Пригожев Олександр Сергійович, кандидат технічних наук, доцент, доцент кафедри інженерії програмного забезпечення Національного університету «Одеська політехніка», Одеса, Україна https://orcid.org/0000-0001-8532-8897

Коваленко Євген Олександрович, магістр з інженерії програмного забезпечення, Одеса, Україна

Prygozhev Oleksandr, Candidate of technical sciences, Assosiate Professor, Associate Professor in Software Engeneering Department Odessa Polytechnic National University, Odessa, Ukraine https://orcid.org/0000-0001-8532-8897

Kovalenko Yevhen, Master of science in software engineering, Odesa, Ukraine

ІНТЕЛЕКТУАЛЬНИЙ ВЕБЗАСТОСУНОК ДЛЯ ПОРІВНЯННЯ РЕЗЮМЕ З ВАКАНСІЯМИ ТА НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ЇХ ПОКРАЩЕННЯ

INTELLIGENT WEB APPLICATION FOR COMPARING RESUMES WITH JOB POSTINGS AND PROVIDING IMPROVEMENT RECOMMENDATIONS

Анотація

Метою роботи є підвищення рівня задоволеності пошукачів роботи процесом працевлаштування шляхом створення інтелектуального вебзастосунку, який забезпечує автоматизований аналіз резюме, порівняння його з вимогами вакансій та формування персоналізованих рекомендацій для покращення.

Методи розробки вебзастосунку ґрунтуються на використанні сучасних вебтехнологій та алгоритмів обробки природної мови. Для серверної частини застосовано мову програмування Python у поєднанні з фреймворком FastAPI. Клієнтська частина створена з використанням JavaScript та бібліотеки React.js із застосуванням Redux Toolkit для керування станом та Bootstrap для адаптивної стилізації інтерфейсу.

У результаті розроблено вебзастосунок, який дозволяє автоматизувати аналіз резюме, виконувати їх порівняння з вакансіями, формувати рейтинг відповідності та надавати персоналізовані рекомендації для покращення.

Ключові слова: резюме, вакансії, обробка природної мови, Python, FastAPI, PostgreSQL, React, JavaScript, ReportLab.

Abstract

The purpose of the work is to increase the level of job seekers' satisfaction with the employment process by creating an intelligent web application that provides automated resume analysis, comparison with job requirements, and the generation of personalized recommendations for improvement.

The methods of web application development are based on the use of modern web technologies and natural language processing algorithms. For the server side, the Python programming language was used in combination with the FastAPI framework. The client side was developed using JavaScript and the React.js library with Redux Toolkit for state management and Bootstrap for adaptive interface styling.

As a result, a web application was developed that enables automated resume analysis, comparison with vacancies, generation of a matching score, and provision of personalized recommendations for improvement.

Keywords: resume, vacancies, natural language processing, Python, FastAPI, PostgreSQL, React, JavaScript, ReportLab.

Постановка проблеми

У сучасному ринку праці конкуренція між кандидатами постійно зростає, і роботодавці висувають все вищі вимоги до резюме. Якість та структура резюме відіграють ключову роль у процесі працевлаштування, адже саме вони є першим інструментом оцінки кандидата. Водночас багато пошукачів роботи не володіють достатніми знаннями про те, як правильно оформити своє резюме та підкреслити навички відповідно до вимог вакансій. Це призводить до зниження шансів навіть кваліфікованих спеціалістів отримати запрошення на співбесіду.

Існуючі онлайн-інструменти для створення резюме здебільшого надають лише шаблони або базове форматування, але не забезпечують інтелектуального аналізу та рекомендацій щодо змісту. Це створює потребу у розробці системи, яка зможе автоматично оцінювати резюме, порівнювати його з конкретними вакансіями та формувати персоналізовані поради для покращення.

