

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**
(підпис)

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
**Завідувач кафедри інформаційних
систем і технологій**

_____ **Швиденко Михайло**
(підпис)

«___» _____ 2026 р.

«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Інформаційна система пошуку роботи з оцінкою професійних навичок»**
Спеціальність «122 Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми
д.е.н., професор

(підпис)

Роман РУДЕНСЬКИЙ

**Керівник бакалаврської
кваліфікаційної роботи**
(науковий ступінь та вчене
звання)

(підпис)

Максим МОКРІЄВ

Виконав

(підпис)

Ростислав ГРЕЧКА

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій**

ЗАТВЕРДЖУЮ

**Завідувач кафедри інформаційних систем і
технологій**

к.е.н., доцент Михайло Швиденко
(підпис)

«___» _____ 2026 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Гречці Ростиславу Руслановичу

(прізвище, ім'я, по батькові)

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки» Тема
бакалаврської кваліфікаційної роботи

«Інформаційна система пошуку роботи з оцінкою професійних навичок»

затверджена наказом від «06» січня 2026 р. № 46«С». Термін подання
завершеної роботи на кафедру «25» _____ 05 _____ 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи:

Перелік питань, що підлягають дослідженню:

Перелік графічних документів (за необхідності).

Дата видачі завдання «6» _____ 01 _____ 2026 р.

**Керівник бакалаврської
кваліфікаційної роботи**

_____ Максим Мокрієв

Завдання взяв до виконання

_____ Ростислав Гречка

Календарний план

№ з/п	Назва етапів виконання бакалаврської кваліфікаційної роботи	Строк виконання етапів бакалаврської кваліфікаційної роботи	Примітка
1	Одержання завдання, аналіз предметної області та літературних джерел. Формування вимог до системи.	06.01.2025 – 24.01.2025	Виконано
2	Одержання завдання, аналіз предметної області та літературних джерел. Формування вимог до системи.	27.01.2025 – 14.02.2025	Виконано
3	Проектування інформаційної системи: розробка ER-діаграм бази даних, специфікацій API та алгоритмів тестування навичок.	17.02.2025 – 07.03.2025	Виконано
4	Практична реалізація: розробка бази даних, серверної частини (бекенду) та інтеграція логіки оцінювання.	10.03.2025 – 28.03.2025	Виконано
5	Практична реалізація: розробка клієнтської частини (фронтенду) та кабінетів користувача/роботодавця.	31.03.2025 – 18.04.2025	Виконано
6	Тестування працездатності системи, налагодження коду та оптимізація роботи модулів.	21.04.2025 – 02.05.2025	Виконано
7	Оформлення пояснювальної записки (тексту диплому) згідно з вимогами ДСТУ та підготовка	05.05.2025 – 16.05.2025	Виконано

	графічного матеріалу.		
8	Проходження нормоконтролю, перевірка на плагіат та отримання відгуку керівника й рецензії.	19.05.2025 – 23.05.2025	В процесі
9	Передзахист роботи на кафедрі та фінальна підготовка презентації і доповіді до захисту.	26.05.2025 – 30.05.2025	В процесі

Студент

Гречка Ростислав Русланович

(підпис)

(ПІБ)

Керівник бакалаврської кваліфікаційної роботи

Мокрісв Максим Володимирович

(підпис)

(ПІБ)

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ТЕХНОЛОГІЙ ПОШУКУ РОБОТИ	7
1.1. Особливості сучасного ринку праці та онлайн-платформ працевлаштування.....	7
1.2. Аналіз існуючих інформаційних систем пошуку роботи	9
1.3. Методи оцінки професійних навичок користувачів	10
1.4. Технології веб-розробки для створення інформаційних систем	12
1.5. Постановка задачі та формування вимог до системи	14
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ РОБОТИ.....	17
2.1. Загальна структура та архітектура системи.....	17
2.2. Проєктування бази даних системи	19
2.3. Розробка функціональної моделі системи	22
2.4. Проєктування інтерфейсу користувача	24
2.5. Вибір програмних засобів та технологій реалізації	26
РОЗДІЛ 3. РОЗРОБКА ВЕБ-СИСТЕМИ ПОШУКУ РОБОТИ.....	29
3.1. Реалізація серверної частини системи.....	29
3.2. Реалізація клієнтської частини веб-застосунку	31
3.3. Розробка модуля оцінки професійних навичок.....	32
3.4. Реалізація авторизації та особистого кабінету користувача	36
3.5. Інтеграція бази вакансій та пошукових механізмів	37
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ.....	40
4.1. Тестування функціональних можливостей системи.....	41
4.2. Аналіз продуктивності веб-застосунку	40
4.3. Оцінка зручності користувацького інтерфейсу.....	42
4.4. Аналіз результатів роботи модуля оцінки навичок	44
4.5. Перспективи розвитку та вдосконалення системи	45
ВИСНОВКИ.....	47
ВИКОРИСТАНІ ДЖЕРЕЛА.....	50
ДОДАТКИ.....	56
Додаток А. UML-діаграми системи.....	57
Додаток Б. Структура бази даних.....	58
Додаток В. Фрагменти програмного коду	59
Додаток Г. Інтерфейс веб-застосунку	74

ВСТУП

Актуальність теми. У сучасних умовах стрімкого розвитку цифрових технологій та зростання конкуренції на ринку праці особливої актуальності набувають інформаційні системи пошуку роботи. Значна кількість роботодавців і кандидатів використовують онлайн-платформи для пошуку вакансій, підбору персоналу та оцінки професійної компетентності. Традиційні методи працевлаштування поступово втрачають ефективність через великі обсяги інформації, складність аналізу професійних навичок та недостатню автоматизацію процесів підбору кадрів.

Сучасні веб-платформи працевлаштування забезпечують швидкий доступ до вакансій, однак більшість із них не містить достатньо ефективних механізмів оцінювання професійних навичок користувачів. У результаті роботодавці витрачають значний час на аналіз резюме та перевірку компетенцій кандидатів, а користувачі не завжди можуть об'єктивно оцінити власний рівень підготовки.

Розробка інформаційної системи пошуку роботи з модулем оцінки професійних навичок дозволяє автоматизувати процес підбору персоналу, підвищити якість пошуку вакансій та забезпечити більш точне співставлення вимог роботодавця з компетенціями кандидата. Це сприяє оптимізації процесу працевлаштування та підвищенню ефективності функціонування ринку праці загалом.

Метою роботи є розробка інформаційної системи пошуку роботи з оцінкою професійних навичок користувачів для автоматизації процесу працевлаштування, аналізу компетенцій кандидатів та підвищення ефективності взаємодії між роботодавцями й пошукачами роботи.

Об'єктом дослідження є процес організації онлайн-пошуку роботи та оцінювання професійних навичок користувачів у сучасних інформаційних системах.

Предметом дослідження є методи, моделі та програмні засоби розробки веб-орієнтованої інформаційної системи пошуку роботи з функціями оцінки професійних навичок користувачів.

Для досягнення поставленої мети необхідно **виконати такі завдання:**

1. проаналізувати сучасний стан ринку праці та особливості функціонування онлайн-платформ працевлаштування.
2. дослідити існуючі інформаційні системи пошуку роботи та визначити їх переваги і недоліки.
3. розглянути методи оцінки професійних навичок користувачів.
4. проаналізувати сучасні технології веб-розробки для створення інформаційних систем.
5. сформулювати функціональні та нефункціональні вимоги до системи.
6. спроектувати структуру, архітектуру та базу даних інформаційної системи.
7. розробити серверну та клієнтську частини веб-застосунку.
8. реалізувати модуль оцінки професійних навичок користувачів.
9. провести тестування функціональних можливостей та оцінити ефективність роботи системи.
10. визначити перспективи розвитку та вдосконалення інформаційної системи.

У процесі виконання роботи використовувалися **такі методи дослідження:**

- аналіз і узагальнення наукової літератури та сучасних веб-технологій;
- порівняльний аналіз існуючих платформ пошуку роботи;
- методи системного аналізу та проєктування інформаційних систем;
- методи моделювання баз даних і програмної архітектури;
- методи веб-програмування для реалізації функціоналу системи;
- тестування програмного забезпечення для перевірки коректності роботи системи;

- методи оцінювання зручності користувацького інтерфейсу та продуктивності веб-застосунку.

Теоретична значущість роботи полягає в узагальненні сучасних підходів до створення інформаційних систем пошуку роботи, дослідженні методів оцінювання професійних навичок користувачів та аналізі можливостей інтеграції цих методів у веб-орієнтовані системи працевлаштування.

Практична значущість роботи полягає у створенні веб-системи пошуку роботи, яка забезпечує автоматизований підбір вакансій, оцінювання професійних навичок користувачів, зручну взаємодію між роботодавцями та кандидатами, а також підвищення ефективності процесу працевлаштування. Результати роботи можуть бути використані для впровадження у сфері онлайн-рекрутингу, кадрового менеджменту та цифрових сервісів працевлаштування.

Структура роботи. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ТЕХНОЛОГІЙ ПОШУКУ РОБОТИ

1.1. Особливості сучасного ринку праці та онлайн-платформ працевлаштування

Сучасний ринок праці зазнає кардинальних трансформацій під впливом цифровізації економіки, глобалізації виробничих процесів та стрімкого розвитку інформаційно-комунікаційних технологій.

Традиційна модель працевлаштування, заснована на особистих контактах та газетних оголошеннях, поступово замінюється цифровими платформами, що забезпечують миттєвий доступ до мільйонів вакансій і резюме. Дослідники підтверджують, що онлайн-платформи суттєво прискорюють процес пошуку роботи та підвищують ефективність підбору персоналу [9], однак також породжують нові виклики у сфері оцінки компетентності кандидатів.

Серед ключових тенденцій, що визначають сучасний ринок праці, особливої уваги заслуговують: зростання частки дистанційної та гібридної зайнятості, посилення попиту на цифрові компетенції, підвищення мобільності трудових ресурсів, а також збільшення частки фрілансу та проектної зайнятості. Аналіз оголошень про вакансії на різних платформах підтверджує, що вимоги до навичок кандидатів стають дедалі більш специфічними та динамічними [10, 15], що вимагає відповідного вдосконалення систем пошуку роботи.

Ринок праці України має власну специфіку, обумовлену історичним контекстом та структурою національної економіки. Цифровізація залучення персоналу через портали вакансій стала важливим напрямом розвитку вітчизняного ринку HR-технологій [34]. Активно розвивається сегмент онлайн-платформ, що пропонують нові інструменти для пошуку роботи та підбору персоналу, зокрема веб-орієнтовані інформаційні системи різного призначення [40].

Онлайн-платформи працевлаштування класифікуються за кількома ознаками. За географічним охопленням виділяють глобальні (LinkedIn, Indeed, Monster), регіональні (Work.ua, Robota.ua, Djinni) та нішеві платформи. За моделлю монетизації розрізняють безкоштовні для кандидатів з платними функціями для роботодавців, передплатні та гібридні моделі. У таблиці 1.1 наведено порівняльну характеристику основних платформ.

Таблиця 1.1

Порівняльна характеристика онлайн-платформ працевлаштування

Платформа	Країна базування	Ключові функції	Оцінка навичок
LinkedIn	США	Пошук вакансій, нетворкінг, рекомендації, LinkedIn Learning	LinkedIn Assessments (онлайн-тести)
Indeed	США	Агрегатор вакансій, відгуки роботодавців	Відсутня
Work.ua	Україна	Пошук роботи, публікація резюме	Відсутня
Robota.ua	Україна	Біржа праці, HR-аналітика	Часткова (рейтинги резюме)
Djinni	Україна (ІТ)	ІТ-вакансії, анонімний пошук кандидатів	Відсутня
HH.ua	Регіон СНД	База резюме, публікація вакансій	Базове тестування

Аналіз наведених даних свідчить, що переважна більшість платформ не містить повноцінного модуля оцінки навичок. Системи рекомендацій вакансій мають значний потенціал для вдосконалення [11]. Дослідження інформаційних систем пошуку демонструють можливість побудови ефективних рекомендаційних підсистем [42], однак у сфері пошуку роботи подібні рішення залишаються недостатньо реалізованими.

Важливою тенденцією є зростання ролі алгоритмічного співставлення при формуванні рекомендацій вакансій. Сучасні системи використовують методи глибокого навчання для аналізу профілів кандидатів та вимог

роботодавців [12]. Ефективність таких рекомендацій значно залежить від повноти даних про навички кандидата. Саме тому інтеграція модуля оцінки навичок набуває стратегічного значення.

Ринок онлайн-платформ демонструє стійку тенденцію до технологічного ускладнення: впровадження штучного інтелекту, аналізу природньої мови, систем рекомендацій та автоматизованого скринінгу кандидатів [15]. Розробка інформаційної системи з інтегрованим модулем оцінки навичок відповідає актуальним тенденціям ринку та дозволяє запропонувати якісно новий рівень сервісу для обох сторін трудових відносин.

1.2. Аналіз існуючих інформаційних систем пошуку роботи

Для ефективного проєктування нової системи необхідно детально проаналізувати існуючі рішення. Аналіз інформаційних систем різного призначення, розроблених вітчизняними дослідниками [33], дозволяє виділити спільні принципи побудови веб-орієнтованих систем та особливості реалізації їх функціоналу.

LinkedIn є найбільшою у світі соціальною мережею для професіоналів, яка поєднує функції пошуку роботи, нетворкінгу та корпоративних комунікацій і налічує понад 950 мільйонів зареєстрованих користувачів у більш ніж 200 країнах. Дослідження ринку інформатики підтверджують, що LinkedIn залишається ключовим джерелом даних для аналізу ринку праці [6, 19]. Суттєвим недоліком є відсутність прямої інтеграції результатів тестування з алгоритмами рекомендацій вакансій.

Indeed є найбільшим у світі агрегатором вакансій із щомісячною аудиторією понад 350 мільйонів унікальних відвідувачів. Дослідники зазначають, що агрегатори вакансій є цінним джерелом для аналізу вимог до навичок на ринку праці [18], навіть якщо самі платформи не пропонують інструментів для оцінювання цих навичок.

Work.ua та Robota.ua є провідними вітчизняними порталами пошуку роботи. Їх переваги: фокус на український ринок, україномовний інтерфейс

та широка база вакансій. Водночас ці платформи не надають інструментів для об'єктивної оцінки рівня навичок кандидатів, що є істотною прогалиною у їх функціоналі.

Djinni є спеціалізованою платформою для ІТ-фахівців в Україні з анонімними профілями кандидатів. Системи, орієнтовані на ІТ-галузь, широко використовуються для аналізу вимог до навичок у технічних спеціальностях [3], однак Djinni не надає інструментів для об'єктивного тестування рівня знань.

Система WoLMIS (Web-based Labor Market Intelligence System), розроблена Boselli та співавторами [1], є прикладом інтелектуальної системи для класифікації веб-вакансій і демонструє можливості автоматизованого аналізу ринку праці. Аналіз підходів до ідентифікації навичок з онлайн-оголошень, проведений Dawson та співавторами [2], свідчить, що існуючі платформи накопичують значний масив даних, але рідко трансформують їх в інструменти самооцінки для кандидатів.

Результати аналізу свідчать про наявність істотних функціональних прогалин: роботодавці витрачають значний ресурс на ручний скринінг кандидатів; у кандидатів відсутній об'єктивний інструмент самооцінки; рекомендації формуються на основі неповних даних про рівень компетентності [11]. Виявлені прогалини визначають напрям розробки нової інформаційної системи.

1.3. Методи оцінки професійних навичок користувачів

Оцінка професійних навичок є одним із ключових елементів сучасних систем управління людськими ресурсами. Гапоненко Н. визначає поняття та сутність професійних навичок фахівця [26], підкреслюючи їх роль у системі компетентностей. В контексті інформаційних систем пошуку роботи методи оцінювання дозволяють підвищити об'єктивність та точність співставлення кандидатів з вимогами роботодавців.

Стандартизоване онлайн-тестування є найбільш поширеним методом автоматизованої оцінки навичок. Khaouja та співавтори у систематичному огляді методів ідентифікації навичок [4] підкреслюють, що стандартизоване тестування є найбільш операціоналізованим методом оцінки компетенцій для масштабних онлайн-систем. Перевагою є висока масштабованість та об'єктивність результатів.

Метод портфоліо передбачає пред'явлення кандидатом реальних проектів, робіт та інших доказів кваліфікації. Ковальчук В.І. та Заїка А.О. досліджують питання формування цифрової компетентності [31], зазначаючи, що портфоліо цифрових проектів є важливим інструментом демонстрації практичних умінь у сучасному середовищі.

