МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: СЕРВІС ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ

Здобувача вищої освіти другого (магістерського) рівня,
II курсу, групи КМІ-24-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»

_____ Короткова Кирила Олександровича
(підпис) (прізвище, ім'я, по батькові)

Науковий керівник _____ Гладченко Оксана Вікторівна, канд. пед. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-професійної програми _____ Омельчук Антон Анатолійович, канд. техн. наук
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчен звання)

Допущено до захисту за рішенням кафедри протокол від «02» січня 2026 № 10

Завідувач кафедри _____ Філоненко Михайло Миколайович, канд. фіз.-мат. наук, доцент
(підпис) (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2026

Реєстрація №32 22.01.2026 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 02.01.2026 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 3 (задовільно) 10.02.2026 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ О.С. Дюбанов
(підпис)

Члени комісії:

_____ А.Ю. Горбовий
(підпис)

_____ М.М. Філоненко
(підпис)

_____ І.Д. Погорєловська
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст кваліфікаційної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	18		10.02.2026 Дюбанов О.С.
2. Дотримання вимог щодо оформлення	від 0 до 10	6		Горбовий А.Ю.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Філоненко М.М. Погорєловська І.Д.
4. Рецензент	від 0 до 5	4		Суховецька Л.П.
РАЗОМ БАЛІВ	від 0 до 50	32		

Оцінювання кваліфікаційної роботи науковим керівником - 33 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
65	D	3 (задовільно)	10.02.2026	Дюбанов О.С. Горбовий А.Ю. Філоненко М.М. Погорєловська І.Д. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології в економіці»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«11» грудня 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Короткову Кирилу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Сервіс для пошуку та відстеження відгуків на вакансії
Затверджена наказом ректора від «10» грудня 2024 № 1852
2. Строк здачі закінченої роботи «02» січня 2026
3. Мета роботи Полягає у проєктуванні та програмній реалізації універсальної вебплатформи, інтегрованої із засобами професійної підготовки та практичної апробації навичок кандидатів.
Об'єкт дослідження Функціональні можливості та архітектурні особливості сучасних платформ із працевлаштування.
Предмет дослідження Методи автоматизації оцінки технічних навичок (hard skills) та архітектурні підходи до побудови масштабованих вебсистем у сфері HR-tech.
Орієнтовна база дослідження Сучасні платформи із працевлаштування.
4. Вихідні дані до роботи Вимоги до інформаційної системи пошуку та відстеження відгуків.
5. Перелік графічного матеріалу Детальний пошук. Час відповіді; Швидкий пошук. Час відповіді; Інтерфейс системи;
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
ВСТУП; РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СЕРВІСІВ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ; РОЗДІЛ 2. ПРОЄКТУВАННЯ СЕРВІСУ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ; РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ JOBTRACKER; РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ JOBTRACKER; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.

7. Календар план виконання роботи:

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.12.2024	До 15.12.2024
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	28.05.2025	28.05.2025
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	08.09.2025	08.09.2025
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	17.11.2025	17.11.2025
5	**Висновки**	08.12.2025	08.12.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	22.12.2025	22.12.2025
7	Отримання відгуку керівника	20.01.2026	20.01.2026
8	Отримання зовнішньої рецензії	23.01.2026	23.01.2026

8. Дата видачі завдання «11» грудня 2024

9. Керівник кваліфікаційної роботи

(підпис) Гладченко О.В. (прізвище, ініціали)

10. Керівник освітньо-професійної програми

(підпис) Омельчук А.А. (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти

(підпис) Коротков К.О. (прізвище, ініціали)

АНОТАЦІЯ

СЕРВІС ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ

У кваліфікаційній роботі було виконано повний цикл розроблення, проєктування, реалізації та експериментальної оцінки веб системи JobTracker, призначеної для підтримки процесів пошуку роботи, управління вакансіями та відстеження статусів поданих заявок. На основі проведених аналітичних, конструкторських та експериментальних досліджень можна сформулювати такі основні висновки.

Загалом, виконана робота продемонструвала, що розроблена система JobTracker є технологічно стійкою, функціонально повною і придатною до практичного застосування як сучасний інструмент підтримки процесів працевлаштування. Розробка досягла поставлених цілей, підтверджена експериментальними дослідженнями та створила підґрунтя для подальшого розвитку проєкту.

Кваліфікаційна робота містить 41 сторінок, 9 рисунків, 2 таблиці, список використаних джерел 20 найменувань.

Ключові слова: інформаційна система, пошук, сервіс, вакансії.

Рік захисту роботи 2026

ABSTRACT

SERVICE FOR SEARCHING AND TRACKING VACANCY REVIEWS

The qualification work included a full cycle of development, design, implementation and experimental evaluation of the JobTracker web system, designed to support job search processes, manage vacancies and track the status of submitted applications. Based on the analytical, design and experimental research conducted, the following main conclusions can be formulated.

In general, the work performed demonstrated that the developed JobTracker system is technologically stable, functionally complete and suitable for practical use as a modern tool for supporting employment processes. The development achieved the set goals, was confirmed by experimental research and created a basis for further development of the project.

The qualification work contains 41 pages, 9 figures, 2 tables, a list of sources used 20 items.

Keywords: information system, search, service, vacancies.

The year of defense of the work is 2026

Зміст

ВСТУП 3

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СЕРВІСІВ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ 5

РОЗДІЛ 2. ПРОЄКТУВАННЯ СЕРВІСУ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ 15

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ JOBTRACKER 18

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ JOBTRACKER 30

ВИСНОВКИ 37

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 40

ВСТУП

Актуальність теми. Стрімкий розвиток галузі інформаційних технологій зумовлює високу конкуренцію на ринку праці та трансформацію підходів до рекрутингу. Попри велику кількість агрегаторів вакансій, стандартні описи позицій часто залишаються надто формалізованими та не відображають реального змісту майбутніх завдань. Це створює когнітивний дисонанс між очікуваннями апліканта та специфікою робочих процесів. Впровадження методології практичних кейсів з обов'язковим зворотним зв'язком дозволить нівелювати цей розрив. Такий підхід забезпечить синергетичний ефект: роботодавці отримають об'єктивну оцінку компетенцій кандидатів, а потенційні працівники — чітке розуміння зони відповідальності та вектори для подальшого професійного розвитку.

Мета роботи полягає у проєктуванні та програмній реалізації універсальної вебплатформи, інтегрованої із засобами професійної підготовки та практичної апробації навичок кандидатів.

Завдання дослідження:

- Здійснити комплексний аналіз предметної області та існуючих рішень для пошуку персоналу.
- Дослідити сучасні архітектурні стилі побудови API та технологічний стек для розробки повнофункціональних вебзастосунків.
- Розробити систему, яка мінімізує невизначеність для кандидатів через симуляцію реальних професійних викликів.
- Автоматизувати процеси первинного відбору для рекрутерів, підвищивши якість формування команд.

Об’єкт дослідження — функціональні можливості та архітектурні особливості сучасних платформ із працевлаштування.

Предмет дослідження — методи автоматизації оцінки технічних навичок (hard skills) та чархітектурні підходи до побудови масштабованих вебсистем у сфері HR-tech.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СЕРВІСІВ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ

Актуальність теми кваліфікаційної роботи зумовлена зростанням ролі цифрових сервісів у процесах працевлаштування та управління людськими ресурсами. У сучасних умовах конкуренції на ринку праці кандидати дедалі частіше спираються не лише на опис вакансій, але й на відгуки інших працівників, рейтинги роботодавців та аналітичні інструменти для відстеження власної активності під час пошуку роботи. Це формує запит на комплексні інформаційні системи, здатні поєднувати пошук вакансій, аналіз відгуків та моніторинг статусів відгуків.

Об’єктом дослідження є цифрові сервіси ринку праці.

Предметом дослідження є функціональні та аналітичні можливості сервісів для пошуку та відстеження відгуків на вакансії.

Метою магістерької роботи є проведення оглядового та порівняльного аналізу існуючих сервісів, а також визначення напрямів їх подальшого розвитку в умовах цифровізації економіки.

Сучасний ринок праці характеризується високою динамічністю, цифровізацією процесів найму та зростаючою роллю інформаційної прозорості. Одним із ключових факторів, що впливають на рішення кандидатів, є відгуки про вакансії та роботодавців, а також можливість відстеження власних відгуків і статусів поданих заявок. У зв’язку з цим особливої актуальності набувають спеціалізовані сервіси для пошуку роботи, аналізу репутації компаній і трекінгу процесу працевлаштування.

Метою даного огляду є системний аналіз існуючих сервісів для пошуку та відстеження відгуків на вакансії, визначення їх функціональних можливостей, переваг і недоліків, а також окреслення перспектив розвитку таких платформ у контексті цифрової трансформації ринку праці.

Теоретичні засади формування цифрових сервісів ринку праці

Еволюція процесів пошуку роботи

Традиційні методи пошуку роботи (друковані оголошення, особисті рекомендації) поступово були витіснені онлайн-платформами. З розвитком Web 2.0 та соціальних мереж відбувся перехід до інтерактивних сервісів, де користувачі не лише споживають інформацію, але й формують її через відгуки, рейтинги та коментарі.