На ринку існує низка онлайн-сервісів та програмних платформ, які допомагають користувачам створювати або покращувати резюме, а також шукати вакансії. Проте більшість із них реалізують лише окремі функції і не забезпечують комплексного підходу до порівняння резюме з описом вакансій та формування персоналізованих рекомендацій.

Jobscan – один із найпопулярніших сервісів у світі, який спеціалізується саме на аналізі резюме [1]. Він дозволяє завантажити резюме та опис вакансії, після чого система обчислює відсоткову відповідність та виділяє ключові слова, яких бракує. Перевагою Jobscan є точність підбору ключових слів та зрозуміла візуалізація. Недоліком можна вважати те, що сервіс орієнтований здебільшого на англомовний ринок та має платну підписку, яка обмежує функціонал безкоштовної версії.

Zety – популярний конструктор резюме, який надає користувачам

широкий вибір сучасних шаблонів та підказок щодо заповнення [2]. Система генерує резюме на основі відповідей користувача, що зручно для новачків. Проте Zety більше фокусується на візуальному оформленні, ніж на глибинному аналізі змісту чи рекомендаціях щодо відповідності конкретній вакансії.

LinkedIn є глобальною професійною мережею, що поєднує можливості публікації резюме (у форматі профілю) та пошуку вакансій [3]. Алгоритми LinkedIn автоматично пропонують релевантні вакансії, виходячи з даних профілю користувача, а також показують ступінь відповідності. Проте ці рекомендації мають узагальнений характер, система не надає детальних порад щодо зміни чи покращення резюме.

Work.ua – одна з найпопулярніших українських платформ з пошуку роботи [4]. Вона дозволяє створювати резюме в системі та відгукуватися на вакансії. Djinni, у свою чергу, є нішевою платформою, орієнтованою на IT-фахівців. Обидва сервіси зручні для пошуку роботи, проте вони не надають інструментів для глибокого аналізу чи оптимізації резюме.

Аналіз показує, що сучасні сервіси переважно поділяються на дві категорії:

- редактори резюме (Zety, Canva, Novoresume), які роблять акцент на дизайні документа, але не аналізують його зміст;

- аналітичні інструменти (Jobscan, SkillSyncer), що проводять поверхневий аналіз текстів і працюють лише з ключовими словами.

Немає рішень, які б одночасно дозволяли автоматично завантажувати та парсити резюме у різних форматах, використовували сучасні NLP-моделі для оцінки не лише ключових слів, але й семантичної відповідності; формували детальні персоналізовані рекомендації, надавали користувачам можливість відстежувати прогрес та покращення резюме з часом.

Це підкреслює доцільність створення інтелектуального вебзастосунку,

який би поєднував функції аналізу, рекомендацій та зручного інтерфейсу для роботи з резюме, орієнтованого на користувачів українського та міжнародного ринку.

Метою дослідження є підвищення рівня задоволеності пошукачів роботи процесом працевлаштування шляхом створення інтелектуального вебзастосунку, який забезпечує автоматизований аналіз резюме, порівняння його з вимогами вакансій та формування персоналізованих рекомендацій для покращення.

Результати

Інформаційна модель є основою для проєктування програмної системи, оскільки вона визначає структуру даних, їх взаємозв'язки та механізми взаємодії між користувачем і вебзастосунком. Для інтелектуального сервісу з аналізу та порівняння резюме з вакансіями важливим завданням є забезпечення зберігання різних типів інформації (резюме, вакансії, результати аналізу, рекомендації) та підтримка гнучкої взаємодії між цими сутностями.