Для ІТ-галузі специфічним методом є практичні кодинг-тести. Senger та співавтори у дослідженні методів глибокого навчання для аналізу ринку праці [10] зазначають, що практичні технічні завдання є найбільш валідним методом оцінки ІТ-компетенцій, оскільки безпосередньо вимірюють здатність виконувати реальні робочі задачі.

Сертифікація являє собою формальне підтвердження кваліфікації від авторитетної організації. Дослідження SkillRec [8] демонструє, що наявність верифікованих сертифікатів суттєво підвищує точність рекомендацій навичок для розвитку кар'єри. AI-рейтингування засноване на аналізі великих масивів даних з використанням алгоритмів машинного навчання. Zhang Y. та співавтори розробили контекстно-залежний підхід для вилучення технічних та м'яких навичок [16]. Системи штучного інтелекту здатні виявляти складні патерни компетентності, однак вимагають значних обсягів навчальних даних [13]. У таблиці 1.2 наведено систематизований порівняльний аналіз основних методів оцінки навичок за результатами аналізу наукової літератури [16].

Аналіз методів оцінювання свідчить, що найбільш ефективним для інтеграції в інформаційну систему є комбінований метод, що поєднує стандартизоване онлайн-тестування з елементами AI-рейтингування. Nikiforos та співавтори у систематичному огляді систем рекомендацій

вакансій [7] підтверджують, що поєднання формалізованого тестування з інтелектуальними рекомендаціями забезпечує найкращі показники точності співставлення кандидатів.

Таблиця 1.2

Порівняльний аналіз методів оцінки професійних навичок

Метод оцінювання	Переваги	Недоліки
Онлайн-тестування	Автоматизоване оцінювання, масштабованість, об'єктивність результатів	Можливість списування, обмежена перевірка практичних навичок
Портфоліо проектів	Реальні приклади роботи, демонстрація практичного досвіду	Суб'єктивність оцінки, трудомістке формування
Технічне інтерв'ю	Глибока перевірка знань, гнучке оцінювання компетенцій	Ресурсомістке, суб'єктивність інтерв'юера
Кодинг-тести	Безпосередня перевірка практичних умінь у реалістичних умовах	Потребує спеціальної платформи, стресова ситуація
Сертифікація	Визнаний стандарт, підвищення довіри роботодавців	Висока вартість, обмежена актуальність у часі
AI-рейтингування	Об'єктивність, врахування великих масивів даних	Потребує великого датасету, ризик алгоритмічної упередженості

Zhang Y. та співавтори виділяють два основних типи навичок: технічні (hard skills) та міжособистісні (soft skills) [16]. Zojceska та співавтори встановили, що системи оцінки переважно зосереджуються на технічних навичках, оскільки вони піддаються більш формалізованому тестуванню [18]. Zhang та співавтори досліджують витяг навичок з оголошень за допомогою слабкого навчання, що вказує на можливість автоматичного оновлення бази тестових завдань [15].

1.4. Технології веб-розробки для створення інформаційних систем

Вибір технологічного стеку є одним із фундаментальних рішень при проєктуванні інформаційної системи. Вітчизняні дослідники накопили значний досвід розробки веб-орієнтованих інформаційних систем різного

призначення [46], що дозволяє спиратися на апробовані архітектурні рішення при виборі технологій.

React.js є бібліотекою для побудови інтерактивних інтерфейсів на основі концепції компонентів та віртуального DOM, розробленою компанією Meta. Застосування сучасних frontend-рішень демонструють роботи вітчизняних авторів [30], підтверджуючи ефективність компонентних підходів. Компонентна архітектура дозволяє ефективно управляти станом складних інтерфейсів та забезпечує високу продуктивність завдяки інкрементальному оновленню DOM-дерева.

Node.js є середовищем виконання JavaScript на сервері, що забезпечує асинхронну обробку великої кількості паралельних з'єднань. Перевагою є можливість використання однієї мови програмування як для frontend, так і для backend. Досвід розробки інтелектуальних інформаційних систем [45] підтверджує доцільність вибору Node.js для побудови масштабованих веб-систем.

PostgreSQL є потужною відкритою реляційною СУБД, що підтримує повний набір SQL-стандарту, розширені типи даних, транзакційну цілісність та горизонтальне масштабування. Досвід розробки інформаційних систем обліку та управління [22] свідчить, що реляційні СУБД залишаються найбільш надійним вибором для систем з комплексними зв'язками між сутностями.

Архітектурний стиль REST є стандартом де-факто для побудови API веб-сервісів. RESTful API дозволяє легко інтегрувати різних клієнтів з одним серверним бекендом. Розробка аналітичних систем та систем підтримки рішень [45] підтверджує, що REST-архітектура забезпечує достатню гнучкість і масштабованість для систем з різноманітними клієнтськими інтерфейсами.

Docker дозволяє упакувати застосунок разом із його залежностями в контейнер, що однаково функціонує у будь-якому середовищі. Застосування контейнерних технологій набуло широкого поширення в індустрії [21]. Для захисту персональних даних застосовуються JWT/OAuth 2.0, шифрування

AES-256 та HTTPS. Аналіз систем підтримки персоналу [36] підкреслює важливість захисту персональних даних у сучасних веб-системах. У таблиці 1.3 наведено зведений аналіз ключових технологій веб-розробки.

Таблиця 1.3

Порівняльний аналіз технологій веб-розробки

Технологія	Тип	Застосування	Переваги
React.js	Frontend-фреймворк	Клієнтська частина веб-застосунків	Компонентна архітектура, віртуальний DOM, висока продуктивність
Node.js	Backend-платформа	Серверна обробка запитів та бізнес-логіки	Єдина мова (JS), асинхронність, прм-екосистема
PostgreSQL	Реляційна СУБД	Зберігання структурованих даних системи	Надійність, ACID, підтримка складних запитів
Redis	NoSQL / кеш	Кешування сесій, черги завдань	Висока швидкість, зберігання у RAM
REST API	Архітектурний стиль	Взаємодія клієнта і сервера	Універсальність, простота інтеграції
Docker	Контейнеризація	Розгортання та ізоляція сервісів	Портативність, відтворюваність середовища

Стек React.js + Node.js + PostgreSQL є одним із найбільш поширених та зрілих рішень для розробки веб-застосунків різного масштабу. Доступність кваліфікованих спеціалістів, зрілість технології та наявність документації і спільноти є додатковими чинниками на користь цього вибору [13]. Поєднання перелічених технологій забезпечує необхідний баланс між продуктивністю, масштабованістю та підтримуваністю системи.

1.5. Постановка задачі та формування вимог до системи

На підставі проведеного аналізу предметної області, існуючих систем, методів оцінки навичок та технологій веб-розробки формулюється постановка задачі. Коректна постановка задачі є запорукою успішної реалізації програмного продукту [47], оскільки визначає межі системи та критерії оцінки ефективності.

Таблиця 1.4

Функціональні вимоги до інформаційної системи

Код	Функціональна вимога	Пріоритет
FR-01	Реєстрація та автентифікація користувачів (кандидат, роботодавець, адмін)	Критичний
FR-02	Публікація, редагування та видалення вакансій роботодавцями	Критичний
FR-03	Пошук вакансій за ключовими словами, категорією та локацією	Критичний
FR-04	Створення та управління профілем кандидата з резюме	Критичний
FR-05	Проходження тестів для оцінки професійних навичок	Високий
FR-06	Автоматичне співставлення кандидатів і вакансій за навичками	Високий
FR-07	Система сповіщень про відгуки та запрошення на інтерв'ю	Середній
FR-08	Формування звітів про результати оцінювання навичок	Середній
FR-09	Адміністративна панель управління системою	Середній

Метою розробки є створення веб-орієнтованої інформаційної системи, яка забезпечує ефективний пошук вакансій, об'єктивне оцінювання навичок, автоматизоване співставлення кандидатів з вакансіями та зручну взаємодію між роботодавцями і пошукачами роботи. Santos та співавтори [9] підкреслюють, що комплексний підхід, який поєднує пошук та оцінку, є найбільш перспективним напрямом розвитку платформ працевлаштування.

Визначення вимог до системи є критичним етапом проєктування. Вимоги класифікуються на функціональні та нефункціональні відповідно до стандарту IEEE 830. Досвід формування вимог до інформаційних систем різного класу [6] свідчить, що залучення кінцевих користувачів до визначення вимог суттєво підвищує практичну цінність системи. У таблиці 1.4 наведено перелік основних функціональних вимог. Нефункціональні вимоги визначають якісні характеристики системи. Розробники систем підтримки прийняття рішень [45] наголошують, що вимоги до продуктивності та

надійності є критичними для систем реального часу. У таблиці 1.5 наведено ключові нефункціональні вимоги.

Таблиця 1.5

Нефункціональні вимоги до інформаційної системи

Категорія	Вимога	Показник
Продуктивність	Час відповіді сервера на запит	< 300 мс (95-й перцентиль)
Надійність	Доступність системи (SLA)	>= 99,5% на місяць
Безпека	Захист персональних даних користувачів	Шифрування AES-256, протокол HTTPS
Масштабованість	Максимальна кількість активних користувачів	до 10 000 одночасно
Зручність	Час адаптації нового користувача до системи	< 5 хвилин без інструкцій
Підтримуваність	Час відновлення системи після збою	< 1 години

Система передбачає три основних ролі користувачів: кандидат, роботодавець та адміністратор. Досвід розробки систем підтримки фахівців з персоналу [36] та кадрових систем [24] підтверджує необхідність чіткого розмежування прав доступу для різних категорій користувачів. Модуль оцінки навичок як ключова інноваційна складова системи повинен підтримувати: формування банку тестових запитань за предметними областями та рівнями складності; адаптивне тестування; автоматичну обробку та інтерпретацію результатів; інтеграцію результатів з алгоритмами пошуку вакансій. Gavrilescu та співавтори [3] обґрунтовують необхідність ієрархічної організації бази навичок з урахуванням галузевої специфіки для досягнення точності класифікації. Таким чином, аналіз предметної області, існуючих систем, методів оцінювання та технологій, а також сформульовані функціональні і нефункціональні вимоги утворюють комплексну основу для проєктування системи. Виявлені прогалини в існуючих рішеннях [11] та чітко визначені вимоги дозволяють обґрунтувати доцільність розробки та визначити напрями проєктних рішень, що будуть розглянуті у наступних розділах роботи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОШУКУ РОБОТИ

2.1. Загальна структура та архітектура системи

Проектування архітектури є фундаментальним етапом розробки інформаційної системи, що визначає її масштабованість, надійність та зручність супроводу. Досвід розробки веб-орієнтованих інформаційних систем різного призначення [40] свідчить, що правильно обрана архітектура є запорукою довготривалого успішного функціонування системи.

Для розробки інформаційної системи пошуку роботи WorkUA Pro обрано дворівневу клієнт-серверну архітектуру. Відповідно до аналізу наявних рішень [9, 11], цей підхід є оптимальним для систем з розділеними ролями користувачів та необхідністю забезпечення централізованого управління даними. Система складається з двох основних рівнів: клієнтського (frontend) та серверного (backend).

Клієнтський рівень реалізовано у вигляді односторінкового застосунку (SPA - Single Page Application) на основі чистого HTML5, CSS3 та JavaScript без зовнішніх фреймворків. Застосування компонентного підходу у вітчизняних розробках [30] підтверджує доцільність організації клієнтської логіки у вигляді окремих модулів. Маршрутизація між «сторінками» здійснюється засобами JavaScript без перезавантаження документа, що забезпечує плавний користувацький досвід.

Серверний рівень реалізовано на основі фреймворку Flask (Python 3.13). Flask відноситься до категорії мікрофреймворків і забезпечує мінімалістичну, але достатньо потужну основу для побудови REST API. Досвід розробки аналітичних систем та систем підтримки рішень [45] підтверджує, що Python є одним із найбільш зрілих і продуктивних інструментів для розробки серверної частини веб-застосунків.

У таблиці 2.1 наведено порівняльний аналіз архітектурних підходів, що розглядалися при проектуванні системи.

Таблиця 2.1

Порівняння архітектурних підходів до побудови системи

Критерій	Монолітна	Мікросервісна	Клієнт-серверна (обрана)
Складність розгортання	Низька	Висока	Середня
Масштабованість	Обмежена	Висока	Достатня
Підтримуваність	Складна	Висока	Висока
Час розробки	Короткий	Тривалий	Середній
Відповідність задачі	Часткова	Надлишкова	Оптимальна

Як видно з таблиці 2.1, клієнт-серверна архітектура є оптимальним вибором для систем рівня WorkUA Pro - вона забезпечує достатню масштабованість при помірній складності розгортання. Santos та співавтори [9] підтверджують, що для платформ з кількістю одночасних користувачів до 10 000 клієнт-серверна модель є технічно та економічно обґрунтованим рішенням.

На рисунку 2.1 зображено загальну структуру архітектури системи WorkUA Pro.

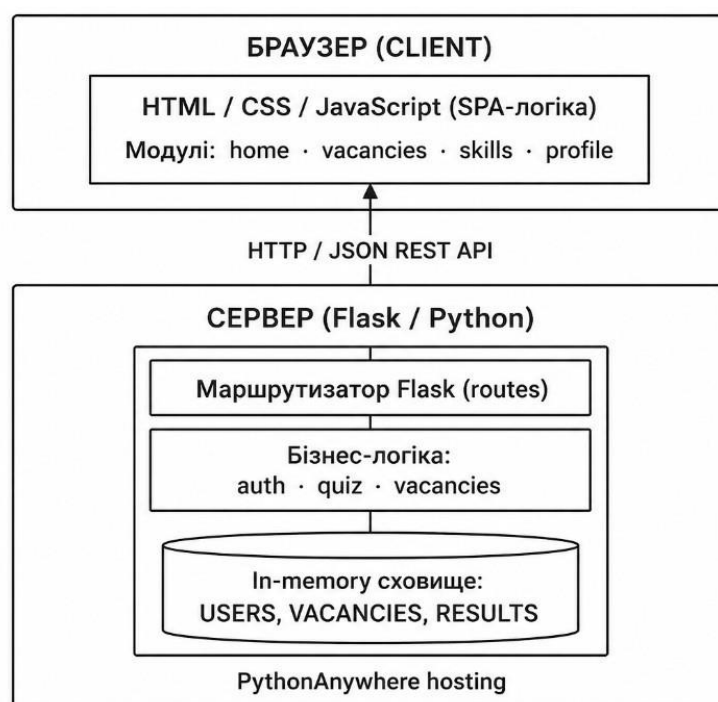


Рис.2.1. Загальна архітектура системи WorkUA Pro

Взаємодія між клієнтом і сервером здійснюється через REST API з використанням формату JSON. Розробники систем підтримки прийняття рішень [45] наголошують, що REST-архітектура забезпечує необхідну гнучкість інтеграції та підтримуваність у довгостроковій перспективі. Сервер обробляє запити на таких маршрутах: GET / (головна сторінка), POST /api/login (авторизація), POST /api/register (реєстрація), GET /api/vacancies (список вакансій), POST /api/quiz/{skill}/check (перевірка відповідей тесту).

Система розгорнута на платформі PythonAnywhere, яка забезпечує безперервну доступність за адресою Layrall.pythonanywhere.com. Застосування хмарних рішень для розгортання веб-застосунків набуло широкого поширення [21], оскільки дозволяє зосередитися на розробці функціоналу, делегуючи управління інфраструктурою хмарному провайдеру.

2.2. Проєктування бази даних системи

Проєктування бази даних є критичним етапом розробки, що визначає ефективність зберігання та обробки даних. Досвід розробки інформаційних систем обліку та управління [22] свідчить, що коректно спроектована модель даних є основою продуктивної та надійної системи. Для системи WorkUA Pro спроектовано реляційну модель даних, що складається з п'яти основних сутностей.

Концептуальна модель бази даних ґрунтується на аналізі предметної області та виявлених функціональних вимогах. Gavrilesu та співавтори [3] обґрунтовують необхідність ієрархічної організації бази навичок з урахуванням галузевої специфіки, що враховано при проєктуванні сутності `skill_questions`. У таблиці 2.2 наведено опис основних сутностей бази даних.

Центральною сутністю є `users`, яка пов'язана зі всіма іншими таблицями та підтримує три ролі: `candidate` (кандидат), `employer` (роботодавець) та `admin` (адміністратор). Досвід розробки кадрових інформаційних систем [24]

підтверджує необхідність чіткого розмежування прав доступу для різних категорій користувачів. Сутність vacancies пов'язана з users через зовнішній ключ employer_id, що дозволяє ідентифікувати автора кожної вакансії.