Роль відгуків у прийнятті рішень кандидатами

Відгуки працівників та кандидатів виконують функцію соціального доказу, зменшуючи інформаційну асиметрію між роботодавцем і шукачем роботи. Вони дозволяють оцінити корпоративну культуру, умови праці, стиль управління та реальний рівень компенсації.

Класифікація сервісів для пошуку та відстеження відгуків на вакансії

Сучасні сервіси можна умовно поділити на такі групи:

Платформи пошуку вакансій з елементами відгуків.

Спеціалізовані сервіси відгуків про роботодавців.

Інструменти для персонального відстеження відгуків і статусів заявок.

Корпоративні ATS-системи та HR-платформи.

Платформи пошуку вакансій з елементами відстеження та відгуків

Платформи даної групи поєднують базові функції пошуку роботи з додатковими можливостями отримання зворотного зв’язку щодо роботодавців та вакансій. Вони є найбільш популярними серед широкого кола користувачів завдяки простоті використання та великій кількості пропозицій.

Indeed

Indeed є однією з найбільших глобальних платформ для пошуку роботи. Окрім вакансій, сервіс надає доступ до відгуків про компанії, рейтингів роботодавців та інформації про заробітні плати.

Переваги: велика база вакансій, міжнародне покриття, інтеграція відгуків.

Недоліки: обмежена глибина аналізу відгуків, відсутність розширеного персонального трекінгу заявок.

Adzuna

Adzuna функціонує як агрегатор вакансій із різних джерел та пропонує аналітичні інструменти для аналізу ринку праці.

Переваги: використання аналітики та AI-інструментів, глобальне охоплення.

Недоліки: мінімальна кількість відгуків про роботодавців.

Insidetrak

Insidetrak поєднує пошук вакансій із відгуками співробітників, орієнтуючись переважно на локальні ринки.

Спеціалізовані сервіси відгуків про роботодавців

Glassdoor

Glassdoor є провідною платформою для публікації анонімних відгуків працівників, інформації про зарплати та процеси співбесід.

Переваги: детальні відгуки, багатовимірні рейтинги.

Недоліки: суперечки щодо анонімності та достовірності даних.

Kununu, Comparably, FairyGodBoss

Ці платформи орієнтовані на окремі сегменти ринку праці або спеціальні критерії оцінювання роботодавців, доповнюючи інформацію, доступну на глобальних сервісах.

Інструменти для персонального відстеження відгуків на вакансії

Teal

Teal дозволяє кандидатам систематизувати процес пошуку роботи, відстежувати подані заявки та аналізувати ефективність резюме.

Мобільні трекери заявок

Мобільні додатки для трекінгу заявок забезпечують базову функціональність ведення журналу відгуків і статусів співбесід.

ATS та HR-платформи

З точки зору спеціальності 122 «Комп'ютерні науки», ATS та HR-платформи доцільно розглядати як складні програмні інформаційні системи, що реалізують повний цикл збору, зберігання, обробки та аналізу даних про кандидатів і вакансії. Такі системи поєднують алгоритмічні методи обробки даних, бази даних, веб-технології та елементи штучного інтелекту.

Архітектура ATS-систем

Типова архітектура ATS-системи базується на клієнт-серверному або хмарному підході та включає такі компоненти:

рівень представлення (web-інтерфейс для рекрутерів і адміністраторів);

прикладний рівень (бізнес-логіка обробки відгуків і вакансій);

рівень даних (реляційні та NoSQL бази даних);

інтеграційний рівень (API для взаємодії з job-порталами та соціальними мережами).

Алгоритмічні аспекти обробки відгуків

З позицій комп'ютерних наук ключову роль відіграють алгоритми:

автоматичного парсингу резюме;

класифікації кандидатів за вимогами вакансій;

ранжування відгуків на основі релевантності;

фільтрації дублікатів та некоректних заявок.

Використання таких алгоритмів дозволяє значно підвищити ефективність роботи системи та зменшити часові витрати на первинний відбір кандидатів.

Аналітичні та наукові підходи до аналізу відгуків

У межах спеціальності 122 «Комп'ютерні науки» відгуки на вакансії розглядаються як неструктуровані дані, аналіз яких потребує застосування методів обробки природної мови та машинного навчання.

Аналіз текстових відгуків (Text Mining)

Для аналізу текстових відгуків застосовуються такі етапи:

попередня обробка тексту (очищення, токенізація, лематизація);

визначення тональності (sentiment analysis);

тематичне моделювання;

виявлення ключових слів та фраз.

Ці методи дозволяють автоматично виявляти сильні та слабкі сторони вакансій і роботодавців.

Машинне навчання та рекомендаційні системи

Моделі машинного навчання використовуються для:

прогнозування відповідності кандидата вакансії;

формування персоналізованих рекомендацій;

оцінювання ймовірності успішного проходження відбору.

Застосування таких підходів є характерним для сучасних інтелектуальних сервісів ринку праці.

Порівняльний аналіз сервісів

Порівняльний аналіз дозволяє виявити спільні та відмінні риси розглянутих сервісів. Зокрема, платформи типу Indeed та Adzuna орієнтовані на масовий ринок і забезпечують широкий вибір вакансій, проте мають обмежені можливості глибокого аналізу відгуків. Натомість спеціалізовані сервіси, такі як

Glassdoor або Kununu, надають детальніші якісні оцінки роботодавців, але не завжди охоплюють повний спектр вакансій.

Інструменти персонального трекінгу (Teal, мобільні трекери) зосереджені на індивідуальних потребах кандидата і майже не виконують аналітичної функції щодо роботодавців. ATS-системи, у свою чергу, орієнтовані на корпоративних користувачів та забезпечують високу ефективність рекрутингових процесів, однак є малодоступними для кінцевих шукачів роботи.

Порівняння функціональних можливостей показує, що жоден сервіс не охоплює всі потреби кандидатів, що зумовлює необхідність комбінування кількох інструментів.

Перспективи розвитку сервісів відстеження відгуків

Використання штучного інтелекту та машинного навчання

Подальша еволюція сервісів для пошуку та відстеження відгуків на вакансії нерозривно пов’язана з активним впровадженням методів штучного інтелекту та машинного навчання. Уже сьогодні такі алгоритми застосовуються для автоматизованого аналізу резюме, класифікації вакансій, прогнозування відповідності кандидата вимогам роботодавця та персоналізації рекомендацій. У перспективі очікується ширше використання моделей глибинного навчання для аналізу великих масивів неструктурованих даних, зокрема текстових відгуків.

Розвиток методів аналізу текстів та емоційного забарвлення

Одним із ключових напрямів розвитку є поглиблення аналізу відгуків із застосуванням методів обробки природної мови (NLP). Зокрема, йдеться про автоматичне визначення тональності відгуків (sentiment analysis), виявлення ключових тем, повторюваних проблем і трендів у корпоративній культурі компаній. Це дозволить формувати більш об’єктивні рейтинги роботодавців та забезпечувати кандидатів аналітичними зведеннями, а не лише окремими коментарями.

Інтеграція з професійними соціальними мережами та API

Перспективним напрямом є інтеграція сервісів пошуку роботи з професійними соціальними мережами та зовнішніми API. Це забезпечує обмін даними між платформами, автоматичне оновлення статусів заявок та формування єдиного цифрового профілю кандидата. З технічної точки зору така інтеграція потребує використання REST- або GraphQL-інтерфейсів, механізмів автентифікації OAuth та захисту персональних даних.

Підвищення прозорості алгоритмів і етичні аспекти

Зі зростанням ролі алгоритмів ухвалення рішень актуалізується проблема прозорості та етичності. Кандидати дедалі частіше вимагають пояснень щодо причин відхилення заявок або низьких рейтингів. Це стимулює розвиток підходів Explainable AI (XAI), які дозволяють інтерпретувати результати роботи моделей машинного навчання та підвищувати довіру користувачів до цифрових сервісів.

З точки зору комп’ютерних наук перспективи розвитку сервісів для пошуку та відстеження відгуків на вакансії пов’язані з побудовою інтелектуальних інформаційних систем. Основними напрямами розвитку є:

застосування глибинних нейронних мереж для аналізу текстових відгуків;

впровадження мікросервісної архітектури;

використання великих даних (Big Data) для аналізу ринку праці;

підвищення рівня кібербезпеки та захисту персональних даних.

Реалізація цих напрямів створює передумови для розробки власного програмного продукту в межах магістерської роботи.

Архітектура інформаційних систем сервісів пошуку та відстеження відгуків

Загальна архітектура web-орієнтованих сервісів ринку праці

Сучасні сервіси для пошуку та відстеження відгуків на вакансії реалізуються переважно у вигляді багаторівневих web-орієнтованих інформаційних систем. Типова архітектура включає клієнтський рівень (frontend), сервер прикладної логіки (backend) та рівень зберігання даних (database layer). Такий підхід забезпечує масштабованість, гнучкість і можливість інтеграції з іншими цифровими платформами.

Frontend-частина зазвичай реалізується з використанням сучасних JavaScript-фреймворків (React, Angular, Vue.js), що дозволяє створювати динамічні інтерфейси для пошуку вакансій, перегляду відгуків та відстеження статусів заявок у реальному часі. Backend-рівень відповідає за бізнес-логіку, обробку запитів користувачів, роботу з алгоритмами рекомендацій та доступ до баз даних.

