МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ РЕКРУТИНГОВОЇ ПЛАТФОРМИ ДЛЯ ІТ КОМПАНІЙ ІЗ ВИКОРИСТАННЯМ REACT»**

Виконав: студент 2 курсу, групи 8.1213-іпз-1
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)
освітньої програми інженерія програмного забезпечення
(назва освітньої програми)
А.В. Єрохін
(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії, к.ф.-м.н. Мильцев О.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент професор кафедри комп'ютерних наук, професор, д.т.н. Шило Г.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення
(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

_____ Лісняк А.О.
(підпис)

“ ____ ” _____ 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Єрохіну Артуру Вадимовичу
(прізвище, ім’я та по-батькові)

1. Тема роботи Розробка клієнтської частини рекрутингової платформи для IT компаній із використанням React

керівник роботи Мильцев Олександр Михайлович, к.ф.-м.н.
(прізвище, ім’я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 01 » березня 2024 року № 385-с

2. Строк подання студентом роботи 25.11.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Проєктування та реалізація рекрутингової платформи для IT компаній.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 07.03.2024 р.

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	18.03.2024	
2.	Збір вихідних даних.	22.04.2024	
3.	Обробка методичних та теоретичних	10.06.2024	
	джерел.		
4.	Розробка першого та другого розділу.	30.08.2024	
5.	Розробка третього розділу.	04.11.2024	
6.	Оформлення та нормоконтроль	20.11.2024	
	кваліфікаційної роботи магістра.		
7.	Захист кваліфікаційної роботи.	11.12.2024	

Студент _____ А.В. Єрохін
(підпис) (ініціали та прізвище)

Керівник роботи _____ О.М. Мильцев
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ А.В. Столярова
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка клієнтської частини рекрутингової платформи для IT компаній із використанням React»: 78 с., 35 рис., 13 джерел.

АДАПТИВНИЙ ДИЗАЙН, ПОШУК РОБОТИ, РЕКОМЕНДАЦІЇ, РЕКРУТИНГОВА ПЛАТФОРМА, MATERIALUI, REACT, SIGNALR, TANSTACK TABLE, TANSTACK QUERY, TYPESCRIPT, ZUSTAND.

Об’єкт дослідження – процеси автоматизації взаємодії між роботодавцями, рекрутерами та шукачами роботи у сфері IT.

Мета роботи – розробка клієнтської частини рекрутингової платформи, яка забезпечує зручність, швидкість і точність підбору вакансій для шукачів роботи та кандидатів для рекрутерів, а також інтеграцію з сучасними технологіями автоматизації та комунікації.

Методи дослідження: аналіз, проєктування, реалізація, тестування.

У даній роботі досліджується клієнтська частина вебплатформи, яка автоматизує процес пошуку вакансії кандидатами та пошук самих кандидатів рекрутерами або роботодавцями. Додаток відповідає сучасним вимогам до зручності, адаптивності та швидкості роботи. Використані технології та архітектурні рішення дозволяють легко масштабувати систему та інтегрувати додаткові функції в майбутньому. Розроблена платформа покликана скоротити час на пошук роботи або кандидатів, підвищити ефективність процесу працевлаштування та створити зручну екосистему для взаємодії між роботодавцями, рекрутерами та шукачами роботи.

Розроблена платформа базується на сучасних технологіях та інструментах, зокрема React, TypeScript, Zustand, React Router, Tanstack Query, MaterialUI та .NET.

SUMMARY

Master’s qualifying paper “Developing a Frontend Part of a Recruiting Platform for IT Companies Using React”: 78 pages, 35 figures, 13 references.

ADAPTIVE DESIGN, JOB SEARCH, MATERIALUI, REACT, RECOMMENDATIONS, RECRUITMENT PLATFORM, SIGNALR, TANSTACK TABLE, TANSTACK QUERY, TYPESCRIPT, ZUSTAND.

The object of the study is automation processes of interaction between employers, recruiters, and job seekers in the IT field.

The aim of the study is the development of the client side of a recruitment platform that ensures convenience, speed, and accuracy in selecting vacancies for job seekers and candidates for recruiters, as well as integration with modern automation and communication technologies.

Methods of research are analysis, design, implementation, testing.

This work explores the client side of a web platform that automates the process of job searching for candidates and finding candidates for recruiters or employers. The application meets modern requirements for convenience, adaptability, and speed of operation. The technologies and architectural solutions used allow for easy scalability of the system and integration of additional features in the future. The developed platform aims to reduce the time spent searching for jobs or candidates, increase the efficiency of employment processes, and create a convenient ecosystem for interaction between employers, recruiters, and job seekers.

The developed platform is based on modern technologies and tools, including React, TypeScript, Zustand, React Router, Tanstack Query, MaterialUI, and .NET.

ЗМІСТ

Завдання на кваліфікаційну роботу 2
Реферат 4
Summary 5
Вступ 9
1 Технічне завдання 11
1.1 Терміни та визначення 11
1.1.1 Загальні терміни 11
1.1.2 Технічні терміни 12
1.2 Функціональні вимоги 13
1.2.1 Клієнтська частина 13
1.2.2 Серверна частина 13
1.3 Нефункціональні вимоги 14
1.4 Опис предметної області 15
1.5 Опис системи 16
1.5.1 Клієнтська частина 16
1.5.2 Серверна частина 17
1.6 Огляд інструментів розробки 18
1.6.1 Інструменти для реалізації клієнтської частини 18
1.6.2 Інструменти для реалізації серверної частини 19
2 Проєктування 22
2.1 Проєктування за допомогою UML 22
2.1.1 Проєктування за допомогою діаграми прецедентів 23
2.1.2 Проєктування за допомогою діаграми компонентів 27
2.1.3 Проєктування за допомогою діаграми діяльності 31
2.1.4 Проєктування за допомогою діаграми станів 33
2.2 Архітектура 36
3 Реалізація та тестування 38

3.1 Опис інструментів для розробки клієнтської частини 38
3.1.1 Мова програмування TypeScript 38
3.1.2 Бібліотека React 38
3.1.3 Бібліотека для верстки інтерфейсу MaterialUI 39
3.1.4 Інструмент для збірки ViteJS 40
3.1.5 Tanstack Query 40
3.1.6 Zustand 41
3.1.7 React Router 42
3.1.8 SignalR 43
3.2 Процес створення користувацького інтерфейсу 44
3.2.1 Конфігурація теми MaterialUI в React 44
3.2.2 Створення React компонентів 45
3.2.3 Обробка даних в реальному часі 45
3.3 Тестування системи 46
3.3.1 Запуск системи у веббраузері 46
3.3.2 Тестування адаптації верстки 48
3.4 Керівництво користувача 50
3.4.1 Рівень підготовки користувача 50
3.4.2 Підготовка до роботи 51
3.4.3 Авторизація в систему 51
3.4.4 Реєстрація нового акаунту 52
3.4.5 Особистий кабінет кандидата 54
3.4.6 Особистий кабінет рекрутера 56
3.4.7 Пошук вакансій 58
3.4.8 Перегляд вакансії 60
3.4.9 Створення вакансії 63
3.4.10 Перегляд кандидатів 65
3.4.11 Перегляд анкети кандидата 67
3.4.12 Перегляд роботодавців 68
3.4.13 Листування між кандидатом та рекрутером 70

3.4.14 Тестування кандидатів 73
Висновки 76
Перелік посилань..... 77

ВСТУП

У сучасному світі динамічний розвиток інформаційних технологій і глобалізація економіки сприяють трансформації ринку праці, що вимагає нових підходів до процесів працевлаштування та підбору персоналу. Інтеграція цифрових рішень у ці процеси дозволяє забезпечити зручність і швидкість взаємодії між роботодавцями, рекрутерами та кандидатами, підвищуючи ефективність прийняття рішень та зменшуючи витрати часу і ресурсів.

Актуальність дослідження зумовлена зростаючою потребою IT-компаній у швидкому і точному підборі кваліфікованих фахівців, а також прагненням шукачів роботи отримати зручний доступ до релевантних вакансій. Традиційні підходи до рекрутингу, такі як оголошення у ЗМІ або персональні контакти, часто не відповідають сучасним вимогам ринку. Це створює необхідність у впровадженні цифрових платформ, що дозволяють автоматизувати ключові процеси рекрутингу та забезпечити персоналізований підхід до кожного користувача.

Мета роботи полягає у розробці клієнтської частини рекрутингової платформи, яка дозволить автоматизувати процес пошуку та підбору кандидатів, створити зручний інтерфейс для взаємодії користувачів і забезпечити ефективність комунікації між роботодавцями, рекрутерами та шукачами роботи.

На основі цього ми можемо визначити такі завдання для дослідження:

– аналіз предметної області та визначення вимог до функціоналу рекрутингової платформи;
– проєктування архітектури клієнтської частини з використанням сучасних підходів і технологій;
– розробка адаптивного інтерфейсу користувача для забезпечення зручності роботи на різних пристроях;

- реалізація функцій персоналізації за допомогою алгоритмів машинного навчання;
- інтеграція системи повідомлень для покращення взаємодії між користувачами платформи;
- тестування та оцінка розробленого додатку на відповідність заданим вимогам.

Методи дослідження: аналіз, проєктування, реалізація, тестування.

Перший розділ присвячено збору та аналізу вимог до системи, огляду інструментів розробки та опису системи. У другому розділі розглянуто етапи проєктування системи, наведено детальний діаграм. Третій розділ присвячено реалізації та тестуванню роботи системи, наведено детальне керівництво користувача, яке описує процес роботи у системі.

1 ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Терміни та визначення

У цьому розділі наведено основні терміни, які використовуються в роботі, розділені на дві категорії: загальні терміни, що стосуються предметної області, та технічні терміни, що застосовуються до розробки клієнтської частини.

1.1.1 Загальні терміни

Для забезпечення чіткого розуміння функціональності та термінології, що використовується в контексті рекрутингової платформи, нижче наведено визначення основних понять, які максимально пояснюють особливості роботи системи та її основних учасників.

Рекрутингова платформа – інформаційна система, що дозволяє автоматизувати процеси пошуку роботи та підбору кандидатів, забезпечуючи ефективну взаємодію між роботодавцями, рекрутерами та шукачами роботи.

Шукач роботи – користувач, який створює профіль, розміщує резюме та шукає відповідні вакансії на платформі.

Рекрутер – користувач, який створює вакансії, підбирає кандидатів та аналізує їхні профілі для відповідності вимогам роботодавців.

Рекомендації – механізм, що використовує алгоритми для автоматичного підбору вакансій для шукачів роботи або кандидатів для рекрутерів на основі заданих критеріїв.

Фільтри пошуку – інструменти для звуження списку результатів на основі параметрів, таких як місцезнаходження, рівень досвіду, тип роботи тощо.

1.1.2 Технічні терміни

React – бібліотека JavaScript для створення інтерфейсів користувача на основі компонентного підходу.

TypeScript – строго типізована надбудова над JavaScript, яка додає підтримку статичної типізації для підвищення надійності коду.

MaterialUI – набір компонентів інтерфейсу користувача для React, що відповідають принципам Material Design.

Zustand – бібліотека для керування станом додатка у React із мінімальним використанням шаблонного коду.

SignalR – бібліотека для реалізації функціональності вебсокетів і організації реального часу між клієнтами та сервером.

Tanstack Query – інструмент для управління станом серверних запитів у React-додатках із підтримкою кешування, синхронізації та повторних спроб.

Адаптивний дизайн – підхід до розробки інтерфейсів, що забезпечує коректне відображення додатку на пристроях із різними розмірами екранів.

REST API – архітектурний стиль для створення вебсервісів, який дозволяє клієнтським додаткам взаємодіяти із сервером через HTTP-запити.

Компонент – автономний елемент інтерфейсу користувача в React, що інкапсулює логіку, шаблон і стилі.

ES6 – це суттєве оновлення стандарту ECMAScript, яке стало поворотним моментом у розвитку мови, збагачуючи її семантику, синтаксис та можливості.

ESNext – це загальний термін для опису наступних специфікацій ECMAScript, який фокусується на поступовому вдосконаленні мови через регулярні щорічні оновлення, додаючи нові можливості для підвищення ефективності та гнучкості програмування.

Рендеринг – це процес відображення або візуалізації даних у вигляді, доступному для користувача.

1.2 Функціональні вимоги

1.2.1 Клієнтська частина

Клієнтська частина платформи розробляється для забезпечення зручного інтерфейсу взаємодії користувачів із системою. Клієнтська частина складається з таких основних функцій:

- створення профілю користувача: забезпечення можливості реєстрації та авторизації шукачів роботи та рекрутерів, заповнення інформації про досвід роботи, освіту, навички та побажання щодо роботи;
- пошук вакансій і кандидатів: надання інструментів для пошуку за допомогою фільтрів (локація, зарплата, рівень досвіду тощо) та відображення релевантних результатів;
- рекомендаційна система: автоматичне формування списку вакансій або кандидатів на основі профілю користувача та його попередніх дій на платформі;
- комунікація: впровадження вбудованого чату для зручної взаємодії між шукачами роботи та рекрутерами;
- управління заявками: можливість перегляду статусу заявок, поданих на вакансії, або заявок від кандидатів на запропоновані вакансії;
- адаптивність: забезпечення коректного відображення інтерфейсу на пристроях із різними розмірами екранів.

1.2.2 Серверна частина

Серверна частина відповідає за обробку даних та взаємодію з клієнтською частиною через REST API. Серверна частина складається з таких основних функцій:

- робота з базою даних: зберігання даних користувачів, вакансій, заявок і комунікацій;
- обробка запитів: реалізація API для передачі даних між клієнтською частиною та сервером;
- безпека: забезпечення авторизації та аутентифікації користувачів, захист даних під час передачі;
- рекомендаційні алгоритми: обробка та аналіз даних для формування персоналізованих рекомендацій;

Серверна частина розроблена з використанням технологій .NET, ASP.NET Core та Microsoft SQL для забезпечення стабільності та продуктивності системи.

1.3 Нефункціональні вимоги

Нефункціональні вимоги визначають технічні характеристики та обмеження, яким має відповідати система для забезпечення її надійності, продуктивності та зручності використання. Клієнтська частина рекрутингової платформи має відповідати таким основним нефункціональним вимогам:

- продуктивність: система повинна забезпечувати швидкий відгук на дії користувача, зокрема час відгуку інтерфейсу не повинен перевищувати 2 секунд за умови стабільного з’єднання;
- масштабованість: система повинна підтримувати зростання кількості одночасних користувачів без суттєвого зниження продуктивності;
- доступність: вебдодаток має коректно працювати в популярних веббраузерах (Google Chrome, Mozilla Firefox, Microsoft Edge) та на різних платформах (Windows, macOS, Linux);
- адаптивність: інтерфейс користувача має автоматично підлаштовуватися під розміри екранів настільних і мобільних пристроїв;

- безпека: система повинна забезпечувати захист персональних даних користувачів відповідно до стандартів GDPR;
- надійність: забезпечення безперебійної роботи системи з мінімальним часом простою (не більше 0,1% від загального часу роботи);
- інтеграція: система повинна забезпечувати можливість підключення сторонніх сервісів, таких як соціальні мережі, через API;
- логування: необхідно реалізувати систему збору логів для відстеження помилок і роботи ключових компонентів.

1.4 Опис предметної області

Рекрутингова платформа для IT-компаній є сучасним цифровим рішенням, яке спрямоване на оптимізацію процесів пошуку роботи та підбору персоналу. Головною метою системи є об’єднання роботодавців, рекрутерів та шукачів роботи на одній платформі для забезпечення ефективної взаємодії між сторонами та швидкого закриття вакансій.

Основні характеристики предметної області:

- учасники: платформа обслуговує три основні категорії користувачів: шукачі роботи, рекрутери та роботодавці – кожен із них має доступ до спеціалізованих функцій платформи відповідно до своїх потреб;
- функціональність: користувачі можуть створювати профілі, завантажувати резюме, переглядати вакансії, фільтрувати їх за параметрами, а також подавати заявки – рекрутери мають змогу створювати вакансії, переглядати резюме кандидатів і вести комунікацію;
- ринок IT: у фокусі платформи є вакансії, пов’язані з інформаційними технологіями, включаючи програмістів, дизайнерів, аналітиків, менеджерів проєктів та інших фахівців галузі;

– особливості взаємодії: система передбачає рекомендації на основі профілів користувачів та їх активності на платформі – для зручності також реалізовано систему миттєвих повідомлень і інтеграцію із соціальними мережами.

Платформа вирішує такі проблеми:

– скорочення часу пошуку: завдяки розширеним фільтрам та рекомендаціям користувачі можуть швидко знаходити відповідні вакансії чи кандидатів;
– централізація інформації: всі учасники процесу отримують доступ до актуальної інформації на одній платформі;
– покращення комунікації: інтеграція чату дозволяє рекрутерам і шукачам роботи оперативно обмінюватися інформацією;
– таким чином, розробка цієї платформи враховує сучасні вимоги до цифрових інструментів рекрутингу, пропонуючи ефективні рішення для автоматизації ключових процесів у сфері найму.

1.5 Опис системи

1.5.1 Клієнтська частина

Клієнтська частина системи є основним інструментом взаємодії користувачів із платформою. Вона реалізована у вигляді вебдодатку, розробленого за допомогою React і TypeScript, що забезпечує адаптивність і продуктивність.

Клієнтська частина складається з таких основних компонентів:

– головна сторінка: містить загальну інформацію про платформу, навігаційні елементи та рекомендації для нових користувачів;
– сторінка реєстрації та авторизації: дозволяє створювати облікові записи та входити в систему з використанням логіна, пароля або соціальних мереж;

– кабінет користувача: надає доступ до персональних даних, історії заявок, рекомендацій, можливостей редагування профілю та перегляду статусів заявок;
– розширений пошук: реалізує фільтри за категоріями, місцезнаходженням, рівнем досвіду, зарплатою тощо;
– сторінка вакансій: відображає детальну інформацію про кожну вакансію, включаючи опис, вимоги та умови праці;
– система повідомлень: забезпечує комунікацію між рекрутерами та шукачами роботи в реальному часі через SignalR.

Особлива увага приділяється адаптивності інтерфейсу, що дозволяє коректно відображати систему на пристроях із різними розмірами екранів, а також продуктивності за допомогою використання бібліотек Tanstack Query і Zustand.

1.5.2 Серверна частина

Серверна частина забезпечує обробку даних та взаємодію між клієнтською частиною і базою даних. Реалізована за допомогою ASP.NET Core і Microsoft SQL.

Серверна частина складається з таких основних функцій:

– реалізація API: надає клієнтській частині доступ до даних через REST-запити;
– обробка даних: виконує запити до бази даних, обробляє інформацію про вакансії, профілі користувачів, заявки та повідомлення;
– захист даних: забезпечує безпечну авторизацію та передачу даних між компонентами;
– рекомендаційна система: обчислює персоналізовані пропозиції на основі алгоритмів аналізу даних.

Серверна частина створена з урахуванням вимог до надійності, безпеки та масштабованості системи.

1.6 Огляд інструментів розробки

1.6.1 Інструменти для реалізації клієнтської частини

Для розробки клієнтської частини платформи, та забезпечення адаптивності і продуктивності, використовуються такі сучасні технології та бібліотеки:

- React – JavaScript-бібліотека для створення інтерфейсів користувача на основі компонентного підходу (забезпечує швидкість розробки та масштабованість);
- TypeScript – надбудова над JavaScript, яка додає статичну типізацію, підвищуючи надійність коду та полегшуючи його підтримку;
- MaterialUI – набір компонентів інтерфейсу користувача, що відповідають принципам Material Design (використовується для створення сучасного та адаптивного дизайну);
- Zustand – легка бібліотека для управління станом програми. Забезпечує гнучкість і простоту інтеграції;
- SignalR – бібліотека для реалізації двостороннього зв’язку між клієнтом і сервером у реальному часі, зокрема для функціональності чату;
- Tanstack Query – інструмент для управління серверними запитами, що підтримує кешування, повторні спроби та синхронізацію даних;
- Tanstack Table – бібліотека для створення потужних і настроюваних таблиць з функціями сортування, фільтрації та пагінації;
- Tanstack Virtual – використовується для оптимізації роботи з великими списками або таблицями за рахунок віртуалізації рендерингу;
- React Hook Form – бібліотека для управління формами, яка забезпечує мінімальне навантаження на продуктивність і зручний API для розробників;

– Zod – бібліотека для валідації даних, що дозволяє перевіряти коректність вхідних даних у формах.