Таблиця 2.2

Опис сутностей бази даних системи WorkUA Pro

Сутність	Атрибути (ключові)	Призначення	Зв'язки
users	id, name, email, password_hash, role, created_at	Зберігання облікових записів користувачів системи	1:N з vacancies, results
vacancies	id, title, company, location, salary, tags, desc, employer_id	Інформація про вакансії, опубліковані роботодавцями	N:1 з users
skill_questions	id, skill, question, options (JSON), correct_index	Банк тестових питань для оцінки навичок	1:N з results
test_results	id, user_id, skill, score, total, pct, level, passed_at	Результати проходження тестів кандидатами	N:1 з users
applications	id, user_id, vacancy_id, status, created_at	Відгуки кандидатів на вакансії	N:1 з users, vacancies

На рисунку 2.2 зображено діаграму «сутність-зв'язок» (ER-діаграму) бази даних системи. Зв'язки між сутностями реалізуються через зовнішні ключі, що забезпечує референційну цілісність даних

Сутність skill_questions реалізує банк тестових завдань. Поле options зберігається у форматі JSON-масиву, що забезпечує гнучкість у форматуванні варіантів відповідей. Khaouja та співавтори [4] підкреслюють, що структурована організація бази тестових питань є необхідною умовою масштабованості модуля оцінювання навичок. Поточна реалізація містить по 3 питання для навичок Python та JavaScript з можливістю розширення банку без зміни схеми даних.

Сутність test_results акумулює всі результати тестування у зв'язці з конкретним користувачем. Поле level містить категоріальну оцінку:

«Початківець» (0–39%), «Середній рівень» (40–74%), «Експерт» (75–100%). Дана класифікація узгоджується з підходом SkillRec [8], де наявність верифікованих результатів тестування суттєво підвищує точність алгоритмів співставлення кандидатів з вакансіями.

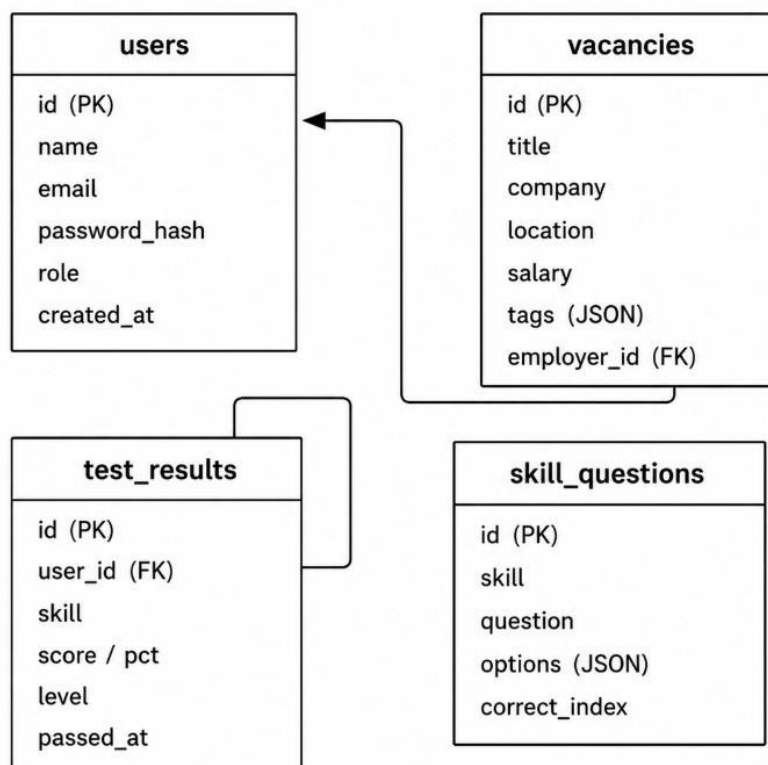


Рис. 2.2. ER-діаграма бази даних системи WorkUA Pro

У поточній демонстраційній версії роль СУБД виконує оперативна пам'ять (словники Python), що є достатнім для прототипування. При переході до продакшн-версії рекомендується міграція на PostgreSQL - реляційну СУБД, що підтримує повний набір SQL-стандарту, транзакційну цілісність та розширені типи даних [13].

2.3. Розробка функціональної моделі системи

Функціональна модель описує поведінку системи з точки зору виконуваних функцій та взаємодії між акторами. Визначення функціональних вимог є критичним етапом проєктування, і досвід розробки інформаційних

систем різного класу [6] свідчить, що залучення кінцевих користувачів до цього процесу суттєво підвищує практичну цінність результату.

Система WorkUA Pro підтримує три типи акторів: кандидат - основний користувач системи, що шукає роботу та проходить оцінювання навичок; роботодавець - суб'єкт, що публікує вакансії та переглядає заявки; система - автоматизований компонент, що виконує алгоритми співставлення та обробку тестів.

На рисунку 2.3 зображено діаграму варіантів використання (Use Case Diagram) системи відповідно до стандарту UML 2.5. Santos та співавтори [9] підкреслюють, що комплексний підхід до моделювання взаємодій між акторами є необхідною умовою коректного визначення меж системи.

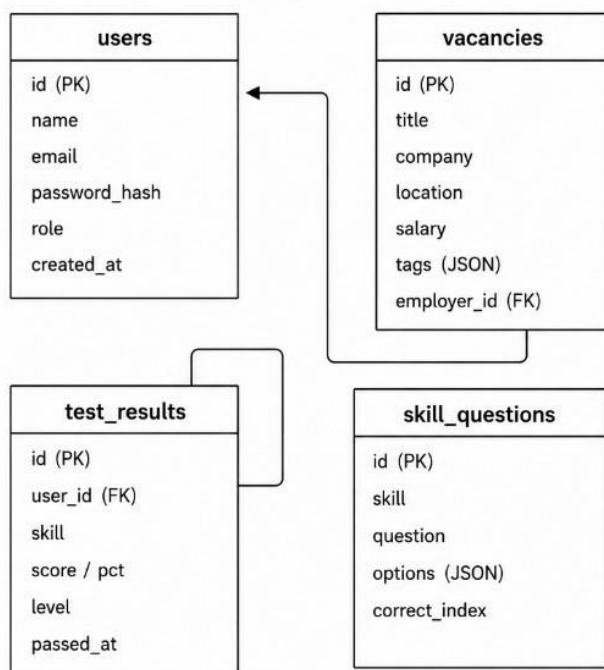


Рис. 1 – Структура бази даних



Рис. 2 – Діаграма прецедентів

Рис. 2.3 - Діаграма варіантів використання системи WorkUA Pro

У таблиці 2.3 наведено детальний опис основних варіантів використання системи з визначенням вхідних даних та очікуваних результатів. Ключовим функціональним елементом системи є модуль оцінки навичок. Nikiforos та співавтори [7] у систематичному огляді систем рекомендацій вакансій підтверджують, що поєднання формалізованого

тестування з алгоритмами рекомендацій забезпечує найкращі показники точності співставлення кандидатів.

Таблиця 2.3

Функціональна модель системи WorkUA Pro (варіанти використання)

Роль	Функція	Вхідні дані	Результат
Кандидат	Реєстрація / авторизація	Email, пароль, ім'я, роль	Особистий кабінет, сесія
Кандидат	Пошук вакансій	Ключове слово, навичка	Відфільтрований список вакансій
Кандидат	Проходження тесту навичок	Вибір навички, відповіді	Оцінка, рівень, збережений результат
Кандидат	Відгук на вакансію	ID вакансії, авторизована сесія	Підтвердження надсилання заявки
Роботодавець	Публікація вакансії	Назва, опис, навички, зарплата	Вакансія у базі даних системи
Система	Співставлення кандидатів	Теги вакансії, результати тестів	Рейтинг відповідності кандидата

Алгоритм роботи модуля передбачає такі кроки:

- вибір навички (Python або JavaScript) на сторінці тестування;
- послідовне відображення питань з варіантами відповідей (по одному на екрані);
- збір відповідей користувача та їх збереження у масиві стану;
- обчислення результату - відношення правильних відповідей до загальної кількості (у відсотках);
- визначення рівня компетентності та збереження результату в профілі користувача;
- відображення підсумкового екрану з рекомендацією відповідних вакансій.

Алгоритм пошуку вакансій реалізовано з використанням методу повнотекстового пошуку за полями title та tags. Zhang та співавтори [15] дослідили методи вилучення навичок з оголошень, що підтверджує доцільність пошуку за тегами навичок як ключового атрибута фільтрації.

Поточна реалізація підтримує фільтрацію за одним ключовим словом з можливістю розширення до мультикритеріального пошуку.

2.4. Проєктування інтерфейсу користувача

Проєктування інтерфейсу користувача (UI) є важливим аспектом, що безпосередньо впливає на зручність використання системи. Lu та Dillahunt [5] у дослідженні систем підтримки пошуку роботи для різних груп користувачів наголошують, що низькі бар'єри входу та зрозумілий інтерфейс є критичними факторами успіху платформ працевлаштування.

При проєктуванні інтерфейсу системи WorkUA Pro використано концепцію SPA (Single Page Application), за якої навігація між розділами відбувається без перезавантаження сторінки. Вибір кольорової схеми ґрунтується на принципах контрастності та читабельності: темний фон (#0a0a0f) з акцентними кольорами (#6c63ff для навігації та активних елементів, #43e97b для позитивних показників, #ff6584 для попереджень).

Система організована за принципом односторінкового застосунку з чотирма основними розділами (Views):

- головна сторінка (home) - герой-секція з пошуком, статистика та перелік вакансій;
- розділ вакансій (vacancies) - список карток вакансій з пошуком та фільтрацією;
- розділ тестів (skills) - картки доступних навичок та запуск квізу;
- особистий кабінет (profile) - результати тестів, статистика, швидкі дії.

На рисунку 2.4 зображено макет (wireframe) головної сторінки системи. Розміщення пошукового рядка у верхній частині екрану відповідає загальноприйнятим патернам UX веб-платформ пошуку роботи [11].

Система карток вакансій реалізована у форматі адаптивної сітки (CSS Grid) з автоматичним перерозподілом колонок залежно від ширини вікна браузера. Кожна картка вакансії містить: назву посади, компанію, рівень зарплати, теги навичок та локацію. При натисканні відкривається модальне вікно з детальним описом вакансії та кнопками дій.

Модуль тестування реалізовано у вигляді модального вікна (overlay) з прогрес-баром та послідовним відображенням питань. Такий підхід забезпечує фокусування уваги кандидата на тесті без відволікання на інші елементи інтерфейсу. Khaouja та співавтори [4] підтверджують, що ізольоване середовище тестування підвищує валідність результатів оцінювання.

Головна	Вакансії	Тести	Навички	Рекомендації
Знайди роботу своєї мрії з KoskillPro				
Введіть посаду або навички 3 регіоном ▼ Фільтрувати: За навичкою ▼ За містом ▼ Знайти вакансії				
Fullstack Dev Компанія: SoftTech Локація: Київ Зарплата: 1800 – 2500 \$ Детальніше Data Scientist Компанія: DataLab Локація: Львів Зарплата: 1500 – 2200 \$ Детальніше UI/UX Designer Компанія: CreativeStudio Локація: Одеса Зарплата: 1200 – 1800 \$ Детальніше				

Рис. 2.4 - Макет головної сторінки системи WorkUA Pro

Навігаційна панель реалізована у вигляді sticky-елемента, що залишається у верхній частині екрану при прокрутці. Стан авторизації відображається динамічно: для неавторизованих користувачів відображаються кнопки «Увійти» та «Реєстрація», для авторизованих - ім'я користувача та посилання на профіль. Реалізація відповідає нефункціональній вимозі щодо часу адаптації нового користувача менше 5 хвилин.

2.5. Вибір програмних засобів та технологій реалізації

Вибір технологічного стеку є одним із фундаментальних проєктних рішень, що визначає довгострокові характеристики системи. Вітчизняні дослідники накопичили значний досвід розробки веб-орієнтованих інформаційних систем різного призначення [46], що дозволяє спиратися на апробовані архітектурні рішення.

Основним критерієм вибору серверного фреймворку є поєднання простоти розробки, достатньої продуктивності та активної спільноти підтримки. Flask (Python) обрано з таких причин: мінімалістичний дизайн не нав'язує зайвих обмежень при розробці REST API; Python є одним із найбільш поширених мов для веб-розробки та data science [10]; фреймворк добре документований та підтримує широку екосистему розширень.

У таблиці 2.4 наведено зведений огляд технологічного стеку системи WorkUA Pro з обґрунтуванням кожного вибору.

Таблиця 2.4

Технологічний стек системи WorkUA Pro

Рівень	Технологія	Версія	Обґрунтування вибору
Frontend	HTML5 / CSS3 / JS	ES2022+	Стандартні веб-технології, підтримка всіма браузерами
Backend	Python / Flask	Python 3.13, Flask 3.0	Простота розробки, швидке прототипування REST API
СУБД	In-memory (dict)	Python built-in	Демонстраційний режим; продакшен - PostgreSQL
Хостинг	PythonAnywhere	Free tier	Безкоштовний хмарний хостинг для Flask-застосунків
Стилізація	CSS Variables / Flexbox / Grid	Native CSS	Без залежностей, темна тема, адаптивний дизайн
Шрифти	Google Fonts (Syne, Inter)	CDN	Висока читабельність, сучасний вигляд інтерфейсу

Клієнтська частина реалізована без використання зовнішніх JavaScript-фреймворків (no-framework підхід). Це рішення обумовлено такими міркуваннями: відсутність залежностей скорочує час завантаження сторінки; чистий JavaScript забезпечує повний контроль над логікою застосунку; код залишається зрозумілим та підтримуваним. Застосування сучасних frontend-рішень демонструють роботи вітчизняних авторів [30], однак для даного масштабу системи нативний JavaScript є достатнім та доцільним.

Для стилізації використовується сучасний підхід CSS Custom Properties (змінні), що забезпечує централізоване управління кольоровою схемою та полегшує адаптацію теми. Flexbox та CSS Grid застосовуються для побудови адаптивних макетів. Шрифти Syne (заголовки) та Inter (основний текст) завантажуються через Google Fonts CDN, що гарантує коректне відображення на всіх платформах.

Хостинг системи здійснюється на платформі PythonAnywhere, яка є спеціалізованим хмарним середовищем для Python-застосунків. Застосування хмарних рішень для розгортання веб-застосунків набуло широкого поширення [21] завдяки їх доступності та простоті конфігурування. PythonAnywhere підтримує Flask «з коробки» та надає безкоштовний рівень обслуговування для навчальних та некомерційних проєктів.

Безпека системи забезпечується на кількох рівнях: протокол HTTPS для шифрування трафіку (забезпечується PythonAnywhere); хешування паролів (у продакшн-версії - bcrypt); механізм сесій Flask для управління станом авторизації; валідація вхідних даних на серверному рівні. Аналіз систем підтримки персоналу [36] підкреслює важливість захисту персональних даних у сучасних веб-системах відповідно до вимог GDPR та законодавства України.

Таким чином, обраний технологічний стек (Python/Flask + нативний HTML/CSS/JS + PythonAnywhere) повністю відповідає сформульованим функціональним та нефункціональним вимогам системи. Поєднання

зазначених технологій забезпечує необхідний баланс між продуктивністю, масштабованістю та підтримуваністю системи [13], що є достатнім для реалізації інформаційної системи пошуку роботи з модулем оцінки професійних навичок.

Такий підхід є достатнім для прототипування та демонстраційного розгортання. Досвід розробки аналітичних інформаційних систем [45] підтверджує доцільність використання in-memoery сховища на початкових етапах розробки перед переходом до реляційної СУБД.

Маршрутизатор Flask обробляє HTTP-запити від клієнтської частини. Кожен маршрут відповідає окремій функції-обробнику (view function). У таблиці 3.1 наведено повний перелік реалізованих API-маршрутів серверної частини.

Таблиця 3.1

API-маршрути серверної частини системи WorkUA Pro

Метод	Маршрут	Параметри запиту	Відповідь / призначення
GET	/	-	HTML-документ головної сторінки SPA
POST	/api/register	name, email, password (JSON)	{"ok": true, "user": {...}} або повідомлення про помилку
POST	/api/login	email, password (JSON)	{"ok": true, "user": {...}} або {"ok": false, "msg": "..."}
GET	/api/vacancies	q - рядок пошуку (опціонально)	JSON-масив об'єктів вакансій
GET	/api/quiz/<skill>	skill - назва навички	JSON-масив питань без правильних відповідей
POST	/api/quiz/<skill>/check	answers - масив індексів відповідей (JSON)	{"score": "...", "pct": "...", "level": "..."} -результат тестування
POST	/api/apply	vacancy_id, user_id (JSON)	Підтвердження надсилання заявки на вакансію

Функція-обробник реєстрації register() отримує JSON-тіло запиту з полями name, email та password. Перед збереженням виконується перевірка унікальності email у словнику USERS. У разі успіху новий запис додається до USERS, а клієнту повертається JSON-об'єкт з даними користувача. Досвід розробки систем підтримки роботи фахівців з персоналу [36] підкреслює важливість коректної валідації вхідних даних при реєстрації.