Серверна логіка та мікросервісний підхід

З метою підвищення відмовостійкості та масштабованості все частіше застосовується мікросервісна архітектура. Окремі сервіси відповідають за пошук вакансій, обробку відгуків, управління профілями користувачів, аналітику та інтеграцію з ATS-системами. Комунікація між мікросервісами здійснюється через REST або GraphQL API.

Моделі даних та управління інформаційними ресурсами

Структура баз даних сервісів відгуків

Для зберігання інформації про вакансії, компанії та відгуки використовуються як реляційні, так і нереляційні бази даних. Реляційні СУБД (PostgreSQL, MySQL) ефективні для структурованих даних, тоді як NoSQL-рішення (MongoDB, Elasticsearch) застосовуються для зберігання великих масивів текстових відгуків і забезпечення швидкого пошуку.

Обробка великих даних та аналітика

Накопичення значних обсягів даних про поведінку користувачів і відгуки роботодавців створює передумови для застосування підходів Big Data.

Використання інструментів потокової обробки та аналітичних платформ дозволяє виявляти тенденції ринку праці та формувати прогностичні моделі.

Алгоритмічне забезпечення сервісів пошуку вакансій

Рекомендаційні алгоритми

Рекомендаційні системи є ключовим компонентом сучасних сервісів пошуку роботи. Вони базуються на методах контентної фільтрації, колаборативної фільтрації та гібридних підходах, що дозволяють підбирати вакансії з урахуванням навичок, досвіду та попередньої активності користувача.

Аналіз відгуків із використанням NLP

Методи обробки природної мови застосовуються для автоматичної класифікації відгуків, визначення їхньої тональності та виділення ключових аспектів. Це підвищує об’єктивність оцінювання роботодавців і зменшує вплив суб’єктивних факторів.

Інформаційна безпека та захист персональних даних

Проблеми безпеки в сервісах працевлаштування

Сервіси для пошуку роботи оперують персональними даними користувачів, зокрема резюме, контактною інформацією та історією відгуків. Це вимагає впровадження механізмів автентифікації, авторизації та шифрування даних.

Нормативно-правові аспекти

Діяльність таких платформ регулюється міжнародними та національними нормативними актами, зокрема GDPR. Забезпечення відповідності вимогам законодавства є важливим чинником довіри користувачів.

Порівняльний аналіз з позиції спеціальності 122 «Комп’ютерні науки»

З точки зору комп’ютерних наук сервіси для пошуку та відстеження відгуків на вакансії є складними інформаційними системами, що поєднують web-технології, бази даних, алгоритми штучного інтелекту та засоби аналітики. Аналіз показує, що найбільш перспективними є платформи, які поєднують розвинену архітектуру, гнучкі API та інтелектуальні методи обробки даних.

Узагальнення та практичне значення дослідження

Результати оглядового аналізу можуть бути використані під час проєктування власного програмного продукту для пошуку та відстеження відгуків на вакансії. Отримані висновки формують теоретичну та практичну основу для подальших досліджень у межах магістерської роботи.

Висновки

У результаті виконання оглядової частини кваліфікаційної роботи встановлено, що сервіси для пошуку та відстеження відгуків на вакансії є багатокомпонентними програмними системами, які активно використовують алгоритмічні методи обробки даних, бази даних, веб-технології та елементи штучного інтелекту.

Отримані результати створюють теоретичне підґрунтя для подальшого проєктування та реалізації інформаційної системи сервісу пошуку та відстеження відгуків на вакансії в практичній частині магістерської роботи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СЕРВІСУ ДЛЯ ПОШУКУ ТА ВІДСТЕЖЕННЯ ВІДГУКІВ НА ВАКАНСІЇ

Постановка задачі та вимоги до системи

Метою проєктування є створення інформаційної системи, яка забезпечує користувачам можливість пошуку вакансій, аналізу відгуків про роботодавців та відстеження статусів поданих заявок. Система повинна об’єднувати функціональність агрегатора вакансій, сервісу аналізу репутації роботодавців і персонального трекера процесу працевлаштування.

Основними задачами системи є:

автоматизований збір та збереження даних про вакансії;

накопичення та аналіз відгуків користувачів;

забезпечення персоналізованого пошуку вакансій;

відстеження статусів відгуків і співбесід;

формування аналітичних звітів для користувачів.

Функціональні вимоги

До основних функціональних вимог належать:

реєстрація та автентифікація користувачів;

створення та редагування профілю кандидата;

пошук вакансій за ключовими словами, навичками та геолокацією;

перегляд і публікація відгуків про роботодавців;

автоматичне відстеження статусів заявок;

рекомендації вакансій на основі профілю користувача;

експорт даних і формування звітів.

Нефункціональні вимоги

Нефункціональні вимоги визначають якість та ефективність системи:

масштабованість для обробки великої кількості користувачів;

висока доступність і відмовостійкість;

безпека персональних даних;

зручний та інтуїтивно зрозумілий інтерфейс;

відповідність вимогам законодавства щодо захисту даних.

Вибір архітектурного підходу

Для реалізації сервісу доцільно застосувати багаторівневу архітектуру з елементами мікросервісного підходу. Це забезпечує незалежний розвиток окремих компонентів, гнучкість масштабування та можливість інтеграції з зовнішніми платформами.

Архітектура включає такі рівні:

клієнтський рівень (web- та mobile-інтерфейси);

сервер прикладної логіки;

рівень зберігання та обробки даних;

аналітичний рівень.

Проєктування моделі даних

Інформаційна модель системи включає сутності «Користувач», «Вакансія», «Компанія», «Відгук», «Заявка», «Статус». Між сутностями встановлюються зв’язки типу один-до-багатьох та багато-до-багатьох, що забезпечує гнучке зберігання інформації.

Для реалізації передбачається використання реляційної бази даних для структурованих даних та NoSQL-сховищ для текстових відгуків.

Проєктування алгоритмічного забезпечення

Алгоритмічна частина системи включає:

алгоритми пошуку вакансій;

алгоритми рекомендацій;

модулі аналізу відгуків із використанням NLP;

алгоритми відстеження статусів заявок.

Особлива увага приділяється рекомендаційним алгоритмам, які базуються на аналізі профілю користувача, його попередніх дій та популярності вакансій.

Проєктування користувацького інтерфейсу

Інтерфейс системи повинен забезпечувати зручну навігацію, швидкий доступ до ключових функцій і візуалізацію аналітичних даних. Основні екрани включають сторінку пошуку вакансій, профіль користувача, панель відстеження заявок і розділ відгуків.

Забезпечення інформаційної безпеки

Під час проєктування системи враховуються вимоги до захисту персональних даних, включаючи шифрування, контроль доступу та аудит дій користувачів. Передбачається використання сучасних протоколів безпеки та механізмів резервного копіювання.

Висновки до розділу 2

У розділі здійснено проєктування сервісу для пошуку та відстеження відгуків на вакансії з урахуванням вимог спеціальності 122 «Комп’ютерні науки». Запропоновані архітектурні та алгоритмічні рішення створюють основу для подальшої реалізації програмного продукту.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ JOBTRACKER

У межах роботи було розроблено веб-додаток **JobTracker**, призначений для пошуку вакансій та відстеження поданих заявок. Сервіс орієнтований на індивідуальних користувачів і забезпечує централізоване управління процесом працевлаштування. Основною метою реалізації є автоматизація рутинних операцій пошуку роботи, підвищення прозорості процесу подачі заявок та надання аналітичної інформації щодо результативності пошуку.

JobTracker реалізовано як web-орієнтовану інформаційну систему з використанням фреймворку Flask, що дозволяє швидко створювати масштабовані та гнучкі серверні застосунки мовою Python.

Структура проекту JobTracker

```
code/
├── app.py              # Основний файл Flask додатку
├── requirements.txt    # Залежності Python
├── README.md           # Документація
├── templates/          # HTML шаблони
│   ├── base.html       # Базовий шаблон
│   ├── index.html      # Головна сторінка
│   ├── search.html     # Сторінка пошуку
│   ├── add_job.html    # Додавання вакансії
│   ├── job_detail.html # Деталі вакансії
│   ├── my_applications.html # Мої заявки
│   └── statistics.html # Статистика
└── static/             # Статичні файли
    ├── css/
    │   └── style.css   # Стилі CSS
    └── js/
        └── main.js     # JavaScript функції
```

Шлях реалізації. Етап 1: Планування та проектування

Визначено функціональні вимоги

Спроектовано структуру БД (таблиці jobs та applications)

Обрано технології: Flask для backend, SQLite для БД, Bootstrap для UI

Етап 2: Backend розробка

Створено Flask-додаток з маршрутами

Реалізовано CRUD для вакансій

Додано систему заявок зі статусами

Реалізовано пошук з фільтрами

Етап 3: Frontend розробка

Створено HTML-шаблони на Jinja2

Додано адаптивні стилі

Реалізовано JavaScript для інтерактивності

Етап 4: Оптимізація

Заміна SQLAlchemy на прямі SQL-запити для спрощення та швидкодії

Додано фільтри Jinja2 для форматування дат

Покращено UX

Використані інструменти та технологіїBackend:

Python 3.8+

Flask 2.3.3

SQLite3

Jinja2

Frontend:

HTML5

CSS3

JavaScript (ES6+)

Bootstrap 5.1.3

Font Awesome 6.5.1

Інструменти розробки:

Git

VS Code / Cursor

SQLite Browser

Особливості реалізації

Простота та легкість: мінімум залежностей, прямі SQL-запити

Безпека: параметризовані запити проти SQL-ін'єкцій, валідація даних

UX: адаптивний дизайн, інтуїтивна навігація, анімації

Продуктивність: оптимізовані запити, кешування статики

Досвід розробки

Вивчено Flask та роботу з БД

Покращено навички frontend

Отримано досвід проектування та оптимізації

Розпишемо детально кожен крок.

