

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кваліфікаційна наукова праця
на правах рукопису

БОГУТ ОЛЕГ МИХАЙЛОВИЧ

УДК 004.42

ДИСЕРТАЦІЯ

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ УПРАВЛІННЯ БАГАТОРІВНЕВИМ
НАВЧАННЯМ ТА ПІДБОРОМ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ**

Ф3 “Комп’ютерні науки”
Ф “Інформаційні технології”

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело О. М. Богут

Наукові керівники:
КАСАТКІН Дмитро Юрійович,
кандидат педагогічних наук, доцент
ЮСКОВИЧ-ЖУКОВСЬКА Валентина Іванівна,
кандидат технічних наук, доцент

Київ - 2026

АНОТАЦІЯ

Богут О. М. Інформаційна технологія управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії. Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю F3 “Комп’ютерні науки”. Національний університет біоресурсів та природокористування України, Київ, 2026.

У дисертації досліджено методи та моделі системи управління процесами навчання та рекрутингу в галузі ІТ. Окреслені теоретичні засади формують методологічну основу для створення інформаційної технології, що дозволяє автоматизувати систему навчання за рівнями: Junior, Middle, Senior та навичками: HTML/CSS, JS, фреймворки, архітектура.

Навчальні програми автоматизованих курсів структуровані таким чином, щоб адаптувати навчання під конкретного кандидата на посаду web-програміста та оцінити його скіли. Впровадження розроблених моделей та алгоритмів відбору кандидатів із застосуванням технологій штучного інтелекту у практику сприятиме автоматизації процесів рекрутингу web-програмістів.

Розроблено прототип інтелектуальної інформаційної системи, яка об’єднує: базу даних кандидатів і вакансій, модуль навчання (LMS), модуль оцінювання, модуль рекрутингу.

Досліджено проблему оптимізації процесів відбору кандидатів на вакансії web-програмістів.

Створений прототип інтелектуальної інформаційної системи дозволив оптимізувати процес кваліфікованої оцінки професійних знань, умінь та навичок (hard- та soft-скілів) web-програмістів для підбору на різні посади в ІТ-компаніях з врахуванням професійного грейду.

Експериментальна перевірка показала, що при автоматизованому прийнятті рішень значно зменшується час на обробку анкетних даних, співставлянні навичок кандидатів з вимогами вакансії, зменшується час найму та підвищується точність підбору кандидатів. Це дозволило провести глибокий аналіз порангового

навчання майбутніх web-програмістів, оцінити їх рівень, автоматично відбирати найкращих для вакансій, допомагає ІТ-компанії швидше і ефективніше наймати web-програмістів.

Складність таких задач полягає в тому, що традиційний процес оцінювання та первинного відбору web-програмістів для ІТ-проектів є суб'єктивним, оскільки залежить від прийняття рішення інтерв'ювером під час проведення співбесіди та може відрізнятись від реального когнітивного стану. Процес відбору ґрунтується на багатьох критеріях hard- та soft-скілів, які є різноважливими для різних рівнів професійної кар'єри web-програміста, що визначає складність задачі багатокритеріального вибору та нелінійність пошуку оптимального рішення.

На основі реальної інформативної бази знань професійної підготовки для кожного рівня кваліфікації web-програмістів: trainee-, junior-, middle-, senior-, lead-автоматизовано багаторівневе навчання та процес відбору на посади командної роботи: frontend-, backend- та fullstack-розробки.

Встановлено, що найбільш ефективними підходами для критеріїв оцінки знань web-програміста є матриця вагових коефіцієнтів, матриця компетенцій ІТ-фахівця, метод 180°, математична модель прийняття рішень в управлінні ІТ-персоналом. Досліджено та програмно реалізовано алгоритм, який інтегрує технології штучного інтелекту та машинного навчання, що дає змогу інформаційній системі кваліфіковано оцінювати рівень знань і навичок за відбору перспективних програмістів-початківців та претендентів на вищі посади, а також прогнозувати освітні потреби в їх подальшому навчанні.

Розроблена інтелектуальна інформаційна система працює в двох режимах: режимі багаторівневого навчання (для кожного рівня web-програмістів пропонується свій окремий контент та контроль знань до нього) та режимі відбору на посаду web-програміста (для кожного кандидата на ту чи іншу посаду штучний інтелект визначає персоналізований варіант тестової співбесіди і обґрунтовує висновки за її результатами).

Створена інформаційна web-технологія здатна модернізувати традиційні форми навчання, вдосконалити підвищення кваліфікації програмістів-початківців і

програмістів-практиків різних рангів та процеси рекрутингу в ІТ-компаніях в умовах цифрового суспільства. Реалізована web-технологія дає можливість індивідуалізувати процес навчання та відбору, самостійно переводити навчаючого з одного рівня на інший, рекомендувати чи відмовляти в участі командної роботи над ІТ-проєктом.

Інтелектуальна інформаційна система розроблена на основі технологій генеративного штучного інтелекту OpenAI GPT та складається з набору промптів, підготовлених наборів даних-датасетів, що формуються окремими підсистемами інтелектуальної інформаційної системи та фреймворку LangGraph, який забезпечує зберігання даних тренування діалогу нейронної мережі.

Проведена експериментальна перевірка використання розроблених модулів навчання та відбору web-програмістів підтвердила дидактичну ефективність створеної інформаційної технології.

Запропонована інформаційна технологія управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії ґрунтується на диференційованому підході до кожного кандидата на рангову вакансію в ієрархії ІТ-компанії, а саме: Junior, Middle, Senior, що дозволяє враховувати його набуті скіли та кар'єрний зріст у розробці та підтримці для клієнтів, зокрема іноземних, web-сайтів на CMF Drupal: e-commerce-платформи, корпоративні сайти, web-калькулятори, навігатори, посадкові сторінки, власних модулів та інтеграції Drupal.

Практичне застосування результатів дослідження створює передумови для покращення взаємодії між закладами вищої освіти, що готують фахівців для ІТ-галузі у сфері web-розробки та безпосередньо роботодавцями – ІТ-компаніями, що спеціалізуються на ІТ-проєктах з web-програмування та web-дизайну.

Ключові слова: інтелектуальна інформаційна система, інформаційні технології, штучний інтелект, web-програмісти, ІТ-проєкт, CMF Drupal.

ANNOTATION

Bogut O. M. Information Technology for Managing multilevel training and selection of Web programmers in an IT Company. The Manuscript.

Dissertation submitted for the degree of Doctor of Philosophy in specialty F3 “Computer Science”.

National University of Life and Environmental Sciences of Ukraine, Kyiv, 2026.

The dissertation investigates methods and models of computer-based learning technologies for the recruitment and professional development of web programmers in IT companies. The outlined theoretical principles form a methodological foundation for the development of an intelligent information system that enables the creation of interactive learning environments, learning management systems (LMS), testing tools, and mechanisms for designing individual educational trajectories adapted to the specific knowledge level of an IT specialist. The implementation of the developed models in practice makes it possible to optimize the process of qualified assessment of professional knowledge, abilities, and skills (hard and soft skills) of web programmers for recruitment to various positions in IT companies, taking into account professional grading.

The complexity of such tasks lies in the fact that the traditional process of evaluation and initial selection of web programmers for IT projects is subjective, as it depends on decisions made by interviewers during job interviews and may differ from the actual cognitive state of candidates. The selection process is based on numerous hard and soft skill criteria that have different weights at various stages of a web programmer’s professional career, which determines the complexity of the multi-criteria decision-making problem and the nonlinearity of the search for an optimal solution.

Global trends in the development of innovative computer-based learning technologies, as well as the implementation of content-advanced methods, tools, and forms of professional training for a new generation of IT specialists within the system of anticipatory education supported by advanced software and hardware solutions, have been analyzed.

Based on a real and informative knowledge base for professional training corresponding to each qualification level of web programmers: trainee, junior, middle, senior, and lead multilevel learning and the recruitment process for team-based positions in frontend, backend, and full-stack development have been automated.

It has been established that the most effective approaches for assessing the knowledge of web programmers include a weighting coefficient matrix, an IT specialist competency matrix, the 180-degree evaluation method, and a mathematical decision-making model in IT personnel management. An algorithm integrating artificial intelligence and machine learning technologies has been studied and implemented in software, enabling the information system to competently assess knowledge and skill levels when selecting promising entry-level programmers and candidates for senior positions, as well as to forecast their educational needs for further training.

The developed intelligent information system operates in two modes: a multilevel learning mode (in which each level of web programmers is provided with dedicated content and corresponding knowledge assessment) and a recruitment mode for web programmer positions (in which artificial intelligence generates a personalized test interview for each candidate and provides a substantiated interpretation of the results).

The created web-based information technology is capable of modernizing traditional forms of education, improving advanced training for novice and practicing programmers of various ranks, and enhancing recruitment processes in IT companies in the context of a digital society. The implemented web technology enables the individualization of learning and selection processes, autonomous progression of learners from one level to another, and recommendations for or refusal of participation in team-based work on IT projects.

The intelligent information system is developed on the basis of generative artificial intelligence technologies, specifically OpenAI GPT, and consists of a set of prompts, prepared datasets generated by individual subsystems of the intelligent information system, and the LangGraph framework, which ensures storage and management of neural network dialogue training data.

Experimental validation of the developed learning and recruitment modules for web programmers confirmed the didactic effectiveness of the proposed information technology.

The proposed information technology for managing multilevel training and recruitment of web programmers in IT companies is based on a differentiated approach to each candidate for ranked positions within the IT company hierarchy: Junior, Middle, and Senior which makes it possible to account for acquired skills and career progression in the development and maintenance of client-oriented, including international, web solutions based on the Drupal CMF, such as e-commerce platforms, corporate websites, web calculators, navigators, landing pages, custom modules, and Drupal integrations.

The practical application of the research results creates prerequisites for improving cooperation between higher education institutions that train specialists for the IT industry in the field of web development and direct employers IT companies specializing in web programming and web design projects.

Keywords: intelligent information system, information technologies, artificial intelligence, web programmers, IT project, Drupal CMF.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових виданнях, включених до міжнародних наукометричних баз даних **Web of Science Core Collection** та/або **Scopus**

1. Valentyna Yuskovych-Zhukovska, **Oleg Bogut**, Yuriy Lotyuk, Olga Kravchuk, Olga Rudenko, Halyna Vasylenko. E-Learning in a Postmodern Society. 2022. Vol. 13. URL:

https://www.researchgate.net/publication/359220790_E-Learning_in_a_Postmodern_Society (Богутом О. М. проведено літературний науковий пошук, розроблено концепцію та технічні вимоги до інформаційної системи, деталізовано контекст впливу електронного навчання на підготовку ІТ фахівців з точки зору потреб бізнесу. Юскович-Жуковською В. І. визначено наукову новизну та теоретичні основи дослідження, описано вплив електронного навчання на підготовку фахівців в галузі ІТ. Лотюком Ю. Г., Кравчук О. Руденко О., Василенко Г. проаналізовано сучасні тенденції та перспективи розвитку електронної освіти в постмодерному суспільстві).

Статті у наукових виданнях, включених до Переліку наукових фахових видань України

2. Касаткін Дмитро, Юскович-Жуковська Валентина, **Богут Олег** Особливості оркестрації мультиагентних систем штучного інтелекту для задач автоматизації HR-процесів в ІТ-компаніях. Вісник Хмельницького національного університету. 2026. Серія: Технічні науки. Том 1. (Богутом О. М. запропоновано використання фреймворку LangGraph для гнучкої оркестрації агентів в мультиагентній системі з можливістю побудови вендорно-незалежної агентної системи. Касаткіним Д. Ю. запропоновано використання багатомірної окрестрації із автокорекцією в мультиагентній системі штучного інтелекту для автоматизації задач оцінювання професійних знань ІТ-фахівців в HR-процесах ІТ-компаній. Юскович-Жуковською В. І. запропоновано використання концепції ізольованого агента для процесів незалежної валідації висновків мультиагентної системи).

3. Valentyna Yuskovych-Zhukovska, **Oleg Bogut**. Methods of preventing information leaks caused by the usage of artificial intelligence agent systems. *Наука і техніка сьогодні*. Серія «Техніка»: журнал. 2025. № 12(53). P. 109-118. URL: https://www.researchgate.net/publication/398000295_METHODS_OF_PREVENTING_INFORMATION_LEAKS_CAUSED_BY_THE_USAGE_OF_ARTIFICIAL_INTELLIGENCE_AGENT_SYSTEMS. (Богутом О. М. досліджено методи впливу штучного інтелекту на витoki інформації, спричинені агентними системами штучного інтелекту. Юскович-Жуковською В. І. Формалізовано методи протидії витокам інформації, спричинені агентними системами штучного інтелекту).

4. Юскович-Жуковська В.І., **Богут О. М.** Управління життєвим циклом адміністративного web-сайту з використанням CMF Drupal. Комп'ютерне моделювання: аналіз, управління, оптимізація. 2019. Вип. 2 (6). С.103-109. Copernicus. URL:

http://www.wydawnictwo.wst.pl/uploads/files/21_Yuskovych-Zhukovska_Bogut.pdf
(Богутом О. М. здійснено декомпозицію і деталізацію моделі та формалізації її у відповідності до сучасних стандартів розробки. Юскович-Жуковською запропоновано концептуальну модель управління життєвим циклом адміністративного веб сайту).

5. Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. An intellectual information system for rank-based selection of web programmers. С.11-20 *Вісник Хмельницького національного університету*. 2024. Технічні науки. Том 345 № 6(2), URL: <https://doi.org/10.31891/2307-5732-2024-345-6> (Богутом О. М. розроблено архітектуру та реалізовано дану інформаційну систему для порангового відбору web-програмістів. Юскович-Жуковською В. І. запропоновано концепт інтелектуальної інформаційної системи для порангового відбору web-програмістів).

Статті в наукових виданнях інших держав (в колективних монографіях)

6. Дмитро Касаткін, Валентина Юскович-Жуковська, **Олег Богут**. Інтеграція штучного інтелекту в освітні програми підготовки ІТ-фахівців. *Edukacja i*

społeczeństwo X. Zbiór prac naukowych. Akademia Śląska: Wydawnictwo Wyższej Szkoły. Technicznej w Katowicach, Polska. 2026. P.166-172. URL: <http://www.wydawnictwo.wst.pl/uploads/files/16e64cd03975f31fe0d3a076e523bb73.pdf> (Богутом О. М. сформовано вимоги до використання штучного інтелекту на основі сучасних практичних потреб ІТ-індустрії. Касаткіним Д. Ю. Сформовано концепцію інтеграції штучного інтелекту в навчальний процес для підготовки ІТ-фахівців. Юскович-Жуковською В. І. Формалізовано підходи інтеграції штучного інтелекту в освітні програми підготовки ІТ-фахівців).

7. Yurii Lotiuk, Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. Features of the use of artificial intelligence in scientific research. Education, economy, and AI: multidisciplinary perspectives for a digital future. Monograph. The University of Technology in Katowice Press. 2025. P. 608-620. URL: <http://www.wydawnictwo.wst.pl/uploads/files/42941e01a402e7084c9e0999d9aca888.pdf> (Богутом О. М. запропоновано підходи до використання штучного інтелекту в наукових дослідженнях, на основі практичних підходів в ІТ-компаніях. Юскович-Жуковською В. І. сформовано концепцію використання великих мовних моделей для систематизації наукових даних в наукових дослідженнях. Лотюком Ю. Г. формалізовано підходи до використання штучного інтелекту для пошуку наукових даних в наукових дослідженнях).

8. Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. Electronic educational environment for acquiring professional skills in web developer training. Contemporary technologies and society: innovations, artificial intelligence, and challenges. Collective Scientific Monograph. 2023. P. 140-148. Katowice. URL: <http://www.wydawnictwo.wst.pl/uploads/files/fc72d78bb069f493b87e716a1f31a05b.pdf> (Богутом О. М. визначено ключові фактори впливу, критичні для ефективного навчання ІТ-фахівців для ІТ-бізнесу. Юскович-Жуковською В. І. проведено аналіз факторів впливу електронного навчального оточення на розвиток професійних скілів ІТ-фахівців).

9. Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. Perspective technologies of the cmf drupal for design and development of the websites and web applications. Zeszyty

naukowe wyższej szkoły technicznej w katowicach. 2021. № 13. P. 235-246. URL: http://www.wydawnictwo.wst.pl/uploads/files/21_Yuskovych-Zhukovska_Bogut.pdf (Богутом О. М. досліджено перспективи практичного використання CMF Drupal як комерційного фреймворку для розробки web-сайтів та web-додатків. Юскович-Жуковською В. І. проаналізовано сильні і слабкі сторони CMF Drupal як сучасного фреймворку для web-розробки).

10. Valentyna Yuskovych-Zhukovska, **Oleg Bogut**. The use of information technologies in recruitment processes. Current challenges and new opportunities in science and education: proceedings of the international scientific conference. 2024, June 29. Nottingham. P. 51-53 URL: <https://researcheurope.org/wp-content/uploads/2024/07/re-29.06.2024.pdf> (Богутом О. М. здійснено аналіз метрик підвищення ефективності застосування іноваційних технологій в процесах рекрутингу. Юскович-Жуковською В. І. проведено аналіз іноваційних технологій в процесах рекрутингу).

11. **Oleg Bogut**, Valentyna Yuskovych-Zhukovska. The impact of artificial intelligence on the efficiency of IT project management. Innovation-driven development in education, digital economy, and applied technologies. 2025. The University of Technology in Katowice Press. P. 228-232. URL: <http://www.wydawnictwo.wst.pl/uploads/files/e28c93cb04ed5369e1c94cb14399b880.pdf> (Богутом О. М. досліджено та виділено фактори впливу штучного інтелекту на ефективність процесів проектного менеджменту в ІТ-компаніях. Юскович-Жуковською В. І. здійснено дослідження та оцінку ефективності факторів впливу штучного інтелекту на ефективність процесів проектного менеджменту в ІТ-компаніях).

12. **Oleg Bogut**, Valentyna Yuskovych-Zhukovska. Peculiarities of using artificial intelligence in the processes of training and evaluation of web programmers in it companies. Exploring the digital landscape: interdisciplinary perspectives. 2024. The University of Technology in Katowice Press. P. 830-842. URL: http://repositc.nuczu.edu.ua/bitstream/123456789/20612/1/Monograph_Poland_KondratenkoOM_LytvyenkoOO_2024%2B%2B%2B.pdf (Богутом О. М. досліджено

ключові компоненти процесів професійного навчання в ІТ-компаніях, що підлягають ефективній автоматизації засобами штучного інтелекту, Valentyna Yuskovich-Zhukovska, В. І. досліджено ключові компоненти процесів оцінювання професійних знань, умінь, та навичок в ІТ-компаніях, що підлягають ефективній автоматизації засобами штучного інтелекту).

13. Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. Features of using artificial intelligence to enhance the qualifications of web developers. Education and economy in the digital age. 2024. The University of Technology in Katowice Press. P. 166-171. URL:

<https://library.megu.edu.ua:9443/jspui/bitstream/123456789/4831/1/%d0%91%d0%be%d0%b3%d1%83%d1%82%20%d0%96%d1%83%d0%ba%d0%be%d0%b2%d1%81%d1%8c%d0%ba%d0%b0%20%d0%9f%d0%be%d0%bb%d1%8c%d1%89%d0%b0%2024.pdf> (Богут О. М. проаналізовано та запропоновано новітні методи аналізу даних метрик в контексті HR-процесів ІТ-компаній. Юскович-Жуковською В. І. досліджено метрики ефективності оцінювання професійних навичок web-розробників в ІТ-компаніях засобами штучного інтелекту).

14. Digital transformation in contemporary society. Edited by Valentyna Yuskovich-Zhukovska, **Oleg Bogut**. 2024. Wydawnictwo Wyższej Szkoły Technicznej w Katowicach. URL:

<http://www.wydawnictwo.wst.pl/uploads/files/30ca7d77debc40d97016d5c12fc0770c.pdf> (Богут О. М. та Юскович-Жуковською В. І. здійснено наукове редагування колективної наукової монографії).

15. Contemporary technologies and society: innovations, artificial intelligence, and challenges. Edited by Valentyna Yuskovich-Zhukovska and **Oleg Bogut**. 2023. Katowice: Wydawnictwo Wyższej Szkoły Technicznej w Katowicach. P. 13-21. URL: http://www.wydawnictwo.wst.pl/oferta_wydawnicza_oraz_zakup_publicacji/wydawnictwa/contemporary_technologies_and_society_innovations_artificial_intelligence_and_challenges/124 (Богут О. М. та Юскович-Жуковською В. І. здійснено наукове редагування колективної наукової монографії).

Наукові статті у нефармових виданнях

16. Юскович-Жуковська В. І., **Богут О. М.** Перспективи автоматизованого управління персоналом в ІТ-компанії. Математика. Інформаційні технології. Освіта. Збірник статей. 2024. № 11/ 2024. С. 171-180. URL: <https://me-qr.com/6g4p522b> (Богут О. М. проаналізовано перспективи та метрики ефективності комплексної автоматизації процесів рекрутингу та HR-процесів. Юскович-Жуковською В. І. проаналізовано сучасний стан ринку технологій автоматизації процесів управління персоналом).

17. Ю. Г. Лотюк, **О. М. Богут.** Формування у студентів умінь та навичок проектування та моделювання комп'ютерних мереж. Науковий часопис НПУ імені М.П. Драгоманова. 2007. Серія №2. Комп'ютерно-орієнтовані системи навчання: Збірник наукових праць. С. 59-63. URL: <https://enpuirb.udu.edu.ua/server/api/core/bitstreams/0e29c358-dfb2-49b9-95a8-d531605e26e7/content> (Богут О. М. запропоновано використання програмного засобу NetCracker для імітаційного моделювання процесів функціонування комп'ютерних мереж в навчальному процесі. Лотюком Ю. Г. запропонована концепція використання засобів імітаційного візуального моделювання топологій комп'ютерних мереж в навчальному процесі).

Тези наукових доповідей

18. Юскович-Жуковська В.І., **Богут О.М.** Концепція людино-машинної взаємодії в контексті штучного інтелекту. Міжнародні тенденції та перспективи розвитку в освіті та науці в умовах глобалізації. Міжнародна науково.-практична конференція до 100-річчя від дня народження акад. С.Дем'янчука. 2025. С. 162-165. Рівне. URL: https://www.megu.edu.ua/sites/default/files/2025-10/ilovepdf_merged.pdf (Богут О. М. досліджено технології людино-машинної взаємодії в контексті діалогових систем штучного інтелекту. Юскович-Жуковською В. І. сформовано концепцію людино-машинної взаємодії в контексті діалогових систем штучного інтелекту).

19. Юскович-Жуковська В.І., **Богут О.М.** Застосування методів централізованої та децентралізованої оркестрації агентного штучного інтелекту до задач автоматизації процесів управління персоналом ІТ-компаній. Матеріали ІХ Міжнародної науково-технічної конференції Комп'ютерне моделювання та оптимізація складних систем. 5-7 листопада, 2025. Дніпро, С. 231-232. URL: https://udhtu.edu.ua/wp-content/uploads/2025/11/zbirnyk-kmoss-2025_compressed.pdf (Богутом О. М. проведено аналіз методів централізованої та децентралізованої оркестрації агентних систем штучного інтелекту в навчальному процесі підготовки ІТ-фахівців. Юскович-Жуковською В. І. запропоновано методи застосування централізованої та децентралізованої оркестрації агентного штучного інтелекту до задач автоматизації процесів управління персоналом ІТ-компаній).

20. Юскович-Жуковська В. І., **Богут О. М.** Використання оркестрованого агентного штучного інтелекту для оцінювання знань та умінь web-програмістів в іт-компаніях. Матеріали V Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації». 25-26 вересня, 2025. Одеса. С. 502-504. URL: https://ontu.edu.ua/download/konfi/2025/Abstracts_Computer_games_and_multimedia_innovative_approach_electronic_communication_2025.pdf (Богутом О. М. досліджено особливості використання оркестрованого агентного штучного інтелекту в застосуванні до задач HR-процесів в ІТ-компаніях. Юскович-Жуковською В. І. досліджено технології оркестрації агентного штучного інтелекту).

21. **Богут О. М.**, Мотриниць Я. Я. Технології агентного штучного інтелекту та їх використання до задач побудови структурованих вибірок даних на основі семантичних запитів. Матеріали V Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації». 25-26 вересня, 2025. Одеса. С. 476-478. URL: https://ontu.edu.ua/download/konfi/2025/Abstracts_Computer_games_and_multimedia_innovative_approach_electronic_communication_2025.pdf (Богутом О. М.

досліджено особливості використання великих мовних моделей до задач семантичного аналізу даних. Мотриниць Я. Я. досліджено та оцінено ефективність використання оркестрованого агентного штучного інтелекту до задач семантичного аналізу даних).

22. Oleg Bogut. The new era of technologies: how AI is changing a modern battlefield. Beyond Europe: Artificial Intelligence in International Relations and Communication - Opportunities and Challenges. 13 грудня, 2024. Poznań. P. 19-20
URL:

<https://beyondeurope.web.amu.edu.pl/wp-content/uploads/2024/12/Copy-of-Beyond-Europe-2024-8.pdf>

23. Юскович-Жуковська В. І., **Богут О. М.** Особливості використання великих мовних моделей в інтелектуальних інформаційних системах для підтримки прийняття рішень. Матеріали XVII міжнародної науково-практичної конференції. 31 жовтня - 1 листопада, 2024. Одеса. С. 702-704. (Богутом О. М. проведено порівняльний аналіз мовних моделей від різних виробників у застосуванні до задач підтримки прийняття рішень. Юскович-Жуковською В. І. здійснено аналіз сучасних великих мовних моделей, та їх особливостей). URL:
<https://library.megu.edu.ua:9443/jspui/bitstream/123456789/4853/1/%d0%ae%d1%81%d0%ba%d0%be%d0%b2%d0%b8%d1%87%2c%20%d0%91%d0%be%d0%b3%d1%83%d1%82%20%d0%9e%d0%b4%d0%b5%d1%81%d0%b0%20%d0%9e%d1%81%d0%be%d0%b1%d0%bb%d0%b8%d0%b2%d0%be%d1%81%d1%82%d1%96.pdf>

24. В. Юскович-Жуковська, Ю. Лотюк, **О. Богут**, А.Грисюк. Актуальний стан та перспективи розвитку інтернету речей. Актуальні питання та перспективи інноваційного розвитку науки та освіти в умовах євроінтеграції. 14 листопада, 2024. Рівне. С. 137-140. URL:
<https://dspace.megu.edu.ua:8443/jspui/handle/123456789/5206> (О. М. Богутом досліджено особливості web-протоколів для технологій інтернету-речей. Юскович-Жуковською В. І. досліджено сучасний стан ринку інтернету речей, Лотюком Ю. Г. досліджено сучасний стан мережевих технологій інтернету речей. Грисюком А. досліджено сучасні екосистеми інтернету речей).

25. Юскович-Жуковська В. І., **Богут О. М.** Використання штучного інтелекту в задачах підбору Web-програмістів в ІТ-компаніях. Матеріали XIII Міжнародної науково-практичної конференції Математика. Інформаційні технології. Освіта. 31 травня -2 червня, 2024. Луцьк-Світязь. С. 193-194. URL: <https://library.megu.edu.ua:9443/jspui/bitstream/123456789/4849/1/%D0%96%D1%83%D0%BA%D0%BE%D0%B2%D1%81%D1%8C%D0%BA%D0%B0%20%D0%91%D0%BE%D0%B3%D1%83%D1%82%20%D0%9B%D1%83%D1%86%D1%8C%D0%BA%20%D1%82%D0%B5%D0%B7%D0%B8%202024%20%D0%92%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%88%D1%82%D1%83%D1%87%D0%BD%D0%BE%D0%B3%D0%BE.pdf> (Богутом О. М. досліджено особливості використання штучного інтелекту для автоматизації бізнес-процесів підбору web-програмістів в ІТ-компаніях. Юскович-Жуковською В. І. систематизовано і формалізовано бізнес-процеси підбору web-програмістів в ІТ-компаніях).

26. Богут О.М. Аналітика ринку персоналу в галузі ІТ. Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. 20-21 квітня, 2023. Одеса. С. 203-204. URL: <https://card-file.ontu.edu.ua/server/api/core/bitstreams/125b3eff-8c1e-40d5-ba1f-6f6d916ba093/content>

27. Юскович-Жуковська В. І., **Богут О. М.**, Цифрові товари для online продажу. Міжнародна науково-практична конференція до 30-річчя Приватного вищого навчального закладу «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука» «Інноваційні дослідження та перспективи розвитку науки і техніки у ххі столітті». 19 жовтня, 2023. URL: https://dspace.megu.edu.ua:8443/jspui/bitstream/123456789/4453/1/%D0%B7%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA_%D1%82%D0%B5%D0%B7_%D0%BD%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B8%D1%85_%D0%B4%D0%BE%D0%BF%D0%BE%D0%B2%D1%96%D0%B4%D0%B5%D0%B9_%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0_3-218-220.pdf (Богутом О. М. досліджено метрики оцінювання та конвертації продажів в галузі

інтернет-маркетингу. Юскович-Жуковською В. І. Досліджено особливості сучасних технологій електронної комерції).

28. **Богут О.М.,** Юскович-Жуковська В.І. Інтелектуальна інформаційна система для автоматизації прийняття рішень в ІТ-компанії. Матеріали VIII Міжнародної науково-технічної конференції. Комп'ютерне моделювання та оптимізація складних систем. 1-3 листопада, 2023. Дніпро. С. 140-141. URL: <https://udhtu.edu.ua/wp-content/uploads/2023/11/zbirnyk-tez-kmoss-2023.pdf> (Богутом О.М. досліджено бізнес-процеси ІТ-компаній, що підлягають автоматизації засобами штучного інтелекту. Юскович-Жуковською В.І. проведено аналіз рішень і технологій штучного інтелекту для автоматизації прийняття рішень в ІТ-компанії).

29. **Богут О.М.,** Юскович-Жуковська В.І. Інтеграція штучного інтелекту в освітній процес. Матеріали III Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів. 28-29 жовтня, 2023. Одеса. С. 28-29. URL: <https://ontu.edu.ua/download/konfi/2023/Abstracts-Computer-games-and-multimedia-as-an-innovative-approach-to-electronic-communication-23.pdf> (Богутом О.М. Досліджено перспективи інтеграції штучного інтелекту в освітній процес. Юскович-Жуковською В.І. досліджено ключові напрямки використання штучного інтелекту в освітньому процесі).

30. В. І. Юскович-Жуковська, **О. М. Богут.** Особливості захисту web-сайтів. Кібербезпека в сучасному світі: матеріали Всеукраїнської науково-практичної конференції. 29 листопада, 2019. Одеса. ВД Гельветика. С. 82-85. (О. М. Богутом проаналізовано сучасні методи аналізу загроз функціонування web-сайтів. В. І. Юскович-Жуковською проаналізовано сучасні підходи до організації захисту web-сайтів).

31. Алгоритм організації функціонування адміністративного сайту в WWW / В. І. Юскович-Жуковська, **О. М. Богут.** Матеріали V Міжнародної науково-технічної конференції «Комп'ютерне моделювання та оптимізація складних систем». 2019. Дніпро. С. 233-235. (О. М. Богутом описано прикладну

архітектуру адміністративного сайту. В. І. Юскович-Жуковською сформовано алгоритм функціонування адміністративного сайту).

32. Лотюк Ю. Г., **Богут О. М.** Формування у студентів умінь та навичок проєктування та моделювання комп'ютерних мереж. Матеріали V Міжнародної науково-технічної конференції “Комп'ютерні технології в будівництві”. 18-21 вересня, 2007. Київ-Севастополь. URL: <https://enpuirb.udu.edu.ua/server/api/core/bitstreams/0e29c358-dfb2-49b9-95a8-d531605e26e7/content> (Богут О.М. запропонував використання програмного продукту NetCracker для візуального імітаційного моделювання комп'ютерних мереж. Лотюк Ю.Г. Запропонував методи умінь та навичок проєктування та моделювання комп'ютерних мереж засобами технологій імітаційного моделювання комп'ютерних мереж).

33. Лотюк Ю. Г., **Богут О. М.** Проблеми підготовки фахівців у сфері мережевих технологій. Новітні комп'ютерні технології: Матеріали VII Міжнародної науково-технічної конференції. 15-18 вересня, 2009. Київ-Севастополь. URL: https://elibrary.kdpu.edu.ua/bitstream/0564/609/1/sev_sbor.pdf (Богутом О.М. запропоновано методи підвищення ефективності підготовки фахівців у сфері мережевих технологій за допомогою збільшення рівня практичних навичок роботи з мережевими технологіями. Лотюк Ю.Г. Виділив ключові проблеми підготовки фахівців у сфері мережевих технологій)

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ТА НЕОБХІДНІСТЬ ЇХ ЗАСТОСУВАННЯ ПРИ ВІДБОРІ НА ПОСАДИ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ	19
1.1. Аналіз існуючих інтелектуальних інформаційних системи для відбору програмістів в ІТ-компанія	19
1.2. Аналіз основних напрямів застосування інтелектуальних інформаційних систем у цифровому суспільстві	30
1.3. Дослідження методології створення інформаційної системи	35
1.4. Проблеми та перспективи підготовки і відбору web-програмістів в ІТ-компаніях	39
1.5. Постановка задачі на основі аналізу процесів автоматизації ІТ-рекрутингу Web-програмістів	50
Висновки до першого розділу	54
РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ОСНОВ WEB-РОЗРОБКИ	56
2.1. Зміст навчання програмістів	56
2.2. Загальні принципи web-розробки, методологічні та наукові аспекти знань для відбору web-програмістів	61
2.3. Методика побудови інтелектуальної інформаційної системи для підготовки та відбору web-програмістів в ІТ-компанії	68
2.4. Багаторівнева перевірка знань з підбору web-програмістів в ІТ-компанії	75
2.5. проект інтелектуальної інформаційної системи, необхідної для підготовки та відбору web-програмістів	86
Висновки до другого розділу	92
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ БАГАТОРІВНЕВИМ НАВЧАННЯМ ТА ПІДБОРОМ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ	94

	3
3.1. Електронний навчальний курс “WEB-програмування”	94
3.2. Інтелектуальна система порангового оцінювання знань web-програмістів	105
3.3. Структура інтелектуальної інформаційної системи з підготовки та підбору web-програмістів в ІТ-компанії	120
Висновки до третього розділу	126
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ БАГАТОРІВНЕВОГО НАВЧАННЯ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ, ЇЇ ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА	128
4.1. Методика проведення навчання та відбору програмістів в ІТ-компанії за допомогою інтелектуальної інформаційної системи	128
4.2. Постановка дослідних задач	132
4.3. Методика проведення експерименту з оцінювання знань здобувачів із застосуванням інтелектуальної інформаційної системи	133
4.4. Результати експериментів	144
Висновки до четвертого розділу	156
ВИСНОВКИ	158
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	161
ДОДАТКИ	
ДОДАТОК А. Стандарти бекенд та фронтенд розробки	182
ДОДАТОК Б. Скіл-матриця для ролі Drupal Backend Developer	185
ДОДАТОК В. Матриця вагових коефіцієнтів для скіл матриці	193
ДОДАТОК Г. Зміст електронного курсу з web-програмування	194
ДОДАТОК Д. Робоча програма автоматизованого курсу з підготовки web-програміста рівня Trainee	198
ДОДАТОК Е. Лістинг вихідного коду	199
ДОДАТОК Є. Довідка про впровадження в Приватному вищому навчальному закладі “Міжнародний Економіко-Гуманітарний Університет”	215
ДОДАТОК Ж. Довідка про впровадження в компанії AnyforSoft	216
ДОДАТОК З. Довідка про впровадження в компанії ТОВ Смарт Кінг	217

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CRM - customer relationship management

ERP - enterprise resource planning

CMF - content management framework

IT - інформаційні технології

IS - інформаційна система

ІС - інтелектуальна інформаційна система

HRIS - human resource information system

ІІТ - інтелектуальні інформаційні технології

HR - human relations

PDP - персональний план розвитку

PPR - поточне оцінювання

CV - реюме

ВСТУП

Актуальність теми. Сучасна світова цифрова економіка базується на знаннях та цінностях інформаційного суспільства. Стрімкий розвиток інформаційних технологій потребує ІТ-фахівців в усі галузі економічної діяльності.

Інформаційні технології протягом свого розвитку стрімко змінювали суспільство, бізнес-процеси та повсякденне життя. Вони стали доступними широкому загалу, що сприяло глобалізації інформації та появи нових форм людино-машинної взаємодії. Сучасні ІТ-технології - від хмарних сервісів до штучного інтелекту підвищують ефективність бізнесу, створюють нові галузі, професії та способи творчого самовираження у цифровому суспільстві.

Дослідження світового лідера в галузі аналітики - компанії Gartner [1] показує, що на професії найближчого майбутнього буде мати вплив штучний інтелект, що в свою чергу змінює вимоги як до самих професій, так і вимоги щодо інтенсивності та ефективності навчання та оцінювання фахівців. Тренди щодо автоматизації та цифровізації рутинних виробничих процесів за допомогою штучного інтелекту визначають зниження попиту до професій, які потребують низької кваліфікації та виконання рутинних операцій, і водночас, підвищують попит до висококваліфікованих фахівців в галузі ІТ, які можуть забезпечувати впровадження рішень в галузі діджиталізації та інформатизації бізнесу.

Відповідно до досліджень Всесвітнього економічного форуму [2], близько 60% працівників потребуватимуть певної форми навчання протягом наступних чотирьох років, щоб залишатися актуальними на своїх посадах або адаптуватися до нових. За даними експертів майбутнє робоче місце вимагатиме поєднання аналітичних здібностей, креативності, адаптивності, логічного мислення та безперервного навчання. Таким чином ІТ-фахівці, зокрема web-програмісти, мають бути готовими безперервно навчатися новітнім технологіям протягом свого життя.

Однією з пріоритетних цілей Стратегії цифрового розвитку інноваційної діяльності України [3] на період до 2030 року є розвиток інноваційної екосистеми в закладах освіти.

Згідно з Концепцією розвитку штучного інтелекту в Україні [4], попит на ринку праці відображає те, що сучасна система освіти повинна набагато якісніше готувати конкурентоздатних ІТ-фахівців, зокрема у галузі штучного інтелекту.

Аналіз Міністерством освіти і науки України [5] проблемних питань підготовки кадрів для ІТ-галузі свідчить про відсутність достатньої взаємодії ІТ-освіти і ІТ-ринку праці, що призводить до виникнення проблем з працевлаштуванням випускників ІТ-спеціальностей та тривалого строку адаптації випускників на робочому місці.

Дослідження у сфері партнерських відносин проводили закордонні науковці, зокрема: С. Брю, М. Брюн, С. Вітте, Б. Джексон, Дж. Кейнс, Дж. Мілль, А. Пейн, М. Портер, Дж. Шет, Ф. Штейн та ін.

У дослідженні [5] проведеному науковцями Japaridze, D., Maghlakelidze, S., Shaverdashvili, E., Tutberidze, D. проаналізована роль вищої освіти у зменшенні невідповідності між навичками випускників та очікуваннями роботодавців.

У роботі вітчизняних науковиці Тетяни Хархаліс [6] досліджено теоретичні та практичні аспекти розвитку співпраці між ІТ-підприємствами та закладами вищої освіти у процесі підготовки висококваліфікованих ІТ-фахівців.

У дослідженні, проведеному вітчизняною науковицею Маріанною Швардак [7] здійснено теоретичне обґрунтування та визначено основні напрями вдосконалення професійної підготовки майбутніх ІТ-фахівців у закладах фахової передвищої освіти в умовах цифровізації суспільства, трансформацій освітнього простору та зростаючих вимог ІТ-ринку праці.

У роботі Р. Н. Кветного [8] розглянуто питання ефективної організації університетської освіти в області ІТ, яка залежить, зокрема, від залучення до цього ІТ-компаній, що працюють на світовому ринку.

До сьогодні трикутник «випускник - роботодавець - університет» не є єдиною екосистемою, оскільки кожен діє окремо, на власний розсуд, без

комплексної координації. Проблема «skills mismatch», тобто невідповідність навичок між тим, чому навчають в університетах, і тим, чого насправді потребує сучасний ІТ-ринок праці, має системний характер.

Університетські ІТ-програми мають теоретичний характер, тоді як ІТ-компанії шукають випускників із практичним досвідом. Університети у багатьох країнах не встигають за швидкими змінами в ІТ-галузі та не забезпечують практичної готовності.

Університети зазвичай продовжують проводити підготовку здобувачів вищої освіти в ІТ-галузі за програмами, які відстають від реалій ІТ-ринку на 5–10 років. Це пояснюється швидким технологічним прогресом, за яким навчальні плани не встигають оновлюватися. Як наслідок, виникають дисбаланси на ринку праці: перенасичення ІТ-фахівців у одних сферах при гострому дефіциті кадрів в інших, критично важливих для економіки України. Тому важливою метою є побудова сталого «win-win» – партнерства зі стейкхолдерами, коли бізнес інвестує в освітні програми (через гранти, обладнання, експертизу), а університети готують для нього потрібних ІТ-фахівців [9].

Сьогодні в усьому світі в умовах стрімких змін на ІТ-ринку праці та розвитку технологій штучного інтелекту додатковим ефективним інструментом для оперативної перепідготовки та безперервного навчання ІТ-фахівців розглядаються мікрокваліфікації.

Успіх мікрокваліфікацій значною мірою залежить від впровадження моделі «соціального замовлення» від бізнесу - коли ІТ-компанії замовляють у закладів вищої освіти підготовку ІТ-фахівців з конкретними soft skills і hard skills та частково фінансують цей процес.

Дослідження [10-15] стверджують, що проблеми з точки зору невідповідності між навичками випускників та очікуваннями роботодавців є суттєвими. тому актуальним є створення інтелектуальної інформаційної системи, роль якої - зменшити рівень невідповідності між навичками випускників та очікуваннями роботодавців в галузі web-розробки.

Не дивлячись на те, що в Україні вже існує успішна практика взаємодії закладів вищої освіти з провідними ІТ-компаніями: EPAM, SoftServe, GlobalLogic, Genesis, що сприяє підвищенню якості підготовки ІТ-кадрів, з іншої сторони, наявний низький рівень інтеграції між ІТ-компаніями і закладами освіти створює дефіцит практичних навичок у студентів [16].

Виходячи із вищеописаного, перед нами постає задача знаходження механізмів та засобів для побудови індивідуального плану навчання здобувача вищої освіти, вирішити яку необхідно для задоволення потреб як для самого закладу так і для здобувача.

Попри значну кількість досліджень, присвячених аналізу взаємовідносин ЗВО з ІТ-компаніями, залишаються невирішені проблеми. Зокрема, потребують додаткового вивчення особливості підготовки web-програмістів.

Визначимо основні проблеми, які виникають на шляху до вирішення такої задачі.

Метою спеціальної професійної освіти з web-програмування та web-розробки є формування в майбутнього web-програміста та web-розробника цифрових професійних знань, умінь та навичок, необхідних для постановки та розв'язання задач з проєктування, реалізації та підтримки web-сайтів, додатків та сервісів.

Управління навчанням та відбором програмістів є важливим аспектом для ІТ-бізнесу, оскільки ефективні стратегії управління персоналом служать вирішальним чинником у досягненні успіху в динамічному та вимогливому ІТ-секторі [18, 19].

Управління ІТ-проєктами в ІТ-компаніях вимагає врахування особливостей діяльності в галузі інформаційних технологій [20]. Це досягається наступним неперервним циклом:

- цілеве управління[21];
- проведення індивідуальних інтерв'ю[22, 23];
- технічне тестування[24];
- оцінка діяльності за професійними результатами [25].

Дані методи взаємодоповнюють один одного та разом формують комплексний підхід до управління та оцінки програмістів в ІТ-компаніях.

Плани персонального розвитку для фахівців ІТ-компаній повинні враховувати наступні критерії:

- адаптацію до швидко змінюваного ІТ-сектору [26];
- максимізацію потенціалу ІТ-співробітника [27, 28];
- зниження рівня оборотності ІТ-персоналу [29];
- стимулювання інновацій [30];
- ефективність управління ресурсами [31];
- підвищення мотивації.

Таким чином, персональний розвиток програмістів відіграє стратегічну роль у досягненні бізнес-цілей ІТ-компанії. Він є інструментом, який сприяє ефективності управління проектними командами, враховуючи особливості та вимоги сучасного ІТ-сектору [32].

Окрім задач оптимального підбору, управління та розвитку програмістів в галузі ІТ важливою задачею є проведення спеціалізованого навчання та оптимальний підбір проектних команд. Ефективність функціонування проектних команд оцінюється за наступними критеріями:

- висока продуктивність;
- синергетичний ефект;
- скорочення часу реалізації ІТ-проекту;
- висока якість ІТ-продукту.

В процесі функціонування проектних команд виникають ризики:

- неоптимальний підбір;
- високий рівень обертаності;
- відсутність синергії;
- перевантаження ресурсів.

Таким чином, розроблена інформаційна технологія враховує наведені критерії для оптимального підбору проектних команд у галузі ІТ [33]. Запропоновано новий підхід до навчання та відбору web-програмістів в

ІТ-компаніях, що базується на технологіях штучного інтелекту та нейронних мереж.

Новий формат короткострокових програм з підготовки, перепідготовки та відбору web-програмістів, які відповідають запитам ІТ-ринку праці, було розроблено, протестовано та впроваджено в ІТ-компаніях AnyforSoft та ТОВ Смарт Кінг та приватному вищому навчальному закладі «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука».

Зв'язок роботи з науковими програмами, планами, темами, грантами.

Тема дисертації “Інформаційна технологія управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії” безпосередньо пов'язана з пріоритетними напрямками державної політики України у сфері цифровізації освіти та розбудови інформаційного суспільства. Дослідження узгоджується із завданнями Стратегії цифрового розвитку інноваційної діяльності України на період до 2030 року, Концепцією цифрової трансформації освіти і науки на період до 2026 року, Концепцією розвитку штучного інтелекту в Україні.

Тема дисертації відповідає науковим напрямкам факультету інформаційних технологій та кафедри Інформаційних систем і технологій Національного університету біоресурсів і природокористування України та науковим напрямкам факультету кібернетики та кафедри Інформаційних систем та обчислювальних методів Приватного вищого навчального закладу “Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, які охоплюють дослідження у галузі інтелектуальних інформаційних систем, обробки великих масивів даних, машинного навчання, систем підтримки прийняття рішень та інформатизації навчального процесу.

Робота також узгоджується із загальною дослідницькою стратегією університетів, спрямованою на цифрову трансформацію освітньої діяльності та використання сучасних інформаційних технологій для підготовки висококваліфікованих ІТ-фахівців.

Це підтверджується актами про впровадження у виробництво результатів дисертації на тему: «Інформаційна технологія управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії», де стверджується, що результати дисертації, як новий формат короткострокових програм з підготовки, перепідготовки та відбору web-програмістів, які відповідають запитам ІТ-ринку праці, було протестовано та впроваджено в ІТ-компаніях AnyforSoft і ТОВ Смарт Кінг та приватному вищому навчальному закладі «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука». Дисертація зв'язана із НДР державний реєстраційний номер 0118U003104) «Створення та впровадження автоматизованої системи керування процесом навчання в електронному освітньому просторі».

Запропонована інтелектуальна інформаційна система реалізована з використанням хмарної екосистеми Amazon AWS. Реалізація здійснена на мовах програмування PHP, Python, JavaScript та з використанням фреймворків Django, Drupal, ReactJS.

Об'єктом дослідження є процеси підбору програмістів в ІТ-компаніях, зокрема web-програмістів на прикладі технологічного напрямку бекенд розробки з використанням CMF Drupal.

Предметом дослідження є новітні інформаційні технології, зокрема інтелектуальні інформаційні системи та методи підтримки прийняття рішень в процесах управління ІТ-проектами в умовах невизначеності з використанням нейронних мереж та штучного інтелекту.

Мета і завдання дослідження. Метою дисертаційної роботи є розробка прикладних та інструментальних засобів інформаційних технологій навчання, орієнтованих на розв'язання задач, призначених для інтенсифікації підготовки та відбору web-програмістів в ІТ-компаніях. При проведенні досліджень використовувались методи штучного інтелекту та нейронних мереж; методологія та програмно-технічні засоби, призначені для інтенсифікації процесу навчання та контролю знань web-програмістів.

Станом на сьогоднішній день існує цілий ряд невирішених питань, які складають науково-методичну проблему в підготовці та відборі в ІТ-компанії ІТ-фахівців, а саме web-програмістів, особливо у зв'язку з використанням новітніх інформаційних технологій.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

- 1) проаналізувати сучасні моделі інтелектуальних інформаційних систем, методи та технології штучного інтелекту з метою виявлення особливостей і проблематики їх застосування у процесі інтенсифікації підготовки та відбору web-програмістів;
- 2) розробити короткостроковий інтенсивний електронний навчальний курс з web-програмування, що складається з лекційних, практичних та лабораторних занять;
- 3) розробити автоматизовану систему рангового оцінювання та контролю знань web-програмістів на посади: trainee, junior, middle, senior, lead ;
- 4) розробити алгоритм та автоматизовану систему прийняття рішень при формуванні проєктних ІТ-команд;
- 5) розробити інтелектуальну інформаційну систему для підготовки та відбору web-програмістів;
- 6) розробити модель інформаційної технології управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії, здійснити експериментальну перевірку ефективності розроблених програмних модулів та на основі отриманих результатів сформулювати рекомендації щодо вдосконалення застосованих методів та алгоритмів.

Методи дослідження. В дисертаційній роботі було використано методи аналізу даних, моделювання процесів підтримки прийняття рішень, математичне моделювання та оцінювання в умовах невизначеності, методів теорії графів та теорії множин, методи моделювання бізнес-процесів, методи стохастичного моделювання, технології нейронних мереж та штучного інтелекту.

Інформаційну базу дисертаційного дослідження становлять нормативно-правові акти та професійні стандарти у сфері інформаційних

технологій, наукові праці вітчизняних і зарубіжних учених з проблем комп'ютерних наук, навчально-методичні матеріали, документація щодо функціонування наявних робочих процесів, а також емпіричні результати власних досліджень моделей, методів і типів інформаційних технологій, що застосовуються для розробки інтелектуальних інформаційних систем.

Методологічну основу PhD-дисертації становить сукупність фундаментальних і загальнонаукових принципів наукового пізнання, а також термінологічний, функціональний, системний і когнітивний підходи, реалізовані з використанням методів математичного та імітаційного моделювання.

Теоретичною основою дисертаційного дослідження є модель інформаційної технології управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії.

Наукова новизна отриманих результатів. На основі проведених теоретичних та практичних досліджень було автоматизовано задачу багаторівневого навчання та прийняття рішень з підбору web-програмістів ІТ-компаній, які спеціалізуються на розробці ІТ-проектів. Автоматизовано первинне та поточне оцінювання рангового рівня знань web-програмістів, автоматизовано формування персональних планів професійного розвитку web-програмістів, проведено автоматизацію підбору web-розробників при формуванні проектних команд та їх рангів в команді. При цьому у дисертаційній роботі отримано такі наукові результати:

вперше:

1) розроблено інтелектуальну інформаційну систему з використанням технології агентного штучного інтелекту LangGraph для аналізу даних професійних компетенцій web-програмістів, динаміки результатів первинного та проміжного оцінювання та автоматизованого формування персональних планів їх професійного ІТ-розвитку;

2) застосовано технологію агентного штучного інтелекту LangGraph для ІТ-проектів з вирішення багатокритеріальної задачі оптимізації по формуванню проектних команд;

3) застосовано в розробленій інтелектуальній інформаційній системі технологію агентного штучного інтелекту LangGraph для аналізу даних професійних компетенцій web-програмістів в процесі багаторівневого навчання, підготовки та відбору web-програмістів в ІТ-компанії; оцінки динаміки результатів первинного та проміжного оцінювання web-програмістів та автоматизованого формування персональних планів їх професійного ІТ-розвитку;

4) застосовано технологію агентного штучного інтелекту LangGraph для ІТ-проектів з вирішення багатокритеріальної задачі оптимізації по формуванню проектних команд;

5) розроблено матрицю hard-скілів для профілю Drupal-розробників рівня trainee, junior, middle, senior, lead;

6) застосовано матрицю вагових коефіцієнтів в якості цифрового засобу визначення важливості скілів та компетенцій для забезпечення ефективного прийняття рішень щодо рівня знань, умінь та навичок в розрізі грейд / компетенція та диференційованої оцінки відповідності до тайтлу;

7) удосконалено матрицю компетенцій ІТ-фахівця шляхом введення ранжування компетенцій web-програмістів в розрізі грейд / компетенція;

8) удосконалено математичну модель прийняття рішень з підбору програмістів ІТ-компанії для використання матриці вагових коефіцієнтів;

9) отримали подальший розвиток методи оцінювання ІТ-персоналу, зокрема web-програмістів, модифіковано метод 180° для його використання з вдосконаленою матрицею компетенцій та матрицею вагових коефіцієнтів.