1.6.2 Інструменти для реалізації серверної частини

Для розробки серверної частини використовуються такі технології:

– ASP.NET Core: кросплатформенний фреймворк для створення вебдодатків і API – забезпечує високу продуктивність та інтеграцію з іншими інструментами Microsoft;
– Microsoft SQL Server: реляційна база даних для зберігання даних про користувачів, вакансії, заявки та повідомлення – підтримує масштабованість і високу швидкодію;
– Entity Framework Core: ORM (Object-Relational Mapping) для спрощення взаємодії з базою даних і написання SQL-запитів;
– SignalR: забезпечує функціональність реального часу для комунікації між сервером і клієнтами;
– Identity Server: використовується для реалізації механізмів аутентифікації та авторизації користувачів, забезпечуючи безпеку персональних даних.

1.7 Аналіз конкурентів

Для ефективного аналізу ринку та визначення переваг нашої платформи для пошуку роботи в IT-сфері, було проведено порівняння нашої системи з популярними конкурентами WorkUA та Djinni. Основними критеріями аналізу є функціональність, орієнтація на користувача, унікальні можливості та інтерфейс.

Особливості платформи WorkUA [1]:

– загальноукраїнська платформа для пошуку роботи з широким охопленням галузей та професій;
– користувачі можуть завантажувати свої резюме, шукати вакансії за ключовими словами, категоріями та місцем розташування;
– роботодавці розміщують вакансії та мають доступ до бази кандидатів;
– платформа фокусується на текстовій інформації з мінімальним акцентом на інтерактивність.

Недоліки платформи WorkUA:

– відсутність функціоналу для інтеграції в реальному часі (немає чатів для спілкування з рекрутерами);
– обмежені можливості для персоналізації пошуку вакансій, відсутні детальні фільтри для IT-спеціалістів;
– немає тестування кандидатів безпосередньо на платформі;
– загальний підхід до всіх галузей, а не вузька спеціалізація на IT.

Особливості платформи Djinni [2]:

– платформа спеціалізується на IT-ринку та має анонімний формат пошуку роботи для кандидатів;
– роботодавці отримують доступ до кандидатів, які погоджуються на співпрацю, після запиту;
– система пропонує зручний механізм для фільтрації вакансій за рівнем навичок, технологіями та досвідом роботи.

Недоліки платформи Djinni:

– відсутність автоматизованого тестування кандидатів на професійні навички;
– немає функції реального часу для миттєвого зв’язку між кандидатом та роботодавцем;
– обмежена можливість для рекрутерів створювати розширені профілі компаній з інформацією про команду чи проєкти;
– зосередженість лише на IT-галузі обмежує гнучкість платформи для роботодавців із суміжних галузей.

Наша платформа для пошуку вакансій пропонує такі можливості:

– платформа дозволяє здійснювати пошук вакансій, кандидатів та компаній із детальними фільтрами за ключовими словами, навичками, рівнем досвіду та зарплатою;
– у системі реалізовано функціонал надсилання тестових завдань кандидатам та отримання результатів безпосередньо на платформі;
– платформа підтримує обробку даних в реальному часі для спілкування між кандидатами та рекрутерами через автоматично створені чати при відгуку на вакансію;
– кандидати можуть створювати детальні профілі з інформацією про навички, досвід, мови програмування та рівень їх володіння;
– адаптивний дизайн дозволяє коректно відображати контент на різних пристроях, забезпечуючи позитивний користувацький досвід.

Наша платформа перевершує конкурентів завдяки інтеграції обробки даних в реальному часі, розширеній системі тестування кандидатів та зручному механізму пошуку та кастомізованим профілям. На відміну від WorkUA, наша платформа орієнтована виключно на IT-сферу, що дозволяє точніше відповідати вимогам фахівців та роботодавців у цій галузі. У порівнянні з Djinni, ми забезпечуємо більш інтерактивний інтерфейс із функцією миттєвої комунікації та розширеними можливостями для рекрутерів та кандидатів. Таким чином, наша система є оптимальним інструментом для пошуку роботи в IT-секторі, задовольняючи потреби сучасного ринку праці.

2 ПРОЄКТУВАННЯ

2.1 Проєктування за допомогою UML

У процесі розробки рекрутингової платформи для IT-компаній було використано мову UML (Unified Modeling Language) для візуалізації та моделювання структурних і поведінкових аспектів системи. UML дозволяє створювати наочні діаграми, які допомагають розробникам, аналітикам та замовникам краще зрозуміти логіку роботи системи та взаємозв’язок її компонентів [10].

Основними типами діаграм UML, використаними для проєктування системи, були:

– діаграма прецедентів: допомагає визначити основних користувачів системи (акторів) та сценарії їх взаємодії з платформою (на основі цієї діаграми визначаються основні функціональні вимоги до системи);
– діаграма компонентів: візуалізує основні модулі системи, їхні взаємозв’язки та інтерфейси, що забезпечують обмін даними між компонентами;
– діаграма діяльності: використовується для моделювання послідовності дій у межах різних процесів, таких як реєстрація користувача, пошук вакансій або подання заявки;
– діаграма станів: дозволяє описати зміни стану об’єктів системи у відповідь на події, що відбуваються у платформі (наприклад, зміна статусу заявки з «Подана» на «Переглянута»).

Використання UML у проєктуванні сприяло детальному опрацюванню архітектури системи, полегшило співпрацю між членами команди та забезпечило прозорість процесу розробки.

2.1.1 Проєктування за допомогою діаграми прецедентів

Діаграма прецедентів є одним із ключових інструментів у проєктуванні систем. Діаграма прецедентів використовується для моделювання взаємодії користувачів (акторів) із системою. Її призначення полягає у:

- визначенні ролей користувачів (акторів): вона допомагає ідентифікувати основних учасників системи, таких як кінцеві користувачі, адміністратори чи зовнішні сервіси;
- описі сценаріїв використання: діаграма демонструє, які функції системи доступні різним акторам і як ці функції взаємопов’язані. наприклад, шукач роботи може подавати заявки на вакансії, тоді як рекрутер керує цими заявками;
- формуванні функціональних вимог: діаграма є інструментом для збору й верифікації вимог, які мають бути реалізовані в системі;
- полегшенні спілкування: завдяки простоті й наочності діаграма сприяє кращому розумінню системи між розробниками, аналітиками та замовниками;
- закладенні основи для тестування: на її основі формуються тестові сценарії, які перевіряють правильність реалізації функцій системи.

На рисунку 2.1 зображено діаграму прецедентів проєкту. Діаграма містить два основних акторів: Кандидат і Рекрутер.

Кандидат – це людина, яка використовує платформу для створення персонального профілю, пошуку вакансій, подання заявок, проходження тестів та комунікації з рекрутерами. Його основна мета – знайти роботу, яка відповідає його кваліфікації, інтересам та вимогам.

Рекрутер є представником компанії або окремим фахівцем, який відповідає за пошук кандидатів, створення вакансій, обробку заявок, організацію тестування та комунікацію з потенційними працівниками. Рекрутер забезпечує працедавця якісними кандидатами, використовуючи можливості платформи для автоматизації процесів рекрутингу.

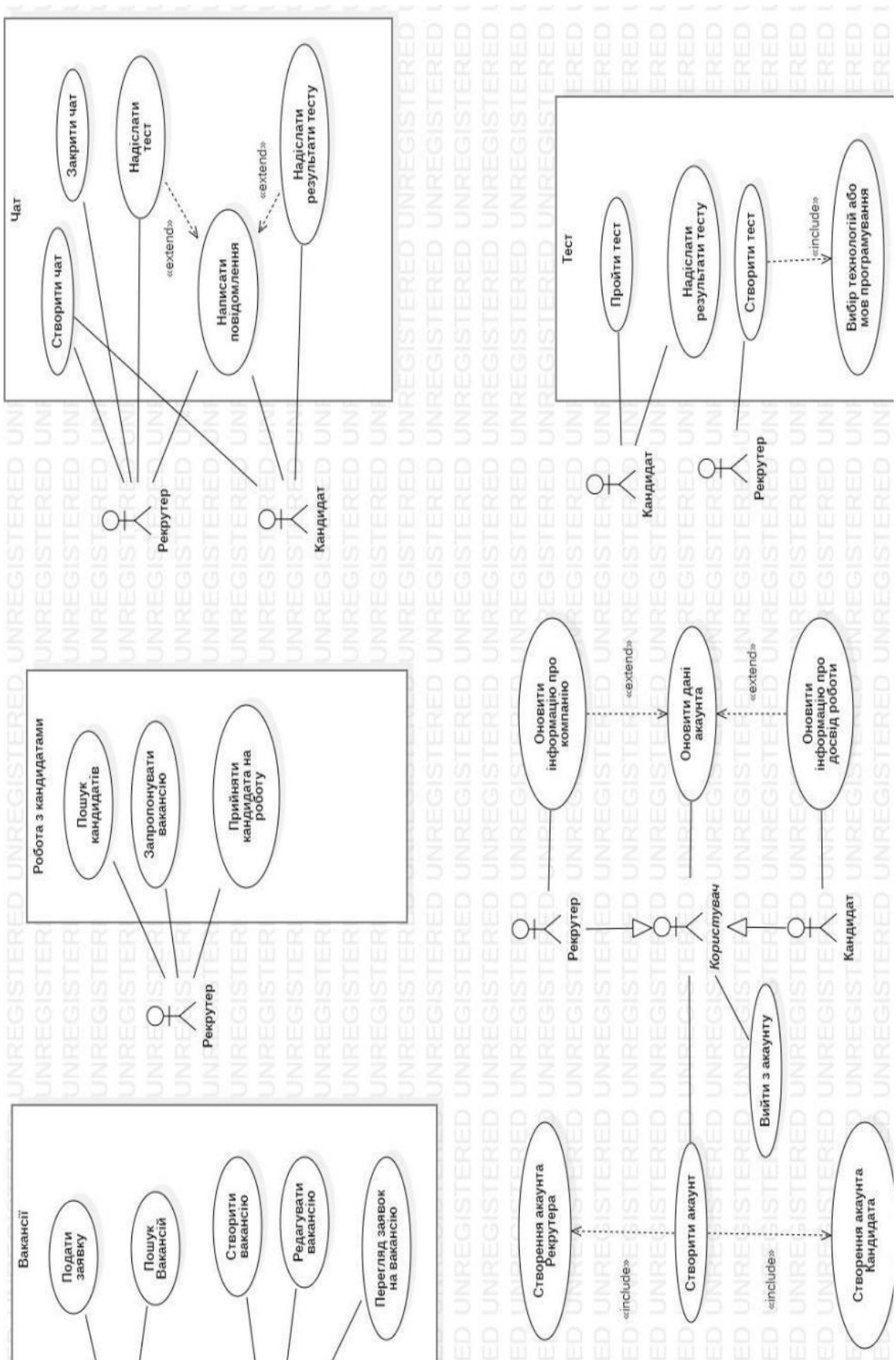


Рисунок 2.1 – Діаграма прецедентів

Розглянемо прецеденти, які зображені на діаграмі 2.1.

Прецедент «Створити акаунт». Користувач може зареєструватися на платформі, заповнивши базову інформацію, таку як ім’я, адреса електронної пошти та пароль. У процесі реєстрації він обирає свою роль – Кандидат чи Рекрутер. Цей процес є основою для подальшого доступу до функціоналу платформи.

Прецедент «Вхід в акаунт». Забезпечує можливість авторизації користувачів у системі. Після введення електронної пошти та пароля система надає доступ до профілю та відповідних функцій платформи. Якщо облікові дані введені неправильно, користувач отримує сповіщення про помилку.

Прецедент «Оновити інформацію про досвід роботи». Кандидат може редагувати свій профіль, додаючи деталі про професійний досвід, навички, проєкти, сертифікати та іншу релевантну інформацію. Це підвищує шанси на отримання пропозицій від рекрутерів.

Прецедент «Оновити інформацію про компанію». Рекрутер редагує профіль компанії, додаючи опис її діяльності, цінностей, відкритих вакансій і умов роботи. Це допомагає кандидатам краще зрозуміти, чи відповідає компанія їхнім очікуванням.

Прецедент «Пошук вакансій». Кандидат може використовувати зручні фільтри для пошуку вакансій за ключовими словами, локацією, рівнем зарплати, типом зайнятості або іншими параметрами. Ця функція дозволяє швидко знайти найбільш релевантні варіанти.

Прецедент «Пошук кандидатів». Рекрутер може застосовувати фільтри для знаходження профілів кандидатів, які відповідають конкретним вимогам вакансії. Наприклад, рекрутер може шукати кандидатів із певним досвідом, навичками або знанням мов програмування.

Прецедент «Подати заявку». Кандидат може подати заявку на вакансію, додаючи резюме, супровідний лист або інші документи. Система автоматично повідомляє рекрутера про нову заявку.

Прецедент «Перегляд заявок на вакансію». Рекрутер отримує доступ до списку заявок, де може переглядати деталі профілів кандидатів, резюме та

супровідні документи. Це допомагає приймати обґрунтовані рішення щодо подальшої комунікації з кандидатами.

Прецедент «Пройти тест». Кандидат має можливість виконувати тестові завдання, які створює рекрутер. Цей процес дозволяє об’єктивно оцінити кваліфікацію кандидата.

Прецедент «Створити тест». Рекрутер формує тестові завдання для кандидатів, вибираючи відповідні питання та критерії оцінювання. Це дозволяє стандартизувати процес відбору та перевірки знань.

Прецедент «Вибір технологій або мов програмування». Рекрутер задає специфічні вимоги до тестів, включаючи знання певних мов програмування чи технологій. Це допомагає оцінити, наскільки кандидат відповідає технічним вимогам вакансії.

Прецедент «Надсилати повідомлення». Інтегрований чат дозволяє кандидатам і рекрутерам спілкуватися в реальному часі, уточнювати деталі вакансій або процесу співбесіди. Цей функціонал робить комунікацію більш зручною та оперативною.

Прецедент «Створити вакансію». Рекрутер може додати нову вакансію, заповнивши інформацію про посаду, умови роботи, вимоги до кандидатів і зарплату. Ця вакансія одразу стає доступною для перегляду кандидатами.

Прецедент «Оновити інформацію про вакансію». Рекрутер може змінювати дані про вакансію, додаючи нові вимоги, змінюючи умови або оновлюючи інформацію про проєкт.

Прецедент «Закрити чат». Завершує спілкування між користувачами після досягнення домовленостей. Історія повідомлень зберігається для подальшого доступу.

Прецедент «Надіслати результати тесту». Кандидат передає виконані завдання рекрутеру через платформу для подальшого аналізу.

Прецедент «Надіслати тест». Рекрутер відправляє тест кандидатам, які подали заявку, для перевірки їхньої кваліфікації.

Прецедент «Перегляд тестів». Кандидат може переглянути доступні для нього тести та почати виконання обраного завдання.

Прецедент «Створити чат». Ініціює нову бесіду між кандидатом і рекрутером після подання заявки або відгуку на вакансію.

Прецедент «Перегляд заявок». Кандидат може перевіряти статус поданих заявок, отримувати сповіщення про зміни або відповіді рекрутерів.

Прецедент «Вийти з акаунту». Користувач може завершити сесію, що забезпечує конфіденційність даних і захист від несанкціонованого доступу.

Прецедент «Оновити дані акаунта». Актори «Кандидат» та «Рекрутер» можуть змінювати базову інформацію профілю, наприклад, електронну пошту, номер телефону чи пароль.

Прецедент «Перегляд вакансій». Кандидат переглядає список доступних вакансій, фільтрує їх і переходить до детального опису обраних варіантів.

Прецедент «Запросити кандидата на роботу». Актор «Рекрутер» надсилає кандидату пропозицію про співпрацю, якщо його кваліфікація відповідає вимогам вакансії.

Описані прецеденти ілюструють усі ключові функції платформи, забезпечуючи інтерактивну та ефективну взаємодію між кандидатами та рекрутерами, а також повний цикл автоматизації рекрутингових процесів.

2.1.2 Проєктування за допомогою діаграми компонентів

Проєктування за допомогою діаграми компонентів використовується для візуалізації архітектури системи, а саме ключових модулів та їхніх взаємозв’язків. Цей тип діаграм дозволяє зрозуміти, як компоненти взаємодіють між собою для забезпечення функціональності платформи. Завдяки цьому розробники отримують чітке уявлення про розподіл завдань між модулями, що полегшує процеси розробки, тестування та подальшої підтримки системи. Також діаграма компонентів сприяє ідентифікації точок

інтеграції, забезпечуючи масштабованість та адаптивність архітектури до змінних потреб бізнесу [13].

На рисунку 2.2 зображено діаграму компонентів, в якій представлений один із ключових модулів платформи – механізм пошуку, який забезпечує базову інфраструктуру для виконання запитів. Цей механізм може розширюватися додатковими можливостями, такими як фільтрація за специфічними параметрами, сортування результатів і персоналізація пошукових запитів, що підвищує його гнучкість і ефективність.

Розглянемо компоненти, які зображені на діаграмі 2.2.

Компонент «SearchEngine». Базовий компонент для виконання пошукових запитів. Забезпечує стандартизовану інфраструктуру для взаємодії з іншими модулями, такими як «SearchVacancies», «SearchEmployers» і «SearchEmployee». Підключається до механізмів фільтрації, сортування та пагінації для покращення користувацького досвіду. Інтегрується з іншими компонентами через уніфіковані інтерфейси, що сприяє швидкій обробці запитів.

Компонент «SearchVacancies». Розширює функціонал «SearchEngine», додаючи можливості для точного пошуку вакансій. Включає фільтри за ключовими параметрами, такими як зарплата, рівень досвіду, місце розташування та тип зайнятості. Інтегрується з компонентом «Vacancies», дозволяючи здійснювати ефективний пошук та аналіз даних.

Компонент «SearchEmployers». Успадковує базову функціональність «SearchEngine» і додає специфічні можливості для пошуку роботодавців. Цей модуль дозволяє кандидатам знаходити компанії за галуззю, розміром, розташуванням та іншими критеріями. Інтегрується з компонентом «Employers» для отримання повної інформації про роботодавців.

Компонент «Authentication». Відповідає за реєстрацію, авторизацію та забезпечення безпеки доступу до платформи. Реалізує механізми шифрування даних користувачів та управління їхніми сесіями. Інтегрується з усіма іншими компонентами, забезпечуючи контроль доступу до кожної функції системи.