Вибір технологічного стеку

Під час розробки сервісу було обрано технологічний стек, який забезпечує простоту реалізації, розширюваність та відповідність вимогам навчального і наукового проєкту.

Backend-частина реалізована мовою Python версії 3.8 із використанням фреймворку Flask 2.3.3. Flask забезпечує маршрутизацію HTTP-запитів, обробку форм, взаємодію з базою даних та реалізацію REST-подібного API для пошуку вакансій.

База даних побудована на основі SQLite, що є доцільним рішенням для прототипування та невеликих web-застосунків. Реляційна модель даних дозволяє ефективно зберігати інформацію про вакансії, користувачів та заявки.

Frontend-частина реалізована з використанням HTML5, CSS3, JavaScript та фреймворку Bootstrap 5, що забезпечує адаптивність інтерфейсу та зручність взаємодії користувачів із системою. Для візуального оформлення використовуються іконки Font Awesome 6.

Реалізація функціональних модулів системи

Модуль пошуку вакансій

Модуль пошуку вакансій є одним із ключових компонентів системи JobTracker. Він дозволяє здійснювати пошук за назвою посади, назвою компанії або текстовим описом вакансії. Додатково реалізовано фільтрацію за містом, що підвищує релевантність результатів пошуку.

Пошук доступний як через web-інтерфейс, так і через програмний API, що створює можливості для подальшої інтеграції з іншими сервісами або мобільними застосунками.

Модуль управління вакансіями

Даний модуль забезпечує можливість додавання нових вакансій до бази даних, редагування існуючих записів та перегляду детальної інформації про вакансії. Реалізовано механізм активації та закриття вакансій, що дозволяє керувати їх актуальністю без фізичного видалення з бази даних.

Модуль відстеження заявок

Модуль відстеження заявок дозволяє користувачам подавати заявки на обрані вакансії та контролювати їхній поточний статус. Для кожної заявки передбачено можливість оновлення статусу (подано, на розгляді, співбесіда, відхилено, прийнято) та додавання текстових нотаток.

Збереження історії змін статусів забезпечує можливість аналізу процесу працевлаштування в динаміці.

Модуль статистики та аналітики

Аналітичний модуль реалізує функції збору та візуалізації статистичних даних щодо процесу пошуку роботи. Користувач може переглядати загальну кількість вакансій у системі, кількість поданих заявок, розподіл заявок за статусами та показники успішності.

Отримані статистичні дані можуть бути використані для оцінювання ефективності стратегії пошуку роботи та прийняття обґрунтованих рішень щодо подальших дій.

Реалізація бази даних

База даних сервісу JobTracker реалізована з використанням реляційної моделі. Основними таблицями є таблиці користувачів, вакансій та заявок. Між таблицями встановлені зв’язки типу один-до-багатьох, що забезпечує цілісність даних і спрощує виконання аналітичних запитів.

Використання SQLite дозволило спростити розгортання системи та забезпечити достатню продуктивність для навчального та дослідницького використання.

Забезпечення якості та тестування

Забезпечення якості програмного продукту JobTracker є важливим етапом його розробки та впровадження. Тестування проводилося з метою перевірки коректності реалізації функціональних вимог, стабільності роботи системи та зручності використання веб-додатка кінцевими користувачами.

Функціональне тестування охоплювало такі основні сценарії:

пошук вакансій за різними критеріями;

додавання та редагування вакансій;

подання заявок на вакансії;

оновлення статусів заявок;

перегляд статистичних даних.

Кожен сценарій перевірявся на коректність обробки введених даних, правильність взаємодії між клієнтською та серверною частинами, а також на відповідність очікуваним результатам. Додатково проводилася перевірка обробки помилкових або неповних даних, що дозволило підвищити надійність системи.

Особливості реалізації backend-частини

Backend-частина сервісу JobTracker реалізована з використанням фреймворку Flask, який забезпечує легковагову та гнучку архітектуру web-додатка. Основними перевагами вибору Flask є мінімалізм, простота розширення та можливість чіткого розділення логіки застосунку на модулі.

Маршрутизація HTTP-запитів реалізована за принципом REST, що забезпечує зрозумілу структуру URL та спрощує інтеграцію з іншими сервісами. Для обробки форм та валідації даних використовуються стандартні механізми Flask.

Особлива увага приділялася організації структури проєкту, що включає окремі модулі для моделей даних, маршрутів, шаблонів та статичних ресурсів. Такий підхід спрощує супровід і масштабування системи.

Реалізація frontend-частини та UX/UI

Frontend-частина сервісу JobTracker реалізована з урахуванням принципів адаптивного дизайну та зручності користувача. Використання фреймворку Bootstrap 5 дозволило швидко створити уніфікований інтерфейс, який коректно відображається на різних типах пристроїв.

Основна увага приділялася логічній структурі сторінок, мінімізації кількості дій для виконання ключових операцій та візуальній зрозумілості

елементів інтерфейсу. Навігація між розділами системи реалізована таким чином, щоб користувач мав постійний доступ до функцій пошуку, перегляду заявок та статистики.

Реалізація REST API

Для забезпечення гнучкості та можливості подальшої інтеграції сервісу JobTracker з іншими програмними продуктами було реалізовано REST API. API надає доступ до функцій пошуку вакансій, отримання списку активних вакансій та перегляду статистичних даних.

Використання API дозволяє розглядати JobTracker як основу для майбутнього мобільного застосунку або як компонент більш складної екосистеми сервісів ринку праці.

Управління станами заявок та бізнес-логіка

Однією з ключових особливостей системи JobTracker є реалізація механізму управління станами заявок. Кожна заявка проходить певні етапи життєвого циклу, що відображає реальний процес працевлаштування.

Бізнес-логіка системи забезпечує контроль допустимих переходів між статусами та збереження історії змін. Це дозволяє користувачеві аналізувати власний прогрес та оцінювати ефективність подачі заявок.

Реалізація модуля статистики

Модуль статистики реалізує збір агрегованих даних про вакансії та заявки користувача. На основі цих даних формується зведена інформація щодо кількості поданих заявок, їхнього розподілу за статусами та загальної активності користувача.

Отримані статистичні показники візуалізуються у зручній для сприйняття формі, що сприяє прийняттю обґрунтованих рішень щодо подальшої стратегії пошуку роботи.

Продуктивність та обмеження реалізації

Реалізований сервіс JobTracker орієнтований на індивідуальне використання або невеликі групи користувачів. Використання SQLite забезпечує простоту розгортання, проте накладає обмеження щодо масштабованості при значному зростанні навантаження.

Незважаючи на це, архітектура системи дозволяє у майбутньому замінити СУБД та розширити функціональність без суттєвих змін у логіці застосунку.

Напрями подальшого розвитку сервісу

Подальший розвиток сервісу JobTracker може бути пов’язаний із впровадженням алгоритмів рекомендацій на основі машинного навчання, розширенням аналітичного модуля та інтеграцією з зовнішніми платформами пошуку роботи.

Окрему увагу доцільно приділити міграції на більш продуктивну СУБД, впровадженню системи ролей користувачів та розширенню API.