Практичне значення отриманих результатів. Практичне значення полягає у створенні інформаційної технології управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії, розроблення та впровадження якої має безпосереднє практичне значення для закладів вищої освіти та ІТ-компаній, які прагнуть підвищити рівень взаємодії, цифровізації та індивідуалізації підготовки web-програмістів, оскільки запропонований підхід забезпечує підвищення ефективності підготовки та рангового відбору кандидатів на вакантну посаду за рахунок персоналізованих рекомендацій.

Практичне значення роботи полягає в наступному:

- 1) створено авторський підручник “WEB-програмування” для підготовки web-розробників та web-програмістів [18];
- 2) розроблено електронний навчальний курс “WEB-програмування” для підготовки web-розробників та web-програмістів;
- 3) проведено декомпозицію бізнес-процесів рекрутингу, управління розвитком IT-персоналу, формування планів їх персонального розвитку, підбору web-програмістів в склад проектних IT-команд;
- 4) створено модифіковані матриці компетенцій для напрямку бекенд розробки з використанням CMF Drupal на основі розробленого авторського підручника “ WEB-програмування”;
- 5) створено шаблони оцінювання за модифікованим методом 180° та проведено ітерації оцінювання IT-персоналу, зокрема web-програмістів та порівняльний аналіз результатів;
- 6) формалізовано гібридну модель управління IT-проектами на базі гнучких методологій;
- 7) здійснено проектування архітектури інтелектуальної інформаційної системи з використанням технологій штучного інтелекту на базі платформи LangGraph для управління багаторівневим навчанням та підбором web-програмістів в IT-компанії;
- 8) здійснено програмну реалізацію інформаційної технології автоматизованого формування планів професійного розвитку IT фахівців; реалізацію проведено з використанням мови Python, CMF Drupal та AI технології LangGraph , проведено порівняльний аналіз традиційних планів розвитку з автоматично сформованими;
- 9) проведено інтеграцію розроблених інформаційних систем в об’єднану корпоративну інформаційну систему на базі технологічної екосистеми Amazon AWS;

10) розроблено інтелектуальну інформаційну систему з багаторівневим навчанням та оцінюванням для підготовки web-програмістів з використанням штучного інтелекту;

11) проведено порівняльний аналіз створених цифрових інформаційних технологій та традиційних підходів з відбору web-програмістів в ІТ-компаніях.

Результати дисертаційного дослідження використовуються в навчальному процесі Міжнародного економіко-гуманітарного університету імені академіка Степана Дем'янчука при підготовці здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальностей F2 “Інженерія програмного забезпечення”, F3 “Комп’ютерні науки”, F6 “Інформаційні системи та технології” в галузі знань F “Інформаційні технології”. Зокрема розроблена матриця компетенцій стала основою для вимог до умінь та навичок бекенд-розробників за напрямком “Digital development”. Виданий підручник “WEB-програмування” [18], використовується для вивчення дисциплін: “Основи веб-програмування”, “Веб-програмування”, “Проектування та підтримка веб-застосувань”. Інтеграцію процесів навчання веб-розробників у закладах вищої освіти з управлінськими процесами ІТ-компаній презентовано ГО “Рівненський ІТ-кластер”, як приклад оптимізації взаємодії навчальних закладів та компаній в межах розвитку регіонального ІТ-ринку.

Результати дисертаційного дослідження були впроваджені в процеси провідних ІТ-компаній SIA AnyforSoft (Латвія), ТОВ Смарт-Кінг (Україна), що засвідчується довідками про впровадження (додатки Є та Ж).

Результати роботи підтверджуються проведенням аналізом теоретичних основ, експериментальною перевіркою, публікаціями в українських та міжнародних наукових виданнях, виступами на науково-практичних та фахових професійних ІТ-конференціях.

Особистий внесок здобувача. Результати, що становлять основний зміст дисертації, отримані здобувачем самостійно. В переліку публікацій, які виконані у співавторстві, внесок здобувача вказано в розрізі відповідної публікації.

Результати дисертаційного дослідження доповідались на конференціях та були опубліковані: на міжнародних та національних науково-практичних

конференціях: Міжнародній науково-практичній конференції "Міжнародні тенденції та перспективи розвитку в освіті та науці в умовах глобалізації" до 100-річчя від дня народження акад. С.Дем'янчука, 2025, Рівне; IX Міжнародній науково-технічній конференції "Комп'ютерне моделювання та оптимізація складних систем", 5-7 листопада, 2025, Дніпро; V Всеукраїнській науково-технічній конференції молодих вчених, аспірантів і студентів "Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації", 25-26 вересня, 2025, Одеса; Міжнародній науково-практичній конференції "Beyond Europe: Artificial Intelligence in International Relations and Communication - Opportunities and Challenges", 13 грудня, 2024, Познань, Польща; XVII міжнародній науково-практичній конференції, 31 жовтня - 1 листопада, 2024, Одеса; міжнародній науково-практичній конференції "Актуальні питання та перспективи інноваційного розвитку науки та освіти в умовах євроінтеграції", 14 листопада, 2024, Рівне; XIII Міжнародній науково-практичній конференції "Математика. Інформаційні технології. Освіта", 31 травня - 2 червня, 2024, Луцьк-Світязь; XXIII Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів, 20-21 квітня, 2023, Одеса; Міжнародній науково-практичній конференції до 30-річчя Приватного вищого навчального закладу "Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука" "Інноваційні дослідження та перспективи розвитку науки і техніки у ххі столітті", 19 жовтня, 2023, Рівне; VIII Міжнародній науково-технічній конференції "Комп'ютерне моделювання та оптимізація складних систем", 1-3 листопада, 2023, Дніпро; III Всеукраїнській науково-технічній конференції молодих вчених, аспірантів і студентів, 28-29 жовтня, 2023, Одеса; Всеукраїнській науково-практичній конференції, 29 листопада, 2019, Одеса; V Міжнародній науково-технічній конференції "Комп'ютерне моделювання та оптимізація складних систем", 2019, Дніпро; VII Міжнародній науково-технічній конференції, 15-18 вересня, 2009, Київ-Севастополь; V Міжнародній науково-технічній конференції "Комп'ютерні технології в будівництві", 18-21 вересня, 2007, Київ-Севастополь.

Публікації: За темою дисертаційної роботи опубліковано 17 наукових праць, яких 10 у зарубіжних колективних монографіях, 4 - у наукових фахових виданнях України, 1 у виданнях, що індексуються у міжнародних науково-метричних базах Index Copernicus, Web of Science, Scopus, 16 доповідей на міжнародних науково-практичних конференціях.

Структура та обсяг дисертації: Дисертаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел та додатків. Загальний обсяг дисертації становить 217 сторінок, з яких 160 становить текст, 44 рисунки, 14 таблиць. Список використаних джерел містить 224 найменування, з яких 189 зарубіжні.

РОЗДІЛ 1.

АНАЛІЗ СУЧАСНИХ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ТА НЕОБХІДНІСТЬ ЇХ ЗАСТОСУВАННЯ ПРИ ВІДБОРІ НА ПОСАДИ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ

1.1. Аналіз існуючих інтелектуальних інформаційних системи для відбору програмістів в ІТ-компаніях

У сфері HR головним є залучення кваліфікованих програмістів, побудова командних відносин, забезпечення позитивного досвіду кандидатів та розширення кадрового резерву, інакше це несе бізнес-ризик ІТ-компанії. Однією з основних задач ІТ-компанії є пошук та виявлення нових і різноманітних кадрових резервів. Цифровий талант – це найважливіший ресурс, яким володіє ІТ-бізнес, він сприяє зростанню та конкурентостійкості. Але, як і багато інших ресурсів, талантів часто не вистачає, і їх може бути важко знайти та розвивати.

Рекрутери традиційно шукають кандидатів з обмеженої кількості місць, щоб керувати обсягом заявок. При цьому штучний інтелект дозволяє розширити коло пошуку та допомагає рекрутеру перевірити набагато більше кандидатів і знайти, чий навички відповідають вимогам вакансії. Штучний інтелект та генеративний штучний інтелект значно може заощадити час, витрачений на рутинні пошуки кандидатів та покращити аналітику, що висвітлює стратегічне планування ІТ-компанії.

Провідні ІТ-компанії активно поєднують новітні технології з віддаленою роботою та автоматизують HR-процеси: рекрутинг, онбординг, оцінку ефективності, фінансові та нефінансові компенсації в ІТ, розвиток цифрових талантів.

В свою чергу, кандидати на оголошені вакансії в ІТ-компанії сподіваються на прості функції подання заявок, співбесіди та швидкі відповіді. Як свідчать результати дослідження BCG та The Network [19] сучасний процес рекрутингу включає кандидатів, які ніколи не отримують відповіді, відразу отримують

відмову або стикаються зі складною системою оцінювання. У даному дослідженні брали участь 90 000 осіб зі 160 країн, понад половина (52%) кандидатів відповіли, що відмовилися б від привабливої пропозиції, якби мали якийсь негативний досвід під час процесу рекрутингу.

Для вирішення цих проблем ІТ-компанії мають використовувати новітні інформаційні технології.

Згідно з опитуванням [19] 70% ІТ-компаній вже використовують штучний інтелект або GenAI у сфері управління персоналом для створення контенту, таке як написання описів вакансій, маркетингових електронних листів, створення оцінок та планування співбесід. Більше половини компаній - 54% використовують штучний інтелект або GenAI при підборі кандидатів шляхом поєднання навичок із вимогами до вакансії. Використання штучного інтелекту під час рекрутингу є частиною швидкозмінного цифрового середовища.

Для кандидатів вибір роботи це дуже особистий процес. Це початок надзвичайно важливих стосунків, які можуть визначити кілька, або навіть багато років їхнього життя.

Якщо для роботодавця - ІТ-компанії рекрутинг - це важливе бізнес-рішення, то для кандидатів на посаду - це особисте рішення і брак відповідних ІТ-фахівців у сферах з високим попитом перешкоджає зростанню ІТ-бізнесу. Іноді роботодавці не знають, які HR-процеси сприяють найму потенційних працівників, а які відлякують їх.

Протягом 2022 року консалтингова фірма BCG та глобальна рекрутингова платформа The Network - альянс web-сайтів для міжнародного пошуку талантів та працевлаштування, провели наймасштабніше у світі опитування потенційних кандидатів на роботу щодо працевлаштування. В опитуванні взяли участь понад 90 000 респондентів, 74% з яких заявили, що до них звертаються кілька разів на рік щодо можливостей працевлаштування, а 36% заявили, що до них звертаються щомісяця. Найбільш бажаними є кандидати, які працюють в ІТ та цифрових технологіях.

68% респондентів вважають, що вони мають сильну переговорну позицію під час пошуку роботи, оскільки впевнені у своїй практичній цінності. З цифровими суперзірками ІТ-компанії можуть не отримати другого шансу, тому вони, як правило приймають рішення на користь такого кандидата.

Бажання кандидатів різняться залежно від того, на якому етапі своєї кар'єри вони знаходяться та як планують кар'єрну траєкторію.

У дослідженні розглядалися респонденти за віковими групами (таблиця 1.1). Головними пріоритетами незалежно від когорти виявилось фінансове задоволення та баланс між роботою та особистим життям.

Таблиця 1.1. Розподіл факторів прийняття рішень кандидатів за віковими фокус групами.

<21	21-30	31-40	41-50	51-60	>60
Work-life баланс	Фінансова коменсації	Фінансова коменсації	Фінансова коменсації	Work-life баланс	Значимість роботи
Фінансова коменсації	Work-life баланс	Work-life баланс	Work-life баланс	Фінансова коменсації	Визнання результатів
Розвиток і навчання	Оплачувані вихідні	Безпечність	Безпечність	Відносини з менеджером	Фінансова коменсації
Визнання результатів	Безпечність	Гнучкий графік та віддалена робота	Гнучкий графік та віддалена робота	Визнання результатів	Відносини з менеджером
Гнучкий графік та віддалена робота	Гнучкий графік та віддалена робота	Оплачувані вихідні	Відносини з менеджером	Цінності компанії	Цінності компанії
Оплачувані вихідні	Здоров'я та страхування	Здоров'я та страхування	Визнання результатів	Значимість роботи	Зацікавленість в роботі
Здоров'я та страхування	Розвиток і навчання	Визнання результатів	Цінності компанії	Зацікавленість в роботі	Work-life баланс
Безпечність	Визнання результатів	Розвиток і навчання	Оплачувані вихідні	Гнучкий графік та віддалена робота	Гнучкий графік та віддалена робота

Цінності компанії	Соціальний пакет	Відносини з менеджером	Значимість роботи	Безпечність	Виклики в роботі
Соціальний пакет	Цінності компанії	Цінності компанії	Здоров'я та страхування	Здоров'я та страхування	Репутація компанії

Як видно, для наймолодшого покоління можливості навчання та розвитку є дуже важливими, але ця характеристика поступово втрачає свою важливість з віком респондентів. Зниження мотивації до навчання може становити проблему, оскільки зі стрімким розвитком інформаційних технологій та технологій штучного інтелекту ІТ-фахівцям необхідно систематично підвищувати свою кваліфікацію або взагалі перекваліфікуватись.

Працівники віком від 30 до 50 років надають пріоритет стабільності зайнятості та гнучкому графіку роботи.

Серед респондентів старше 60 років працевлаштування та визнання їхньої роботи займають високі місця, це свідчить про старіння робочої сили в багатьох країнах.

42% респондентів дослідження повідомили, що активно шукають роботу, 41% відмітили, що не шукають активно роботу, але були б готові змінити роботу в разі слушної пропозиції.

Мотивація цих кандидатів, які активно та пасивно шукають роботу, відрізняється. Активних кандидатів мотивує майбутній кар'єрний зріст, більша відповідальність або можливість опанувати нову професію. Їх мотивує зміст та обсяг самої майбутньої роботи.

Пасивних кандидатів цікавить покращений компенсаційний пакет і кращий баланс між роботою та особистим життям.

Роботодавцям необхідно залучати кандидатів з різними талантами, щоб мати значну перевагу.

Невдалий досвід рекрутингу може зірвати навіть надзвичайно привабливу пропозицію роботи. У проведеному опитуванні 52% респондентів заявили, що відмовилися б від привабливої пропозиції, якби мали негативний досвід під час

рекрутингу. Тому IT-компаніям важливо розуміти ідеальний шлях кандидата до рекрутингу на кожному етапі.

Для підвищення довіри потенційних кандидатів до оголошених вакансій в IT-компаніях у дослідженні пропонуються наступні дії:

- збирати дані з різних джерел (соціальні мережі, опитування нових співробітників, фокус-групи) для формування портретів талановитих фахівців;
- відстежувати великі IT-компанії на предмет звільнень або переїздів, щоб залучити Agile-коучів, які можуть звернути увагу на місцевий стартап-майданчик;
- дотримуватись найкращих оптимізованих практик, а не застарілих систем, скорочуючи процес до основних кроків та зосереджуючись на доданій цінності для кандидата;
- розширювати кадровий резерв, приділяти менше уваги формальним вимогам щодо дипломів, сертифікатів та стажу роботи, натомість зосередитися на навичках, мотивації та потенціалі; чим більший буде кадровий резерв, тим більша ймовірність отримати доступ до потрібних талантів;
- звернутися до міжнародних IT-ринків, що розвиваються, де кандидати можуть бути готові працювати віддалено;
- працювати з програмами підвищення кваліфікації та інклюзії, спрямованими на пошук роботи для меншин та малозабезпечених верств населення;
- оцінити респондентів за методологією STAR - фокус на кандидатів, які отримали кваліфікацію альтернативними шляхами, вони не мають ступеня бакалавра, але мають досвід роботи та навички і можуть бути джерелом талантів; лише в США налічується близько 70 мільйонів STAR [20].

Інтелектуальні інформаційні системи для відбору програмістів уже існують і застосовуються в IT-компаніях та рекрутингових службах, але вони найчастіше не є вузькоспеціалізованими «для програмістів» у вигляді окремого продукту - натомість це комплекси HR/ATS/AI рішень, що автоматизують та інтелектуалізують процес найму [19].

Штучний інтелект стрімко змінює роботу HR-менеджерів та рекрутерів. За даними консалтингової фірми BCG, у 2024 році його використання у підборі кадрів значно зросло, і вже 92% компаній відзначають позитивний вплив AI на ефективність роботи.

В даний час AI активно використовують в наступних процесах рекрутингу:

- при пошуку кандидатів AI швидко аналізує великі бази даних (LinkedIn, джоб-борди, внутрішні CRM) та формує короткі списки кандидатів за ключовими компетенціями;
- при первинному скринінгу чати-боти проводять первинні інтерв'ю, де AI автоматично аналізує резюме (CV parsing) і заповнює ATS-систему;
- при аналітиці та прогнозуванні AI допомагає рахувати time-to-hire, cost-per-hire, прогнозує, скільки часу реально займе закриття вакансії;
- при onboarding та адаптації AI-асистенти можуть допомагати новачкам: відповідати на типові питання, пояснювати правила та політики IT-компанії.

AI також може взяти на себе нагадування, follow-up, відповіді на типові питання, може оцінити, які канали дають найкращих кандидатів (LinkedIn, рекомендації, job-борди).

При цьому AI - це допоміжний інструмент рекрутера або HR, що бере на себе великий обсяг рутинної роботи, вивільняючи час на роботу з hiring-менеджерами та стратегічне планування пошуку, залучення та утримання цифрових талантів.

AI-рекрутинг платформи (зокрема Workable, Personio, Recrutee) належать до класу хмарних інформаційних систем управління персоналом та систем відстеження кандидатів (HRIS/ATS), у яких реалізовано алгоритми аналізу даних і елементи інтелектуальних технологій. Основним призначенням таких платформ є автоматизація та інтелектуалізація процесів відбору кандидатів, зокрема в IT-компаніях.

Функціонально AI-рекрутинг платформи забезпечують збір, зберігання та обробку великих обсягів кадрових даних, включаючи резюме кандидатів,

результати тестувань, історію взаємодії та дані про вакансії. На основі цих даних реалізується автоматизований попередній скринінг кандидатів, який ґрунтується на алгоритмах обробки природної мови (NLP), методах інформаційного пошуку та машинного навчання. Такі алгоритми дозволяють виділяти ключові компетентності, досвід, технологічні навички та порівнювати їх із вимогами вакансій.

Важливою функцією AI-рекрутинг платформ є ранжування кандидатів за рівнем відповідності визначеним критеріям. Ранжування здійснюється на основі моделей оцінювання, які враховують як формалізовані параметри (досвід роботи, володіння мовами програмування, освіта), так і напівструктуровані або неструктуровані дані, отримані з текстових описів резюме. Результатом є формування рекомендаційних списків кандидатів, що значно скорочує час прийняття рішень рекрутерами.

Крім того, такі платформи підтримують інтеграцію з онлайн-тестуванням, технічними завданнями та відео-інтерв'ю, що дозволяє здійснювати багатокритеріальне оцінювання кандидатів, у тому числі web-програмістів. У деяких системах застосовуються елементи предиктивної аналітики, спрямовані на прогнозування успішності кандидата на посаді або ймовірності його довгострокової співпраці з компанією.

Таким чином, AI-рекрутинг платформи можна розглядати як практичну реалізацію інтелектуальних інформаційних систем у сфері управління людськими ресурсами, однак більшість існуючих рішень орієнтовані на універсальні сценарії рекрутингу і не враховують специфіку відбору IT-фахівців, зокрема web-програмістів. Це зумовлює необхідність подальших досліджень і розроблення спеціалізованих інтелектуальних інформаційних систем, адаптованих до вимог IT-галузі.

Використання методів обробки природної мови (Natural Language Processing, NLP) є одним із ключових напрямів інтелектуалізації процесів відбору web-програмістів. Основне призначення NLP-модулів полягає в автоматизованому аналізі текстових даних, зокрема резюме кандидатів, описів вакансій та

супровідних матеріалів, з метою виділення професійних компетентностей і оцінювання відповідності кандидатів вимогам посади.

У процесі відбору web-програмістів NLP-алгоритми застосовуються для структуризації неформалізованої інформації резюме, що містить описи досвіду роботи, технічних навичок, реалізованих проєктів та використаних технологій. За допомогою методів лексичного аналізу, розпізнавання іменованих сутностей та семантичного аналізу здійснюється автоматичне виявлення ключових технологічних компетентностей, таких як мови програмування, фреймворки, системи керування базами даних, інструменти контролю версій та методології розроблення програмного забезпечення.

Окремим етапом є аналіз відповідності резюме вимогам вакансії web-програміста, що реалізується шляхом зіставлення семантичних представлень текстів резюме та опису вакансії. Для цього можуть використовуватися векторні моделі подання тексту, методи семантичної подібності та алгоритми машинного навчання, які дозволяють оцінювати рівень релевантності кандидатів не лише за ключовими словами, а й за змістовою близькістю професійного досвіду до вимог посади.

Застосування NLP у відборі web-програмістів забезпечує автоматизоване ранжування кандидатів за рівнем відповідності технічним вимогам вакансії, що сприяє зменшенню часу первинного скринінгу та підвищенню об'єктивності оцінювання. Крім того, використання семантичного аналізу дозволяє виявляти приховані або неявно сформульовані компетентності, зокрема досвід роботи з певними архітектурними підходами або технологічними стеком, які не завжди прямо зазначені у резюме.

Таким чином, застосування NLP-технологій у процесах відбору web-програмістів є важливим інструментом інтелектуальних інформаційних систем рекрутингу, що забезпечує більш точне, масштабоване та формалізоване оцінювання професійних компетентностей кандидатів.

Чат-боти та віртуальні асистенти є складовими інтелектуальних інформаційних систем рекрутингу, призначеними для автоматизації взаємодії з

кандидатами на етапах попереднього відбору, збору інформації та організації співбесід. У контексті відбору web-програмістів такі інструменти забезпечують оперативну комунікацію з великою кількістю кандидатів та зменшують навантаження на рекрутерів.

Функціонально чат-боти реалізують автоматичні відповіді на типові запити кандидатів, зокрема щодо вимог до вакансії, технологічного стеку, формату роботи, етапів відбору та умов працевлаштування. Використання методів обробки природної мови дозволяє забезпечити коректну інтерпретацію запитів кандидатів та формування релевантних відповідей у діалоговому режимі.

Окремим завданням чат-ботів є автоматизований збір додаткових даних, необхідних для первинного скринінгу web-програмістів. Це може включати уточнення рівня володіння мовами програмування, досвіду роботи з фреймворками, участі в проєктах, наявності портфоліо або посилань на репозиторії з програмним кодом. Зібрана інформація передається до HRIS або ATS та використовується для подальшого аналітичного оброблення.

Віртуальні асистенти також застосовуються для планування співбесід, зокрема узгодження часу технічних і HR-інтерв'ю, інтеграції з календарними сервісами та автоматичного інформування кандидатів про заплановані етапи відбору. Це забезпечує підвищення організаційної ефективності процесу рекрутингу та зменшує кількість помилок, пов'язаних із людським фактором.

З точки зору інтелектуальних інформаційних систем чат-боти та віртуальні асистенти виконують роль інтерфейсних та сервісних компонентів, що забезпечують безперервний обмін інформацією між кандидатами та системами підтримки прийняття рішень. Їх використання у відборі web-програмістів сприяє підвищенню масштабованості рекрутингових процесів, стандартизації первинного оцінювання та покращенню якості взаємодії з кандидатами.

Проведений аналіз свідчить про активний розвиток інтелектуальних моделей автоматизованого відбору кандидатів, що базуються на методах обробки природної мови (NLP), великих мовних моделях (Large Language Models, LLM) та мультиагентних підходах. У межах таких досліджень розробляються алгоритми

автоматичного скринінгу резюме, здатні аналізувати неструктуровані текстові дані, виділяти професійні компетентності кандидатів та оцінювати їх відповідність вимогам вакансій.

Застосування LLM та сучасних NLP-моделей дозволяє реалізувати семантичний аналіз резюме і описів вакансій, що забезпечує більш точне зіставлення професійного досвіду кандидатів з вимогами роботодавців порівняно з традиційними підходами, заснованими на ключових словах. Такі моделі здатні враховувати контекст, синонімію та приховані зв'язки між компетентностями, що є особливо важливим у сфері IT-рекрутингу, зокрема при відборі web-програмістів.

Окремий напрям досліджень пов'язаний із використанням мультиагентних систем, у яких різні агенти виконують спеціалізовані функції, такі як аналіз резюме, оцінювання технічних навичок, ранжування кандидатів та формування рекомендацій. Такий підхід забезпечує модульність, масштабованість і гнучкість інтелектуальних систем відбору персоналу.

Важливим аспектом сучасних наукових робіт є розроблення пояснюваних інтелектуальних систем (Explainable AI, XAI) для підбору кандидатів. У таких системах результати автоматизованого скринінгу та ранжування супроводжуються поясненнями щодо причин прийняття рішень, що підвищує прозорість, довіру користувачів та відповідність етичним і регуляторним вимогам.

У сучасних IT-компаніях активно застосовуються інтелектуальні інформаційні системи (ІС) для автоматизації процесів управління персоналом. До класу таких систем належать хмарні HRIS/ATS-платформи (Workable, Personio, Recruitee), які інтегрують модулі автоматизованого скринінгу резюме, ранжування кандидатів та підтримки прийняття рішень. Дослідження показують, що використання таких систем дозволяє скоротити час рекрутингу та підвищити об'єктивність оцінки, проте більшість рішень орієнтовані на універсальні критерії і недостатньо враховують специфіку відбору IT-фахівців, зокрема web-програмістів [1–3].

Сучасні інформаційні системи управління персоналом і рекрутингу все частіше інтегрують інтелектуальні модулі, що дозволяють суттєво підвищити

ефективність відбору кандидатів шляхом автоматизації рутинних процесів, таких як аналіз резюме, оцінювання кандидатів та формування рекомендацій. Зокрема в наукових роботах розглядаються підходи до проектування ІС на основі штучного інтелекту та автоматизованого аналізу даних про кандидатів для підтримки прийняття рішень у HR-процесах [21].

Методи обробки природної мови (NLP) широко використовуються для аналізу резюме та оцінки відповідності кандидатів вимогам вакансій. NLP дозволяє виділяти ключові компетенції, досвід роботи, використані технології та навички програмування, включаючи володіння мовами та фреймворками, що критично для web-програмістів [4–6]. Семантичний аналіз тексту та моделі великих мов (LLM) дозволяють оцінювати резюме не лише за ключовими словами, а й за змістовною відповідністю до вимог вакансії. Це підвищує точність скринінгу і дозволяє автоматично формувати рейтинг кандидатів.

Наприклад, системи рекомендацій на основі NLP поєднують видобування ознак тексту, розпізнавання іменованих сутностей та тематичний аналіз з традиційними алгоритмами фільтрації, що підвищує ефективність пошуку релевантних співвідношень між кандидатами і вакансіями [22].

Сучасні AI-рекрутинг платформи, такі як Canditech, поєднують тестування технічних компетенцій, симуляції професійних задач та відео-співбесіди, інтегруючи результати у загальну модель оцінки кандидата. Інтелектуальні алгоритми таких платформ дозволяють здійснювати ранжування кандидатів за рівнем відповідності технічним та когнітивним вимогам вакансії, що особливо актуально для web-програмістів.

У сфері автоматичного скринінгу резюме та оцінювання кандидатів запропоновано інноваційні підходи, які використовують великі мовні моделі (LLM) і мультиагентну архітектуру. Такі моделі забезпечують семантичний та контекстуальний аналіз тексту, де агенти виконують різні етапи процесу - від вилучення інформації до оцінювання та формування підсумкових балів відповідності. Наприклад, у дослідженнях показано, що інтегрована система

LLM-агентів може значно прискорити скринінг резюме та підвищити якість оцінок у порівнянні з традиційними методами [23].

Чат-боти та віртуальні асистенти застосовуються для автоматизації взаємодії з кандидатами, включаючи відповіді на типові запити, збір додаткових даних та планування співбесід. У поєднанні з NLP вони дозволяють стандартизувати первинний етап відбору і значно зменшують навантаження на рекрутерів.

Деякі наукові розробки зосереджені на створенні пояснюваних систем (Explainable AI) для рекрутингу, що дозволяють не лише автоматизувати процеси аналізу резюме, але й формально обґрунтувати отримані результати. Наприклад, сучасні NLP-конвеєри автоматично видобувають структуру з неструктурованого тексту, здійснюють семантичне зіставлення резюме і описів вакансій у загальному векторному просторі та забезпечують прозорість прийнятих рішень [24].

Практичні реалізації інтелектуальних систем рекрутингу, які включають NLP-модулі для аналізу текстів резюме і вакансій, показують потенційні переваги у створенні об'єктивних механізмів оцінки кандидатів. Наприклад, в освітніх та молодших наукових проєктах розглядається розробка систем обліку вакансій і кандидатів з персоналізованим ШІ-асистентом, що автоматично оцінює відповідність резюме вимогам та формує ранжування кандидатів [25].

1.2. Аналіз основних напрямів застосування інтелектуальних інформаційних систем у цифровому суспільстві

Сучасні інтелектуальні інформаційні системи (ІС) є складовою інструментально-технологічного забезпечення цифрового суспільства, заснованого на знаннях. ІС є комплексом уніфікованих методологічних, психолого-педагогічних, логіко-математичних та програмно-технічних засобів, призначених для підтримки діяльності користувача в режимі розширеного діалогу природною мовою [47].

В даний час можна виділити такі технологічні модулі (ТМ) ІС:

- ТМ 1 - обробка та аналіз даних: ІС спроможні обробляти величезні обсяги даних, надаючи інсайти, які раніше були недоступні через обмеження традиційних методів обробки;
- ТМ 2 - автоматизація та оптимізація бізнес-процесів: ІС сприяють автоматизації рутинних та складних завдань, зменшуючи людський фактор і підвищуючи продуктивність;
- ТМ 3 - підтримка прийняття рішень: штучний інтелект та машинне навчання в ІС дозволяють краще розуміти складні сценарії та надавати обґрунтовані рекомендації;
- ТМ 4 - інтерактивність та користувацький досвід: ІС вносять значний вклад у поліпшення користувацького досвіду через персоналізацію та інтерактивність;
- ТМ 5 - прогнозування та планування: ІС ефективно використовують аналітичні дані спостережень та моніторингу для прогнозування майбутніх тенденцій та планування стратегій.

Модуль ТМ 1 використовує складні алгоритми для виявлення шаблонів, тенденцій та відхилень, що дозволяє краще розуміти феномени і приймати обґрунтовані рішення. Цей модуль достатньо добре вивчений та використовується для повноцінних досліджень з візуалізацією даних у режимі реального часу.

Модуль ТМ 2 набув широкого поширення в системах управління та автоматизації. Вони оптимізують процеси за допомогою розумного планування та розподілу ресурсів, що є важливим у секторах з високим рівнем складності та великим обсягом даних.

Модуль ТМ 3 передбачає узагальнення результатів досліджень та вибір рішень у первинних неструктурованих або слабо структурованих задачах, у тому числі й тих, що мають багато критеріїв. Це особливо важливо у сферах, де потрібні швидкі та точні рішення.

Модуль ТМ 4 передбачає взаємодію між користувачами та комп'ютерами, гаджетами через зручний інтерфейс користувача. Це включає персоналізовані

рекомендації, інтерактивні навчальні платформи, а також розумні асистенти та чат-боти для забезпечення кращої комунікації з користувачами.

Модуль ТМ 5 передбачає створення потужної аналітики для детального прогнозування та планування стратегій та процесів. Вони допомагають в антиципації майбутніх подій та підготовці до них, що є важливим у цифровому суспільстві.

За критерієм ступеня інтелектуалізації, який у першому наближенні може характеризуватися обсягом інформації, що обробляється, інформаційні системи (ІС) можна поділити наступним чином [17]:

- системи перебору варіантів рішень згідно з встановленою пріоритетністю для наперед змодельованих ситуацій;
- ІС, які приймають рішення за детермінованими вирішальними правилами без навчання;
- ІС, що реалізують алгоритми компараторного розпізнавання за еталонами;
- експертні системи, що з метою прийняття ефективних рішень маніпулюють спеціальними знаннями, накопиченими фахівцями-експертами у конкретній предметній галузі знань;
- системи підтримки прийняття рішень, які здатні самостійно створювати бази знань;
- ІС, що самонавчаються;
- знання-орієнтовані (knowledge-based) ІС, що здатні утворювати базу знань за якісною шкалою виміру ознак розпізнавання.

Отже, інформаційні системи, які здатні до навчання можна поділити на такі основні класи:

- системи, що ґрунтуються на інструктивних знаннях (rulebased reasoning);
- системи, що ґрунтуються на автоматичному доведенні теорем (automatic theorem-proving techniques);

- системи, що ґрунтуються на автоматичному породженні гіпотез (automatic hypothesizing);
- системи, що ґрунтуються на доведенні за аналогією (analogical reasoning);
- об'єктно-орієнтовані інтелектуальні системи (object-oriented intelligent systems);
- об'єктно-логічні інтелектуальні системи, що поєднують окремі переваги об'єктно-орієнтованих систем з системами автоматичного доведення теорем і використовують об'єктно-логічні мови, фреймові логіки (F-logics), логіки транзакції (transaction logics) та інше.

Наведена вище класифікація є не остаточною, оскільки відбувається неперервне розширення функціоналу та інструментарію ІС, отже інтелектуальні інформаційні системи перебувають в стані постійного розвитку та вдосконалення.

Розвиток інтелектуальних інформаційних систем є комплексним та багатоетапним процесом, що охоплює різні аспекти, від теоретичних досліджень до практичного застосування. Цей процес можна розділити на шість ключових етапів:

I етап - визначення вимог та збір даних:

- на цьому етапі відбувається аналіз потреб користувачів та визначення основних функцій системи;
- збір та аналіз даних, які будуть використовуватися ІС для навчання та обробки.

II етап –теоретичне моделювання та розробка алгоритмів:

- розробка математичних та комп'ютерних моделей, які ляжуть в основу ІС;
- створення алгоритмів штучного інтелекту, машинного навчання та обробки даних, які дозволяють системі виконувати свої функції.

III етап - розробка прототипу та тестування:

- створення робочих прототипів системи для перевірки її функціональності та ефективності;

- тестування прототипу на реальних даних та у реальних умовах для виявлення та виправлення помилок.

IV етап - інтеграція та оптимізація:

- інтеграція ІС з існуючими системами та процесами користувача;
- оптимізація продуктивності, надійності та масштабованості системи.

V етап - пілотне впровадження та оцінка ефективності:

- розгортання системи в обмеженому масштабі для оцінки її ефективності та виявлення можливих проблем при її використанні;
- збір зворотного зв'язку від користувачів та аналіз ефективності системи.

VI етап - масштабне впровадження та підтримка:

- впровадження системи на повну шкалу для широкого спектра користувачів;
- надання підтримки та оновлень для підтримки актуальності та ефективності системи.

Кожен з цих етапів вимагає співпраці між різними ІТ-фахівцями, включаючи дослідників у галузі штучного інтелекту (ШІ), розробників програмного забезпечення, аналітиків даних, а також кінцевих користувачів. Такий підхід дозволяє створити ефективні та адаптивні ІС, які відповідають потребам користувачів та викликам сучасного цифрового світу.

Враховуючи ці аспекти, ІС виступають як каталізатори в розвитку цифрового суспільства, підвищуючи його ефективність та інноваційність [43].

Проблема використання інтелектуальних інформаційних систем актуальна для багатьох галузей економіки та виробництва. Одним з напрямків застосування ІС в цифровому суспільстві в даний час можна виділити: використанні ШІ у фінансовій та банківській справі, використанні ШІ для оптимізації логістики та управління запасами, інтелектуальне управління розумними містами та інфраструктурою, цифровізація виробництва та використання ШІ в освіті для автоматизованого оцінювання та надання зворотного зв'язку та ін.

Особливо, інтелектуальні інформаційні системи відіграють велику роль у розвитку освіти та навчання, забезпечуючи інноваційність, ефективність та виконують функції [43]:

- персоналізація навчання: ПС дозволяють створити індивідуальний навчальний план, адаптований до потреб та здібностей кожного навчаного, це забезпечує глибше засвоєння матеріалу та підвищує мотивацію до навчання;
- адаптивне навчання: системи можуть аналізувати відповіді і автоматично налаштовувати складність завдань або контенту, забезпечуючи при цьому оптимальний рівень траєкторії навчання;
- доступність ресурсів: ПС можуть забезпечити доступ до освітніх ресурсів, роблячи освіту більш інклюзивною;
- оцінювання та зворотний зв'язок: автоматизація процесу оцінювання дозволяє викладачам та менторам зосередитися на більш важливих аспектах навчання, а навчаваним надає швидкий та об'єктивний зворотний зв'язок;
- підготовка до майбутнього: використання ПС у навчанні готує до роботи в сучасному технологічному світі, розвиваючи важливі навички роботи з інноваційними технологіями.

Проте, поки не вироблена науково обгрунтована модель оптимального використання інтелектуальних інформаційних систем не лише в освіті, а й загалом в галузях економіки. Темпи розвитку інформаційних технологій випереджають темпи розвитку освітньої галузі.

У зв'язку з цим, для прискорення впровадження ПС у сферу освіти з підготовки майбутніх програмістів необхідна розробка автоматизованих навчальних програм з елементами штучного інтелекту для різних учбових дисциплін.

1.3. Дослідження методології створення інформаційної системи

Необхідність використання інтелектуальних інформаційних систем викликана стрімким прискоренням науково-технічного прогресу цифрового

суспільства, який обумовлює необхідність діджиталізації системи освіти, зокрема у підготовці майбутніх програмістів.

Для будь-якої ІТ-організації є істотним встановлення регламенту функціонування інформаційної системи – від виявлення інформаційних потреб до використання інформації. Йдеться про типізації завдань, що вирішуються в ІТ-компанії, встановлення періодичності отримання, обробки та використання інформації, стандартизації вхідних і вихідних документів, процедур обробки інформації.

Кожну ІТ-компанію або її частину можна розглядати як систему, яка прагне у своєму функціонуванні до досягнення поставленої мети. Для системи характерні наступні основні властивості: складність, подільність, цілісність, різноманіття елементів, відмінність їх природи, структурованість.

Методологія створення ІС має певні фундаментальні принципи побудови. Головним з яких є системний підхід, який передбачає декомпозицію системи на частини (підсистеми) згідно з цілями її функціонування. Таким чином, структуру інформаційної системи становить сукупність окремих її частин, підсистем. Сучасна концепція структури ІС містить функціональну і забезпечувальну частини (рис. 1.1)



Рис. 1.1. Концепція структури інформаційної системи

Функціональна частина ІС (рис.2.2.) – це інформаційно-організаційна модель управління діяльністю ІТ-компанії. Основне призначення цієї частини – автоматизоване розв'язання задач і виконання необхідних обчислень.

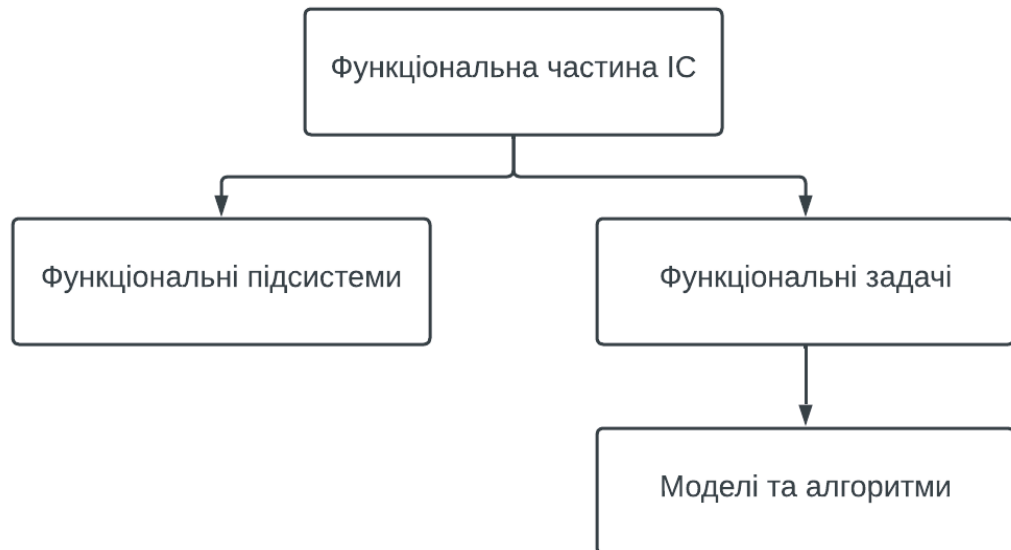


Рис. 1.2. Типовий склад функціональної частини ІС.

Інформаційні системи незалежно від сфери їх застосування включають однаковий склад забезпечувальної частини ІС (рис. 1.3.).

Система опрацювання даних (СОД) призначена для інформаційного обслуговування фахівців різних рівнів управління організацією, що приймають управлінські рішення.

Забезпечувальна частина відповідає цілям функціонування всієї системи і забезпечує автоматизоване розв'язання задач різних управлінських служб.

Специфічними цілями ІС на сучасному етапі її розвитку є: забезпечення інтеграції завдань, вирішуваних ІС, інтерпретація даних, діагностика, моніторинг, проектування, прогнозування, планування, навчання, керування, підтримка прийняття рішень з метою широкого використання ІС чи її компонентів в учбових та наукових закладах, на виробництві та в побуті.

Призначення підсистеми інформаційного забезпечення полягає у своєчасному формуванні й видачі достовірної інформації для прийняття управлінських рішень.



Рис. 1.3. Типовий склад забезпечувальної частини ІС.

Математичне забезпечення включає засоби моделювання процесів управління, методи оптимізації досліджуваних процесів та прийняття рішень (методи багатокритеріальної оптимізації, математичного програмування, математичної статистики, теорії масового обслуговування та ін.). МЗ є основою для розробки спеціалізованого програмного забезпечення.

Актуальне значення при визначенні складу ПЗ має забезпечення інформаційної безпеки ІС (регламентація доступу до ресурсів, антивірусний захист, резервне копіювання, шифрування, електронний цифровий підпис та ін.).

В процесі розв'язання задач управління організаційне забезпечення визначає взаємодію працівників управлінських служб і персоналу ІС з технічними засобами та між собою.

Правове забезпечення розробки інформаційної системи включає нормативні акти договірних взаємовідносин між замовником і розроблювачем ІС, правове регулювання відхилень.

Лінгвістичне забезпечення використовується для забезпечення зручності спілкування між користувачем і комп'ютером.

Ергономічне забезпечення формується сукупністю методів і засобів, призначених для створення оптимальних умов високоякісної, високоефективної і безпомилкової діяльності фахівців.

Впровадження ІС в процеси управління ІТ компанією сприяє удосконаленню організаційних структур, оскільки обґрунтовано визначається чисельність апарату управління, чисельність та ефективність штату програмістів, з розподілом по структурних підрозділах службових обов'язків.

1.4 Проблеми та перспективи підготовки і відбору web-програмістів в ІТ-компаніях

Стрімкий розвиток інформаційних технологій у цифровому суспільстві визначає необхідність підготовки творчих ІТ-фахівців, зокрема програмістів. Тому стає актуальною розробка сучасних методичних підходів до використання новітніх інформаційних технологій в процесі підготовки програмістів та створення нових технологій і методів вирішення як навчальних так і практичних завдань.

Разом з тим розробка програмного забезпечення та його супровід, надання ІТ-послуг залежить від якісного командного складу програмістів. Сучасні

підприємства з їх ІТ-інфраструктурою потребують фахівців, які володіють спеціальними знаннями та навичками в галузі інформаційних технологій. Створення й підтримка web-сайтів та web-додатків вимагає ефективної розробки та відлагодження, контролю якості та тестування, що є в компетенції професійної діяльності web-програмістів.

Як показує дослідження, кваліфіковані web-програмісти віддають перевагу професійній діяльності в колективі ІТ-професіоналів, тобто в ІТ-компаніях, де є можливість взаємодіяти в перспективних ІТ-проектах, з новими технологіями та прозорою системою організації професійної діяльності.

ІТ-компанії займаються створенням та впровадженням ІТ-продуктів та ІТ-послуг, де робота програмістів відіграє ключову роль. Згідно даних досліджень [1, 2], основну частину українського ІТ-експорту складає аутсорсинг ІТ-послуг.

Незважаючи на велику кількість ІТ-фахівців, яких щорічно готують заклади вищої освіти в Україні, пошук кваліфікованих web-програмістів є нетривіальною задачею для ІТ-компаній. Згідно статистики, яку надає сервервіс DOU, відсоток програмістів серед нових фахівців, які увійшли на ІТ-ринок протягом 2023 року, складає 45%, тобто абсолютну більшість в порівнянні з іншими технічними та не технічними ІТ-професіями [3]. Також, згідно статистики DOU, 66% нових програмістів, які увійшли на ІТ-ринок протягом 2023 року мають вищу освіту. Для загального обсягу програмістів на українському ІТ-ринку цей показник становить 86%.

Це визначає необхідність активної взаємодії ІТ-компаній із закладами вищої освіти, зокрема сприяння первинному відбору потенційних здобувачів через проходження практики, стажування, професійного корпоративного навчання та організації контрактованої співпраці з студентами старших курсів, які успішно проходять стажування та показують перспективні результати. Таким чином, ІТ-компанії забезпечують підготовку майбутніх контракторів із забезпеченням високого рівня професійної сумісності з процесами і технічними особливостями компанії.

Тобто, підбір кваліфікованих програмістів, зокрема web-програмістів є одним з найскладніших та найактуальніших завдань для менеджера з персоналу в ІТ-компанії. Основною проблемою при відборі програмістів є об'єктивна оцінка hard- та soft-скілів.

Вимоги до кандидатів змінюються в залежності від рангу, складу команди, типу та особливостей ІТ-проекту. Розробка методики автоматизованої оцінки кандидатів при відборі на посаду web-програміста служить доповненням до традиційних співбесід, та є ефективною альтернативою, за допомогою якої можна здійснювати підбір web-програмістів на різні ІТ-позиції.

З впровадженням новітніх інформаційних технологій задачі автоматизації прийняття рішень в підборі програмістів для широкого кола предметних задач стають все більш актуальними. Глобалізація та діджиталізація світової економіки визначають необхідність для цього використання різноманітного програмного забезпечення.

За даними порталу roadmap.sh специфіка роботи web-програміста безпосередньо пов'язана з глобальною мережею Інтернет [50]. Тому web-програмісту необхідно володіти специфічними для Інтернету мовами розмітки HTML та CSS, та сучасних підходів до верстки та володіти сучасними фреймворками Bootstrap, Wingtail; мовами програмування PHP, JavaScript, Python; сучасними базами даних MySQL, MariaDB, MongoDB; знати та володіти основними фреймворками: Drupal, WordPress, Laravel, Django, Flask, FastAPI; знати та розуміти основні протоколи передачі даних: HTTP, HTTPS, FTP, SSH, UDP; знати та розуміти принципи захищеної передачі даних за протоколами SSL/TLS.

Згідно досліджень [35] найпопулярнішими інструментальними засобами для web розробки є сучасні середовища інтегрованої розробки (IDE), які відіграють важливу роль у процесі веб-розробки, оскільки вони забезпечують розробникам комплексний набір інструментів для написання коду, його відладки, тестування та впровадження. Для мов програмування, таких як JavaScript (JS), PHP та Python,

існує широкий вибір IDE, кожен з яких пропонує різноманітні функції, що задовольняють специфічні потреби веб-розробників.

Visual Studio Code (VS Code) - це легке, але потужне середовище інтегрованої розробки від Microsoft, яке підтримує багато мов програмування, включаючи JavaScript. VS Code пропонує широкий набір розширень, які можуть бути інтегровані для покращення процесу розробки, включаючи підтримку фреймворків, лінтинг коду, автодоповнення та інші корисні інструменти.

WebStorm - IDE від JetBrains, спеціально розроблене для веб-розробки. Воно забезпечує глибоку інтеграцію з сучасними фреймворками JS, такими як React, Angular та Vue.js, пропонуючи автоматичне завершення коду, відладку на льоту та інструменти для рефакторингу. WebStorm також підтримує розробку серверних додатків на Node.js.

PHPStorm - продукт від JetBrains, PHPStorm є лідером серед IDE для розробки на PHP. Він включає всі основні можливості WebStorm для роботи з JavaScript та CSS, а також надає глибоку інтеграцію з PHP, включаючи підтримку всіх основних PHP-фреймворків, таких як Laravel, Symfony та Zend, інструменти для відладки та тестування, а також управління базами даних.

NetBeans - безкоштовне, відкрите середовище розробки, яке підтримує багато мов програмування, включаючи PHP. NetBeans пропонує інтегроване середовище з підтримкою HTML5, JavaScript, CSS, і PHP, забезпечуючи інструменти для відладки, управління проектами та роботи з системами контролю версій.

PyCharm - IDE від JetBrains, спеціалізоване на Python, яке пропонує широкий набір інструментів для професійної розробки на Python, включаючи підтримку веб-розробки з Django, Flask та іншими фреймворками. PyCharm надає потужні інструменти для аналізу коду, відладки, рефакторингу, інтеграції з системами контролю версій та віртуальними середовищами.

Visual Studio Code - завдяки своїй гнучкості та великій кількості розширень, VS Code також є популярним вибором для розробки на Python. Він підтримує інтеграцію з Python через розширення, яке забезпечує підтримку інтелектуального

автодоповнення коду, відладку, управління пакетами та віртуальними середовищами.

Кожне з цих IDE має свої унікальні особливості та переваги, а вибір конкретного інструменту залежить від особистих уподобань розробника, специфіки проєкту та вимог до функціональності.

Для успішного використання мов програмування, технологій та технічних засобів web-розробки, та бути затребуваним у конкурентному середовищі, web-програміст повинен володіти відповідними знаннями, уміннями та навичками (рис. 1.4.).



Рис. 1.4. Структура сучасних знань web-програміста.

Крім практичного досвіду в IT-галузі необхідно дотримуватись загально прийнятих стилів, підходів та вимог до розробки web-додатків:

- реалізовувати сумісність з мобільними пристроями;
- використовувати фреймворки та бібліотеки;

- використовувати основні принципи і методи SEO-оптимізації;
- реалізовувати сумісність з сучасними мультимедіа технологіями.

Варто відзначити, що для досягнення успіху в ІТ галузі web-програмісту не завжди обов'язково мати університетську освіту у галузі комп'ютерних наук чи суміжних дисциплін. Сучасні технологічні та освітні ресурси надають потенційним web-програмістам широкий спектр можливостей для самоосвіти та професійного розвитку, як з університетським дипломом, так і без нього.

Університетська освіта в галузі інформаційних технологій чи комп'ютерних наук надає студентам теоретичну базу, з вивчення алгоритмів, програмування, баз даних та інших фундаментальних аспектів комп'ютерних наук [43].

Важливим аспектом є також той факт, що багато роботодавців у сфері веб-розробки все більше звертають увагу на практичні навички та досвід кандидатів, а не виключно на наявність диплому про вищу освіту. Це підкреслює можливість досягнення успіху в цій області для осіб з різним освітнім та професійним досвідом.

Таким чином, можливість стати веб-програмістом існує як для осіб з університетською освітою, так і для тих, хто вибрав шлях самоосвіти. Головними чинниками успіху в цій сфері є неперервне навчання, адаптація до новітніх технологій, розвиток практичних навичок та активна участь у професійній спільноті.

Виділимо три окремих проблеми щодо ефективного функціонування системи набуття знань для професії web-програміста.

Перший підхід - здобути вищу освіту в галузі ІТ з поглибленим вивченням напрямку веб-розробки. У підсумку здобувач матиме диплом бакалавра (магістра), багато теоретичних знань, але при цьому йому бракуватиме достатньо практичних умінь та навичок.

Другий підхід - здобути знання з напрямку веб-розробки самостійно, за допомогою доступних безкоштовних різноманітних відкритих Інтернет-ресурсів. Недоліком такого підходу є відсутність диплому про вищу освіту, що є перешкодою для просування кар'єрними сходами в галузі ІТ.

Третій підхід - пройти навчання у спеціалізованих навчальних ІТ-центрах, ІТ-академіях за online/offline форматом проведення навчання. Перевагою такого навчання є здобуття практичних навичок та отримання пропозицій від ІТ-компаній зайняти відкриту вакансію.