Основними об'єктами інформаційної моделі виступають:

- користувач (User) – містить персональні дані, інформацію про роль (пошукач роботи, HR, адміністратор), історію дій у системі;
- резюме (Resume) – включає розділи про освіту, досвід роботи, навички, проєкти, сертифікати та інші важливі відомості. Користувач може завантажувати документ у різних форматах (PDF, DOCX, TXT) або створювати його безпосередньо в редакторі системи;
- вакансія (Vacancy) – опис посади, вимог до кандидата, навичок та досвіду, а також інформацію про компанію; джерела отримання вакансій – ручне додавання або імпорт через API зовнішніх платформ (LinkedIn, Work.ua, Djinni);
- результат аналізу (AnalysisResult) – дані про відсоток відповідності

резюме вакансії, список збігів та прогалин, ключові слова, які необхідно додати;

- рекомендації (Recommendation) – автоматично сформовані поради щодо покращення змісту резюме: виділення релевантних навичок, оптимізація формулювань, додавання ключових слів, побудова альтернативної версії документа;

- адміністративні дані (AdminData) – містять інформацію про модерацію користувачів і вакансій, а також статистику використання сервісу.

Інформаційна модель застосунку показана на рис. 1.

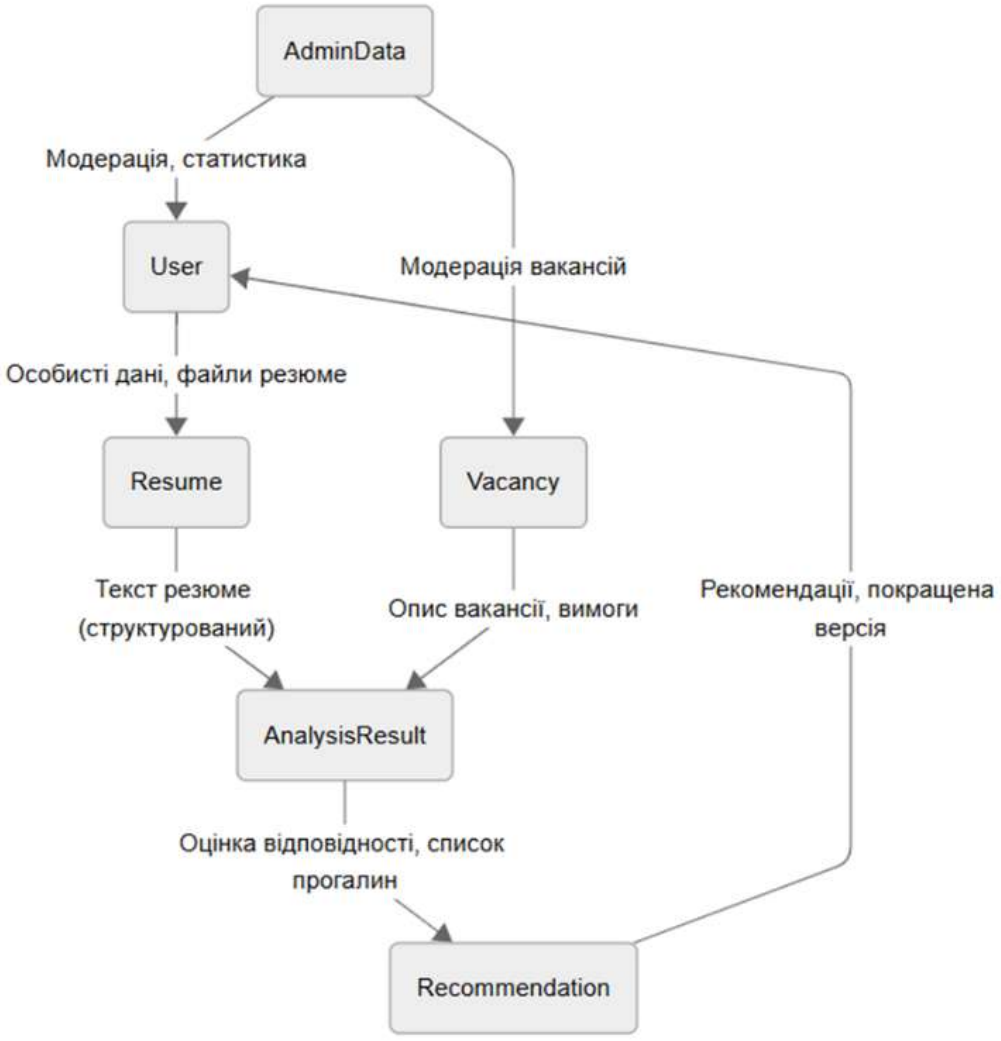


Рис. 1 – Інформаційна модель застосунку

Центральним елементом є користувач (User), який може виступати в ролі пошукача роботи, HR-спеціаліста або адміністратора. Користувач передає до системи особисті дані та файли резюме у різних форматах (PDF, DOCX, TXT), які після завантаження перетворюються у структуровану форму та зберігаються у сутності Resume.

Подальша взаємодія відбувається через порівняння резюме з описами вакансій. Вакансії надходять до системи двома шляхами: шляхом ручного введення або через інтеграцію з API зовнішніх ресурсів, після чого вони потрапляють у сутність Vacancy, що зберігає дані про посаду, вимоги до кандидата та умови праці.

Для забезпечення ефективної роботи інтелектуального вебзастосунку необхідно чітко визначити його функціональні вимоги. Вони описують конкретні дії, які система повинна виконувати для взаємодії з користувачами та досягнення поставленої мети – автоматизованого порівняння резюме з вакансіями і формування персоналізованих рекомендацій.