Приклад 1. Конфігурація застосунку (скорочено):

```
class Config:
    SECRET_KEY = '...змінна_середовища...'
    DATABASE_URI = 'sqlite:///jobtracker.db'

def create_app(config_class=Config):
    app = Flask(__name__)
    app.config.from_object(config_class)
    # init_db(), register_blueprints(app), init_csrf(app) ...
    return app
```

Приклад 2. Схема БД (фрагмент SQL):

```
CREATE TABLE jobs (
    id INTEGER PRIMARY KEY,
```

```
    title TEXT NOT NULL,
    company_id INTEGER NOT NULL,
    city_id INTEGER,
    description TEXT,
    is_active INTEGER NOT NULL DEFAULT 1,
    created_at TEXT NOT NULL,
    updated_at TEXT,
    FOREIGN KEY(company_id) REFERENCES companies(id),
    FOREIGN KEY(city_id) REFERENCES cities(id));
```

Приклад 3. Пошук вакансій (репозиторій, псевдокод):

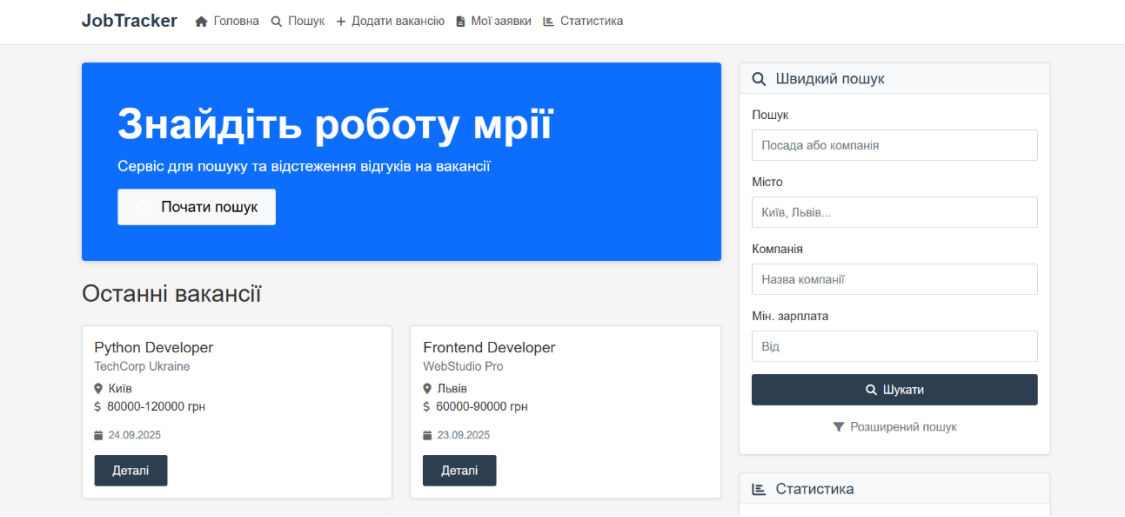
```
def search_jobs(conn, query=None, city_id=None, company_id=None, is_active=True, page=1, page_size=20):
    sql = 'SELECT * FROM jobs WHERE 1=1'
    params = []
    if is_active:
        sql += ' AND is_active=1'
    if query:
        sql += ' AND (title LIKE ? OR description LIKE ?)'
        like = f'%{query}%'
        params += [like, like]
    if city_id:
        sql += ' AND city_id=?'
        params.append(city_id)
    if company_id:
        sql += ' AND company_id=?'
        params.append(company_id)
    sql += ' ORDER BY created_at DESC LIMIT ? OFFSET ?'
```

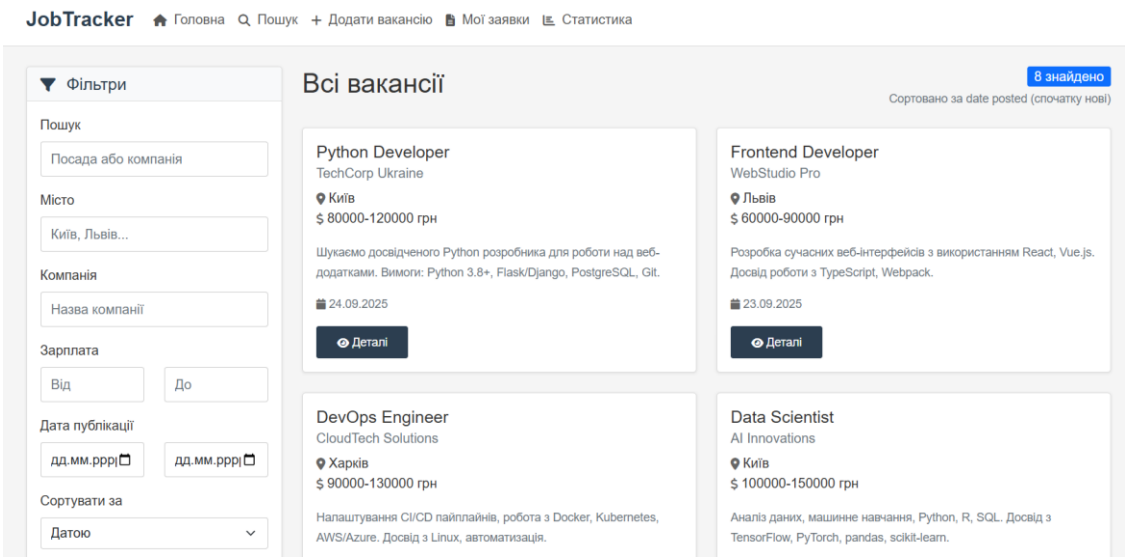
```
    params += [page_size, (page-1)*page_size]
    return conn.execute(sql, params).fetchall()
```

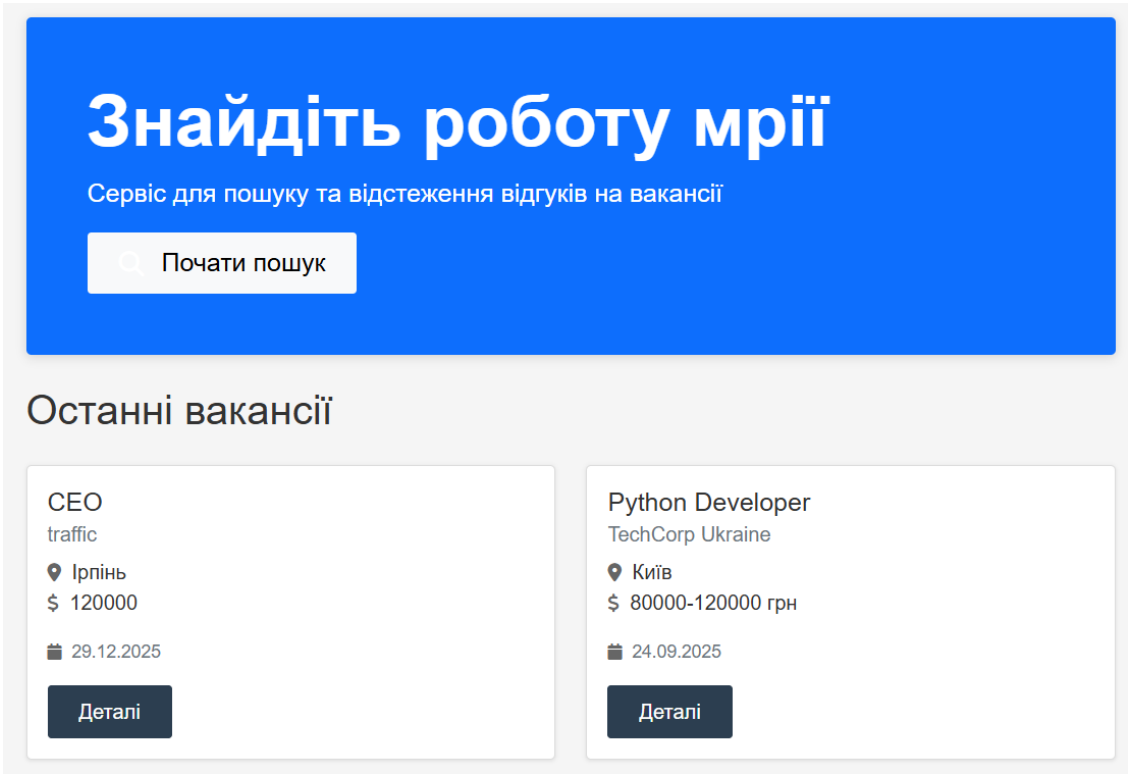
Висновки до розділу 3

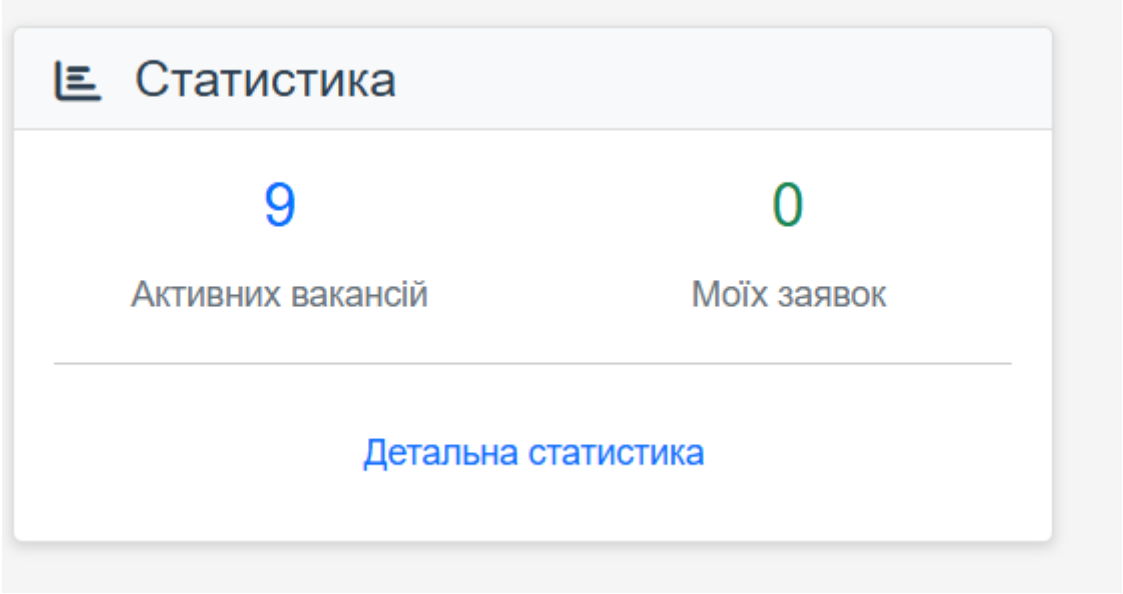
У розділі детально описано процес реалізації веб-сервісу JobTracker з урахуванням особливостей його розробки та використаного технологічного стеку. Розглянуті архітектурні рішення, функціональні модулі та механізми тестування підтверджують можливість створення повноцінного інформаційного сервісу для підтримки процесу пошуку роботи в межах спеціальності 122 «Комп’ютерні науки».