Web-програміст - це окремий напрям у професії програміста, який займається розробкою сайтів. Крім того, напрям web-програмування в ІТ-компаніях поділяється на три спеціалізації: фронтенд-розробка, бекенд-розробка, фуллстек-розробка. В залежності від зайнятої посади під час працевлаштування в ІТ-компаніях, роботодавець ставить вимоги до якості набутих soft- та hard- скілів [50].

Розвиток персоналу є важливою галуззю знань, яка стосується не тільки ІТ, але й всіх інших галузей науки та бізнесу. Протягом десятиліть багато науковців вивчали цю проблематику, і деякі з них сформуvalи основи сучасного управління персоналом. Розглянемо основних вчених та їх вклад в науку, які сформуvalи теоретичні основи сучасних підходів до управління персоналом.

Фредерик Тейлор (Frederick Taylor) - відомий своєю теорією наукового управління. Він розглядав працівників як частину процесу і намагався оптимізувати їх продуктивність через стандартизацію роботи [55].

Практиком-експериментатором, який виявив важливість соціальних факторів та взаємодії в колективі був Елтон Майо (Elton Mayo), що провів ряд експериментів з персоналом заводу Western Electric в Хоутонії [56].

Вченим, що заклав фундаментальні основи аналізу та управління потребами був Абрахам Маслоу (Abraham Maslow) - він розробив ієрархію потреб, що описує, які потреби мотивують людей [57].

Визначив та формалізував фактори мотивації працівників був Герцберг Фредерик (Frederick Herzberg) [58]. Науковцем, який почав розглядати управління як мистецтво, а не як науку був Пітер Друкер (Peter Drucker) - його підходи стали основою для сучасних методів управління в галузі ІТ [59].

Формування загальних наукових підходів до управління персоналом стали основою до розвитку науки управління персоналом в галузі інформаційних

технологій. Розглянемо науковців, які сформували основи до сучасних підходів управління персоналом в ІТ.

Основний науковий вклад Тома ДеМарко (Tom DeMarco) і Тімоті Лістера (Timothy Lister) полягає у вивченні того, як фактори середовища робочого місця впливають на продуктивність програмістів [45].

Кент Бек (Kent Beck) є одним з авторів гнучкої методології розробки (Agile) і Extreme Programming (XP). Бек розглядає програмування як командну діяльність і акцентує увагу на командній співпраці. [60] Гнучкі методології наразі є “підходом по замовчуванню”, який наразі використовується в абсолютній більшості проєктів в галузі ІТ.

Основа сучасних підходів до управління проєктами та персоналом належить Джефу Сазерленду (Jeff Sutherland) [19].

Фундаментальні праці в галузі психології програмування та управління ІТ-проєктами належать Джеральду Вайнбергу (Gerald Weinberg) [61].

Суттєвої уваги до процесів управління командами приділив Мартін Фаулер (Martin Fowler) - відомий своїми роботами з об'єктно-орієнтованого проєктування та архітектури програмного забезпечення [62].

В умовах інтенсифікації світової економіки та стрімкого розвитку інформаційних технологій все більшої популярності набуває підхід “економної розробки”, який дозволяє оперативно перевіряти бізнес гіпотези з використанням обмежених ресурсів. Такий підхід вимагає особливих методів до управління проєктами та персоналом. Одним з відомих науковців, які вивчає Lean Development є Ерік Ріс (Eric Ries) [47].

Окремої уваги заслуговує вклад науковців в розробку сучасних методів оцінювання та грейдуння персоналу. Зокрема, одним з основоположників був Максвелл Малц (Maxwell Maltz), який виклав основи принципів самооцінки та саморозвитку ІТ-фахівців в фундаментальній праці “Психо-кібернетика” [63].

Вплив емоційного інтелекту на процеси оцінювання персоналу вивчає Деніел Гоулман (Daniel Goleman) [64]. Його ідеї активно використовуються в методах оцінки та розвитку персоналу.

Оснoву використання оцінювання як елементу керування персоналом заклав Пітер Друкер (Peter Drucker), який в своїй праці “Практика менеджменту” розглядав оцінку як інструмент управління та наголошував на важливості вимірювань та об'єктивності [59].

Принципи використання балансових карток для кількісного та якісного оцінювання продуктивності та ефективності заклали Роберт Каплан (Robert Kaplan) та Девід Нортон (David Norton) [34].

Фундаментальне вивчення природи роботи та мотивації здійснив Дуглас Макгрегор (Douglas McGregor), який представив дві теорії (Теорія X та Теорія Y), які мають прямий вплив на підходи до оцінки працівників [66].

Однією з наукових праць, яка є основою для сучасних підходів до оцінювання та грейдуння персоналу є “A Handbook of Human Resource Management Practice” за авторством Майкла Армстронга (Michael Armstrong) [37].

Українська наукова школа в галузі інформаційних технологій тісно пов'язана з закордонною. Зокрема, Україна відіграє важливу роль в розвитку інформаційних технологій та кібернетики на пострадянському просторі та світі загалом. Величезну роль в становленні та сучасному функціонуванні ІТ науки відіграє наукова школа, сформована академіком В. М. Глушковим [68, 69].

Його наукова школа – це одне з основних ядер розвитку кібернетики та прикладної математики в Україні. Ця школа надала світові численні інноваційні рішення у сфері теорії автоматів, математичної логіки, системного аналізу, інтелектуальних систем, систем автоматизованої обробки інформації, та ряду інших наукових напрямків.

В галузі досліджень процесів управління персоналом існують наступні наукові школи, представники якої здійснюють наукові дослідження та розвивають теоретичні та практичні аспекти управління персоналом. Зокрема А. В. Колот [81], О. І. Васильєва [82], Н. П. Борецька.

Грунтовний теоретичний і практичний фундамент, сформований як вітчизняними так і закордонними науковцями дозволили створити передумови до

появи окремого класу інформаційних технологій та систем - інформаційних система управління персоналом (HRIS).

В науковому контексті інформаційні технології автоматизації процесів управління персоналом ІТ-компаній базуються на комбінації теорій управління, соціології, психології, математики та інформатики.

Стан ринку інформаційних технологій має великий вплив на стратегічне планування та прогнозування для сфери бізнесу з ряду причин:

- швидкість змін: в ІТ-секторі технології постійно оновлюються, якщо підприємство не відстежує ці зміни, воно може втратити конкурентні переваги;
- інновації: інновації в ІТ можуть різко змінити ринкову динаміку, нові продукти та рішення можуть перетворити існуючі бізнес-моделі або створити нові ринкові ніши;
- залежність від технологій: більшість компаній зараз залежать від ІТ, щоб оптимізувати свої операції, забезпечити краще обслуговування клієнтів та підтримувати ефективність;
- кібербезпека: зі зростанням кіберзлочинності стан ринку ІТ може впливати на загрози безпеці для бізнесу, підприємства повинні розуміти поточні та майбутні ризики, щоб правильно вкладати в захист своїх активів;
- інвестиції: оцінка стану ринку ІТ допомагає компаніям визначити, куди краще вкладати кошти, це може бути вкладення в нові технології, партнерства або навіть поглинання та придбання;
- регулятивне середовище: у деяких країнах регулювання може суттєво впливати на ІТ-бізнес, оцінка цих трендів є важливою для розуміння можливих обмежень або вимог;
- споживчі очікування: технологічні звички споживачів змінюються зі швидкістю введення нових технологій, розуміння цих змін допомагає компаніям адаптуватися до змінюючихся потреб споживачів.

Враховуючи стан українського та світового ринків ІТ, існуючі економічні та технологічні тренди - можна зробити висновок, що інтелектуальні інформаційні технології (ІІТ) будуть продовжувати відігравати критично важливу роль у

сучасному цифровому світі, особливо у контексті автоматизації прийняття рішень в процесах управління персоналом ІТ-компаній.

Ринок праці в галузі інформаційних технологій має ряд особливостей, які відрізняють його від інших ринків праці:

- висока динамічність: технології швидко змінюються, і вимоги до спеціалістів в галузі ІТ також постійно адаптуються, це може вимагати від фахівців постійного навчання та самовдосконалення;
- дефіцит кваліфікованих кадрів: високий попит на ІТ-фахівців у багатьох країнах призводить до дефіциту висококваліфікованих програмістів, що може підвищувати зарплати в цій галузі;
- широкий спектр спеціалізацій: від програмування до кібербезпеки, від аналізу даних до дизайну галузь ІТ пропонує величезний діапазон спеціалізацій;
- важливість софт скілів: незважаючи на технічний характер професії, софт скіли, такі як комунікабельність, критичне мислення та здатність працювати в команді, стають все важливішими;
- безперервний розвиток та безперервна освіта: оскільки технології неперервно розвиваються, професійці ІТ повинні бути готові до постійного навчання та адаптації до нових інструментів і підходів.

Таким чином, в порівнянні з іншими ринками праці, ІТ-галузь вимагає більш високого рівня спеціалізації, адаптивності до швидко змінюваних умов та гнучкості у виборі місця та формату роботи.

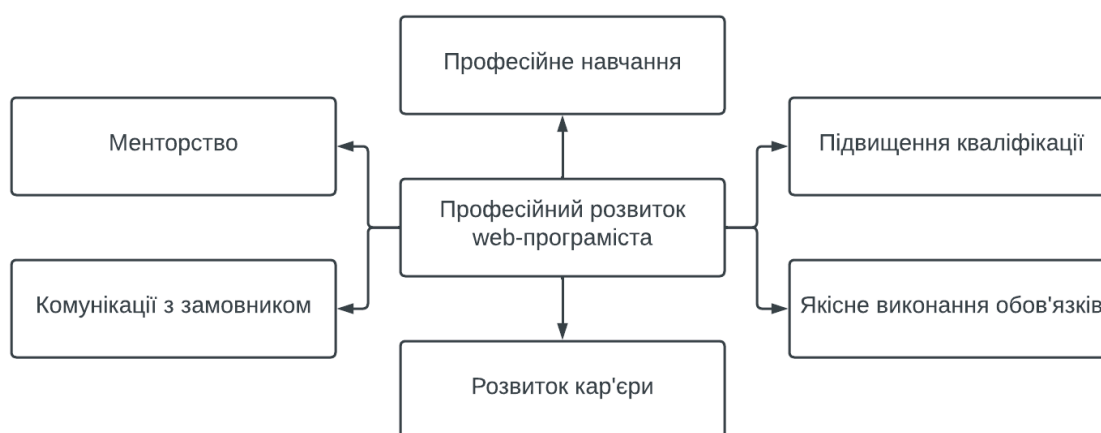


Рис. 1.5. Вимоги до безперервного професійного розвитку web-програміста.

Для того, щоб забезпечити максимальний рівень ефективності функціонування ІТ компаній, необхідно враховувати актуальні вимоги до скілів, яким має відповідати сучасний веб-девелопер.

Сучасний веб-розробник повинен володіти широким спектром технічних та м'яких навичок (рис. 1.6.), щоб ефективно впоратися з викликами розвитку та підтримки веб-застосунків у швидко змінюваному технологічному середовищі.

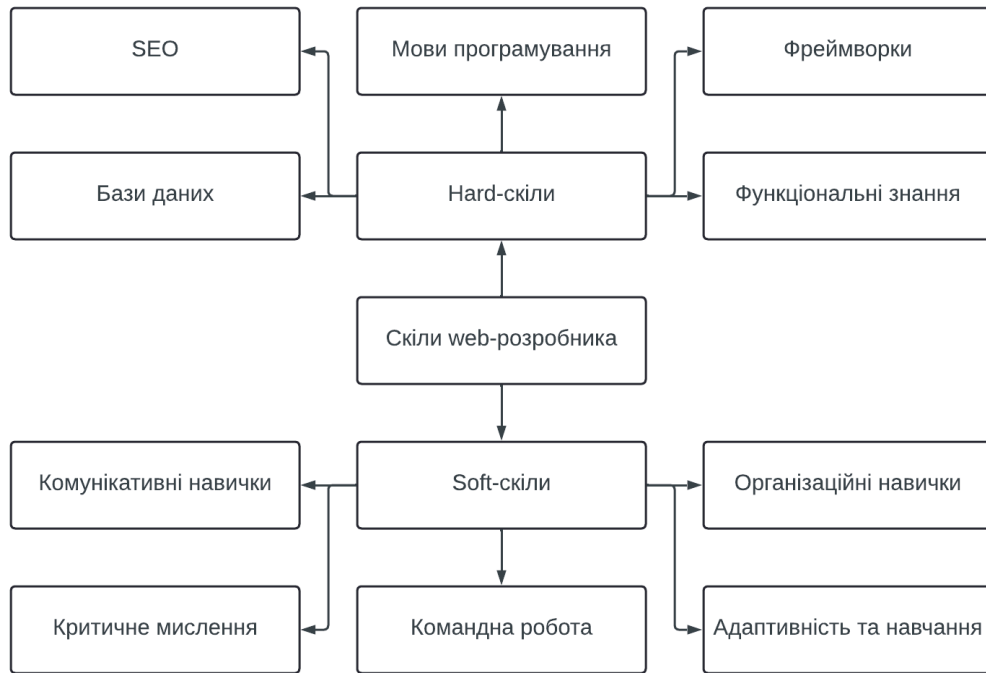


Рис. 1.6. Необхідні скіли сучасного web-розробника.

Враховуючи швидкий розвиток технологій у сфері веб-розробки, сучасному веб-розробнику важливо постійно оновлювати свої навички та знання, щоб відповідати поточним тенденціям і вимогам ІТ-ринку.

1.5. Постановка задачі на основі аналізу процесів автоматизації ІТ-рекрутингу web-програмістів

Наукові роботи демонструють розробку моделей на основі NLP/LLM та мультиагентних підходів для автоматичного скринінгу резюме та ранжування кандидатів. Окремо виділяються пояснювані системи (Explainable AI, XAI), які

підвищують прозорість оцінки і довіру користувачів. Використання мультиагентних систем забезпечує масштабованість, модульність та гнучкість інтегрованих рішень, що потенційно може бути застосовано для оптимізації відбору web-програмістів [13–15].

Проведений аналіз процесів автоматизації IT-рекрутингу web-програмістів свідчить про те, що сукупність сучасних наукових підходів і комерційних рішень створює передумови для розроблення спеціалізованих інтелектуальних систем відбору web-програмістів. Водночас існують обмеження існуючих платформ щодо адаптації до специфіки IT-вакансій, що обґрунтовує необхідність розробки спеціалізованої інформаційної технології з інтеграцією NLP, LLM, мультиагентних моделей та ХАІ для ранжування кандидатів.

Таким чином сукупність сучасних моделей NLP, LLM та мультиагентних підходів створює надійну наукову основу для розробки інтелектуальних систем рекрутингу, зокрема для відбору web-програмістів [100]. Водночас існуючі підходи переважно розробляються як загальні рішення і не завжди враховують специфіку IT-посад, що обґрунтовує потребу у розробленні спеціалізованої інформаційної технології, адаптованої до вимог сучасних вакансій web-розробки. Порівняємо підходи до інтелектуального відбору кандидатів на посади web-програмістів (таблиця 2).

Аналіз таблиці 1.2. демонструє, що кожен із розглянутих підходів має власні сильні сторони та обмеження. Методи NLP ефективні для швидкого вилучення ключових компетенцій з резюме та описів вакансій, проте обмежені у врахуванні складного контексту та взаємозв'язків між навичками [101]. Великі мовні моделі (LLM) дозволяють здійснювати глибокий семантичний аналіз і ранжування кандидатів з урахуванням контексту, а також формувати пояснювані рекомендації, однак потребують значних обчислювальних ресурсів та складніші в інтеграції [102-108]. Мультиагентні системи забезпечують модульність та масштабованість процесів рекрутингу, дозволяючи паралельно виконувати оцінку технічних та когнітивних компетенцій, взаємодіяти з кандидатами через чат-боти або

віртуальних асистентів, але вимагають ретельного проектування координації агентів [110].

Таблиця 1.2. Порівняння підходів до інтелектуального відбору кандидатів на посади web-програмістів

Параметр	NLP (обробка природної мови)	LLM (великі мовні моделі)	Мультиагентні системи
Мета застосування	Вилучення ключових компетенцій, аналіз резюме та вакансій	Глибоке семантичне та контекстуальне розуміння тексту, оцінка складних взаємозв'язків	Розподіл завдань між агентами: аналіз, оцінка, ранжування, взаємодія з кандидатами
Технічні особливості	Токенізація, лексичний аналіз, розпізнавання іменованих сутностей, TF-IDF, тематичне моделювання	Векторні подання тексту (embeddings), трансформерні архітектури, генеративні можливості	Модульна архітектура, агенти виконують окремі підзадачі, координація між агентами
Переваги	Легка інтеграція в існуючі ATS/HRIS, швидка обробка великих обсягів даних	Висока точність оцінки відповідності, можливість роботи з контекстом, генерація пояснень	Масштабованість, гнучкість, можливість паралельного виконання завдань, інтеграція різних типів аналізу
Обмеження	Обмежена здатність враховувати контекст, не враховує складні взаємозв'язки	Високі обчислювальні ресурси, складність інтеграції в корпоративні системи	Складна реалізація, потребує чіткої координації агентів і стандартизації протоколів
Приклади застосування у відборі web-програмістів	Вилучення знань про мови програмування, фреймворки, досвід роботи	Ранжування кандидатів за рівнем відповідності вакансії, пояснювані рекомендації	Координація тестів на технічні навички, симуляцій, відео-інтерв'ю та оцінювання soft skills

З огляду на це, оптимальним є комбіноване використання підходів, де NLP використовується для первинного скрінінгу та вилучення ключових компетенцій, LLM для семантичного аналізу, ранжування та формування пояснень, а

мультиагентна архітектура забезпечує інтеграцію різних модулів і автоматизацію взаємодії з кандидатами. Такий підхід дозволяє створити спеціалізовану інтелектуальну систему відбору web-програмістів, що забезпечує високий рівень об'єктивності, прозорості та ефективності процесу рекрутингу [111-112].

Відповідно до проведеного аналізу для оцінки релевантності кандидатів і визначення відповідності їхнього профілю вимогам вакансії пропонуються спеціалізовані моделі штучного інтелекту [113]. До таких моделей належать алгоритми семантичної та векторної схожості, інтегровані NLP-модулі для автоматичного вилучення компетенцій та текстових характеристик, а також системи на базі Explainable AI (XAI), які забезпечують прозорість і обґрунтованість прийнятих рішень [114]. Наявність цих досліджень підтверджує, що розробка інтелектуальних інформаційних систем для відбору кандидатів є не лише теоретичною концепцією, а й активною науковою та прикладною сферою, що включає розробку прототипів, експериментальні дослідження та практичну інтеграцію у HR-процеси [115].

Інтелектуальні інформаційні системи для відбору кандидатів, зокрема web-програмістів, існують як у вигляді комерційних продуктів, таких як Workable, Personio, Canditech та ін., так і у вигляді науково-дослідних рішень, що використовують елементи штучного інтелекту. Ці системи дозволяють автоматизувати процеси первинного скринінгу, оцінки компетенцій та ранжування кандидатів, підвищуючи ефективність і об'єктивність HR-процесів у IT-сфері. Наявність як практичних, так і дослідницьких рішень підтверджує, що використання інтелектуальних інформаційних систем у відборі персоналу є актуальною, ефективною та перспективною задачею як для досліджень, так і для впровадження [116-119].

Таким чином, аналіз сучасних підходів обґрунтовують необхідність розробки інтелектуальної інформаційної системи, яка поєднає методи NLP, LLM та мультиагентних систем для автоматизованого скринінгу, оцінки компетенцій і ранжування кандидатів на позиції web-програмістів [124]. Саме це стає науковою основою для постановки задачі дисертаційного дослідження.

Для вирішення цієї проблеми в дисертаційній роботі ставляться та розв'язуються такі задачі:

- аналіз сучасних методів та засобів розробки інтелектуальних інформаційних систем; розробка структури та принципів побудови інформаційної технології для підготовки та рангового відбору web-програмістів; розробка методів та подання знань в багаторівневому середовищі;
- інтеграція переваг підходів у моделі інтелектуальної інформаційної системи: NLP (обробка природної мови), LLM (великі мовні моделі), мультиагентні системи;
- реалізація даного інформаційного середовища для підготовки та відбору певного рангу web-програмістів;
- розробка та реалізація експериментальної версії інтелектуальної інформаційної системи, що складається з окремих модулів: навчання, тестування та відбору;
- проведення експериментів по визначенню дидактичної ефективності розробленої інформаційної технології.

Висновки до першого розділу

За даними ряду досліджень [56, 61, 62, 63] web-розробник це затребувана професія в сучасному світі інформаційних технологій. Для того, щоб web-програмування було якісним та ефективним, web-програміст повинен володіти наступними навичками та знаннями:

- Володіння HTML, CSS, і JavaScript для розробки фронтенду;
- Знання серверних мов програмування: PHP, Python, Node.js;
- Досвід роботи з фронтенд фреймворками (React, Angular, Vue.JS);
- Знання бекенд фреймворків (FastAPI, Django, Drupal, Symfony, Laravel);
- Вміння працювати з реляційними (PostgreSQL, MySQL, MariaDB) та нереляційними (MongoDB, Redis, Elasticsearch) базами даних;

- Знання принципів проектування баз даних та оптимізації запитів;
- Володіння системами контролю версій, зокрема Git;
- Знання та досвід застосування агільних методологій розробки (Agile, Scrum, Kanban);
- Основи DevOps і неперервної інтеграції/доставки (CI/CD);
- Розуміння загальних веб-вразливостей (SQL-ін'єкції, XSS) і методів їх запобігання;
- Аналітичні здібності та вміння вирішувати складні задачі;
- Комунікативні здібності, вміння працювати в команді, управління часом.

РОЗДІЛ 2.

ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ОСНОВ WEB-РОЗРОБКИ

2.1. Зміст навчання програмістів

В Міжнародному економіко-гуманітарному університеті імені академіка С. Дем'янчука згідно освітньої програми “Інженерія програмного забезпечення, Інтернет речей” розроблена модель навчального плану підготовки здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальності 121/F2 Інженерія програмного забезпечення (рис. 2.1).

На кафедрі інформаційних систем та обчислювальних методів опрацьовані дистанційні курси з дисциплін “Web-програмування”, “Web-дизайн”, “Web-технології”, “Web-розробка”, “Управління ІТ-проектами”.

Управління проектами в ІТ-компаніях - це складний і багатогранний процес, що включає планування, виконання, моніторинг та завершення ІТ-проектів. Вибір правильної методології управління в ІТ-компаніях є важливим, оскільки він визначає, як будуть вирішуватися ключові завдання, розподілятися людські, технічні та програмні ресурси та виконуватися контроль якості.

Вибір правильної методології впливає на успішність проекту, ефективність команди, задоволення клієнтів, та остаточну якість продукту. Не існує універсальної методології, яка підійде всім проектам, тому вибір залежить від специфічних вимог проекту та кваліфікованої команди, у відповідності до специфіки та особливостей бізнес-процесів клієнтського бізнесу, композиції команди, очікуваних строків реалізації проекту, обраного стеку технології для реалізації, очікуваного переліку можливих інтеграцій.

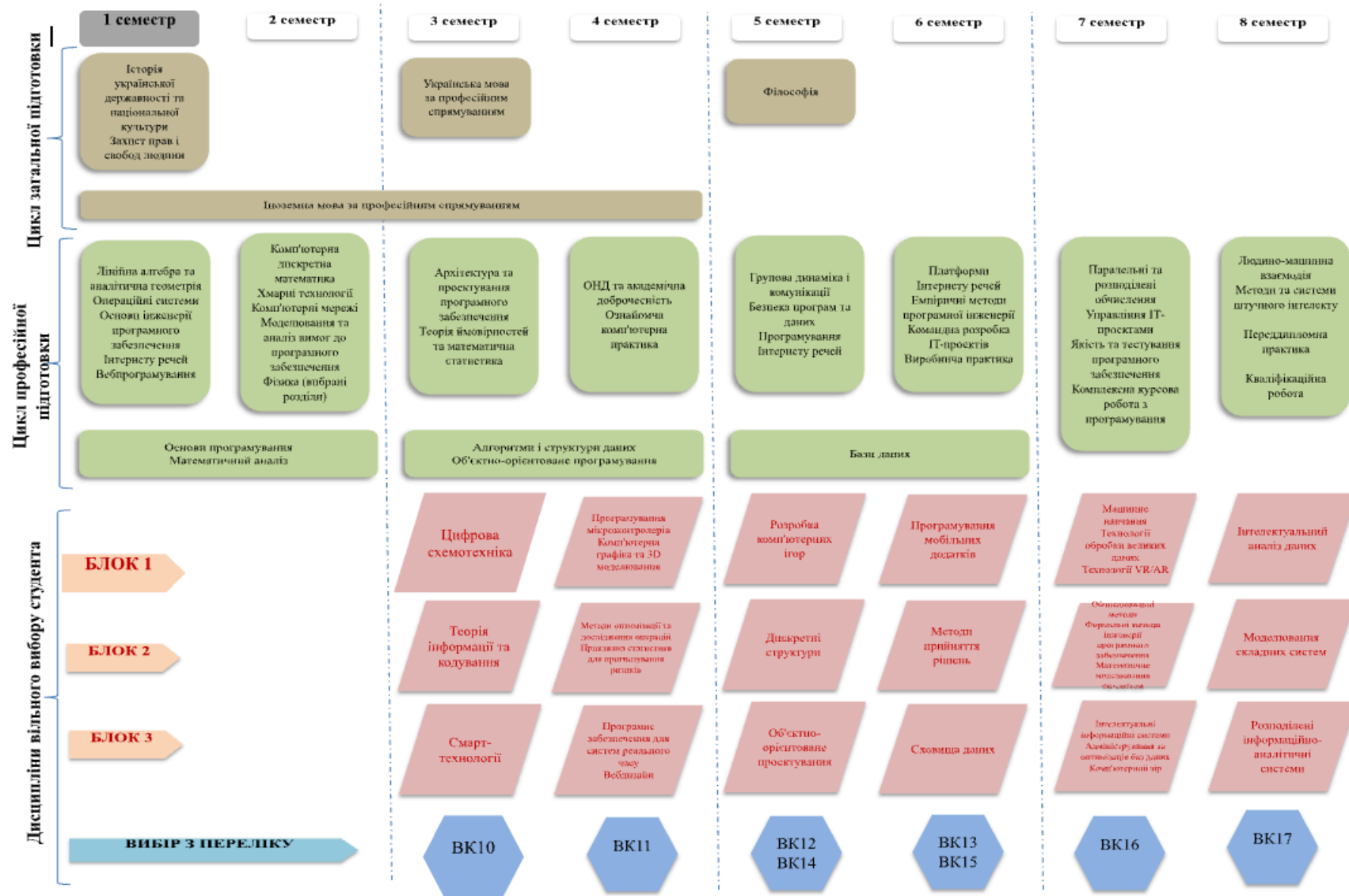


Рис. 2.1. Схема неперервної освіти здобувачів спеціальності 121/F2 Інженерія програмного забезпечення в МЕГУ.

Основними методологіями, які застосовуються в ІТ-галузі для управління проектами є наступні гнучкі методології (Agile), зокрема:

- Scrum: це ітеративний підхід, що включає короткі цикли розробки (спринти), що дозволяє команді швидко адаптуватися до змін у проекті.
- Kanban: орієнтований на поточну роботу команди та її оптимізацію через візуалізацію робочого процесу.
- Extreme Programming (XP): фокусується на покращенні якості програмного забезпечення та відповідності потребам замовника.
- Водоспадна модель (Waterfall) є менш гнучкою, це традиційний, секвенційний підхід, де кожен етап проекту (аналіз вимог, проектування, розробка, тестування, впровадження) є передумовою для наступного, вона забезпечує чітку структуру.

Автором проаналізовано методи управління ІТ-проектами та визначено їх переваги та недоліки (табл. 2.1):

Таблиця 2.1. Порівняльний аналіз переваг і недоліків методологій.

Методологія	Переваги	Недоліки
Scrum	<ul style="list-style-type: none"> - гнучкість та адаптивність; - прозорість; - покращення співпраці; - контроль якості; - ефективність через ітеративний підхід. 	<ul style="list-style-type: none"> - залежність від команди; - складність для великих проєктів; - можливість перевантаження команди.
Kanban	<ul style="list-style-type: none"> - простота впровадження; - висока гнучкість; - оптимізація робочого навантаження; - прозорість процесів; - підвищення продуктивності та ефективності. 	<ul style="list-style-type: none"> - відсутність чітких часових рамок; - менша підходящість для складних проєктів; - залежність від самодисципліни команди; - ризик ігнорування стратегічного планування; - потреба в постійному моніторингу.

Waterfall	<ul style="list-style-type: none"> - простота та структурованість; - легкість управління; - документування; - визначеність. 	<ul style="list-style-type: none"> - негнучкість; - ризик при зборі вимог; - затримка зворотного зв'язку; - ізоляція етапів;
-----------	---	--

На основі проведеного аналізу обрано характеристики ключових методологій для їх застосування в управлінні ІТ-проєктами.

Таблиця 2.2. Порівняльний аналіз основних характеристик методологій.

Методологія	Гнучкість	Ефективність	Оперативність	Ризики	Управління
Scrum	Висока	Висока	Висока	Середні	Складне
Canban	Висока	Середня	Середня	Середні	Складне
Waterfall	Низька	Низька	Низька	Високі	Просте

З результатів порівняльного аналізу методологій управління ІТ-проєктами, жодна з методологій не забезпечує ефективного покриття всіх категорій на високому рівні, отже не є універсальною для проєктів різного розміру, виду та командного складу (табл. 2.2).

Для здійснення ефективного управління ІТ-проєктами автором створено гібридну модель управління проєктами, що базується на ефективних принципах Scrum та Kanban, а також окремих компонентах методології Waterfall (рис. 2.2). Така модель є корисною в середовищах, де вимоги систематично змінюються, а процеси управління є чітко структурованими.

Гібридна модель управління ІТ-проєктами ґрунтується на ітеративному підході, ретроспективі та неперервному вдосконаленні (Scrum), візуалізації та управлінні потоком робіт (Kanban), гнучкості та адаптивності, формалізації чекпойнтів, гібридній ізоляції потоків.

Пропонований варіант гібридної моделі управління ІТ-проєктами на базі цілісного підходу до проблем підготовки та відбору web-програмістів полягає у

формування у здобувачів soft- і hard- скілів, а у відповідності з ними дозволяє вдосконалювати інформаційні технології.

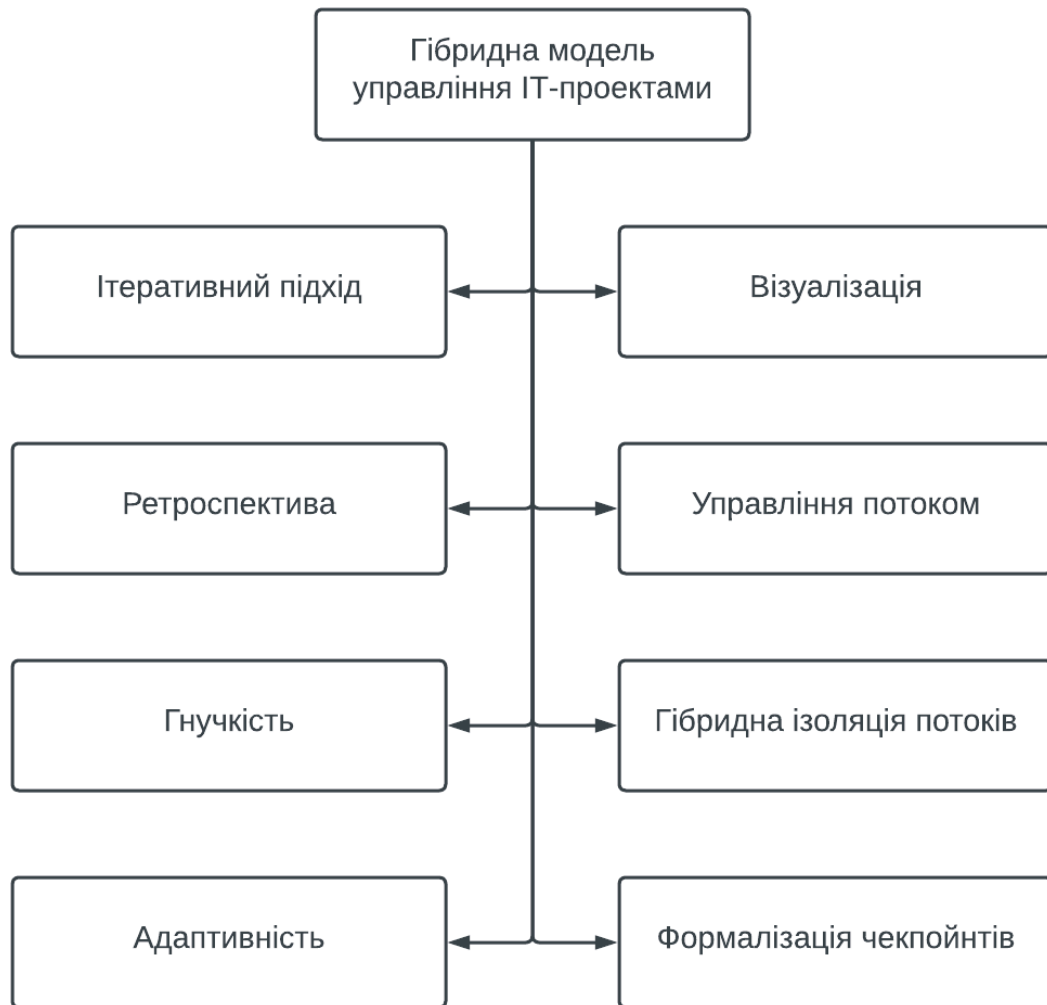


Рис. 2.2. Гібридна модель управління ІТ-проектами.

Така модель дозволить:

- підвищити продуктивність та ефективність роботи команди;
- гнучко управляти змінами у проєкті;
- оптимально використовувати обчислювальні ресурси;
- знизити ризики та затримки з web-розробки;
- поліпшити якість кінцевого програмного продукту.

Управління ІТ-проектом на основі розробленої гібридної методології являє наступні фази (рис. 2.3):

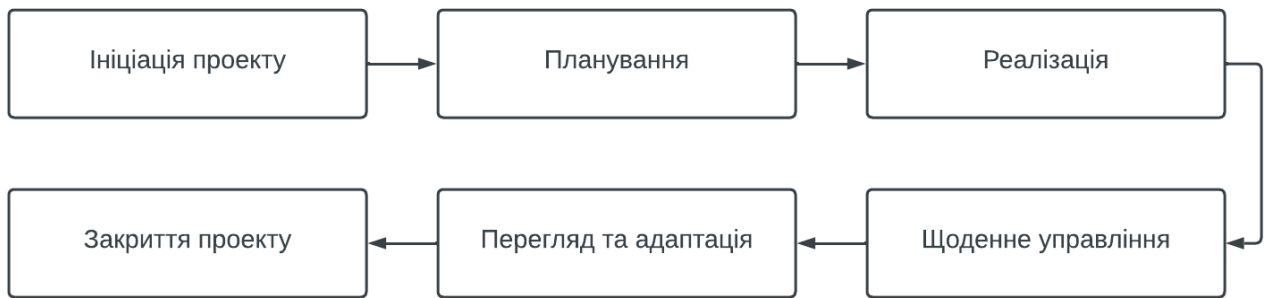


Рис. 2.3. Розроблені фази управління ІТ-проектом.

Баланс між структурованістю та гнучкістю дозволяє проєктній команді пристосовуватися до змін, одночасно зберігаючи чітку структуру та напрямок проєкту. Позитивна сторона створеної моделі дозволяє забезпечити високу продуктивність та ефективне управління ресурсами, що є важливим у динамічному ІТ-середовищі.

2.2. Загальні принципи web-розробки, методологічні та наукові аспекти знань для відбору web-програмістів

Веб-програмування сьогодні - це одна з найбільш перспективних сфер діяльності ІТ-компаній. У сучасному світі, будь-який бізнес потребує візуальної ідентифікації в мережі Інтернет, корпоративних мережах, глибокої системної інтеграції для забезпечення конкурентноспроможності на світовому ринку.

Завданнями web-програмістів є:

- створення web-сайтів, додатків, сервісів та гібридних мобільних додатків;
- верстка та інтеграція UI/UX
- інтеграція з базами та сховищами даних;
- системна інтеграція додатків із сторонніми сервісами;
- організація безпечного функціонування додатків та сервісів;

- активна взаємодія із суміжними фахівцями, зокрема з галузевим маркетингом, комерцією, для реалізації їх прикладних задач.

У веб-розробці існує поділ на дві основні області: фронтенд (клієнтська сторона) та бекенд (серверна сторона). Кожна з цих областей використовує унікальний набір технологій та інструментів, які в сукупності формують стек веб-розробки.

Ефективна веб-розробка вимагає глибокого розуміння як фронтенд, так і бекенд технологій, а також вміння інтегрувати їх для створення безпечних, швидких та відповідних веб-додатків.

Насьогодні не існує єдиних теоретичних основ та загальноприйнятих єдиних галузевих регламентів та стандартів, які б дозволяли однозначно регулювати на світовому рівні питання оцінювання професійних умінь та навичок ІТ фахівців в галузі web-розробки.

Натомість, існують міжнародні стандарти, системи сертифікацій, загальноприйняті але не строго регламентовані підходи, які покривають частину компетенцій та дозволяють класифікувати та оцінити знання, вміння та навички web-розробника. Методологічною основою web-розробки служить науковий метод дослідження різних мов програмування, фреймворків, бібліотек, наборів компонентів, платформ.

Методологічною основою web-розробки служить науковий метод дослідження різних мов програмування, фреймворків, бібліотек, наборів компонентів, платформ. Результати наукових розробок в галузі ІТ - це основа методологічних принципів web-програмування. Методологія відбору web-програмістів заснована на відомих галузевих стандартах. Методологічною основою відбору web-програмістів служить науковий метод дослідження всіх існуючих ключових стандартів, таких як PSR, W3C, 1Z0-882, ISO, тощо.

Таким чином, основою системи відбору web-програмістів повинні стати сформовані на комплексному науковому підході вимоги, що включають в себе вимоги до:

- розуміння базових технологій та протоколів;

- розуміння алгоритмів;
- розуміння мов програмування та фреймворків;
- розуміння технологій роботи з базами даних;
- розуміння типових підходів та практик до вирішення класичних задач;
- володіння технологіями і методиками командної взаємодії;
- володіння комунікативними навичками.

Така кількість професійних скілів повинна забезпечити командну підтримку ІТ-проєкту.

Сукупність ключових індустріальних стандартів web-розробки викладено в Додатку А.

Одним з ключових стандартів для підготовки web-програмістів є фреймворк Drupal.

Acquia Certification [51] є одним із провідних програм сертифікації для Drupal, фреймворку та системи управління контентом відкритого коду.

Для формування вимог до знань, умінь та навичок фахівців за напрямом бекенд-розробки з використанням фреймворку CMF Drupal було використано Acquia Certified Drupal 10 Backend Specialist Study Guide [52].

Для покриття знань, умінь та навичок по роботі з базами даних було використано стандарт сертифікації MySQL 5.6 Developer (1Z0-882)[53]

Для покриття знань, умінь та навичок по роботі з проєктами було використано стандарт сертифікації ICAgile Certified Professional [54].

Сукупність таких знань дозволяє визначити професійну придатність web-програміста. Наведемо структуру компетенцій для профілю Drupal Backend Developer (рис. 2.4).

Дана структура є загальною і обов'язковою не залежно від поточного професійного рівня розробника, і регулюється технологічними вимогами щодо фреймворків, мов програмування, типових підходів, кращих практик розробки, тестування та впровадження ІТ-рішень .

Усі ці знання усіляко сприятимуть командній розробці ІТ-проєктів.

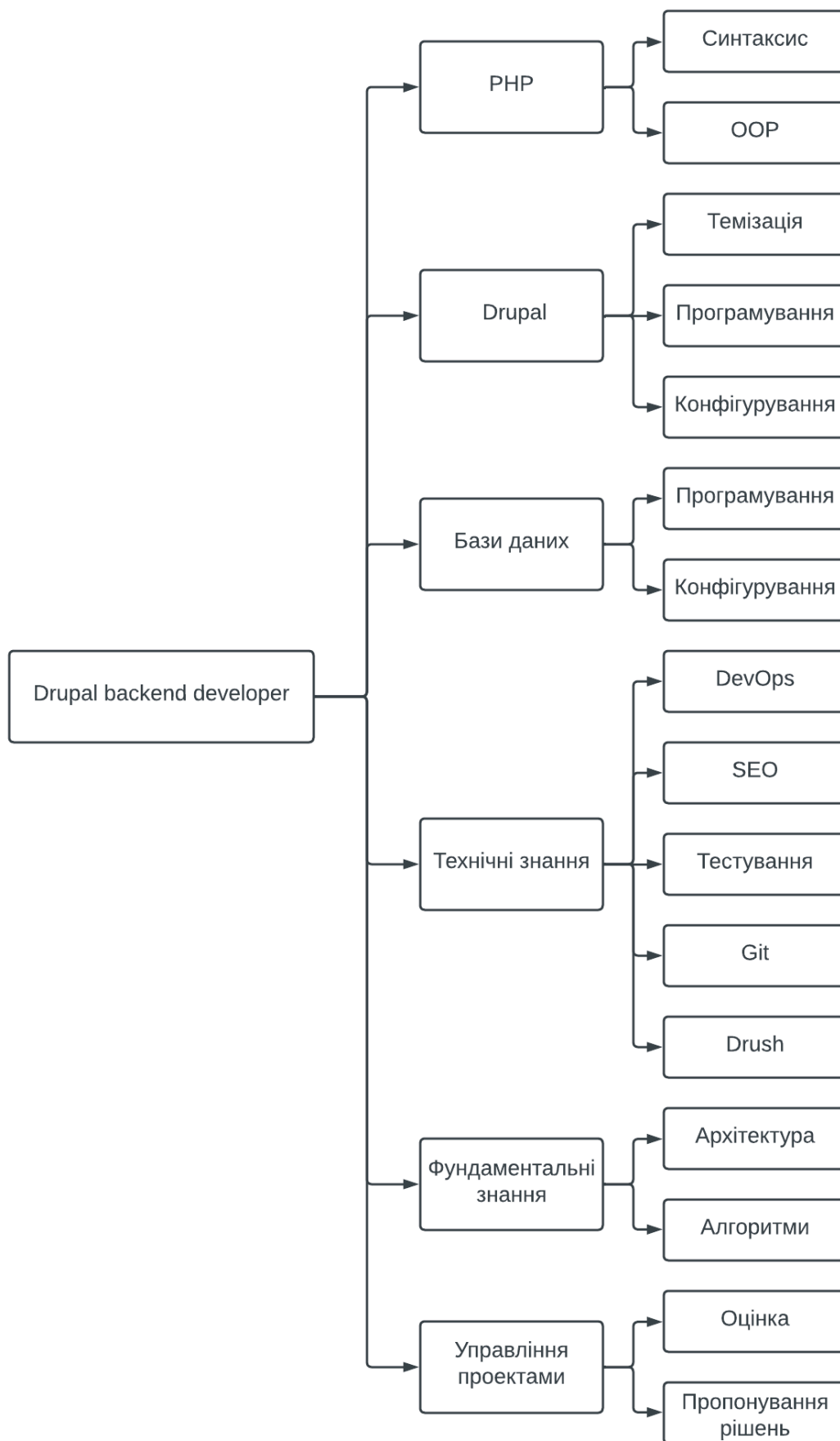


Рис. 2.4. Структура знань для профілю Drupal Backend Developer.

Розроблена автором структура скілів для профілю Drupal Backend Developer є валідною для різних рівнів професійного розвитку і використовується в градації професійних рівнів web-програмістів. Наведемо систему градації професійних рівнів в ІТ-компанії (рис. 2.5).

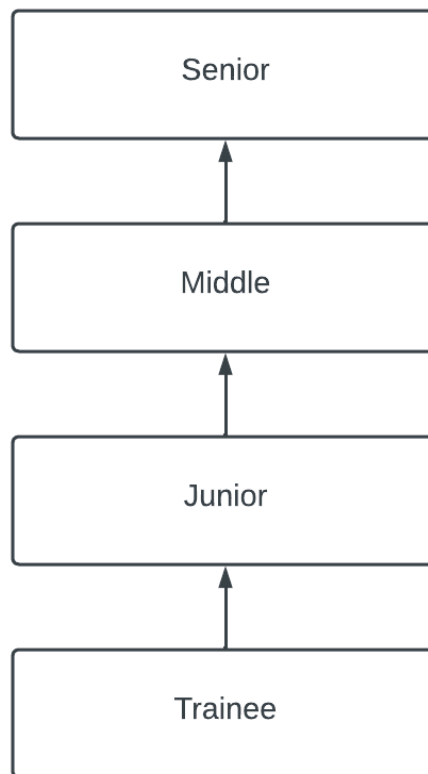


Рис. 2.5. Вертикаль розвитку для профілю Drupal Backend Developer в ІТ-компанії.

Градація професійних рівнів web-програмістів в ІТ-компаніях створює структуру для оцінки та розвитку технічних навичок, професійного досвіду та вмінь співпрацювати в команді. Професійні рівні включають Trainee, Junior, Middle та Senior, кожен з яких має свої характеристики та вимоги.

Подальший розвиток після рівня Senior є нелінійним, специфічним та індивідуальним.

Таблиця 2.3. Характеристика професійних рівнів web-програмістів в ІТ-компанії.

Рівень	Досвід	Навички	Роль у проєкті	Відповідальність
Trainee	Часто є студентами або випускниками без досвіду роботи.	Основні знання у вибраній області, такі як програмування, тестування чи системне адміністрування.	Працюють під керівництвом більш досвідчених колег, виконують прості завдання, активно вчаться та набираються досвіду.	Мінімальна, зосереджені на навчанні та розвитку своїх навичок.
Junior	1-2 роки	Добрі базові технічні знання та деякі практичні навички.	Виконують відносно прості завдання під наглядом більш досвідчених колег.	Відповідальні за свої конкретні завдання, починають працювати більш самостійно, але все ще потребують керівництва.
Middle	2-5 років	Глибоке розуміння технічних аспектів, здатність вирішувати складні завдання.	Активна робота над основними частинами проєкту, можуть починати керувати менш досвідченими колегами.	Відповідають за значну частину проєкту, часто вносять внесок у технічні рішення
Senior	Понад 5 років	Високий рівень технічних знань, стратегічне мислення, здатність вести проєкт або команду.	Лідирують у середніх проєктах, розробляють архітектуру простих проєктів та вирішують складні технічні проблеми.	Велика відповідальність за успіх проєкту, керівництво командою, менторство молодших спеціалістів.

Кар'єрний ріст в ІТ-галузі вимагає безперервної освіти, розвитку технічних навичок, навичок комунікації, управління проєктами, здатності працювати в команді та адаптуватися до змін у технологіях і методологіях.

Баланс між soft скілами (м'якими навичками) та хард скілами (технічними навичками) ІТ-фахівця змінюється в залежності від рівня професійного розвитку, починаючи від Trainee до Senior рівня. Розглянемо цю динаміку в табл. 2.4:

Таблиця 2.4. Порівняння софт і хард скілів ІТ-фахівця.

Рівень	Софт скіли	Хард скіли
Trainee	Навички навчання, відкритість до нового знання, комунікація та здатність працювати в команд	Акцент робиться на набуття базових технічних знань і навичок. Стажери фокусуються на вивченні мов програмування, інструментів, технологій, а також основ програмування та системного аналізу.
Junior	Комунікативні навички, здатність приймати відгуки та навички ефективної роботи в команді стають важливішими. Важливо також вміння ефективно вирішувати проблеми.	Продовжують розвивати свої технічні знання, починаючи працювати над реальними проєктами, але все ще потребують певного рівня нагляду.
Middle	Критичне мислення, здатність керувати часом та проєктами, а також навички лідерства починають набувати більшої ваги. Middle фахівці часто починають виступати в ролі менторів для молодших колег.	Мають глибокі знання у своїй сфері та можуть самостійно вирішувати складні технічні завдання.
Senior	На цьому етапі софт скіли є критично важливими. Вони включають сильні лідерські якості, вміння управляти командою, розвинуті комунікативні навички для взаємодії з клієнтами та стейкхолдерами, а також вміння ефективно вирішувати конфлікти.	Мають широкі та глибокі технічні знання, вони відповідають за високорівневе планування та стратегічні технічні рішення в проєктах.

Розвиток програмістів, зокрема web-програмістів в ІТ-галузі вимагає як систематичного покращення технічних знань, так і розвитку soft скілів, які допомагають управляти ІТ-проєктами, командною розробкою та ефективно

взаємодіяти з клієнтами та колегами. Цей баланс є ключовим для повноцінного професійного розвитку в ІТ-галузі.

2.3. Методика побудови інтелектуальної інформаційної системи для підготовки та відбору web-програмістів в ІТ-компанії

Головною методичною передумовою проєктування та реалізації інтелектуальної інформаційної системи для інтенсифікації підготовки та відбору web-програмістів є уявлення про навчання та відбір, як про процес розв'язку задачі управління навчанням в людинно-машинній системі, що складається з підсистем управління “викладач (тьютор) - автоматизований навчальний курс” та об'єкта управління - студентів спеціальності 121/F2 “Інженерія програмного забезпечення”, або інших користувачів, що бажають пройти відбір на посаду web-програміста в ІТ-компанію.

Методологія побудови і функціонування інтелектуальної інформаційної системи для інтенсифікації підготовки та відбору web-програмістів має ієрархічну структуру побудови, де основним елементом є автоматизований навчальний курс “Web-програмування”, розроблений автором. Курс має п'ять рівнів підготовки та відбору web-програмістів. Завантаживши цей курс, користувач потрапляє в спеціальне середовище вибраного рівня підготовки. В інтелектуальній інформаційній системі розкривається освітнє середовище з переліком тем, питань, які необхідно засвоїти користувачу, претенденту на посаду відповідного рівня. Освітнє середовище надає користувачу цілу групу автоматизованих послуг:

- інформаційно-довідкове обслуговування;
- розв'язок задач з курсу;
- контроль знань користувача зі сторони системи.

Надбудовою над автоматизованим навчальним курсом є система прийняття рішень, яка контролює та управляє процесом навчання та контролем знань в усіх рівнях при роботі з інтелектуальною інформаційною системою (рис. 2.6.).

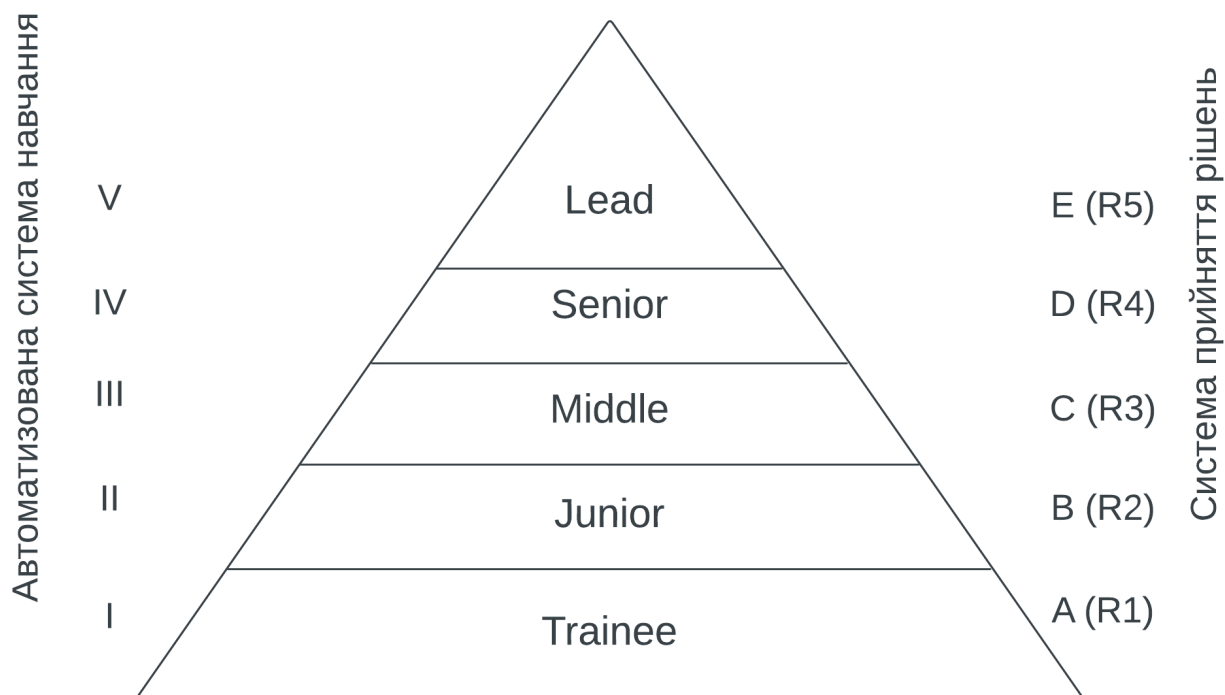


Рис. 2.6. Багаторівнева модель інтелектуальної інформаційної системи.