У вебзастосунку передбачено три основні типи користувачів, або акторів, кожен з яких виконує власні функції та має різний рівень доступу до системи. Головним актором є пошукач роботи, який взаємодіє з усіма ключовими можливостями: він може завантажувати або створювати резюме, додавати вакансії для порівняння, отримувати результати аналізу та рекомендації щодо покращення, а також зберігати історію перевірок і формувати звіти у зручному форматі. Другим актором виступає HR-фахівець або ментор, роль якого полягає у додаванні вакансій та перегляді відповідності кандидатів. У розширеній версії системи такий користувач може надавати поради пошукачам роботи, що робить платформу більш інтерактивною. Третім актором є адміністратор, який відповідає за модерацію користувачів і вакансій, контроль якості даних, технічне

адміністрування та підтримку стабільної роботи сервісу. Завдяки такому розподілу ролей система забезпечує повноцінну взаємодію між учасниками ринку праці, а також надійність і зручність використання для всіх категорій користувачів.

Для наочного представлення взаємодії користувачів із системою побудовано Use Case діаграму (рис. 2). Вона відображає основні сценарії використання вебзастосунку та показує, які дії доступні для різних категорій акторів

Розглянемо далі формулювання нефункціональних вимог до системи. Система повинна демонструвати високу продуктивність і стабільність роботи, забезпечуючи швидке оброблення запитів користувачів навіть при значному навантаженні. Зокрема, час виконання операції порівняння резюме з вакансією не має перевищувати 3–5 секунд, що гарантує комфортну взаємодію користувача з вебзастосунком у режимі реального часу. Такий показник досягається за рахунок оптимізації запитів до бази даних, попереднього кешування результатів обробки та використання асинхронних механізмів FastAPI.

У разі одночасної роботи понад 100 активних користувачів система повинна забезпечувати стабільну швидкодію без збоїв і затримок. Для цього передбачено застосування балансування навантаження на рівні сервера, а також використання пулу з’єднань із базою даних PostgreSQL, що дозволяє ефективно обробляти паралельні запити.

Система повинна забезпечувати високий рівень безпеки, надійності та захисту даних користувачів. Усі персональні дані, що зберігаються або передаються, мають бути зашифровані з використанням алгоритмів AES-256 для збереження у сховищі та TLS/SSL-сертифікатів для передачі між клієнтом і сервером.