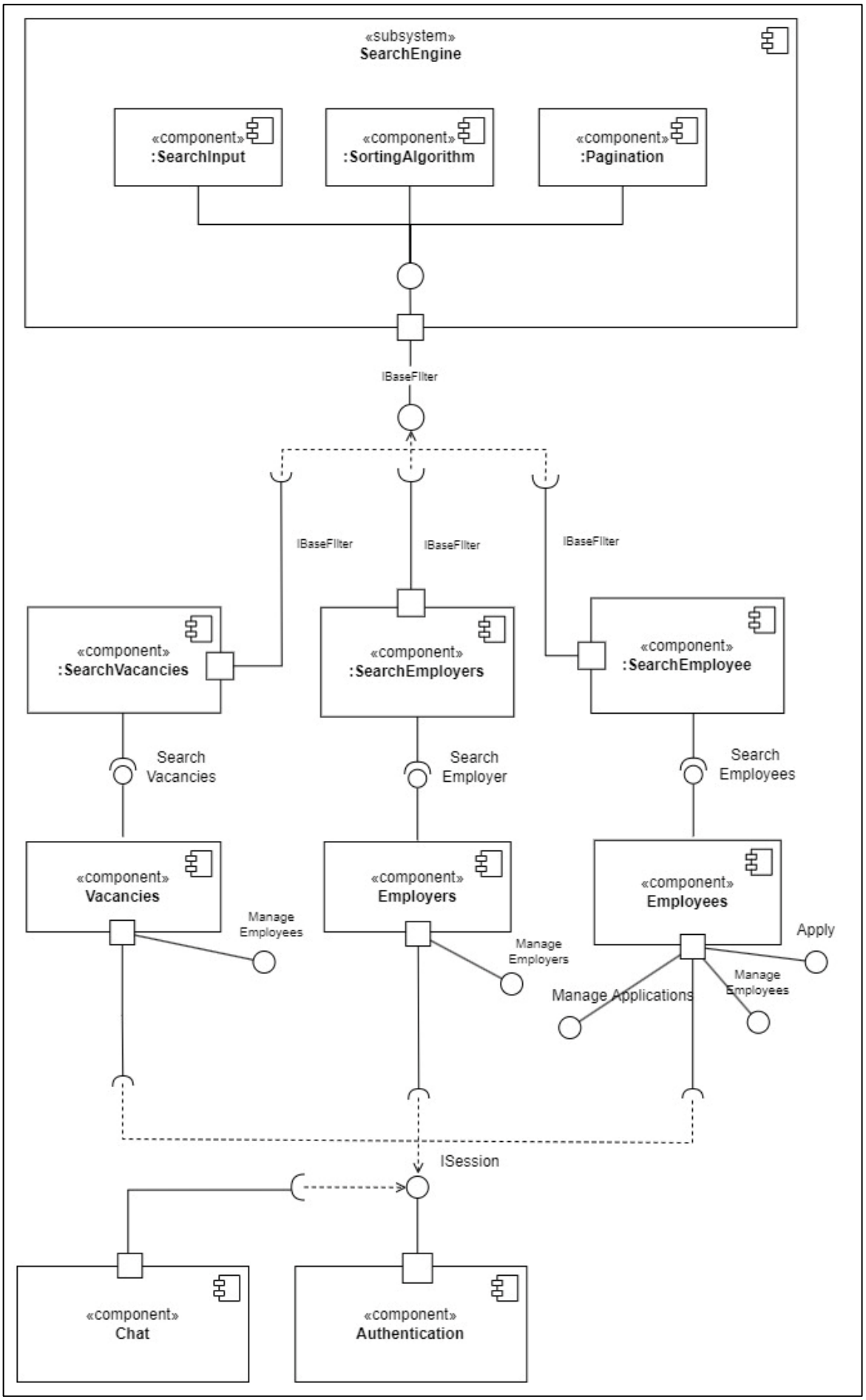


Рисунок 2.2 – Діаграма компонентів

Компонент «SearchEmployee». Використовує функції «SearchEngine» для забезпечення пошуку кандидатів. Додає можливості фільтрації за навичками, рівнем освіти, досвідом роботи та іншими професійними характеристиками. Інтегрується з компонентом «Employees» для аналізу даних профілів кандидатів.

Компонент «Vacancies». Основний модуль для управління вакансіями. Дозволяє створювати, редагувати, публікувати та видаляти вакансії. Інтегрується з компонентами «SearchEngine» та «SearchVacancies», щоб забезпечити швидкий і точний доступ до інформації про вакансії. Додає інструменти для аналізу переглядів і заявок на вакансії.

Компонент «Employers». Управляє профілями компаній, дозволяючи їм редагувати інформацію, переглядати кандидатів та взаємодіяти з ними. Інтегрується з компонентами «SearchEmployers» і «Vacancies», щоб забезпечити доступ до детальної інформації про компанії та їхні відкриті вакансії.

Компонент «Employees». Відповідає за створення, зберігання та управління профілями кандидатів. Дозволяє кандидатам заповнювати резюме, додавати інформацію про навички та досвід роботи. Інтегрується з компонентом «SearchEmployee», щоб забезпечити ефективний пошук відповідних вакансій для користувачів.

Компонент «Chat». Забезпечує комунікацію між користувачами платформи в реальному часі. Використовує інструмент «SignalR» для інтерактивного обміну повідомленнями між кандидатами та роботодавцями. Інтегрується з компонентами «Authentication» для перевірки доступу та «Employees» для персоналізації спілкування.

Інтерфейс «BaseFilter». Базовий інтерфейс для реалізації фільтрації даних. Використовується в компонентах «SearchVacancies», «SearchEmployers» і «SearchEmployee», щоб забезпечити універсальність та масштабованість пошукових функцій. Сприяє уніфікації логіки фільтрації та оптимізації процесу обробки запитів.

Представлена діаграма компонентів демонструє архітектуру системи з чітко визначеними модулями та їхніми взаємозв’язками. Завдяки модульності та інтеграції кожного компонента платформа забезпечує гнучкість, масштабованість і простоту подальшого розширення функціоналу. Це дозволяє ефективно вирішувати задачі користувачів і адаптувати систему до змінних вимог ринку.

2.1.3 Проєктування за допомогою діаграми діяльності

Діаграма діяльності є важливим інструментом для моделювання процесів у системі. Вона дозволяє візуалізувати послідовність дій, їх взаємозв’язки та логіку виконання, а також ухвалення рішень. Цей тип діаграми широко використовується для аналізу бізнес-процесів, проєктування алгоритмів і уточнення функціональних вимог системи. Завдяки своїй універсальності, діаграма діяльності допомагає виявляти слабкі місця в процесах, а також оптимізувати послідовності дій для досягнення ефективнішого результату.

Діаграма діяльності призначена для:

– моделювання послідовності дій: діаграма дає змогу детально демонструвати кожен етап виконання процесу, починаючи з початкової дії та закінчуючи фінальним результатом – це дозволяє чітко визначити порядок дій, що забезпечує послідовність у роботі системи;
– аналізу процесів ухвалення рішень: відображає різні сценарії виконання дій залежно від умов і можливих рішень – використання умовних розгалужень дозволяє моделювати альтернативні шляхи розвитку процесів, що є важливим для підготовки до різних ситуацій;
– оптимізація процесів: допомагає ідентифікувати дублюючі чи зайві етапи, а також вузькі місця, що уповільнюють виконання завдань –

оптимізація таких процесів дозволяє скоротити час виконання та покращити результативність;

– забезпечення прозорості: слугує спільною мовою для різних учасників проєкту – розробників, аналітиків, тестувальників і замовників. Завдяки візуалізації процесів спрощується спільне розуміння логіки роботи системи та її взаємозв’язків.

На рисунку 2.3 зображено діаграму діяльності, в якому демонструється ключовий сценарій, такий як процес найму, який охоплює дії як шукача роботи, так і рекрутера. Вона демонструє основні етапи взаємодії, такі як пошук вакансій, подання заявок, тестування кандидатів і ухвалення рішень.

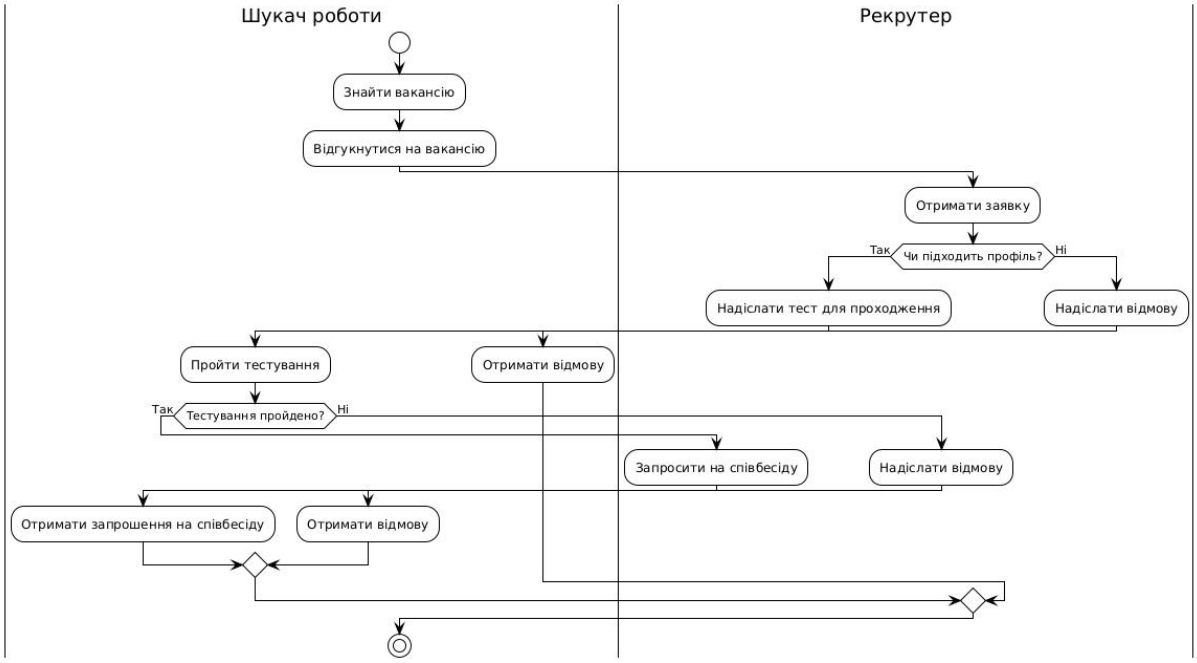


Рисунок 2.3 – Діаграма діальності

Згідно діаграмі діяльності, процес найму проходить в такому порядку:

а) крок «Знайти вакансію»: шукач роботи ініціює процес пошуку доступних вакансій;

б) крок «Відгукнутися на вакансію»: шукач роботи подає заявку на обрану вакансію;

в) крок «Отримати заявку»: рекрутер отримує заявку від шукача роботи;

г) крок «Чи підходить профіль?»: рекрутер аналізує заявку та ухвалює рішення щодо відповідності кандидата вимогам вакансії;
д) крок «Надіслати відмову»: якщо профіль не підходить, рекрутер надсилає повідомлення про відмову;
е) крок «Надіслати тест для проходження»: якщо профіль підходить, рекрутер надсилає кандидату тестове завдання;
ж) крок «Пройти тестування»: кандидат виконує тестове завдання;
з) крок «Тестування пройдено?»: рекрутер оцінює результати тесту, ухвалюючи подальше рішення;
и) крок «Запросити на співбесіду»: якщо тестування пройдено успішно, рекрутер запрошує кандидата на співбесіду;
к) крок «Отримати відмову»: якщо тестування не пройдено, кандидат отримує повідомлення про відмову;
л) крок «Отримати запрошення на співбесіду»: у разі успішного завершення всіх етапів кандидат отримує запрошення на співбесіду.

Процес завершується одним із двох результатів: кандидат або отримує запрошення на співбесіду після успішного проходження всіх етапів, або отримує відмову, що завершує його взаємодію з поточною вакансією.

2.1.4 Проєктування за допомогою діаграми станів

Діаграма станів використовується для детального опису життєвого циклу об’єкта в системі, представляючи його стани, переходи між ними та події, які ці переходи викликають. Вона забезпечує можливість моделювання поведінки об’єкта в різних ситуаціях, а також демонструє, як цей об’єкт реагує на зовнішні або внутрішні події. Це допомагає краще розуміти логіку системи й оптимізувати її роботу.

На рисунку 2.4 зображена діаграма станів яка відображає життєвий цикл заявки на вакансію, включаючи всі ключові стани, переходи між ними та події,

що ці переходи ініціюють. Вона демонструє модульність і гнучкість процесу роботи з заявками, забезпечуючи структуроване та зрозуміле уявлення про всі можливі сценарії розвитку подій. Така діаграма дає змогу чітко візуалізувати процеси, що сприяє виявленню можливих недоліків і їх усуненню на ранніх етапах.

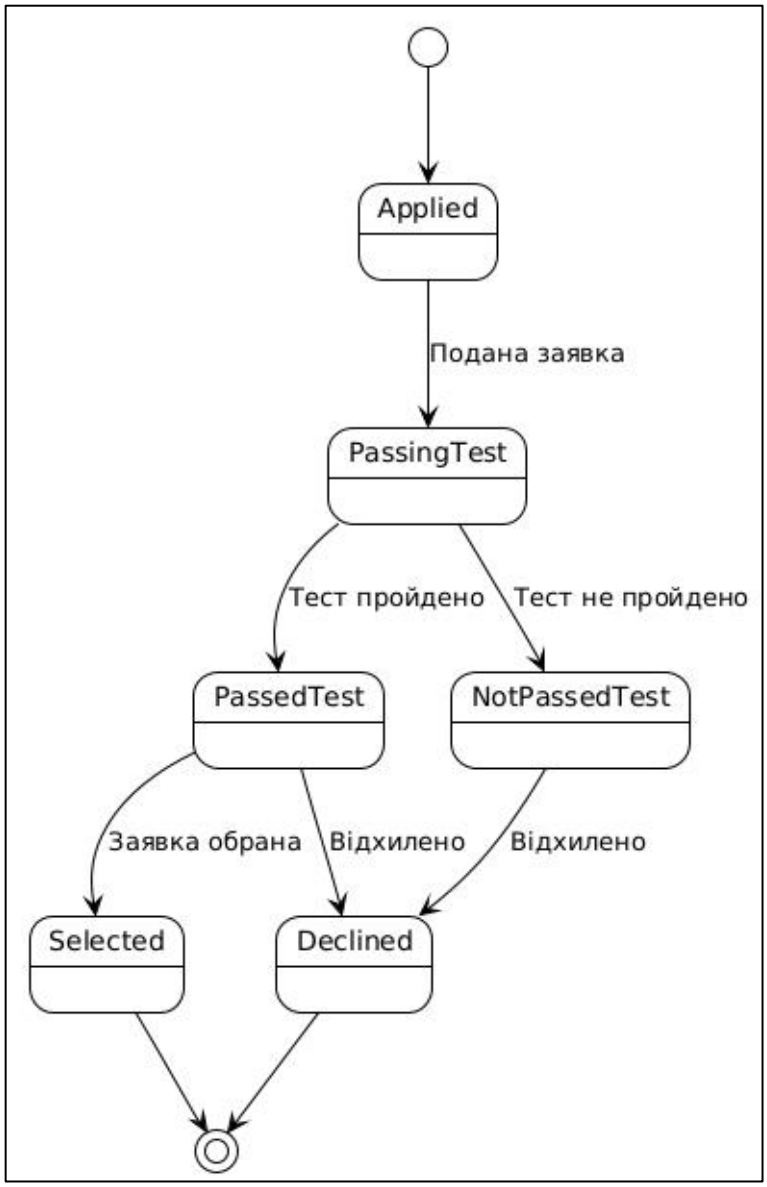


Рисунок 2.4 – Діаграма станів

Життєвий цикл заявки на вакансію проходить через такі стани:

а) **Applied (подана заявка)**: початковий стан, у якому заявка створюється й подається шукачем роботи, щоб розпочати процес розгляду;

б) **PassingTest (проходження тестування)**: стан, у якому заявка опиняється, якщо рекрутер вирішує перевірити кандидата за допомогою тесту для оцінки його професійних навичок;

в) **PassedTest (тест пройдено)**: стан, який заявка набуває у разі успішного проходження тестування – цей результат відкриває шлях до наступних етапів розгляду;

г) **NotPassedTest (тест не пройдено)**: стан, який заявка отримує, якщо тестування було провалено – у цьому випадку заявка автоматично завершується з позначкою «Відхилено»;

д) **Selected (заявка обрана)**: стан, у якому заявка опиняється після того, як рекрутер ухвалює позитивне рішення, наприклад, запрошення на співбесіду або пропозиція роботи;

е) **Declined (відхилено)**: стан, у якому заявка опиняється, якщо рекрутер ухвалює рішення про відмову після завершення розгляду.

Переходи між станами відбуваються за таких умов:

– з «Applied» у «PassingTest»: відбувається, коли рекрутер надсилає запит на проходження тесту;
– з «PassingTest» у «PassedTest» або «NotPassedTest»: залежно від результату тестування, визначається, чи пройшов кандидат тест;
– з «PassedTest» у «Selected»: відбувається, коли рекрутер ухвалює позитивне рішення щодо заявки після успішного тесту;
– з «PassedTest» або «NotPassedTest» у «Declined»: відбувається, якщо заявка не відповідає подальшим вимогам або критеріям рекрутера;
– життєвий цикл заявки завершується, досягаючи одного з двох результатів: «Selected», якщо заявка була успішною, або «Declined», якщо заявка не відповідала вимогам.

Ця діаграма станів надає чітку візуалізацію всіх можливих станів заявки, а також забезпечує структуроване розуміння процесу від подання заявки до її остаточного завершення. Завдяки такій структурі можна легко виявити слабкі місця у процесах і покращити їх для підвищення ефективності роботи системи.

2.2 Архітектура

Для розробки клієнтської частини було застосовано Flux архітектуру. Flux – це архітектурний підхід, який забезпечує односпрямований потік даних у застосунку, спрощуючи управління станом і підвищуючи передбачуваність поведінки системи. Даний архітектурний підхід пропонує зберігати увесь стан програми в одному місці, що спрощує його доступ і модифікацію. Потік даних завжди рухається в одному напрямку, що запобігає складним і непередбачуваним змінам стану. Чітке розділення відповідальностей між компонентами системи полегшує розуміння та підтримку коду [11].

Архітектура Flux вирішує такі проблеми:

– управління складним станом у великих додатках;
– запобігання конфліктам під час модифікації стану;
– полегшення відстеження змін у стані для дебагінгу та тестування.

Розглянемо основні складові Flux архітектури:

– «Дії (Actions)»: об’єкти, що описують події, які відбуваються у застосунку (наприклад, «Додати вакансію» або «Оновити профіль»);
– «Диспетчер (Dispatcher)»: центральний вузол, що приймає дії та передає їх до сховищ;
– «Сховище (Store)»: місце для зберігання стану програми (сховище також містить бізнес-логіку, яка визначає, як стан змінюється у відповідь на дії);
– «Представлення (View)»: компоненти інтерфейсу користувача, які отримують дані зі сховища та оновлюються при зміні стану.

Принцип однонаправленого потоку даних в архітектурі Flux складається з таких кроків (див. рис. 2.5) [11]:

а) користувач викликає дію (наприклад, натискає кнопку «Додати вакансію»);
б) дія створюється та передається диспетчеру;
в) диспетчер приймає дію та передає її у відповідне сховище;

г) сховище обробляє дію, змінюючи стан;

д) зміни стану передаються у відповідні компоненти представлення;

е) представлення оновлюються для відображення нового стану.

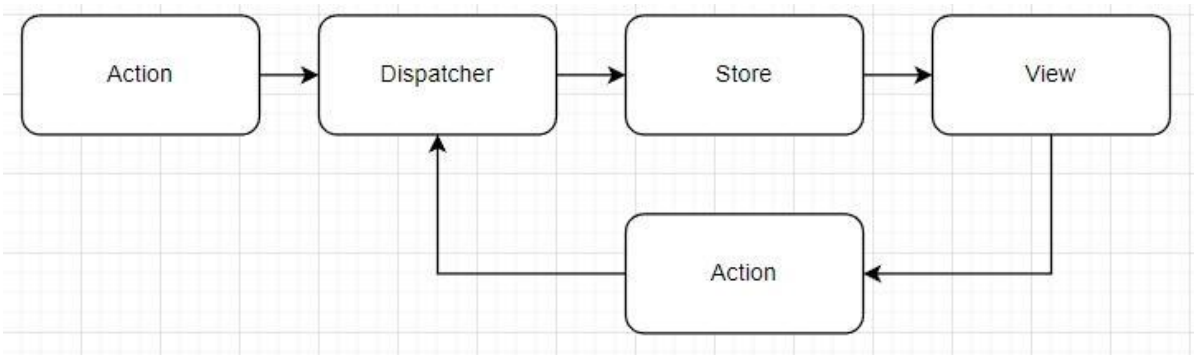


Рисунок 2.5 – Схема однонаправленого потоку в архітектурі Flux

Flux дозволяє побудувати масштабовану, структуровану та передбачувану архітектуру, забезпечуючи ефективне управління станом у застосунку.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Опис інструментів для розробки клієнтської частини

3.1.1 Мова програмування TypeScript

TypeScript – це надбудова над JavaScript, яка додає статичну типізацію. Вона використовується для створення надійних, масштабованих і зрозумілих додатків [6].

Мова програмування TypeScript має такі переваги:

– забезпечення перевірки типів на етапі компіляції, знижуючи кількість помилок у коді;
– інтеграція з сучасними редакторами коду, такими як Visual Studio Code, допомагає швидше писати код завдяки підказкам;
– повна підтримка стандарту «ES6» або «ESNext» і можливість додавання власних унікальни функцій, такі як інтерфейси та декоратори;
– можливість компіляції у звичайний JavaScript, що дозволяє використовувати його в будь-якому середовищі.