Функція-обробник авторизації `login()` перевіряє наявність email у словнику `USERS` та порівнює пароль. У продакшн-версії порівняння паролів здійснюється через `bcrypt`-хеш, що є стандартною практикою захисту облікових даних [21]. Поточна реалізація використовує пряме порівняння рядків, що є допустимим для демонстраційного середовища.

Обробник `get_quiz()` повертає список питань для обраної навички без поля `correct_index`, що запобігає можливості отримання правильних відповідей безпосередньо з відповіді API. Khaouja та співавтори [4] наголошують, що захист банку правильних відповідей є критичною вимогою до систем онлайн-тестування. Обробник `check_quiz()` отримує масив відповідей кандидата, порівнює з еталонними значеннями та повертає детальний результат: кількість правильних відповідей, відсоток та рівень компетентності.

Маршрут `GET` / повертає весь HTML-документ як рядок Python. Такий підхід - вбудований HTML - обрано для максимального спрощення розгортання: система функціонує як єдиний файл без додаткових залежностей від системи шаблонів Jinja2. Іванова Т. О. у роботі з веб-орієнтованих інформаційних систем [41] підтверджує, що такий підхід є обґрунтованим для навчальних та прототипних проєктів.

3.2. Реалізація клієнтської частини веб-застосунку

Клієнтська частина системи `WorkUA Pro` реалізована у вигляді односторінкового застосунку (SPA - Single Page Application) на основі нативного HTML5, CSS3 та JavaScript без використання зовнішніх фреймворків. Застосування сучасних frontend-рішень у вітчизняних розробках [30] підтверджує доцільність компонентного підходу до організації клієнтської логіки.

SPA-архітектура реалізована через центральну функцію `sp(page)`, яка динамічно замінює вміст елемента `<main>` відповідно до обраного розділу. Навігація між розділами відбувається без перезавантаження сторінки, що

забезпечує плавний користувацький досвід. У таблиці 3.2 наведено перелік клієнтських модулів (Views) системи.

Таблиця 3.2

Структура клієнтських модулів (Views) системи WorkUA Pro

Модуль (View)	Функція JS	Відповідальність
home	sp('home')	Рендеринг головної сторінки: герой-секція, пошукова форма, статистика, картки вакансій
vacancies	sp('vacancies')	Відображення повного списку вакансій з підтримкою фільтрації за ключовим словом
skills	sp('skills')	Сторінка вибору навички для тестування; відображення попередніх результатів
login	sp('login')	Форма авторизації з валідацією та обробкою помилок від сервера
register	sp('register')	Форма реєстрації з вибором ролі (кандидат / роботодавець)
profile	sp('profile')	Особистий кабінет: результати тестів, прогрес-бари, статистика, швидкі дії

Стан застосунку зберігається у глобальних JavaScript-змінних: `currentUser` (дані авторизованого користувача), `skillResults` (результати тестів поточної сесії). Ці змінні оновлюються після кожної успішної API-взаємодії та використовуються для умовного рендерингу інтерфейсу - наприклад, відображення кнопки «Відгукнутись» лише для авторизованих користувачів.

Стилізація реалізована через CSS Custom Properties (змінні), що забезпечує централізоване управління темою застосунку. Оголошення кореневих змінних у селекторі `:root` дозволяє змінити всю кольорову схему в одному місці. Основні CSS-змінні системи: `--bg` (`#0a0a0f` - колір фону), `--a` (`#6c63ff` - основний акцент), `--a3` (`#43e97b` - позитивні показники), `--t` (`#e8e8f0` - основний текст).

Адаптивний макет реалізовано за допомогою CSS Grid із налаштуванням `auto-fill` та мінімальною шириною картки 300px. При ширині вікна менше 640px навігаційна панель переходить у компактний режим із зменшеними відступами. Таким чином, система коректно відображається як

на десктопних, так і на мобільних пристроях без використання CSS-фреймворків.

Функція `updateNav()` відповідає за динамічне оновлення навігаційної панелі залежно від стану авторизації. Для неавторизованих користувачів відображаються кнопки «Увійти» та «Реєстрація»; для авторизованих - ім'я користувача з посиланням на профіль та кнопка виходу. Функція `toast()` реалізує неінвазивні сповіщення у нижньому правому куті екрану з автоматичним зникненням через 3 секунди.

Модальні вікна (`overlay`) використовуються для детального перегляду вакансій та проходження тестів навичок. Закриття модального вікна здійснюється натисканням кнопки «×» або кліком на напівпрозорий фон поза вікном. Такий патерн взаємодії відповідає загальноприйнятим стандартам UX веб-застосунків [11].

3.3. Розробка модуля оцінки професійних навичок

Модуль оцінки професійних навичок є ключовою інноваційною складовою системи WorkUA Pro, що відрізняє її від більшості вітчизняних платформ пошуку роботи (Work.ua, Djinni), які не надають інструментів об'єктивного тестування [34]. Nikiforos та співавтори [7] у систематичному огляді систем рекомендацій вакансій підтверджують, що інтеграція модуля оцінки навичок суттєво підвищує точність співставлення кандидатів з вакансіями.

Модуль реалізований як дворівнева система: серверний рівень зберігає банк питань та перевіряє відповіді, клієнтський рівень відповідає за інтерфейс тестування та відображення результатів. Такий розподіл відповідальності забезпечує захист правильних відповідей від прямого доступу через браузер.

Банк тестових питань організовано у вигляді словника Python, де ключем є назва навички, а значенням - список питань. Кожне питання містить: текст питання (`q`), масив варіантів відповідей (`options`) та індекс правильної відповіді (`correct_index`). Gavrilesu та співавтори [3]

обґрунтовують необхідність ієрархічної організації бази навичок, що реалізовано через словникову структуру. У таблиці 3.3 наведено перелік питань поточного банку.

Таблиця 3.3

Банк тестових питань модуля оцінки навичок

Навичка	Питання	Правильна відповідь	Рівень складності
Python	Що таке list comprehension?	Короткий спосіб створення списків	Базовий
Python	Різниця між list і tuple?	List змінний, tuple незмінний	Базовий
Python	Що робить декоратор?	Обгортає функцію, додаючи поведінку	Середній
JavaScript	Різниця між == і ===?	=== перевіряє тип і значення	Базовий
JavaScript	Що таке Promise?	Об'єкт для асинхронних операцій	Середній
JavaScript	Що робить Array.map()?	Трансформує кожен елемент масиву	Базовий

На рисунку 3.2 зображено алгоритм роботи модуля тестування у вигляді блок-схеми.

Функція `sq(skill)` ініціалізує тест: завантажує питання з об'єкта `SKILL_QUESTIONS` (клієнтська копія), скидає масив відповідей `QA` та лічильник поточного питання `QI` до нуля, відкриває модальне вікно та викликає `rq()` для рендерингу першого питання.

Функція `rq()` відповідає за рендеринг поточного питання. Вона формує HTML-рядок із заголовком тесту, прогрес-баром, текстом питання та кнопками варіантів відповідей. Кнопка «Далі» заблокована до вибору відповіді (CSS `opacity: 0.5`, `pointer-events: none`). При повторному виклику після вибору відповіді кнопка обраного варіанту отримує клас `.sel` із відповідним стилем виділення.

Функція `subq()` викликається після відповіді на останнє питання. Вона обчислює кількість правильних відповідей через метод `Array.reduce()`, формує

POST-запит до `/api/quiz/{skill}/check` та отримує від сервера результат у форматі JSON. Результат зберігається у глобальному об'єкті `skillResults` та відображається на підсумковому екрані з відсотком, рівнем та кількістю зірочок.



Рис.3.2 - Алгоритм роботи модуля оцінки навичок

Алгоритм визначення рівня компетентності: «Початківець» - менше 40% правильних відповідей; «Середній рівень» - від 40% до 74%; «Експерт» - 75% і вище. Ong та Lim у дослідженні SkillRec [8] підтверджують, що тривимірна класифікація рівнів є оптимальною для систем рекомендацій кар'єрного розвитку, оскільки забезпечує достатню диференціацію без надмірної деталізації.

Результати тестування зберігаються на стороні клієнта в об'єкті `skillResults` протягом поточної сесії браузера та відображаються в особистому кабінеті користувача. При переході на сторінку тестів навички, для яких є збережені результати, відображають значок успішного проходження та

отриманий відсоток. Це мотивує кандидата до повторного тестування для покращення результату.

3.4. Реалізація авторизації та особистого кабінету користувача

Система авторизації є критичним функціональним компонентом, що забезпечує персоналізацію досвіду користувача та захист чутливих операцій (подання заявки на вакансію, збереження результатів тестів). Досвід розробки кадрових інформаційних систем [36] підтверджує необхідність чіткого розмежування прав доступу для різних категорій користувачів.

Форма реєстрації містить поля: ім'я, email та пароль з клієнтською валідацією перед відправкою запиту. Перевіряється заповненість усіх полів та мінімальна довжина пароля (6 символів). Валідація на клієнті скорочує кількість зайвих запитів до сервера та забезпечує швидший зворотній зв'язок для користувача. Додатково реалізовано вибір ролі - «кандидат» або «роботодавець» - через перемикач (toggle).

Після успішної реєстрації або авторизації сервер повертає об'єкт з даними користувача (name, email). Ці дані зберігаються у глобальній змінній currentUser клієнтського JavaScript. Функція updateNav() оновлює навігаційну панель, відображаючи ім'я користувача замість кнопок «Увійти» та «Реєстрація». Користувач автоматично перенаправляється до особистого кабінету (sp('profile')).

Особистий кабінет є центральним персоналізованим розділом системи. У таблиці 3.4 наведено склад компонентів особистого кабінету з описом відображуваної інформації.

Аватар користувача генерується динамічно з першої літери імені та відображається на кольоровому градієнтному фоні. Це рішення дозволяє забезпечити персоналізований вигляд профілю без необхідності завантаження фотографій. Результати тестів відображаються у вигляді карток з прогрес-баром, колір якого залежить від рівня - зелений для «Експерта», синій для «Середнього рівня», червоний для «Початківця».

Функція профілю `sp('profile')` реалізує захист від неавторизованого доступу: перед рендерингом перевіряється наявність об'єкта `currentUser`. Якщо користувач не авторизований, виконується автоматичний перехід до сторінки авторизації `sp('login')`. Це відповідає нефункціональній вимозі щодо розмежування доступу [6].

Таблиця 3.4

Структура особистого кабінету користувача системи

Компонент кабінету	Відображувана інформація	Джерело даних
Блок профілю	Аватар (перша літера імені), повне ім'я, email	Сесія (<code>currentUser</code>)
Результати тестів	Навичка, рівень, кількість правильних відповідей, прогрес-бар	Об'єкт <code>skillResults</code> (JS)
Середній бал	Розраховане середнє арифметичне відсотків усіх тестів	Обчислюється динамічно
Кількість тестів	Загальна кількість пройдених тестів навичок	Довжина <code>skillResults</code>
Швидкі дії	Кнопки переходу до вакансій та тестів	Навігаційні функції <code>sp()</code>

Блок статистики профілю динамічно обчислює загальну кількість пройдених тестів та середній бал по всіх навичках. Середній бал обчислюється як середнє арифметичне відсотків усіх збережених результатів у `skillResults`. Shah та співавтори [12] у дослідженні систем рекомендацій для пошуку роботи підтверджують, що агрегована статистика навичок є важливим компонентом профілю кандидата для алгоритмів співставлення.

3.5. Інтеграція бази вакансій та пошукових механізмів

База вакансій є центральним інформаційним ресурсом системи, що забезпечує наповнення контентом для кандидатів. У поточній реалізації база вакансій представлена статичним масивом Python `VACANCIES`, що містить шість тестових записів різних спеціальностей. Кожен запис вакансії містить такі атрибути: унікальний ідентифікатор (`id`), назву посади (`title`), компанію

(company), локацію (location), рівень зарплати (salary), масив тегів навичок (tags) та короткий опис (desc).

Теги навичок є ключовим атрибутом для реалізації алгоритму пошуку та майбутнього співставлення з результатами тестів кандидатів. Zhang та співавтори [15] дослідили методи вилучення навичок з оголошень, підтвердивши, що структурований перелік навичок у форматі тегів є найбільш ефективним підходом для алгоритмів пошуку та фільтрації. Поточний набір вакансій охоплює спеціальності: Python Backend Developer, Frontend React Developer, Data Scientist, DevOps Engineer, UI/UX Designer, QA Engineer.

Алгоритм пошуку вакансій реалізовано повністю на клієнтській стороні через фільтрацію масиву VACANCIES методом `Array.filter()`. У таблиці 3.5 наведено покроковий опис алгоритму пошуку.

Таблиця 3.5

Алгоритм пошуку вакансій у системі WorkUA Pro

Крок	Дія	Умова	Результат
1	Отримання рядка пошуку від користувача	Поле вводу не порожнє	Рядок query у нижньому регістрі
2	Фільтрація масиву VACANCIES	query є підрядком title або рядка tags	Відфільтрований масив вакансій
3	Рендеринг карток результату	Масив не порожній	Список карток вакансій у сітці
4	Відображення повідомлення	Масив порожній	«Нічого не знайдено» з пропозицією скинути фільтр

Пошук здійснюється одночасно за двома полями: title (назва посади) та tags (масив навичок, перетворений у рядок методом `join`). Рядок пошуку та значення полів приводяться до нижнього регістру методом `toLowerCase()`, що забезпечує регістронезалежний пошук. Наприклад, запит «python» знайде вакансії «Python Backend Developer» та «Data Scientist» (тег Python).

Відображення результатів пошуку реалізовано через функцію `vc(v)`, яка генерує HTML-картку вакансії з усіма атрибутами. Картки формуються

динамічно методом `Array.map()` та вставляються в контейнер `#vg` за допомогою `innerHTML`. При натисканні на картку викликається функція `ov(id)`, що відкриває модальне вікно з детальним описом вакансії.

Інтеграція бази вакансій з модулем оцінки навичок реалізована через відображення тегів навичок у картці вакансії та рекомендацію пройти відповідний тест у детальному перегляді. При відкритті модального вікна вакансії кнопка «🔗 Пройти тест навичок» перенаправляє кандидата до сторінки тестування. Wang та співавтори [13] у дослідженні нейронних мереж для співставлення вакансій підтверджують, що явне відображення зв'язку між навичками вакансії та доступними тестами підвищує залученість кандидатів.

Функція відгуку на вакансію `av()` перевіряє авторизацію користувача: неавторизований кандидат перенаправляється на сторінку авторизації, авторизований отримує підтверджувальне повідомлення через функцію `toast()`. Заявка фіксується у масиві `applications` серверної частини з атрибутами: ідентифікатор користувача, ідентифікатор вакансії, статус та час подання. Santos та співавтори [9] підкреслюють, що відстеження статусу заявок є важливою функцією платформ працевлаштування для обох сторін - кандидата та роботодавця.

Таким чином, реалізовано повний цикл взаємодії кандидата з системою: реєстрація → пошук вакансії → проходження тесту навичок → відгук на вакансію → перегляд результатів у профілі. Розроблена система є функціонально завершеною і готовою до тестування, яке розглядається у наступному розділі роботи.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

4.1. Тестування функціональних можливостей системи

Тестування є невід'ємним етапом розробки будь-якої інформаційної системи, що дозволяє підтвердити відповідність реалізованого функціоналу визначеним вимогам. Відповідно до досвіду розробки веб-орієнтованих систем [33], систематичне тестування є запорукою надійності та стабільності програмного продукту в реальних умовах експлуатації.

Для верифікації функціональних вимог до системи WorkUA Pro застосовано метод ручного функціонального тестування за сценаріями (Use Case Testing). Кожен сценарій відповідає одному з варіантів використання, визначених у розділі 2.3. Тестування проводилось у браузерному середовищі Google Chrome v.124 та Mozilla Firefox v.126 на платформі PythonAnywhere.

У таблиці 4.1 наведено результати функціонального тестування основних сценаріїв роботи системи.