В основі архітектури автоматизованого навчального курсу ІС покладена п'ятирівнева модель процесу навчання.

Перший рівень містить навчальний контент К1, що містить базові поняття з web-програмування, призначений для web-програмістів початківців рівня Trainee.

Другий рівень містить навчальний контент К2, що містить поняття з web-програмування, алгоритмів, підходів та технік рівня Junior.

Третій рівень містить навчальний контент К3, що містить поняття з web-програмування, особливості підходів оптимізації, особливості фреймворків та бібліотек у відповідності до рівня Middle.

Четвертий рівень містить навчальний контент К4, що містить поняття з web-програмування, основ проєктування, методів оптимізації та інтеграції web-додатків рівня Senior.

П'ятий рівень містить навчальний контент К5, що містить поняття з web-розробки рівня Lead.

На основі біжучого стану і прийнятої методики навчання генерується чергове завдання, що вимагає відповідних дій користувача; відповідь користувача

порівнюється з варіантами правильної відповіді і на основі відмінностей проводиться діагностика відповідних дій користувача; за результатами контролю знань та результатів діагностики система управління прийняттям рішень коректує біжучі характеристики користувача та надає пропозиції щодо його подальшої підготовки та відбору на посаду web-програміста. Кожен рівень має свій набір характеристик, вимог і критеріїв відбору.

Перший рівень (підсистема прийняття рішень А) містить критерії відбору R1, призначені для web-програмістів рівня Trainee.

Другий рівень (підсистема прийняття рішень В) містить критерії відбору R2, призначені для web-програмістів рівня Junior.

Третій рівень (підсистема прийняття рішень С) містить критерії відбору R3, призначені для web-програмістів рівня Middle.

Четвертий рівень (підсистема прийняття рішень D) містить критерії відбору R4, призначені для web-програмістів рівня Senior.

П'ятий рівень (підсистема прийняття рішень Е) містить критерії відбору R5, призначені для web-програмістів рівня Lead.

Головною методичною передумовою для проектування та реалізації інтелектуальної інформаційної системи для відбору та управління кадрами ІТ-компанії є уявлення про відбір кадрів як процес розв'язання задачі управління особовим складом в автоматизованій системі прийняття рішень з технологіями штучного інтелекту.

Методологія побудови і функціонування інтелектуальної інформаційної системи для відбору web-програмістів має багаторівневу ієрархічну структуру побудови (рис. 2.7), де основним елементом першого рівня є автоматизований навчальний курс “Web-програмування”. Основним елементом другого рівня є автоматизоване тестове середовище. Основним елементом третього рівня є автоматизована система підтримки прийняття рішень (АСППР) для формування проєктних команд.



Рис. 2.7. Ієрархічна структура побудови інтелектуальної інформаційної системи.

Навчальне середовище надає користувачу групу послуг:

- інформаційно-довідкове обслуговування;
- розв'язок практичних задач з курсу web-програмування;
- автоматизований контроль знань.

Для побудови інтелектуальної інформаційної системи з підготовки та відбору web-програмістів в ІТ-компанії, що включає підсистеми для управління рекрутингом, навчанням персоналу, тестуванням та управлінням проєктними командами, було використано проєктний підхід, який охоплює як новітні ІТ-технології, так і сучасні методології управління персоналом.

Виділимо основні технології, які можна використовувати для розробки інтелектуальної інформаційної системи для підбору особового складу ІТ-компанії, а саме web-програмістів.

В результаті проведеного системного аналізу існуючих процесів управління особовим складом ІТ-компаній, ідентифіковано ключові вимоги та визначено межі автоматизованої системи, проведено формалізацію бізнес-процесів. В результаті

дослідження запропоновано високорівневу структуру (рис. 2.8) яку було розроблено у відповідності до нотації С4 [56].



Рис. 2.8. Структура ІС з підбору web-програмістів в ІТ компанії.

Застосування алгоритмів машинного навчання та штучного інтелекту для автоматизації та оптимізації процесів відбору персоналу, аналізу ефективності роботи співробітників, а також для прогнозування потреб у навчанні та розвитку персоналу дозволило інтегрувати технології штучного інтелекту до інформаційної системи. Інтеграцію було проведено з використанням технології LangGraph, яка дозволяє використовувати OpenAI GPT-4 [57] як технологічну основу для реалізації прикладних процесів аналізу даних та підтримки прийняття рішень.

Інтеграція з зовнішніми системами, такими як модулі ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) та інші ІТ-ресурси компанії для забезпечення цілісності даних та ефективного управління ресурсами. На основі раніше визначених меж систему було здійснено крос-системну інтеграцію (рис. 2.9).

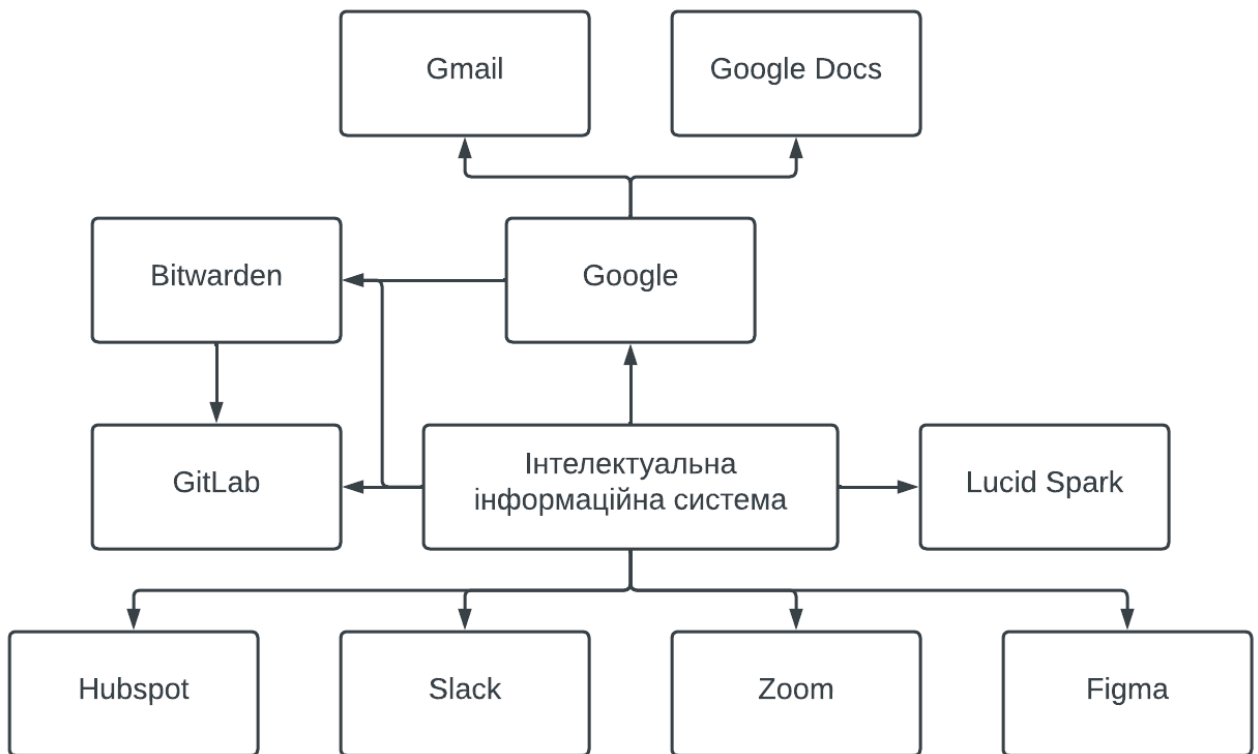


Рис. 2.9. Схема інтеграції з зовнішніми системами.

В результаті дослідження інтелектуальну інформаційну систему було інтегровано з платформами:

- Hubspot - хмарною CRM-платформою для управління продажами та комунікації з клієнтами;
- Slack - хмарною платформою для онлайн-комунікації серед менеджменту та проєктних команд;
- Zoom - хмарною платформою для організації телеконференцій;
- Lucid Spark - хмарною платформою для прототипування;
- Figma - хмарною платформою для візуального прототипування;
- Bitwarden - хмарною платформою для управління захищеними сховищами даних;
- GitLab - хмарною платформою для управління вихідними кодами проєктів та управління доставкою коду;
- Google Mail - хмарною платформою для управління електронною поштою;

- Google Documents - хмарною платформою для управління документами.

Розробка надійної та масштабованої архітектури баз даних, яка забезпечить високий рівень безпеки, доступності та швидкодії обробки даних. Для побудови інтелектуальної інформаційної системи було використано комплексний підхід до зберігання даних (рис. 2.10).

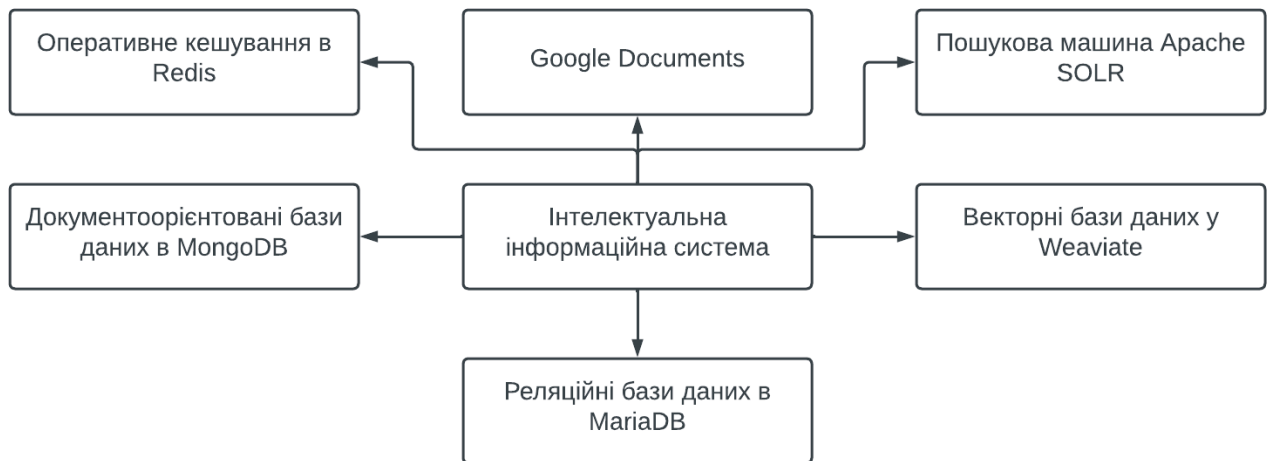


Рис. 2.10. Структура управління даними в ІС.

В якості основного засобу зберігання даних використовуються реляційні бази даних, які працюють під управлінням системи MariaDB. Для системи штучного інтелекту використовується векторна база даних на основі технології Weaviate. Для проміжного зберігання даних використовується документо-орієнтована база даних на основі технології MongoDB. Для оперативного кешування даних використовується сховище даних на основі технології Redis. Для побудови корпоративної бази знань використовується пошукова машина на основі технології Apache SOLR. Для командної взаємодії при роботі з документами використовується екосистема Google Documents.

Розробка інтуїтивно зрозумілих та зручних користувацьких інтерфейсів, забезпечення гарного UX (User Experience) та відповідності UI (User Interface) сучасним стандартам відіграє ключову роль в побудові сучасної інтелектуальної інформаційної системи. Для побудови користувацького інтерфейсу було

використано сучасний фреймворк Chakra UI, який є одним з найбільш ефективних та широко використовуваних рішень. Вибір даного фреймворку пояснюється можливістю покриття ключових вимог до сучасної інтелектуальної інформаційної системи:

- простота використання: Chakra UI надає прості та інтуїтивно зрозумілі API для компонентів, що робить процес розробки швидшим та ефективнішим - це допомагає розробникам зосередитися на бізнес-логіці, а не на деталях імплементації інтерфейсу;
- модульність та гнучкість: компоненти Chakra UI розроблені з урахуванням модульності, що дозволяє легко їх налаштовувати та розширювати відповідно до потреб проекту;
- доступність: бібліотека забезпечує високий рівень доступності, зокрема, клавіатурний доступ до компонентів та підтримку ARIA атрибутів;
- тематизація та стилізація: Chakra UI включає систему тематизації, яка дозволяє легко кастомізувати зовнішній вигляд компонентів, також підтримується стилізація за допомогою Styled System та Emotion;
- респонсивний дизайн: Бібліотека підтримує респонсивний дизайн, дозволяючи компонентам автоматично адаптуватися до різних розмірів екрану.

2.4. Багаторівнева перевірка знань з підбору web-програмістів в IT-компанії

На основі зробленого аналізу застосування інтелектуальних інформаційних систем при підготовці web-програмістів, приведеного у висновках розділу 1, виходячи із змісту підготовки web-програмістів, виникає потреба у багаторівневому поданні знань з інформаційних технологій та перевірці засвоєння таких знань. Розглянемо детально рівні подання інтелектуальної інтегрованої технології навчання (рис. 2.11).

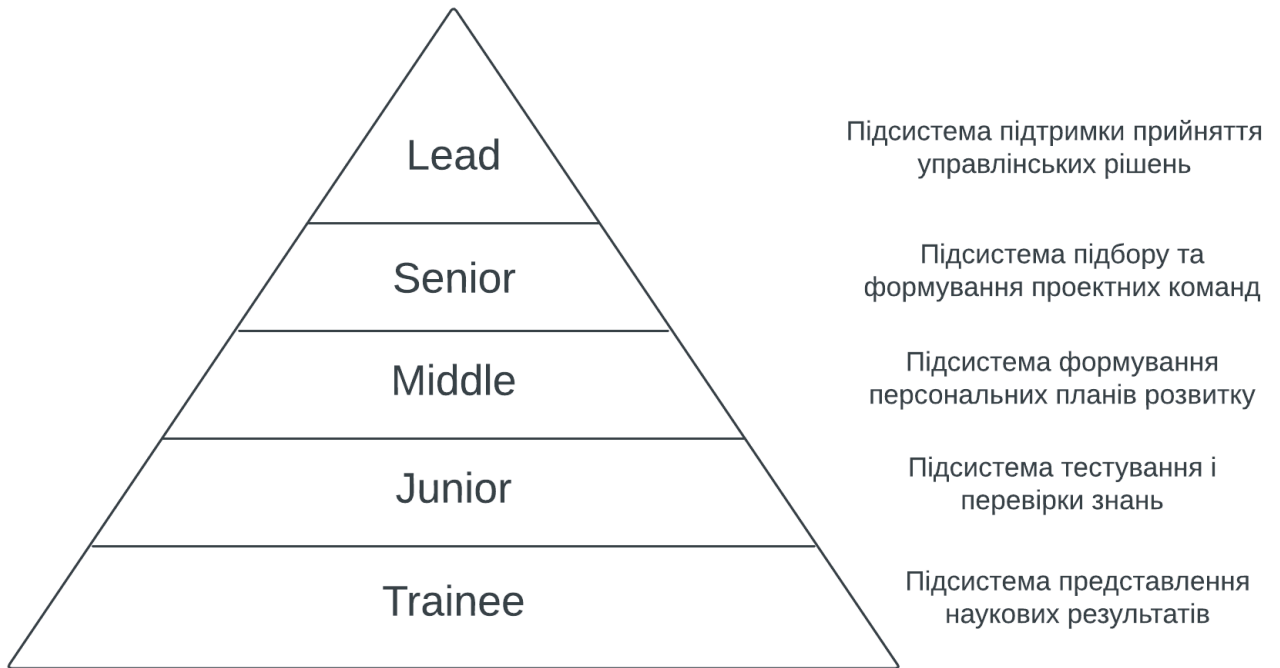


Рис. 2.11. Рівні подання інтелектуальної інтегрованої технології навчання.

Предметна область з web-розробки у загальному вигляді характеризується як трійка:

$$M = (\{O\}, \{V\}, \{W\}), \text{ де}$$

$\{O\}$ - множина моделей об'єктів;

$\{V\}$ - множина відношень між об'єктами;

$\{W\}$ - множина операцій над об'єктами і відношеннями.

Розглянемо варіанти множин моделей об'єктів відповідно ієрархічного рівня створюваної інтелектуальної інформаційної системи (рис. 2.10).

1. $M=\{O\}$ - найпростіший варіант, де $\{O\}$ - множина навчального контенту для рівня Trainee, який являє собою набір гіпертексту та гіпермедіа, структурований у відповідності до змісту курсу підготовки web-програмістів для рівня Trainee;

2. $M=(\{O_i\}, \{V\})$ - де $\{O_i\}$ - множина фрагментів навчального контенту для рівня Junior; $\{V\}$ - множина зв'язків між цими фрагментами. Дані такої структури потребують застосування спеціальних інструментальних засобів для

виділення фрагментів навчального контенту та встановлення зв'язків між цими фрагментами;

3. $M = (\{o, v\}, V)$, де o - семантично цілісні елементи деяких знань (поняття, зображення, визначення, питання, відповіді, підказки, тощо для рівня Middle); v - відношення, що пов'язують їх у семантичні структури (списки, фрейми, дерева, мережі, тощо); V - відношення між структурами даних. На основі таких даних та знань такої структури, створюються автоматизовані навчальні системи та курси, інформаційні системи, бази даних та знань;

4. $M = (\{o\}, \{v\}, \{w\}, \{V\}, \{W\})$, де $\{o, v, w\}$ - динамічні об'єкти предметної області для рівня Senior; $\{V\}, \{W\}$ - відношення і операції над ними. На основі таких структур створюються інтелектуальні ігри і тренажери, бази знань, автоматизовані інформаційні навчальні системи.

Така модель багаторівневого подання знань предметної області дозволила створити функціонуючий макет інтелектуальної інформаційної системи. При цьому, при переході від рівня до рівня зберігається спадкоємність не лише інформації та даних, але й їх структур, створених на попередньому рівні.

До проєктування інтелектуальної інформаційної системи для відбору web-програмістів будемо використовувати основи "задачного підходу".

Однією з цілей задачного підходу є побудова структури різних штучних систем, в тій чи іншій мірі здатних до вирішення задач та діалогу; іншою ціллю служить визначення змісту навчання та тестування користувача, який вступає у взаємодію з інтелектуальною інформаційною системою [47].

Нашою основною методологічною ідеєю є розгляд взаємодії користувача та інтелектуальної інформаційної системи як процесу прийняття рішень в різних задачах для web-розробників.

В плані нашої методології розробка інтелектуальної інформаційної системи переслідує такі основні цілі:

- перевірка працездатності запропонованої структури інтелектуальної інформаційної системи і вироблення рекомендацій по створенню таких систем;

- визначення ефективності окремих навчальних модулів інтелектуальної інформаційної систем та системи в цілому;
- визначення ефективності окремих тестових модулів інтелектуальної інформаційної системи;
- дослідна реалізація основних режимів роботи взаємодії навчальних процедур;
- дослідна реалізація основних режимів роботи взаємодії тестових процедур;
- дослідна реалізація основних режимів роботи взаємодії одночасно навчальних та тестових процедур;
- створення конкретної інтелектуальної інформаційної системи, підготовленої до широкого впровадження в навчальний процес підготовки web-розробників та web-програмістів.

Вирішення поставленої задачі будемо розглядати як деякий інтелектуальний творчий процес.

Завдяки наявності моделі предметної області, інтелектуальна інформаційна система контролює відповіді на запитання навчаючих як за черговою, пред'явленою йому порцією навчального контенту, запитанням, завданням, так і по всьому матеріалу, представленому в цій моделі (аналізатор запитань).

Для створення інтелектуальної інформаційної системи з відбору web-програмістів необхідно:

1. Розробити алгоритми різних рівнів взаємодії користувачів та інтелектуальної інформаційної системи.
2. Виробити вимоги й розробити програмне забезпечення (інструментарій) для реалізації цих алгоритмів з відбору web-програмістів, включаючи засоби формування багаторівневої моделі предметної області.
3. Розподілити функції з реалізації процедури навчання, процедури тестування, та процедури прийняття рішень інтелектуальною інформаційною системою.

4. Встановити співвідношення між фіксованими і формованими частинами навчального контенту, вибрати та опрацювати інструментальні засоби програмування ключових блоків інтелектуальної інформаційної системи.

В основу розробки формалізованого опису інтелектуальної інформаційної системи покладений “задачний” підхід, описаний у науковому виданні Довгялло О. М. “Діалогові системи: сучасний стан та перспективи розвитку” [74]. Цей підхід дозволяє представити процес взаємодії навчаючого та системи як процес вирішення задач певного виду.

За такого підходу, перевірка знань користувача інтелектуальної інформаційної системи, здійснювана самою системою бачиться як вирішення задачі контролю знань, формування роз’яснення помилки - як вирішення задачі діагнозу причини помилки, і задачі адаптації до індивідуальних особливостей навчаючого, тощо.

Кожна задача, що вирішується такою інтелектуальною інформаційною системою визначається як трійка:

$$Z = (K^*, K_{akt}, K_{*potr}), \text{ де}$$

K^* - предметна область задач з web-програмування - в свою чергу є трійкою:

$$K = (\{A\}, \{W\}, \{R\}), \text{ де}$$

$\{A\}$ - множина об’єктів;

$\{W\}$ - множина операцій над $\{A\}$;

$\{R\}$ - множина відношень (предикатів) на $\{A\}$.

K_{akt} - актуальний, тобто наявний в даний момент стан предметної області задачі;

K_{*potr} - модель потрібного стану предметної області з web-програмування.

Модель K_{*potr} розуміється як деякий опис стану предметної області в залежності від пореб навчаючого та тестованого.

Перетворення актуального стану K_{akt} у потрібний K_{*potr} називається процесом чи процедурою вирішення задачі Z . Потрібний стан є результатом вирішення.

Результатом роботи інтелектуальної інформаційної системи є отримання необхідного стану предметної області з відбору web-програмістів.

Процес грейдування персоналу, тобто безперервний процес оцінювання є системою, за допомогою якої ІТ-компанії класифікують різні робочі позиції відповідно до їх значимості, відповідальності та складності (рис. 2.12). Скіл матриця виступає інструментом для оцінки навичок, знань та досвіду web-програмістів на різних позиціях.

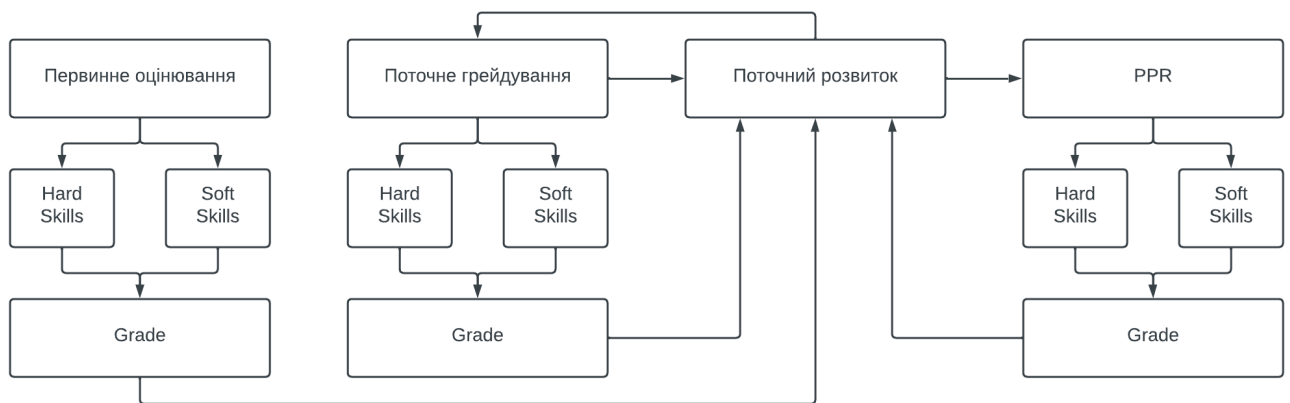


Рис. 2.12. Схема багаторівневої перевірки знань в ІС.

Для оцінювання автором розроблена матриця компетенцій, відповідно до заданих технічних доменів (Додаток Б).

В таблиці скіл матриці дана диференціація компетенцій реалізована виділенням компетенцій кольором:

- required (обов'язковий) - червоний;
- expected (очікуваний) - жовтий;
- extra (додатковий) - зелений.

Опишемо математичну модель оцінки компетенцій, на основі якої функціонує інтелектуальна інформаційна система для оцінювання персоналу ІТ-компаній

Множина компетенцій може бути задана як трійка:

$$A = (\{I\}, \{J\}, \{K\}), \text{ де}$$

$\{I\}$ - множина можливих компетенцій;

$\{J\}$ - множина кваліфікаційних рівнів;

$\{K\}$ - множина технічних доменів.

Враховуючи описану множину компетенцій A , скіл матриця може бути представлена як тривимірна матриця $[A^*]$ з розмірностями i, j, k , де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів.

Тоді окремо компетенцію можна представити як елемент матриці $a = A^*[i,j,k]$.

В процесі оцінювання окремо визначеного web-програміста в ІТ компанії індекс j буде вважатись константою, оскільки оцінювання проводиться в рамках окремого фіксованого кваліфікаційного рівня.

Кожна компетенція може бути оцінена у відповідності до наступних квантованих рівнів, а отже оцінка компетенції може набувати відповідних фіксованих значень (табл. 2.5).

При проєктуванні математичної моделі інтелектуальної інформаційної системи для оцінювання web-програмістів було задано мінімальний рівень володіння компетенцією в розмірі 50%, що відповідає коефіцієнту 0.50. Таким чином, володіння матеріалом на рівні 50% буде вважатись мінімально достатнім для присвоєння грейду і тайтлу web-програмісту.

Таблиця 2.5. Рівні володіння компетенціями.

Рівень	Коефіцієнт	Статус
0%	0	Не зараховано
25%	0.25	Не зараховано
50%	0.50	Зараховано
75%	0.75	Зараховано
100%	1	Зараховано

Квантування рівнів володіння компетенціями було реалізовано на основі модифікованої шкали ECTS в становленні мінімального прохідного балу що відповідає успішно освоєному 50% обсягу навчального матеріалу.

Володіння вищими рівнями компетенцій визначає переваги окремо взятого web-програміста в рамках визначеного грейду, і є факторами конкурентного відбору при формуванні проєктних команд.

Для забезпечення гнучкості оцінювання, автором було запропоновано ввести додаткові умови необхідності володіння окремо взятою компетенцією. Так, у відповідності до оголошених рівнів володіння компетенціями, було введено наступні коефіцієнти-мультиплікатори:

Таблиця 2.6. Значення коефіцієнтів-мультиплікаторів для рівнів володіння компетенціями.

Ознака необхідності володіння компетенцією	Коефіцієнт-мультиплікатор
Required	1
Expected	1.1
Extra	1.3

На основі механізму рівнів володіння компетенціями та механізму ознак необхідності володіння компетенцією, автором було створено наступний алгоритм оцінювання окремо взятої компетенції, в якому логіка прийняття рішення базується на рівнях володіння компетенціями, заданих в таблиці 2.5 (рис. 2.13).

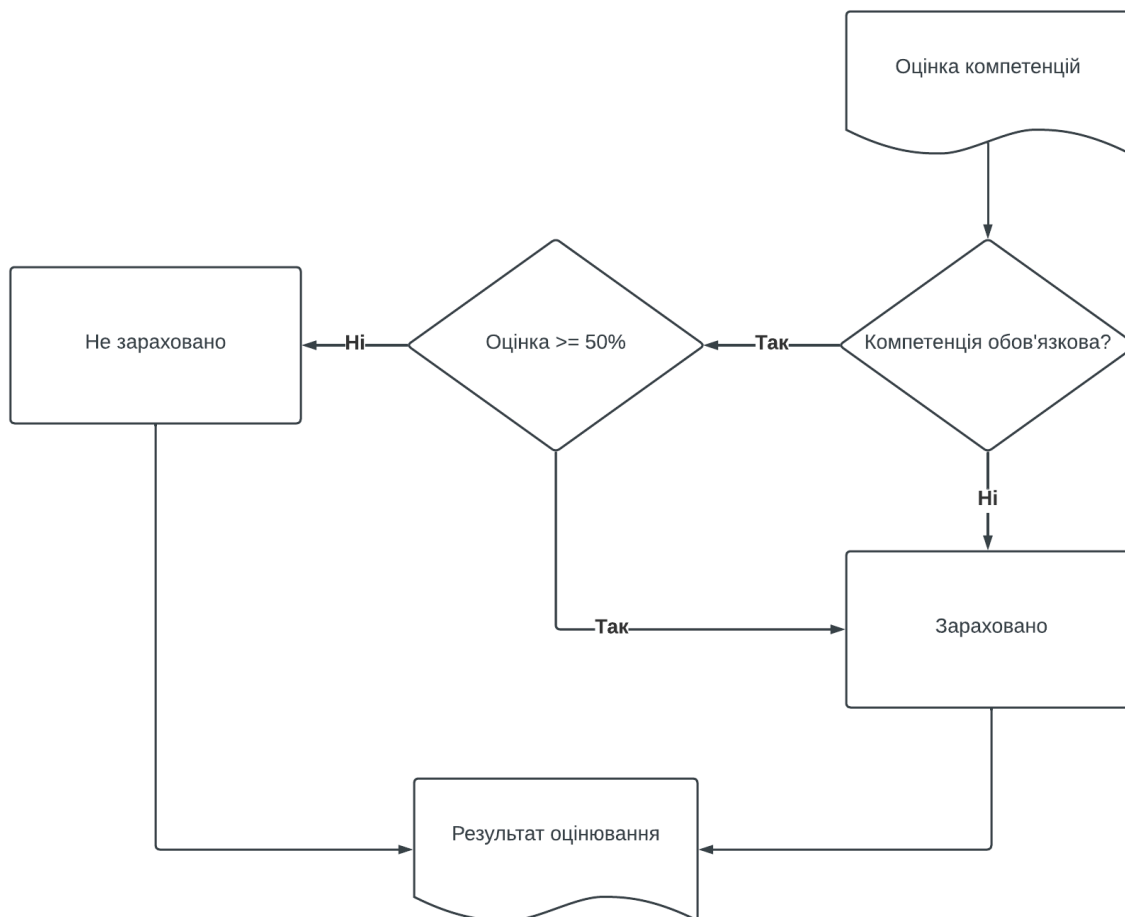


Рис. 2.13. Алгоритм оцінювання компетенції.

Технічна реалізація даного алгоритму в інтелектуальній інформаційній системі реалізована на основі механізму калькуляції добутку коефіцієнта володіння компетенцією на коефіцієнт-мультиплікатор і порівняння його з відповідним значенням еталонної матриці [E] що має розмірності i, j, k у відповідності до матриці компетенцій [A*].

Еталонна матриця [E] сформована з врахуванням мінімально допустимого рівня володіння компетенціями, тобто кожен елемент матриці $e[i, j, k]$ є константою, і завжди має значення 0.5.

Таким чином, допустимий інтервал для рівня володіння окремою компетенцією можна представити як сукупність умов (1) та (1.1):

$$1 > a[i, j, k] \geq 0.5 \quad (1)$$

$$e[i, j, k] = 0.5 \quad (1.1)$$

де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів, $a[i, j, k]$ - рівень володіння компетенцією, $e[i,j,k]$ - еталонний рівень володіння компетенцією.

Для визначеного технічного домену k можемо отримати загальну оцінку володіння технічним доменом, яку можна представити як суму оцінок компетенцій в рамках технічного домену (2):

$$\sum_{i=1}^n a[i, j, k] \quad (2)$$

де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів, $a[i, j, k]$ - рівень володіння компетенцією.

Еталонну оцінку володіння технічним доменом k можна представити як суму еталонних оцінок компетенції:

$$\sum_{i=1}^n e[i, j, k] \quad (2.1)$$

де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів, $e[i, j, k]$ - рівень володіння компетенцією

Достатній рівень володіння технічним доменом в цьому разі можна представити як умову:

$$\sum_{i=1}^n a[i, j, k] \geq \sum_{i=1}^n e[i, j, k] \quad (3)$$

де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів, $a[i, j, k]$ - рівень володіння компетенцією, $e[i,j,k]$ - еталонний рівень володіння компетенцією.

Актуальний рівень технічної експертизи (4) та мінімально можливий рівень технічної експертизи (4.1) в рамках комплексного оцінювання рівня знань web-програміста можна представити наступним чином:

$$\sum_{k=1}^m \sum_{i=1}^n a[i, j, k] \quad (4)$$

$$\sum_{k=1}^m \sum_{i=1}^n a[i, j, k] \geq \sum_{k=1}^m \sum_{i=1}^n e[i, j, k] \quad (4.1)$$

де i - кількість компетенцій, j - кількість кваліфікаційних рівнів, k - кількість технічних доменів, $a[i, j, k]$ - рівень володіння компетенцією, $e[i, j, k]$ - еталонний рівень володіння компетенцією.

Зазначимо, що умова 4.1 є достовірною за дотримання умов 1 та 2.1.

Для вибору компетенцій було використано вдосконалену матрицю компетенцій з диференційованим розподілом скілів та матрицею вагових коефіцієнтів (додаток В).

В якості базового інструментарію було розроблено шаблони для проведення оцінювання, які використовують технологію Google Spreadsheets в якості засобу для збору та первинного оперування даними. Вибір даної технології обгрунтовується відкритою ліцензією, відсутністю обмежень до використання, хорошими можливостями до системної інтеграції завдяки можливостям використання Google API.

Проведення оцінки відбувається в ході трьох етапів:

- самооцінювання - під час якого фахівець, який проходить оцінювання самостійно заповнює шаблон, даючи відповіді на питання та вибираючи найбільш відповідні варіанти відповідей зі списку;
- оцінювання - під час якого відбувається оцінка фахівцем, який заповнює шаблон оцінювання на основі проведення технічного інтерв'ю та вибираючи найбільш відповідні варіанти відповідей зі списку;
- проведення порівняння результатів самооцінювання та оцінювання з мінімально допустимими результатами та формування диференційованої оцінки.

Результатом технічного оцінювання є формальний документ - план персонального розвитку (Personal Development Plan, PDP), який регулює подальший розвиток фахівця в рамках компанії, визначає здобуття фахівцем того чи іншого професійного рівня в компанії або визначає конкретні кроки, які необхідно виконати ІТ фахівцю для його отримання.

Personal Development Plan (PDP) - це структурований план, що створюється окремою особою для визначення та відстеження розвитку своїх навичок, знань та компетенцій. PDP використовується для підвищення особистої ефективності, досягнення професійних цілей та покращення якості життя.

Основні елементи Personal Development Plan:

1. Аналіз поточного стану - включає в себе інтроспективну оцінку своїх сильних сторін, слабких місць, можливостей та загроз.
2. Визначення цілей - конкретні, вимірювані, досяжні, релевантні та обмежені в часі (SMART-критерії) цілі, які особа прагне досягти.
3. Стратегії та дії - розробка конкретних кроків, стратегій або технік, які допоможуть досягти визначених цілей.
4. Моніторинг та оцінка - регулярний огляд та аналіз своїх досягнень, вивчення зібраної обратної інформації та коригування плану відповідно до змін.
5. Зворотний зв'язок - отримання відгуків від наставників, колег, керівників або консультантів з метою покращення.

В рамках існуючої моделі було розроблено шаблони оцінювання для наступних рівнів:

- Trainee Drupal Backend Developer;
- Junior Drupal Backend Developer;
- Middle Drupal Backend Developer;
- Senior Drupal Backend Developer;
- Drupal Tech Lead.

2.5. Проєкт інтелектуальної інформаційної системи, необхідної для підготовки та відбору web-програмістів

На основі проведеного аналізу, який був зроблений вище в п.п. 2.1, 2.2, 2.3, 2.4 можна скласти таблицю інформаційних технологій, які необхідно використовувати для блоку навчального курсу з web-програмування та блоку-підсистеми для перевірки знань ієрархічних рівнів web-програмування.

Таблиця 2.7. Технології для побудови інтелектуальної інформаційної системи.

Технологія	Реалізація	Призначення
Бази даних	SQL	Зберігання, відбір та первинна обробка даних
WEB	HTTPS, Nginx, PHP	Взаємодія з користувачами, передача даних, практична реалізація бізнес-процесів
Авторизація та автентифікація	OAuth	Безпечний доступ до даних
Штучний інтелект	OpenAI GTP-4, LangGraph	Автоматизована інтелектуальна обробка даних

У розробці інтелектуальної інформаційної системи для підготовки та відбору web-програмістів кожен описаний вище компонент відіграє ключову роль, сприяючи загальній ефективності та інноваційності системи. Відмітимо роль кожного з цих компонентів:

- база даних є фундаментальним компонентом інтелектуальної інформаційної системи, вона забезпечує структуроване зберігання, управління та отримання даних, що є критично важливим для аналізу, обробки та виведення інформації; ефективне використання баз даних дозволяє системі швидко обробляти великі обсяги даних, забезпечуючи точність та надійність інформації;

- веб-технології відіграють ключову роль у розробці інтелектуальних інформаційних систем, оскільки вони забезпечують універсальний доступ до системи через інтернет - це дозволяє користувачам взаємодіяти з системою з будь-якого місця та на будь-якому пристрої, що значно розширює її функціональність та доступність; WEB також сприяє інтеграції з іншими онлайн-сервісами та даними, що розширює можливості системи.

- авторизація та автентифікація критично важливою для забезпечення безпеки інтелектуальної інформаційної системи; автентифікація забезпечує перевірку особи користувача, тоді як авторизація контролює доступ до ресурсів

системи - вони відіграють ключову роль у захисті конфіденційності даних та запобіганні несанкціонованому доступу;

- штучний інтелект є ключовим драйвером інновацій у інтелектуальних інформаційних системах - і використання штучного інтелекту дозволяє системі вчитися з даних, автоматизувати складні завдання, а також забезпечувати розширені аналітичні та прогнозні можливості - це сприяє підвищенню ефективності системи, покращенню взаємодії з користувачем та здатності системи адаптуватися до змінних умов та потреб користувачів.

У сукупності, ці компоненти створюють міцний фундамент для інтелектуальної інформаційної системи, забезпечуючи її надійність, ефективність та інноваційність.

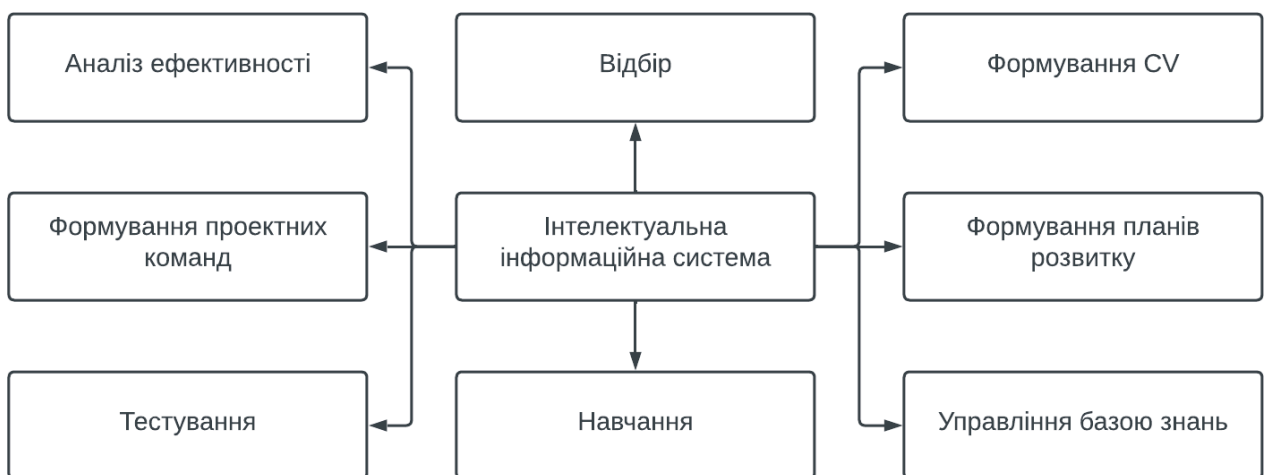


Рис. 2.14. Структура підсистем інтелектуальної інформаційної системи для підготовки та відбору web-програмістів.

Інтелектуальну інформаційну систему для підготовки та відбору web-програмістів можна представити як сумупність окремих підсистем, які виконують окремі функції а також глибоко інтегровані між собою (рис. 2.14).

У ІТ компаніях особливий акцент зазвичай робиться на процесах підбору та розвитку персоналу, оскільки технологічна галузь швидко розвивається, а потреба в висококваліфікованих спеціалістах росте. Тому здатність адаптуватися до

нововведень, навчати своїх співробітників та стимулювати їх до постійного професійного росту є ключовою для успіху ІТ організацій.

Найбільш цікавими в рамках даного дослідження є бізнес-процеси підбору та найму персоналу, а також поточного оцінювання та розвитку.

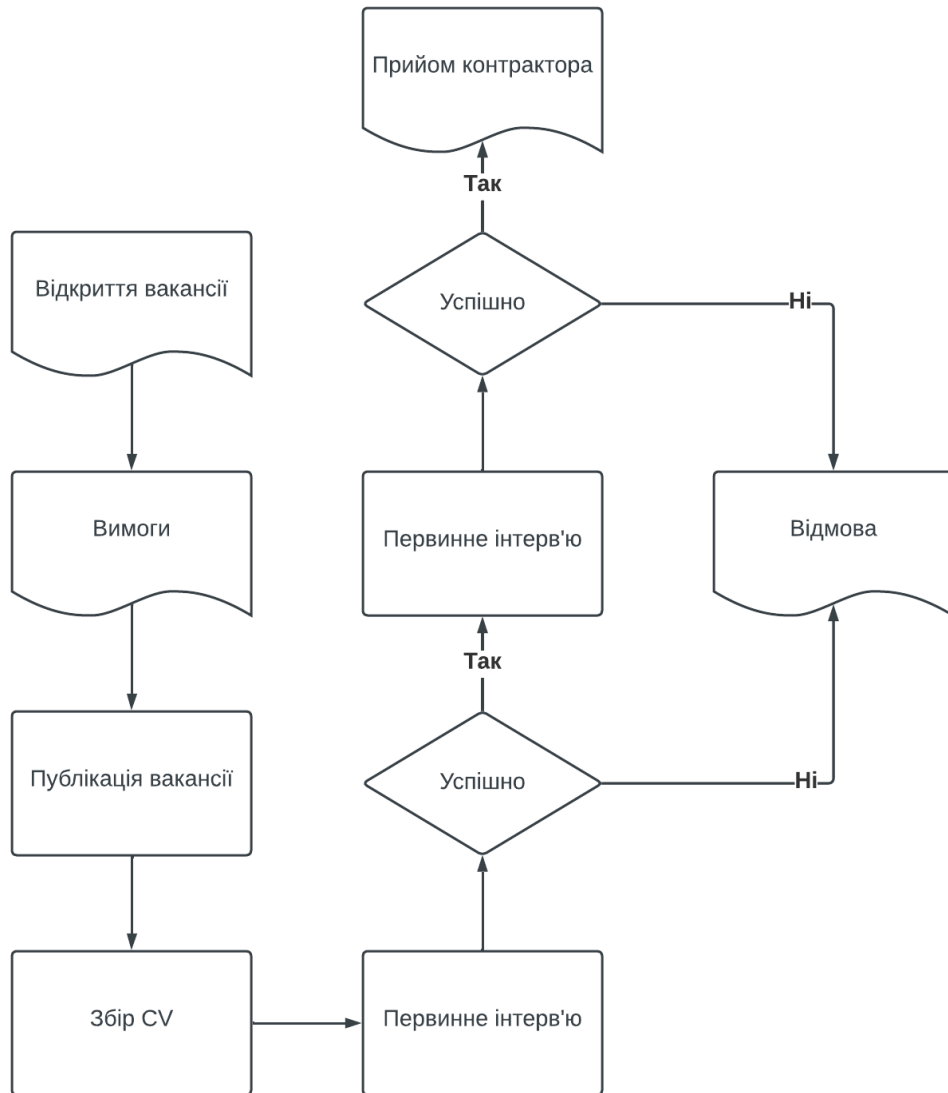


Рис. 2.15. Бізнес-процес найму персоналу в ІТ-компанії.

Розглянемо схему процесу найму персоналу в ІТ компанії (рис. 2.15). Як бачимо з наведеної схеми бізнес-процесу, найм персоналу є ступінчастим процесом, який регламентується первинним документом, що має назву “Запит на відкриття позиції”, який ініціюється та формується керівником відділу, в інтересах якого буде відбуватись процес рекрутингу та найму. Системним артефактом, який

визначається даним документом є вимоги до фахівця, на основі яких відбувається процес інтерв'ю та безпосередньо технічне опитування і тестування.

Процес найму включає в себе два етапи інтерв'ю:

- первинне інтерв'ю;
- технічне інтерв'ю.

На етапі первинного інтерв'ю відбувається первинне знайомство претендента з рекрутером, здійснюється первинна оцінка відповідностей досвіту та оголошених претендентом технічних умінь та навичок.

На етапі технічного інтерв'ю відбувається повноцінна технічна співбесіда, яка проводиться у відповідності до очікуваного рівня, з використанням матриці компетенцій та відповідного шаблону оцінювання. Результатом технічного інтерв'ю є технічний відгук на кандидата. В разі, якщо кандидат задовільняє всім необхідним вимогам, та є оптимальним за вартістю - відбувається фінальне інтерв'ю з керівником структурного підрозділу.

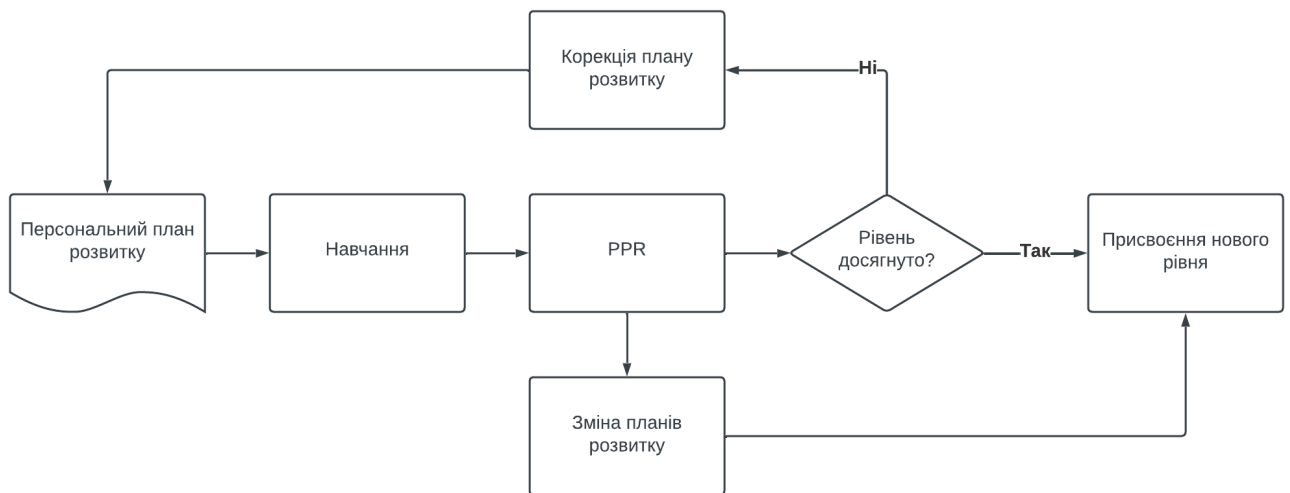


Рис. 2.16. Бізнес-процес професійного розвитку web-програміста.

В разі успішного фінального інтерв'ю відбувається ініціація процесу найму фахівця та інтеграція його у відповідний структурний підрозділ та відповідну проектну команду.

Іншим важливим бізнес процесом є безперервне навчання та та підготовка персоналу. Як ми бачимо з рис. 2.16, процес професійного розвитку web-програміста є безперервним, і поєднує в собі навчання згідно персонального плану розвитку, періодичні перевірки знань, умінь та навичок згідно очікуваного професійного рівня (Personal Performance Review), корекцію персонального плану розвитку у разі підвищення рівня або внесення поправок для швидшого прогресу в професійному розвитку за поточним рівнем. Окремо слід відзначити випадок зміни професійного напрямку. В цьому разі зазвичай відбувається пониження професійного рівня з переходом за новим напрямком, що в свою чергу викликає корекцію персонального плану розвитку.

Розглянемо модель даних та взаємозв'язків між даними в інтелектуальній інформаційній системі для підготовки та відбору web-програмістів (рис. 2.17):



Рис. 2.17. Модель даних інтелектуальної інформаційної системи для підготовки та відбору web-програмістів.

Як бачимо з рис. 2.17, модель даних інтелектуальної інформаційної системи для підготовки та відбору web-програмістів включає в себе набір структур даних,

які відповідають за зберігання та обробку відповідних елементів даних бізнес процесів, і технічно в інформаційній системі представлені у вигляді таблиць в базі даних. Детальні описи структури таблиць подано в додатку 4.

Висновки до другого розділу

В результаті проведеного аналізу сучасних методів та засобів розробки інтелектуальних інформаційних систем, змісту навчання web-програмістів, загальних принципів підготовки програмістів за спеціальністю 121/F2. Інженерія програмного забезпечення, загальних принципів командної розробки IT-проектів, методологічних та наукових аспектів знань з програмування, багаторівневого подання знань з підготовки та відбору web-програмістів були вирішені наступні задачі:

1. Розроблена структура та принципи побудови програмного забезпечення для інтенсифікації підготовки web-програмістів.
2. Розроблені моделі і методи відбору web-програмістів в IT-компанії.
3. Складена таблиця професійних рівнів web-програмістів;
4. Запропонована і впроваджена методики побудови інтелектуальної інформаційної системи для відбору web-програмістів в IT-компанії.

Сформовано новий підхід до оцінювання знань, умінь та навичок web-програмістів IT компаній, з акцентом на бекенд-розробників, що використовують фреймворк CMF Drupal.

Сфоровані вимоги до знань, умінь та навичок web-програмістів дали змогу визначити критерії професійної діяльності web-програмістів, враховуючи специфіку роботи з CMF Drupal. Встановлені вимоги стали фундаментом для подальшого моделювання процесу оцінювання.

Завдяки побудові та верифікації скіл-матриці автором було створено набір компетенцій, який є необхідним для web-розробників, та оцінено актуальність та релевантність кожної з них.

Побудована та верифікована автором матриця вагових коефіцієнтів дозволила встановити ієрархію навичок та компетенцій, яка важлива для багаторівневого навчання при підготовці та перепідготовці web-програмістів.

Розроблена автором математична модель оцінювання дозволила реалізувати в інтелектуальній інформаційній системі процес автоматизованого тестування знань, умінь та навичок web-програмістів.

Розроблена автором методика оцінювання технічних знань враховує особливості та потреби сучасних ІТ компаній та пропонує інтегрований підхід до вимірювання ефективності web-програмістів.

Формалізація багаторівневого навчання та контролю знань дозволила побудувати модель інтелектуальної інформаційної системи для підготовки та відбору web-програмістів, сформувати модель даних та забезпечити системний підхід до регулювання взаємодії між різними учасниками бізнес-процесів та оптимізувати їх діяльність.

Автором розроблено класифікацію знань, необхідних для сучасної підготовки web-програмістів, та розроблено модель багаторівневого комп'ютерного курсу з web-програмування.

РОЗДІЛ 3.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ БАГАТОРІВНЕВИМ НАВЧАННЯМ ТА ПІДБОРОМ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ

3.1. Електронний навчальний курс “WEB-програмування”

Електронний навчальний курс “WEB-програмування” являє собою комплекс електронних навчально-методичних матеріалів, створених для організації індивідуального та групового багаторівневого навчання з використанням дистанційних технологій, як під керівництвом викладача, так і самостійно.