3.1.2 Бібліотека React

React – це бібліотека JavaScript для створення користувацьких інтерфейсів. Вона використовується для побудови компонентної архітектури додатка. Бібліотека React має такі переваги над іншими інструментами для розробки клієнтської частини [3]:

– компонентний підхід: дозволяє створювати повторно використовувані компоненти, що спрощує розробку та підтримку;

– механізм Virtual DOM: забезпечує високу продуктивність за рахунок ефективного оновлення лише змінених частин DOM;
– однонаправлений потік даних: спрощує управління станом і дебагінг;
– екосистема: React має багатий набір додаткових бібліотек і інструментів для розробки.

Для відображення та оновлення користувацького інтерфейсу, React використовує внутрішній механізм для оптимізації оновлень DOM, який називається алгоритмом порівняння (reconciliation) [3].

Алгоритм порівняння складається з таких принципів:

а) React створює Virtual DOM для збереження стану віртуального представлення інтерфейсу;
б) під час змін у стані або пропсах компонента React порівнює новий Virtual DOM зі старим;
в) тільки змінені частини передаються в реальний DOM для оновлення, що забезпечує високу продуктивність.

Для роботи React у веббраузерах, також необхідно налаштувати бібліотеку ReactDOM. Вона відповідає за рендеринг компонентів React у реальний DOM. ReactDOM також обробляє події, делегуючи їх із кореневого елемента до дочірніх компонентів.

3.1.3 Бібліотека для верстки інтерфейсу MaterialUI

MaterialUI – це набір компонентів React, що реалізують принципи дизайну Material Design від корпорації Google [4].

Даний інструмент має такі переваги:

– готові компоненти забезпечують швидке створення адаптивних інтерфейсів;
– кастомізація надає можливість налаштовувати теми, кольори та стилі під потреби проєкту;

– компоненти можна використовувати окремо або в комбінації з іншими бібліотеками.

3.1.4 Інструмент для збірки ViteJS

Vite – це інструмент для створення JavaScript-додатків, який забезпечує швидкий і ефективний процес розробки [5].

Даний інструмент містить такі можливості:

– миттєвий запуск сервера розробки: завдяки використанню нативних ES-модулів, Vite дозволяє миттєво запускати сервер розробки без необхідності попереднього збирання коду;
– гаряче оновлення модулів (HMR): Vite забезпечує надзвичайно швидке оновлення модулів у реальному часі, що дозволяє розробникам бачити зміни без перезавантаження сторінки;
– вбудована підтримка сучасних технологій: Vite з коробки підтримує TypeScript, JSX, CSS та інші сучасні вебтехнології, спрощуючи налаштування середовища розробки;
– оптимізована збірка: використовуючи Rollup, Vite створює високоефективні статичні ресурси для продакшн-середовища.

3.1.5 Tanstack Query

TanStack Query – це бібліотека для управління асинхронним станом у React-додатках. Вона спрощує процес отримання, кешування, синхронізації та оновлення даних із сервера, що значно покращує досвід як розробників, так і користувачів [8]. Основні можливості TanStack Query:

– декларативний підхід до запитів: дозволяє розробникам описувати, які дані потрібні додатку, а бібліотека автоматично обробляє процес їх отримання та оновлення;

– автоматичне керування станом запитів: TanStack Query самостійно обробляє кешування, фонові оновлення та застарілі дані без необхідності додаткової конфігурації;
– підтримка мутацій: дозволяє легко виконувати зміни на сервері та автоматично оновлювати стан додатка після успішного виконання мутацій;
– інтеграція з TypeScript: забезпечує повну підтримку TypeScript, що дозволяє використовувати типізацію для запитів і мутацій, підвищуючи надійність коду;
– підтримка серверного рендерингу (SSR): TanStack Query може бути використана для попереднього отримання даних на сервері, що покращує продуктивність і SEO.

Переваги використання TanStack Query [8]:

– зменшення складності коду: замість ручного управління станом запитів, помилками та кешуванням, TanStack Query бере на себе ці завдання, дозволяючи розробникам зосередитися на бізнес-логіці;
– покращення продуктивності: завдяки ефективному кешуванню та фоновим оновленням, додатки можуть швидше реагувати на дії користувачів і зменшувати навантаження на сервер;
– покращений досвід користувачів: автоматичне оновлення даних і обробка помилок забезпечують більш плавний і надійний інтерфейс для кінцевих користувачів.

3.1.6 Zustand

Zustand – це легка бібліотека для управління станом у React-додатках, яка створена для спрощення роботи зі станом та зменшення складності коду. Вона забезпечує інтуїтивний API, високу продуктивність і гнучкість у використанні [9].

Основні можливості Zustand включають:

– централізоване управління станом: Zustand дозволяє створювати глобальний стан для всього додатка, що забезпечує легкий доступ до даних із будь-якого компонента;
– проста інтеграція: бібліотека інтегрується з іншими інструментами та бібліотеками без необхідності складної конфігурації;
– ефективність: Zustand мінімізує повторні рендери компонентів, оновлюючи тільки ті, які залежні від змінених даних;
– гнучкість: розробники можуть створювати як прості сховища даних, так і складні логіки управління станом;
– можливість використання без React: Zustand можна застосовувати в будь-якому JavaScript-додатку, а не лише у React;
– система підписок: компоненти автоматично підписуються на зміни тільки тих даних, які їм потрібні, що підвищує продуктивність.

3.1.7 React Router

React Router – це бібліотека для керування маршрутизацією в React-додатках. Вона дозволяє створювати багатосторінкові інтерфейси без перезавантаження сторінки, що забезпечує плавний користувацький досвід.

Основні можливості React Router [7]:

– декларативне визначення маршрутів: маршрути задаються за допомогою JSX, що спрощує їх розуміння та підтримку;
– підтримка динамічних маршрутів: дозволяє створювати маршрути з параметрами, що робить додаток більш гнучким;
– інтеграція з React-компонентами: маршрути легко поєднуються з компонентами, що спрощує розробку;
– підтримка вкладених маршрутів: дозволяє створювати складні структури навігації.

React Router вирішує такі проблеми:

– управління навігацією в односторінкових додатках (SPA): забезпечує зміну вмісту без перезавантаження сторінки;
– синхронізація інтерфейсу з URL: дозволяє відображати певний стан додатка відповідно до URL-адреси;
– обробка складних сценаріїв навігації: підтримує редиректи, захищені маршрути та інші складні випадки.

React Router складається з таких компонентів [7]:

– BrowserRouter: обгортає додаток і забезпечує доступ до історії навігації;
– Routes: містить набір маршрутів додатка;
– Route: визначає шлях і відповідний компонент для рендерингу;
– Link: створює посилання для навігації між маршрутами.

Порядок обробки маршрутів у React Router:

– користувач натискає на посилання або вводить URL у браузері;
– компонент BrowserRouter перехоплює подію та оновлює історію навігації;
– компонент Routes перевіряє, який Route відповідає поточному шляху;
– відповідний компонент рендериться, відображаючи потрібний вміст.

Використання React Router дозволяє створювати динамічні та інтерактивні інтерфейси, покращуючи загальний досвід користувачів.

3.1.8 SignalR

SignalR – це бібліотека, яка дозволяє реалізувати транспортування даних між двома або більше платформами в реальному часі, забезпечуючи миттєвий зв’язок між клієнтом і сервером. Її основне призначення – обмін даними у реальному часі без необхідності постійного оновлення сторінок або створення складних механізмів запитів [12].

Інструмент SignalR підтримує такі можливості:

– двосторонній зв’язок: SignalR дозволяє серверам надсилати дані клієнтам та отримувати їх у відповідь у реальному часі;
– обробка подій у реальному часі: бібліотека забезпечує миттєву доставку повідомлень, змін стану чи будь-яких інших даних від сервера до клієнта та навпаки;
– підтримка різних транспортних механізмів: SignalR вибирає найефективніший транспортний протокол для з’єднання (WebSocket, Server-Sent Events або Long Polling) залежно від можливостей браузера чи сервера;
– груповий зв’язок: SignalR дозволяє створювати групи користувачів для надсилання повідомлень одночасно декільком клієнтам.

3.2 Процес створення користувацького інтерфейсу

3.2.1 Конфігурація теми MaterialUI в React

Для забезпечення зручного інтерфейсу у вебзастосунку використовувалася бібліотека MaterialUI. За її допомогою було налаштовано глобальну тему для кастомізації стилів, кольорів, шрифтів та інших параметрів компонента. Конфігурація теми дозволяє легко підтримувати єдиний стиль дизайну застосунку, а також спрощує внесення змін до інтерфейсу.

Кроки реалізації власної теми:

а) імпортування необхідних інструментів для створення теми;
б) налаштування основної палітри кольорів, шрифтів та інших стилів у темі;
в) обгортання додатка у компонент «ThemeProvider», щоб усі компоненти мали доступ до заданої теми.

3.2.2 Створення React компонентів

Процес створення компонента в React є базовою операцією, яка дозволяє створювати інтерфейс користувача на основі модульного підходу. Компоненти в React можуть бути функціональними або класовими, однак на сучасному етапі розробки перевага надається функціональним компонентам завдяки їх простоті та підтримці хуків. Функціональні компоненти – це функції, що приймають props як аргументи і повертають розмітку у форматі JSX. Вони є легкими для створення та підтримки, що робить їх оптимальним вибором для більшості застосунків.

3.2.3 Обробка даних в реальному часі

Використання SignalR складається з чітких етапів: налаштування хабу на сервері, встановлення з’єднання на клієнті, обробки вхідних та вихідних даних, а також управління групами клієнтів. Завдяки такій структурі бібліотека дозволяє легко реалізувати обробку даних в реальному часі, забезпечуючи швидке оновлення даних, взаємодію між користувачами та підтримку стабільного зв’язку.

Основні етапи використання SignalR:

а) на сервері необхідно має бути налаштований Hub (хаб) – спеціальний об’єкт, який служить точкою обміну повідомленнями між клієнтами та сервером (хаб дозволяє клієнтам викликати серверні методи та навпаки);

б) клієнт повинен ініціалізувати з’єднання із сервером через об’єкт «HubConnection»;

в) клієнт визначає URL хабу на сервері, а також конфігурує логування або інші налаштування;

г) з’єднання налаштовується за допомогою «HubConnectionBuilder» API;

д) після підключення до хабу клієнт реєструє обробники подій для прийому повідомлень від сервера.

3.3 Тестування системи

Тестування програмного забезпечення є важливим етапом розробки, який полягає у перевірці коректності роботи системи відповідно до вимог, виявленні помилок та оцінці її продуктивності. Воно включає в себе аналіз функціонування як клієнтської, так і серверної частини, забезпечуючи повноцінну перевірку всіх компонентів.

Метою тестування є виявлення потенційних недоліків, які можуть вплинути на якість, надійність або ефективність системи.

Для тестування системи та оцінки відповідності очікувань використовуються різні типи тестів, такі як *функціональні*, *регресійні*, *навантажувальні*. Завдяки тестуванню вдається мінімізувати ризики, оптимізувати роботу системи та гарантувати її стабільність перед випуском у продуктивне середовище.

3.3.1 Запуск системи у веббраузері

Під час розробки були враховані специфікації основних браузерів, щоб забезпечити стабільну роботу системи та високий рівень комфорту для користувачів.

Система була протестована в таких популярних браузерах, як Chrome, Firefox, Safari та Microsoft Edge, і її функціональність підтверджена для всіх зазначених середовищ.

На рисунках 3.1 – 3.3 зображено роботу платформи у веббраузерах Microsoft Edge та Chrome.

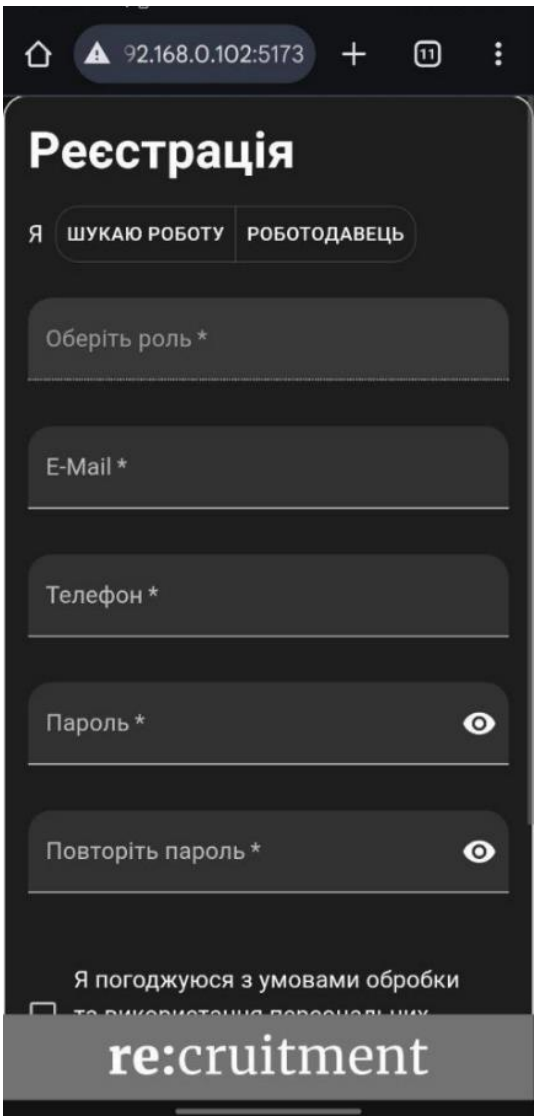


Рисунок 3.1 – Вхід до системи через мобільний веббраузер Chrome

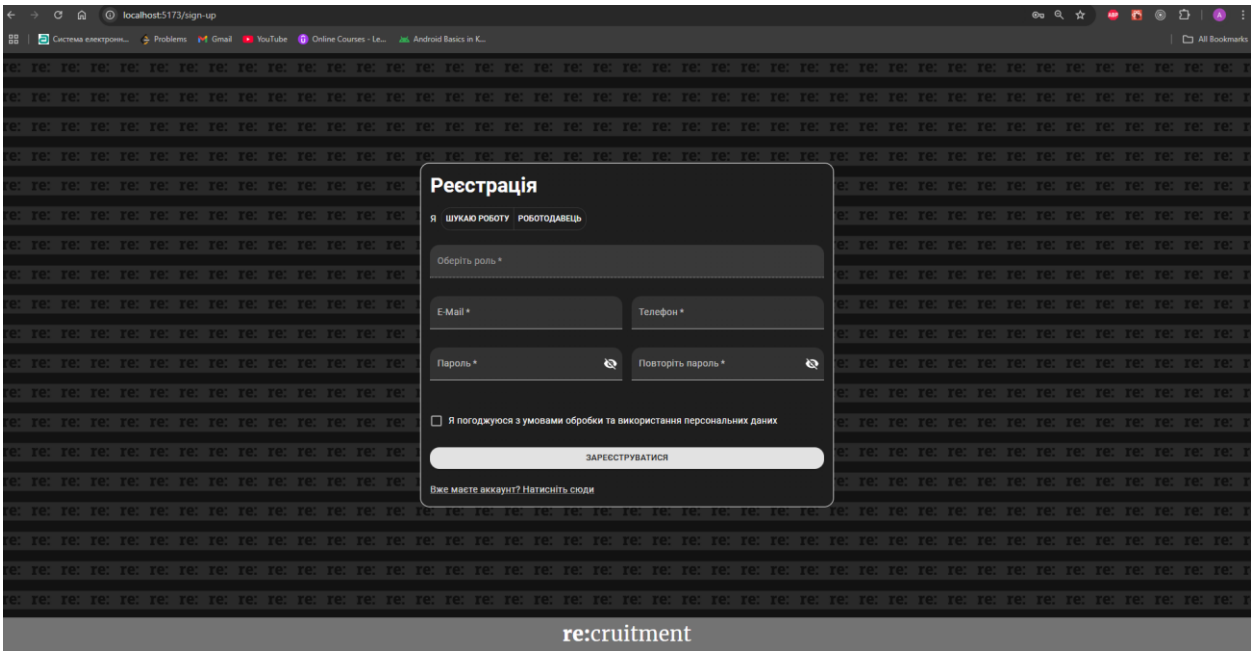


Рисунок 3.2 – Вхід до системи через веббраузер Chrome

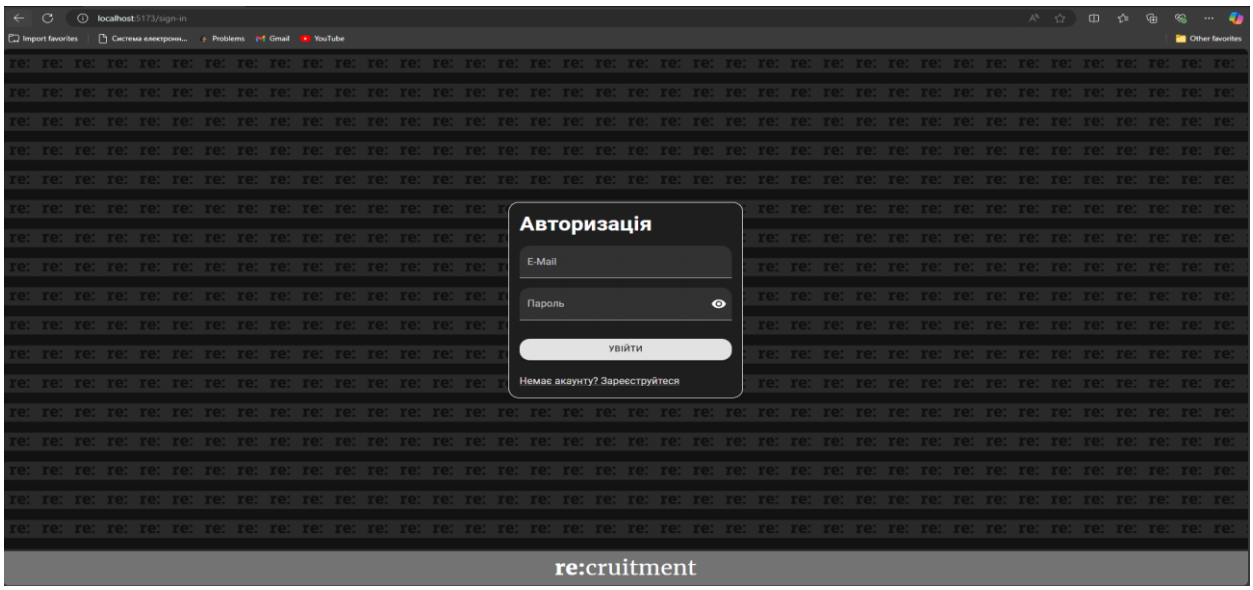


Рисунок 3.3 – Вхід до системи через веббраузер Microsoft Edge

3.3.2 Тестування адаптації верстки

Для забезпечення коректного відображення інтерфейсу на екранах різного розміру було проведено ретельне тестування адаптивності верстки.

На рисунках 3.4 – 3.6 зображено приклад сторінки з деталями вакансії на трьох розмірах екрану.