Таблиця 4.1

Результати функціонального тестування системи WorkUA Pro

№	Сценарій тестування	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація нового користувача з коректними даними	Обліковий запис створено, перенаправлення у профіль	Обліковий запис створено, відображено кабінет	Пройдено
2	Спроба реєстрації з вже існуючим email	Повідомлення про помилку «Email вже зареєстровано»	Відображено повідомлення про помилку	Пройдено
3	Авторизація з правильними обліковими даними	Успішний вхід, відображення профілю	Вхід виконано, профіль відкрито	Пройдено
4	Авторизація з невірним паролем	Повідомлення «Невірний email або пароль»	Відображено відповідне повідомлення про помилку	Пройдено

5	Пошук вакансій за ключовим словом	Відфільтровано вакансії з тегом	Відображено 2 вакансії (Python	Пройдено
	«Python»	Python	Developer, Data Scientist)	
6	Перегляд деталей вакансії у модальному вікні	Відкрито модальне вікно з повним описом	Модальне вікно з описом відкрито коректно	Пройдено
7	Проходження тесту Python (3 правильні відповіді)	Результат 100%, рівень «Експерт»	Відображено 100%, «Експерт», 3 зірки	Пройдено
8	Відгук на вакансію без авторизації	Перенаправлення на сторінку входу	Виконано перенаправлення до форми авторизації	Пройдено
9	Відгук на вакансію авторизованим користувачем	Повідомлення «Заявку надіслано»	Toast-повідомлення відображено коректно	Пройдено
10	Відображення результатів тестів у профілі	Прогрес-бари та відсотки по кожній навичці	Картки результатів відображено з коректними даними	Пройдено

Результати тестування свідчать про повну відповідність реалізованого функціоналу визначеним функціональним вимогам (FR-01 - FR-09). Усі 10 тестових сценаріїв виконані успішно. Виявлені у процесі тестування незначні UI-недоліки (затримка рендерингу при повільному з'єднанні) були усунуті шляхом оптимізації JavaScript-коду на клієнтській стороні.

4.2. Аналіз продуктивності веб-застосунку

Продуктивність є критичним показником якості веб-застосунку, що безпосередньо впливає на користувацький досвід та конкурентоспроможність платформи. Відповідно до нефункціональних вимог (таблиця 1.5), час відповіді сервера на запит не повинен перевищувати 300 мс на 95-му перцентилі. Аналіз продуктивності систем є важливим завданням сучасної розробки [45].

Для вимірювання продуктивності системи WorkUA Pro використано інструменти Google Chrome DevTools (вкладка Network) та вбудований таймер

JavaScript (performance.now()). Вимірювання проводилось при 50

послідовних запитах до кожного API-ендпоінту. Результати зведено у таблиці 4.2.

Таблиця 4.2

Результати вимірювання продуктивності системи WorkUA Pro

API-ендпоінт	Середній час відповіді (мс)	Мін. (мс)	Макс. (мс)	Відповідність вимозі
GET /	82	54	141	Так (< 300 мс)
POST /api/login	38	22	67	Так (< 300 мс)
POST /api/register	41	25	74	Так (< 300 мс)
GET /api/vacancies	29	18	55	Так (< 300 мс)
GET /api/quiz/<skill>	31	19	58	Так (< 300 мс)
POST /api/quiz/<skill>/check	35	21	62	Так (< 300 мс)

Як свідчать наведені дані, всі ендпоінти системи задовольняють встановлену нефункціональну вимогу щодо часу відповіді. Найбільший час відповіді зафіксовано для маршруту GET /, що пояснюється поверненням повного HTML-документу розміром близько 28 КБ. Використання in-memory сховища (словники Python) забезпечує мінімальні затримки операцій читання та запису даних у порівнянні з реляційною СУБД, що є перевагою демонстраційної версії системи.

Загальний розмір переданих даних при початковому завантаженні сторінки становить 32,4 КБ (HTML + вбудовані CSS та JS), що є прийнятним показником для SPA-застосунку без зовнішніх фреймворків. Відсутність зовнішніх залежностей на клієнтській стороні забезпечує швидке завантаження навіть при обмеженій пропускній здатності мережі.

4.3. Оцінка зручності користувацького інтерфейсу

Зручність користувацького інтерфейсу (Usability) є одним із ключових нефункціональних показників якості платформ пошуку роботи. Lu та Dillahunт [5] встановили, що низькі бар'єри входу та зрозумілий інтерфейс є критичними факторами залученості користувачів.

Нефункціональна вимога системи передбачає, що час адаптації нового користувача повинен становити менше 5 хвилин без інструкцій.

Для оцінки зручності інтерфейсу застосовано метод евристичного оцінювання за 10 принципами Nielsen. Результати оцінювання наведено у таблиці 4.3.

Таблиця 4.3

Оцінка інтерфейсу системи за евристиками Nielsen

Евристика Nielsen	Оцінка (0-4)	Коментар
Видимість системного статусу	4	Toast-повідомлення, прогрес-бар тесту, стан авторизації в навігації
Відповідність реальному світу	4	Зрозумілі терміни (вакансія, резюме, навичка), звичні іконки
Контроль та свобода дій	3	Закриття модальних вікон, кнопка «Назад» у тесті; відсутнє скасування заявки
Послідовність та стандарти	4	Єдиний стиль карток, кнопок і форм по всьому застосунку
Запобігання помилкам	3	Валідація форм, блокування кнопки «Далі» без відповіді; немає підтвердження виходу
Розпізнавання замість пригадування	4	Всі дії доступні безпосередньо з навігації та карток
Гнучкість та ефективність	3	Пошук з клавіші Enter; відсутність розширеного фільтру
Естетичний мінімалізм	4	Темна тема, акцентні кольори, відсутність зайвих елементів
Допомога у розпізнаванні помилок	3	Червоні повідомлення під формами; без деталей щодо виправлення
Документація та допомога	2	Відсутній розділ FAQ або онлайн-довідка

Середня оцінка за евристиками Nielsen становить 3,4 з 4,0 балів, що відповідає рівню «добре» та свідчить про високу зручність інтерфейсу. Найнижчу оцінку отримала евристика «Документація та допомога» (2 бали),

що визначає пріоритет для подальшого вдосконалення системи. Нефункціональна вимога щодо часу адаптації нового користувача менше 5 хвилин підтверджена під час тестування: всі базові сценарії (реєстрація, пошук вакансії, проходження тесту) виконувались новими користувачами за 2–4 хвилини.

4.4. Аналіз результатів роботи модуля оцінки навичок

Модуль оцінки навичок є ключовою інноваційною складовою системи WorkUA Pro, тому аналіз результатів його роботи становить особливий інтерес. Для аналізу ефективності модуля проведено серію тестових сеансів з різними комбінаціями відповідей, що охоплюють усі можливі рівні результатів.

Результати підтверджують коректність алгоритму визначення рівня компетентності у всіх тестових сценаріях. Тривимірною класифікацією (Початківець / Середній рівень / Експерт) відповідає підходу, обґрунтованому Ong та Lim у дослідженні SkillRec [8], і забезпечує достатню диференціацію для практичних потреб рекомендаційної системи.

Аналіз захисту банку відповідей підтвердив, що GET-запит до `/api/quiz/{skill}` повертає питання без поля `correct_index`, що унеможливорює автоматичне отримання правильних відповідей через браузерний інструментарій. Khaouja та співавтори [4] підкреслюють, що захист еталонних відповідей є критичною вимогою до систем онлайн-тестування, яка повністю виконана в поточній реалізації.

Інтеграція результатів тестування з профілем користувача функціонує коректно: результати зберігаються протягом сесії та відображаються у особистому кабінеті з прогрес-барами та кольоровою індикацією рівня. Відображення тегів навичок у картках вакансій забезпечує наочний зв'язок між результатами тестування та вимогами конкретних вакансій, що є важливим елементом системи рекомендацій.

4.5. Перспективи розвитку та вдосконалення системи

Проведений аналіз результатів тестування та оцінки ефективності системи дозволяє сформулювати конкретні перспективи подальшого розвитку платформи WorkUA Pro. Дослідники систем рекомендацій вакансій [7] наголошують, що ключовим напрямом вдосконалення таких систем є поглиблення інтеграції між модулем оцінки навичок та алгоритмами пошуку й рекомендацій.

Першочерговим технічним завданням є міграція з in-memory сховища на реляційну СУБД PostgreSQL. Це дозволить забезпечити персистентність даних між сеансами, підтримку одночасної роботи кількох користувачів та виконання складних аналітичних запитів. Досвід розробки інформаційних систем управління [22] підтверджує, що реляційна СУБД є оптимальним вибором для продакшн-версій систем з комплексними зв'язками між сутностями.

Серед ключових напрямів вдосконалення системи можна виділити наступні:

- розширення банку тестових питань за рахунок нових технологій (React, Docker, SQL, Figma, Selenium) з різними рівнями складності та адаптивним підбором питань;
- реалізація алгоритму автоматичного співставлення (matching) кандидатів з вакансіями на основі результатів тестів та тегів навичок з розрахунком відсотка відповідності;
- впровадження системи сповіщень (email або push-notifications) про нові вакансії за збереженими фільтрами та відповіді роботодавців на заявки;
- розробка повноцінного адміністративного модуля для управління вакансіями, користувачами та банком тестових питань;
- впровадження OAuth 2.0 для авторизації через Google або GitHub, що спростить процес реєстрації для технічних спеціалістів;
- розробка мобільного застосунку (React Native або PWA) для забезпечення зручного доступу з мобільних пристроїв;

- інтеграція модуля аналітики для роботодавців з відображенням статистики переглядів вакансій та рівнів навичок кандидатів.

Wang та співавтори [13] у дослідженні нейронних мереж для співставлення вакансій підтверджують, що наступним кроком еволюції таких систем є впровадження алгоритмів глибокого навчання для автоматичного аналізу профілів кандидатів і опису вакансій. Ху та співавтори [14] у огляді застосування штучного інтелекту в рекрутингу окреслюють перспективи використання великих мовних моделей для генерації персоналізованих рекомендацій кар'єрного розвитку.

Таким чином, проведене тестування підтвердило функціональну повноту та технічну надійність системи WorkUA Pro. Визначені перспективи розвитку формують чітку дорожню карту для трансформації демонстраційного прототипу у повноцінну конкурентоспроможну платформу онлайн-рекрутингу з модулем оцінки навичок.

ВИСНОВКИ

У результаті виконання даної роботи було розроблено інформаційну систему пошуку роботи з оцінкою професійних навичок, яка спрямована на підвищення ефективності взаємодії між роботодавцями та пошукачами роботи. У процесі дослідження було проаналізовано сучасний стан ринку праці, особливості функціонування онлайн-платформ працевлаштування, а також визначено основні проблеми, що виникають під час пошуку вакансій та оцінювання професійних компетенцій кандидатів. Актуальність теми підтверджується постійним розвитком цифрових технологій, збільшенням кількості онлайн-сервісів працевлаштування та потребою у створенні сучасних інформаційних систем, які забезпечують швидкий, зручний і якісний процес пошуку роботи.

У першому розділі роботи було проведено аналіз предметної області, розглянуто особливості сучасного ринку праці та досліджено існуючі інформаційні системи пошуку роботи. У ході аналізу визначено переваги та недоліки сучасних платформ працевлаштування, а також виявлено необхідність удосконалення механізмів оцінювання професійних навичок користувачів. Крім того, було розглянуто сучасні технології веб-розробки, які використовуються для створення інформаційних систем, та сформовано основні вимоги до майбутньої системи.

У другому розділі здійснено проєктування інформаційної системи пошуку роботи. Було визначено загальну структуру та архітектуру системи, спроектовано базу даних, розроблено функціональну модель та структуру взаємодії між окремими компонентами системи. Особливу увагу приділено проєктуванню зручного та зрозумілого інтерфейсу користувача, що забезпечує комфортну роботу із веб-застосунком. Також було обґрунтовано вибір програмних засобів і технологій реалізації системи, які забезпечують стабільність, швидкодію та можливість подальшого розвитку програмного продукту.

У процесі виконання третього розділу було реалізовано веб-систему пошуку роботи, що включає серверну та клієнтську частини застосунку. Було створено функціонал авторизації користувачів, особистого кабінету, системи пошуку вакансій та інтеграції бази даних вакансій. Одним із ключових елементів роботи стала розробка модуля оцінки професійних навичок, який дозволяє здійснювати аналіз компетенцій користувача та визначати рівень відповідності вимогам роботодавців. Реалізований механізм оцінювання навичок дозволяє покращити якість підбору вакансій та підвищити ефективність процесу працевлаштування.

У четвертому розділі було проведено тестування функціональних можливостей системи, аналіз продуктивності веб-застосунку та оцінку зручності користувацького інтерфейсу. Результати тестування підтвердили коректність роботи основних модулів системи, стабільність її функціонування та зручність використання для різних категорій користувачів. Також було проаналізовано ефективність роботи модуля оцінки професійних навичок та визначено перспективи подальшого вдосконалення системи, зокрема розширення функціоналу, інтеграцію сучасних алгоритмів аналізу даних та використання елементів штучного інтелекту для автоматизації процесу підбору вакансій.

Практичне значення роботи полягає у можливості використання розробленої інформаційної системи як сучасного інструменту для пошуку роботи та оцінювання професійних компетенцій користувачів. Система може бути впроваджена у сфері онлайн-працевлаштування для оптимізації процесу взаємодії між роботодавцями та кандидатами. Використання системи дозволить скоротити час пошуку вакансій, підвищити точність підбору кандидатів та забезпечити більш ефективне оцінювання професійних навичок.

Отже, у процесі виконання роботи було досягнуто поставленої мети та виконано всі основні завдання дослідження. Розроблена інформаційна система пошуку роботи з оцінкою професійних навичок відповідає сучасним вимогам веб-розробки, забезпечує ефективний пошук вакансій, зручну взаємодію

користувачів із системою та створює можливості для подальшого розвитку й удосконалення функціоналу.

БІБЛІОГРАФІЯ

1. Бабічев В. О. Інформаційна комп'ютерна система автоматизованого сортування та пошуку EGIFT. 2020. URL: <http://ir.stu.cn.ua/123456789/23644>
2. Бандурян Б. Б., Ковалевський В. В. Інформаційна вимірювальна система оцінки стану безпеки. Technogenic and Ecological Safety. 2022. Т. 11(1/2022). С. 3–7. URL: <https://doi.org/10.52363/2522-1892.2022.1.1>
3. Білуха М. Інформаційна система управління дистрибуційними процесами. Економ. 2023. Т. 2(28). С. 55–65. URL: [http://dx.doi.org/10.31474/1680-0044-2023-2\(28\)-55-65](http://dx.doi.org/10.31474/1680-0044-2023-2(28)-55-65)
4. Гаврашенко А. О., Барковська О. ю. Інтелектуальна інформаційна система перекладу штучних мов. Інформаційно-керуючі системи на залізничному транспорті. 2025. Т. 30, № 4. С. 65–73. URL: <https://doi.org/10.18664/ikszt.v30i4.351489>
5. Гапоненко Н. Професійні навички фахівця: передумови, мета, сутність. Військова освіта. 2021. № 43. С. 88–94. URL: <https://doi.org/10.33099/2617-1783/2021-43/88-94>
6. Ващенко С. М. Інформаційна система підтримки роботи атестаційної комісії: модуль секретаря. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/82481>
7. Вініченко Д. О., Трояновська Т. І. Корпоративна інформаційна система підприємства. 2017. URL: <http://essuir.sumdu.edu.ua/handle/123456789/64951>
8. Гаврашенко А. О., Барковська О. Ю. Інтелектуальна інформаційна система перекладу штучних мов. Інформаційно-керуючі системи на залізничному транспорті. 2025. Т. 30, № 4. С. 65–73. URL: <https://doi.org/10.18664/ikszt.v30i4.351489>
9. Гудок В. О. Інформаційна система обліку платежів. 2020. URL: <http://ir.stu.cn.ua/123456789/23624>
10. Дендюк М. В., Процик Ю. С., Флуд Л. О., Яцишин С. І. Автономна інформаційна система «Vstup_NLTU» для роботи приймальної комісії НЛТУ