Основними характеристиками електронного навчального курсу “WEB-програмування” є:

- структурованість навчальних матеріалів;
- відповідність основним структурним елементам процесу вивчення навчального курсу: лекції, практичні роботи, лабораторні роботи, самостійні роботи, перевірку знань (залік або іспит);
- чіткий графік виконання здобувачем навчального плану;
- налагоджена система інтегрованої взаємодії викладача і студента засобами електронного навчального курсу та дистанційних технологій Moodle і OpenEDX протягом усього часу вивчення навчального курсу;
- розміщені навчальні матеріали в системі Moodle і OpenEDX дозволяють отримати компетенції, заплановані в навчальній програмі;
- система оцінювання результатів навчальної діяльності здобувачів охоплює форми та критерії оцінювання всіх видів навчальної діяльності;
- система самоконтролю всіх видів навчальної діяльності дозволяє здобувачу пройти навчальний матеріал ще раз та покращити свій результат оцінювання.

Електронний навчальний курс “WEB-програмування” містить електронні ресурси двох типів:

- електронний підручник “WEB-програмування”, електронні конспекти лекцій, мультимедійні презентації лекцій, методичні рекомендації;
- завдання з кожної теми, призначені для закріплення вивченого матеріалу, для формування вмій та навичок, самооцінювання та оцінювання навчальних досягнень здобувачів, тестування, навчальний чат з менторами.

Електронний підручник “WEB-програмування” на платформі Drupal є доступним з будь-якого пристрою: ПК, ноутбука, мобільних гаджетів.

Сучасний web-програміст повинен володіти основами інформаційних технологій, серед яких знаннями мов програмування клієнтського та серверного середовища, знання баз даних, операційних систем та системного адміністрування, знання сучасних інструментів розробки, а також фундаментальні знання з технологій програмування.

Перевагою електронного підручника є:

- кросплатформенність;
- адаптивність;
- масштабованість;
- самоперевірка знань;
- інтерактивність.

Електронний підручник розміщено на web-сайті МЕНУ, внутрішніх корпоративних ресурсах ІТ-компаній - корпоративних LMS системах.

Електронний підручник працює в двох режимах:

- режимі навчання;
- режимі самоперевірки знань.

Процес навчання відбувається інформаційним методом з послідовною деталізацією поданого контенту. Вибравши окрему тему, навчаючий отримує навчальний контент.

Основою електронного підручника є класифікація знань (рис. 3.1), яких має набувати майбутній Web-програміст:

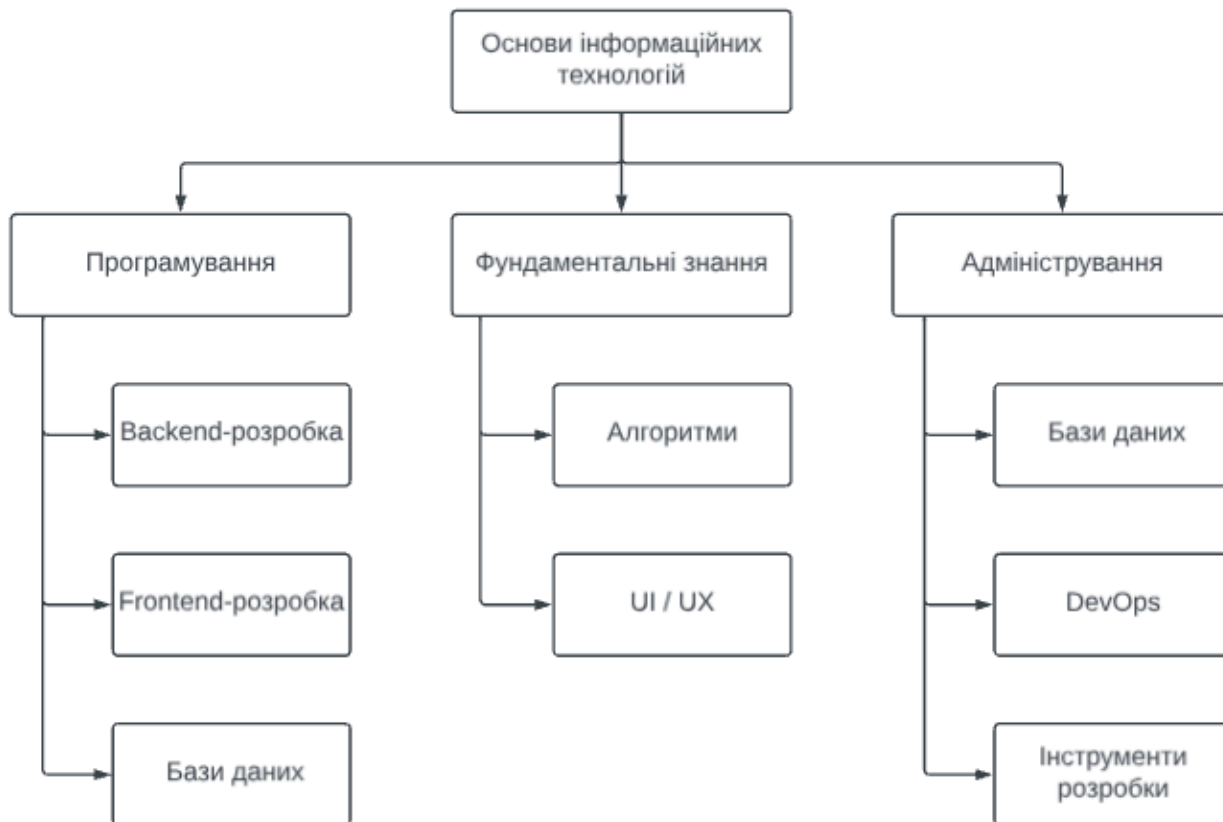


Рис. 3.1. Основи інформаційних технологій, необхідних для підготовки web-програмістів.

Автором було виділено основи інформаційних технологій для майбутнього web-програміста, яка сформована у вигляді наступних блоків та модулів:

Основи програмування:

- основні концепції програмування (змінні, цикли, умовні оператори, функції);
- об'єктно-орієнтоване програмування (класи, об'єкти, спадкування, поліморфізм);
- основи алгоритмів та структур даних.

Основи PHP:

- синтаксис та основи PHP;
- робота з формами та обробка даних користувача;
- робота з сесіями та куки.

Розробка Front-end:

- HTML/CSS, включаючи HTML5 та CSS3;
- JavaScript, включаючи DOM-маніпуляції та AJAX.

Бази даних та SQL:

- основи реляційних баз даних;
- основи SQL, включаючи CRUD операції;
- робота з MySQL або іншими СУБД, сумісними з PHP.

Розширений PHP:

- поглиблені теми PHP (робота з файлами, регулярні вирази, обробка помилок);
- розробка за допомогою фреймворків PHP (наприкладі Drupal);
- безпека веб-додатків (XSS, CSRF, SQL Injection).

Інструменти розробника:

- системи контролю версій (наприклад, Git);
- основи роботи з командним рядком та Linux;
- використання інструментів для автоматизації розробки (наприклад, Composer).

Веб-сервери та розгортання:

- конфігурація та оптимізація веб-серверів (Apache, Nginx);
- основи хмарних технологій та розгортання додатків;
- основи DevOps та CI/CD процесів.

Принципи веб-дизайну та UX/UI:

- основи веб-дизайну та розробки інтерфейсів;
- принципи створення зручного та привабливого UI/UX.

Ці основи інформаційних технологій охоплюють широкий спектр знань, які потрібні для формування компетентного web-програміста в сучасному цифровому середовищі. Важливо підкреслити, що окрім hard-скілів, розвиток soft-скілів також є важливим для успішної кар'єри в ІТ.

На основі виділених hard-скілів, було створено структуру електронного курсу з web-програмування (рис. 3.2), який є електронною реалізацією підручника з web-програмування [1].

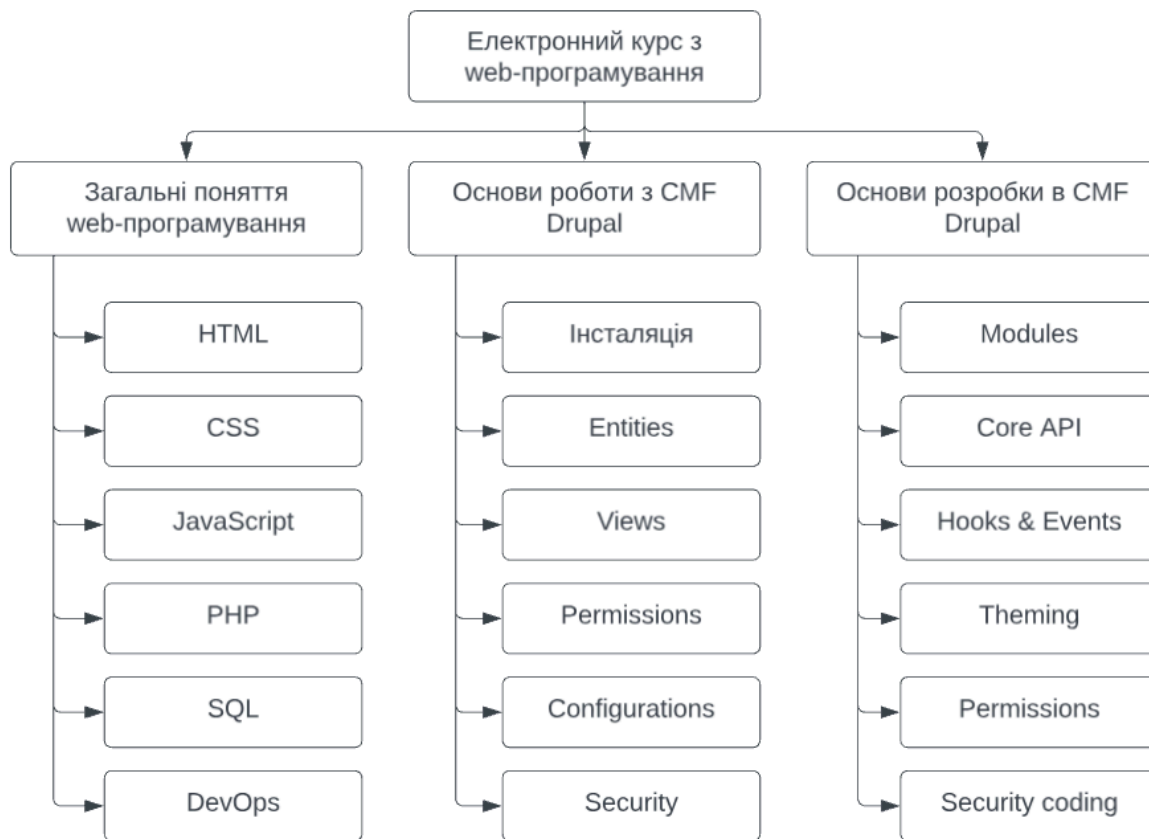


Рис. 3.2. Структура електронного навчального курсу з web-програмування.

Розроблений автором електронний навчальний курс “WEB-програмування” враховує hard-скіли, необхідні сучасному web-програмісту, а також галузеві стандарти, технічні специфікації, класичні підходи та технології, які використовуються в сучасному програмуванні на базі технологічного стеку PHP. Збалансована структура курсу дозволяє використовувати його як для підготовки здобувачів першого (бакалаврського) рівня вищої освіти, так і при підготовці та розвитку web-програмістів в ІТ-компаніях. Зміст електронного курсу з web-програмування подано в додатку Г.

Електронний навчальний курс з web-програмування являє собою багаторівневий курс підготовки для всіх рівнів web-програмістів та адаптований під основні рівні web-програмістів, а саме:

Рівень Trainee (стажер) – це вступний етап кар'єри програміста, на якому основна увага приділяється набуттю практичних навичок та теоретичних знань.

Цьому рівню відповідають знання з електронного навчального курсу:

- знання мови гіпертекстової розмітки HTML;
- знання мови каскадних таблиць стилів CSS;
- знання мови програмування PHP;
- базові знання конфігурування Drupal: типи контенту, поля, таксономія, блоки, користувачі, ролі.

- Базові знання систем контролю версій Git.

Рівень Junior передбачає, що програмісти мають невеликий досвід у розробці і зазвичай можуть вирішувати базові технічні задачі.

Цьому рівню відповідають знання з електронного навчального курсу:

- знання основ семантичної верстки;
- знання основ кросбраузерності;
- впевнені знання основ ООП в PHP;
- знання з конфігурування Drupal: views, Paragraphs, Layout Builder;
- базові знання з програмування на Drupal: роути, контролери, сервіси, форми.

Рівень Middle передбачає наявність накопиченого досвіду у програмуванні та здатність самостійно розв'язувати прикладні задачі.

Цьому рівню відповідають знання з електронного навчального курсу:

- знання типових CSS фреймворків;
- знання типових підходів з оптимізації верстки;
- базові знання з accessibility;
- впевнені знання з конфігурування Drupal: здатність самостійно проєктувати конфігурації на основі типових підходів та контрібних рішень;

- впевнені знання з програмування на Drupal: декоратори, активне використання OOP, пакетні операції, кешування;
- знання типових рішень вирішення задач;
- володіння типовими стандартами роботи з кодом;
- володіння типовими паттернами проектування.

Рівень Senior передбачає, що програмісти відрізняються високим рівнем майстерності, глибокими знаннями в різних технологічних стеках та багаторічним досвідом роботи.

Цьому рівню відповідають знання з електронного навчального курсу:

- знання з основ UI/UX проектування;
- впевнені знання з побудови інтегрованих рішень на основі конфігурацій в Drupal: активне використання комбінацій контрибних модулів для вирішення прикладних задач;
- впевнені знання з програмування на Drupal: вільне використання API для реалізації прикладних задач в комбінації з контрибним кодом;
- впевнені знання з оптимізації швидкодії;
- впевнені знання з типових рішень;
- впевнені знання з стандартів коду.

Рівень Lead (ведучий програміст) передбачає високий рівень технічної експертизи та здатність до управління IT-проектами та командами. Ліди беруть на себе відповідальність за планування, розподіл завдань, контроль виконання проєктів та взаємодію зі стейкхолдерами. Вони також визначають технічну стратегію розвитку продуктів, забезпечують високу продуктивність та мотивацію команди, і є зв'язковою ланкою між технічною командою та керівництвом компанії. Розвиток програмістів рівня Lead є індивідуальним, тому не має чітких шаблонів та форматів, і відбувається за допомогою персональної програми розвитку, яка включає знання, уміння та навички, характерні для окремого програміста.

Ця ієрархія рівнів дозволяє чітко визначити шлях професійного росту для web-програмістів, а також допомагає ІТ-компаніям у створенні ефективних команд для реалізації ІТ-проектів будь-якої складності.

Електронний навчальний курс передбачає зручне та інтуїтивно зрозуміле навчальне середовище на базі платформи OpenEDX (рис. 3.3)

Begin your course today [Start course](#)

Welcome to the **Web Development!** This course is designed for growing up with Development in Drupal

To begin, click on the **"Start/Resume Course"** button at the top of this page.

[Dismiss](#) [Show More](#)

[Expand all](#)

- Module 1: HTML Basics
- Module 2: Crafting Captivating Content
- Module 3: CSS basics
- Module 4: PHP Basics
- Module 5: Drupal site building basics
- Module 6: Code delivery basics

Course Tools

- Bookmarks
- Updates
- Launch tour

Important dates

Wed, Apr 3, 2024

Course starts

[View all course dates](#)

Course Handouts

Рис. 3.3. Інтерфейс курсу для рівня Trainee.

Курс структуровано за модулями, кожен з яких містить набір тем, які покривають знання, уміння та навички, яких має набувати здобувач у відповідності до свого професійного рівня.

В залежності від необхідності кожна тема може містити структурований текстовий контент, зображення, мультимедійні елементи у вигляді відео- та аудіо-контенту, а також необхідні файли презентацій, посилання на репозиторії програмного коду, і т. д.

Здобувач може здійснювати вільну навігацію між темами курсу для забезпечення детального проходження навчального матеріалу (рис. 3.4.).

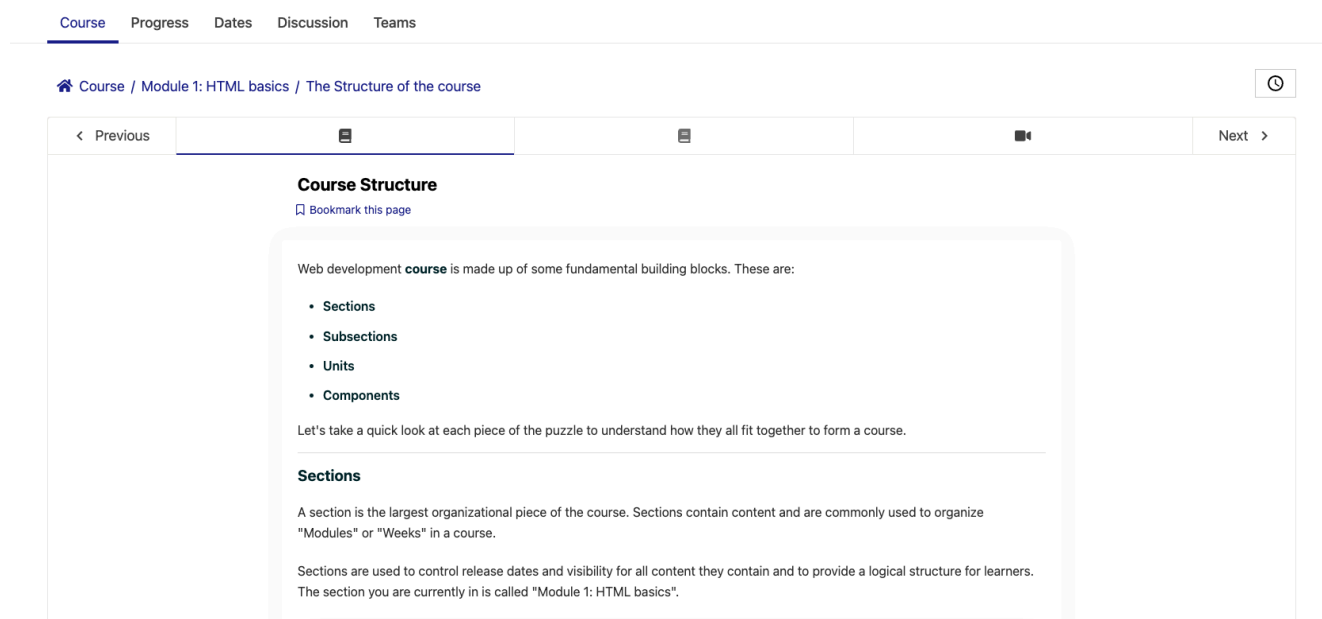


Рис. 3.4. Інтерфейс проходження курсу

Інтерактивність електронного навчального курсу забезпечує високу якість взаємодії здобувача з навчальним матеріалом в будь якій локації, за будь-якого часу доби, що реалізовує принцип неперервності навчання.

У сфері web-програмування спеціалізація IT-фахівців є ключовим фактором, який визначає їхній професійний шлях та область експертизи. Незалежно від рівня кваліфікації від Trainee до Senior програмісти обирають спеціалізацію, що відповідає їхнім інтересам, навичкам та кар'єрним амбіціям. Ця спеціалізація може охоплювати широкий спектр напрямків у галузі інформаційних технологій, включаючи фронтенд та бекенд розробку, розробку мобільних додатків, бази даних, системне програмування, штучний інтелект, машинне навчання, та інші технології.

Фронтенд програмісти зосереджені на розробці інтерфейсу та користувацького досвіду, використовуючи технології, такі як HTML, CSS, JavaScript та їх фреймворки (наприклад, React, Angular, Vue). Вони працюють над

тим, щоб веб-сайти та додатки були не тільки функціональними, але й зручними та привабливими для користувача.

Бекенд програмісти, у свою чергу, фокусуються на серверній частині додатків, обробці даних, розробці API, взаємодії з базами даних та іншими серверними технологіями. Вони забезпечують стабільність, безпеку та ефективність web-додатків і систем, використовуючи мови програмування, такі як Python, Java, Ruby, або Node.js.

Спеціалісти з розробки мобільних додатків спеціалізуються на створенні програм для мобільних пристроїв, використовуючи Swift для iOS, Kotlin для Android, або крос-платформні рішення, такі як Flutter або React Native. Їхня робота вимагає розуміння особливостей мобільних платформ, оптимізації продуктивності та споживання ресурсів.

Фахівці в галузі штучного інтелекту та машинного навчання займаються розробкою систем, здатних аналізувати великі обсяги даних, вчиться на досвіді, та виконувати завдання, які зазвичай вимагають людського інтелекту. Ця область вимагає глибоких знань у математиці, статистиці, алгоритмах та програмуванні.

Швидкий розвиток технологій, таких як штучний інтелект, машинне навчання, блокчейн, та інтернет речей (IoT), що інтегруються в web-розробку, відкривають нові можливості для створення інноваційних продуктів та послуг.

Web-додатки є ключовими інструментами для збору, обробки та аналізу великих обсягів даних, що дозволяє компаніям приймати обґрунтовані бізнес-рішення, покращувати продукти та послуги та забезпечувати кращу взаємодію з клієнтами.

Web-технології дозволяють IT-компаніям виходити на міжнародні ринки, надаючи свої продукти та послуги глобальній аудиторії. Така тенденція стимулює попит на web-розробників, здатних створювати високоякісні, масштабовані та багатомовні web-сайти та додатки.

Отже, професія web-програміста має стабільні перспективи розвитку та важливість у майбутньому. Адаптація до швидких змін у технологіях, здатність до

навчання та професійного зростання будуть ключовими для фахівців, які прагнуть досягти успіху в цій галузі.

Становлення web-розробника може здійснюватися через різні шляхи, кожен з яких має свої особливості, переваги та виклики. Основними маршрутами до професії є отримання вищої освіти, самоосвіта та участь у корпоративних програмах навчання.

Основним шляхом до кар'єри web-розробника є отримання вищої освіти в галузі комп'ютерних наук, інформаційних технологій або суміжних спеціальностей. Університетські програми надають студентам глибокі теоретичні знання та практичні навички у широкому спектрі дисциплін, включаючи програмування, алгоритми та структури даних, бази даних, web-дизайн та розробку. Вища освіта також сприяє розвитку критичного мислення, здатності до навчання та адаптації, що є важливими вміннями в динамічній сфері ІТ. Окрім того, університетське середовище надає студентам можливість для налагодження професійних контактів, участі в дослідницьких проєктах та стажуваннях.

Другим популярним шляхом до становлення web-розробником є самоосвіта. Сучасний цифровий світ надає безліч онлайн-ресурсів, таких як відеокурси, інтерактивні платформи для навчання web-програмуванню, спеціалізовані блоги, форуми та спільноти. Цей підхід вимагає високої самодисципліни та мотивації, дозволяє індивідуально підібрати темп та напрямки навчання відповідно до особистих інтересів і професійних цілей. Електронний підручник “Web-програмування” може використовуватись для самоосвіти.

По-третє, корпоративні програми навчання та стажування в ІТ-компаніях становлять ще один шлях до професії web-розробника. Багато ІТ-компаній пропонують внутрішні навчальні програми для молодих фахівців або інтенсивні курси з подальшою можливістю працевлаштування. Такі програми зазвичай зосереджені на практичних аспектах роботи та навичках, необхідних для конкретної компанії, що може забезпечити високий рівень підготовки та швидку інтеграцію в професійне середовище. Корпоративне навчання також може

включати менторство та роботу над реальними проєктами, що додає цінності до досвіду молодих спеціалістів.

Кожен із цих шляхів має свої унікальні переваги та може бути вибраний відповідно до індивідуальних переваг, життєвої ситуації та кар'єрних амбіцій майбутнього web-розробника.

3.2. Інтелектуальна система порангового оцінювання знань web-програмістів

Для перевірки знань web-програмістів початкового рівня (Trainee) використовується розроблена автором автоматизована система, що являє собою інтегроване середовище, яке дозволяє забезпечити взаємодію між програмістом що навчається, ментором та менеджером:

Як бачимо з рис. 3.3, процес оцінювання є ступінчатим, і складається з чотирьох основних етапів:

- формування та заповнення опитувальника технологій: на цьому етапі програміст заповнює опитувальник, що формується інтелектуальною інформаційною системою та надається програмісту в форматі документу Google Spreadsheet;

- генерація та проходження тесту: на цьому етапі інтелектуальна інформаційна система на основі заповнення даних опитувальника генерує тест, який проходить програміст;

- перевірка тесту: інтелектуальна інформаційна система перевіряє результати проходження тесту та генерує звіт;

- валідація звіту та надання результатів: ментор та менеджер валідують результати проходження тесту та заповненого опитувальника, оцінюють згенерований інтелектуальною інформаційною системою персональний план розвитку та надають зворотній зв'язок програмісту.

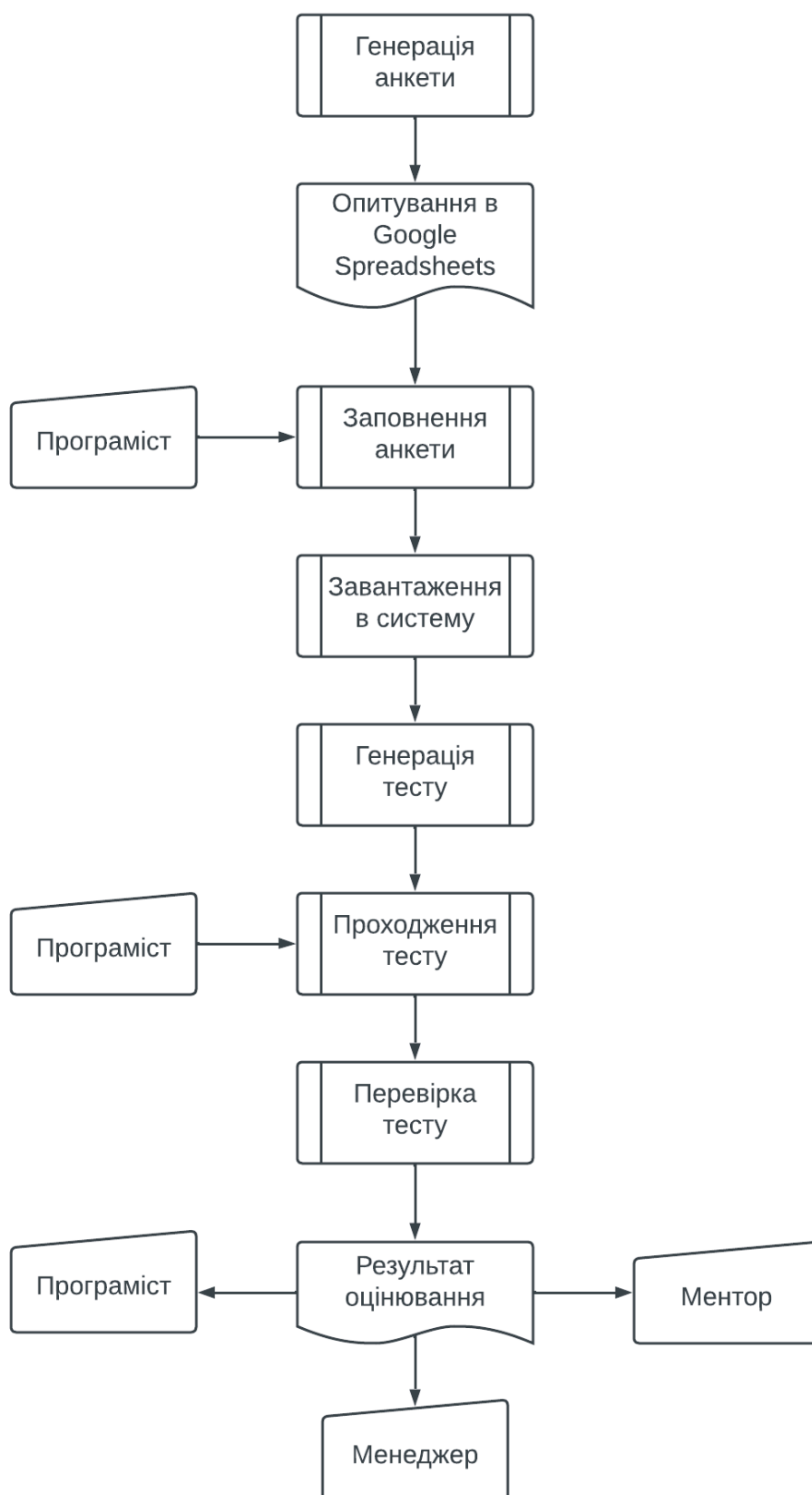


Рис. 3.3. Алгоритм автоматизованої системи перевірки знань початкового рівня

Skill	Level	Statement	Weight	Score	Status
PHP					
Basic syntax	50%	<ul style="list-style-type: none"> 0% - Completely do not understand about PHP syntax 25% - General understanding about PHP file structure, defining variables, basic syntax, basic operators 50% - understanding principals of basic output with print/echo 75% - understanding of principals of commenting and formatting 100% - understanding using "include", "require", "require_once", understanding constants 	Required ▾	0.5	Passed
Conditional operator if	50%	<ul style="list-style-type: none"> 0% - Do not understand conditional operator 25% - Understanding how to use simple conditional operator 50% - Understanding how to use full statement if - elseif - else 75% - Understanding Ternary operator 100% - Understanding conditional operator micro-optimizations 	Required ▾	0.5	Passed
Switch operator	50%	<ul style="list-style-type: none"> 0% - Do not understand conditional operator 25% - Understanding basic statement 50% - Understanding full statement 75% - Understanding optimal distribution of the selectors 100% - Understanding effective usage if / switch 	Required ▾	0.5	Passed
Cycles: for, foreach, while, do-while	25%	<ul style="list-style-type: none"> 0% - Do not understand conditional cycles 25% - Understanding for and foreach cycles 50% - Understanding for, foreach, while, do-while cycles 75% - Understanding incremental and decremental cycles 100% - Understanding cycles micro-optimizations 	Required ▾	0.25	Failed
Functions and arguments	25%	<ul style="list-style-type: none"> 0% - Do not understand functions 25% - Understanding functions definition syntax 50% - Understanding passing arguments, returning results 75% - Understanding default arguments, passing by value and reference 100% - Understanding recursive functions 	Required ▾	0.25	Failed

Рис. 3.4. Структура опитувальника для первинної оцінки знань, умінь та навичок (hard скілів) web-програміста.

Процес роботи із заповненням опитувальника виглядає як робота з документом Google Spreadsheet і тому не потребує від програміста додаткових знань, умінь та навичок (рис. 3.4):

Від web-програміста очікується заповнення колонки “Level” через вибір одного з п’яти можливих значень від 0 до 100 відсотків у відповідності із співвідношенням власних знань, умінь та навичок із шкалою з колонки “Statement”.

Створений формат опитувальника дозволяє також проведення додаткової очної сесії оцінювання ментором, де аналогічну колонку заповнює ментор web-програміста на основі заданих запитань та співвідношення отриманих відповідей то наявних критеріїв в шкалі оцінювання.

Додатковим інструментом для аналізу результатів як самим web-програмістом, так і його ментором та менеджером є звіт, який формується в опитувальнику, і є первинним інструментом рефлексії та первинних висновків для внесення змін у персональний план розвитку.

Як видно з рис. 3.5, звіт дозволяє оцінити співвідношення між результатами самооцінювання програміста, оцінкою ментора (відбувається опційно) та базовим мінімально необхідним рівнем знань, що визначений матрицею компетенцій. Візуалізація результатів аналізу оцінювання дозволяє приймати попередні рішення навіть web-програмістам початківцям, що не володіють глибокими знаннями в галузі управління та розвитку персоналу.

Подальшим етапом процесу оцінювання є проходження тесту, який формується з врахуванням максимальної гнучкості та адаптивності, у відповідності між набором питань, ранжованим відносно матриці компетенцій, та актуальними результатами первинного оцінювання з опитувальника. Так, якщо web-програміст в одному модулі показує базові знання, а в іншому високі - питання тесту також будуть відібрані та сформовані з врахуванням рівнів складності для кожного модуля.

Module	Total	Status
PHP	6.45	Failed
PHP OOP	3.8	Passed
Databases	2.5	Passed
Drupal Site Building	5.7	Passed
Drupal API & Development	5	Passed
Drupal Theming	2.05	Passed
Drush	0.5	Passed
Git	0.5	Passed
Docker & DevOps	0.5	Passed
Solution Providing	1	Passed
Total status:	28	Failed
Assessment results		
Module	Total	Status
PHP	5.75	Passed
PHP OOP	3.8	Passed
Databases	2.5	Passed
Drupal Site Building	6.25	Passed
Drupal API & Development	4.75	Failed
Drupal Theming	2.05	Passed
Drush	0.5	Passed
Git	0.5	Passed
Docker & DevOps	0.5	Passed

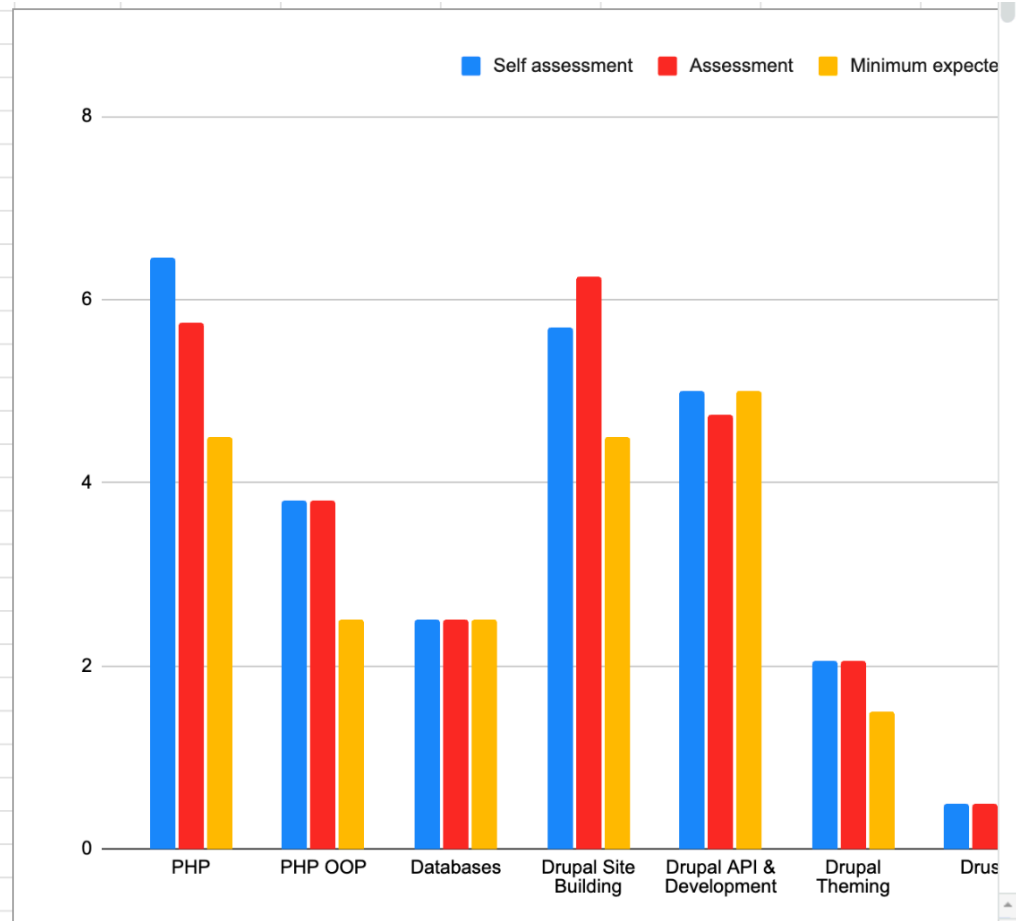


Рис. 3.5. Форма звіту порівняльного аналізу результатів опитування

Отже тестування за даною методикою реалізує адаптивний підхід, і тестування є засобом покращення точності та деталізації результатів первинного оцінювання, отриманого в результаті заповнення опитувальника.

Підсистема оцінювання знань web-програмістів (рис. 3.6), інтегрована в інтелектуальну інформаційну систему, була реалізована на основі фреймворку OpenEDX, та була розроблена з метою забезпечення ефективної, гнучкої та масштабованої оцінки компетенцій студентів.

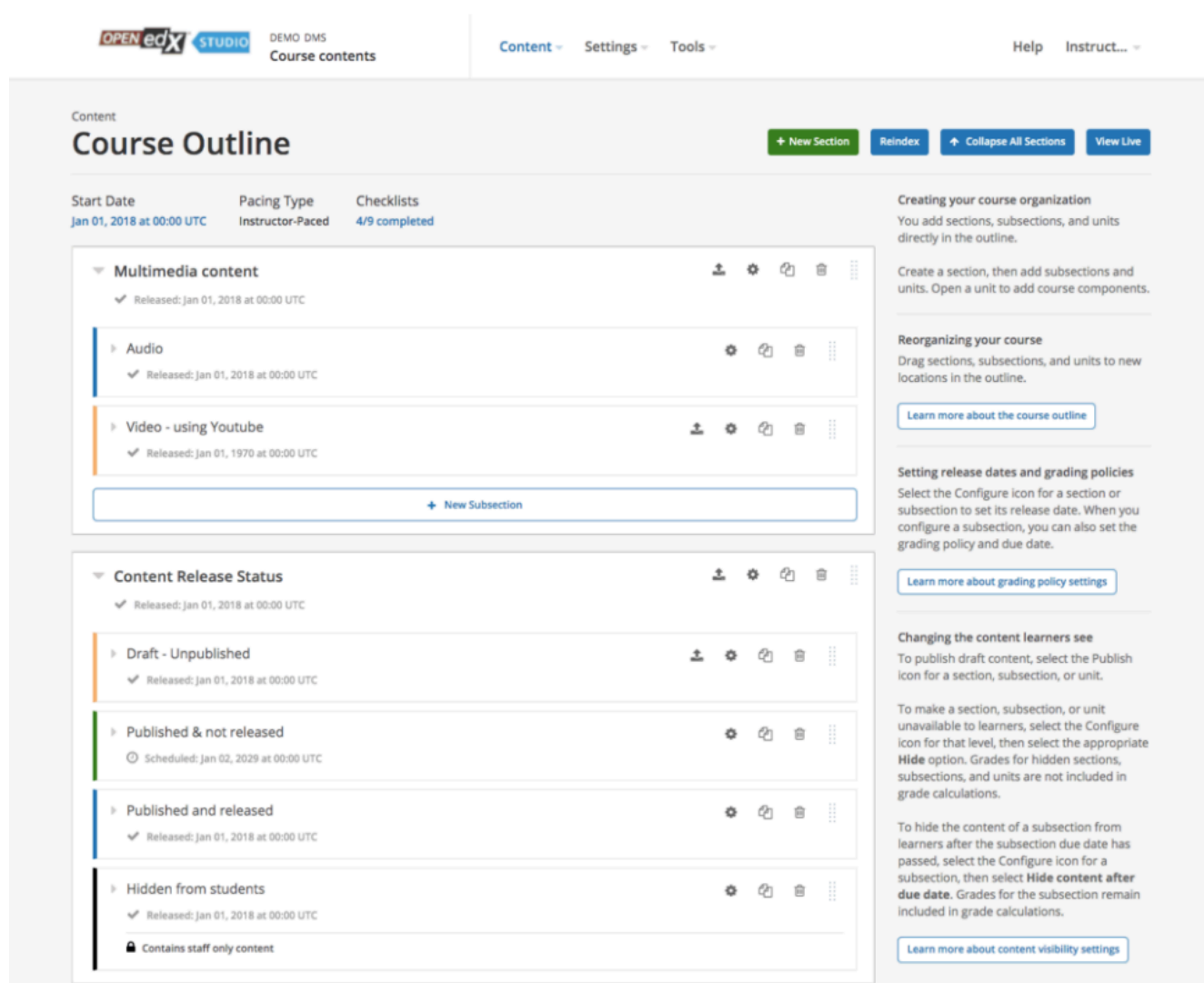


Рис. 3.6. Підсистема оцінювання знань

Основна увага при її створенні приділялася автоматизації процесів оцінювання, інтеграції сучасних технологій програмування та забезпеченні

високого рівня адаптивності до індивідуальних особливостей навчання кожного студента.

Процес розробки підсистеми тестування включав кілька ключових етапів:

- аналіз вимог і планування: На цьому етапі було визначено ключові вимоги до підсистеми, включаючи необхідність підтримки різних типів завдань (одиначний вибір, множинний вибір, завдання на введення коду тощо), можливість інтеграції з існуючою базою даних і інтерфейсами OpenEDX, а також вимоги до масштабованості та безпеки.

- проектування архітектури: В цьому контексті було прийнято рішення про використання мікросервісної архітектури для забезпечення високої масштабованості та легкості інтеграції з іншими компонентами системи OpenEDX. Такий підхід дозволив також легко впроваджувати нові функціональні можливості та проводити оновлення без зупинки системи.

- Розробка та тестування: На цьому етапі була реалізована функціональність підсистеми з використанням сучасних технологій програмування. Особлива увага приділялася безпеці та ефективності обробки даних. Тестування включало як модульні тести, так і інтеграційні тести з основною системою OpenEDX.

Розроблена підсистема тестування з використанням OpenEDX надає ряд значних переваг, зокрема:

- масштабованість: мікросервісна архітектура забезпечує легку масштабованість підсистеми, дозволяючи ефективно обробляти зростаючі обсяги користувачів та даних.

- гнучкість: система підтримує широкий спектр типів завдань, що дозволяє адаптуватись.

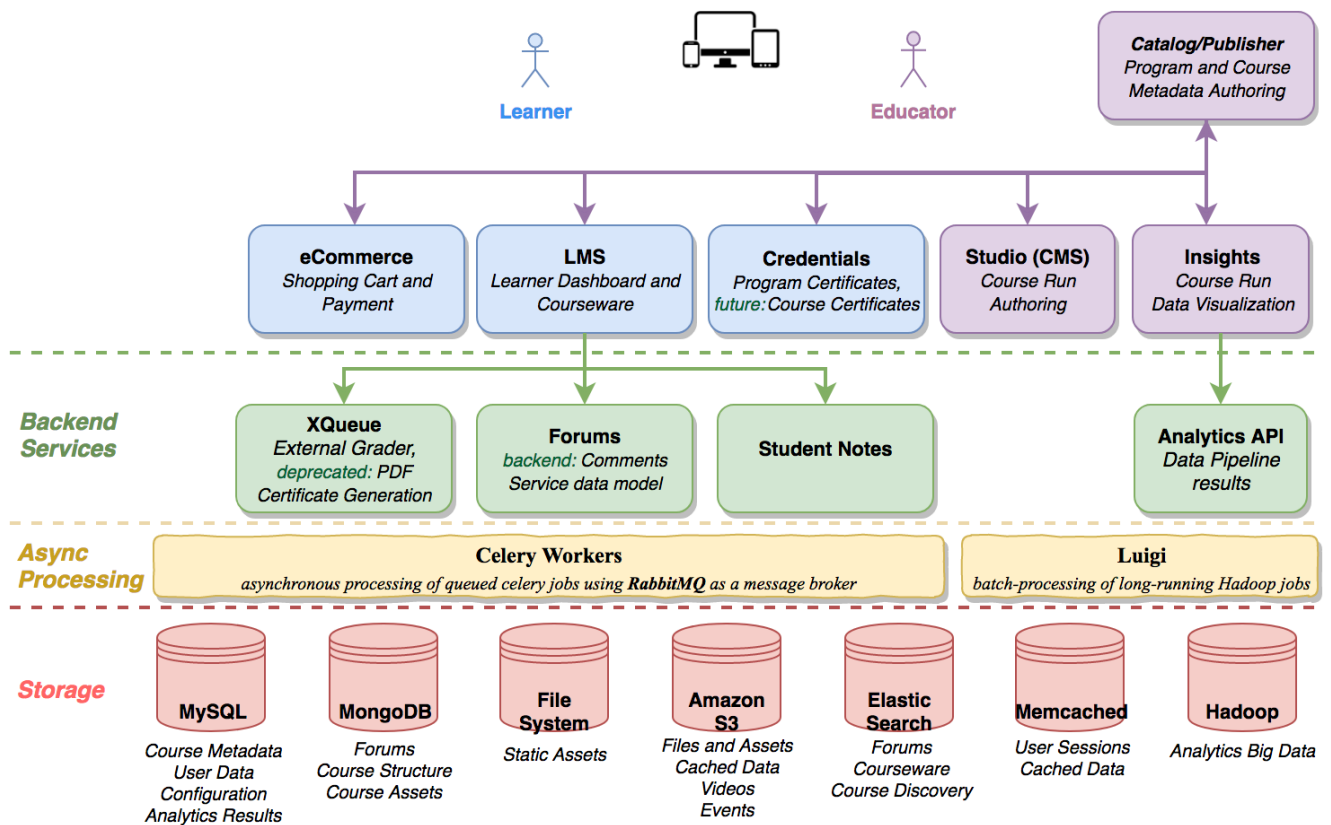


Рис. 3.7. Архітектура підсистеми оцінювання знань

Модульна архітектура підсистеми оцінювання знань (рис. 3.7) є однією з ключових характеристик, яка забезпечує її високу адаптивність, гнучкість і масштабованість. Підсистема оцінювання знань складається з декількох ключових компонентів, кожен з яких відповідає за певні аспекти функціональності платформи:

- **LMS (Learning Management System):** система управління навчанням є основним інтерфейсом для студентів, де вони можуть проходити курси, виконувати завдання, отримувати оцінки.
- **Studio:** це інструмент для авторів курсів, який дозволяє створювати і редагувати курси, додавати навчальні матеріали, створювати тести та інші елементи курсу.
- **XBlock:** це фреймворк для розробки модулів навчального контенту, який дозволяє розширювати функціональність курсів за допомогою вставних блоків. XBlocks можуть включати відео, завдання, опитування, текстові матеріали та багато іншого.

- EDX Insights: інструмент аналітики, що забезпечує збір даних про курси та студентів, дозволяючи аналізувати ефективність навчальних програм та досягнення студентів.

На основі проходження тестів та автоматизованого прийняття рішень приймається рішення щодо поточного професійного рівня та формується персональний план професійного розвитку.

Персональний план професійного розвитку web-програміста є детально спланованим документом, який окреслює цілі, стратегії та дії, спрямовані на підвищення професійних компетенцій та досягнення кар'єрного прогресу в галузі веб-розробки.

Персональний план професійного розвитку web-програміста визначає довгострокові та короткострокові професійні цілі. Ці цілі включають освоєння нових програмувальних мов, фреймворків, інструментів розробки, участь у великих проєктах, отримання сертифікатів, підвищення на посаді або перехід у суміжні області веб-розробки.

Персональний план включає самостійне вивчення нових технологій, проходження онлайн-курсів, участь у воркшопах та конференціях, читання фахової літератури та статей які надаються як рекомендовані матеріали для самостійного опрацювання.

Для забезпечення ефективності плану професійного розвитку передбачений поточний моніторинг прогресу для аналізу досягнутих результатів та внесення коректив в план з урахуванням отриманого досвіду та змін на ринку праці.

Автоматизована розробка та реалізація персонального плану професійного розвитку дозволяє web-програмістам систематично просуватися у своїй кар'єрі, адаптуватися до змін у технологіях і ринкових умовах, а також забезпечувати високий рівень професіоналізму та конкурентоспроможності в ІТ-галузі.

На основі результатів оцінювання та персональних планів розвитку інтелектуальна інформаційна система дозволяє здійснювати автоматизовану генерацію CV для учасників проєктних команд.

Автоматизована генерація резюме (CV) web-програмістів у контексті інтелектуальних інформаційних систем відіграє важливу роль у процесі формування проектних команд, забезпечуючи ефективність, об'єктивність та швидкість підбору необхідних фахівців. Ця технологія сприяє оптимізації ресурсів компанії та підвищенню якості проектних робіт за рахунок точного відбору кандидатів з необхідними навичками та досвідом.

Автоматизація процесу генерації CV дозволяє значно скоротити час, необхідний для пошуку та відбору кандидатів, завдяки використанню алгоритмів фільтрації та сортування, що базуються на певних критеріях, таких як технічні навички, досвід роботи, освітній рівень та інші ключові показники. Це дозволяє менеджерам проектів та HR-спеціалістам швидко ідентифікувати найбільш підходящих кандидатів для подальших етапів відбору.

Інтелектуальна інформаційна система мінімізує суб'єктивний фактор у процесі відбору персоналу, забезпечуючи більш об'єктивне оцінювання кандидатур. Це сприяє вибору найбільш компетентних кандидатів, що підвищує ефективність роботи проектної команди.

Точний відбір web-програмістів, що відповідають конкретним вимогам проекту, є критично важливим для успіху проектних ініціатив (рис. 3.8). Автоматизована генерація CV та використання інтелектуальної системи для відбору web-програмістів дозволяє формувати високоефективні проектні IT-команди, здатні ефективно вирішувати складні задачі та досягати проектних цілей у встановлені терміни.

Ефективна автоматизована генерація резюме (CV) є важливим інструментом для підтримки актуальності інформації про web-програмістів при формуванні проектних IT-команд.

У контексті динамічного технологічного середовища IT-компанії, навички та досвід web-програмістів швидко змінюються та розвиваються. Автоматизована генерація CV дозволяє оперативно оновлювати інформацію про фахівців, що включає нові проекти, набуті навички, отримані сертифікати та інші професійні досягнення.

The screenshot shows a user profile management interface. On the left is a sidebar with a 'Dashboard' header and buttons for 'Users', 'Projects', 'Skills', and 'Sign Out'. At the bottom of the sidebar is a link: 'Report bugs or share your ideas here'. The main content area features a table with columns: 'USER NAME', 'GRADE', 'LAST UPDATED', 'ACTIVE', and 'PREVIEW'. The first row shows 'Taras M.', 'junior', '2023-11-16', an active status indicator, and a 'User preview' link. Below the table are several form sections: 'System information' with fields for 'Grade' (Junior), 'Work Direction' (Frontend), 'CV Type' (Green & Yellow), and an 'Active' toggle; 'About' with fields for 'Name' (Taras M.), 'Experience' (Total experience 2 years), and 'Position' (Frontend Javascript developer), plus a 'Description' text area; 'Links' with an 'Add Link Item' button; and 'Hard Skills' with a grid of skill cards including TypeScript (1 year), Redux (2 years), Redux Toolkit (2 years), Git (3 years), React Native (1 year), Ant Design, Material UI, and HTML 5. Each skill card has a trash icon for removal.

Рис. 3.8. Інтерфейс підсистеми автоматизованої генерації CV.

Підсистема автоматизованої генерації CV покриває ряд задач, зокрема:

- визначення технічного напрямку web-програміста (фронтенд, бекенд, фуллстек);
- визначення технічного досвіду (кількості років практичного досвіду web-програмування);
- визначення поточної позиції web-програміста;
- централізований облік хард-скілів web-програміста (мови програмування, фреймворки, бібліотеки, підходи, технології);
- централізований облік софт-скілів web-програміста (знання мов, комунікативних навичок, критичного мислення, лідерства, вміння дотримуватись дедлайнів);
- централізований облік досвіду участі web-програміста в IT-проектах.

Сукупність даних, яку опрацьовує, систематизовує та зберігає підсистема автоматизованої генерації CV дозволяє генерувати CV і таким чином забезпечує менеджерам проєктів доступ до найактуальнішої інформації при виборі кандидатів для команди.

FRONTEND JAVASCRIPT DEVELOPER — Junior

Total experience 2 years

Taras is a forward-looking JavaScript developer interested in building usable, useful, well-constructed websites and applications. He is enthusiastic about the continuation of learning and cooperation with all members of the project team.

EXPERIENCE	HARD SKILLS	SOFT SKILLS	LANGUAGES
<ul style="list-style-type: none"> • Markup development - 3 years • JavaScript development - 3 years • Fundamental knowledge - 3 years • Testing - 1 year • Integration - 1 year 	<ul style="list-style-type: none"> • Ant Design - 1 year • CSS3 - 3 years • Firebase - 1 year • Git - 3 years • HTML 5 - 3 years • Javascript - 3 years • Jest - 1 year • Material UI - 1 year • Next.js - 1 year • React - 2 years • React Native - 1 year • Redux - 2 years • Redux Toolkit - 2 years • SCSS - 3 years • TypeScript - 1 year 	<ul style="list-style-type: none"> • Adaptability • Communication • Responsible 	<ul style="list-style-type: none"> • English - B1

Рис. 3.9. Результат автоматизованої генерації CV.

Як бачимо з рис. 3.9, автоматично згенероване CV містить всі дані, якими оперує підсистема автоматизованої генерації CV, зокрема:

- дані про технічний досвід web-програміста конвертуються в колонку experience, яка дозволяє проєктному менеджеру (або стейкхолдеру який приймає рішення) оцінити досвід та порівняти його відповідно до потреб ІТ-проєкту;

- дані про hard-скіли конвертуються в колонку hard-skills, яка відображає повний набір даних щодо hard-скілів, якими володіє web-програміст, що дозволяє технічному керівнику проєкту оцінити відповідність web-програміста до потреб ІТ-проєкту.