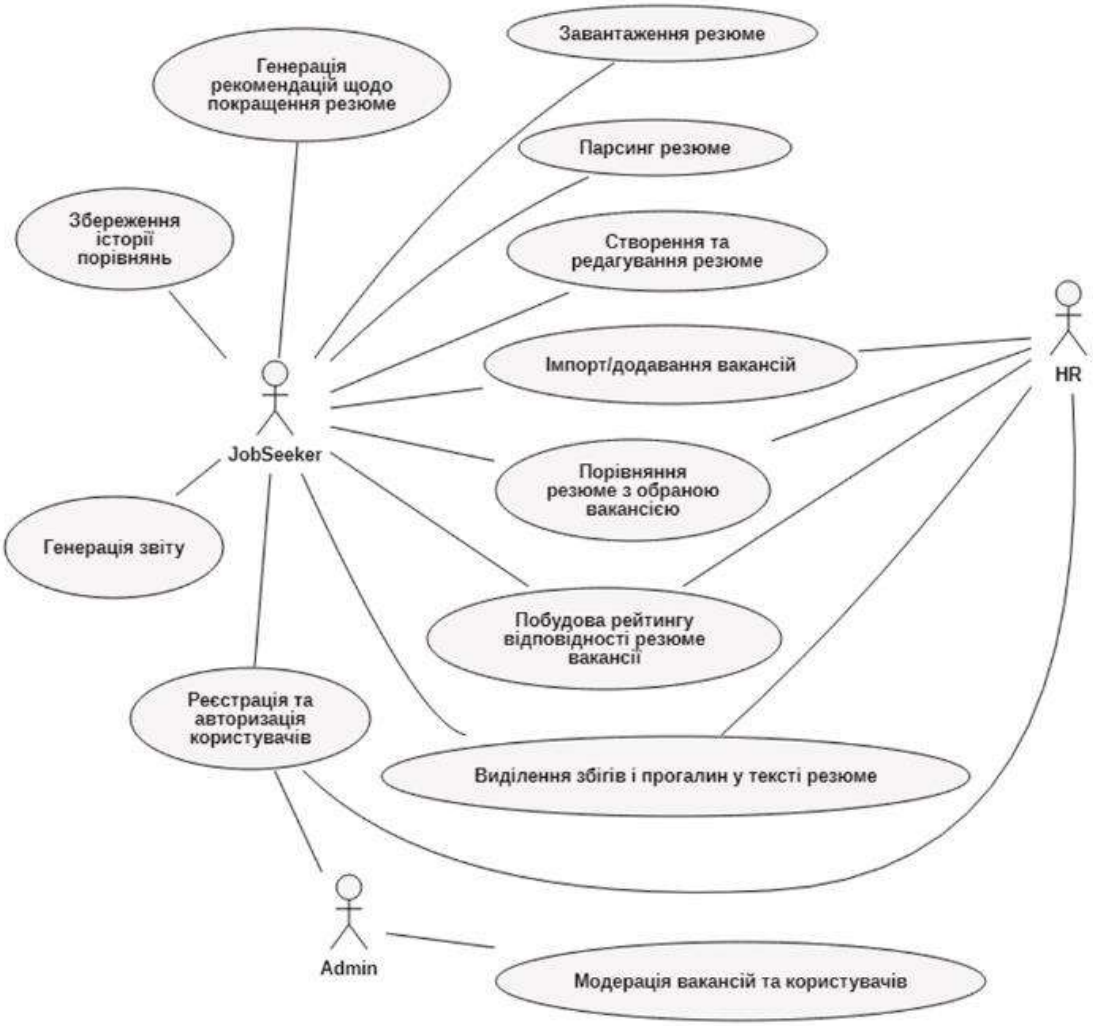


Рис. 2 – Діаграма варіантів використання

Архітектура інтелектуального вебзастосунку представлена діаграмою концептуальних класів (рис. 3). Центральним елементом є узагальнений клас User, що містить базові атрибути (ідентифікатор, ім’я, email, пароль) та методи для виконання автентифікації, зокрема вхід та вихід із системи. Від нього успадковуються три спеціалізовані підкласи – JobSeeker, HR та Admin, які реалізують специфічний функціонал відповідно до своєї ролі.