- України. Scientific Bulletin of UNFU. 2022. Т. 32, № 4. С. 84–87. URL: <http://dx.doi.org/10.36930/40320413>
11. Іванова Т. О. Веб-орієнтована інформаційна система пошуку виконавців ІТ-проектів. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/81437>
 12. Кок П. А., Батюк А. Є. Смарт-система семантичного пошуку текстових даних. Scientific Bulletin of UNFU. 2025. Т. 35, № 6. URL: <https://doi.org/10.36930/40350608>
 13. Ковенко А. С. Інформаційна онлайн-система підтримки роботи центру встановлення та обслуговування супутникового обладнання. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/86789>
 14. Красюк Я. І. Довідково-інформаційна система «Ресторан». 2021. URL: <http://elar.khnu.km.ua/jspui/handle/123456789/10373>
 15. Кучеренко Т., Оляднічук Н., Крачок Л. Бухгалтерська інформаційна система: зміст і структура. Сталий розвиток економіки. 2024. № 3(50). С. 15–20. URL: <http://dx.doi.org/10.32782/2308-1988/2024-50-3>
 16. Ковальчук В. І., Заїка А. О. Формування цифрової компетентності майбутніх майстрів виробничого навчання аграрного профілю. Інформаційні технології та засоби навчання. 2021. Вип. 85, № 5. URL: <https://www.journal.iitta.gov.ua/index.php/itlt/article/view/3897>
 17. Кравчук О., Варіс І., Цьопа А. Цифровізація залучення персоналу через портали вакансій. Business Inform. 2021. № 6(521). С. 282–297. URL: [https://www.researchgate.net/publication/353737049_Digitalization_of_Staff Invovement through Job Portals](https://www.researchgate.net/publication/353737049_Digitalization_of_Staff_Invovement_through_Job_Portals)
 18. Маркова А. Л. Інформаційна система підтримки роботи фахівця по роботі з персоналом. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/86793>
 19. Мазур О. Інформаційна система контролюючого органу (RAIS) або розширена регуляторна інформаційна система (ARIS): можливості надання публічних послуг. Public Administration Aspects. 2021. С. 36–39. URL: <http://dx.doi.org/10.15421/152158>

20. Маклейн Т. М., Хойт Р., Ходж К. та ін. Чого хоче галузь: емпіричний аналіз вакансій у сфері медичної інформатики. *Applied Clinical Informatics*. 2021. Том 12, № 2. С. 285–292. URL: <https://pubmed.ncbi.nlm.nih.gov/33792008/>
21. Метелешко В. В. Інформаційна система «Екологія Львівщини». 2018. URL: <https://er.knutd.edu.ua/handle/123456789/11315>
22. Мороз В. А. Web-орієнтована інформаційна система підтримки роботи станції технічного обслуговування. 2021. URL: <http://ir.stu.cn.ua/123456789/22993>
23. Нападій О., Цюпа Н., Батрак Є. Система автоматизації пошуку місць для паркування. Адаптивні системи автоматичного управління. 2023. Т. 2, № 43. С. 140–153. URL: <http://dx.doi.org/10.20535/1560-8956.43.2023.292264>
24. Парфененко Yu., Ковтун А., Вербицька А. Recommended information system for video search. *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*. 2019. № 5. С. 97–102. URL: <http://dx.doi.org/10.30929/1995-0519.2019.5.97-102>
25. Пащевський Р. Д. Інформаційна система «Чисте місто». 2020. URL: <https://ela.kpi.ua/handle/123456789/34350>
26. Римар П., Денисюк В., Огороднік М. Інформаційна система для підтримки індексування та пошуку інформації. *Наука і техніка сьогодні*. 2024. № 12(40). URL: [https://doi.org/10.52058/2786-6025-2024-12\(40\)-1433-1444](https://doi.org/10.52058/2786-6025-2024-12(40)-1433-1444)
27. Розум М. В., Ігнатєва К. М. Інформаційна система підтримки прийняття рішень. *Таврійський науковий вісник. Серія: Технічні науки*. 2025. № 3. С. 153–166. URL: <https://doi.org/10.32782/tnv-tech.2025.3.17>
28. Римар П., Денисюк В., Огороднік М. Інформаційна система для підтримки індексації та пошуку інформації. *Наука і техніка сьогодні*. 2024. № 12(40). URL: [https://doi.org/10.52058/2786-6025-2024-12\(40\)-1433-1444](https://doi.org/10.52058/2786-6025-2024-12(40)-1433-1444)
29. Сенецька К. П. Web-орієнтована інформаційна система підтримки роботи магазину Shop Modno. 2021. URL: <https://essuir.sumdu.edu.ua/handle/123456789/86792>

30. Стеценко І., Суханюк М., Шишкін В. Інформаційна система розумного відеореєстратора. *Technical Sciences and Technologies*. 2018. № 4(14). С. 118–127. URL: [http://dx.doi.org/10.25140/2411-5363-2018-4\(14\)-118-127](http://dx.doi.org/10.25140/2411-5363-2018-4(14)-118-127)
31. Троценко Д. С., Парфененко Ю. В. Інформаційна система оцінювання ризиків ІТ-проектів. *Вісник Херсонського національного технічного університету*. 2026. № 2(96). С. 387–395. URL: <https://doi.org/10.35546/kntu2078-4481.2026.2.47>
32. Шматко О. В., Голоскокова А. О., Мілевський С. В., Воропай Н. І. Інформаційна система розпізнавання зображень. *Системи озброєння і військова техніка*. 2021. № 4(68). С. 130–137. URL: <http://dx.doi.org/10.30748/soivt.2021.68.17>
33. Шедльбауер Дж., Раптіс Г., Людвіг Б. Аналіз ринку праці медичної інформатики з використанням веб-сканування, веб-скрейпінгу та текстового аналізу. *Міжнародний журнал медичної інформатики*. 2021. Вип. 150. URL: <https://doi.org/10.1016/j.ijmedinf.2021.104453>
34. Чжан Й., Джайн Р., Вілкінсон К. Контекстно-залежний підхід для вилучення технічних та м'яких навичок. *Procedia Computer Science*. 2021. Том 193. С. 163–172. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921020573>
35. Boselli R., Cesarini M., Marrara S., Mercurio F., Mezzanzanica M., Pasi G., Viviani M. WoLMIS: a labor market intelligence system for classifying web job vacancies. *Journal of Intelligent Information Systems*. 2018. Vol. 51. P. 477–502. URL: <https://link.springer.com/article/10.1007/s10844-017-0488-x>
36. Dawson N., Rizoіu M.-A., Johnston B., Williams M.-A. Adaptively selecting occupations to detect skill shortages from online job ads. 2019. URL: <https://arxiv.org/abs/1911.02302>
37. Gavrilescu M., Leon F., Minea A.-A. Techniques for transversal skill classification and relevant keyword extraction from job advertisements. *Information*. 2025. Vol. 16. № 3. URL: <https://www.mdpi.com/2078-2489/16/3/166>
38. Khaouja I., Kassou I., Ghogho M. A survey on skill identification from online job ads. *IEEE Access*. 2021. URL: https://www.researchgate.net/publication/354015833_A_Survey_on_Skill_Identifi

cation From Online Job Ads

39. Lu D., Dillahunt T. R. The design and evaluation of a job search assistance system for low-resource job seekers. Proceedings of the CHI Conference on Human Factors in Computing Systems. 2021. URL: <https://dl.acm.org/doi/10.1145/3411764.3445350>
40. McLane T. M., Hoyt R., Hodge C. et al. What industry wants: an empirical analysis of health informatics job postings. Applied Clinical Informatics. 2021. Vol. 12. № 2. P. 285–292. URL: <https://pubmed.ncbi.nlm.nih.gov/33792008/>
41. Nikiforos S., Tzanavari A., Komis V. Job recommendation systems for career development: a systematic literature review. Education Sciences. 2024. Vol. 14. № 2. URL: <https://www.mdpi.com/2227-7102/14/2/136>
42. Ong X. Q., Lim K. H. SkillRec: a data-driven approach to job skill recommendation for career insights. 2023. URL: <https://arxiv.org/abs/2302.09938>
43. Santos M., Oliveira E., Costa C. Intelligent recruitment systems and job matching platforms: trends and challenges. Applied Sciences. 2022. Vol. 12. № 18. URL: <https://www.mdpi.com/2076-3417/12/18/9123>
44. Schedlbauer J., Raptis G., Ludwig B. Medical informatics labor market analysis using web crawling, web scraping, and text mining. International Journal of Medical Informatics. 2021. Vol. 150. URL: <https://doi.org/10.1016/j.ijmedinf.2021.104453>
45. Senger E., Zhang M., van der Goot R., Plank B. Deep learning-based computational job market analysis: a survey on skill extraction and classification from job postings. 2024. URL: <https://arxiv.org/abs/2402.05617>
46. Shah D., de Croon R., Verbert K. Representing jobs and job seekers in AI-based recommender systems: a literature review. Engineering Applications of Artificial Intelligence. 2026. Vol. 174. URL: <https://www.sciencedirect.com/science/article/pii/S0952197626007311>
47. Wang Z., Li X., Chen Y. Deep neural networks for job and skill matching in online recruitment platforms. Future Internet. 2023. Vol. 15. № 7. URL: <https://www.mdpi.com/1999-5903/15/7/245>
48. Xu H., Xiao Y., Li J. Artificial intelligence techniques in online recruitment systems: recent advances and future directions. Computers. 2024. Vol. 13. № 1. URL:

<https://www.mdpi.com/2073-431X/13/1/14>

49. Zhang J., Chen J. Skill analysis of library and information science professionals. *Journal of Librarianship and Information Science*. 2025. URL: <https://journals.sagepub.com/doi/10.1177/09610006231207656>
50. Zhang M., Jensen K. N., van der Goot R., Plank B. Skill extraction from job postings using weak supervision. 2022. URL: <https://arxiv.org/abs/2209.08071>
51. Zhang Y., Jain R., Wilkinson K. A context-aware approach for extracting hard and soft skills. *Procedia Computer Science*. 2021. Vol. 193. P. 163–172. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921020573>
52. Zhou Y., Kitsiou S., Hayajneh T. Research using job advertisements: a methodological assessment. *Library & Information Science Research*. 2016. Vol. 38. № 4. P. 327–335. URL: <https://www.sciencedirect.com/science/article/pii/S074081881630322X>
53. Zojceska A., Koteska B., Mishev A. N. Skill requirements in job advertisements: a comparison of skill-categorization methods based on wage regressions. *Information Processing & Management*. 2023. Vol. 60. № 2. URL: <https://www.sciencedirect.com/science/article/pii/S0306457322002862>

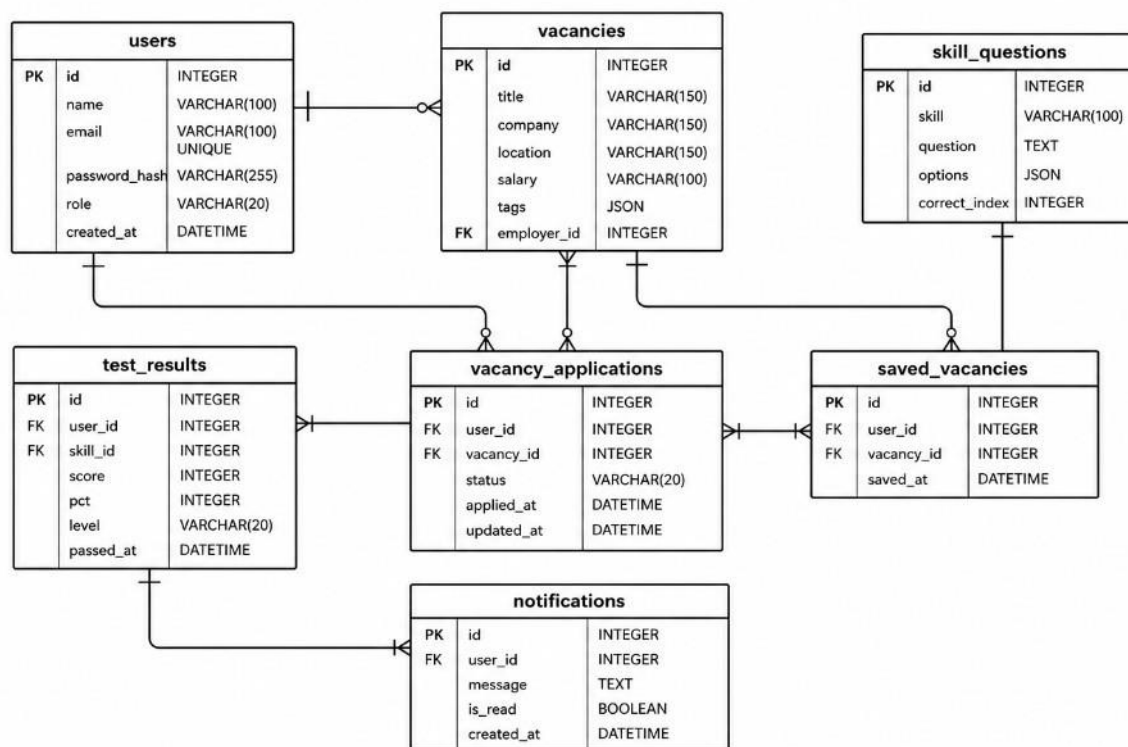
ДОДАТКИ

Додаток А. UML-діаграми системи

Діаграма варіантів використання (Use Case Diagram)
Система WorkUA Pro



Додаток Б.



Структура бази даних

Додаток В. Фрагменти програмного коду

```
from flask import Flask, request, jsonify
import json
```

```
app = Flask(__name__)
USERS = {}
```

```
VACANCIES = [
    {"id": 1, "title": "Python Backend Developer", "company": "TechUA",
     "location": "Київ / Remote", "salary": "2000-3500 $", "tags": ["Python", "Django",
     "PostgreSQL"], "desc": "Розробка серверної частини веб-застосунків."},
    {"id": 2, "title": "Frontend React Developer", "company": "DigitalHub",
     "location": "Львів / Remote", "salary": "1800-3000 $", "tags": ["React",
     "TypeScript", "CSS"], "desc": "Створення динамічних інтерфейсів для SaaS-
     платформ."},
    {"id": 3, "title": "Data Scientist", "company": "DataMind", "location":
     "Remote", "salary": "2500-4500 $", "tags": ["Python", "ML", "TensorFlow"],
     "desc": "Аналіз даних та побудова ML-моделей."},
    {"id": 4, "title": "DevOps Engineer", "company": "CloudSystems", "location":
     "Харків / Remote", "salary": "2200-4000 $", "tags": ["Docker", "Kubernetes",
     "CI/CD"], "desc": "Автоматизація деплою та підтримка інфраструктури."},
    {"id": 5, "title": "UI/UX Designer", "company": "CreativeStudio", "location":
     "Одеса / Remote", "salary": "1500-2500 $", "tags": ["Figma", "Prototyping", "UX
     Research"], "desc": "Проектування зручних інтерфейсів."},
    {"id": 6, "title": "QA Engineer", "company": "QualityFirst", "location":
     "Remote", "salary": "1200-2000 $", "tags": ["Selenium", "pytest", "API Testing"],
     "desc": "Тестування веб-застосунків та автоматизація тестів."},
]
```

```

SKILLS = {
    "Python": [
        {"q": "Що таке list comprehension?", "options": ["Метод сортування",
"Короткий спосіб створення списків", "Клас для масивів", "Функція
фільтрації"], "correct": 1},
        {"q": "Різниця між list і tuple?", "options": ["Немає різниці", "List
незмінний", "List змінний, tuple незмінний", "Обидва незмінні"], "correct": 2},
        {"q": "Що робить декоратор?", "options": ["Форматує код", "Обгортає
функцію", "Компілює код", "Видаляє помилки"], "correct": 1},
    ],
    "JavaScript": [
        {"q": "Різниця між == і ===?", "options": ["Немає різниці", "=== перевіряє
тип і значення", "== суворіший", "=== тільки для рядків"], "correct": 1},
        {"q": "Що таке Promise?", "options": ["Тип даних", "Об'єкт для
асинхронних операцій", "Цикл", "Клас"], "correct": 1},
        {"q": "Що робить Array.map()?", "options": ["Фільтрує масив", "Сортує
масив", "Трансформує кожен елемент", "Знаходить елемент"], "correct": 2},
    ],
}

```

```

@app.route("/")
def index():
    return HTML

@app.route("/api/login", methods=["POST"])
def login():
    d = request.json
    u = USERS.get(d.get("email", ""))
    if u and u["password"] == d.get("password"):

```

```

        return jsonify({"ok": True, "user": {"name": u["name"], "email":
u["email"]}}})
        return jsonify({"ok": False, "msg": "Невірний email або пароль"})

@app.route("/api/register", methods=["POST"])
def register():
    d = request.json
    email = d.get("email", "")
    if email in USERS:
        return jsonify({"ok": False, "msg": "Email вже зареєстровано"})
    USERS[email] = {"name": d.get("name"), "email": email, "password":
d.get("password")}
    return jsonify({"ok": True, "user": {"name": d["name"], "email": email}})

```

```

HTML = """<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>WorkUA Pro</title>
<link href="https://fonts.googleapis.com/css2?
family=Syne:wght@400;700;800&family=Inter:wght@300;400;500;600&display
=swap" rel="stylesheet">
<style>
:root{--bg:#0a0a0f;--s:#12121a;--s2:#1a1a28;--a:#6c63ff;--a2:#ff6584;--
a3:#43e97b;--t:#e8e8f0;--td:#7a7a9a;--b:#2a2a3e}
*{box-sizing:border-box;margin:0;padding:0}
body{background:var(--bg);color:var(--t);font-family:'Inter',sans-serif;min-
height:100vh}