Дані про soft-скіли конвертуються в колонку soft-skills, яка відображає soft-скіли, якими володіє web-програміст, що дозволяє проєктному менеджменту оцінити рівень відповідності web-програміста до потреб команди ІТ-проєкту, оцінити рівень ризиків та сумісність програміста та команди. Володіння мовами є окремим видом soft-скілів, проте цей скіл подано окремою колонкою, яка дозволяє швидко оцінити рівень володіння web-програміста мовами, і таким чином оцінити ризики, пов'язані з спілкуванням між командою ІТ-проєкту та web-програмістом.

Інтелектуальна інформаційна система здатна аналізувати великі обсяги даних, ефективно фільтруючи кандидатів за заданими параметрами, такими як спеціалізація, рівень кваліфікації, досвід роботи з конкретними технологіями. Це значно підвищує швидкість та якість процесу відбору, дозволяючи визначити найбільш підходящих кандидатів для задач ІТ-проєкту.

Актуальна і точна інформація про кваліфікацію та спеціалізацію web-програмістів дозволяє формувати команди, оптимально специфіці проєкту. Автоматизоване управління CV сприяє правильному розподілу ролей і завдань у проєктній команді, заснованому на реальних навичках та досвіді її членів, що, у свою чергу, збільшує продуктивність та якість виконання проєктних робіт.

Уміння швидко знаходити та ангажувати найкращих фахівців є ключовим фактором успіху в ІТ-індустрії. Автоматизація процесу генерації та управління CV не тільки підвищує ефективність внутрішніх процесів ІТ-компанії, але й сприяє зміцненню її конкурентоспроможності на ринку, забезпечуючи здатність швидко адаптуватися до змін у технологіях та вимогах замовників.

Таким чином, розроблена підсистема автоматизованої генерації CV є значущим засобом підтримки актуальності інформації про web-програмістів, що відіграє ключову роль у формуванні проєктних команд, забезпечуючи високу якість розробки та зміцнення конкурентних позицій ІТ-компанії в ІТ-секторі.

Автоматизація формування проєктних команд в ІТ-компаніях за допомогою інтелектуальних інформаційних систем є важливим елементом, що сприяє підвищенню ефективності, гнучкості та конкурентоспроможності організацій у динамічному технологічному середовищі. Цей підхід дозволяє значно оптимізувати процес відбору персоналу, підвищити якість проєктних робіт та забезпечити ефективне використання ресурсів компанії.

Інтелектуальна інформаційна система, розроблена в рамках даного дисертаційного дослідження, здатна автоматично аналізувати великі обсяги даних про потенційних учасників проєкту, включаючи їхні навички, досвід роботи, освіту та інші релевантні характеристики. Завдяки алгоритмам машинного навчання та штучного інтелекту, система можуть визначати найбільш підходящих кандидатів для конкретних проєктних ролей, значно скорочуючи час та зусилля, необхідні для формування команди (рис. 3.10).

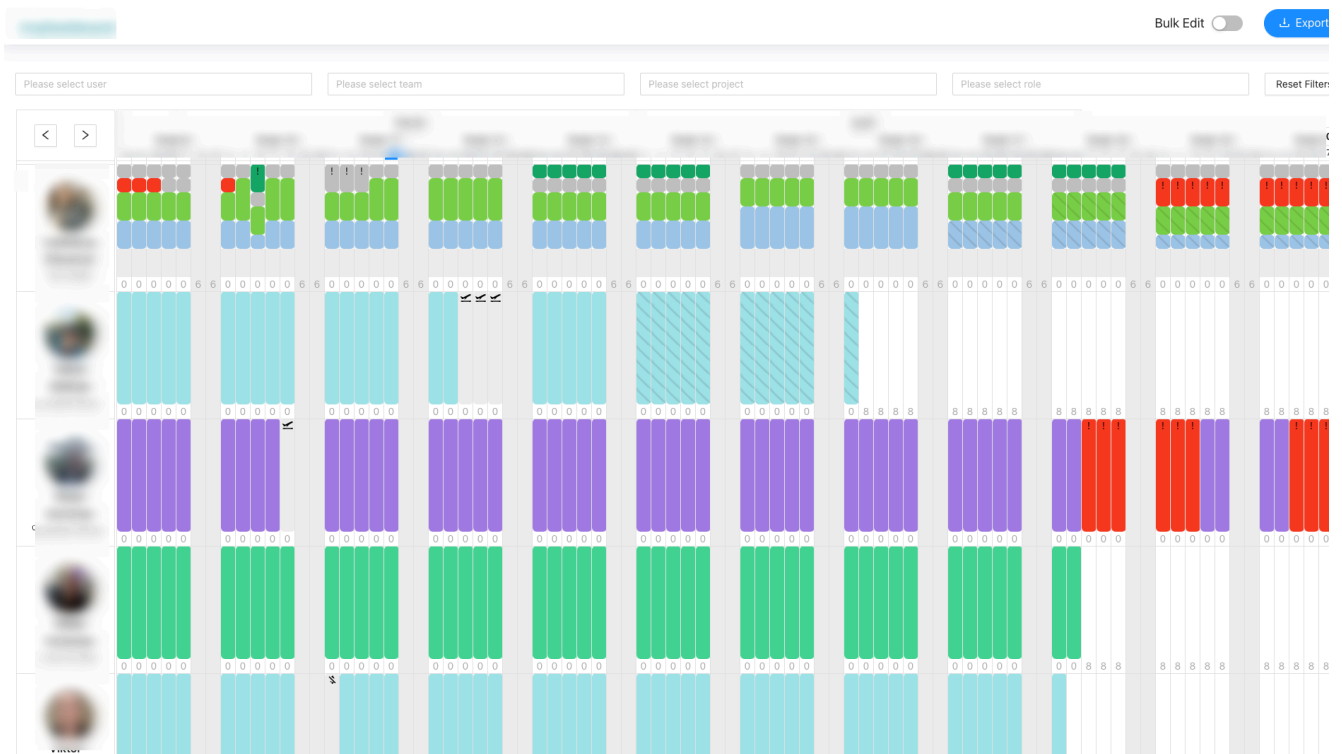


Рис. 3.10. Підсистема автоматизованого формування проєктних команд

Підсистема автоматизованого формування проєктних команд дозволяє оцінити вільні ресурси (наявних web-програмістів), які представлені у вигляді рядків таблиці, у співвідношенні з часом що представлено у вигляді стовбців таблиці (елементом квантування часу є день).

Таким чином, проєктний менеджер або технічний керівник проєкту може задати для проєкту потрібну кількість web-програмістів, їх рівень, часові рамки ІТ-проєкту, і підсистема автоматизованого формування проєктних команд відобразить перелік можливих web-програмістів, які можуть бути залучені до виконання ІТ-проєкту, ступінь доступності часових ресурсів (web-програміст може бути частково залучений до іншого проєкту, проте мати частину продуктивного часу незаконтрактованою).

Отримані дані дозволяють проєктному менеджеру або технічному керівнику додати web-програміста до ІТ-проєкту, та законтрактувати часові ресурси web-програміста.

Таким чином, автоматизоване формування команд сприяє вибору найбільш кваліфікованих фахівців для реалізації проєктів, що, у свою чергу, підвищує якість кінцевих продуктів та послуг. Інтелектуальна інформаційна система враховує не тільки технічні навички кандидатів, але й їхній досвід співпраці в команді, адаптивність та інші soft-скіли, що є критично важливими для успішної реалізації складних ІТ-проєктів.

Автоматизоване формування команд підвищує ефективність робочих процесів та якість продуктів ІТ-компаній, зміцнює конкурентоспроможність на ринку ІТ-послуг. Здатність швидко мобілізувати висококваліфіковані команди для реалізації інноваційних проєктів відкриває нові бізнес-можливості та сприяє залученню інвестицій.

3.3. Структура інтелектуальної інформаційної системи з підготовки та підбору web-програмістів в ІТ-компанії

Для забезпечення підготовки особового складу ІТ-компанії автором реалізовано інтелектуальну інформаційну систему на базі технологічної платформи OpenEDX та глибокої інтеграції з системою штучного інтелекту LangGraph, що використовує технологію OpenAI GPT 4 покоління в якості моделі генеративного штучного інтелекту.

Архітектура створеної інтелектуальної інформаційної системи складається з наступних компонентів (рис. 3.11):



Рис. 3.11. Архітектура системного рівня для інтелектуальної інформаційної системи для відбору персоналу ІТ компаній

Як бачимо з рис. 3.11, інтелектуальна інформаційна система для відбору web-програмістів являє собою сукупність інтегрованих підсистем, які об'єднані завдяки інтелектуальній підсистемі [139], що забезпечує системну інтеграцію з використанням технологій генеративного штучного інтелекту:

- електронний навчальний курс “web-програмування” надає можливості для навчання за допомогою лекційних та практичних матеріалів, і є основою для підготовки web-програмістів;
- корпоративна база знань містить приклади прикладних знань, якими діляться досвідчені web-програмісти, кодові сніпети, бібліотеку відео та медіа матеріалів;
- підсистема контролю знань надає ресурси для самоконтролю web-програмістів в процесі підготовки;
- підсистема управління користувачами дозволяє розподіляти доступи на основі використання рольової моделі;
- підсистема прийняття рішень дозволяє оцінити рівень ризиків при формуванні проєктних команд [140], оцінюванні ІТ-проєктів, відборі web-програмістів, і таким чином дає проєктному менеджеру і технічному керівнику ІТ-проєкту інструмент для ефективного прийняття рішень;
- підсистема управління проєктними командами забезпечує моніторинг основних метрик ефективності проєктних команд, таких як рівень утилізації, показники якості доставки задач, відповідність графіку виконання проєкту, і таким чином забезпечує зниження рівня ризиків та ефективне функціонування проєктної команди;
- підсистема штучного інтелекту забезпечує автоматизовану систематизацію даних на основі технології генеративного штучного інтелекту, і таким чином вирішує задачу автоматизації типових задач управління ІТ-проєктами.

На технологічному рівні архітектуру інтелектуальної інформаційної системи можна представити як сукупність інтегрованих технологічних компонентів [145].

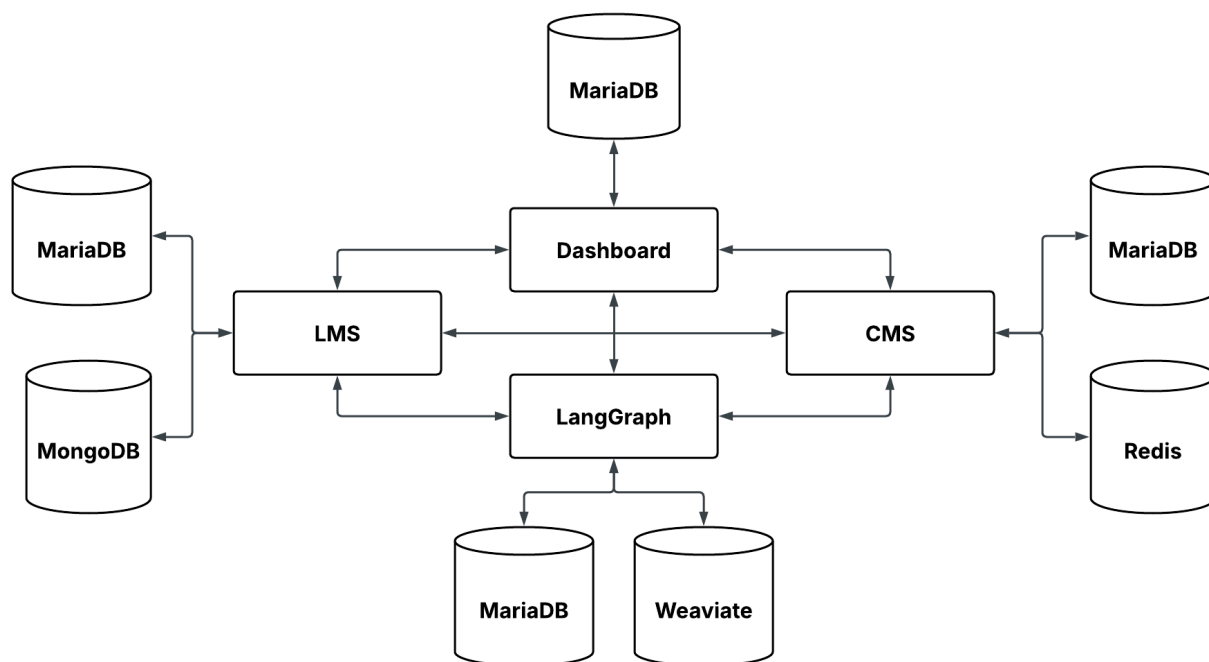


Рис. 3.12. Технологічна архітектура інтелектуальної інформаційної системи для відбору web-програмістів ІТ компаній.

З рис. 3.12. бачимо, що інтелектуальна інформаційна система для відбору та управління web-програмістів складається з чотирьох основних компонентів [147]: системи управління навчанням, системи управління контентом, системи управління ІТ-персоналом, системи підтримки прийняття рішень на основі генеративного штучного інтелекту. Всі чотири технологічних підсистеми є глибоко інтегрованими та функціонують як єдина інформаційна система в складі комплексу.

Однією з найбільш важливих підсистем інтелектуальної інформаційної системи для відбору web-програмістів для ІТ компаній є інтелектуальна підсистема підтримки прийняття рішень [150] (рис. 3.13).

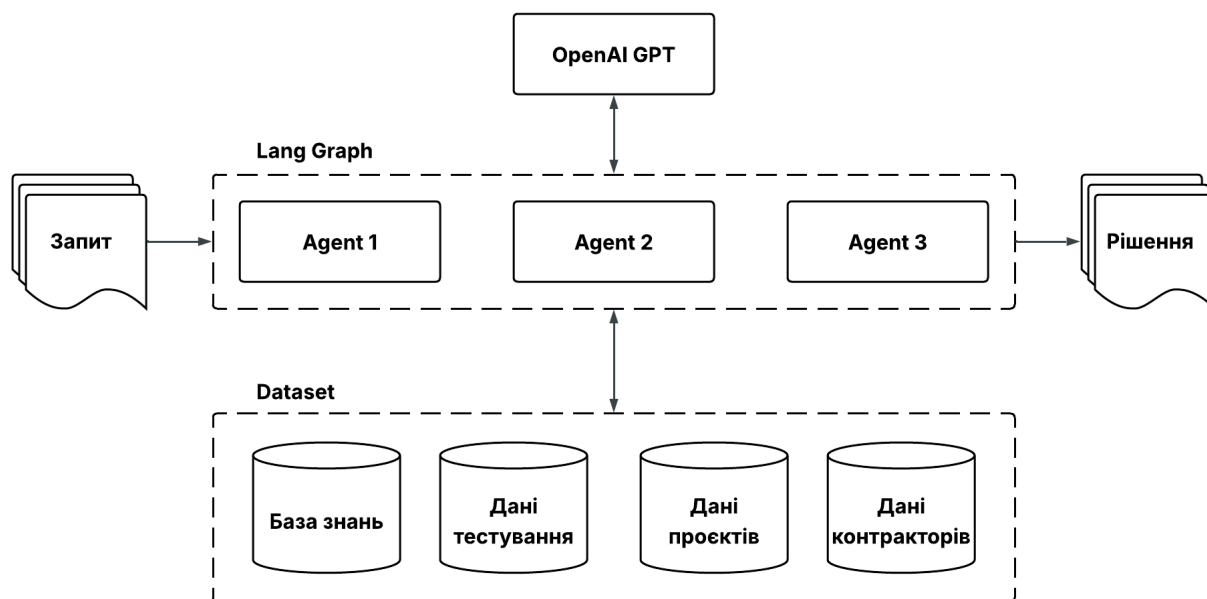


Рис. 3.13. Архітектура інтелектуальної підсистеми підтримки прийняття рішень.

Інтелектуальна підсистема реалізована як програмний комплекс, що побудований на основі технології генеративного штучного інтелекту OpenAI GPT-4, та складається з набору промптів [142-144] (шаблонів команд, для виконання чітко визначених сценаріїв у відповідності до потреб бізнес-процесу), підготовлених наборів даних - датасетів, що формуються окремими підсистемами інтелектуальної інформаційної системи, та платформи LangGraph [110], яка забезпечує організацію короткотривалого та довготривалого зберігання даних тренування діалогу нейронної мережі.

Діалог користувача з інтелектуальною інформаційною системою для підготовки та відбору web-програмістів має ранговий характер, і передбачає сегментацію послуг за основними потребами:

- навчання;
- оцінювання;
- персональні плани розвитку;
- генерація CV;
- формування проєктних команд.

Інтелектуальна інформаційна система для підготовки та відбору web-програмістів є комплексним рішенням, що інтегрує ряд компонентів для ефективного управління кар'єрою та розвитку спеціалістів у галузі веб-розробки [147-150]. Розроблена система використовує принципи штучного інтелекту та машинного навчання для оптимізації процесів навчання, оцінювання, планування розвитку кар'єри, автоматизації генерації CV та формування проєктних команд.

Компонент навчання передбачає індивідуалізовану програму підготовки для кожного web-програміста з урахуванням його поточного рівня знань, навичок та професійних інтересів. Система аналізує доступні дані про спеціаліста та рекомендує персонально сформований курс [151]. Інтелектуальна підтримка навчального процесу допомагає web-програмістам ефективно розвивати потрібні навички та адаптуватися до новітніх технологій.

Компонент оцінювання включає в себе автоматизовані тести для визначення рівня компетенцій та ефективності роботи web-програмістів. Система враховує результати оцінювання для корекції навчальних планів та подальшого професійного розвитку. Такий підхід забезпечує об'єктивне та всебічне оцінювання кваліфікації web-програмістів [148].

На основі даних про поточний рівень навичок та цілей web-програміста, система допомагає розробити персональний план розвитку. Цей план включає конкретні кроки для досягнення професійних цілей, таких як освоєння нових технологій, підвищення кваліфікації чи перехід на нову посаду. План регулярно оновлюється та коригується з урахуванням прогресу програміста [151].

Автоматизована генерація CV є ключовим компонентом інтелектуальної інформаційної системи, що дозволяє швидко формувати актуальні резюме на основі зібраних даних про навчання, проєкти, навички та досягнення спеціаліста. Цей процес спрощує підготовку до пошуку нових проєктів або робочих місць, а також забезпечує представлення найбільш повної та актуальної інформації потенційним клієнтам.

Останній компонент системи спрямований на автоматизацію процесу формування проєктних команд з урахуванням специфіки проєктів та

індивідуальних компетенцій програмістів. Інтелектуальна система аналізує вимоги до проекту та визначає найбільш підходящих кандидатів для кожної ролі, оптимізуючи розподіл ресурсів та забезпечуючи високу продуктивність роботи ІТ-команди [152, 153].

У сукупності, ці компоненти формують інтегровану систему, що забезпечує комплексний підхід до підготовки, відбору та розвитку web-програмістів в ІТ-компаніях, сприяючи підвищенню якості робочої сили та ефективності проектної діяльності [155].

Система прийняття рішень щодо оцінки знань та рекомендацій щодо присвоєння професійного рівня або призначення до проектної команди в інтелектуальній інформаційній системі базується на комплексному аналізі даних, використанні алгоритмів машинного навчання та штучного інтелекту [157-160]. Процес прийняття рішень можна розділити на кілька ключових етапів.

На початковому етапі інтелектуальна інформаційна система збирає та аналізує всю доступну інформацію про кандидатів або контракторів, включаючи їх знання, досвід роботи, професійні навички, результати тестувань та оцінок, а також історію участі у проектах [160]. Ця інформація використовується для створення детальних профілів, які відображають загальну кваліфікацію та спеціалізацію кожного web-програміста.

Для оцінки знань та компетенцій співробітників система використовує алгоритми машинного навчання, які можуть ідентифікувати закономірності та зв'язки у великих обсягах даних [161-164]. В тому числі, використовуються методи машинного навчання для прогнозування рівня кваліфікації на основі історичних даних про успішність web-програміста у подібних проектах.

На основі аналізу даних та прогнозів, сформованих алгоритмами машинного навчання, система рекомендує присвоєння конкретного професійного рівня співробітнику, наприклад, Junior, Middle, Senior, залежно від кваліфікації, досвіду та навичок. Такий підхід забезпечує об'єктивність та консистентність у визначенні рівнів кваліфікації web-програміста.

Для призначення контракторів до проєктних ІТ-команд система враховує не тільки їх професійний рівень, але й специфіку проєкту, вимоги до навичок, досвід роботи з певними технологіями та потенціал для розвитку [169]. Штучний інтелект аналізує ці параметри для створення оптимального складу команди, що відповідає усім проєктним вимогам та максимізує шанси на успіх [170-174].

Завдяки інтеграції цих компонентів, інтелектуальна інформаційна система дозволяє значно підвищити ефективність процесів підбору та розвитку web-програмістів в ІТ-компанії, забезпечуючи високу якість виконання проєктів і сприяючи професійному зростанню співробітників [177-180].

Висновки до третього розділу

Методологія створення інтелектуальної інформаційної системи, призначеної для багаторівневого навчання та підбору web-програмістів, має істотні відмінності від традиційної форми навчання, оскільки поєднує в собі методи і рівні активації пізнавальної діяльності та багаторангового підбору web-програмістів за допомогою штучного інтелекту на основі складеної матриці скілів, необхідних для підготовки web-програмістів, виникла необхідність в розробці інтелектуальної інформаційної системи з підготовки та відбору web-програмістів в ІТ компаніях [184-189].

Реалізація даного інструментального та прикладного програмного забезпечення служить для розв'язання задач з підготовки та підбору програмістів для ІТ-компаній, що спеціалізуються в галузі web-розробки.

Розробка інтелектуальної інформаційної системи з підготовки та відбору web-програмістів в ІТ компаніях дасть можливість майбутнім web-програмістам поглибити та закріпити свої знання в галузі web-програмування, а ІТ-компаніям - забезпечити ефективний підбір, навчання та професійний розвиток контракторів, покращити ефективність створення CV та формування проєктних ІТ-команд.

Розроблений програмний продукт є інтегрованою інтелектуальною інформаційною системою [200-205], що забезпечує повний цикл операцій,

призначених для підготовки контракторів ІТ-компаній, а саме web-програмістів. Розроблена інтелектуальна інформаційна система може використовуватись як в ІТ-компаніях, так і при підготовці здобувачів за напрямом F2 Інженерія програмного забезпечення та F3 Комп'ютерні науки.

РОЗДІЛ 4.

РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ БАГАТОРІВНЕВОГО НАВЧАННЯ WEB-ПРОГРАМІСТІВ В ІТ-КОМПАНІЇ, ЇЇ ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА

4.1. Методика проведення навчання та відбору програмістів в ІТ-компанії за допомогою інтелектуальної інформаційної системи

Основними цілями та задачами даної інтелектуальної інформаційної системи є:

- отримання базових та практичних знань з web-програмування відповідно до рівня web-програміста: trainee, junior, middle, senior, lead;
- навчання web-програмування в електронному курсі “Web-програмування”;
- проходження тестів згідно заявленого рівня;
- отримання практичних навиків роботи з інтелектуальною інформаційною системою в рамках інтенсивного навчання, підвищення кваліфікації, контролю знань;
- автоматизація процесів прийняття рішень при відборі web-програмістів на відповідну посаду згідно результатів тестування;
- трансформація управлінських рішень із застосуванням систем штучного інтелекту;
- підвищення ефективності і якості підготовки та відбору web-програмістів для командної роботи в ІТ-проектах.

Автоматизований навчальний курс з web-програмування для практикуючих програмістів складає 1020 годин, з яких 148 годин лекційних, 192 години практичних занять, 516 годин для роботи над ІТ-проектом та 64 годин заплановано для менторства, консультацій та майстер-класів.

контролю знань. Система передбачає два режими роботи: для здобувачів вищої освіти та для практикуючих програмістів.

На рис. 4.1. зображено алгоритм підготовки та відбору web-програмістів в ІТ-компанії.

Всі бази даних являють собою сукупність взаємопов'язаних об'єктів, кожний з яких призначений для виведення на екран пристрою (ПК, ноутбука, мобільного гаджета) запитуваної інформації.

Всі бази даних працюють в двох режимах: редагування та додавання інформації.

Виділимо основні кроки які відбуваються в процесі підготовки та відбору web-програмістів в ІТ-компанії, згідно алгоритму роботи інтелектуальної інформаційної системи, представленого на рис. 4.1.

Крок 1. Оголошення вакансії web-програміста:

Крок 2. Обробка вхідних запитів:

Крок 3. Обробка заявки кандидата:

Крок 4. Перевірка CV:

Крок 5. Невідповідність вимогам CV:

Крок 6. Запрошення на первинну співбесіду:

Крок 7. Первинна співбесіда:

Крок 8. Первинну співбесіду не пройдено:

Крок 9. Запрошення на технічну співбесіду:

Крок 10. Технічна співбесіда:

Крок 11. Технічну співбесіду не пройдено:

Крок 12. Відхилення можливості подальшого навчання:

Крок 13. Пропозиція пройти підготовку:

Крок 14. Процес навчання web-програмуванню

Крок 15. Валідація успішності підготовки:

Крок 16. Підготовка не була успішною:

Крок 17. Підготовка була успішною:

Крок 18. Формування оферу:

Крок 19. Початок випробного терміну:

Крок 20. Закриття вакансії:

Процес відбору web-програмістів за допомогою інтелектуальної інформаційної системи, опрацьований автором дисертації. Підсистеми прийняття рішень враховують по

Метою підготовки web-програмістів є проходження лекційного матеріалу, що міститься в інтелектуальній інформаційній системі, який класифікований у відповідності до посад trainee, junior, middle, senior, lead web-розробника.

Кожен багаторівневий курс завершується автоматизованим контролем знань, в результаті якого встановлюється кількість набраних балів, вказуються виявлені недоліки, надаються відповідні рекомендації [208-210].

Для Trainee web-розробника курс розрахований на 126 годин, з яких 24 години складають лекційні заняття, реалізовані у вигляді відео-лекцій з можливістю задавати питання в корпоративному чаті, 48 годин практичних занять в форматі домашніх робіт, 70 резервованих для розробки практичного проєкту та 2 годин відведених для консультацій та захисту практичного проєкту.

Для Junior web-розробника курс розрахований на 130 годин, з яких 28 години складають лекційні заняття, реалізовані у вигляді відео-лекцій з можливістю задавати питання в корпоративному чаті, 48 годин практичних занять в форматі домашніх робіт, 70 резервованих для розробки практичного проєкту та 2 годин відведених для консультацій та захисту практичного проєкту.

Для Middle web-розробника курс розрахований на 252 години, з яких 48 годин складають лекційні заняття, реалізовані у вигляді відео-лекцій, онлайн майстер-класів та вебінарів з можливістю живої комунікації та можливості задавати питання в корпоративному чаті, 48 годин практичних занять в форматі домашніх робіт, 120 резервованих для розробки практичного проєкту та 18 годин відведених для онлайн консультацій з персональним ментором та 2 години для захисту практичного проєкту.

Для Senior web-розробника курс розрахований на 512 годин, з яких 48 годин складають лекційні заняття, реалізовані у вигляді онлайн майстер-класів та

вебінарів з можливістю живої комунікації та можливості задавати питання в корпоративному чаті, 48 годин практичних занять в форматі воркшопів, 120 резервованих для самостійних досліджень в рамках персонального плану розвитку, 40 годин виділених на менторство, та 256 години для виконання практичного проєкту в контексті внутрішніх задач компанії, а також захисту практичного проєкту.

Для Lead розробника курс не лімітований часом та структурою, і наповнення курсу створюється персонально у відповідності до планів розвитку та корпоративної потреби у вузьких технологічних вимогах щодо кожного конкретного фахівця рівня Lead.

Зразок робочої програми електронного навчального курсу для рівня Trainee наведено в додатку Д.

Розроблений електронний навчальний курс “Web-програмування” може змінюватись, доповнюватись відповідно з апробацією в системі навчального процесу закладів вищої освіти та ІТ-компаній.

4.2. Постановка дослідних задач

При розробці інтелектуальної інформаційної системи для інтенсифікації підготовки та відбору web-програмістів в ІТ-компанії ставились дослідні цілі з оцінювання дидактичної ефективності розроблених інструментальних та прикладних засобів інформаційних технологій.

Перевірялась гіпотеза: розроблена інтелектуальна інформаційна система скорочує час на навчання при підготовці web-програмістів, поліпшує успішність в групі, а при відборі web-програмістів скорочує час на проведення співбесід та технічного оцінювання та підвищує ефективність прийняття рішень на кожному етапі відбору web-програмістів.

Експеримент складався з двох частин. У першій частині з цією метою були виділені параметри, за якими проводилось оцінювання розробленої інтелектуальної інформаційної системи. Цими параметрами є:

- середні групові показники вхідного контролю;
- середні групові показники вихідного контролю;
- дані про час навчання на кожному з етапів;
- дані про час відповіді при контролі знань;
- дані про показник відмови під час проходження навчання.

В другій частині експерименту проводилось оцінювання знань, умінь та навичок з побудови ІТ-проєкту тестованим здобувачем. Оцінювання виконаної версії ІТ-проєкту проводилось за такими критеріями:

- відповідність виконаного ІТ-проєкту початковим вимогам (acceptance criteria);
- стан готовності ІТ-проєкту (definition of done);
- відповідність технічних рішень в ІТ-проєкті до існуючих стандартів (best practices);
- відповідність фактично витраченого часу до запланованого (estimates).

Напрямок даних досліджень - оптимізація підготовки web-програміста, підвищення інтенсифікації підготовки web-програміста з будь-яким рівнем початкових професійних знань за допомогою інтелектуальної інформаційної системи.

4.3. Методика проведення експерименту з оцінювання знань здобувачів із застосуванням інтелектуальної інформаційної системи

В нижчеописаному експерименті була зроблена спроба оцінити взаємозалежність між рівнем якості навчання з використанням традиційного лекційно-практичного підходу, та рівнем якості навчання із застосуванням для багаторівневого навчання інтелектуальної інформаційної системи.

Теорія методу контрольних груп є однією з основоположних концепцій у сфері експериментального дослідження. Вона базується на принципі порівняння результатів між двома або більше групами, з яких одна (або декілька) піддається впливу експериментального втручання, тоді як інша (контрольна група)

залишається без такого втручання. Цей метод дозволяє оцінити вплив певного фактора або інтервенції, забезпечуючи більш об'єктивні та надійні результати [41, 42].

Цей науковий підхід до експерименту покладено в основу роботи з контрольними групами для оцінки знань web-програмістів. Основоположним аспектом є об'єктивність в оцінюванні впливу, де контрольна група виступає як бенчмарк для порівняння з групою, що піддається експериментальному втручання. Таке порівняння дозволяє виявити реальний вплив експериментальних втручань на навчання та розвиток навичок web-програмістів [41, 42].

Досліджувані. В експерименті взяли участь 184 здобувачів першого (бакалаврського) рівня вищої освіти 2 та 3 курсів Міжнародного економіко-гуманітарного університету імені академіка Степана Дем'янчука спеціальності F2 Інженерія програмного забезпечення та та F3 Комп'ютерні науки.

Були вибрані рівноцінні навчальні групи, з яких 1 та 3 були контрольними, а групи 2 та 4 - експериментальними. Поділ на групи проведений рівномірно, на основі результатів зимової сесії, перед проведенням експерименту.

Експериментальні групи отримували навчання з використанням розробленої інтелектуальної інформаційної системи, що включає спеціалізоване програмне забезпечення, інтерактивні методики навчання та онлайн-платформи для самостійного вивчення матеріалу. Контрольні групи продовжували навчання за традиційною методикою без активного використання зазначених інформаційних технологій. Такий підхід дозволив оцінити вплив застосування інтелектуальної інформаційної системи на ефективність навчального процесу та відбір web-програмістів.

У контексті IT-компанії AnyforSoft експеримент було спрямовано на оцінку впливу використання новітніх інформаційних технологій на процеси відбору нових web-програмістів.

Застосування методики контрольних груп дало змогу об'єктивно порівняти результати, отримані в обох групах, і визначити статистичні відмінності в

ефективності навчання та процесах відбору web-програмістів. Це дозволило не лише підтвердити або спростувати вихідну гіпотезу дослідження, але й виявити конкретні аспекти, в яких використання інформаційних технологій має найбільш виражений позитивний вплив.

Експериментальний матеріал. Для оцінювання знань з web-програмування були відібрані теми та відповідні тести з курсу “Web-програмування”.

План і процедура експерименту. Лекційний матеріал, лабораторні роботи були міжгруповою змінною, тобто кожна з груп отримувала завдання на вивчення одного й того ж основного контенту. Ефект навчання розглядався як внутрішньогрупова та внутрішньоіндивідуальна змінна.

До уваги брались результати лише тих досліджуваних, які приймали участь в експерименті, вони тестувались двічі - на початку навчання (вихідний контроль) та в кінці навчання (вихідний контроль).

Результати першої частини експериментів (навчання за допомогою інтелектуальної інформаційної системи) представлені у вигляді таблиці 4.1.

Таблиця 4.1. Середні групові показники вхідного контролю

№ групи	Якісна успішність (%)	Загальна успішність	Середній бал	Кількість здобувачів
1к	19.1%	74.0%	46.55	46
2е	17.2%	65.3%	41.25	46
3к	16.3%	60.8%	38.55	46
4е	15.7%	57.3%	36.5	46

Після проведення навчання контрольні та експериментальні групи було протестовано, і за результатами вхідного контролю було отримано показники, представлені в таблиці 4.2.

Таблиця 4.2. Середні групові показники вихідного контролю

№ групи	Якісна успішність	Загальна успішність	Середній бал	Кількість здобувачів
1к	62%	82.4%	72.2	46
2е	82%	98.2%	90.1	46
3к	76.6%	83.1%	79.85	46
4е	85.2%	98.7%	91.4	46

Для оцінки дидактичної ефективності використання інформаційних технологій у навчальному процесі було проведено експеримент за методом контрольних груп на базі спеціальності F2 "Інженерія програмного забезпечення" та та F3 Комп'ютерні науки. Експериментальна група нараховувала 38 студентів, які використовували інноваційні інструментальні та прикладні засоби навчання інтелектуальної інформаційної системи, тоді як контрольна група складалася з 38 студентів, що продовжували навчання за традиційною методикою.

Основними критеріями оцінки ефективності навчання були визначені:

- успішність студентів;
- засвоєння навчального матеріалу;
- час, витрачений на навчання.

Для аналізу результатів було використано статистичні методи, зокрема t-критерій Стьюдента для незалежних вибірок.

Проведемо розрахунки, щоб визначити статистичну значущість відмінностей між експериментальною та контрольною групами. Зокрема, обчислимо середні значення успішності, стандартні відхилення та розрахуємо t-критерій для двох незалежних вибірок.

Середній бал успішності в експериментальній групі складає 77.26 зі стандартним відхиленням 10, а в контрольній групі середній бал становить 64.31 зі стандартним відхиленням 12.

Проаналізуємо отримані дані за допомогою статистичних розрахунків.

Для оцінки ефективності навчального процесу за допомогою методу контрольних груп та визначення статистичної значущості відмінностей між експериментальною та контрольною групами було використано t-критерій Стьюдента для незалежних вибірок.

Формула t-критерію Стьюдента для незалежних вибірок:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (1)$$

де: \bar{x}_1, \bar{x}_2 - середні значення вибірок (експериментальної та контрольної відповідно);

s_1^2, s_2^2 - дисперсії вибірок;

n_1, n_2 - кількість спостережень у вибірках.

Для кожної групи було визначено середнє значення (\bar{X}) та стандартне відхилення (s), які відображають успішність студентів.

Використовуючи вищенаведену формулу, ми розраховуємо t-статистику, яка дозволяє оцінити ступінь відмінності між середніми значеннями двох груп. Розрахунок було здійснено на основі даних успішності студентів, що відповідають нормальному розподілу.

P-значення отримуємо як результат порівняння розрахованої t-статистики з табличними значеннями розподілу Стьюдента. Це значення допомагає зрозуміти, наскільки ймовірно отримати спостережувану різницю між групами, якщо насправді жодної різниці не існує (тобто припускаючи, що нульова гіпотеза вірна).

$$p = P(T > |t|) \quad (2)$$

Проведемо розрахунки, використовуючи наступні дані, отримані в результаті експерименту:

$\bar{x}_1 = 75$ - середнє значення успішності експериментальної групи;

$\bar{x}_2 = 65$ - середнє значення успішності контрольної групи;

$s_1 = 10$ - стандартне відхилення успішності експериментальної групи;

$s_2 = 12$ - стандартне відхилення успішності контрольної групи;

$n_1 = 32$ - кількість студентів у експериментальній групі;

$n_2 = 34$ - кількість студентів у контрольній групі.

Обчислимо t-критерій Стьюдента:

$$t = \frac{75 - 65}{\sqrt{\frac{10^2}{32} + \frac{12^2}{34}}} = 3.686$$

Отже, розрахована t-статистика показує значення близько 3.686. Порівнюючи це значення з критичним значенням t-розподілу (яке залежить від ступенів свободи та обраного рівня значущості), ми можемо зробити висновок про статистичну значущість різниці між групами. Розрахунок показав, що різниця між середніми балами успішності експериментальної та контрольної груп є статистично значущою, що дозволяє підтвердити ефективність використання інформаційних технологій у навчальному процесі.

У рамках дослідження ефективності використання інноваційних інформаційних технологій для підготовки контракторів до отримання чергового професійного рівня в ІТ-компанії було проведено експеримент за методом контрольних груп. Експеримент залучив дві групи контракторів: експериментальну групу з 38 учасників, які застосовували інтелектуальну інформаційну систему у своєму навчанні, та контрольну групу з 38 учасників, що дотримувалися традиційних методів підготовки.

Середній час підготовки у експериментальній та контрольній групах становив 4 місяців. Ці результати вказують на потенційну ефективність впровадження інноваційних технологій у процес навчання.

Для визначення статистичної значущості спостережуваної різниці між групами було використано t-критерій Стьюдента для незалежних вибірок.

Розрахунок здійснювався на основі середніх значень часу підготовки та відомих стандартних відхилень для обох груп.

Проведемо розрахунки, використовуючи наступні дані, отримані в результаті експерименту:

$\bar{x}_1 = 4.8$ - середнє значення часу підготовки експериментальної групи;

$\bar{x}_2 = 6$ - середнє значення часу підготовки контрольної групи;

$s_1 = 0.4$ - стандартне відхилення експериментальної групи;

$s_2 = 0.5$ - стандартне відхилення контрольної групи;

$n_1 = 25$ - кількість контракторів в експериментальній групі;

$n_2 = 28$ - кількість контракторів в контрольній групі.

Обчислимо t-критерій Стьюдента:

$$t = \frac{4.8 - 6}{\sqrt{\frac{0.4^2}{25} + \frac{0.5^2}{28}}} = -9.69$$

Це значення t-статистики вказує на сильну статистичну значущість різниці у середньому часі підготовки між експериментальною та контрольною групами. З цих даних можна зробити висновок, що застосування новітніх інформаційних технологій в процесі підготовки суттєво скорочує час, необхідний для досягнення нового професійного рівня серед контракторів у IT-компанії.

Для отримання групових відмінностей співставимо розрахункові та експериментальні дані між собою, для цього проведемо порівняння дві метрики:

- кількість безпомилкових відповідей під час контролю знань;
- час на виконання тестового проєкту.

Результати залежності між кількістю відповідей та часом на виконання тестового проєкту подано в таблиці 4.3

Залежність між часом виконання та кількістю безпомилкових відповідей в розрізі контрольних та експериментальних груп.

№ групи	Кількість безпомилкових відповідей	Час на виконання тестового проєкту
1к	65	72
1к	61	76
1к	63	72
1к	67	73
1к	61	75
1к	81	65
1к	75	68
1к	74	70
1к	64	78
1к	63	77
1к	91	61
1к	62	78
1к	64	74
1к	51	81
1к	58	76
1к	65	73
1к	48	85
1к	47	78
1к	62	73
2e	78	65
2e	77	65
2e	79	63
2e	82	61
2e	85	57
2e	76	63
2e	88	51
2e	91	45
2e	74	61
2e	76	64
2e	74	63

2e	63	69
2e	85	53
2e	72	63
2e	78	61
2e	61	79
2e	68	70
2e	88	53
2e	73	62
3κ	68	70
3κ	64	71
3κ	68	69
3κ	71	65
3κ	65	71
3κ	82	61
3κ	77	61
3κ	75	68
3κ	68	71
3κ	65	73
3κ	88	67
3κ	63	76
3κ	64	75
3κ	56	78
3κ	61	79
3κ	65	73
3κ	58	79
3κ	59	81
3κ	64	73
4e	81	63
4e	83	52
4e	82	61
4e	82	60
4e	85	54
4e	76	62
4e	89	49

4e	92	44
4e	74	59
4e	76	64
4e	74	63
4e	71	62
4e	87	51
4e	73	59
4e	78	64
4e	71	61
4e	72	59
4e	91	48
4e	84	58

Використовуючи час на виконання тестового проєкту в якості трудомісткості через відповідність витрат часу на виробничу операцію (виконання тестового проєкту), можемо отримати інтегральні показники ефективності за кожною з груп.

Залежність між часом виконання завдання та кількістю правильних відповідей має чітко виражений характер. Візуальне представлення даних можна організувати у вигляді графіків залежності між часом виконання проєкту та кількості правильних відповідей при вихідному оцінюванні у розрізі контрольних та експериментальних груп

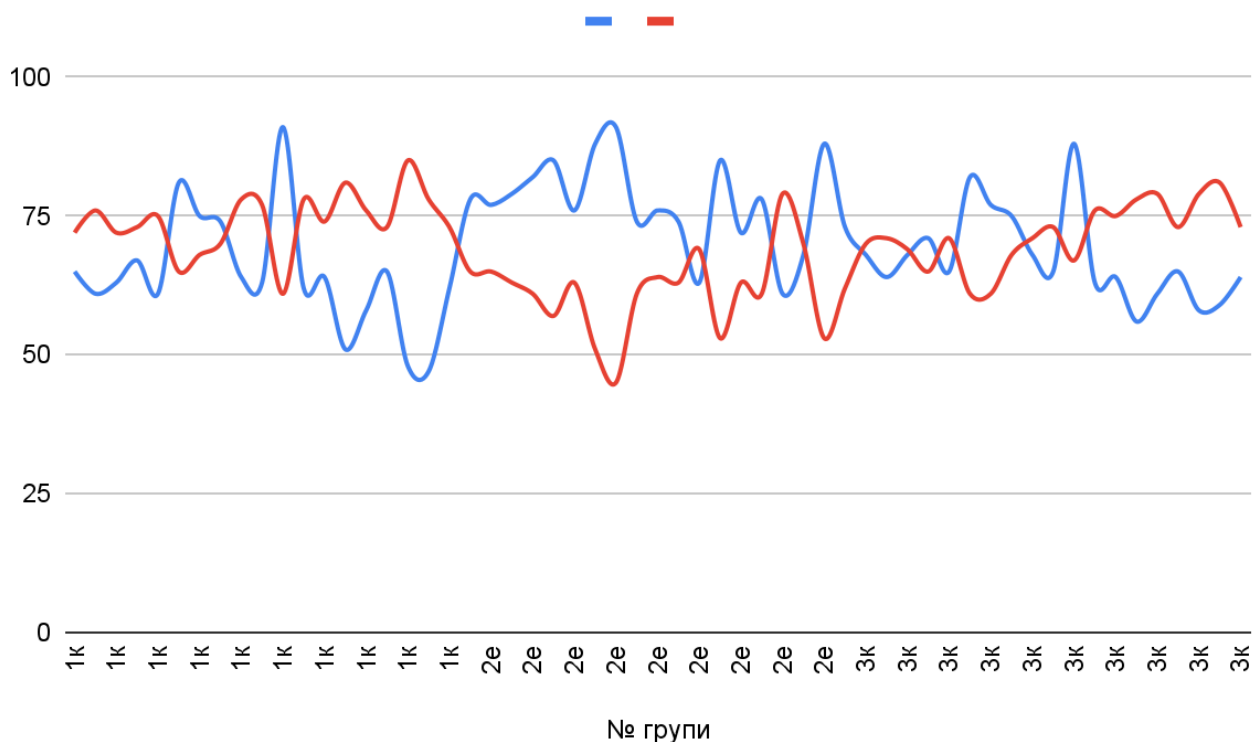


Рис. 4.2. Графік працездатності контрольних та експериментальних груп.

З графіка працездатності (рис. 4.2) видно, що в експериментальних групах за меншого затраченого часу на розробку тестового ІТ-проєкту спостерігається більш висока продуктивність.

В контрольних групах при більшій затраті часу на виконання тестового ІТ-проєкту продуктивність значно нижча, ніж в експериментальних групах. Спостерігалась більша кількість помилок при розробці ІТ-проєктів. Статистична обробка анкетування вказала на високий відсоток бажань студентів навчатись розробці ІТ-проєктів із застосуванням необхідних знань з інтелектуальної інформаційної системи у порівнянні з традиційною формою навчання (88% проти 12%).

4.4. Результати експериментів

Практичне значення роботи було перевірено за допомогою проведення експерименту, що включав перевірку гіпотези, щодо очікувано вищої ефективності оцінювання та відбору web-програмістів зі застосуванням інтелектуальної інформаційної системи, в порівнянні з традиційними методами рекрутингу.

Для проведення експерименту було використано та адаптовано практичний процес пошуку, відбору, та оцінювання фахівців на посаду Junior Drupal Developer в компанію AnyforSoft, який проводився в серпні-жовтні 2025 року. Для первинного пошуку кандидатів було використано платформи Djinni, DOU, LinkedIn а також існуючі корпоративні бази кандидатів та реферальні посилання соціальних мереж.

Всього було проведено три етапи відбору для закриття трьох рівноцінних вакансій, з загальною кількістю 144 кандидати. В кожному з етапів було відібрано по 48 кандидатів, які пройшли попередній скрінінг з рекрутером. З кандидатів в кожному періоді було сформовано 2 групи: тестову і контрольну.

Вимоги до кандидатів на етапі скрінінгу складали:

- від 1 року комерційного досвіду роботи з фреймворком Drupal;
- володіння технологіями семантичної верстки з використанням HTML та CSS;
- володіння синтаксисом та базовими конструкціями мови PHP;
- володіння синтаксисом та вміння будувати комплексні запити мовою SQL;
- володіння технологією сайт-білдингу в фреймворку Drupal;
- досвід створення модулів в Drupal: роутинг, форми, контролери, сервіси, Entity API, Plugin API, Config API;
- володіння технологіями темізації в Drupal: структура темплейтів, Twig, препроцес-функції;
- володіння базовими DevOps практиками для локального розгортання з використанням технології контейнеризації Docker.

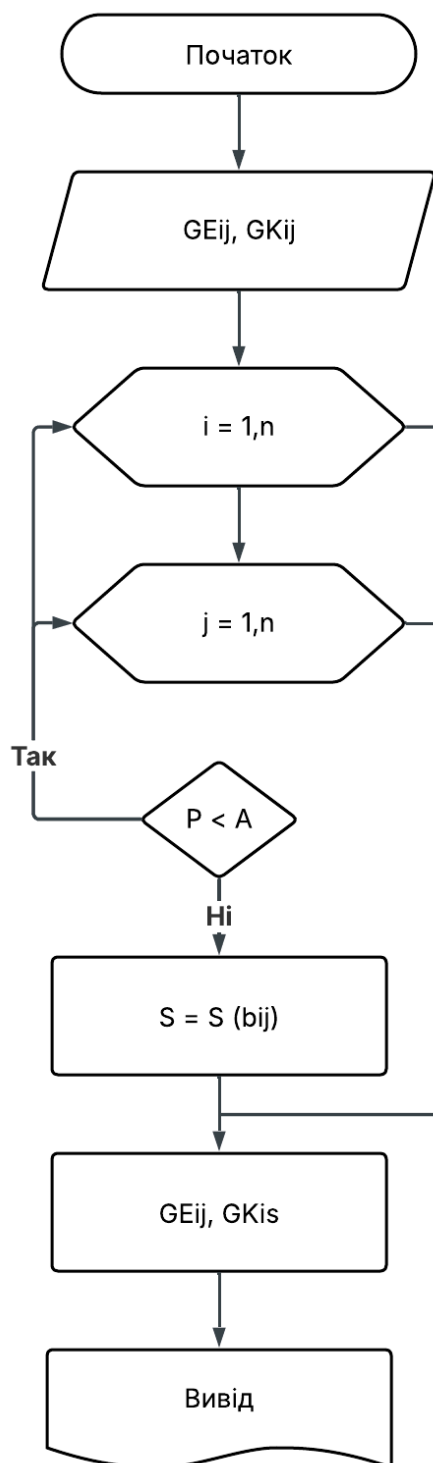


Рис. 4.3. Блок-схема експерименту з відбору web-програмістів засобами інтелектуальної інформаційної системи

Процес проведення експерименту складається з окремих блоків (рис. 4.3), які визначені набором рекрутингових операцій:

- Блок 1. ІТ-компанія відкриває вакансію web-програміста, публікуючи вимоги на відповідних платформах та соціальних мережах (Djinni, DOU, Linkedin).

- Блок 2. Кандидати, що відповідають базовим вимогам та проходять базовий скринінг розділяються на дві групи, однакові за чисельністю, де GE - експериментальна група, в якій проводиться відбір web-програмістів засобами інтелектуальної інформаційної системи; GR - експериментальна група, в якій здійснюється відбір web-програмістів традиційними засобами: проведенням співбесід, виконання тестових завдань, та інших типових операцій ІТ-рекрутингу.

- Блок 3. Відкриває цикл по кількості кандидатів на посаду web-розробника.

- Блок 4. Перевірка відповідності умов та валідація правильності відповідей на генерований масив D сформований з загального масиву питань K.

- Блок 5. Підрахунок суми S балів, які отримав кандидат на посаду web-програміста, з врахуванням грейдингової матриці та матриці вагових коефіцієнтів. Ранжування кандидатів за отриманими сумами балів.

- Блок 6. Порівняння отриманих балів в групах GE, GR за всіма критеріями.

- Блок 7. Остаточна співбесіда з працевлаштування, оголошення результатів з відбору web-програмістів.

Інтегральна оцінка рейтингу:

$$R = \sum_{i=1}^n w_i \cdot s_i ,$$

де:

w_i - вага критерію,

s_i - оцінка кандидата за критерієм.

Математична модель відбору web-програмістів.

Нехай $C = \{c_1, c_2, \dots, c_m\}$, множина кандидатів на посаду web-програміста.

Для кожного кандидата на кожну посаду (позицію) визначається набір критеріїв: $K = \{k_1, k_2, \dots, k_n\}$.

Перелік критеріїв для відбору web-програмістів за грейдом Junior подано в додатку Б (скіл-матриця).

Для кожного кандидата c_j формується вектор оцінок:

$$S_j = (S_{j1}, S_{j2}, \dots, S_{jn}),$$

де $S_{ji} \in [0, 1]$ – нормалізована оцінка за критерієм k_i .

Нормалізація:

$$S_{ji} = \frac{x_{ji} - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

Визначаємо вагові коефіцієнти критеріїв:

$$W = (w_1, w_2, \dots, w_n),$$

де

$$\sum_{i=1}^n w_i = 1, w_i \geq 0,$$

ваги визначаються інтелектуальною інформаційною системою автоматично за допомогою засобів агентного штучного інтелекту, на основі співвідношення критерія до елемента скіл матриці та матриці вагових коефіцієнтів.

Загальний рейтинг кандидата на посаду web-програміста визначається як:

$$R_j = \sum_{i=1}^n w_i \cdot S_{ji}$$

Встановлюємо порогову модель відсіву, для цього вводимо мінімально допустимий рівень: R_{min} .

Кандидат допускається до наступного етапу співбесіди з працевлаштування з техлідом, якщо $R_j \geq R_{min}$.

Встановлюємо модель жорстких обмежень (фільтрацію) для потрібного рангу web-програмістів. Для критично важливих вимог: $S_{ji} \geq \alpha_i$,

Наприклад:

RНР Basic syntax: $S_{j2} \geq 0.7$

RНР Class syntax $S_{j3} \geq 0.6$

Складемо модель прогнозу успішності кандидата на посаду web-програміста (ML-компонент). Ймовірність успішної роботи кандидата визначимо, як:

$$P_j = f(S_j),$$

де f - навчена модель (fine tuned LLM).

Представимо інтегральну модель рішення:

$$Q_j = \lambda R_j + (1 - \lambda)P_j,$$

де $0 \leq \lambda \leq 1$.