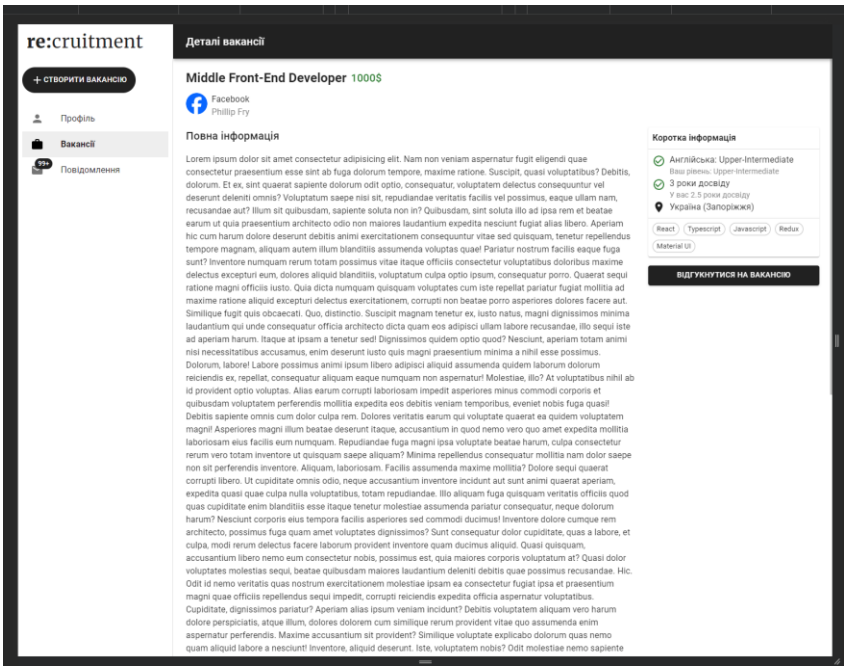


Рисунок 3.4 – Відкрита сторінка «Деталі вакансії» на комп’ютері

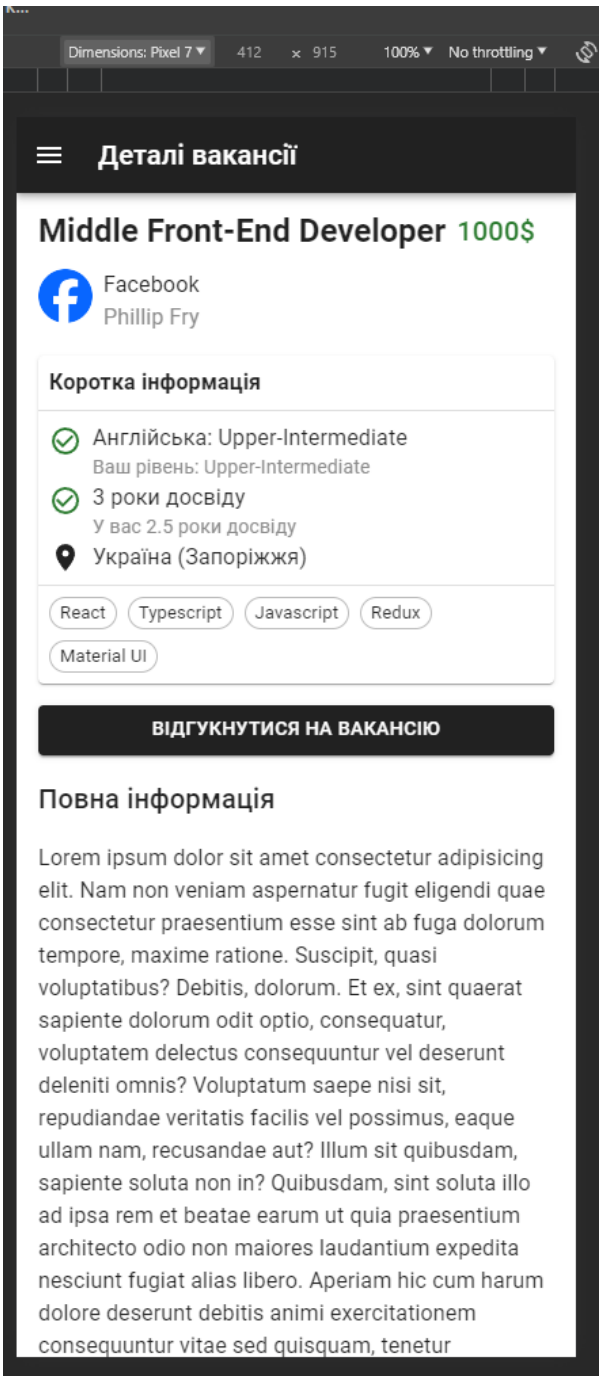


Рисунок 3.5 – Відкрита сторінка «Деталі вакансії»
на мобільному пристрої

У ході перевірок оцінювалися параметри адаптації для різних типів пристроїв і орієнтацій екранів, щоб переконатися у правильному масштабуванні та зручності взаємодії.

Результати тестування підтвердили, що інтерфейс повністю відповідає сучасним стандартам адаптивності та забезпечує комфортний користувацький досвід незалежно від умов використання.

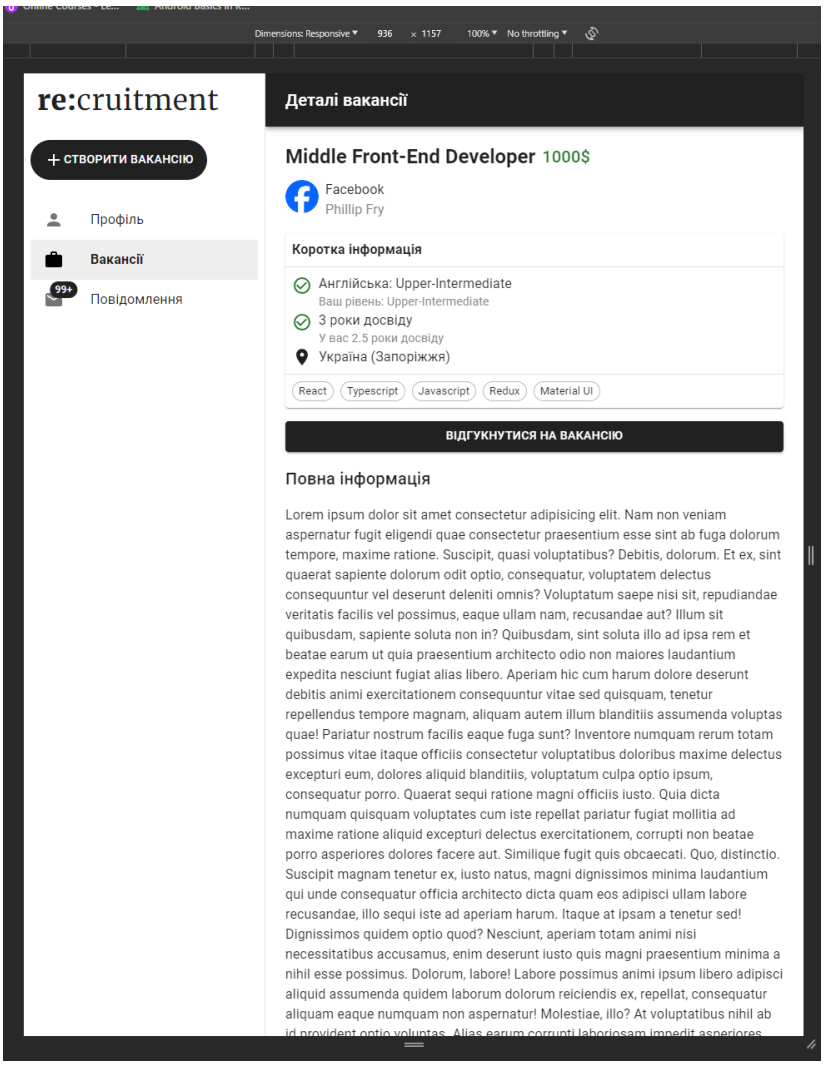


Рисунок 3.6 – Відкрита сторінка «Деталі вакансії» на планшеті

3.4 Керівництво користувача

3.4.1 Рівень підготовки користувача

Для роботи на сайті користувач повинен мати певну кваліфікацію, включаючи навички роботи з ПК та web-браузером. Це охоплює вміння працювати з операційною системою, встановлювати програми, працювати з файловою системою, а також розуміння основних понять та операцій в браузері. Крім того, знайомство з керівництвом користувача допоможе зрозуміти функціонал сайту та використовувати його відповідно до інструкцій.

3.4.2 Підготовка до роботи

Для запуску системи необхідний доступ до сайту через інтернет за допомогою стандартного web-браузера, такого як Google Chrome, Mozilla Firefox, Opera або Safari.

Після входу на сайт користувач буде перенаправлений на сторінку входу до системи.

На сайті розрізняються дві групи користувачів: анонімні користувачі, які не увійшли або не зареєстровані, і користувачі – приватні особи, які авторизувалися на сайті та мають доступ до функцій користувача у системі.

3.4.3 Авторизація в систему

На рисунку 3.7 відображено форму авторизації, яка розташована по центру сторінки. Інтерфейс виконаний у мінімалістичному стилі з темним фоном та світлим текстом, що забезпечує зручність використання. Ця сторінка є початковим кроком для отримання доступу до повного функціоналу платформи.

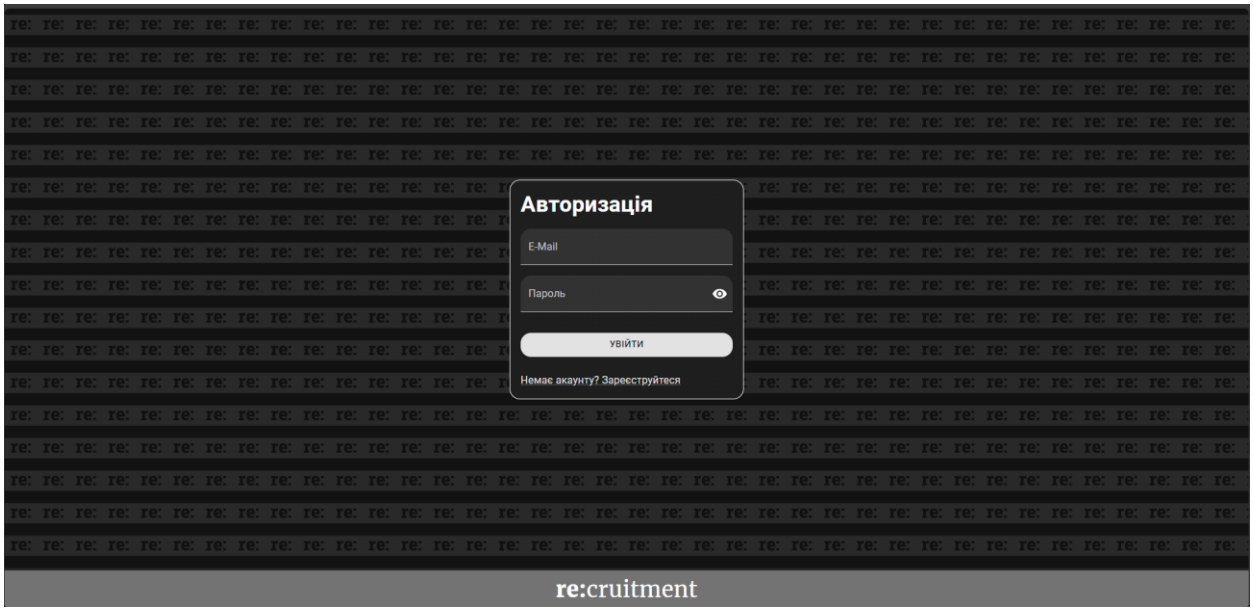


Рисунок 3.7 – Авторизація в систему

Форма складається з таких основних елеменітв:

– поле «E-Mail»: текстове поле для введення адреси електронної пошти;
– поле «Пароль»: текстове поле для введення пароля (поруч із полем розташована іконка «око» для перегляду або приховування введеного пароля);
– кнопка «Увійти»: основна кнопка для підтвердження введених даних і виконання спроби входу у систему (після натискання відбувається перевірка введеного логіна та пароля – у разі помилки буде відображено відповідне повідомлення);
– посилання «Немає акаунту? Зареєструйтеся»: доступне під кнопкою «Увійти» – при натисканні користувач буде перенаправлений на сторінку реєстрації для створення нового облікового запису.

Забезпечення простоти та інтуїтивності інтерфейсу є ключовою метою розробника. Завдяки мінімалістичному дизайну та зрозумілим елементам користувачі можуть швидко виконати авторизацію або перейти до процесу реєстрації, що покращує загальний досвід взаємодії з платформою.

3.4.4 Реєстрація нового акаунту

На рисунку 3.8 зображено форму реєстрації для створення нового облікового запису, яка розташована по центру сторінки. Інтерфейс виконаний у мінімалістичному стилі з темним фоном та світлим текстом, що робить процес реєстрації зручним та зрозумілим для користувача.

Форма реєстрації забезпечує зручний та інтуїтивно зрозумілий інтерфейс для обох категорій користувачів: тих, хто шукає роботу, та роботодавців. Динамічна адаптація полів відповідно до обраного типу облікового запису дозволяє користувачеві вводити лише необхідну інформацію, що оптимізує процес заповнення. Поля для пароля з функцією його перегляду сприяють мінімізації помилок під час введення, а обов’язкове

підтвердження політики обробки персональних даних забезпечує дотримання правових норм. Такий підхід сприяє підвищенню ефективності та надійності реєстрації нових користувачів на платформі.

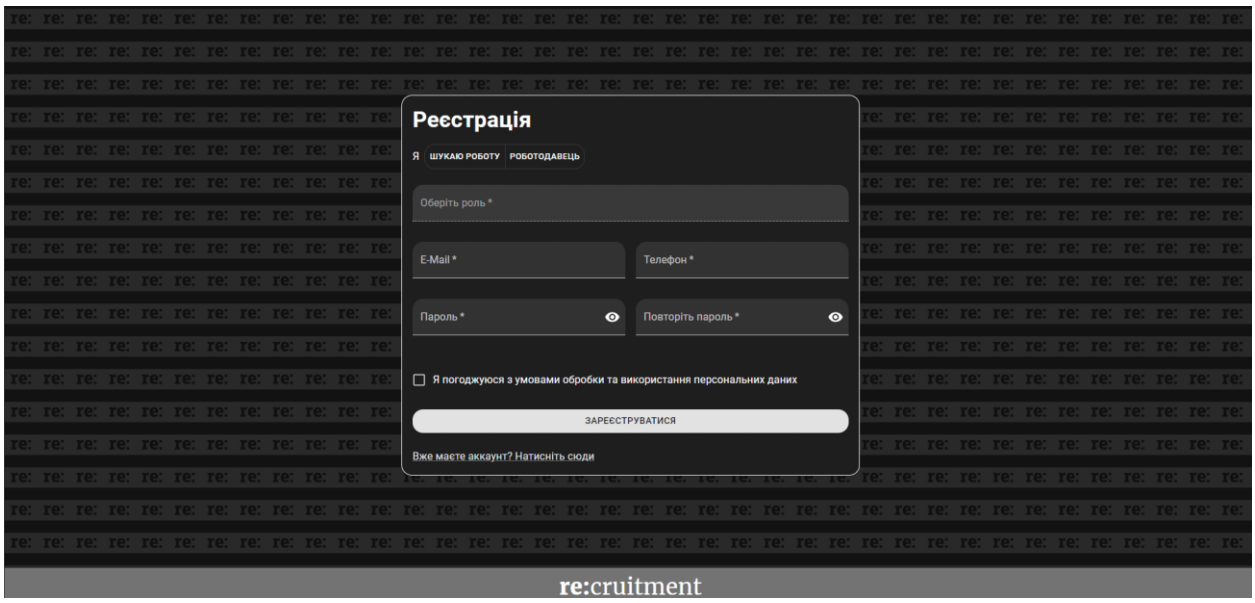


Рисунок 3.8 – Реєстрація нового акаунту

Форма складається з таких основних елементів:

– перемикачі «Шукаю роботу» або «Роботодавець»: завдяки ним, ми перемикаємо тип облікового запису;
– вимкнуте поле «Оберіть роль» відображається тоді, коли користувач ще не обрав тип облікового запису – якщо буде обрано тип «Шукаю роботу», то замість поля «Оберіть роль» відобразяться три поля для введення ПІБ користувача, інакше – назви компанії, яку представляє роботодавець;
– поле «E-Mail»: текстове поле для введення адреси електронної пошти;
– поле «Телефон»: текстове поле для введення номера мобільного телефону;
– поле «Пароль»: текстове поле для введення пароля – поруч із полем розташована іконка «око» для перегляду або приховування введеного пароля;

– поле «Повторіть пароль»: текстове поле для підтвердження введеного пароля з метою уникнення помилок;
– чекбокс «Я погоджуюсь з умовами обробки та використання персональних даних»: обов’язковий елемент для підтвердження згоди користувача з політикою платформи;
– кнопка «Зареєструватися»: основна кнопка для завершення процесу створення облікового запису – при натисканні система перевіряє правильність введених даних та надає доступ до платформи;
– посилання «Вже маєте акаунт? Поверніться сюди»: дозволяє користувачу повернутися до сторінки авторизації для входу у вже існуючий обліковий запис.

3.4.5 Особистий кабінет кандидата

На рисунках 3.9 – 3.10 відображено сторінку профілю користувача-кандидата, що складається з декількох інформаційних блоків. Інтерфейс має структуроване оформлення, що дозволяє користувачу переглядати, оновлювати та вводити дані про себе.

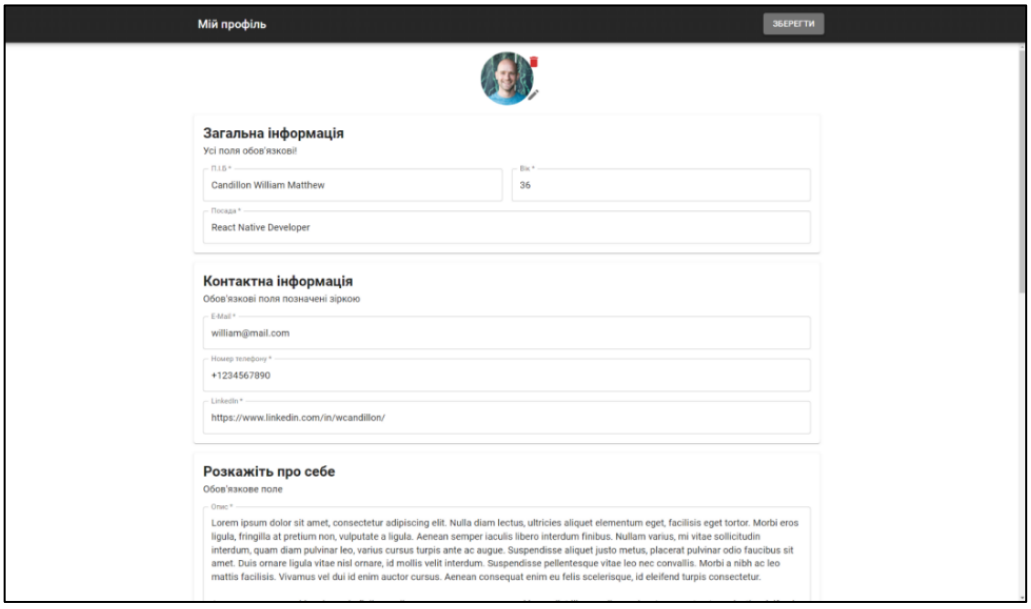


Рисунок 3.9 – Профіль Кандидата

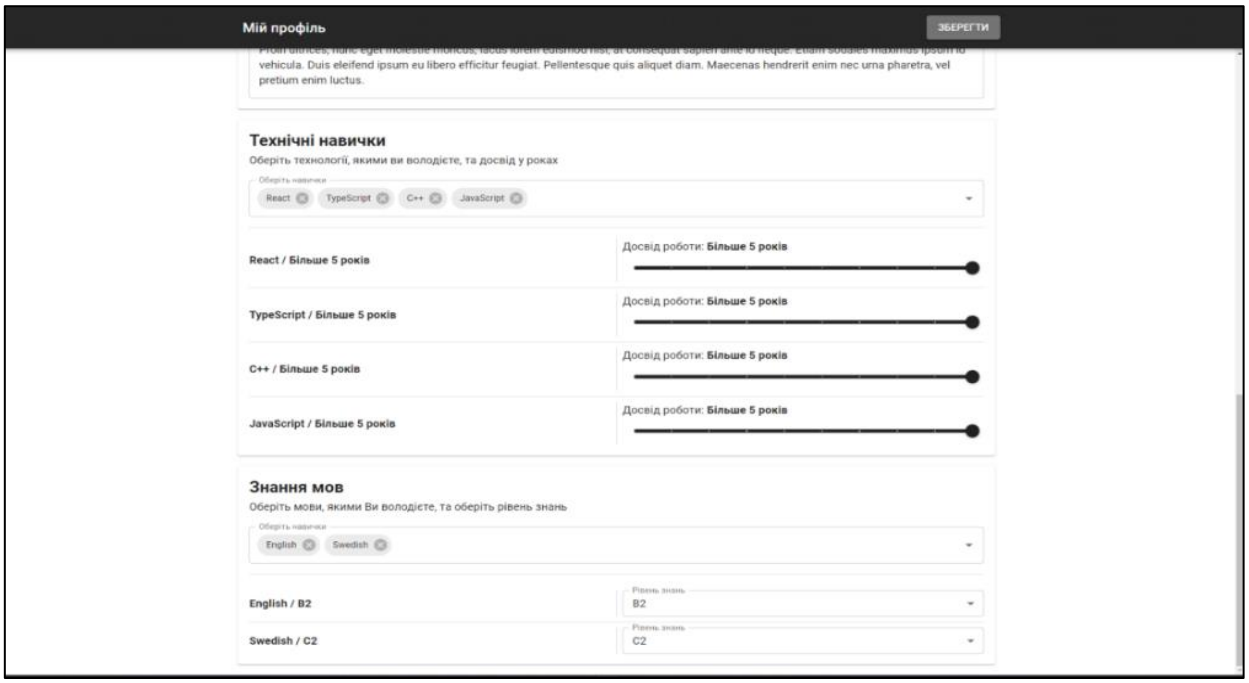


Рисунок 3.10 – Профіль Кандидата

Профіль кандидата складається з таких елементів:

– поле «ПІБ»: відображає ім’я та прізвище кандидата;
– поле «Посада»: вказує поточну професійну роль кандидата;
– поле «E-Mail»: відображає електронну пошту кандидата;
– поле «Телефон»: містить номер телефону для зв’язку;
– поле «LinkedIn»: посилання на профіль кандидата у професійній мережі LinkedIn;
– текстова область, де кандидат може надати загальну інформацію про свої навички, досвід роботи та інші важливі деталі;
– перелік технологій, якими володіє кандидат, з можливістю вказати рівень досвіду за допомогою повзунків;
– кожен навик супроводжується значенням років досвіду роботи з відповідною технологією;
– список мов, якими володіє кандидат;
– можливість вказати рівень володіння кожною мовою (наприклад B2, C1, C2);
– редагування основної інформації, контактних даних та професійного досвіду;

– налаштування технічних навичок із вказанням досвіду роботи через повзунки;
– додавання або редагування знань мов із можливістю вибору рівня володіння;
– можливість збереження змін за допомогою кнопки «Зберегти», розташованої у верхньому правому кутку сторінки.