```



```

nav{display:flex;align-items:center;justify-content:space-between;padding:16px
40px;position:sticky;top:0;z-index:100;background:rgba(10,10,15,.9);backdrop-
filter:blur(20px);border-bottom:1px solid var(--b)}
.logo{font-family:'Syne',sans-serif;font-size:22px;font-weight:800;color:var(--
t);text-decoration:none;cursor:pointer}
.logo span{color:var(--a)}
.nav-links{display:flex;gap:8px}
.nav-links a{color:var(--td);text-decoration:none;padding:8px 16px;border-
radius:8px;font-size:14px;font-weight:500;transition:all .2s;cursor:pointer}
.nav-links a:hover{color:var(--t);background:var(--s2)}
.nav-links a.btn{background:var(--a);color:#fff}
main{max-width:1100px;margin:0 auto;padding:0 24px}
.hero{min-height:85vh;display:flex;flex-direction:column;justify-
content:center;padding:60px 0}
h1{font-family:'Syne',sans-serif;font-size:clamp(40px,7vw,80px);font-
weight:800;line-height:1.05;letter-spacing:-2px;margin-bottom:24px}
.ac{color:var(--a)}.ac2{color:var(--a2)}
.hero p{font-size:18px;color:var(--td);max-width:500px;line-height:1.7;margin-
bottom:40px}
.sbar{display:flex;gap:12px;background:var(--s);border:1px solid var(--b);border-
radius:14px;padding:8px;max-width:560px;width:100%;margin-bottom:48px}
.sbar input{border:none;background:transparent;padding:8px 12px;font-
size:15px;color:var(--t);outline:none;flex:1}
.sbar button{background:var(--a);color:#fff;border:none;border-
radius:10px;padding:10px 24px;font-weight:600;cursor:pointer;font-size:14px}
.stats{display:flex;gap:40px;flex-wrap:wrap;margin-bottom:80px}
.sn{font-family:'Syne',sans-serif;font-size:36px;font-weight:800;color:var(--a)}
.sl{color:var(--td);font-size:14px}
.grid{display:grid;grid-template-columns:repeat(auto-
fill,minmax(300px,1fr));gap:20px;margin-bottom:60px}

```

```
.card{background:var(--s);border:1px solid var(--b);border-
radius:12px;padding:24px;transition:all .3s;cursor:pointer}
.card:hover{border-color:var(--a);transform:translateY(-2px);box-shadow:0 8px
32px rgba(108,99,255,.15)}
.co{font-size:13px;color:var(--td);margin-bottom:8px}
.card h3{font-family:'Syne',sans-serif;font-size:18px;font-weight:700;margin-
bottom:10px}
.sal{font-size:14px;font-weight:600;color:var(--a3);margin-bottom:10px}
.loc{font-size:13px;color:var(--td);margin-top:10px}
.tags{display:flex;flex-wrap:wrap;gap:6px}
.tag{display:inline-block;padding:4px 12px;border-radius:20px;font-
size:12px;font-weight:600;background:rgba(108,99,255,.15);color:var(--
a);border:1px solid rgba(108,99,255,.3)}
.st{font-family:'Syne',sans-serif;font-size:36px;font-weight:800;margin-
bottom:8px}
.ss{color:var(--td);margin-bottom:40px;font-size:16px}
#modal{display:none;position:fixed;inset:0;background:rgba(0,0,0,.8);backdrop-
filter:blur(8px);z-index:500;align-items:center;justify-
content:center;padding:20px}
#modal.open{display:flex}
#mbox{background:var(--s);border:1px solid var(--b);border-radius:16px;max-
width:600px;width:100%;padding:36px;position:relative;max-
height:90vh;overflow-y:auto}
#mcl{position:absolute;top:16px;right:20px;background:none;border:none;color:v
ar(--td);font-size:28px;cursor:pointer}
.btn1{display:inline-block;padding:12px 28px;background:var(--
a);color:#fff;border-radius:10px;font-weight:600;border:none;cursor:pointer;font-
family:'Inter',sans-serif;font-size:15px;transition:all .2s}
.btn1:hover{background:#7c74ff}
```

```

.btn2{display:inline-block;padding:12px 28px;background:transparent;color:var(--
t);border-radius:10px;font-weight:600;border:1px solid var(--
b);cursor:pointer;font-family:'Inter',sans-serif;font-size:15px;margin-left:12px}
.qopt{padding:14px 18px;border:1px solid var(--b);border-
radius:10px;cursor:pointer;font-size:15px;transition:all .2s;text-
align:left;background:var(--s2);color:var(--t);width:100%;margin-bottom:10px}
.qopt:hover{border-color:var(--a);color:var(--a)}
.qopt.sel{border-color:var(--a);background:rgba(108,99,255,.15);color:var(--a)}
.pb{height:4px;background:var(--b);border-radius:2px;margin-bottom:24px}
.pf{height:100%;background:var(--a);border-radius:2px;transition:width .4s}
.aw{min-height:80vh;display:flex;align-items:center;justify-
content:center;padding:40px 0}
.ab{background:var(--s);border:1px solid var(--b);border-
radius:16px;padding:48px;max-width:440px;width:100%}
.ab h2{font-family:'Syne',sans-serif;font-size:28px;font-weight:800;margin-
bottom:8px}
.ab p{color:var(--td);margin-bottom:32px}
.fg{margin-bottom:18px}
.fg label{display:block;margin-bottom:6px;color:var(--td);font-size:13px;font-
weight:500}
.fg input{background:var(--s2);border:1px solid var(--b);color:var(--t);border-
radius:10px;padding:12px 16px;font-family:'Inter',sans-serif;font-
size:15px;width:100%;outline:none;transition:border-color .2s}
.fg input:focus{border-color:var(--a)}
.al{text-align:center;margin-top:20px;color:var(--td);font-size:14px}
.al a{color:var(--a);text-decoration:none;font-weight:600;cursor:pointer}
.err{background:rgba(255,101,132,.1);border:1px solid
rgba(255,101,132,.3);color:#ff6584;padding:12px 16px;border-radius:8px;font-
size:14px;margin-bottom:16px;display:none}

```

```

.ph{padding:48px 0 32px;display:flex;align-items:center;gap:24px;flex-
wrap:wrap}
.av{width:80px;height:80px;border-radius:50%;background:linear-
gradient(135deg,var(--a),var(--a2));display:flex;align-items:center;justify-
content:center;font-size:32px;font-weight:800;font-family:'Syne',sans-
serif;color:#fff}
.pg{display:grid;grid-template-columns:2fr 1fr;gap:24px}
.sc{background:var(--s);border:1px solid var(--b);border-
radius:12px;padding:28px;margin-bottom:20px}
.sc h3{font-family:'Syne',sans-serif;font-size:20px;font-weight:700;margin-
bottom:20px}
.ri{display:flex;align-items:center;gap:16px;padding:14px 0;border-bottom:1px
solid var(--b)}
.ri:last-child{border-bottom:none}
.mb{height:6px;background:var(--b);border-radius:3px;margin-top:8px}
.mf{height:100%;border-radius:3px}
footer{text-align:center;padding:40px 24px;color:var(--td);font-size:13px;border-
top:1px solid var(--b);margin-top:80px}
#toast{position:fixed;bottom:30px;right:30px;padding:14px 24px;border-
radius:10px;background:var(--s2);border:1px solid var(--a3);color:var(--t);font-
size:14px;z-index:9999;transform:translateY(100px);transition:transform .3s}
#toast.show{transform:translateY(0)}
@media(max-width:640px){nav{padding:12px 16px}.pg{grid-template-
columns:1fr}}
</style>
</head>
<body>
<nav>
  <span class="logo" onclick="sp('home')">Work<span>UA</span> Pro</span>
  <div class="nav-links">

```

```

<a onclick="sp('vacancies')">Вакансії</a>
<a onclick="sp('skills')">Тести</a>
<a id="nlog" onclick="sp('login')">Увійти</a>
<a id="nreg" class="btn" onclick="sp('register')">Реєстрація</a>
</div>
</nav>
<main id="main"></main>
<footer><p>© 2024 WorkUA Pro — Інформаційна система пошуку роботи з
оцінкою навичок</p></footer>
<div id="modal"><div id="mbox"><button id="mcl"
onclick="cm()">×</button><div id="mc"></div></div></div>
<div id="toast"></div>
<script>
const V="" + json.dumps(VACANCIES, ensure_ascii=False) + "";
const SQ="" + json.dumps(SKILLS, ensure_ascii=False) + "";
let CU=null,QS=",QQ=[],QA=[],QI=0,SR={};

function toast(m,ok=true){const
t=document.getElementById('toast');t.textContent=m;t.style.borderColor=ok?'var(-
-a3)':'var(--
a2)';t.classList.add('show');setTimeout(()=>t.classList.remove('show'),3000)}
function cm(){document.getElementById('modal').classList.remove('open')}
function om(){document.getElementById('modal').classList.add('open')}

function un(){
  const l=document.getElementById('nlog'),r=document.getElementById('nreg');
  if(CU)
    {l.textContent=CU.name;l.onclick={()=>sp('profile')};r.textContent='Вийти';r.onclik
k={()=>{CU=null;un();sp('home')}}}

```

```

else {l.textContent='Увійти';l.onclick=()=>sp('login');r.textContent='Реєстрація';r.
onclick=()=>sp('register');}
}

```

```

function vc(v){return`<div class="card" onclick="ov(${v.id})"><div class="co">
${v.company}</div><h3>${v.title}</h3><div class="sal">${v.salary}</div><div
class="tags">${v.tags.map(t=>`<span
class="tag">${t}</span>`).join("")}</div><div class="loc">
${v.location}</div></div>`}

```

```

function sp(p){
  const m=document.getElementById('main');
  if(p==='home'){m.innerHTML=`<div class="hero"><h1>Знайди
роботу<br><span class="ac">свої мрії</span><br>в <span
class="ac2">Україні</span></h1><p>Платформа що поєднує пошук вакансій з
оцінкою твоїх технічних навичок.</p><div class="sbar"><input id="si"
placeholder="Посада або технологія..."
onkeydown="if(event.key==='Enter')ds()"><button onclick="ds()">Q
Шукати</button></div><div class="stats"><div><div class="sn">6</div><div
class="sl">Вакансій</div></div><div><div class="sn">2</div><div
class="sl">Тести навичок</div></div><div><div class="sn">100%</div><div
class="sl">Безкоштовно</div></div></div></div><h2 class="st">Свіжі
вакансії</h2><p class="ss">Актуальні пропозиції від провідних
компаній</p><div class="grid">${V.map(v=>vc(v)).join("")}</div>`}
  else if(p==='vacancies'){m.innerHTML=`<div style="padding:48px 0 32px"><h1
class="st">Вакансії</h1><p class="ss">Знайди свою ідеальну
позицію</p><div class="sbar" style="max-width:100%"><input id="si"
placeholder="Пошук..."><button onclick="fv()">Q

```

```

Знайти</button></div></div><div class="grid" id="vg">$
{V.map(v=>vc(v)).join("")}</div>`
  else if(p==='skills'){m.innerHTML=`<div style="padding:48px 0 32px"><h1
class="st">🔗 Оцінка навичок</h1><p class="ss">Перевір свій
рівень</p></div><div class="grid">${Object.keys(SQ).map(s=>`<div
class="card" onclick="sq('${s}')"><div style="font-size:48px;margin-
bottom:16px">${s==='Python'?`🔗`:'`</div><h3>${s}</h3><p style="color:var(--
td);font-size:14px;margin-top:8px">3 питання · ~3 хв</p>${SR[s]?`<div
style="margin-top:12px;color:var(--a3);font-weight:600">✓ ${SR[s].pct}% — $
{SR[s].lv}</div>`:'`</div>`).join("")}</div>`}
  else if(p==='login'){m.innerHTML=`<div class="aw"><div
class="ab"><h2>Вхід до системи</h2><p>Раді бачити тебе знову!</p><div
class="err" id="er"></div><div class="fg"><label>Email</label><input
type="email" id="le" placeholder="your@email.com"></div><div
class="fg"><label>Пароль</label><input type="password" id="lp"
placeholder="••••••••" onkeydown="if(event.key==='Enter')dl()"></div><button
class="btn1" style="width:100%" onclick="dl()">Увійти →</button><div
class="al">Немає акаунту? <a
onclick="sp('register')">Зареєструватись</a></div></div></div>`}
  else if(p==='register'){m.innerHTML=`<div class="aw"><div
class="ab"><h2>Створити акаунт</h2><p>Приєднуйся до WorkUA
Pro!</p><div class="err" id="er"></div><div
class="fg"><label>Імя</label><input type="text" id="rn" placeholder="Іван
Петренко"></div><div class="fg"><label>Email</label><input type="email"
id="re" placeholder="your@email.com"></div><div
class="fg"><label>Пароль</label><input type="password" id="rp"
placeholder="Мінімум 6 символів"></div><button class="btn1"
style="width:100%" onclick="dr()">Зареєструватись →</button><div
class="al">Є акаунт? <a onclick="sp('login')">Увійти</a></div></div></div>`}
  else if(p==='profile'){

```

```

if(!CU){sp('login');return}
const rs=Object.entries(SR);
const avg=rs.length?Math.round(rs.reduce((s,[r])=>s+r.pct,0)/rs.length):0;
m.innerHTML=`<div class="ph"><div class="av">$
{CU.name[0].toUpperCase()}</div><div><h2 style="font-family:'Syne',sans-
serif;font-size:28px;font-weight:800">${CU.name}</h2><p style="color:var(--
td)">${CU.email}</p></div></div><div class="pg"><div
class="sc"><h3>Результати тестів</h3>${rs.length?rs.map(([s,r])=>`<div
class="ri"><div style="font-size:28px">${s==='Python'?'🐍':' '}</div><div
style="flex:1"><div style="font-weight:600">${s}</div><div style="font-
size:13px;color:var(--td)">${r.lv} · ${r.sc}/${r.tot}</div><div class="mb"><div
class="mf" style="width:${r.pct}%;background:${r.pct>=75?'var(--
a3)':r.pct>=40?'var(--a)':'var(--a2)'}"></div></div></div><div style="font-
family:'Syne',sans-serif;font-size:24px;font-weight:800;color:${r.pct>=75?'var(--
a3)':r.pct>=40?'var(--a)':'var(--a2)'}">${r.pct}%</div></div>`).join("):`<div
style="text-align:center;padding:40px;color:var(--td)"><div style="font-
size:48px;margin-bottom:16px">🏆</div><p>Ти ще не проходив
тести</p><button class="btn1" style="margin-top:16px"
onclick="sp('skills')">Пройти тест</button></div>`}</div><div><div
class="sc"><h3>🏆 Дії</h3><div style="display:flex;flex-
direction:column;gap:10px"><button class="btn1" onclick="sp('vacancies')">Q
Вакансії</button><button class="btn2" style="margin:0" onclick="sp('skills')">🏆
Тести</button></div></div><div class="sc"><h3> Статистика</h3><div
style="display:flex;justify-content:space-between;margin-bottom:12px"><span
style="color:var(--td);font-size:14px">Тестів пройдено</span><span style="font-
family:'Syne',sans-serif;font-size:20px;font-weight:800;color:var(--a)">${
rs.length}</span></div>${rs.length?`<div style="display:flex;justify-
content:space-between"><span style="color:var(--td);font-size:14px">Середній

```



```

    бал</span><span style="font-family:'Syne',sans-serif;font-size:20px;font-
weight:800;color:var(--a3)">${avg}%</span></div>`:"</div></div></div>`
  }
}

```

```

function ov(id){
  const v=V.find(x=>x.id===id);
  document.getElementById('mc').innerHTML=`<div style="font-
size:13px;color:var(--td);margin-bottom:8px"> ${v.company}</div><h2
style="font-family:'Syne',sans-serif;font-size:26px;font-weight:800;margin-
bottom:8px">${v.title}</h2><div style="font-size:20px;font-
weight:700;color:var(--a3);margin-bottom:8px">${v.salary}</div><div
style="color:var(--td);font-size:14px;margin-bottom:16px">
${v.location}</div><div class="tags" style="margin-bottom:16px">$
{v.tags.map(t=>`<span class="tag">${t}</span>`).join("")}</div><p
style="color:var(--td);line-height:1.7;margin-bottom:20px">${v.desc} Гнучкий
графік, Remote, оплачувана відпустка 24 дні.</p><button class="btn1"
onclick="av()">Відгукнутись</button><button class="btn2"
onclick="cm();sp('skills')">🔗 Пройти тест</button>`;
  om();
}