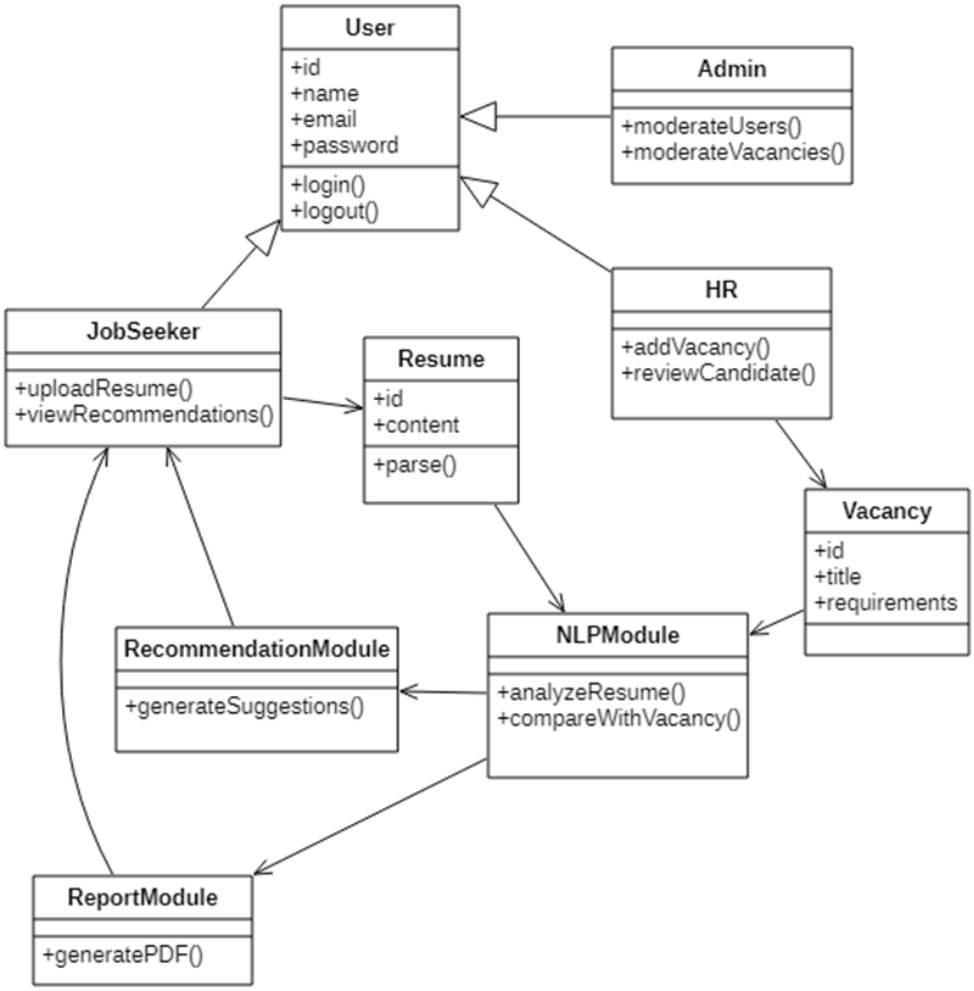


Рис.3 – Діаграма концептуальних класів веб-застосунку

Клас JobSeeker є користувачем, що шукає роботу. Він має можливість завантажувати резюме, переглядати результати його аналізу, отримувати рекомендації та формувати звіти для подальшого удосконалення власного профілю. Клас HR відповідає за управління вакансіями: додавання нових оголошень, перегляд інформації про кандидатів та оцінювання їх відповідності вимогам. Admin реалізує функції адміністратора системи, зокрема модерацію користувачів, контроль якості опублікованих вакансій і забезпечення безпеки роботи застосунку.