На підставі цієї моделі відбувається ранжування кандидатів, формується впорядкований список рекомендованих web-програмістів на кожен ранг у відповідності до вимог, заданих скіл-матрицею та матрицею вагових коефіцієнтів:

$$Q_1 \geq Q_2 \geq \dots \geq Q_m.$$

Вклад кожного критерію для кожного рангу web-програміста рахується окремо, це дозволяє інтелектуальній інформаційній системі пояснити, чому кандидат на вакантну посаду був рекомендований або відхилений.

Інтерпретація результатів обчислюється наступним чином:

$$I_{ji} = w_i \cdot S_{ji}, \text{ де}$$

w_i - вага критерію,

S_{ji} - сумарна оцінка кандидата за критерієм.

За результатами проведеного оцінювання було отримано наступні диференціальні показники оцінок кандидатів по тестовій (таблиця 4.4) та контрольній (4.5) групам.

Таблиця 4.4.

Диференціальні показники оцінок по тестовій групі

Номер кандидата в групі	Диференціальна оцінка
1	62
2	66
3	52
4	76
5	84
6	83
7	57
8	68
9	65
10	64
11	72
12	57
13	53
14	60
15	57
16	78
17	68
18	78
19	74
20	63

21	54
22	62
23	79
24	76
25	66
26	75
27	66
28	54
29	90
30	73
31	57
32	47
33	80
34	71
35	76
36	89
37	69
38	83
39	71
40	52
41	84
42	72
43	69
44	72
45	63
46	72
47	63
48	92
49	85

50	67
51	66
52	75
53	66
54	54
55	90
56	73
57	57
58	47
59	80
60	71
61	76
62	89
63	69
64	83
65	71
66	52
67	84
68	72
69	69
70	72
71	63
72	92

Проаналізуємо отримані результати, побудувавши гістограму розподілу диференціальних оцінок тестової групи (рис. 4.4).

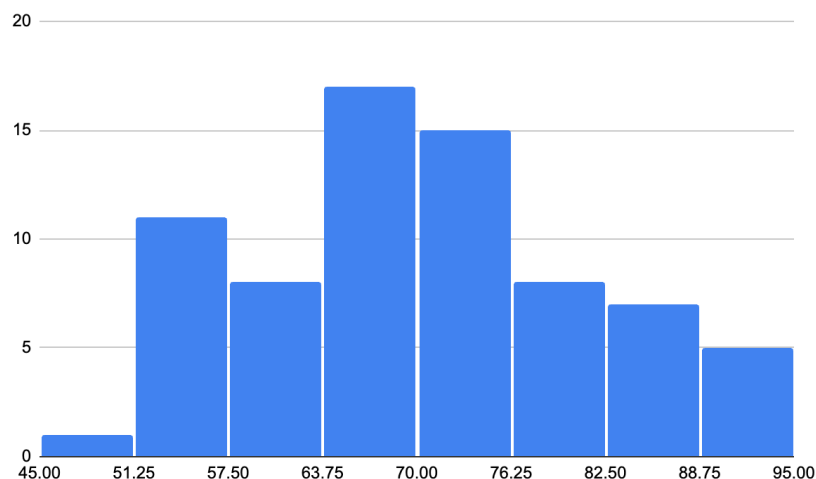


Рис. 4.4. Гістограма розподілу диференціальних оцінок тестової групи

Середнє значення вибірки для вибірки тестової групи становить 74.82

Проаналізуємо гістограму розподілу вибірки тестової групи:

1. центральну частину розподілу становлять кандидати з діапазону оцінок 64-76 балів;
2. центр розподілу зміщений в діапазон 64-76 балів;
3. медіана розподілу на відмітці 75 балів;
4. Стандартне відхилення 8.79;
5. Дисперсія 72.22.
6. сегментація і фокусування розподілу відносно центру свідчить вищу якість оцінювання, про що свідчить низька варіативність та диференціюванн рівня і відсікання слабких кандидатів.

Таблиця 4.5.

Диференціальні показники оцінок по контрольній групі

Номер кандидата в групі	Диференціальна оцінка
1	89
2	70
3	74
4	64

5	64
6	81
7	59
8	76
9	84
10	71
11	80
12	68
13	72
14	88
15	69
16	72
17	79
18	68
19	84
20	80
21	92
22	71
23	78
24	75
25	85
26	74
27	69
28	75
29	60
30	89
31	84
32	45

33	69
34	87
35	89
36	86
37	80
38	65
39	68
40	78
41	86
42	82
43	73
44	74
45	77
46	67
47	62
48	78
49	77
50	70
51	71
52	64
53	89
54	65
55	81
56	82
57	75
58	80
59	77
60	70
61	76

62	75
63	87
64	65
65	78
66	71
67	79
68	70
69	73
70	59
71	74
72	69

Проаналізуємо отримані результати, побудувавши гістограму розподілу диференціальних оцінок контрольної групи (рис. 4.5).

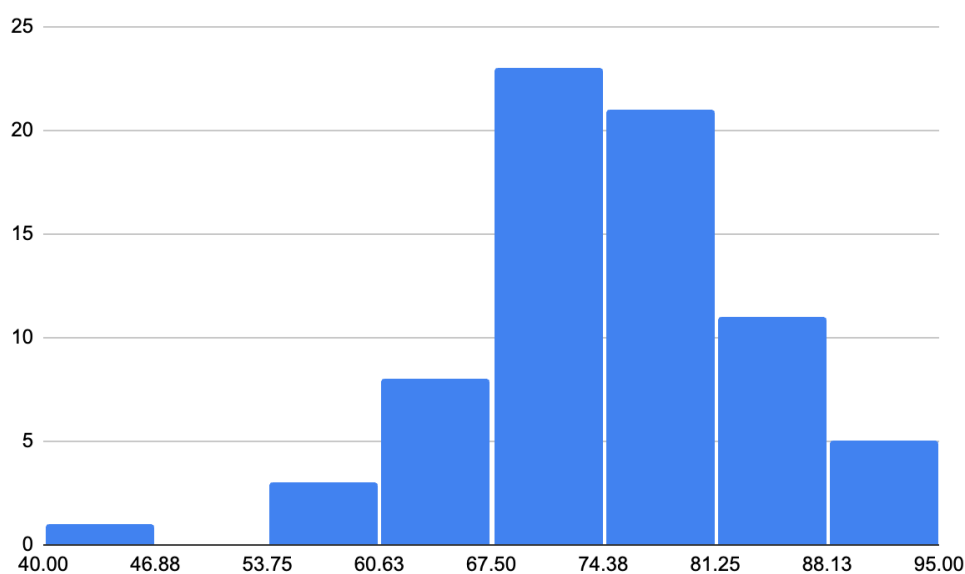


Рис. 4.5. Гістограма розподілу диференціальних оцінок контрольної групи

Середнє значення вибірки для вибірки контрольної групи становить 70.24

Проаналізуємо гістограму розподілу вибірки контрольної групи:

1. центральну частину розподілу становлять кандидати з діапазону оцінок 67-81 балів;
2. медіана розподілу 69 балів;
3. стандартне відхилення 10.91;
4. дисперсія 118.97;
5. наявні відносно симетричні “хвости розподілу” щодо центральної частини розподілу в інтервалах 54-67 балів, та 81-95 балів, що виявляють аномально слабких та аномально сильних кандидатів;
6. слабка сегментація і згладжування розподілу відносно центру свідчить нижчу якість оцінювання відносно результатів розподілу тестової групи.

Чітка сегментація розподілу свідчить про доволі нестабільний рівень точності оцінювання в контрольній групі

Результати аналізу результатів свідчать, що застосована технологія відбору та оцінювання знань web-програмістів засобами інтелектуальної інформаційної системи забезпечує статистично значуще підвищення диференціальної оцінки кандидатів порівняно з контрольною групою, оціненої з традиційними підходами.

Висновки до четвертого розділу

Проведено постановку експерименту, який був спрямований на визначення ефективності застосування розробленої інтелектуальної інформаційної системи у процесі підготовки та відбору web-програмістів. Описано метрики та критерії оцінювання, які були використані для аналізу результативності системи.

Методика проведення експерименту включає в себе комплексну оцінку якісних та кількісних характеристик процесу підготовки та відбору кандидатів з використанням розробленої системи. Результати експерименту, підтвердили високу ефективність розробленої інтелектуальної інформаційної системи у контексті підготовки та відбору web-програмістів. Зокрема, було встановлено значне зростання якості відбору кандидатів та оптимізацію часових та ресурсних витрат на процес відбору. Також система сприяла підвищенню об'єктивності та

прозорості процесу оцінювання кандидатів, що, у свою чергу, позитивно позначилося на репутації компанії серед потенційних працівників.

Отримані результати свідчать про значний потенціал впровадження інтелектуальних інформаційних систем в процеси підготовки та відбору web-програмістів в IT-компаніях.

Головна особливість отриманих кількісних даних пов'язана з використанням різних методик навчання (комп'ютерної та традиційної), з їх впливом на поліпшення успішності в групах при відборі web-програмістів.

Всі отримані результати свідчать про ефективність застосування інтелектуальної інформаційної системи при багаторівневому навчанні та відборі web-програмістів в порівнянні з традиційним способом навчання та традиційним відбором web-програмістів.

ВИСНОВКИ

Дисертаційна робота присвячена дослідженню необхідності створення нової інформаційної технології для управління багаторівневим навчанням та підбором web-програмістів в ІТ-компаніях.

В рамках проведених досліджень було запропоновано новий підхід до автоматизованого процесу підготовки, відбору та оцінювання web-програмістів, який базується на використанні модифікованої скіл-матриці, та використанні матриці вагових коефіцієнтів. Згідно створеного набору ієрархічних професійних компетенцій web-програміста для ІТ-компанії оцінено актуальність та релевантність кожної компетенції у відповідності до професійного рангу. Модифікована матриця компетенцій та матриця вагових коефіцієнтів склали основу датасету інтелектуальної інформаційної системи для автоматизованого оцінювання знань, умінь та навичок web-програмістів у відповідності до рівня ієрархії.

Отримані такі основні результати дослідження:

- проведено аналіз сучасних наукових праць, технологій, методів та підходів до оцінювання знань, умінь та навичок web-програмістів; проведено аналіз існуючих вимог та галузевих стандартів до знань, якими мають володіти web-програмісти різних рівнів; проведено декомпозицію бізнес-процесів рекрутингу, управління розвитком ІТ-персоналу, формування планів їх персонального розвитку, підбору web-програмістів в склад проектних ІТ-команд; проведено інтеграцію розроблених інформаційних систем в об'єднану корпоративну інформаційну систему на базі технологічної екосистеми Amazon AWS; проведено порівняльний аналіз створеної інтелектуальної інформаційної системи та традиційного підходу з відбору web-програмістів в ІТ-компаніях.

- розроблено класифікацію знань, необхідних для підготовки web-програмістів; на основі розробленої класифікації знань було розроблено модель комп'ютерного курсу з web-програмування, орієнтованого як на здобувачів вищої освіти за напрямками F2 Інженерія програмного забезпечення та F3

Комп'ютерні науки, так і на практикуючих web-програмістів з професійними рівнями Trainee, Junior, Middle, Senior; на основі розробленої класифікації знань було створено підручник “Web-програмування”; розроблено матрицю hard-скілів для профілю Drupal-розробників рівня trainee, junior, middle, senior, lead; розроблено інтелектуальну інформаційну систему з використанням технології штучного інтелекту LangGraph для підготовки та аналізу даних професійних компетенцій web-програмістів, динаміки результатів первинного та проміжного оцінювання та автоматизованого формування персональних планів їх професійного IT-розвитку;

- запропоновано та застосовано застосовано в розробленій інтелектуальній інформаційній системі технологію штучного інтелекту LangGraph для аналізу даних професійних компетенцій web-програмістів в процесі підготовки та відбору web-програмістів в IT-компанії; запропоновано використання технології штучного інтелекту LangGraph для IT-проектів з вирішення багатокритеріальної задачі оптимізації по формуванню проектних команд; запропоновано та застосовано матрицю вагових коефіцієнтів в якості цифрового засобу визначення важливості скілів та компетенцій для забезпечення ефективного прийняття рішень щодо рівня знань, умінь та навичок в розрізі грейд / компетенція та диференційованої оцінки відповідності до тайтлу;

- створено авторський підручник “ WEB-програмування” для підготовки web-розробників та web-програмістів; створено модифіковані матриці компетенцій для напрямку бекенд розробки з використанням CMF Drupal на основі розробленого авторського підручника “ WEB-програмування”; створено шаблони оцінювання за модифікованим методом 180° та проведено ітерації оцінювання IT-персоналу, зокрема web-програмістів та порівняльний аналіз результатів;

- проведено проведено експериментальну перевірку результатів дослідження в ході експерименту, який був спрямований на визначення ефективності застосування розробленої інтелектуальної інформаційної системи у

процесі підготовки та відбору web-програмістів; описано метрики та критерії оцінювання, які були використані для аналізу результативності системи.

Отримані результати проведених досліджень свідчать про ефективність застосування інтелектуальної інформаційної системи в процесах управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії у порівнянні з традиційним способом навчання та відбору.

Результати дисертаційної роботи можуть використовуватись в навчальному процесі вищих навчальних закладів для підготовки здобувачів вищої освіти за спеціальностями F2 Інженерія програмного забезпечення та F3 Комп'ютерні науки, підготовки та відбору web-програмістів в ІТ-компаніях за напрямком РНР та Drupal розробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Emily Rose McRae, Kaelyn Lowmaster. 9 Future of Work Trends for 2026. Gartner. 2026. URL: <https://www.gartner.com/en/articles/future-of-work-trends>
2. Mauro Cazzaniga, Florence Jaumotte, Longji Li, and others. Gen-AI: Artificial Intelligence and the Future of Work. Internatioal Monetary Fund. 2024. URL: <https://www.imf.org/-/media/files/publications/sdn/2024/english/sdnea2024001.pdf> (дата звернення: 08.10.2025)
3. Статегія цифрового розвитку інновацій України до 2030 року. URL: https://winwin.gov.ua/assets/files/WINWIN_%D0%A1%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D1%96%D1%8F%20%D1%96%D0%BD%D0%BD%D0%BE%D0%B2%D0%B0%D1%86%D1%96%D0%B9%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82.pdf (дата звернення: 08.10.2025)
4. Розпорядження Кабінету Міністрів України від 2 грудня 2020 року №1556-р Про схвалення Концепції розвитку штучного інтелекту в Україні. 2020. URL: <https://zakon.rada.gov.ua/laws/show/1556-2020-%D1%80#Text> (дата звернення: 08.10.2025)
5. Dimitri Japaridze, Shorena Maghlakelidze, Ekaterine Shaverdashvili, Davit Tutberidze. The Role of Higher Education in Reducing the Mismatch between Graduates' Skills and Employer Expectations. Матеріали міжнародної наукової конференції "Education, Research, Practice". Том 2024. URL: (дата звернення: 08.10.2025)<https://journals.4science.ge/index.php/erp/article/view/3140>
6. Т. С. Хархаліс. Дослідження співпраці ІТ-підприємств з ВНЗ у підготовці кадрів. Менеджмент та підприємництво в Україні: етапи становлення та проблеми розвитку. № 2 (14), 2025. С. 267-273. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2025/dec/41332/101225maket16122025-269-275.pdf> (дата звернення: 08.10.2025)
7. М. Швардак Професійна підготовка майбутніх ІТ-фахівців у закладах фахової передвищої освіти. Наукові записки. Серія: Педагогічні науки. 2025. Вип.

22. С. 111-116. URL:

<https://pednauk.cusu.edu.ua/index.php/pednauk/article/view/2337/2351> (дата звернення: 10.10.2025)

8. Р. Н. Кветний. Проблеми ефективної організації університетської освіти в ІТ галузі. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ІНЖЕНЕРІЯ. 2017. №2. С.20-23. URL:

<https://itce.vntu.edu.ua/index.php/itce/article/view/678/418> (дата звернення: 10.10.2025)

9. Вища освіта в Україні. «Skills mismatch»: розрив між освітою та потребами ринку праці. Освіта.ua. 2025. URL: <https://osvita.ua/vnz/94853> (дата звернення: 10.10.2025)

10. А. МакГован, М. та Д. Ендрюс Невідповідність на ринку праці та продуктивність праці: дані РІААС, Робочі документи Департаменту економіки ОЕСР, № 1209. Париж: Видавництво ОЕСР. 2015.

11. А. Амоа. Роль освіти та навчання у зміцненні робочої сили, підприємстві та управлінні організацією. Hillary Publishing. 2023. URL: <https://www.hilarispublisher.com/open-access/the-role-of-education-and-training-in-strengthening-the-labor-force.pdf> (дата звернення: 10.10.2025)

12. Б. Пандья, У. Рухі та Л. Паттерсон. Підготовка майбутньої робочої сили до 2030 року: роль вищих навчальних закладів. Front. Educ. Sec. Higher Education. Том 8. 2023.

13. К. Венер, А. Гріп та Г. Пфайфер. Чи підбирають рекрутери працівників з різними рисами особистості для різних завдань? Дискретний експеримент з вибором, Економіка праці, том 78. 2022.

14. Чаморро-Премюзік, Франкевич. 6 причин, чому вищу освіту потрібно реорганізувати. Harvard Business Review. 2019. URL: <https://danielschristian.com/learning-ecosystems/2019/11/29/6-reasons-why-higher-education-needs-to-be-disrupted-chamorro-premuzic-frankiewicz/> (дата звернення: 14.10.2025)

15. Глобальні тенденції розвитку людського капіталу – Лідерство в соціальному підприємстві: переосмислення з акцентом на людину. Deloitte Insights. 2019. URL:
https://www2.deloitte.com/content/dam/insights/us/articles/5136_HC-Trends-2019/DI_HC-Trends-2019.pdf (дата звернення: 14.10.2025)
16. Д. Парк. Самооцінка працівниками вищої освіти відповідності людини організації: роль умов праці та задоволеності роботою, Міжнародний огляд державного управління, том 23, 55-77. 2018. URL:
<https://doi.org/10.1080/12294659.2018.1447883> (дата звернення: 14.10.2025)
17. Т. Хархаліс. Дослідження співпраці ІТ-підприємств з ВНЗ у підготовці кадрів. Management and Entrepreneurship in Ukraine: the stages of formation and problems of development. Випуск 7. Номер 2. 2025. URL:
https://science.lpnu.ua/uk/smeu/vsi-vypusky/vypusk-7-nomer-2-2025/doslidzhennya-spi-vpraci-it-pidpryemstv-z-vnz-u-pidgotovci?utm_source (дата звернення: 14.10.2025)
18. В. І. Юскович-Жуковська, О. М. Богут. WEB-Програмування. 2023, Волинські Обереги, ISBN 978-617-8260-52-1
19. Аналітичні матеріали BCG. 2025 URL:
<https://www.bcg.com/publications/2025/ai-changing-recruitment> (дата звернення: 14.10.2025)
20. Йенс Баєр, Орсоля Ковач-Ондрейкович, П'єр Антебі, Боян Дівчич, Кармен Маркес Кастро, Катерина Мала, Вінчіан Бошен, Генріке Барт, Ніхаріка Джаджорія. Звіт BCG. Стратегія персоналу. 2023. URL:
<https://www.bcg.com/publications/2023/recruitment-recommendations-for-employers> (дата звернення: 14.10.2025)
21. П. Іваницький. Проектування інтелектуальної системи відбору персоналу для працедавців. КНУ імені Вадима Гетьмана. 2025. URL:
<https://ir.kneu.edu.ua/items/b9ed4a82-4ec4-451b-80da-2c0001d8cc5a> (дата звернення: 14.10.2025)
22. В. Черкасова, Б. Бочаров. Імплементація системи рекомендацій на основі nlp у вакансійному аналізі. Information technology: Computer Science,

Software Engineering and Cyber Security. №2. 2024. URL:

<https://journals.politehnica.dp.ua/index.php/it/article/view/598> (дата звернення: 16.10.2025)

23. Frank P.-W. Lo, Jianing Qiu, Zeyu Wang, Haibao Yu, Yeming Chen, Gao Zhang, Benny Lo. AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening. 2025. URL: <https://arxiv.org/pdf/2504.02870> (дата звернення: 16.10.2025)

24. Kenza Khelkhal, Dihia Lanasri. Smart-Hiring: An Explainable end-to-end Pipeline for CV Information Extraction and Job Matching. 2025. URL: <https://arxiv.org/pdf/2511.02537> (дата звернення: 16.10.2025)

25. І. В. Кравченко. Інформаційна система обліку вакансій та кандидатів з персоналізованим AI-помічником з вбудованим NLP-модулем. Репозитарій ХНУ. 2025. URL: <https://openarchive.nure.ua/entities/publication/04daabb7-0da0-4946-8fbc-442851d301cb> (дата звернення: 16.10.2025)

26. Kenza Khelkhal, Dihia Lanasri. Smart-Hiring: An Explainable end-to-end Pipeline for CV Information Extraction and Job Matching. 2025. URL: <https://arxiv.org/pdf/2511.02537> (дата звернення: 16.10.2025)

27. А. Міхель. Від will be a plus до must-have: як AI-інструментарій змінює рекрутинг у геймдеві. DOU. 2025. URL: <https://gamedev.dou.ua/articles/why-does-hr-use-ai/> (дата звернення: 16.10.2025)

28. А. В. Тівецька, С. М. Невмержицька. Удосконалення системи управління персоналом організації з врахуванням вимог міжнародних стандартів iso. Київський Національний Університет технологій та дизайну. ISSN 2413-0117, Сучасні тенденції та проблеми, вісник КНУТД №6 (93), 2015

29. Peters, Tom. The Brand You 50: Fifty Ways to Transform Yourself from an 'Employee' into a Brand That Shouts Distinction, Commitment, and Passion!, KNOPE US, 1999, reprint 2021

30. Статистика IT-ринку України в 2023 році. DOU. URL: <https://dou.ua/lenta/articles/it-newcomers-2023/> (дата звернення: 16.10.2025)

31. Locke, Edwin A. *Handbook of Principles of Organizational Behavior: Indispensable Knowledge for Evidence-Based Management*, Wiley, 2009
32. Tulgan, Bruce. *It's Okay to Be the Boss: The Step-by-Step Guide to Becoming the Manager Your Employees Need*, Collins, 2007
33. Harter, Jim. *Wellbeing: The Five Essential Elements*, Brilliance Audio, 2016
34. Kaplan, Robert S., David P. Norton. *The Balanced Scorecard: Translating Strategy into Action*, Harvard Business Review Press, 1996, reprint 2022
35. Harvard Business Review. *Performance Management: Measure and Improve The Effectiveness of Your Employees*. Harvard Business Review Press, 2006
36. Bersin, Josh. *The Blended Learning Book: Best Practices, Proven Methodologies, and Lessons Learned.*, Pfeiffer, 2004, reprint 2021
37. Kouzes, James M., and Barry Z. Posner. *The Leadership Challenge: How to Make Extraordinary Things Happen in Organizations*, Jossey-Bass, 2017
38. Kirkpatrick, Donald L., and James D. Kirkpatrick. *Evaluating Training Programs: The Four Levels*, Berrett-Koehler Publishers, 2006
39. Noe, Raymond A. *Employee Training and Development*, McGraw-Hill Education, 2019
40. Aguinis, Herman. *Performance Management*, Pearson, 2012
41. Lawler III, Edward E. *Talent: Making People Your Competitive Advantage*, Jossey-Bass, 2008
42. Phoenix, Gene Kim, Kevin Behr, and George Spafford. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*, IT Revolution Press, 2013
43. Sullivan, John. *Talent Magnetism: How to Build a Workplace That Attracts and Keeps the Best*, Quercus, 2013
44. Hatvany, Nina D. *Inside the Technical Consulting Business: Launching and Building Your Independent Practice*, Wiley, 1997
45. DeMarco, Tom and Timothy Lister. *Peopleware: Productive Projects and Teams*, Dorset House, 1999

46. Pfeffer, Jeffrey. *Dying for a Paycheck: How Modern Management Harms Employee Health and Company Performance and What We Can Do About It*, HarperBusiness, 2018
47. Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Currency, 2011
48. Schwaber, Ken and Jeff Sutherland. *Scrum: The Art of Doing Twice the Work in Half the Time*, Currency, 2014
49. Pink, Daniel H. *Drive: The Surprising Truth About What Motivates Us*, Riverhead Books, 2011
50. Weinberg, Gerald M. *Becoming a Technical Leader: An Organic Problem-Solving Approach*, Dorset House Publishing, 1986
51. Laloux, Frederic. *Reinventing Organizations*, Nelson Parker, 2014
52. Rock, David. *Your Brain at Work: Strategies for Overcoming Distraction, Regaining Focus, and Working Smarter All Day Long*, Harper Business, 2009
53. Stanley, Andy. *Next Generation Leader: 5 Essentials for Those Who Will Shape the Future*, Multnomah, 2006
54. Collins, Jim. *Good to Great: Why Some Companies Make the Leap... and Others Don't*, HarperBusiness, 2001, reprint 2019
55. Frederick Winslow Taylor. *The Principles of Scientific Management*, Dover Publications, 1997
56. Elton Mayo. *The Social Problems of an Industrial Civilisation: With an Appendix on the Political Problem*, Routledge, 2007
57. Abraham H. Maslow. *A Theory of Human Motivation*, Martino Fine Books, 2013
58. Frederick Herzberg. *The Motivation to Work*, Routledge, 1993
59. Друкер Питер. *Задачи менеджмента в XXI веке*, Williams, 2000
60. Kent Beck, Cynthia Andres. *Extreme Programming Explained: Embrace Change*, 2nd Edition, Addison-Wesley, 2004
61. Gerald M. Weinberg. *The Psychology of Computer Programming: Silver Anniversary Edition*, Dorset House, 1998

62. Martin Fowler. Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional, 2018
63. Maxwell Maltz. Psycho-Cybernetics, TarcherPerigee, 2015
64. Daniel Goleman. Emotional Intelligence: Why It Can Matter More Than IQ, Random House Publishing Group, 2005
65. Peter F. Drucker. The Practice of Management, Harper Business, 2006
66. Douglas McGregor. The Human Side of Enterprise, McGraw Hill, 2006
67. Michael Armstrong, Stephen Taylor. Armstrong's Handbook of Human Resource Management, Kogan Page Ltd, 2014
68. Бесіди про управління / В. М. Глушков, Г. М. Добров, В. І. Терещенко. К. : Політвидав України, 1973
69. В. М. Глушков. Основы безбумажной информатики. 2-е изд., испр. М. : Наука, 1987
70. Гриценко В.І. Інформаційно-комунікаційні технології в освіті для всіх - в ракурсі проблем суспільства знань. – К.: Академперіодика, 2007. – 28 с.
71. Гриценко В.І. Перспективні технології навчання - основа стратегії побудова суспільства знань. УСiМ. – 2005. – № 6. – С. 5–10.
72. Гриценко В.І. Фундаментальні проблеми Е-навчання. – К.: Академперіодика, 2008. – 38 с.
73. Манако А.Ф., Синиця Е.М. Інформаційне суспільство та ІКТ в освіті. – Lambert Acad. Publ. ISBN 978-3-659-59949-1, 2014. – 99 с.
74. Довгялло А.М. Діалог користувача та ЕОМ: Основи проектування та реалізації. – Київ: Наук. думка, 1981. – 231 с.
75. В.І. Гриценко, С.П. Кудрявцева, В.В. Колос та ін. Дистанційна освіта: теорія та практика. Київ: Наук. думка, 2004. – 376 с.
76. Комп'ютерна технологія навчання: Словник-довідник. Під ред. Гриценко В.І., Довгялло А.М. В 2-х томах. – Київ: Наук. думка, 1992. – 784 с.
77. Von Brevern H., Synytsya K. A systemic activity based approach for holistic learning & training systems. Educational Technology & Society. – 2006. – 9, № 3. – P. 100–111.

78. Манако А.Ф., Синиця Е.М. Неперервна освіта та іноваційні електронні освітньо-наукові простори. / Нові інформаційні технології в освіті для всіх: безперервна освіта / В.І. Гриценко, В.Б. Артеменко та ін. – К.: Академперіодика, 2013. – С. 121–205.
79. Манако А.Ф., Синиця Е.М. ІКТ в освіті: погляд крізь призму трансформацій // Освітні технології та суспільство. – 2012. – 15, № 3. – С. 392–414.
80. Гриценко В.І., Манако А.Ф. Використання навчального мультимедіа в електронних підручниках та дистанційних курсах що надаються через мережу Інтернет: Метод. посібник. – Київ: Вітус, 2003. – 123 с.
81. Колот А. В. Економіка праці та соціально-трудоі відносини: Підручник, Київ, КНЕУ, 2009.
82. Васильєва О. І. Управління персоналом у публічній службі : навч. посіб. для здобувачів другого (магістерського) рівня вищої освіти за спеціальністю 281 «Публічне управління та адміністрування» ; за заг. ред. С. К. Хаджирадєвої. Київ: НАДУ, 2020. 196 с.
83. Нотація проектування системної архітектури C4. URL: <https://c4model.com/> (дата звернення: 21.10.2025)
84. Система LangGraph . URL: <https://LangGraph.net/> (дата звернення: 21.10.2025)
85. Звіт ІТ-асоціації України за 2022 рік. URL: <https://itukraine.org.ua/files/reports/2022/DoITLikeUkraine2022.pdf> (дата звернення: 21.10.2025)
86. Аналітика Львівського ІТ-кластеру про обсяг ІТ-експорту. URL: <https://itcluster.lviv.ua/obsyag-it-eksportu-u-sichni-vrav-majzhe-na-30/> (дата звернення: 21.10.2025)
87. Аналітика IDC про перспективи зростання глобального ІТ-ринку. URL: <https://www.idc.com/getdoc.jsp?containerId=prUS50573223> (дата звернення: 21.10.2025)

88. Аналітика причин звільнень та розвитку ІТ галузі. URL:
<https://economictimes.indiatimes.com/tech/technology/big-tech-is-firing-employees-by-the-thousands-why-and-how-worried-should-we-be/articleshow/97339177.cms?from=mdr> (дата звернення: 21.10.2025)

89. Аналітика масових звільнень в ІТ-галузі. URL:
<https://www.indiatoday.in/technology/features/story/why-are-tech-companies-firing-thousands-of-engineers-it-is-not-about-saving-money-honey-2375516-2023-05-06> (дата звернення: 21.10.2025)

90. Сучасні тренди ІТ-галузі. URL:
<https://www.forbes.com/sites/bernardmarr/2022/11/21/the-top-10-tech-trends-in-2023-everyone-must-be-ready-for/?sh=b43a1b67df0f> (дата звернення: 21.10.2025)

91. Аналітика ринку праці Dou в 2023 році. URL:
<https://dou.ua/lenta/articles/it-job-market-april-2023/> (дата звернення: 25.10.2025)

92. Аналітика ринку праці від Національного інституту стратегічних досліджень. URL:
<https://niss.gov.ua/news/komentari-ekspertiv/rynok-pratsi-it-sektoru-v-umovakh-viyny-realnyi-ta-perpektyvu> (дата звернення: 25.10.2025)

93. Стандарт PSR. Офіційна специфікація. URL:
<https://www.php-fig.org/psr/> (дата звернення: 25.10.2025)

94. Стандарт W3C. Офіційна специфікація. URL:
<https://www.w3.org/standards/about/> (дата звернення: 25.10.2025)

95. Сертифікаційна програма Acquia. URL:
<https://docs.acquia.com/certification/> (дата звернення: 25.10.2025)

96. Офіційна документація з підготовки для сертифікації Acquia. URL:
<https://docs.acquia.com/certification/study-guides/d10-backend-specialist/>

97. Специфікація сертифікації Oracle. URL:
http://www.oraclecertificationprep.com/apex/f?p=OCPSG:EXAM_DETAILS::::::P2_EXAM:1Z0-882 (дата звернення: 25.10.2025)

98. Специфікація сертифікації ICAgile. URL: <https://www.icagile.com/media/pdfs/learning-outcomes/agile-fundamentals-learning-outcomes.pdf> (дата звернення: 25.10.2025)
99. Phoenix, J., Taylor, M. Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs. 2024. O'Reilly Media. URL: <https://www.oreilly.com/library/view/prompt-engineering-for/9781098153427/> (дата звернення: 28.10.2025)
100. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., Chadha, A. A systematic survey of prompt engineering in large language models: Techniques and applications. 2024. URL: <https://arxiv.org/abs/2402.07927> (дата звернення: 28.10.2025)
101. Gu, J., Han, Z., Chen, S., et al. A systematic survey of prompt engineering on vision-language foundation models. 2024. URL: <https://arxiv.org/abs/2307.12980> (дата звернення: 28.10.2025)
102. White, J., Fu, Q., Hays, S., et al. A prompt pattern catalog to enhance prompt engineering with ChatGPT. 2023. URL: <https://arxiv.org/abs/2302.11382> (дата звернення: 28.10.2025)
103. Prompt Engineering Guide. 2024. URL: <https://www.promptingguide.ai>
104. DeepLearning.AI OpenAI. ChatGPT Prompt Engineering for Developers (short course). 2023. URL: <https://www.deeplearning.ai/short-courses/> (дата звернення: 02.11.2025)
105. OpenAI. Prompt engineering examples — OpenAI Cookbook. 2024 URL: <https://cookbook.openai.com/examples> (дата звернення: 02.11.2025)
106. OpenAI. Best practices for prompt engineering with the OpenAI API. 2024 URL: <https://platform.openai.com/docs/guides/prompt-engineering> (дата звернення: 02.11.2025)
107. Anthropic. Prompting with Claude: Best practices. 2024 URL: <https://docs.anthropic.com/claude/docs/prompting> (дата звернення: 02.11.2025)
108. Google Cloud. Generative AI prompt engineering guide. 2024. URL: <https://cloud.google.com/architecture/prompts> (дата звернення: 02.11.2025)

109. Microsoft. Prompt engineering techniques for Azure OpenAI Service. 2023
URL: <https://learn.microsoft.com/azure/ai-services/openai/how-to/prompt-engineering>
(дата звернення: 02.11.2025)
110. Chase, H. LangChain documentation: Prompt templates, prompt management. 2023. URL:
https://python.langchain.com/docs/concepts/prompt_templates (дата звернення: 02.11.2025)
111. LlamaIndex. LlamaIndex core concepts: Prompts query engines. 2023
URL: <https://docs.llamaindex.ai/en/stable/> (дата звернення: 02.11.2025)
112. Debnath, T., et al. A comprehensive survey of prompt engineering techniques in large language models. TechRxiv preprint. 2023. URL:
<https://www.techrxiv.org/> (дата звернення: 05.11.2025)
113. Vatsal, S., Dubey, H. A survey of prompt engineering methods in large language models for different NLP tasks. 2024. URL: <https://arxiv.org/abs/2407.12994>
(дата звернення: 05.11.2025)
114. Qin, C., Zhang, X., et al. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. 2023. URL: <https://arxiv.org/abs/2307.16789> (дата звернення: 05.11.2025)
115. Mialon, G., et al. Augmented language models: A survey. 2023. URL:
<https://arxiv.org/abs/2302.07842> (дата звернення: 05.11.2025)
116. Gao, L., et al. Retrieval-augmented generation for large language models: A survey. 2023. URL: <https://arxiv.org/abs/2312.10997> (дата звернення: 05.11.2025)
117. Prompting Techniques for Enhancing the Use of Large Language Models. Cybersecurity: Education, Science, Technique, 6(2). 2024. URL
<https://science.lpnu.ua/csn/all-volumes-and-issues/volume-6-number-2-2024/prompting-techniques-enhancing-use-large-language> (дата звернення: 05.11.2025)
118. Prompt Engineering for Developers OpenAI Community Thread. 2023.
URL:
<https://community.openai.com/t/chatgpt-prompt-engineering-for-developers/191028>
(дата звернення: 05.11.2025)

119. Liu, J., et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in NLP. *ACM Computing Surveys*, 55(9). 2023. URL: <https://doi.org/10.1145/3560815> (дата звернення: 05.11.2025)
120. Liu, P., et al. GPT understands, too: A survey on prompt-based learning in NLP. 2021. URL: <https://arxiv.org/abs/2107.13586> (дата звернення: 05.11.2025)
121. Schick, T., Schütze, H. Exploiting cloze questions for few-shot text classification and natural language inference. *EACL 2021*. URL: <https://aclanthology.org/2021.eacl-main.20/> (дата звернення: 05.11.2025)
122. Gao, T., Fisch, A., Chen, D. Making pre-trained language models better few-shot learners. *ACL 2021*. URL: <https://aclanthology.org/2021.acl-long.295/> (дата звернення: 08.11.2025)
123. Min, S., et al. Rethinking the role of demonstrations: What makes in-context learning work? *EMNLP 2022*. URL: <https://aclanthology.org/2022.emnlp-main.89/> (дата звернення: 08.11.2025)
124. Dong, Q., et al. A survey on in-context learning. 2023. URL: <https://arxiv.org/abs/2301.00234> (дата звернення: 08.11.2025)
125. Mishra, S., et al. Reframing instruction learning as a question-answering problem. *ACL 2022*. URL: <https://aclanthology.org/2022.acl-long.229/> (дата звернення: 08.11.2025)
126. Sanh, V., et al. Multitask prompted training enables zero-shot task generalization. *ICLR 2022*. URL: <https://openreview.net/forum?id=9Vrb9D0WI4> (дата звернення: 08.11.2025)
127. Wei, J., et al. Finetuned language models are zero-shot learners. 2022. URL: <https://arxiv.org/abs/2109.01652> (дата звернення: 08.11.2025)
128. Chung, H. W., et al. Scaling instruction-finetuned language models. 2022. URL: <https://arxiv.org/abs/2210.11416> (дата звернення: 08.11.2025)
129. Ouyang, L., et al. Training language models to follow instructions with human feedback. *NeurIPS 2022*. URL: <https://arxiv.org/abs/2203.02155> (дата звернення: 08.11.2025)

130. Honovich, O., et al. Instruction induction: From few examples to natural language task descriptions. 2022. URL: <https://arxiv.org/abs/2205.10782> (дата звернення: 08.11.2025)
131. Wang, D., et al. Self-instruct: Aligning language models with self-generated instructions. ACL 2023. URL: <https://arxiv.org/abs/2212.10560> (дата звернення: 08.11.2025)
132. Lester, B., Al-Rfou, R., Constant, N. The power of scale for parameter efficient prompt tuning. EMNLP 2021. URL: <https://aclanthology.org/2021.emnlp-main.243/> (дата звернення: 08.11.2025)
133. Li, X. L., Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. ACL 2021. URL: <https://aclanthology.org/2021.acl-long.353/> (дата звернення: 10.11.2025)
134. Liu, X., et al. P-Tuning v2: Prompt tuning can be comparable to fine-tuning across scales and tasks. ACL 2022 Findings. URL: <https://aclanthology.org/2022.findings-acl.214/> (дата звернення: 10.11.2025)
135. Qin, G., Eisner, J. Learning how to ask: Querying LMs with mixtures of soft prompts. NAACL 2021. URL: <https://aclanthology.org/2021.naacl-main.410/> (дата звернення: 10.11.2025)
136. Liu, N. F., et al. What makes good in-context examples for GPT-3? DEEM 2022. URL: <https://dl.acm.org/doi/10.1145/3533028.3533313> (дата звернення: 10.11.2025)
137. Rubin, O., Herzig, J., Berant, J. Learning to retrieve prompts for in-context learning. NAACL 2022. URL: <https://aclanthology.org/2022.naacl-main.248/> (дата звернення: 10.11.2025)
138. Reynolds, L., McDonell, K. Prompt programming for large language models: Beyond the few-shot paradigm. 2021. URL: <https://arxiv.org/abs/2102.07350> (дата звернення: 10.11.2025)
139. Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. NeurIPS 2022. URL: <https://arxiv.org/abs/2201.11903> (дата звернення: 10.11.2025)

140. Kojima, T., et al. Large language models are zero-shot reasoners. NeurIPS 2022. URL: <https://arxiv.org/abs/2205.11916> (дата звернення: 10.11.2025)
141. Wang, X., et al. Self-consistency improves chain-of-thought reasoning in language models. ICLR 2023. URL: <https://arxiv.org/abs/2203.11171> (дата звернення: 15.11.2025)
142. Yao, S., et al. Tree of thoughts: Deliberate problem solving with large language models. 2023. URL: <https://arxiv.org/abs/2305.10601> (дата звернення: 15.11.2025)
143. Wang, X., et al. Plan-and-solve prompting: Improving chain-of-thought reasoning via planning. 2023. URL: <https://arxiv.org/abs/2305.04091> (дата звернення: 15.11.2025)
144. Yao, S., et al. ReAct: Synergizing reasoning and acting in language models. ICLR 2023. URL: <https://arxiv.org/abs/2210.03629> (дата звернення: 15.11.2025)
145. Schick, T., et al. Toolformer: Language models can teach themselves to use tools. 2023. URL: <https://arxiv.org/abs/2302.04761> (дата звернення: 17.11.2025)
146. Shen, Y., et al. HuggingGPT: Solving AI tasks with ChatGPT and its friends in HuggingFace. 2023. URL: <https://arxiv.org/abs/2303.17580> (дата звернення: 17.11.2025)
147. Gao, L., et al. PAL: Program-aided language models. ICLR 2023. URL: <https://arxiv.org/abs/2211.10435> (дата звернення: 17.11.2025)
148. Chen, J., et al. Teaching large language models to self-debug. 2023. URL: <https://arxiv.org/abs/2304.05128> (дата звернення: 17.11.2025)
149. Huang, W., et al. Language models can self-improve. 2023. URL: <https://arxiv.org/abs/2210.11610> (дата звернення: 17.11.2025)
150. Nye, M., et al. Show your work: Scratchpads for intermediate computation with language models. 2021. URL: <https://arxiv.org/abs/2112.00114> (дата звернення: 21.11.2025)
151. Zelikman, E., et al. STaR: Bootstrapping reasoning with reasoning. NeurIPS 2022. URL: <https://arxiv.org/abs/2203.14465> (дата звернення: 21.11.2025)

152. OpenAI. Function calling and other API features. 2023. URL: <https://platform.openai.com/docs/guides/function-calling> (дата звернення: 21.11.2025)
153. OpenAI. Structured outputs with JSON Schema. 2024. URL: <https://platform.openai.com/docs/guides/structured-outputs> (дата звернення: 21.11.2025)
154. Anthropic. Using tools and function calling in Claude. 2024. URL: <https://docs.anthropic.com/claude/docs/tool-use> (дата звернення: 21.11.2025)
155. OpenAI. Building agents with the Assistants API. 2024. URL: <https://platform.openai.com/docs/assistants/overview> (дата звернення: 21.11.2025)
156. Microsoft. Agent patterns with Azure OpenAI and orchestration frameworks. 2024. URL: <https://learn.microsoft.com/azure/ai-services/openai/> (дата звернення: 25.11.2025)
157. Karpas, E., et al. MRKL systems: Modular reasoning, knowledge and language. AI21 Labs Blog / arXiv. 2024. URL: <https://arxiv.org/abs/2205.00445> (дата звернення: 25.11.2025)
158. Schuurmans, D., et al. A process for systematically eliciting reliable reasoning from language models. 2023. URL: <https://arxiv.org/abs/2312.08360> (дата звернення: 25.11.2025)
159. Lewis, P., et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. NeurIPS 2020. URL: <https://arxiv.org/abs/2005.11401> (дата звернення: 25.11.2025)
160. Guu, K., et al. REALM: Retrieval-augmented language model pre-training. ICML 2020. URL: <https://arxiv.org/abs/2002.08909> (дата звернення: 25.11.2025)
161. Karpukhin, V., et al. Dense passage retrieval for open-domain question answering. EMNLP 2020. URL: <https://arxiv.org/abs/2004.04906> (дата звернення: 25.11.2025)
162. Izacard, G., Grave, E. Leveraging passage retrieval with generative models for open-domain QA. EACL 2021. URL: <https://aclanthology.org/2021.eacl-main.74> (дата звернення: 27.11.2025)

163. Gao, L., et al. Retrieval-augmented generation for large language models: A survey. 2023. URL: <https://arxiv.org/abs/2312.10997> (дата звернення: 27.11.2025)
164. OpenAI. RAG patterns — OpenAI Cookbook. 2024. URL: https://cookbook.openai.com/examples/retrieval_augmentation (дата звернення: 27.11.2025)
165. Pinecone. Vector Databases: A Comprehensive Introduction. 2023. URL: <https://www.pinecone.io/learn/vector-database/> (дата звернення: 27.11.2025)
166. Chroma. Chroma DB Documentation. 2023. URL: <https://docs.trychroma.com/> (дата звернення: 27.11.2025)
167. LangChain. Retrieval and RAG documentation. 2023. URL: https://python.langchain.com/docs/use_cases/question_answering/ (дата звернення: 29.11.2025)
168. LlamaIndex. RAG architectures and query engines. 2023. URL: https://docs.llamaindex.ai/en/stable/module_guides/querying/ (дата звернення: 29.11.2025)
169. Hendrycks, D., et al. Measuring massive multitask language understanding. ICLR 2021 (MMLU). URL: <https://arxiv.org/abs/2009.03300> (дата звернення: 29.11.2025)
170. Srivastava, A., et al. Beyond the imitation game benchmark: Measuring and extrapolating the capabilities of language models. 2022. URL: <https://arxiv.org/abs/2206.04615> (дата звернення: 29.11.2025)
171. Lin, S. C., et al. TruthfulQA: Measuring how models mimic human falsehoods. ACL 2022. URL: <https://arxiv.org/abs/2109.07958> (дата звернення: 29.11.2025)
172. Ribeiro, M. T., et al. Beyond accuracy: Behavioral testing of NLP models with CheckList. ACL 2020. URL: <https://aclanthology.org/2020.acl-main.442/> (дата звернення: 29.11.2025)
173. Carlini, N., et al. Extracting training data from large language models. USENIX Security 2021. URL: <https://arxiv.org/abs/2012.07805> (дата звернення: 29.11.2025)

174. Ganguli, D., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. 2022. URL: <https://arxiv.org/abs/2209.07858> (дата звернення: 29.11.2025)
175. NIST. AI Risk Management Framework (AI RMF 1.0). 2023. URL: <https://www.nist.gov/itl/ai-risk-management-framework> (дата звернення: 03.12.2025)
176. OWASP Foundation. OWASP Top 10 for Large Language Model Applications. 2023. URL: <https://owasp.org/www-project-top-10-for-large-language-model-applications/> (дата звернення: 03.12.2025)
177. Solaiman, I., Dennison, C. Process for adapting language models to society. 2021. URL: <https://arxiv.org/abs/2106.10328> (дата звернення: 03.12.2025)
178. Weidinger, L., et al. Ethical and societal risks of language models. 2021. URL: <https://arxiv.org/abs/2112.04359> (дата звернення: 03.12.2025)
179. Bender, E. M., Gebru, T., McMillan-Major, A., Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? FAccT 2021. URL: <https://dl.acm.org/doi/10.1145/3442188.3445922> (дата звернення: 03.12.2025)
180. Floridi, L., Chiriatti, M. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*. 30(4). 2020. P 681–694. URL: <https://doi.org/10.1007/s11023-020-09548-1> (дата звернення: 05.12.2025)
181. Kasneci, E., et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*. 2023. P. 103. URL: <https://doi.org/10.1016/j.lindif.2023.102274> (дата звернення: 05.12.2025)
182. Sun, T., et al. Safety-prompts: Language model safety via adversarial prompting. 2023. URL: <https://arxiv.org/abs/2309.02705> (дата звернення: 05.12.2025)
183. OpenAI. GPT-4 system card. 2023. URL: <https://cdn.openai.com/papers/gpt-4-system-card.pdf> (дата звернення: 05.12.2025)
184. Vaswani, A., et al. Attention is all you need. *NeurIPS 2017*. URL: <https://arxiv.org/abs/1706.03762> (дата звернення: 05.12.2025)

185. Radford, A., et al. Improving language understanding by generative pre-training. OpenAI Technical Report. 2018. URL: <https://openai.com/research/language-unsupervised> (дата звернення: 05.12.2025)
186. Radford, A., et al. Language models are unsupervised multitask learners. OpenAI Technical Report. 2019. URL: <https://openai.com/research/language-unsupervised> (дата звернення: 05.12.2025)
187. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL 2019. URL: <https://arxiv.org/abs/1810.04805> (дата звернення: 06.12.2025)
188. Raffel, C., et al. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR. 21(140). 2020. P. 1–67. URL: <https://jmlr.org/papers/v21/20-074.html> (дата звернення: 06.12.2025)
189. Brown, T. B., et al. Language models are few-shot learners. NeurIPS 2020. URL: <https://arxiv.org/abs/2005.14165> (дата звернення: 06.12.2025)
190. OpenAI. GPT-3 model card API documentation. 2020. URL: <https://platform.openai.com/docs/models/gpt-3> (дата звернення: 06.12.2025)
191. OpenAI. GPT-4 technical report. 2023. URL: <https://arxiv.org/abs/2303.08774> (дата звернення: 06.12.2025)
192. Bubeck, S., et al. Sparks of artificial general intelligence: Early experiments with GPT-4. 2023. URL: <https://arxiv.org/abs/2303.12712> (дата звернення: 06.12.2025)
193. Touvron, H., et al. LLaMA: Open and efficient foundation language models. 2023. URL: <https://arxiv.org/abs/2302.13971> (дата звернення: 09.12.2025)
194. Touvron, H., et al. LLaMA 2: Open foundation and fine-tuned chat models. 2023. URL: <https://arxiv.org/abs/2307.09288> (дата звернення: 09.12.2025)
195. Meta AI. Llama 3 model card and technical report. 2024. URL: <https://ai.meta.com/blog/meta-llama-3/> (дата звернення: 09.12.2025)
196. OpenAI. GPT-4o and GPT-4o mini model cards. 2024 URL:
197. <https://platform.openai.com/docs/models#gpt-4o> (дата звернення: 09.12.2025)

198. Anthropic. Claude 2 and Claude 3 model cards. 2023. URL: <https://www.anthropic.com/product> (дата звернення: 09.12.2025)
199. Zhao, W. X., et al. A survey of large language models. 2023. URL: <https://arxiv.org/abs/2303.18223> (дата звернення: 11.12.2025)
200. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. MIT Press. 2016. URL: <https://www.deeplearningbook.org/> (дата звернення: 11.12.2025)
201. Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2005. URL: <https://link.springer.com/book/9780387310732> (дата звернення: 11.12.2025)
202. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press. 2012. URL: https://books.google.com/books/about/Machine_Learning.html?id=NZP6AQAA (дата звернення: 11.12.2025)
203. Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer. 2009. URL: <https://link.springer.com/book/10.1007/978-0-387-84858-7> (дата звернення: 11.12.2025)
204. Russell, S. J., Norvig, P. Artificial Intelligence: A Modern Approach (4th ed.). Pearson. 2021. URL: <http://aima.cs.berkeley.edu/> (дата звернення: 11.12.2025)
205. Sutton, R. S., Barto, A. G. Reinforcement Learning: An Introduction (2nd ed.). MIT Press. 2018. URL: <https://incompleteideas.net/book/the-book-2nd.html> (дата звернення: 15.12.2025)
206. Mitchell, T. M. Machine Learning. McGraw-Hill. 1997. URL: <https://www.cs.cmu.edu/~tom/mlbook.html> (дата звернення: 15.12.2025)
207. Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification (2nd ed.). Wiley. 2000. URL: <https://www.wiley.com/en-us/Pattern+Classification%2C+2nd+Edition-p-9780471056690> (дата звернення: 15.12.2025)
208. Shalev-Shwartz, S., Ben-David, S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press. 2014. URL:

<https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/> (дата звернення: 15.12.2025)

209. Alpaydin, E. Introduction to Machine Learning (4th ed.). MIT Press. 2020. URL: <https://mitpress.mit.edu/9780262043793/introduction-to-machine-learning/> (дата звернення: 15.12.2025)

210. Abu-Mostafa, Y. S., Magdon-Ismael, M., Lin, H.-T. Learning from Data: A Short Course. AMLBook. 2012. URL: <https://learningfromdata.com/> (дата звернення: 17.12.2025)

211. MacKay, D. J. C. Information Theory, Inference, and Learning Algorithms. Cambridge University Press. 2003. URL: <https://www.inference.org.uk/itprnn/book.pdf> (дата звернення: 17.12.2025)

212. Cover, T. M., Thomas, J. A. Elements of Information Theory (2nd ed.). Wiley. 2006. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X> (дата звернення: 17.12.2025)

213. Jurafsky, D., Martin, J. H. Speech and Language Processing. Pearson (manuscript draft). 2023. URL: <https://web.stanford.edu/~jurafsky/slp3/> (дата звернення: 17.12.2025)

214. Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. Morgan Claypool. URL: <https://www.morganclaypool.com/doi/abs/10.2200/S00762ED1V01Y201703HLT037> (дата звернення: 17.12.2025)

215. Bird, S., Klein, E., Loper, E. Natural Language Processing with Python. O'Reilly Media. 2009. URL: <https://www.nltk.org/book/> (дата звернення: 17.12.2025)

216. Eisenstein, J. Introduction to Natural Language Processing. MIT Press. 2019. URL: <https://mitpress.mit.edu/9780262042840/introduction-to-natural-language-processing/> (дата звернення: 19.12.2025)

217. Géron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media. 2023. URL:

<https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967> (дата звернення: 19.12.2025)

218. Raschka, S., Mirjalili, V. Python Machine Learning (3rd ed.). Packt Publishing. 2019. URL:

<https://www.packtpub.com/product/python-machine-learning-third-edition/9781789955750> (дата звернення: 19.12.2025)

219. Chollet, F. 2021. Deep Learning with Python (2nd ed.). Manning Publications. URL:

<https://www.manning.com/books/deep-learning-with-python-second-edition> (дата звернення: 19.12.2025)

220. Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. Dive into Deep Learning. 2020. URL: <https://d2l.ai/> (дата звернення: 19.12.2025)

221. Nielsen, M. A. Neural Networks and Deep Learning. Determination Press (online book). 2015. URL: <http://neuralnetworksanddeeplearning.com/> (дата звернення: 21.12.2025)

222. Szeliski, R. Computer Vision: Algorithms and Applications (2nd ed.). Springer. 2020. URL: <https://szeliski.org/Book/> (дата звернення: 21.12.2025)

223. Murphy, K. P. Probabilistic Machine Learning: An Introduction. MIT Press. 2022. URL: <https://mitpress.mit.edu/9780262046824/probabilistic-machine-learning/> (дата звернення: 21.12.2025)

224. Mohri, M., Rostamizadeh, A., Talwalkar, A. Foundations of Machine Learning (2nd ed.). MIT Press. 2018. URL: <https://mitpress.mit.edu/9780262039406/foundations-of-machine-learning/> (дата звернення: 21.12.2025)

ДОДАТКИ

ДОДАТОК А.