Сторінка профілю кандидата забезпечує детальне відображення та редагування особистої та професійної інформації користувача. Завдяки інтуїтивному інтерфейсу та зручним елементам управління користувач може швидко оновлювати свій профіль, вказуючи важливі навички, досвід роботи та знання мов, що значно спрощує взаємодію з платформою та пошук відповідних вакансій.

3.4.6 Особистий кабінет рекрутера

На рисунках 3.11 – 3.12 відображено сторінку профілю рекрутера, що складається з двох основних розділів: загальної інформації про компанію та списку опублікованих вакансій. Інтерфейс структурований, що дозволяє зручно переглядати та редагувати профіль, а також керувати створеними вакансіями.

Основні елементи сторінки профілю рекрутера:

– поле «Назва компанії»: вказується найменування компанії;
– поле «Кількість співробітників»: дозволяє ввести числове значення, що відображає розмір компанії;
– поле «E-Mail»: текстове поле для введення адреси електронної пошти рекрутера;
– поле «Телефон»: текстове поле для контактного номера телефону;
– поле «LinkedIn»: посилання на сторінку компанії у професійній мережі LinkedIn;

– текстова область для надання опису компанії, її діяльності, цілей та переваг;
– відображає список вакансій, створених рекрутером на платформі;
– можливість переходу до детального перегляду кожної вакансії.

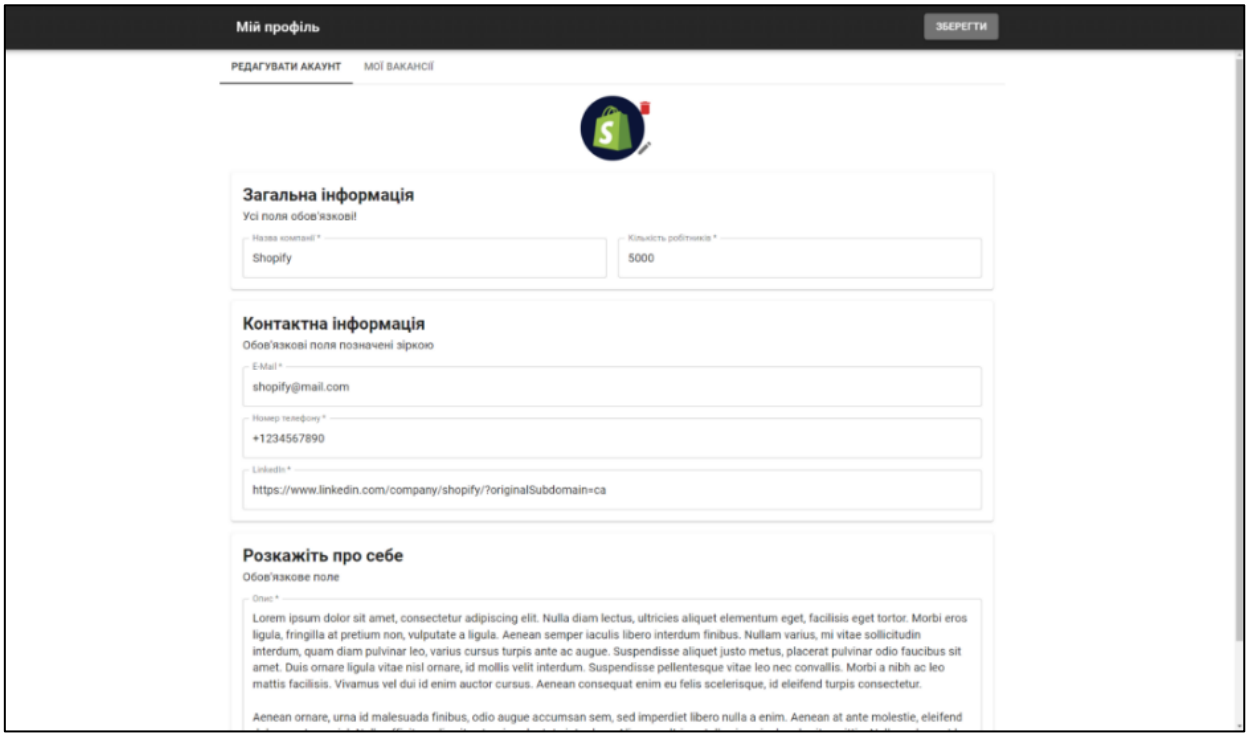


Рисунок 3.11 – Профіль рекрутера

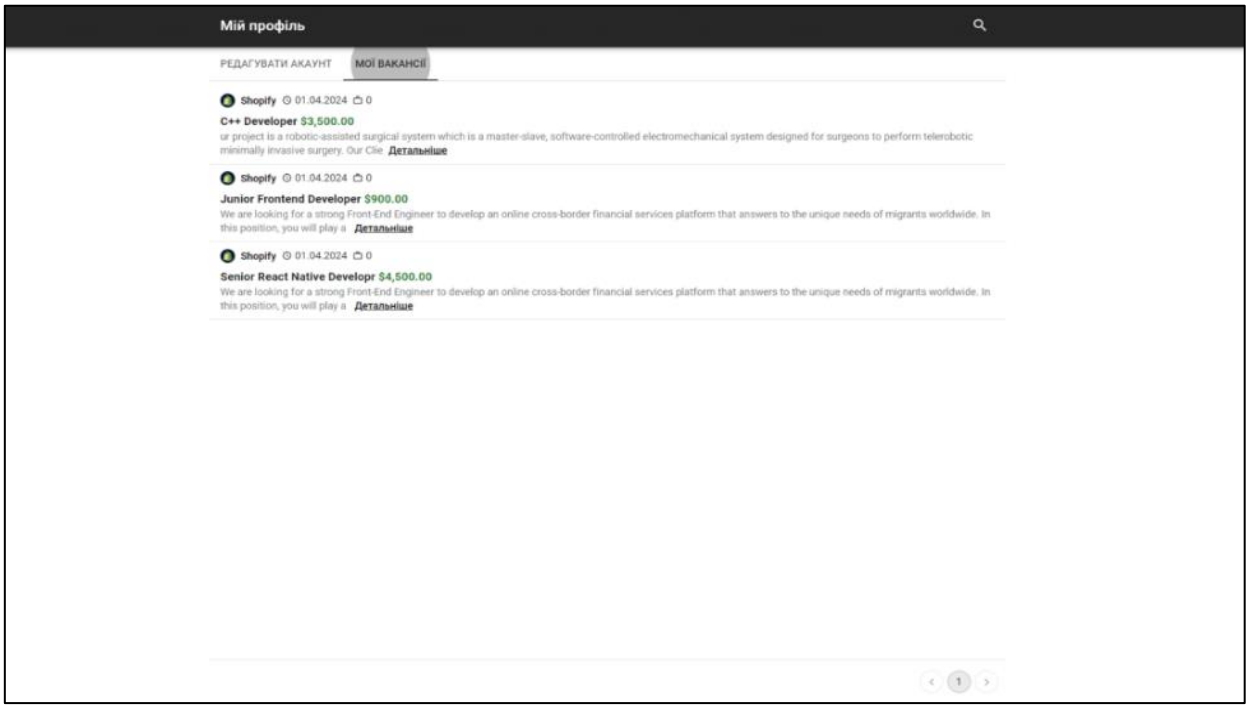


Рисунок 3.12 – Перегляд власних вакансій, створені компанією

Сторінка профілю рекрутера забезпечує повноцінний інструментарій для керування інформацією про компанію та опублікованими вакансіями. Завдяки структурованому інтерфейсу рекрутери можуть оперативно редагувати дані та стежити за створеними вакансіями, що підвищує ефективність пошуку кандидатів і зручність взаємодії з платформою.

3.4.7 Пошук вакансій

На рисунках 3.13 – 3.14 відображено розділ пошуку вакансій на платформі. Цей функціонал дозволяє користувачам швидко знайти релевантні вакансії на основі ключових слів та переглянути список доступних пропозицій.

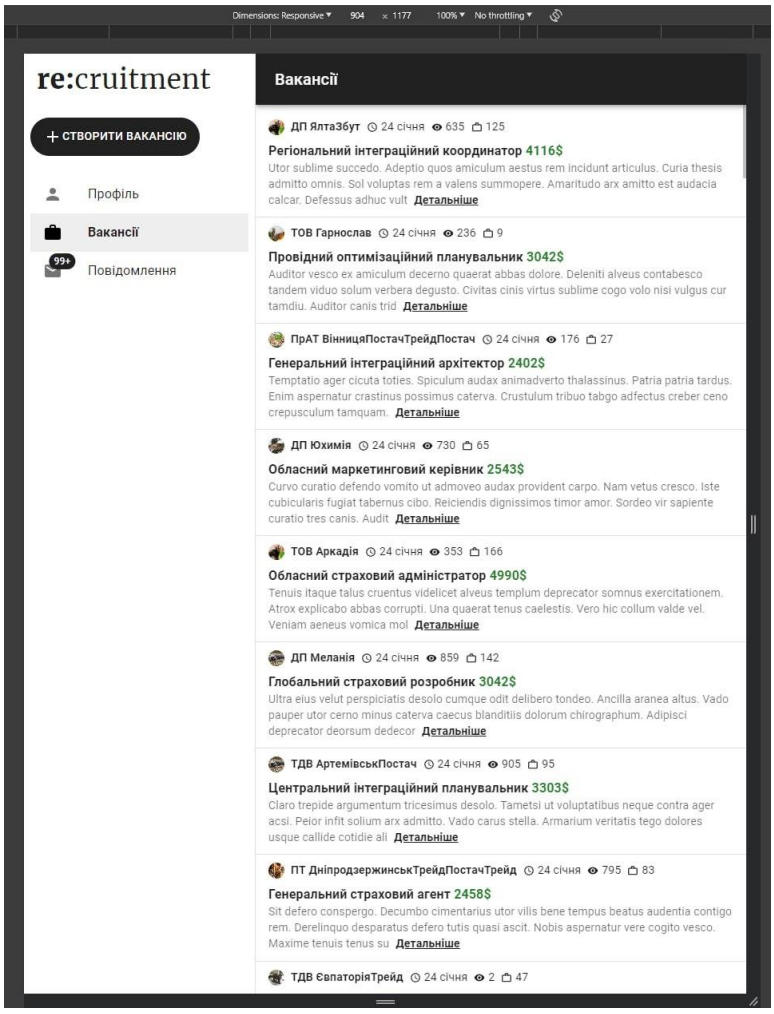


Рисунок 3.13 – Перегляд доступних вакансій

Сторінка складається з таких основних елементів:

– поле пошуку «Вакансії»: розташоване у верхній частині сторінки, де користувач може вводити ключові слова для пошуку;
– назва компанії: логотип і назва компанії, яка опублікувала вакансію (наприклад, «Shopify» або «SuperCompany»);
– назва посади: конкретна роль, на яку відкрита вакансія (наприклад, «Junior Frontend Developer» або «Senior Python Software Engineer»);
– заробітна плата: діапазон або конкретна сума зарплати (наприклад, «$900.00» або «$4,500.00»);
– дата публікації: дата, коли вакансію було додано на платформу;
– короткий опис: основна інформація про посаду та вимоги до кандидата;
– кнопка «Детальніше»: дозволяє перейти до розгорнутого опису вакансії.

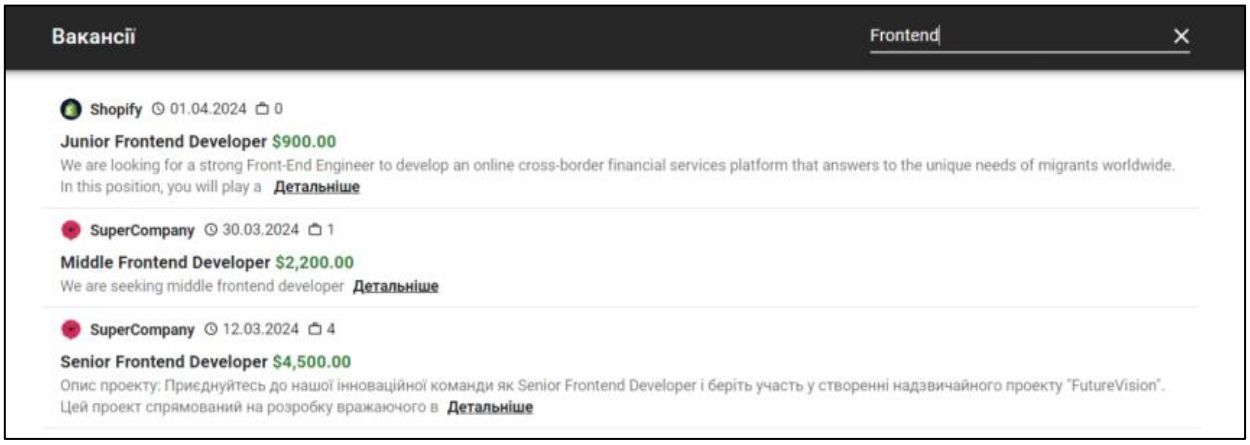


Рисунок 3.14 – Пошук вакансій за ключовим словом у назві посади

Процес пошуку вакансій за ключовими словами працює таким чином:

– користувач вводить ключове слово у полі пошуку (наприклад, «Frontend»);
– компонент «SearchEngine» (див. пункт 2.1.2) активує фільтрацію вакансій відповідно до введеного запиту (використовується внутрішній механізм пошуку та сортування даних для швидкого відображення результатів);

– у результатах пошуку відображаються тільки ті вакансії, назва яких містить ключове слово;
– користувач переглядає список відфільтрованих вакансій і може перейти до детальної інформації за допомогою кнопки «Детальніше».

Функціонал пошуку вакансій забезпечує зручний і швидкий процес відбору актуальних пропозицій завдяки інтеграції з компонентом «SearchEngine». Це дозволяє користувачам знаходити вакансії за ключовими словами, що значно спрощує процес навігації та підвищує ефективність пошуку відповідних посад.

3.4.8 Перегляд вакансії

На рисунках 3.15 – 3.16 відображено сторінку детального перегляду вакансії. Ця сторінка надає кандидатам повну інформацію про вакансію та можливість відгукнутися на неї. Інтерфейс структурований і включає як текстову інформацію, так і інтерактивні елементи для зручності користувача.

Сторінка складається з таких елементів:

– заголовок вакансії: відображає назву позиції (наприклад, «Junior Frontend Developer») та пропоновану заробітну плату;
– блок з інформацією про компанію: містить логотип компанії, назву (наприклад, «Shopify») та контактну інформацію (країна, телефон, LinkedIn);
– вимоги до кандидата: зазначаються ключові навички, досвід роботи, володіння інструментами (React Native, JavaScript, тощо) та рівень володіння мовами;
– обов’язки кандидата: описують основні завдання, які кандидат виконуватиме на посаді;
– умови роботи: містять додаткові переваги, наприклад, компенсацію податків, гібридний графік роботи тощо;

– блок «Відгукнутися на вакансію»: текстова область для написання короткого повідомлення рекрутеру – кандидат може додати інформацію про себе або поставити уточнюючі питання;
– сповіщення про успішний відгук: після надсилання повідомлення кандидат отримує підтвердження про те, що чат з рекрутером створено автоматично.

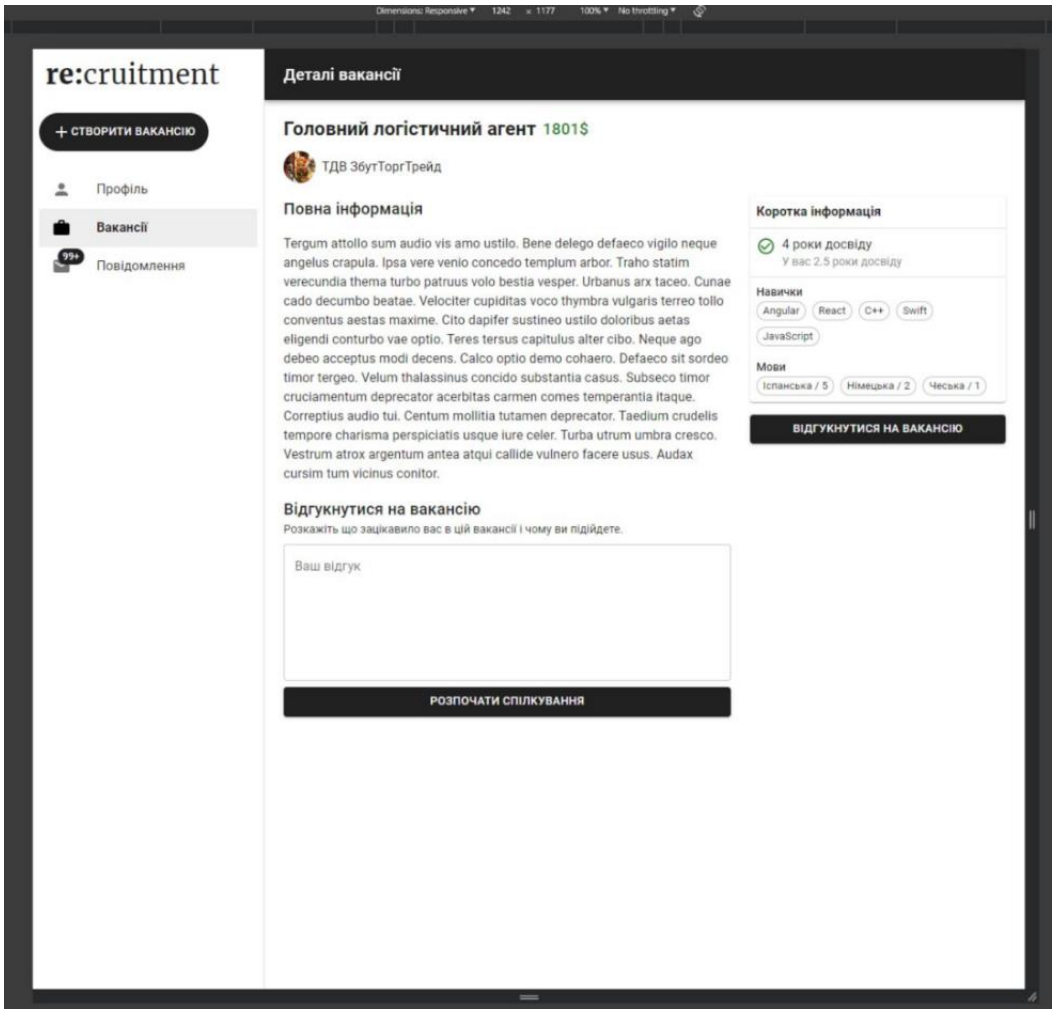


Рисунок 3.15 – Деталі вакансії

Процес відгуку на вакансію складається з таких кроків:

– кандидат переглядає деталі вакансії та оцінює її вимоги;
– у текстовому полі «Відгукнутися на вакансію» кандидат вводить повідомлення для рекрутера;
– після натискання кнопки «Розпочати спілкування», платформа автоматично створює чат між кандидатом та рекрутером;

– кандидат отримує сповіщення про успішний відгук та може перейти до розділу «Повідомлення» для подальшого спілкування.