NLP-модуль виконує ключову роль у вебзастосунку, оскільки він відповідає за аналіз тексту резюме та вакансій і формування показника

їхньої відповідності. Його робота побудована на поєднанні кількох етапів: попередньої обробки тексту, векторизації, обчислення міри подібності та виділення збігів і прогалин.

На першому етапі виконується попередня обробка текстів. До неї належать токенізація (розбиття тексту на окремі слова), нормалізація, лематизація та видалення стоп-слів. Завдяки цьому система залишає лише суттєві слова, які впливають на зміст. Наприклад, фраза «маю досвід роботи з програмуванням на Python» після обробки перетворюється на набір ключових термінів: досвід, робота, програмування, Python.

Далі тексти резюме та вакансії перетворюються у числове представлення. Для цього використовується метод TF-IDF (Term Frequency – Inverse Document Frequency), який враховує не тільки частоту слів у документі, але й їхню унікальність у корпусі текстів. Формально значення TF-IDF для терміна t у документі d обчислюється як:

$$TF - IDF(t,d) = TF(t,d) * IDF(t),$$

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}},$$

$$IDF(t) = \log\frac{N}{n_t},$$

де $f_{t,d}$ – кількість входжень терміна t у документ d;

N – кількість документів у колекції;

n_t – кількість документів, що містять термін t.

Таким чином, TF-IDF дозволяє виділити найважливіші слова у вакансії та резюме.

Після векторизації обидва документи представлені у вигляді векторів. Для їх порівняння використовується косинусна міра подібності:

$$S(A,B) = \frac{\vec{A} * \vec{B}}{\|\vec{A}\| * \|\vec{B}\|},$$

де $\vec{A}$ – вектор резюме;

$\vec{B}$ – вектор вакансії;

$\vec{A} * \vec{B}$ – скалярний добуток векторів;

$\|\vec{A}\|$ та $\|\vec{B}\|$ – їх довжини відповідно.

Значення *Sim(A,B)* знаходиться у діапазоні від 0 до 1: чим ближче воно до 1, тим вищий рівень відповідності. Для зручності користувача результат додатково відображається у відсотках.

Окрім розрахунку загального показника подібності, NLP-модуль виконує аналіз вимог вакансії. Якщо певна навичка, позначена роботодавцем як обов'язкова, не зустрічається у резюме, система автоматично фіксує цю прогалину.

Отримані результати передаються у модуль рекомендацій, де на основі виявлених прогалин система формує поради для JobSeeker, а також у модуль звітів, який створює зведені документи для HR. Таким чином, використання формальних методів, зокрема TF-IDF та косинусної подібності, дозволяє перетворити простий текстовий опис у кількісні показники, що підвищують об'єктивність і точність аналізу.

Експериментальне тестування було проведено з метою перевірки, наскільки використання розробленого вебзастосунку сприяє підвищенню задоволеності пошукачів роботи процесом підготовки та використання резюме. Для цього було організовано експеримент за участю десяти користувачів, які оцінювали систему за двома критеріями: «Якість резюме» та «Зручність користування». Кожен учасник спочатку виконував завдання без використання вебзастосунку, а потім повторював їх із його допомогою.

У ситуації без системи користувачі складали резюме самостійно, орієнтуючись на загальні поради або стандартні шаблони. Це призводило до того, що документи були перевантажені другорядною інформацією, недостатньо сфокусовані на вимогах конкретних вакансій, а процес їх підготовки був тривалим та незручним. У випадку використання нашого вебзастосунку система автоматично аналізувала резюме, порівнювала його з вакансіями, виділяла ключові збіги та прогалини і пропонувала персоналізовані рекомендації для покращення. Завдяки цьому користувачі отримували структурований, змістовний і більш конкурентоспроможний документ, а сам процес взаємодії з інтерфейсом став простішим та зрозумілішим.

Результати експерименту подано на рис. 4. Як видно з даних, оцінки користувачів за обома критеріями суттєво зросли після впровадження вебзастосунку. У середньому якість резюме покращилася в 1,5 раза, а зручність користування – в 1,6 разів. Це підтверджує, що створена система дійсно підвищує ефективність та комфорт пошуку роботи.