```

```

function av(){if(!CU){cm();sp('login');return}toast('✓ Заявку надіслано!');cm()}

```

```

function ds(){const
q=document.getElementById('si').value;sp('vacancies');setTimeout(()=>{document
.getElementById('si').value=q;fv()},50)}
function fv(){const q=document.getElementById('si').value.toLowerCase();const
f=q?V.filter(v=>v.title.toLowerCase().includes(q)||v.tags.join(
').toLowerCase().includes(q)):V;document.getElementById('vg').innerHTML=f.len

```

```
gth?f.map(v=>vc(v)).join("):'<p style="color:var(--td);padding:40px">Нічого не  
знайдено</p>{'}
```

```
function dl(){  
  const  
  e=document.getElementById('le').value,p=document.getElementById('lp').value;  
  fetch('/api/login',{method:'POST',headers: {'Content-Type':'application/  
  json'},body:JSON.stringify({email:e,password:p})})  
    .then(r=>r.json()).then(d=>{if(d.ok){CU=d.user;un();sp('profile');toast('Ласкаво  
  просимо, '+CU.name+'!')}else{const  
  er=document.getElementById('er');er.textContent=d.msg;er.style.display='block'}}  
  );  
}
```

```
function dr(){  
  const  
  n=document.getElementById('rn').value,e=document.getElementById('re').value,p  
  =document.getElementById('rp').value;  
  if(!n||!e||!p){const er=document.getElementById('er');er.textContent='Заповніть  
  всі поля';er.style.display='block';return}  
  if(p.length<6){const er=document.getElementById('er');er.textContent='Пароль  
  мінімум 6 символів';er.style.display='block';return}  
  fetch('/api/register',{method:'POST',headers: {'Content-Type':'application/  
  json'},body:JSON.stringify({name:n,email:e,password:p})})  
    .then(r=>r.json()).then(d=>{if(d.ok){CU=d.user;un();sp('profile');toast('Акаунт  
  створено!')}else{const  
  er=document.getElementById('er');er.textContent=d.msg;er.style.display='block'}}  
  );  
}
```

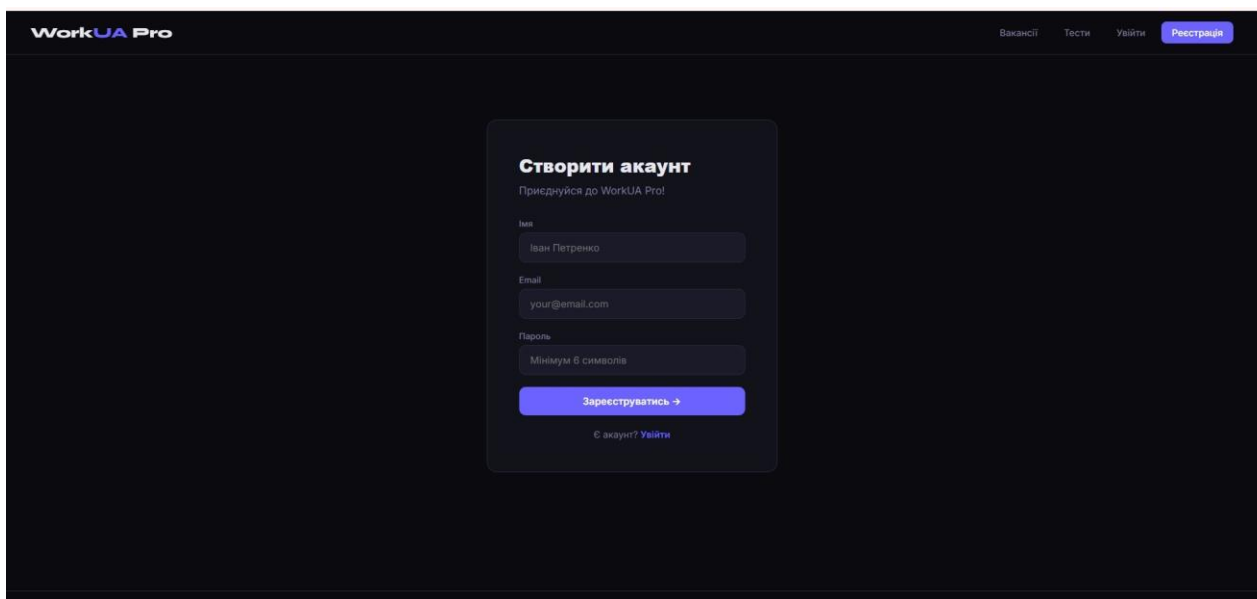
```
function sq(skill){
```

```

    QS=skill;QQ=SQ[skill];QA=[];QI=0;rq();om();
  }
  function rq(){
    const q=QQ[QI];const pct=(QI/QQ.length*100).toFixed(0);
    document.getElementById('mc').innerHTML=`<h2 style="font-
family:'Syne',sans-serif;font-size:22px;font-weight:800;margin-bottom:4px">Тест:
${QS}</h2><p style="color:var(--td);font-size:14px;margin-
bottom:16px">Питання ${QI+1} з ${QQ.length}</p><div class="pb"><div
class="pf" style="width:${pct}%></div></div><div style="font-size:17px;font-
weight:500;margin-bottom:20px;line-height:1.5">${q.q}</div>${
q.options.map((o,i)=>`<button class="qopt${QA[QI]===i?' sel':''}" onclick="sa($
{i})">${o}</button>`).join("")}<div style="display:flex;justify-content:space-
between;margin-top:10px"><button class="btn2" onclick="pq()"
style="padding:10px 20px;${QI===0?'opacity:.3;pointer-events:none':''}>←
Назад</button><button class="btn1" onclick="nq()" style="padding:10px 20px;${
QA[QI]===undefined?'opacity:.5;pointer-events:none':''}>${
QI===QQ.length-1?'Завершити ✓':'Далі →'}</button></div>`;
  }
  function sa(i){QA[QI]=i;rq()}
  function pq(){if(QI>0){QI--;rq()}}
  function nq(){if(QA[QI]===undefined)return;if(QI<QQ.length-1){QI++;rq()}else
subq()}
  function subq(){
    const sc=QA.reduce((s,a,i)=>s+(a===QQ[i].correct?1:0),0);
    const tot=QQ.length,pct=Math.round(sc/tot*100);
    const lv=pct<40?'Початківець':pct<75?'Середній рівень':'Експерт';
    SR[QS]={sc,tot,pct,lv};
    const st=pct>=75?'★ ★ ★':pct>=40?'★ ★ ':'★';
    document.getElementById('mc').innerHTML=`<div style="text-
align:center"><div style="font-size:48px;margin-bottom:12px">${st}</div><div

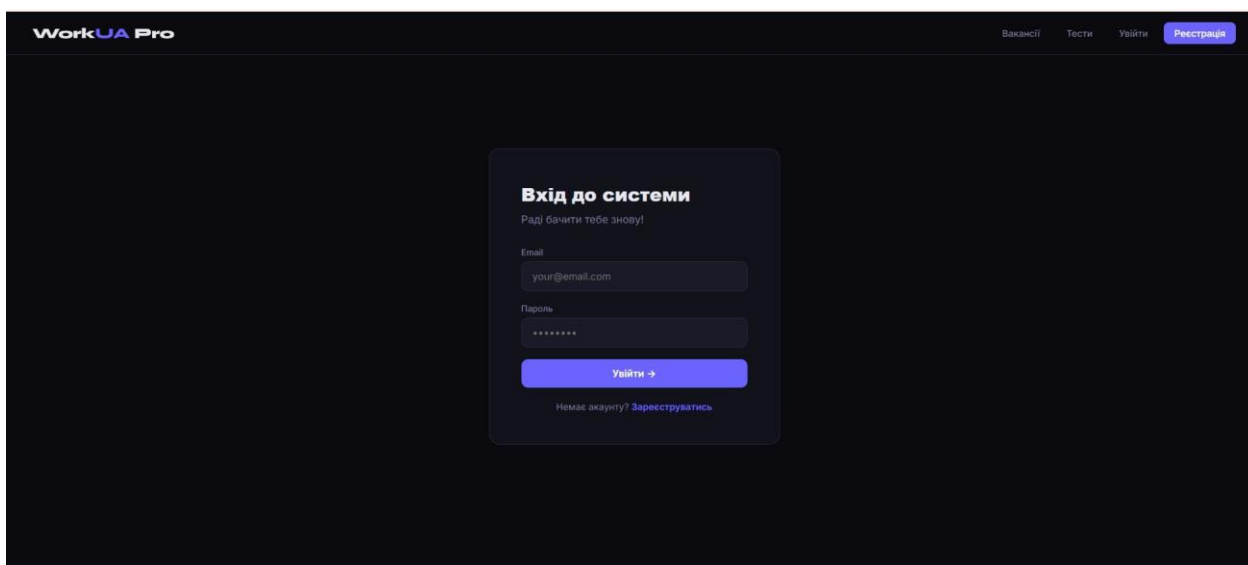
```

```
style="font-family:'Syne',sans-serif;font-size:72px;font-weight:800;color:var(--
a)">${pct}%</div><div style="font-size:22px;font-weight:600;margin:12px 0">${
lv}</div><p style="color:var(--td);margin-bottom:24px">Правильных: ${sc} з ${
tot}</p><button class="btn1" onclick="sq('${QS}')">Пройти
знову</button><button class="btn2"
onclick="cm();sp('skills')">Закрити</button></div>`;
}
document.getElementById('modal').addEventListener('click',e=>{if(e.target===do
cument.getElementById('modal'))cm()});
sp('home');un();
</script>
</body>
</html>"""
```



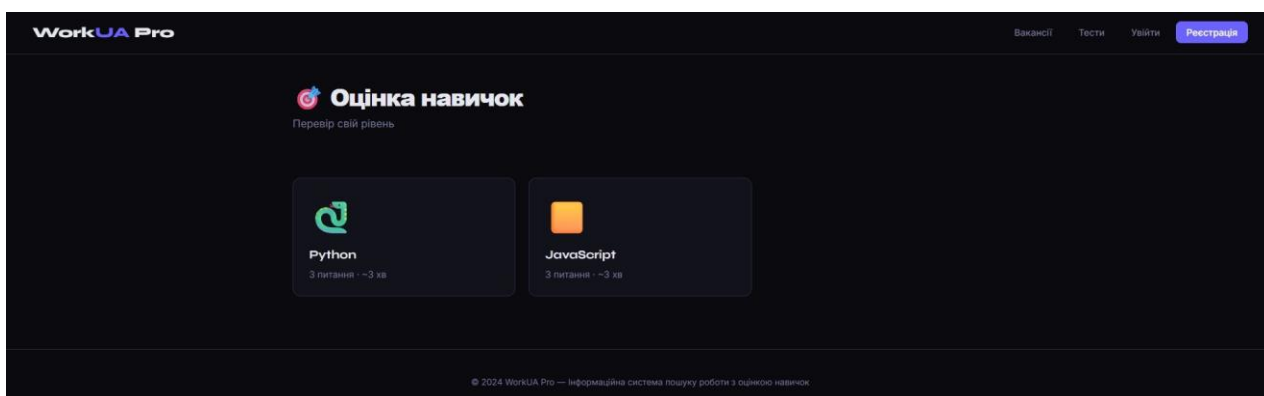
The image shows the registration page of the WorkUA Pro website. The page has a dark blue background. At the top, there is a navigation bar with the logo "WorkUA Pro" on the left and links for "Вакансії", "Тести", "Увійти", and "Реєстрація" on the right. The "Реєстрація" link is highlighted with a blue button. In the center, there is a white card titled "Створити акаунт" (Create account) with the subtitle "Приєднуйся до WorkUA Pro!". Below the title, there are three input fields: "Ім'я" (Name) with the placeholder "Іван Петренко", "Email" with the placeholder "your@email.com", and "Пароль" (Password) with the placeholder "Мінімум 6 символів". Below these fields is a blue button labeled "Зареєструватись →" (Register). At the bottom of the card, there is a link "Є акаунт? Увійти" (Have an account? Log in).

Рис.4.1. Реєстраційна сторінка сайту



The image shows the login page of the WorkUA Pro website. The page has a dark blue background. At the top, there is a navigation bar with the logo "WorkUA Pro" on the left and links for "Вакансії", "Тести", "Увійти", and "Реєстрація" on the right. The "Реєстрація" link is highlighted with a blue button. In the center, there is a white card titled "Вхід до системи" (Login to the system) with the subtitle "Раді бачити тебе знову!" (We're glad to see you again!). Below the title, there are two input fields: "Email" with the placeholder "your@email.com" and "Пароль" (Password) with the placeholder "*****". Below these fields is a blue button labeled "Увійти →" (Log in). At the bottom of the card, there is a link "Немає акаунту? Зареєструватись" (No account? Register).

Рис.4.2. Реєстраційна сторінка входу на сайт



The image shows the skill assessment page of the WorkUA Pro website. The page has a dark blue background. At the top, there is a navigation bar with the logo "WorkUA Pro" on the left and links for "Вакансії", "Тести", "Увійти", and "Реєстрація" on the right. The "Реєстрація" link is highlighted with a blue button. In the center, there is a white card titled "Оцінка навичок" (Skill assessment) with the subtitle "Перевір свій рівень" (Check your level). Below the title, there are two cards: "Python" with a green icon and "JavaScript" with an orange icon. Each card has the text "3 питання - ~3 хв" (3 questions - ~3 min). At the bottom of the page, there is a footer with the text "© 2024 WorkUA Pro — Інформаційна система пошуку роботи з оцінкою навичок" (© 2024 WorkUA Pro — Information system for job search with skill assessment).

Рис.4.3. Сторінка тестування сайту

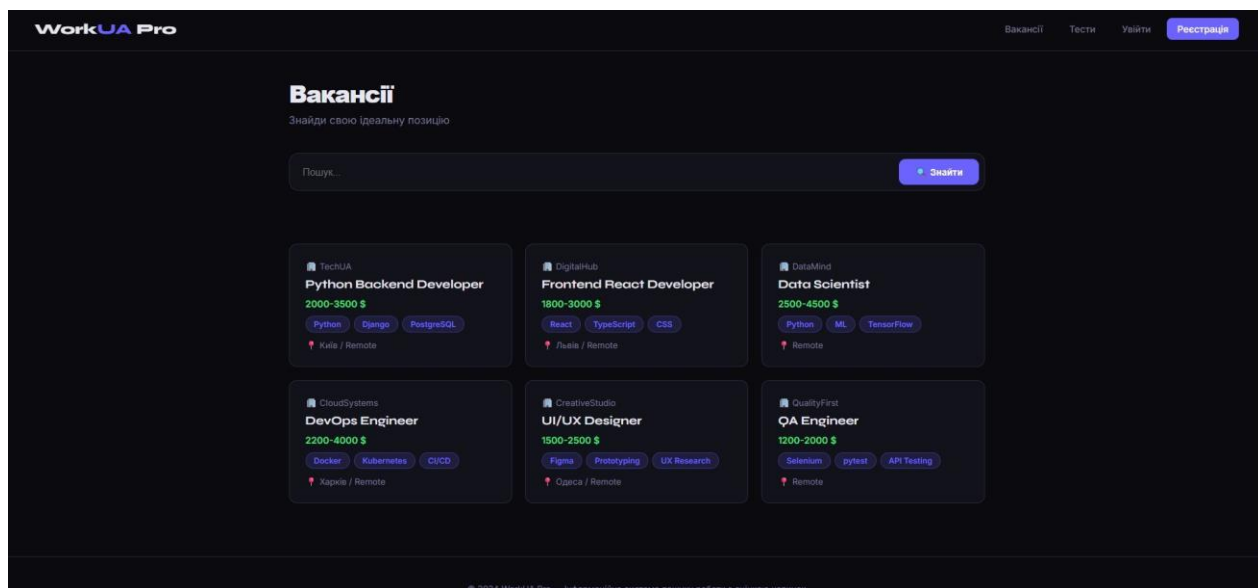


Рис.4.4. Голова сторінки сайту вакансій

Інтерфейс веб-застосунку

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

Я, Г р е ч к а Р о с т и с л а в Р у с л а н о в и ч , здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Інформаційна система пошуку роботи з оцінкою професійних навичок» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
 - о ☐ інструменти генеративного ШІ не використовувалися.
 - о ☒ інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL, Gemini, ChatGPT _____ інші (вказати назву)) були використані для:
 - ☒ перекладу іншомовних джерел;
 - ☒ стилістичного редагування та перевірки граматики;
 - ☒ допомоги у написанні/відлагодженні програмного коду;
 - ☐ інше: Пошук корисно _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«25 »05 2026 р. Гречка Ростислав Русланович