Стандарти бекенд та фронтенд розробки

Галузеві стандарти, системи сертифікації та загальноприйняті підходи, які в основному використовуються для формування вимог до знань, умінь та навичок фахівців за напрямом бекенд-розробки з використанням фреймворку CMF Drupal.

PSR (PHP Standard Recommendations) [49] вказує на набір рекомендацій та стандартів для написання PHP-коду. Ці стандарти розроблені групою PHP-FIG (PHP Framework Interop Group), яка включає представників найбільш популярних PHP-фреймворків і проєктів.

Основна мета PSR - забезпечити консистентність та чіткість коду, що сприяє його легкому читанню та підтримці. Зокрема, PSR зосереджується на стандартах форматування коду, стандартних інтерфейсах та підходах до автозавантаження.

Виділимо ключові PSR-стандарти:

1. PSR-1: Basic Coding Standard: Цей стандарт визначає базові елементи, які мають використовуватися для забезпечення високої інтероперабельності коду PHP. Він включає в себе вимоги до тегів, кодування символів, назв змінних та інших базових аспектів написання коду.
2. PSR-2: Coding Style Guide: Він базується на PSR-1 та вводить додаткові правила форматування: відступи, розташування фігурних дужок, довжина рядка та інші стилістичні деталі. Ця рекомендація забезпечує єдність стилю коду серед розробників.
3. PSR-3: Logger Interface: Цей стандарт описує загальний інтерфейс для систем реєстрації журналів, дозволяючи розробникам створювати об'єкти, які можуть передаватися між різними системами реєстрації журналів.
4. PSR-4: Autoloading Standard: Замінюючи PSR-0, ця рекомендація визначає, як автозавантаження класів має відбуватися, щоб забезпечити легше взаємодію між різними PHP-кодами.

5. PSR-7: HTTP Message Interface: Описує загальний інтерфейс для репрезентації HTTP-повідомлень, які є ключовими для веб-додатків, що працюють на PHP.
6. PSR-11: Container Interface: Описує інтерфейс контейнера, що використовується для отримання об'єктів та їх залежностей.
7. PSR-12: Extended Coding Style: Оновлення PSR-2, що включає новіші елементи мови PHP.

Крім того, існує ще ряд інших PSR-рекомендацій, які покривають різні аспекти розробки на PHP. Вони служать як основа для забезпечення якісного, чіткого та консистентного коду серед PHP-спільноти.

W3C (World Wide Web Consortium) [50] є міжнародним співтовариством, яке включає організації-члени, постійний персонал та громадськість, що працює разом над розробкою веб-стандартів. W3C засновано Тімом Бернерсом Лі, винахідником World Wide Web, у 1994 році.

Основна мета W3C полягає в розробці та узгодженні стандартів та протоколів, які забезпечують довгостроковий ріст Інтернету як платформи для інформаційного обміну, комунікації та колективної роботи.

Ось деякі ключові аспекти W3C:

1. Стандарти: W3C розробляє веб-стандарти, які забезпечують єдність та інтероперабельність в мережі. Ці стандарти включають такі, як HTML, CSS, XML, SVG та багато інших.
2. Сумісність: Однією з основних цілей W3C є забезпечення того, щоб всі веб-стандарти були сумісними один з одним. Це гарантує, що веб-сайти та веб-додатки будуть працювати належним чином в усіх браузерах, незалежно від платформи.
3. Доступність: W3C наголошує на необхідності робити веб доступним для всіх користувачів, незалежно від їхніх фізичних або технічних обмежень. Вони розробляють рекомендації та стандарти, щоб забезпечити доступність веб-контенту.

4. Інтернаціоналізація: З метою забезпечення глобальної доступності Інтернету, W3C працює над тим, щоб веб-стандарты підтримували багато мов та локалей.
5. Приватність і безпека: W3C також розробляє стандарти, які захищають конфіденційність користувачів і забезпечують безпеку даних в Інтернеті.
6. Взаємодія: Щоб сприяти розвитку Інтернету, W3C стимулює взаємодію між різними зацікавленими сторонами: організаціями, вченими, промисловістю та громадськістю.

Зрештою, W3C діє як керівний орган, який стежить за тим, щоб Інтернет продовжував розвиватися в напрямку, який забезпечує доступ, сумісність, стабільність та інноваційність.

Скіл-матриця для ролі Drupal Backend Developer.

	Trainee	Junior	Middle	Middle+	Senior
Technical skills					
PHP	<ul style="list-style-type: none"> - Basic syntax - Conditional operators: if, switch - Cycles: for, foreach, while, do-while - functions, passing arguments - data types - arrays - sessions - HTTP and HTTPS protocols 	<ul style="list-style-type: none"> - Cycles: iterations & recursions - Recursive functions - Strings operation - Cookies - Requests: GET, POST, PUT, DELETE - ssh, ftp, sftp protocols 	<ul style="list-style-type: none"> - Able to manage typical data structures - Using includes - Using try-catch - Able to use debug tools - Understanding general PHP settings - Able to use PHP logs - Able to do simple PHP refactoring 	<ul style="list-style-type: none"> - Understanding PHP performance - Able to use profiler - Understanding PHP memory usage - Understanding basic PHP security - Able to refactor deprecations - Knowledge about affective code - Able to do refactoring - Able to do code review 	<ul style="list-style-type: none"> - Using basic design patterns in the practical development - Good knowledge of code optimization - Writing effective code - Writing safe code - Able to do code review of difficult code - Able to do code optimization and using performance metrics
PHP OOP	<ul style="list-style-type: none"> - Understanding classes and objects - Abstraction, Encapsulation, Inheritance, Polymorphism - Class syntax - Class properties - Class methods - Constructors - Destructors 	<ul style="list-style-type: none"> - Autoloadning classes - Visibility properties - Static keyword - Final keyword - Object inheritance - Scope resolution operator - Class abstraction - Object interfaces 	<ul style="list-style-type: none"> - Traits - Anonymous classes - Overloading - Magic methods - Late static bindings - Covariance and contravariance - Native designing classes and methods - Understanding advantages and disadvantages of OOP 	<ul style="list-style-type: none"> - Refactoring of simple OOP code - Understanding of OOP performance - Understanding static calls and using constructors - Understanding basic OOP design patterns 	<ul style="list-style-type: none"> - Strong experience of the OOP refactoring - Experience with practical usage of the OOP design patterns - Native designing the application architecture using OOP

Databases	<ul style="list-style-type: none"> - Understanding relational DB concept - Understanding DB tables - Understanding DB rows and fields - Able to do simple queries: SELECT, INSERT, UPDATE, DELETE, TRUNCATE - Using PHPMyAdmin for the elementary DB management 	<ul style="list-style-type: none"> - Understanding primary and secondary keys - Able to use JOIN operator - Able to use GROUP, ORDER, DISTINCT operators 	<ul style="list-style-type: none"> - Able to use subqueries - Able to use UNION operator - Able to use aggregated data in the query - Basic understanding of the query execution principals - Basic understanding of the query optimization principals - Understanding indexes - Understanding data normalization principles: normal forms N1-N4 and other rules 	<ul style="list-style-type: none"> - Good understanding of the query optimization principals - Good knowledge of indexes usage - Basic understanding of MySQL tuning - Able to build complex queries - Able to optimize DB using indexes - Basic understanding how MySQL works - Basic experience with MySQL console - Basic understanding of the big data processing - Basic knowledge about No-SQL DB 	<ul style="list-style-type: none"> - Deep and com understanding. - MySQL syntax - Able to manag through console - Able to use to DB performance analysis - Able to do the performance optimization for queries and DB structure - Able to use an analyse DB log
Drupal Site Building	<ul style="list-style-type: none"> - Able to install and setup Drupal locally - Able to create content type - Able to create and manage fields - Able to setup entity edit forms - Able to setup entity displays (and create new if necessary) - Able to create and manage taxonomy - Able to create and manage blocks - Able to create and manage simple Views - Able to enable and disable modules - Able to setup simple permissions - Able to setup Pathauto and redirects - Basic understanding of Drupal configurations 	<ul style="list-style-type: none"> - Able to use Paragraphs - Able to build complex content solution based on nested Paragraphs, referenced entities, etc - Able to build complex Views - Able to manage text formats - Able to install and setup contrib modules - Able to search for contrib modules at drupal.org - Strong understanding of the name conventions for fields and entities 	<ul style="list-style-type: none"> - Good understanding of the Drupal UI - Understanding principals of re-using components - Using Paragraphs library - Using Media library - Ability to desing content entities structure based on the userstory - Basic UI/UX knowledge - Knowledge of typical modules for the building functionality using contribs 	<ul style="list-style-type: none"> - Understanding basic UI/UX best practices - Able to provide advices for UI/UX optimization - Able to design difficult content structure - Experience with Workflows - Knowledge and experience with most part of the typical modules and libraries 	<ul style="list-style-type: none"> - Knowledge of UI/UX for optim designing applications - Able to design difficult complex content structure - Able to design application using Workflow / Workbench - Able to build application using sets of modules wide integration between the modules - Able to build application using wide integration between contrib custom modules - Knowledge for building applica using alternative solutions and d comparison of alternative solu based on cost a effectivity

<p>Drupal API & Development</p>	<ul style="list-style-type: none"> - Defining a custom module - Routes - Controllers - Forms - Services - Permissions - Basic hooks - Basic entity operations - Following indentions and basic code standards - Following commenting standards 	<ul style="list-style-type: none"> - Dynamic access rules - Ajax form submit and validation - Using most used hooks - Being able to use dependency injection - Entity query - Being able to create simple plugins (blocks) - Understanding render arrays 	<ul style="list-style-type: none"> - Being able to create plugins: widgets, formatters, views fields, etc - Being able to create Batch - Being able to create Queue - Being able to use 3rd-party APIs in code - Being able to use event listeners and dispatchers - Being able to mixing custom and contrib functionality in code - Being able to use most popular Drupal APIs - Understanding sync / async contexts of code execution 	<ul style="list-style-type: none"> - Knowledge about most used classes - Deep understanding how Dependency Injection works - Being able to override typical functionality - Being able to use class decorators - Being able to define custom hooks - Being able to create custom entity types - General understanding how to process big data - General understanding of the code execution flow in Drupal - General understanding advantages and disadvantages of the custom and contrib solutions 	<ul style="list-style-type: none"> - Good understanding Drupal architecture - Being able to implement any functionality based on contrib or custom modules - Being able to hihg amount of and / or highload - Understanding Symfony works - Being able to own APIs - Being able to integrate Drupal other applications - Being able to integrate Drupal other technologies
<p>Drupal Theming</p>	<ul style="list-style-type: none"> - Understanding theme definition - Understanding templates structure - General understanding Twig - Understanding templates names convention - Being able to edit simple templates 	<ul style="list-style-type: none"> - Understanding render arrays - Able to create templates based on existing ones - Able to use standard templates name suggestions - Able to create standard render arrays - Able to create simple theme functions and put variables to custom templates - Being able to use simple preprocess hooks 	<ul style="list-style-type: none"> - Understanding theme_suggestions_alter() - Being able to define custom theme suggestion - Deep understanding Twig: being able to use conditional operators, cycles, arrays, etc - Understanding translation in Twig - Understanding how to define library - Being able to put CSS to library - Being able to put JS to library - Being able to use preprocess hooks widely 	<ul style="list-style-type: none"> - Understanding weight management for libraries components - Understanding regions theme model - General understanding of the content rendering flow - Being able to integrate existing markup and divide it between different templates - General understanding of the render caching - General understanding of client side caching 	<ul style="list-style-type: none"> - Understanding Drupal render flow - Understanding Render cache flow - Understanding BigPipe and lazy loading - Deep understanding of Drupal Ajax from client side - Being able to create CSS and JS for simple / mid cases - Being able to integrate existing markup and provide improvements according to UI/UX practices

<p>Decoupling / Integration</p>	<p>-</p>	<p>-</p>	<ul style="list-style-type: none"> - General understanding of decoupled architecture - Understanding external services concept - Understanding API concepts: endpoint, service, session, etc - Able to integrate existing services using contrib solutions: MailChimp, LiqPay, etc - Understanding JSON syntax - Understanding XML syntax - Understanding REST principals - Understanding data transfer model - Understanding HTTP protocol, ability to work with different types of requests, working with headers information - Able to use authentication: JWT, OAuth 	<ul style="list-style-type: none"> - Understanding difference between coupled (monolithic) / decoupled / headless architecture - Understanding and experience with REST API - Understanding and experience with JSON:API - Basic understanding GraphQL - Understanding SOAP - Understanding CORS 	<ul style="list-style-type: none"> - Understanding microservice approach - Ability to create custom non-standard solutions using REST - Ability to optimize solutions based REST and JSON - Ability to analyze and improve performance for REST and JSON endpoints - General understanding of highload and big data exchange
<p>SEO</p>	<p>-</p>	<ul style="list-style-type: none"> - General understanding of the HTML document structure, keywords, semantic, and general search optimization rules - Able to setup metatags - Understanding the principles of the relevancy - Understanding structure of sitemap.xml and robots.txt 	<ul style="list-style-type: none"> - Understanding HTML semantics - able to setup metatags based on tokens - able to setup OG metatags - understanding search optimization for desktop and mobile 	<ul style="list-style-type: none"> - Able to setup multilingual sitemap and huge sitemaps - Understanding how to use redirects - Able to use Google analytics and Google page speed insights for simple audit 	<ul style="list-style-type: none"> - Doing initial SEO audit - Collaboration with SEO specialists - Wide usage of Google analytics

System performance	-	<ul style="list-style-type: none"> - Able to set and disable standard caching in Drupal - Able to enable aggregation rules - Able to setup AdvAgg - Understanding general cache concepts - Able to use cache tags, contexts, keys, max-age - General understanding about BigPipe module - General understanding about client-side cache - General understanding about the cache lifetime and cache invalidation 	<ul style="list-style-type: none"> - Able to use Memcache, Redis: installation and basic setup - Able to setup simple integration with CDN: images, css, js - Able to use profiler for simple performance measuring 	<ul style="list-style-type: none"> - Able to set and override cache tags, contexts, keys - Understanding principals of complex usage for cache tags, contexts, keys - Understanding Varnish usage - Understanding principals of reverse proxying - Able to use tools for complex performance analysis - General understanding of the complex performance tuning for Nginx/Apache, PHP, MySQL 	<ul style="list-style-type: none"> - Deep understanding of caching architecture in Drupal - Understanding principals of multilayer caching - Detailed tuning memcache and - Ability to do the complex audit and optimization of performance
Testing	-	<ul style="list-style-type: none"> - being able to understand the concepts of DoD and DoR - being able to write the "how to test" section - being capable to test someones' feature/task 	<ul style="list-style-type: none"> - knowing the concepts of how to write the acceptance criterias - knowing the meaning of a regression test and also smoke testing - being able to run the automated testing for a certain vendor 	<ul style="list-style-type: none"> - knowing at least 1 php testing framework (like phpunit, selenium, behat) - being able to create a custom automated test (at least functional test) - being able to fix an existing automated test 	<ul style="list-style-type: none"> - knowing all the types of the automated tests - being able to create any type of test and understand it well - being able to create a custom gitlab pipeline job for automated testing
Drush	<ul style="list-style-type: none"> - Able to use elementary commands: - drush cr - drush cim - drush cex - drush updb 	<ul style="list-style-type: none"> - pm-enable - pm-uninstall - knows about drush versions - knows about drush aliases 	<ul style="list-style-type: none"> - understanding how to run php with Drush - ability to use specific config using Drush - using combination of commands - ability to write custom commands - understanding difference between drush versions 	<ul style="list-style-type: none"> - understanding execution limitations in Drush - understanding execution user contexts - understanding global and local contexts 	
Git	<ul style="list-style-type: none"> - Understanding GIT basic: - clone repo - select existing remote branch - pull changes - create local branch - add changes 	<ul style="list-style-type: none"> - Understanding branch naming - create branch - merge branch - rebase branch - revert commits - revert merge 	<ul style="list-style-type: none"> - interactive rebase - partial commits - squash commits - cherry-pick - tags 		

	<ul style="list-style-type: none"> - commit - create MR 				
Docker & DevOps	<ul style="list-style-type: none"> - Able to use existing environment based on Docker (DDEV, Docker compose, etc) - General understanding of the contenerization - Able to log in into container, and do simple operations - Able to setup LAMP / MAMP stack natively 	<ul style="list-style-type: none"> - Understanding what is container - Understanding what is image - Able to do small changes in Dockerfile - Able to do small tuning in docker-compose.yml or ddev config.yml 	<ul style="list-style-type: none"> - Able to setup ddev environment from scratch - Able to setup and do a tuning for docker-compose.yml - Able to use multiple containers in the local environment - Able to manage resource limits and quotas 	<ul style="list-style-type: none"> - Understanding how to write bash scripts - Able to setup multi-environments (using docker network) 	<ul style="list-style-type: none"> - Understanding using GitLab and Bitbucket, GitHub - Understanding Docker and basic part of CI pipeline - Understanding principles of scalability and orchestration of Docker containers (based on Kubernetes or other technology) - Understanding what an internal networking is and how to communicate between 2 different docker networks - Understanding portainer tool - Knowing how to expose particular ports using docker
Computer Science & Architecture	-	<ul style="list-style-type: none"> - Understanding general OOP flow - Understanding main data types - Understanding the difference between classes, objects, arrays and able to use them depends on the situation 	<ul style="list-style-type: none"> - Understanding OOP paradigms and could use them for real projects - understanding the basic algorithms (sorting, search) - understanding the basic data structures: tree, list - understanding KISS principle 	<ul style="list-style-type: none"> - Basic understanding of the SOLID principle - Strong understanding OOP - Strong understanding FP - Understanding design patterns - Understanding algorithms complexity 	<ul style="list-style-type: none"> - Able to compare and OOP - Understanding OOD (Object Oriented Design) - Understanding principles of decomposition

<p>Solution providing</p>	<ul style="list-style-type: none"> - Able to apply ready-made solutions, that are already described in details and already approved by seniorish developer - Requires help of and advices from mentor and seniorish developer 	<ul style="list-style-type: none"> - Able to analyze simple technical documentations - Able to integrate simple ready-made solution - Able to create simple solution from scratch with help of seniorish developer 	<ul style="list-style-type: none"> - Able to implement mid difficult solutions by himself - Able to do simple technical decisions - Able to find out mid difficult ready-made solutions - Understanding and trying to find out the simplest way for the solution - Able to read and understand source code - Able to discuss mid difficult technical solutions 	<ul style="list-style-type: none"> - Able to provide difficult technical solutions with help of senior developer - Able to think about technical solution using already decomposed business process - Able to think and provide ideas about most optimal technical solutions - Able to use own experience for providing technical decision at native level 	<ul style="list-style-type: none"> - Able to provide solutions based on own experience and knowledge in development best practices - Able to understand the business processes and propose solutions based on understanding of business needs and scope of separate business processes
<p>Estimates</p>	<p>-</p>	<ul style="list-style-type: none"> - Able to estimate elementary tasks, but require mandatory approval from the seniorish person at project 	<ul style="list-style-type: none"> - Able to make estimates by himself for simple tasks - Estimates for mid difficulty and difficult tasks requires to be approved by senior developer - General understanding of estimation components: development, testing, setup, code review, documentation, QA deploy, Live deploy - General understanding about Agile methodologies and aspects of development related to estimation: sprints, releases, story points, etc 	<ul style="list-style-type: none"> - Being able to estimate mid difficulty tasks, but requires approval for difficult tasks estimation from senior developer - Good understanding of estimation components, and levels of affecting risks - Understanding and basic experience with Agile development: Scrum / Kanban - Being able to use Agile estimation practices: planning poker, etc - Understanding and take part in sprint plannings - Understanding sprint delivery life cycle - Ability to reflect on 	<ul style="list-style-type: none"> - Able to estimate type of tasks - Extra complex tasks requires estimation validation from Tech Lead - Able to analyze and reflect on incorrect estimations by himself - Able to review and validate estimations - Able to do correct estimations on pre-sale phase (requires validation from Tech Lead for difficult complex cases) - Able to follow Agile estimation - Able to provide conclusions related to tasks estimation from technical sprints retros

				"lessons learned" at retros	
Collaboration with Drupal.org	-	<ul style="list-style-type: none"> - Have profile at drupal.org - Knows how to search at drupal.org - Knows how to ask at drupal.org - Knows how to create issue at drupal.org 	<ul style="list-style-type: none"> - Knows how to apply existing patches - Able to re-roll patch - Understanding how to put custom patch to drupal.org 	<ul style="list-style-type: none"> - Helping with resolving issues at drupal.org by providing comments - contributing patches to drupal.org - knows how to create a module in the sandbox 	<ul style="list-style-type: none"> - Able to maintain custom module - be a co-maintainer at drupal.org - Contributing to documentation at drupal.org - Understanding how to configure automated testing a certain module/theme/patch - Being able to prepare a fork for certain issue and provide a MR for review - Being able to create a new module/theme/patch release
Technical mentorship	-	-	- Could be a mentor for trainee	- Could be a mentor for junior	- Could be a mentor for Middle / Middle +

ДОДАТОК В.

Матриця вагових коефіцієнтів для скіл матриці.

	Trainee	Junior	Middle	Middle+	Senior
Technical skills multiplier					
PHP	1.5	1.5	1.3	1.1	1
PHP OOP	1.5	1.5	1.3	1.1	1
MySQL	1	1	1.5	1.5	1.3
Drupal Site Building	1.5	1.5	1.3	1.1	1
Drupal API & Development	1.5	1.5	1.5	1.3	1.1
Drupal Theming	1.5	1.5	1.5	1.3	1.1
Decoupling / Integration	1	1	1.1	1.3	1.5
SEO	1	1.1	1.3	1.5	1.5
System performance	1	1	1.1	1.3	1.5
Testing	1	1	1.1	1.3	1.5
Drush	1	1		1	1
Git	1		1.1		1.3
Docker & DevOps	1	1	1.1	1.3	1.5
Computer Science & Architecture	1	1	1.1	1.3	1.5
Solution providing	1	1	1.1	1.3	1.5
Estimates	1	1	1.1	1.3	1.5
Collaboration with Drupal.org	1	1	1	1	
Technical mentorship	1	1	1.1	1.3	1.5

1. Загальні поняття веб-програмування
 - 1.1. Поняття про веб-програмування. Бекенд та фронтенд розробка
 - 1.2. Класифікація мов програмування та розмітки
 - 1.2.1. Мова HTML та її особливості
 - 1.2.2. Мова CSS та її особливості
 - 1.2.3. Мова Javascript та її особливості
 - 1.2.4. Мова PHP та її особливості
 - 1.2.5. Мова запитів реляційних баз даних SQL
 - 1.3. Локальне та серверне оточення
 - 1.3.1. Стеки LAMP та WAMP
 - 1.3.2. Веб-сервери Apache та Nginx
 - 1.3.3. Інтерпретатор мови PHP. Особливості версій PHP 7.x та 5.x
 - 1.3.4. Сервери MySQL / MariaDB
 - 1.4. Засоби відлагодження. Дебагери
 - 1.5. Огляд систем інтегрованої розробки веб-додатків
2. Основи роботи з CMF Drupal (Site building)
 - 2.1. Загальна характеристика системи
 - 2.2. Інсталяція та базові налаштування
 - 2.2.1. Системні вимоги
 - 2.2.2. Налаштування локального оточення
 - 2.2.3. Інсталяція
 - 2.2.4. Використання інтерпретатора Drush
 - 2.2.5. Встановлення модулів та тем
 - 2.2.6. Базові налаштування
 - 2.3. Сутності (Entities) в Drupal
 - 2.3.1. Поняття сутності. Основні сутності в Drupal
 - 2.3.2. Типи контенту, поля, налаштування форм і режимів відображення

- 2.3.3. Таксономія та її використання
- 2.3.4. Профілі користувачів та їх налаштування
- 2.3.5. Блоки та їх налаштування
- 2.3.6. Меню та їх налаштування
- 2.4. Конструктор запитів Views
 - 2.4.1. Загальні відомості про конструктор запитів Views
 - 2.4.2. Принципи використання Views
 - 2.4.3. Побудова сторінок та блоків з використанням Views
 - 2.4.4. Використання режимів відображення у Views
 - 2.4.5. Статичні та динамічні фільтри у Views
 - 2.4.6. Налаштування агрегації даних у Views
- 2.5. Керування правами доступу
 - 2.5.1. Структура облікових записів користувачів
 - 2.5.2. Користувацькі ролі
 - 2.5.3. Права доступу
 - 2.5.4. Підходи до організації розподіленого доступу до контенту
 - 2.5.5. Засоби для розширення функціоналу управління доступом
 - 2.5.6. Використання облікових записів та ролей
- 2.6. Управління конфігураціями в Drupal
 - 2.6.1. Поняття конфігурації
 - 2.6.2. Менеджер конфігурацій та його використання
 - 2.6.3. Експорт конфігурацій
 - 2.6.4. Імпорт конфігурацій
 - 2.6.5. Використання Drush для роботи з конфігураціями
 - 2.6.6. Типові рішення для управління конфігураціями
- 2.7. Рішення для безпеки та захисту даних в Drupal
 - 2.7.1. Загальні правила щодо забезпечення безпеки і захисту даних
 - 2.7.2. Системне логування
 - 2.7.3. Управління правами доступу до файлової системи
 - 2.7.4. Потенційно небезпечні елементи конфігурацій

2.7.5. Оновлення та патчі

2.7.6. Типові методи виявлення та нейтралізації вразливостей

3. Основи розробки веб-програмування в CMF Drupal

3.1. Поняття про модулі та теми. Стандарти кодування. Відлагодження

3.2. Розробка модулів

3.2.1 Структура файлів та директорій модуля

3.2.2. Структура info.yml

3.2.3. Роути

3.2.4. Контроллери

3.2.5. Плагіни

3.2.6. Форми

3.2.7. Сервіси

3.2.8. Права доступу

3.3. Хуки та події

3.4. Робота з сутностями. Entity API

3.5. Робота з базою даних. Database API

3.6. Робота з Configuration API

3.7. Робота з State API

3.8. Розробка тем

3.8.1. Структура теми

3.8.2. Структура .info.yml

3.8.3. Структура .theme

3.8.4. Технологія twig

3.8.5. Темплейти та їх використання

3.8.6. Бібліотеки

3.8.7. Робота з CSS

3.8.8. Робота з JavaScript

3.8.9. Preprocess-функції та theme suggestions

3.8.10. Особливості розробки тем, відлагодження тем

3.9. Розробка безпечного коду

- 3.9.1. Стандарти кодування
- 3.9.2. Фільтрація вхідного потоку даних
- 3.9.3. Захист від SQL injection
- 3.9.4. Best practices for writing secure code

Робоча програма автоматизованого курсу з підготовки web-програміста рівня

Trainee.

№ п/п	Назва теми та її зміст	Об'єм в годинах
	<p>Лекційні заняття:</p> <p>1.1. Синтаксис і семантика HTML 2</p> <p>1.2. Принципи використання CSS 2</p> <p>1.3. Типові підходи до верстки. Моделі flexbox та grid. 2</p> <p>1.4. Основи JS 2</p> <p>1.5. Синтаксис мови PHP, основні конструкції, типи і структури даних 2</p> <p>1.6. Типові конструкції PHP: умовний оператор, оператор вибору, цикли, функції 2</p> <p>1.7. ООП в PHP 2</p> <p>1.8. Робота з БД в PHP 2</p> <p>1.9. Основи фреймворку Drupal 2</p> <p>1.10. Основи сайт-білдінгу 2</p> <p>1.11. Основи темізації в Drupal 2</p> <p>1.12. Drupal core API 2</p> <p style="text-align: right;">Разом: 48</p> <p>Практичні заняття:</p> <p>1.1. Створення HTML-розмітки 2</p> <p>1.2. Створення базових CSS стилів 2</p> <p>1.3. Верстка респонсивного макету сторінки 2</p> <p>1.4. Інтеграція JS бібліотеки в код HTML сторінки 2</p> <p>1.5. Створення лінійного PHP скрипта 2</p> <p>1.6. Використання типових конструкцій для ітеративної генерації контенту 2</p> <p>1.7. Створення базового класу 2</p> <p>1.8. Використання БД MariaDB 2</p> <p>1.9. Інсталяція Drupal, розгортання оточення для розробки 2</p> <p>1.10. Конфігурування лендінг-сторінки згідно макету 2</p> <p>1.11. Темізація лендінг-сторінки згідно макету 2</p> <p>1.12. Створення кастомної форми зворотнього зв'язку та інтеграція її в лендінг-сторінку згідно макету 2</p> <p style="text-align: right;">Разом: 48</p> <p>Розробка практичного проєкту 70</p> <p>Захист практичного проєкту 2</p>	
	Разом:	126

Лістинг вихідного коду

(модуль статичного аналізу з використанням LangGraph)

```
"""LangGraph workflow: load skill matrix + CV, analyze relevance per level (OpenAI LLM)."""
```

```
from __future__ import annotations
```

```
from typing import TypedDict
```

```
from langgraph.graph import StateGraph, START, END
```

```
from app.services.google_sheets import get_skill_matrix
```

```
from app.services.pdf_loader import load_pdf_text_from_url
```

```
from app.services.llm_analyzer import analyze_level_with_llm
```

```
class AnalysisState(TypedDict, total=False):
```

```
    cv_url: str
```

```
    target_level: str | None
```

```
    sheet_id: str | None
```

```
    skill_matrix: dict[str, list[dict[str, str]]]
```

```
    cv_text: str
```

```
    level_scores: list[dict]
```

```
    best_fit_level: str | None
```

```
    error: str | None
```

```
def node_load_skill_matrix(state: AnalysisState) -> AnalysisState:
```

```
"""Load skill matrix from Google Sheets."""
```

```
try:
```

```
    matrix = get_skill_matrix(sheet_id=state.get("sheet_id"))
```

```
except Exception as e:
```

```
    return {**state, "error": str(e)}
```

```
return {**state, "skill_matrix": matrix, "error": None}
```

```
def node_load_cv(state: AnalysisState) -> AnalysisState:
```

```
    """Fetch PDF from URL and extract text."""
```

```
    if state.get("error"):
```

```
        return state
```

```
    try:
```

```
        text = load_pdf_text_from_url(state["cv_url"])
```

```
    except Exception as e:
```

```
        return {**state, "error": str(e)}
```

```
    return {**state, "cv_text": text}
```

```
def node_analyze(state: AnalysisState) -> AnalysisState:
```

```
    """Score CV against each level using OpenAI LLM."""
```

```
    if state.get("error"):
```

```
        return state
```

```
    matrix = state["skill_matrix"]
```

```
    cv_text = state["cv_text"]
```

```
    target = state.get("target_level")
```

```
    levels_to_eval = [target] if target and target in matrix else list(matrix.keys())
```

```
    level_scores = []
```

```

best_fit = None
best_score = -1.0

for level in levels_to_eval:
    skills = matrix.get(level, [])
    if not skills:
        continue
    try:
        llm_matches = analyze_level_with_llm(level_name=level, skills=skills,
cv_text=cv_text)
    except Exception as e:
        return {**state, "error": f"LLM analysis failed: {e}"}
    matches = []
    for m in llm_matches:
        matched = m.get("matched", False)
        matches.append({
            "skill": m.get("skill", ""),
            "level_required": level,
            "level_found": level if matched else None,
            "matched": matched,
            "evidence": m.get("evidence"),
        })
    matched_count = sum(1 for m in matches if m["matched"])
    total = len(matches)
    score_percent = round(100.0 * matched_count / total, 1) if total else 0.0
    level_scores.append({
        "level": level,
        "score_percent": score_percent,
        "matched_skills": matched_count,
        "total_skills": total,
    })

```

```

        "skill_matches": matches,
    })
    if score_percent > best_score:
        best_score = score_percent
        best_fit = level

return {
    **state,
    "level_scores": level_scores,
    "best_fit_level": best_fit,
}

```

```
def build_analysis_graph():
```

```
    """Build LangGraph: START -> load_skill_matrix -> load_cv -> analyze -> END."""
```

```
    workflow = StateGraph(AnalysisState)
```

```
    workflow.add_node("load_skill_matrix", node_load_skill_matrix)
```

```
    workflow.add_node("load_cv", node_load_cv)
```

```
    workflow.add_node("analyze", node_analyze)
```

```
    workflow.add_edge(START, "load_skill_matrix")
```

```
    workflow.add_edge("load_skill_matrix", "load_cv")
```

```
    workflow.add_edge("load_cv", "analyze")
```

```
    workflow.add_edge("analyze", END)
```

```
    return workflow.compile()
```

```
def run_analysis(cv_url: str, target_level: str | None = None, sheet_id: str | None =
None) -> dict:
```

```
    """
```

```
    Run the full analysis pipeline and return result suitable for API response.
```

```
    """
```

```
    state: AnalysisState = {
```

```
        "cv_url": cv_url,
```

```
        "target_level": target_level,
```

```
        "sheet_id": sheet_id,
```

```
        "skill_matrix": {},
```

```
        "cv_text": "",
```

```
        "level_scores": [],
```

```
        "best_fit_level": None,
```

```
        "error": None,
```

```
    }
```

```
    graph = build_analysis_graph()
```

```
    final = graph.invoke(state)
```

```
    if final.get("error"):
```

```
        return {
```

```
            "cv_url": cv_url,
```

```
            "levels": [],
```

```
            "best_fit_level": None,
```

```
            "message": f'Error: {final["error"]}',
```

```
        }
```

```
    return {
```

```
        "cv_url": cv_url,
```

```
        "levels": final["level_scores"],
```

```
        "best_fit_level": final["best_fit_level"],
```

```
        "message": None,
```

```
    }
```

Лістинг вихідного коду
(модуль аналізу скіл матриці)

```
"""Google Sheets client for Drupal Developer skill matrix."""
from __future__ import annotations

import os
from typing import Any

from google.oauth2.credentials import Credentials
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError

from app.config import get_settings

# Scope for read-only Sheets access
SCOPES = ["https://www.googleapis.com/auth/spreadsheets.readonly"]

def _get_credentials() -> Credentials | None:
    """Obtain credentials from credentials.json and optional token.json."""
    settings = get_settings()
    creds = None
    token_path = settings.google_token_path
    creds_path = settings.google_credentials_path

    if os.path.exists(token_path):
        creds = Credentials.from_authorized_user_file(token_path, SCOPES)
```

```
if not creds or not creds.valid:
```

```
    if creds and creds.expired and creds.refresh_token:
```

```
        creds.refresh(Request())
```

```
    elif os.path.exists(creds_path):
```

```
        flow = InstalledAppFlow.from_client_secrets_file(creds_path, SCOPES)
```

```
        creds = flow.run_local_server(port=0)
```

```
    else:
```

```
        return None
```

```
if creds:
```

```
    with open(token_path, "w") as f:
```

```
        f.write(creds.to_json())
```

```
return creds
```

```
# Rows to skip (section headers or footer) in Drupal BE Grade sheet
```

```
_SKIP_SKILL_NAMES = frozenset({
```

```
    "technical skills",
```

```
    "points",
```

```
    "required",
```

```
    "expected",
```

```
    "strong",
```

```
    "total",
```

```
    "to be passed",
```

```
    "knowledge or skill expected as mandatory for the current level",
```

```
    "knowledge or skill expected as good point, but not required",
```

```
    "knowledge or skill is not mandatory, but recognizing as additional strong point",
```

```
})
```

```

def _parse_skill_matrix_rows(values: list[list[Any]]) -> dict[str, list[dict[str, str]]]:
    """
    Parse sheet rows into level -> list of skills.
    Supports Drupal BE Grade v2 layout:
    - Row 0: headers; column A = skill column (may be "Skill" or a character), B..G =
    Trainee, Junior, Middle, Middle+, Senior, Tech Lead.
    - Rows 1..N: column A = skill name, other columns = requirement text per level.
    - Skips section rows (e.g. "Technical skills") and stops at "Points" / "Required" etc.
    """
    if not values or len(values) < 2:
        return {}

    headers = [str(h).strip() for h in values[0]]
    skill_col_idx = 0
    for i, h in enumerate(headers):
        if h and (h.lower() == "skill" or h.lower() == "skills"):
            skill_col_idx = i
            break

    level_columns: dict[str, int] = {}
    for i, h in enumerate(headers):
        if i != skill_col_idx and h and h.lower() not in ("skill", "skills"):
            level_columns[h.strip()] = i

    result: dict[str, list[dict[str, str]]] = {level: [] for level in level_columns}

    for row in values[1:]:
        if len(row) <= skill_col_idx:

```

```

        continue
    skill_name = str(row[skill_col_idx]).strip()
    if not skill_name:
        continue
    if skill_name.lower() in _SKIP_SKILL_NAMES:
        continue
    for level, col_idx in level_columns.items():
        requirement = ""
        if col_idx < len(row) and row[col_idx]:
            requirement = str(row[col_idx]).strip()
        result[level].append({"skill": skill_name, "requirement": requirement})

return result

```

```

def _get_first_sheet_range(service: Any, spreadsheet_id: str) -> str:
    """Return range for the first sheet (e.g. 'Sheet1!A1:H30') so any sheet name works."""
    try:
        meta = service.spreadsheets().get(spreadsheetId=spreadsheet_id).execute()
        sheets = meta.get("sheets", [])
        if sheets:
            title = sheets[0].get("properties", {}).get("title", "Sheet1")
            return f"{title}!A1:H30"
    except HttpError:
        pass
    return "Sheet1!A1:H30"

```

```

def get_skill_matrix(sheet_id: str | None = None, range_name: str | None = None) ->
dict[str, list[dict[str, str]]]:

```

```

"""
Fetch Drupal Developer skill matrix from Google Sheets.
Returns dict: level name -> list of { "skill": str, "requirement": str }.
If range_name is not set, uses the first sheet and range A1:H30 (Drupal BE Grade
layout).
"""

settings = get_settings()
sid = sheet_id or settings.google_sheet_id
if not sid:
    raise ValueError("Google Sheet ID is not configured (google_sheet_id or
parameter)")

creds = _get_credentials()
if not creds:
    raise RuntimeError(
        "Google credentials not found. Place credentials.json in working directory and
run once to authorize."
    )

service = build("sheets", "v4", credentials=creds)
sheet = service.spreadsheets()
if not range_name:
    range_name = _get_first_sheet_range(service, sid)
try:
    result = sheet.values().get(spreadsheetId=sid, range=range_name).execute()
except HttpError as e:
    raise RuntimeError(f"Failed to fetch Google Sheet: {e}") from e

values = result.get("values", [])
return _parse_skill_matrix_rows(values)

```

Лістинг вихідного коду
(модуль LLM аналізу)

```
"""OpenAI LLM-based CV vs skill matrix analysis."""
from __future__ import annotations

from pydantic import BaseModel, Field

from app.config import get_settings

# Max CV chars to send to the LLM to stay within context
CV_TEXT_TRUNCATE = 14_000

class SkillMatchResult(BaseModel):
    """One skill match from LLM."""

    skill: str = Field(description="Exact skill name as in the list")
    matched: bool = Field(description="True if the CV demonstrates this skill or requirement")
    evidence: str | None = Field(default=None, description="Short quote or phrase from the CV that supports the match, or null if not matched")

class LevelAnalysisResult(BaseModel):
    """LLM output for one level: list of skill match results in the same order as input skills."""

    skill_matches: list[SkillMatchResult] = Field(description="One entry per skill, in the same order as the skills list provided")

def analyze_level_with_llm(
```

```

level_name: str,
skills: list[dict[str, str]],
cv_text: str,
) -> list[dict]:
    """
    Use OpenAI GPT to analyze CV text against one level's skills.
    Returns list of { "skill", "matched", "evidence" } in same order as skills.
    """

    from langchain_openai import ChatOpenAI
    settings = get_settings()
    if not settings.openai_api_key:
        raise RuntimeError("OPENAI_API_KEY is not set. Configure it in .env for
LLM-based analysis.")
    cv_truncated = cv_text.strip()
    if len(cv_truncated) > CV_TEXT_TRUNCATE:
        cv_truncated = cv_truncated[: CV_TEXT_TRUNCATE] + "\n\n[... CV text
truncated ...]"
    skills_blob = "\n".join(
        f"- **{s.get('skill', '')}** : {s.get('requirement', '')} or 'No specific requirement'"
        for s in skills
    )
    system = (
        "You are an expert recruiter evaluating a developer's CV against a Drupal Backend
skill matrix. "
        "For each skill and its requirement, decide whether the CV demonstrates that skill
(matched: true) or not (matched: false). "
        "Use only evidence from the CV text. If matched, provide a short evidence quote
from the CV; if not matched, set evidence to null. "
        "Return skill_matches in the EXACT same order as the skills list given."
    )

```

```

user = f'""""**Level:** {level_name}
**Skills and requirements for this level:**
{skills_blob}
**CV text:**
{cv_truncated}
Analyze the CV against each skill above. For each skill, set matched to true only if the
CV clearly demonstrates that skill or the requirement. Return skill_matches in the same
order as the list above.'""
llm = ChatOpenAI(
    model=settings.openai_model,
    api_key=settings.openai_api_key,
    temperature=0,
)
structured_llm = llm.with_structured_output(LevelAnalysisResult)
messages = [{"role": "system", "content": system}, {"role": "user", "content": user}]
result = structured_llm.invoke(messages)
# Map LLM results by skill name so we preserve order and handle missing/extra
items
by_skill: dict[str, dict] = {}
for r in result.skill_matches:
    by_skill[r.skill.strip()] = {"skill": r.skill, "matched": r.matched, "evidence":
r.evidence}
out = []
for s in skills:
    name = s.get("skill", "").strip()
    if name in by_skill:
        out.append(by_skill[name])
    else:
        out.append({"skill": name, "matched": False, "evidence": None})
return out

```

Лістинг вихідного коду
(модуль завантаження CV)

```
"""Fetch PDF from URL and extract text."""
from __future__ import annotations

import io
import httpx
from pypdf import PdfReader

def load_pdf_text_from_url(url: str, timeout: float = 30.0) -> str:
    """
    Download PDF from URL and extract text content.
    """
    with httpx.Client(timeout=timeout, follow_redirects=True) as client:
        response = client.get(url)
        response.raise_for_status()
        if "application/pdf" not in response.headers.get("content-type", ""):
            # Allow if no content-type or octet-stream (some servers don't set pdf)
            if response.headers.get("content-type") and "application/pdf" not in
response.headers.get("content-type", ""):
                if "text" in (response.headers.get("content-type") or ""):
                    raise ValueError("URL did not return a PDF document")
        buffer = io.BytesIO(response.content)
        reader = PdfReader(buffer)
        parts = []
        for page in reader.pages:
            text = page.extract_text()
            if text:
                parts.append(text)
        return "\n\n".join(parts) if parts else ""
```

Лістинг вихідного коду
(API для аналізу CV)

```
"""API routes for CV analysis."""
from app.models import AnalyzeCVRequest, AnalyzeCVResponse, LevelScore,
SkillMatch
from app.services import run_analysis
from fastapi import APIRouter, HTTPException

router = APIRouter(prefix="/api/v1", tags=["CV Analysis"])

def _to_level_score(raw: dict) -> LevelScore:
    return LevelScore(
        level=raw["level"],
        score_percent=raw["score_percent"],
        matched_skills=raw["matched_skills"],
        total_skills=raw["total_skills"],
        skill_matches=[SkillMatch(**m) for m in raw.get("skill_matches", [])],
    )

@router.post(
    "/analyze",
    response_model=AnalyzeCVResponse,
    summary="Analyze CV against Drupal skill matrix",
    description="Fetches the PDF from the given URL, extracts text, and scores it against
the Drupal Developer skill matrix from Google Sheets for each level (e.g. Junior,
Middle, Senior).",
)
```

```

async def analyze_cv(request: AnalyzeCVRequest) -> AnalyzeCVResponse:
    """
    Analyze a CV (PDF) from a URL against the Drupal Developer skill matrix.
    Returns relevance scores per level and the best-fit level.
    """
    try:
        result = run_analysis(
            cv_url=str(request.cv_url),
            target_level=request.target_level,
            sheet_id=request.sheet_id,
        )
    except ValueError as e:
        raise HTTPException(status_code=400, detail=str(e))
    except RuntimeError as e:
        raise HTTPException(status_code=503, detail=str(e))

    levels = [_to_level_score(lev) for lev in result["levels"]]
    return AnalyzeCVResponse(
        cv_url=result["cv_url"],
        levels=levels,
        best_fit_level=result.get("best_fit_level"),
        message=result.get("message"),
    )

    @router.get(
        "/health",
        summary="Health check",
        description="Simple health check for the service.",
    )

async def health() -> dict:
    return {"status": "ok", "service": "cv-skill-matrix"}

```

Довідка про впровадження в Приватному вищому навчальному закладі
 “Міжнародний Економіко-Гуманітарний Університет”

ЗАТВЕРДЖУЮ
 Проректор з наукової роботи
 Приватного вищого навчального закладу
 «Міжнародний економіко-гуманітарний
 університет імені академіка Степана Дем'ячука»
 проф. НАУЧНИЙ МЕДИНСЬКА



АКТ
 про впровадження у виробництво
 результатів дисертації на здобуття ступеня доктора філософії

Цим актом стверджується, що результати дисертації на тему: «Інформаційна технологія управління багаторівневим навчанням та підбором web-програмістів в ІТ-компанії», яку представлено на здобуття ступеня доктора філософії з галузі знань F «Інформаційні технології» зі спеціальності F3 «Комп'ютерні науки», виконаної Богумом Олегом Михайловичем, впроваджено у Приватному вищому навчальному закладі «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'ячука».

1. Вид впроваджуваних результатів: Комп'ютерна програма «Інтелектуальна інформаційна система з відбору web-програмістів в ІТ-компанії», яка працює у двох режимах: освітній процес у закладі вищої освіти та рекрутингу в ІТ-компанії. У закладі вищої освіти впровадження відбувається в межах викладання освітньої компоненти «Web-програмування», «Web-технології» та ін.


2. Назва отриманих результатів: Акт про впровадження у виробничі процеси SIA AnyFor (Латвія).

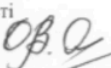
3. Значущість отриманих результатів: впровадження інноваційних інформаційних технологій у систему вищої освіти України та ІТ-галузь; розробка та апробація комп'ютерних технологій навчання, орієнтованих на відбір та порангову підготовку web-програмістів для сучасного цифрового суспільства; створення основи для подальших досліджень у галузі інтелектуальних інформаційних систем з використанням технологій штучного інтелекту; вдосконалення екосистеми «університет - випускник - роботодавець».

4. Зв'язок роботи з науковими програмами, планами, темами: кафедри інформаційних систем та обчислювальних методів Приватного вищого навчального закладу «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'ячука»:

- № 0118U003103 «Програмна модифікація методів штучного інтелекту для аналізу графічної інформації на антиплагіат on-line»;

- № 0118U003104 «Створення та впровадження автоматизованої системи керування процесом навчання в електронному освітньому просторі».

Завідувач кафедри Інформаційних систем
 та обчислювальних методів, де безпосередньо
 впроваджені результати дисертації  Юрій ЛОТЮК 13.02 2026 р.

Начальник відділу з наукової діяльності
 та аспірантури  Оксана ЯРОМЕНКО 13.02 2026 р.

Заступник декана
 факультету кібернетики  Валентина ЮСКОВИЧ-ЖУКОВСЬКА 13.02 2026 р.

Здобувач  Олег БОГУТ 13.02 2026 р.

Довідка про впровадження в компанії AnyforSoft



SIA "Anyfor" d.b.a. AnyforSoft

Mārupes novads, Mārupe,
Mazā Briežu iela 3, LV-2167, Latvia
VAT: LV40103520457

CERTIFICATE

On the Implementation of an Intelligent Information System for Managing Recruitment and Learning Processes in IT Companies

This is to certify that an Intelligent Information System for managing recruitment and learning processes within IT companies has been successfully designed, implemented, and put into operational use by Oleg Bogut.

The system was developed with the objective of optimizing and formalizing key organizational processes related to the engagement, evaluation, and professional development of project teams and contractors. The implemented solution integrates data-driven decision-making mechanisms, automated workflow management, and analytical tools to enhance the transparency, efficiency, and scalability of recruitment and learning activities.

The system has been fully implemented and has been continuously used in operational activities from October 2025 to the present date.

This certificate is issued for the purpose of confirming the implementation and practical utilization of the aforementioned Intelligent Information System in the context of academic research and may be submitted as supporting documentation for a PhD dissertation defense.

Aleksejs Pavlovs
Board member at SIA "Anyfor", d.b.a AnyforSoft
aleksejs.pavlovs@anyforsoft.com

13.02.2026

A handwritten signature in black ink, appearing to be 'AP', written over a horizontal line.

ДОДАТОК 3.

Довідка про впровадження в компанії ТОВ Смарт Кінг

ТОВ СМАРТ КІНГ ЛТД
33015, Україна
М. Рівне, Київська, 110А
ЄДРПОУ 42479185
Виконавчий директор
Клап А. В.

ДОВІДКА

Про впровадження інтелектуальної інформаційної системи управління процесами підбору та навчання контракторів ІТ-компаній

Цим підтверджується, що інтелектуальна інформаційна система управління процесами підбору та навчання в ІТ-компаніях була розроблена, впроваджена та введена в експлуатацію.

Система створена з метою оптимізації та формалізації ключових організаційних процесів, пов'язаних із залученням, оцінюванням та професійним розвитком команд і підрядників. Впроваджене рішення інтегрує механізми прийняття рішень на основі даних засобами агентного штучного інтелекту, автоматизоване управління бізнес-процесами та аналітичні інструменти для підвищення прозорості, ефективності та масштабованості процесів підбору й навчання персоналу.

Архітектура системи побудована відповідно до сучасних принципів програмної інженерії, зокрема модульності, рольової моделі доступу, механізмів захисту даних та масштабованої інфраструктури. Для підтримки процесів оцінювання кандидатів, зіставлення компетентностей і формування персоналізованих рекомендацій щодо навчання застосовуються інтелектуальні алгоритми і технології агентного штучного інтелекту.

Система повністю впроваджена та використовується в операційній діяльності з вересня 2024 року по теперішній час.

Довідку видано Богуту Олегу Михайловичу для підтвердження факту впровадження та практичного використання зазначеної інтелектуальної інформаційної системи в межах наукового дослідження та подання її під час захисту дисертації на здобуття ступеня доктора філософії (PhD).

Виконавчий директор
Клап А. В.



12.02.2026