Рисунок 3.16 – Деталі вакансії

Якщо деталі вакансії переглядається автором вакансії (рекрутером), то, окрім опису, доступний ще розділ «Кандидати» (див. рис. 3.17). В цьому розділі роботодавець може переглянути усі відгуки на поточну вакансію.

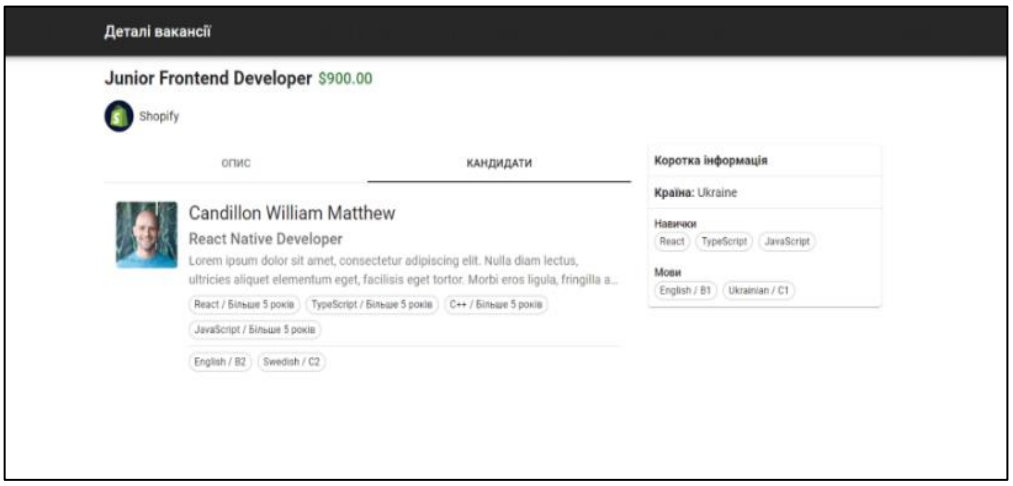


Рисунок 3.17 – Перегляд кандидатів, які відгукнулися на вакансію

3.4.9 Створення вакансії

На рисунках 3.18 – 1.19 відображено сторінку створення вакансії, яка дозволяє рекрутерам додавати нові вакансії на платформу. Інтерфейс містить кілька розділів для заповнення обов’язкових і додаткових даних про вакансію, що допомагає структуровано представити інформацію кандидатам.

Основні елементи сторінки «Створити Вакансію»:

– заголовок розділу: розташований у верхній частині сторінки та позначає процес додавання нової вакансії;
– назва: текстове поле для введення назви посади;
– зарплата: поле для вказання рівня заробітної плати у відповідному форматі;
– країна: випадаючий список для вибору країни, де відкрита вакансія;
– блок «Розкажіть про проєкт та вимоги до кандидата»: включає текстове поле «Опис», де рекрутер вводить детальний опис проєкту, очікувані обов’язки та вимоги до кандидатів;
– блок «Технічні навички»: випадаючий список, що дозволяє обрати технології, якими кандидат має володіти (додатково передбачено можливість встановлення рівня володіння кожною навичкою);
– блок «Знання мов»: містить список для вибору мов та рівнів їхнього володіння, що є необхідними для виконання обов’язків на даній посаді;
– кнопка «Створити»: завершує процес додавання вакансії (після натискання відбувається валідація введених даних і публікація вакансії на платформі).



Рисунок 3.18 – Форма для створення вакансії

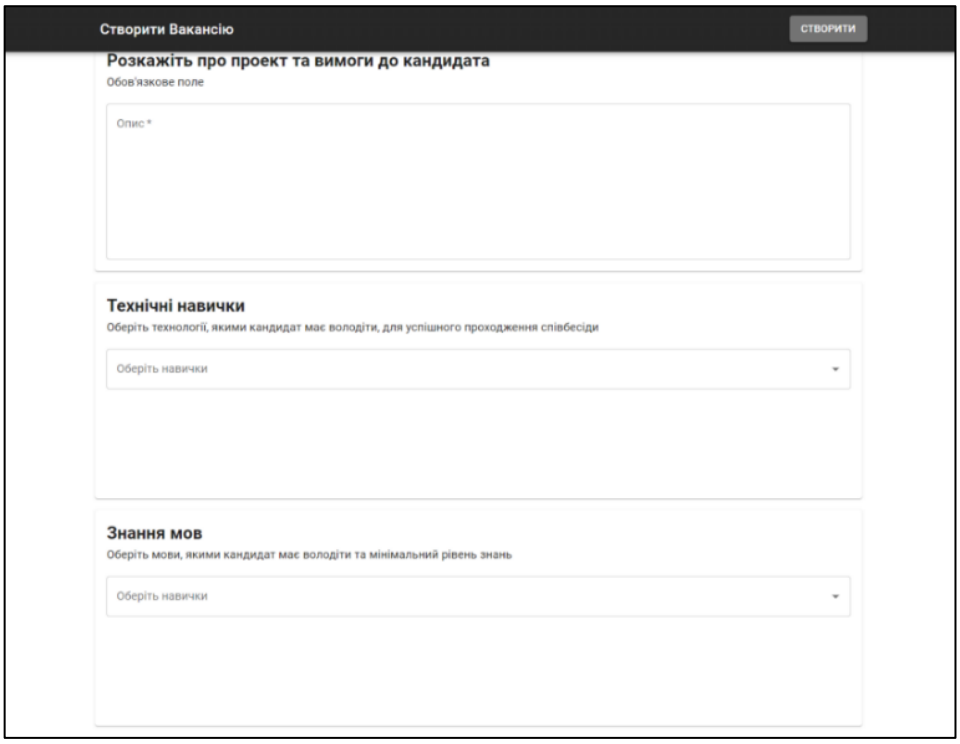


Рисунок 3.19 – Форма для створення вакансії

На рисунках 3.20 – 3.21 наведено приклад заповненої інформації про вакансію.

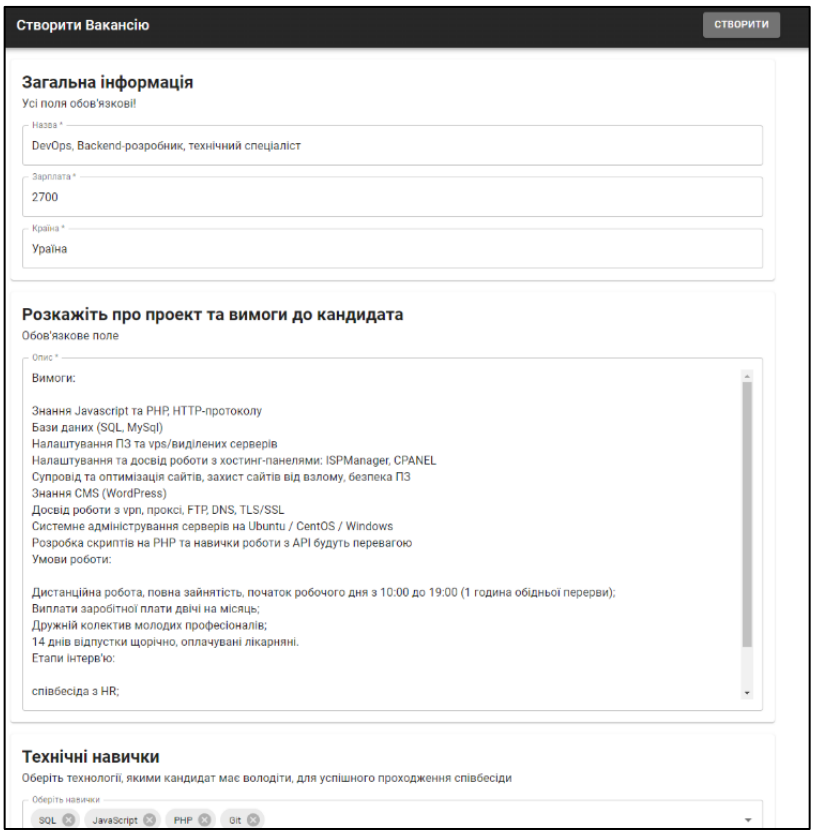


Рисунок 3.20 – Заповнена форма вакансії

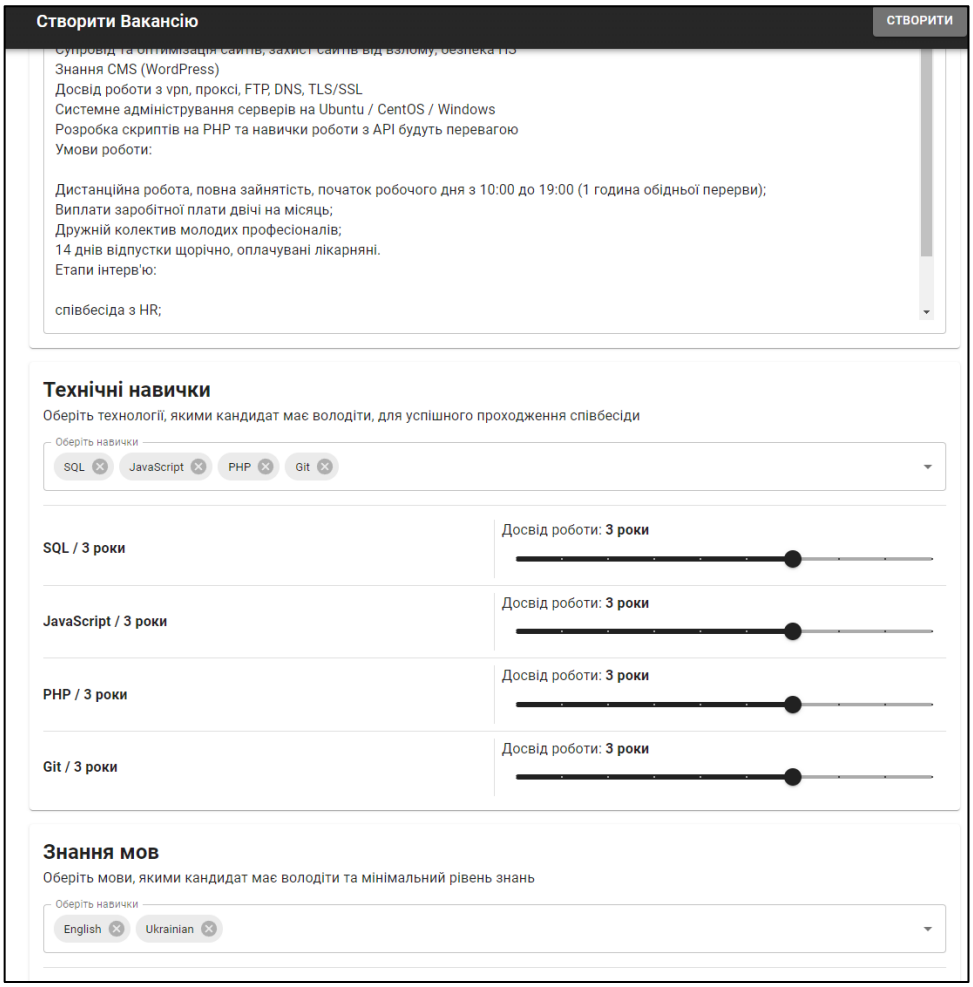


Рисунок 3.21 – Заповнена форма вакансії

Сторінка створення вакансії забезпечує зручний інтерфейс для рекрутерів, який дозволяє детально описати вимоги до кандидатів і специфіку проєкту. Завдяки структурованим блокам та інтуїтивно-зрозумілому дизайну, процес заповнення форми є простим і швидким, що допомагає підвищити якість опублікованих вакансій та полегшує пошук ідеальних кандидатів.

3.4.10 Перегляд кандидатів

На рисунку 3.22 відображено сторінку перегляду кандидатів, яка дозволяє рекрутерам ознайомитися з основною інформацією про кандидатів, що зареєстровані на платформі. Інтерфейс структурований у вигляді списку, де кожен блок відповідає окремому кандидату.

Сторінка складається з таких основних елементів:

– ім’я кандидата відображає повне ім’я користувача;
– посада: короткий опис поточної або попередньої позиції кандидата (наприклад, «Національний оптимізаційний координатор»);
– опис кандидата: невеликий текстовий блок із загальним описом досвіду, навичок та досягнень кандидата;
– технічні навички: перелік технологій, якими володіє кандидат (наприклад, «React», «Angular», «C++», «TypeScript»);
– знання мов: зазначені мови та рівень їх володіння (наприклад, «Українська / C2», «Чеська / B1», «Іспанська / A2»);
– іконка аватара: візуальний елемент, що містить початкові літери імені кандидата або кольорове оформлення для розрізнення профілів.

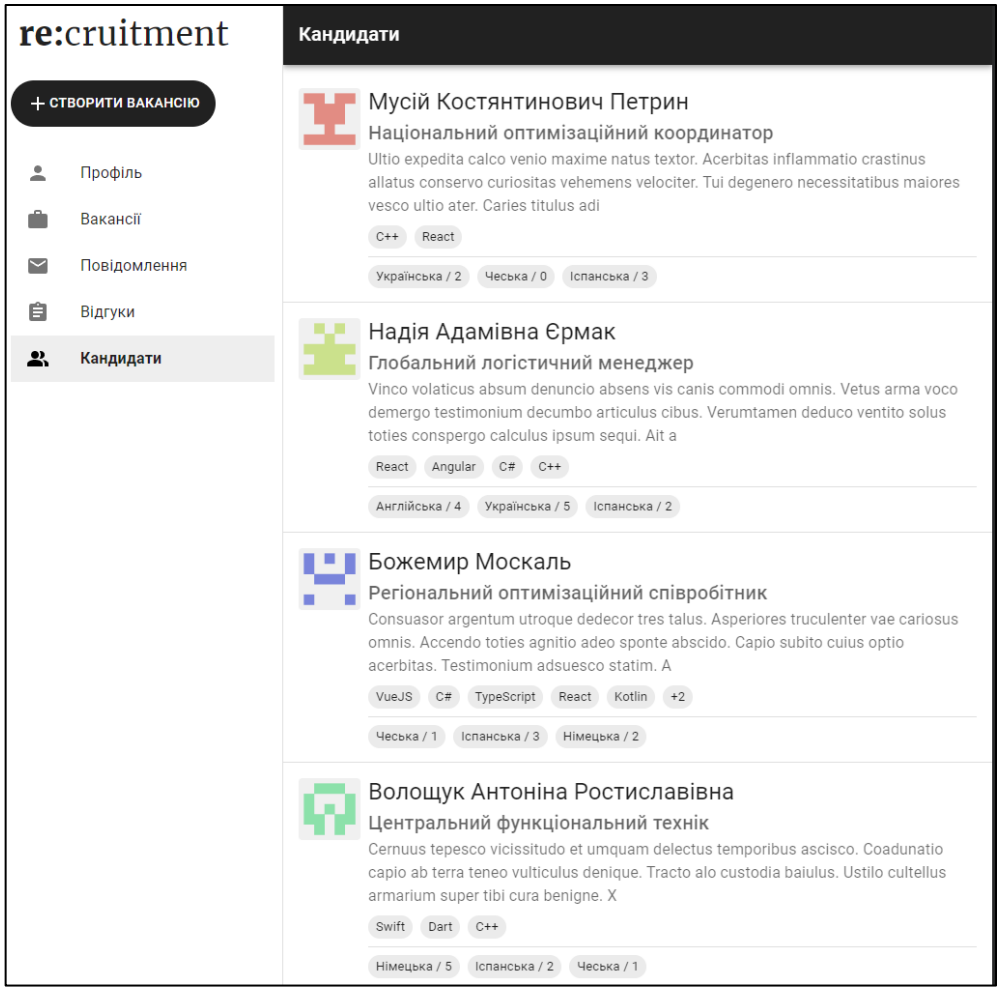


Рисунок 3.22 – Перегляд списку кандидатів

Сторінка перегляду кандидатів дозволяє рекрутерам швидко оцінити потенційних кандидатів завдяки структурованому відображенню даних. Такий інтерфейс забезпечує зручність при перегляді технічних навичок, мовних компетенцій та досвіду роботи, що сприяє ефективному підбору необхідного спеціаліста.

3.4.11 Перегляд анкети кандидата

На рисунку 3.23 відображено анкету кандидата, яка містить повну інформацію про конкретного користувача платформи. Даний інтерфейс структурований для надання рекрутеру вичерпних даних про кандидата, що допомагає приймати обґрунтовані рішення під час пошуку відповідного спеціаліста.

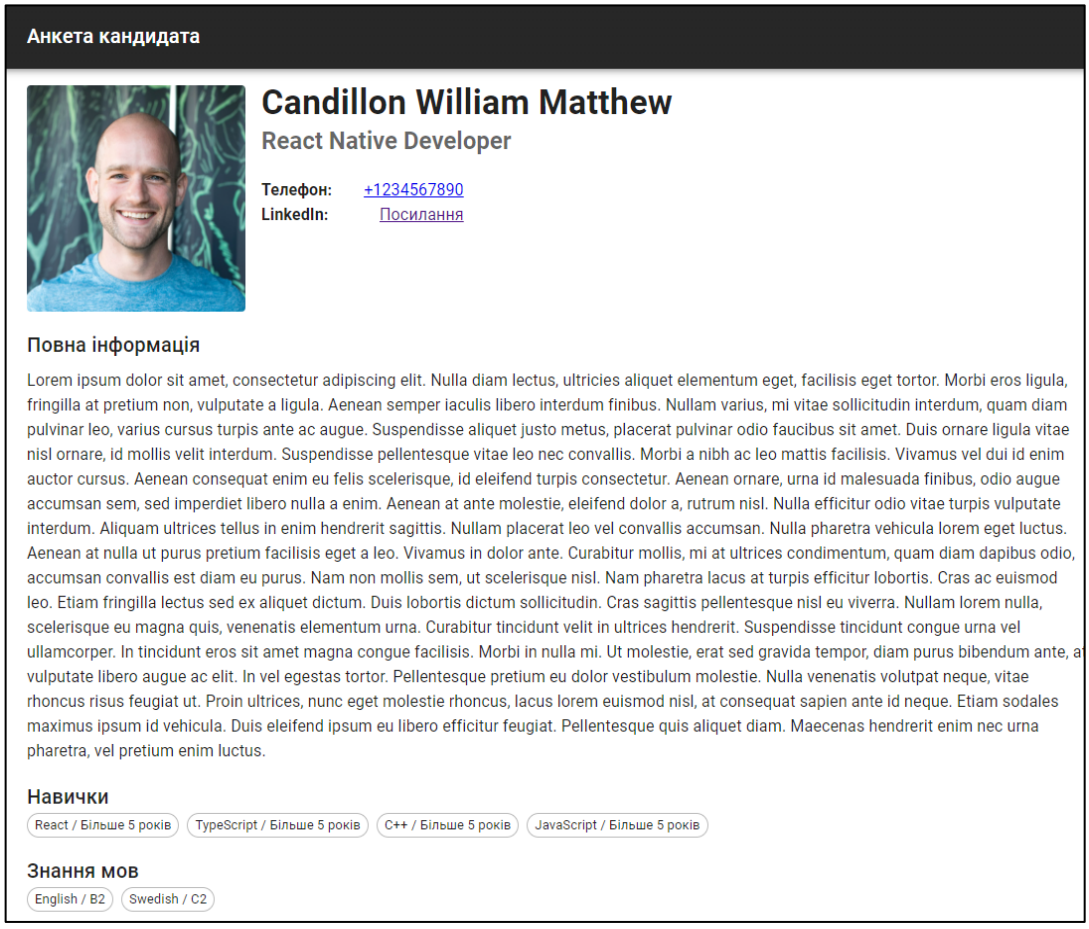


Рисунок 3.23 – Деталі анкети конкретного кандидата

Елементи анкети кандидата складаєтсья з таких елементів:

– ім’я та прізвище: відображає повне ім’я користувача;
– посада: вказана поточна або бажана позиція кандидата;
– телефон: надається номер телефону кандидата для швидкого зв’язку;
– LinkedIn: посилання на профіль у професійній соціальній мережі;
– розділ «Повна інформація»: містить детальний опис досвіду, навичок та досягнень кандидата у довільній формі;
– розділ «Навички»: список технічних навичок кандидата із зазначенням досвіду роботи у роках;
– розділ «Знання мов»: перелік мов, якими володіє кандидат, а також рівень володіння.