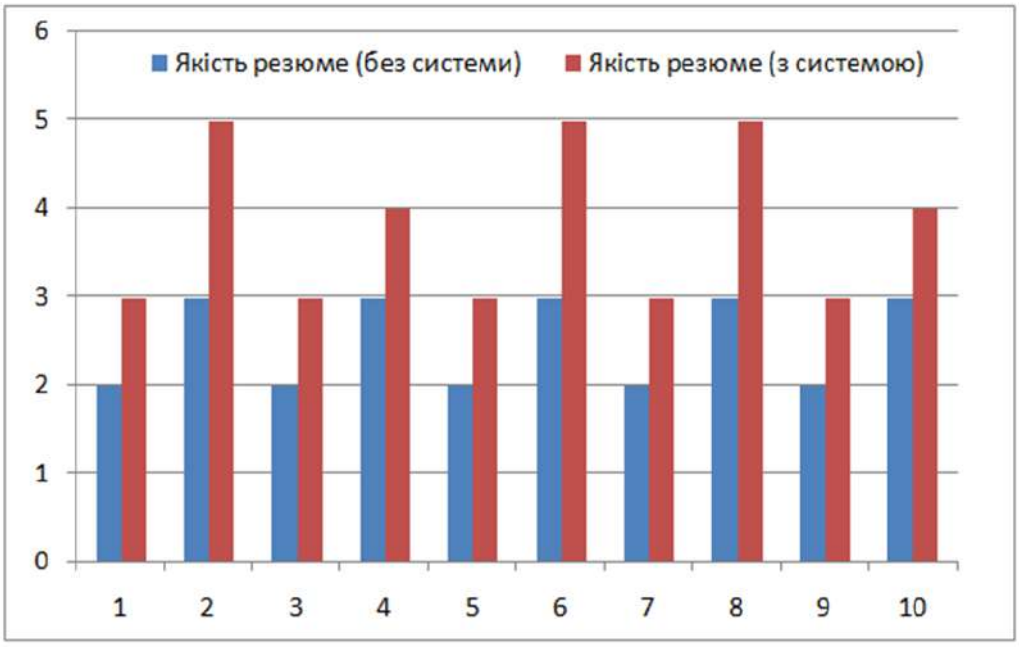


Рис. 4 – Результати експериментів

Висновки. Результатом дослідження є розробка інтелектуального вебзастосунку для порівняння резюме з вакансіями та формування персоналізованих рекомендацій щодо їх покращення. Запропонована система дозволяє автоматизувати процес аналізу резюме, оцінювати його відповідність вимогам конкретних вакансій, виділяти збіги та прогалини у навичках і надавати користувачам зрозумілі поради щодо вдосконалення. Це підвищує ефективність процесу підготовки резюме у 1.5 рази та сприяє зростанню задоволеності пошукачів роботою із сучасними цифровими інструментами. За результатами експериментального тестування було підтверджено, що використання вебзастосунку підвищує якість резюме та рівень задоволеності користувачів у півтора та більше разів відповідно в порівнянні з традиційним підходом.

Список використаних джерел

1. Jobscan. URL: https://www.jobscan.co (дата звернення: 01.06.2026).
2. Zety. URL: https://zety.com (дата звернення: 01.06.2026).
3. LinkedIn. URL: https://www.linkedin.com (дата звернення: 01.06.2026).
4. Work.ua. URL: https://www.work.ua (дата звернення: 01.06.2026).

References

5. Jobscan. URL: https://www.jobscan.co (дата звернення: 01.06.2026).
6. Zety. URL: https://zety.com (дата звернення: 01.06.2026).
7. LinkedIn. URL: https://www.linkedin.com (дата звернення: 01.06.2026).
8. Work.ua. URL: https://www.work.ua (дата звернення: 01.06.2026).