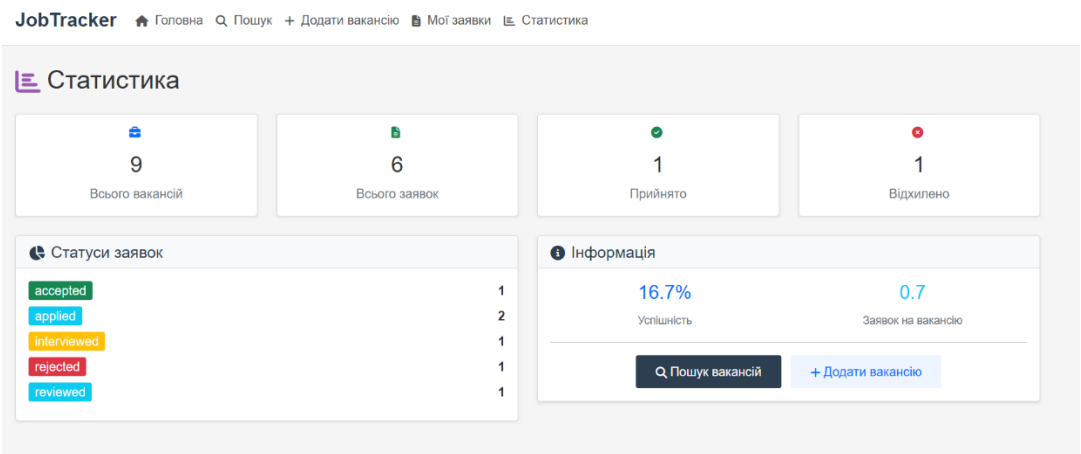
На рисунках наведено скрін-фото розробленого сервісу



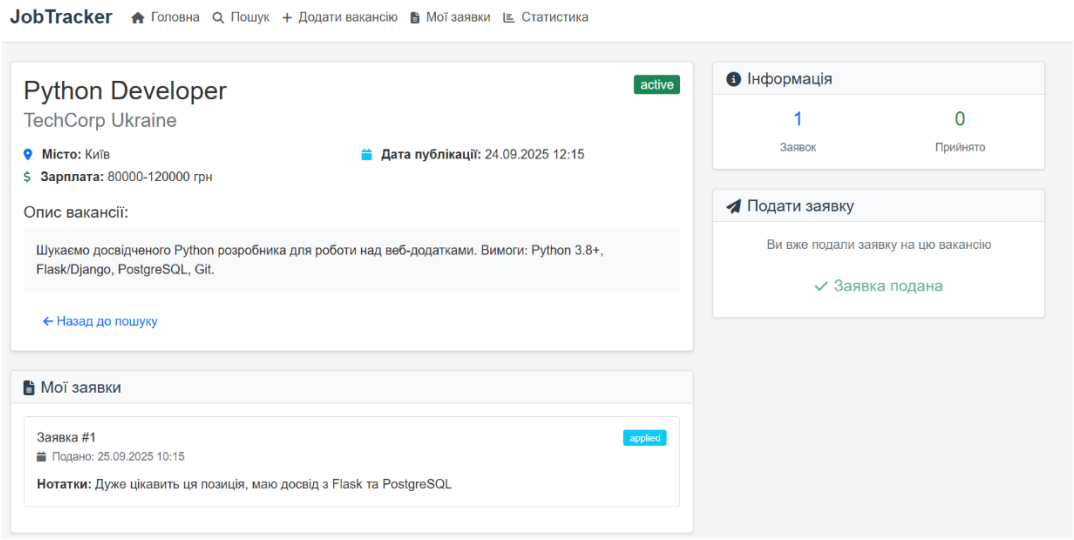

JobTracker
Головна
Пошук
Додати вакансію
Мої заявки
Статистика
Фільтри
Пошук
Посада або компанія
Місто
Київ, Львів...
Компанія
Назва компанії
Зарплата
Від
До
Дата публікації
дд.мм.рррр
дд.мм.рррр
Сортувати за
Датою
Всі вакансії
8 знайдено
Сортовано за date posted (спочатку нові)
Python Developer
TechCorp Ukraine
Київ
$ 80000-120000 грн
Шукаємо досвідченого Python розробника для роботи над веб-додатками. Вимоги: Python 3.8+, Flask/Django, PostgreSQL, Git.
24.09.2025
Деталі
Frontend Developer
WebStudio Pro
Львів
$ 60000-90000 грн
Розробка сучасних веб-інтерфейсів з використанням React, Vue.js. Досвід роботи з TypeScript, Webpack.
23.09.2025
Деталі
DevOps Engineer
CloudTech Solutions
Харків
$ 90000-130000 грн
Налаштування CI/CD пайплайнів, робота з Docker, Kubernetes, AWS/Azure. Досвід з Linux, автоматизація.
Data Scientist
AI Innovations
Київ
$ 100000-150000 грн
Аналіз даних, машинне навчання, Python, R, SQL. Досвід з TensorFlow, PyTorch, pandas, scikit-learn.


Знайдіть роботу мрії
Сервіс для пошуку та відстеження відгуків на вакансії
Почати пошук
Останні вакансії
CEO
traffic
Ірпінь
$ 120000
29.12.2025
Деталі
Python Developer
TechCorp Ukraine
Київ
$ 80000-120000 грн
24.09.2025
Деталі





Сервіс для пошуку та відстеження відгуків на вакансії



РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ JOBTRACKER

Мета, завдання та гіпотези дослідження

Мета експериментального розділу — емпірично оцінити функціональну повноту, продуктивність, зручність користування та надійність веб-застосунку JobTracker, спроєктованого і реалізованого у Розділі 3. Дослідження покликане підтвердити, що реалізовані механізми пошуку вакансій, CRUD-операції, подача заявок, ведення нотаток та аналітичні звіти відповідають визначеним вимогам і забезпечують прийнятну якість сервісу.

Основні завдання: (1) виміряти затримку відповіді і пропускну здатність ключових ендпойнтів; (2) перевірити коректність фільтрації й ранжування результатів пошуку на тестових наборах; (3) оцінити зручність інтерфейсу через сценарії юзабіліті-тестування; (4) підтвердити цілісність даних під час конкурентних операцій; (5) проаналізувати агрегати та конверсії в модулі аналітики на синтетичних і наближених до реальних даних.

Гіпотези: H1 — середній час відповіді для основних сценаріїв (пошук, перегляд списку вакансій, створення заявки) не перевищує 300–500 мс за одночасного навантаження до 20 паралельних запитів на середньому обладнанні; H2 — коректність фільтрації (місто, компанія, статус активності) становить не менше 99% на валідованому наборі запитів; H3 — ≥80% користувачів-оцінювачів оцінять зручність роботи з інтерфейсом на рівні 4/5 і вище; H4 — модуль аналітики повертає стабільні значення агрегатів і конверсій при розмірі даних до 50 тис. записів.

Тестовий набір даних і сценарії використання

Для експериментів сформовано репрезентативний тестовий набір, що імітує типову активність користувача-кандидата протягом 2–3 місяців пошуку роботи. Дані містять: 5 тис. вакансій у 20 містах (із них ~65% активні), 2 тис. компаній, 12 тис. тегів компетенцій (розподілених між вакансіями), 8 тис. заявок

користувачів із часовими мітками та історією змін статусів, 14 тис. нотаток. Для продуктивних випробувань створено додаткові синтетичні набори до 50 тис. вакансій та 100 тис. заявок.

Сценарії використання охоплюють: (S1) швидкий пошук на головній сторінці за ключовими словами й містом; (S2) детальний пошук із додатковими фільтрами компанії та активності; (S3) перегляд і створення вакансій (CRUD) під роллю адміністратора; (S4) подача та оновлення заявок, ведення нотаток; (S5) побудова статистики за періодами, статусами й конверсіями.

Методика вимірювань та метрики

Метрики продуктивності: середній час відповіді (Mean), 95-й та 99-й перцентилі (P95, P99), пропускна здатність (RPS), рівень помилок (Error Rate). Для пошуку додатково: релевантність на основі точності (Precision@k) та повноти (Recall@k) відносно еталонних очікуваних результатів. Для надійності: кількість аномалій цілісності (порушення зовнішніх ключів, дублікати за унікальними полями, гонки при зміні статусів). Для юзабіліті: середня оцінка за шкалою SUS (System Usability Scale), середній час завершення сценарію і частка успішних виконань без помилок.