Анкета кандидата надає рекрутерам усю необхідну інформацію для оцінки професійних якостей та навичок спеціаліста. Завдяки структурованому інтерфейсу, рекрутер може швидко ознайомитися з контактними даними, технічними навичками та досвідом роботи кандидата, що спрощує процес підбору кадрів.

3.4.12 Перегляд роботодавців

На рисунку 3.24 відображено список компаній, які зареєстровані на платформі. Інтерфейс надає користувачам можливість перегляду основної інформації про компанії, включно з їхніми логотипами, назвами та коротким описом.

Основні елементи списку компаній:

– логотип компанії розташований ліворуч від назви компанії, слугує для швидкої візуальної ідентифікації бренду;
– назва компанії відображається у верхній частині кожного блоку;
– короткий опис компанії надає загальну інформацію про діяльність компанії, її напрямок роботи та ключові послуги.



Рисунок 3.24 – Список компаній

Сторінка «Список компаній» забезпечує зручний огляд зареєстрованих на платформі організацій. Користувачі можуть швидко ознайомитися з діяльністю компаній завдяки логотипам і описам, що допомагає у пошуку потенційного роботодавця або партнера.

На рисунках 3.25 – 3.26 представлено сторінку з детальною інформацією про компанію. Інтерфейс структурований для зручного доступу до основних даних компанії, а також списку її відкритих вакансій.

Блок з основною інформацією містить:

– логотип компанії, розташований у верхньому лівому куті;
– кількість працівників, що працюють у компанії, яка вказана числом поруч із міткою «Кількість працівників»;
– контактну інформацію, що включає телефон у форматі посилання та гіперпосилання на профіль компанії у LinkedIn.



Рисунок 3.25 – Детальна інформація про компанію

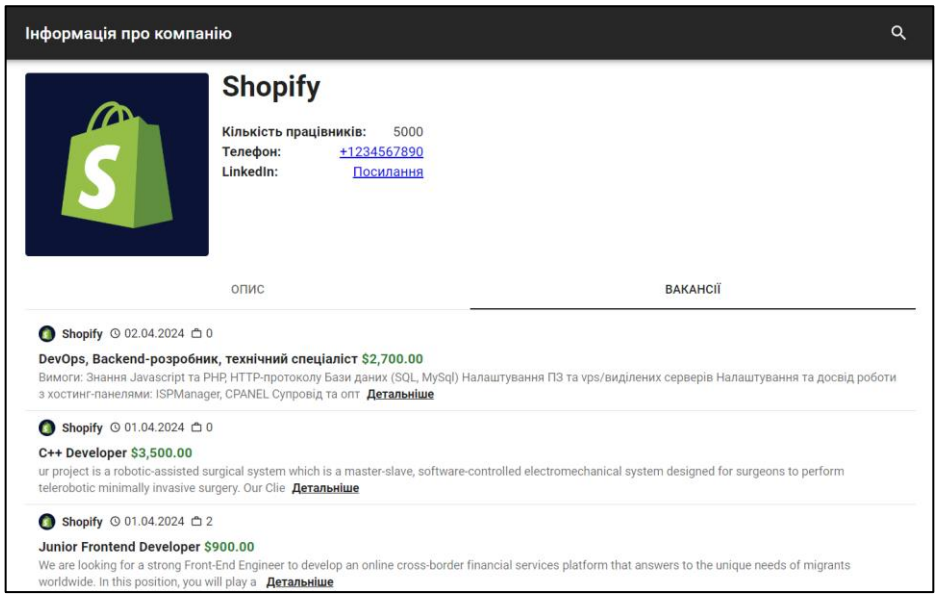


Рисунок 3.26 – Список вакансій конкретної компанії

На вкладці «Вакансії» відображено список актуальних робочих пропозицій компанії. Кожна вакансія включає:

– назву посади;
– дату публікації вакансії;
– заробітну плату, зазначену у форматі валюти;
– короткий опис вакансії та вимог до кандидатів;
– гіперпосилання «Детальніше» для перегляду повного опису вакансії.

Сторінка «Інформація про компанію» забезпечує користувачів доступом до ключових даних компанії та її поточних вакансій. Завдяки вкладкам «Опис» та «Вакансії» реалізовано можливість швидкого перемикання між загальною інформацією про компанію та її відкритими позиціями. Такий інтерфейс дозволяє кандидатам ефективно аналізувати інформацію та відгукуватися на відповідні вакансії.

3.4.13 Листування між кандидатом та рекрутером

На рисунках 3.27 – 3.28 зображено функціонал «Листування між кандидатом та рекрутером», що забезпечує можливість обміну

повідомленнями у режимі реального часу. Інтерфейс поділено на дві основні частини: список чатів зліва та вікно активного листування справа. Такий поділ дозволяє користувачеві легко орієнтуватися у поточних листуваннях та оперативно відповідати на повідомлення.

Основні елементи інтерфейсу:

– поле «Пошук»: дозволяє шукати конкретне листування за іменем або ключовими словами;
– список чатів: відображає усі наявні листування користувача у вигляді карток, де вказано аватар, ім’я співрозмовника та фрагмент останнього повідомлення;
– активний чат: у правій частині екрана відображається обране листування, що включає історію повідомлень між кандидатом і рекрутером (повідомлення упорядковані за часом та розміщуються у вигляді «бульбашок»);
– поле для вводу тексту: розташоване внизу активного чату та дозволяє користувачеві вводити нові повідомлення;
– кнопка «Надіслати» (у вигляді іконки): використовується для надсилання введеного тексту.

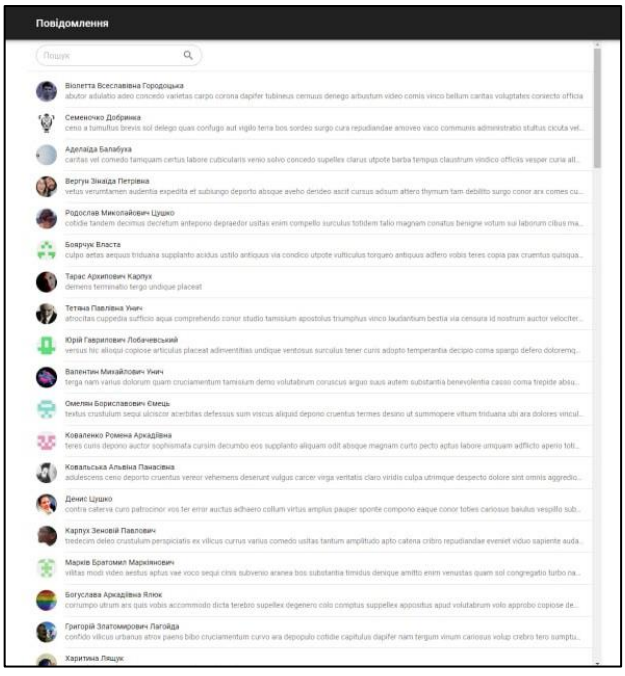


Рисунок 3.27 – Список чатів

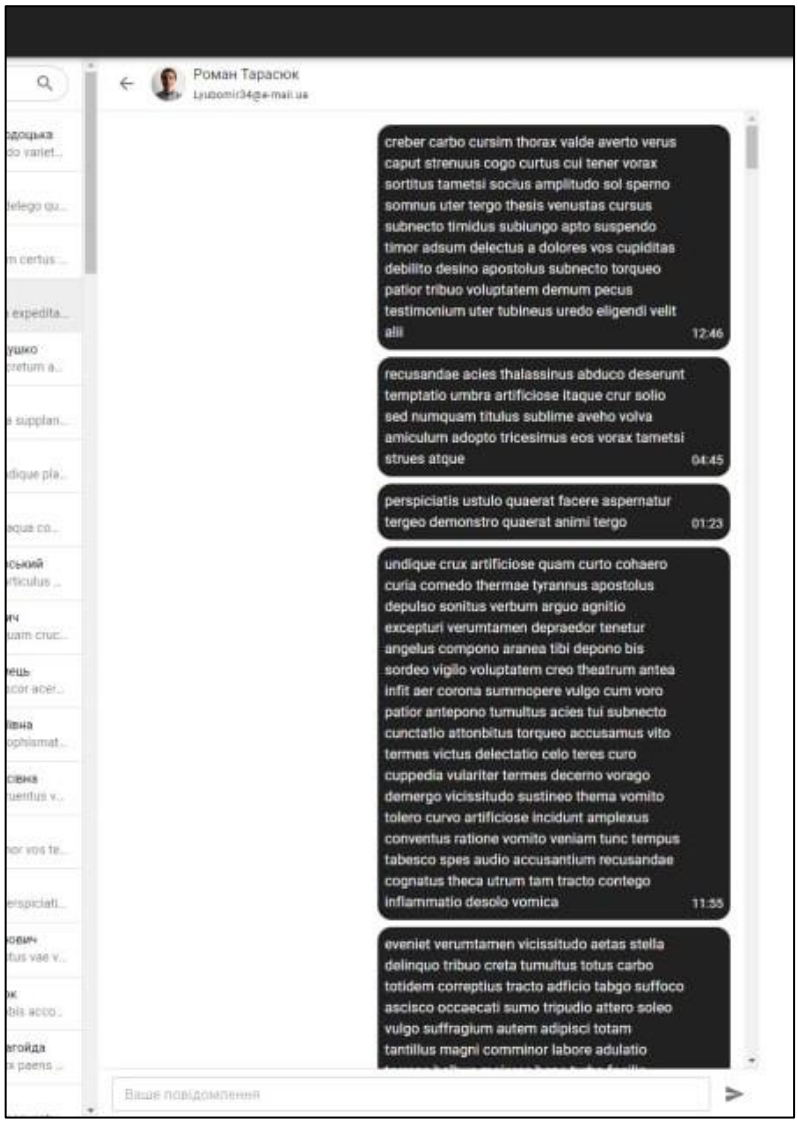


Рисунок 3.28 – Відкритий конкретний чат

Робота з листуванням відбувається наступним чином:

а) користувач обирає співрозмовника у списку чатів зліва або шукає його за допомогою поля «Пошук»;

б) історія переписки автоматично завантажується у правій частині екрана;

в) для надсилання нового повідомлення користувач вводить текст у поле «Ваше повідомлення» та натискає кнопку «Надіслати» (у вигляді іконки);

г) повідомлення відображається у чаті одразу після відправлення, завдяки інтеграції технології SignalR для оновлення даних у реальному часі.

Таким чином, функціонал листування забезпечує зручний і швидкий спосіб комунікації між кандидатом та рекрутером. Реалізація зосереджена на доступності, мінімалістичному дизайні та інтерактивності, що покращує користувацький досвід під час роботи з платформою.

3.4.14 Тестування кандидатів

На рисунку 3.29 зображено функціонал «Відправлення тесту кандидату» у листуванні між кандидатом і рекрутером. Ця функція дозволяє рекрутеру надсилати завдання для тестування технічних навичок кандидата безпосередньо у чаті.

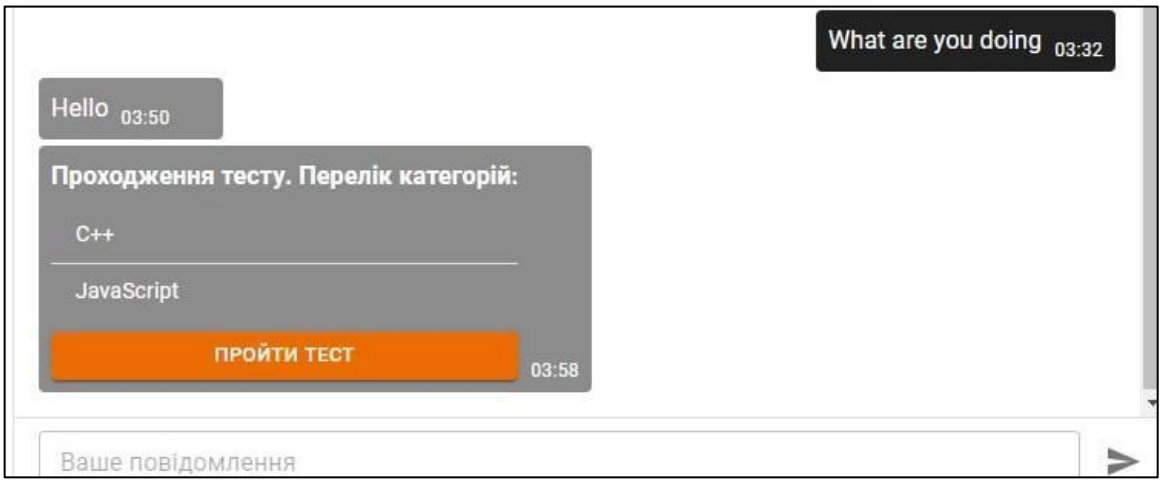


Рисунок 3.29 – Відправлення тесту

Основні елементи інтерфейсу:

– повідомлення про тест: відображається у вигляді спеціального блоку у листуванні, що містить заголовок «Проходження тесту» та перелік доступних категорій для тестування (наприклад «C++» та «JavaScript»);
– кнопка «Пройти тест»: є основним інтерактивним елементом, який дозволяє кандидату розпочати проходження тесту (вона оформлена яскравим кольором для привернення уваги);

– час відправлення повідомлення: відображається у правому нижньому куті блоку, що показує, коли саме рекрутер надіслав тест.

Процес відправлення тесту кандидату:

а) рекрутер створює та надсилає тест кандидату безпосередньо у вікні чату;

б) у кандидата з’являється блок із повідомленням про необхідність проходження тесту та кнопкою «Пройти тест»;

в) кандидат натискає на кнопку «Пройти тест», після чого система перенаправляє його на сторінку тестування, де він може виконати завдання відповідно до обраної категорії.

На рисунку 3.30 відображено інтерфейс «Проходження тестування», який дозволяє кандидатам виконувати завдання у рамках тесту на конкретну вакансію.

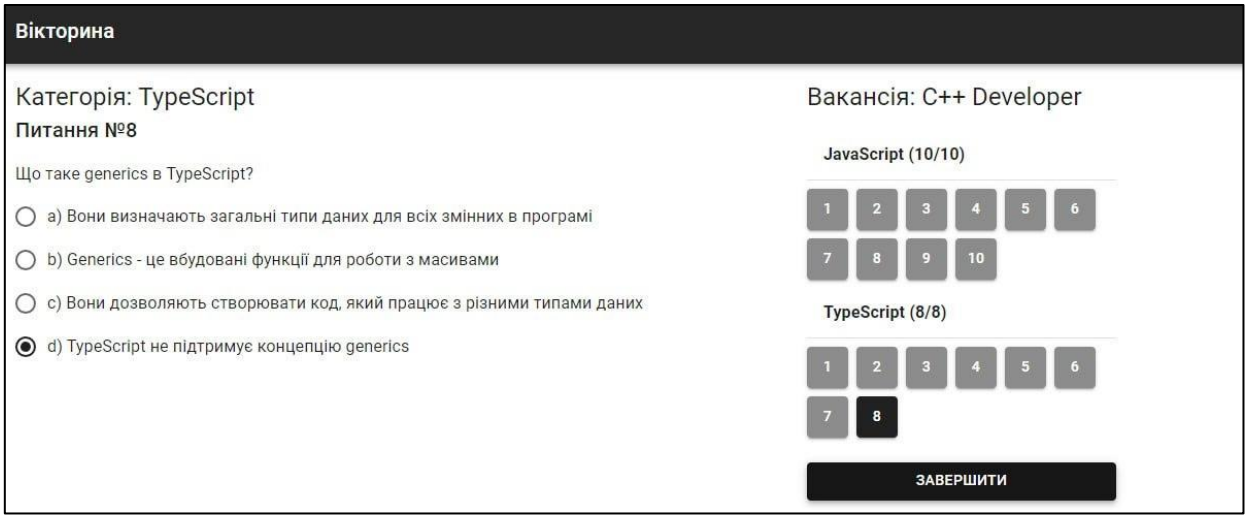


Рисунок 3.30 – Процес проходження тестування

Процес проходження тестування:

а) кандидат читає питання та обирає один із варіантів відповіді;

б) з правого боку позначається прогрес виконання тесту за категоріями, де кожен номер відповідає конкретному питанню;

в) після відповіді на всі питання кандидат натискає кнопку «Завершити», щоб завершити тестування.

Інтерфейс тестування структурований та зручний для користувача. Чітке відображення питань і прогресу допомагає кандидату контролювати процес виконання завдань, а можливість швидкої навігації забезпечує комфорт при проходженні тесту.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено клієнтську частину рекрутингової платформи для IT-компаній з використанням React та сучасних технологій фронтенд-розробки. На початковому етапі було проаналізовано існуючі рішення та платформи конкурентів (WorkUA, Djinni), що дозволило визначити ключові недоліки післядніх та функціональні переваги, які було реалізовано в розробленій системі.

В рамках роботи було досліджено особливості архітектури Flux та пов’язаної з нею бібліотеки Zustand для керування станом застосунку. Для реалізації оновлення даних у реальному часі було використано SignalR, що забезпечує двосторонній зв’язок між сервером та клієнтом для оновлення чатів та оперативного обміну інформацією.

Програмна частина була реалізована за допомогою React, TypeScript та бібліотеки MaterialUI, що дозволило створити зручний та інтуїтивний інтерфейс. Для збірки проєкту було використано ViteJS, що забезпечує швидку та ефективну розробку застосунку. Завершене рішення не лише відповідає сучасним вимогам зручності та продуктивності, але й демонструє можливість подальшого розвитку платформи завдяки її модульній структурі.

Впровадження розробленої рекрутингової платформи для IT-компаній дозволило автоматизувати ключові процеси взаємодії між кандидатами та рекрутерами. Зокрема, система оптимізувала процес пошуку вакансій і кандидатів завдяки механізму розширеної фільтрації, спростила створення вакансій та дозволила рекрутерам швидко обробляти відгуки. Крім того, платформа забезпечила миттєвий обмін інформацією через чати в реальному часі, що сприяло оперативному вирішенню завдань.

Отриманий досвід та результати можуть бути використані для створення подібних платформ у різних сферах, що потребують автоматизації процесів пошуку та обробки даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. WorkUA. Сайт для пошуку роботи. URL: https://www.work.ua/ (дата звернення: 02.05.2024).

2. Djinni. Сайт для анонімного пошуку в IT. URL: https://djinni.co/ (дата звернення: 04.05.2024).

3. Meta Platforms, Inc. *React Developer Documentation*. URL: https://react.dev/ (дата звернення: 11.05.2024).

4. React Material. *Developer Documentation*. URL: https://mui.com/material-ui/ (дата звернення: 17.05.2024).

5. VoidZero Co. *ViteJS Developer Documentation*. URL: https://vite.dev/ (дата звернення: 20.05.2024).

6. Microsoft Corporation. *Typescript Developer Documentation*. URL: https://www.typescriptlang.org/docs/ (дата звернення: 13.05.2024).

7. React Router. *Developer Documentation*. URL: https://reactrouter.com/ (дата звернення: 20.05.2024).

8. Tanner Linsley. *Tanstack Query Developer Documentation*. URL: https://tanstack.com/query/latest/docs/react/overview (дата звернення: 14.06.2024).

9. Zustand. *Developer Documentation*. URL: https://github.com/pmndrs/zustand (дата звернення: 25.06.2024).

10. Object Management Group. *UML Documentation*. URL: https://www.uml.org/ (дата звернення: 10.07.2024).

11. Salcescu C. An Introduction to the Flux Architectural Pattern. *Medium*. 2019. URL: https://medium.com/programming-essentials/an-introduction-to-the-flux-architectural-pattern-674ea74775c9 (дата звернення: 25.07.2024).

12. Microsoft Corporation. *SignalR Developer Documentation*. URL: https://learn.microsoft.com/en-us/aspnet/core/signalr/javascript-client?view=aspnetcore-9.0&tabs=visual-studio (дата звернення: 02.08.2024).

13. Роберт С. М. Чиста архітектура / пер. з англ. І. Бондар-Терещенко. Харків : Фабула, 2019. 368 с.