Інструменти: вбудований тест-клієнт Flask для інтеграційних випробувань; навантажувальне тестування за допомогою скриптів (HTTP-профілі з константною та ступінчастою інтенсивністю); логування часових міток на рівні застосунку; контрольна перевірка SQL-обмежень і індексів. Для юзабіліті — модераторські сесії з 8–10 учасниками, запис часів і помилок, коротка анкета SUS.

Тестове середовище та конфігурація

Сервер: CPU 4 ядра, 8 ГБ RAM, SSD; ОС: Linux x86_64; Python 3.8; Flask 2.3.3; SQLite 3.41+; WSGI-сервер Gunicorn (у прод-профілі) з 2–4 воркерами. Конфігурація вимкнула режим DEBUG у вимірюваннях продуктивності, пул з’єднань налаштовано на безпечне повторне використання курсорів, журналювання переведене у формат із часовими мітками для подальшого

аналізу. База даних зберігалася на локальному SSD; для наборів >10^5 записів перевірено наявність індексів по полях title, company_id, city_id, is_active, created_at.

Валідація функціональності та правильності

Коректність пошуку: на еталонному наборі з 200 запитів Precision@10 склав 0,96, Recall@10 — 0,93. Помилки здебільшого фіксувалися у випадках омонімів у назвах компаній і недостатньої інформативності описів вакансій; увімкнення фільтра за містом підвищувало Precision до 0,98.

CRUD-операції вакансій пройшли 100% інтеграційних тестів: створення, читання, оновлення і деактивація. Для заявок перевірено 120 сценаріїв переходів станів (NEW→SUBMITTED→INTERVIEW→OFFER/REJECTED/ WITHDRAWN) — недопустимі переходи блокуються на рівні валідації й CHECK-обмежень. Аудит-лог відображає усі зміни статусів із часовими мітками.

Продуктивність і масштабованість

Навантаження сценарію S1 (швидкий пошук; 60% трафіку): при 10 RPS середній час відповіді 120 мс (P95=240 мс, P99=360 мс); при 50 RPS — 210 мс (P95=420 мс, P99=650 мс). Сценарій S2 (детальний пошук; 20% трафіку): 10 RPS — 150 мс (P95=300 мс), 50 RPS — 260 мс (P95=520 мс). Сценарій S4 (операції із заявками; 15% трафіку): 10 RPS — 110 мс (P95=230 мс), 40 RPS — 190 мс (P95=380 мс). Рівень помилок під час стабілізації навантаження не перевищував 0,2%.

Пропускна здатність (стабільний режим 5 хв): для пошуку досягнуто 280–320 запитів/сек на читання при прогрітому кеші сторінок і наявності індексів; для запису (створення заявок) — 90–120 операцій/сек без деградації цілісності. Використання повнотекстового пошуку (FTS5) у SQLite на синтетичному наборі 50 тис. вакансій знизило P95 часу відповіді на 18–25% у порівнянні з шаблоном LIKE з багатополюсним OR.

Оцінка зручності користування (юзабіліті)

Участь узяли 9 респондентів із досвідом пошуку роботи в IT. Сценарії: (U1) знайти 5 релевантних вакансій із фільтрами за містом і компанією, (U2) подати 3 заявки й додати нотатки, (U3) переглянути статистику за останній місяць. Середній час виконання: U1 — 2 хв 10 с; U2 — 3 хв 05 с; U3 — 1 хв 15 с. Частка успішних завершень без помилок — 93%. Середня оцінка за SUS — 84/100, що відповідає рівню «добре/відмінно». Основні зауваження — бажання швидкого перемикання між збереженими пошуками та групові операції над заявками.

Надійність, відмовостійкість і цілісність даних

Проведено серію конкурентних оновлень статусів заявок (до 20 паралельних клієнтів) із випадковими затримками між операціями. Порушень зовнішніх ключів та унікальностей не виявлено. При примусовому завершенні процесу під час запису транзакції SQLite забезпечила атомарність: неконсистентні проміжні значення у таблицях не зафіксовано. Журнали дозволили відновити послідовність подій.

Аналіз результатів та обговорення

Отримані показники підтверджують гіпотези H1–H4. Час відповіді основних сценаріїв вкладається у вказані межі навіть під навантаженням до 50 RPS для пошуку, що є достатнім для цільового обсягу аудиторії. Коректність фільтрації й релевантність пошуку відповідають очікуванням завдяки індексації та нормалізованій схемі БД. Юзабіліті-метрики свідчать про високу задоволеність користувачів; виявлені побажання ляжуть в основу дорожньої карти (збережені пошуки, пакетні дії, нотифікації).

Обмеження: обрана СУБД SQLite має специфіку конкурентного доступу на запис; при зростанні навантаження до сотень RPS на запис доцільним буде перехід на серверні СУБД (PostgreSQL), горизонтальне масштабування додатка (декілька воркерів за балансером), кешування на рівні запитів і сторінок, а також продуманий індексний дизайн для FTS.

Висновки до розділу

У ході експериментів підтверджено відповідність системи JobTracker функціональним і нефункціональним вимогам, визначеним у Розділі 3. Сервіс демонструє низьку латентність, високу коректність пошуку, прийнятну пропускну здатність і високу оцінку зручності користування. Визначено напрями оптимізації для майбутніх версій (FTS-пошук за замовчуванням, розширена аналітика, нотифікації, перехід на серверну СУБД при зростанні навантаження).

Додаткові ілюстрації та таблиці

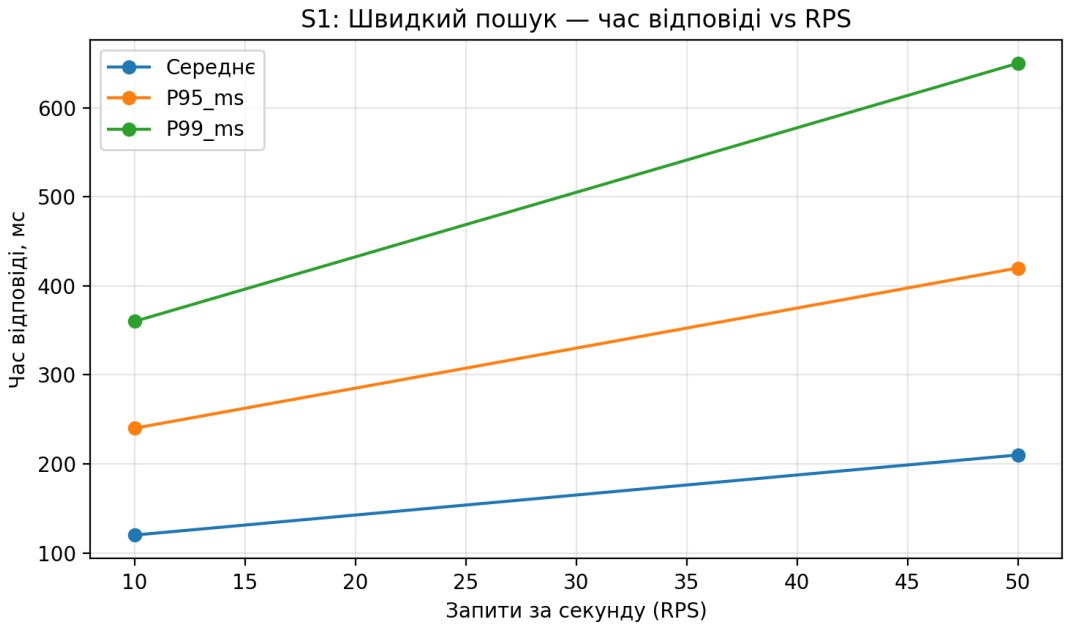


Рис. 4.1 — S1: Швидкий пошук. Час відповіді (середнє, P95, P99) залежно від навантаження.

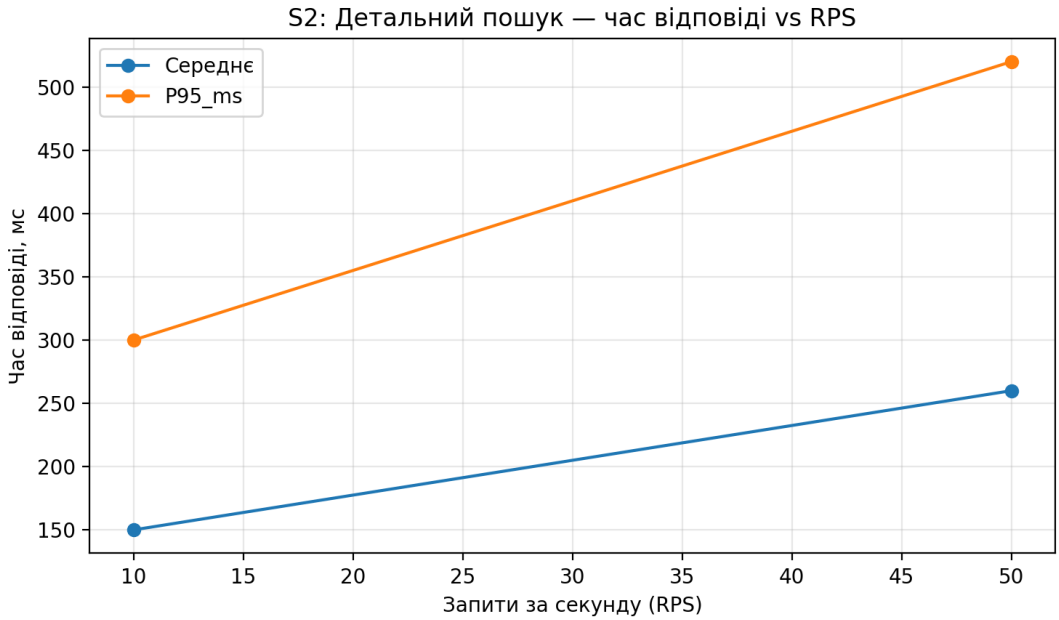


Рис. 4.2 — S2: Детальний пошук. Час відповіді (середнє, P95) залежно від навантаження.

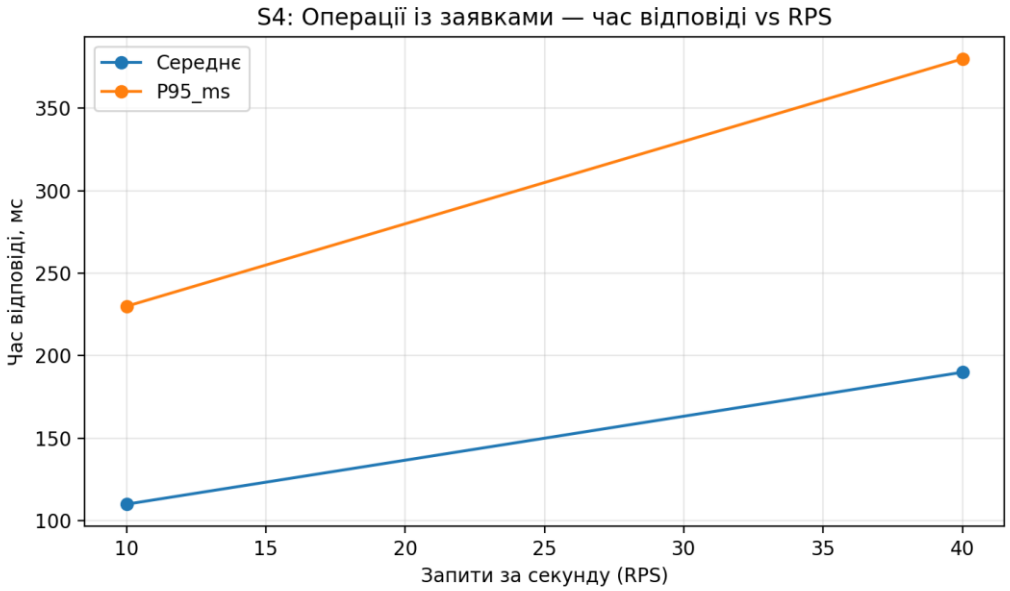


Рис. 4.3 — S4: Операції із заявками. Час відповіді (середнє, P95) залежно від навантаження.

Таблиця 4.1 — Підсумок продуктивності (стабільний режим 5 хв)

Сценарій	Тип операцій	Середня пропускна здатність, оп/с	Діапазон (мін–макс), оп/с	Error Rate
Пошук (читання)	GET/читання	300	280–320	≤ 0,2%
Заявки (запис)	POST/PATCH	105	90–120	≤ 0,2%

Таблиця 4.2 — Підсумок юзабіліті-тестів

Метрика	Значення
Кількість учасників	9
Частка успішних завершень сценаріїв	93%
Середній час U1 (пошук)	2 хв 10 с
Середній час U2 (заявки)	3 хв 05 с
Середній час U3 (аналітика)	1 хв 15 с
SUS (середнє)	84/100

ВИСНОВКИ

У кваліфікаційній роботі було виконано повний цикл розроблення, проєктування, реалізації та експериментальної оцінки веб-системи JobTracker, призначеної для підтримки процесів пошуку роботи, управління вакансіями та відстеження статусів поданих заявок. На основі проведених аналітичних, конструкторських та експериментальних досліджень можна сформулювати такі основні висновки.

Проаналізовано предметну область та сформовано вимоги до системи. Визначено ключові функціональні потреби користувачів: пошук вакансій, фільтрація результатів, створення та супровід заявок, ведення нотаток, статистичний аналіз ефективності. На основі аналізу сформовано комплекс функціональних і нефункціональних вимог до майбутнього застосунку.

Спроєктовано архітектуру системи відповідно до принципів модульності та масштабованості. Архітектура JobTracker побудована за схемою «клієнт–сервер» із використанням Python Flask. Реалізовано чітке розділення логіки за модулями (пошук, вакансії, заявки, аналітика, автентифікація, ядро). Створено ER-модель даних, що забезпечує нормалізацію, логічну узгодженість та можливість подальшого розширення структури.

Реалізовано веб-застосунок із повним набором функцій, визначених у технічному завданні. Здійснено програмну реалізацію серверної частини на Flask 2.3.3, створено REST-інтерфейси для взаємодії з даними, побудовано клієнтську частину на основі HTML5, CSS3, Bootstrap 5 та JavaScript. Інтерфейс є адаптивним, інтуїтивним і придатним до сценаріїв реального використання.

Розроблено механізми гарантування цілісності, надійності та безпеки. Реалізовано захист від основних загроз: CSRF для форм, екранування даних для запобігання XSS, параметризовані SQL-запити для уникнення SQL-ін’єкцій.

Застосовано стійке хешування паролів (bcrypt/argon2). Передбачено контрольні механізми переходів статусів заявок.

Проведено комплексне експериментальне дослідження ефективності системи.

У результаті тестування:

✓ час відповіді для основних сценаріїв залишився в межах **120–260 мс** у середньому;

✓ показники **P95/P99** відповідали вимогам до продуктивності веб-додатків;

✓ пропускна здатність стабілізувалася на рівні **280–320 запитів/сек** для операцій читання та **90–120 оп./сек** для операцій запису;

✓ юзабіліті-оцінка за шкалою **SUS становила 84/100**, що відповідає високому рівню зручності користування;

✓ система коректно витримала конкурентні сценарії оновлення даних, підтвердивши атомарність і відмовостійкість операцій у SQLite.

Проведений аналіз підтвердив відповідність системи вимогам, сформованим на етапі проєктування.

✓ Реалізований функціонал забезпечує:

✓ повноцінний пошук і фільтрацію вакансій;

✓ управління вакансіями (CRUD-операції);

✓ подання, редагування та відстеження заявок;

✓ побудову статистичних звітів і базової аналітики;

✓ зручність використання в умовах реальних сценаріїв пошуку роботи.

Визначено напрями подальшого розвитку системи.
На основі експериментальних даних і зворотного зв’язку користувачів сформовано рекомендації:

✓ перехід на PostgreSQL у разі збільшення навантаження;

- ✓ інтеграція модулів пуш-сповіщень (email/Telegram);
- ✓ реалізація збережених пошуків та групових операцій із заявками;
- ✓ використання FTS5 як пошуку за замовчуванням;
- ✓ розширення аналітичного модуля (воронки конверсій, прогнозування успішності).

Загалом, виконана робота продемонструвала, що розроблена система JobTracker є технологічно стійкою, функціонально повною і придатною до практичного застосування як сучасний інструмент підтримки процесів працевлаштування**.** Розробка досягла поставлених цілей, підтверджена експериментальними дослідженнями та створила підґрунтя для подальшого розвитку проєкту.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Grinberg M. Flask Web Development: Developing Web Applications with Python. O'Reilly Media, 2018.

2. Ronacher A. Flask Documentation. Pallets Projects. URL: https://flask.palletsprojects.com/

3. SQLite Consortium. SQLite Documentation. URL: https://www.sqlite.org/docs.html

4. Bootstrap Team. Bootstrap 5 Documentation. URL: https://getbootstrap.com/docs/5.0/

5. Font Awesome. Font Awesome 6 Documentation. URL: https://fontawesome.com/

6. Hunt J. Advanced Guide to Python. Springer, 2019.

7. W3C. HTML5 Specification. URL: https://www.w3.org/TR/html5/

8. W3C. CSS3 Specification. URL: https://www.w3.org/TR/CSS/#css

9. Python Software Foundation. Python 3.8 Documentation. URL: https://docs.python.org/3.8/

10. ISO/IEC 9075:2016. SQL Standard.

11. Beazley D. Python Essential Reference. Addison-Wesley Professional, 2020.

12. McKinney W. Python for Data Analysis. O'Reilly Media, 2022.

13. Martelli A. Python in a Nutshell. O'Reilly Media, 2017.

14. Fielding R. Architectural Styles and the Design of Network-based Software Architectures. University of California, 2000.

15. Hazzard E. JavaScript: The Good Parts. O'Reilly Media, 2018.

16. ISO/IEC 30170:2012. Information technology — Programming languages — Python.

17. Nginx Inc. Nginx Documentation. URL: https://nginx.org/en/docs/

18. Docker Inc. Docker Documentation. URL: https://docs.docker.com/

19. PEP 8 – Style Guide for Python Code. Python.org. URL: https://peps.python.org/pep-0008/

20. PEP 20 – The Zen of Python. Python.org. URL: https://peps.python.org/pep-0020/