

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

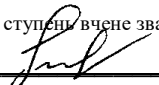
на тему: Розробка веб-сервісу пошуку роботи «Fast-Work»

Виконав: студент групи 3157ст

 _____ Чопенко М.С.
(підпис, ПІБ)

Керівник роботи:

доцент кафедри ПЗАС, к.т.н., доцент

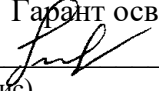
(посада, науковий ступінь/вчене звання)
 _____ Макарова Л.М.
(підпис, ПІБ)

Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

 (підпис) доц. Макарова Л.М.
 «__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Чопенко Микиті Станіславовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка веб-сервісу пошуку роботи «Fast-Work»

Керівник роботи: Макарова Л.М.

Затверджені наказом ректора №257-уч від «17» березня 2023 р.

2. Термін подання роботи: 29.05.2023 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний аркуш; 2. Аналіз існуючих аналогів веб-сервісів пошуку роботи; 3. Аналіз методів та інструментів розробки веб-сервісів; 4. Модель варіантів використання; 5. Архітектура програмного забезпечення; 6. Діаграма компонентів; 7. Діаграма розміщення; 8. Діаграма діяльності; 9. Інформаційна модель; 10. Діаграма послідовності; 11. Логічна модель даних; 12. Фізична модель бази даних; 13. Результати розробки веб-сервісу; 14. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 20.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	24.03.2023	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	31.03.2023	ПП*
3.	Аналіз вимог до ПЗ	07.04.2023	ПП*
4.	Розробка архітектури ПЗ	21.04.2023	
5.	Розробка проекту ПЗ	05.05.2023	
6.	Реалізація та тестування ПЗ	12.05.2023	
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023	
8.	Підготовка розділу з охорони праці	19.05.2023	ПП*
9.	Підготовка розділу ВИСНОВКИ	24.05.2023	
10.	Оформлення списку використаних джерел та додатків	26.05.2023	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	29.05.2023	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	02.06.2023	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	05.06.2023	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	12.06.2023	

* - за результатами переддипломної практики (ПП), яка була з 01.02.2023 до 19.03.2023 р.

Студент

(підпис)

Чопенко М.С.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Дана кваліфікаційна робота присвячена розробці веб-сервісу пошуку роботи для користувачів в Україні. Метою роботи є створення ефективного та зручного програмного забезпечення для пошуку вакансій, що допоможе зменшити час та зусилля, які будуть витрачені при пошуку роботи.

Проведено аналіз практичної задачі інженерії програмного забезпечення з розробки веб-сервісу пошуку роботи «Fast-Work». Було розглянуто архітектурні аспекти роботи веб-сервісу та вибір технологій для реалізації. Було розроблено проект веб-сервісу, а також реалізований веб-сервіс пошуку роботи.

Отриманні результати дозволять рекомендувати розроблений веб-сервіс як ефективний інструмент пошуку роботи в Україні. Веб-сервіс допоможе збільшити ефективність процесу пошуку роботи та сприяє взаємовигідним зв'язкам на ринку праці.

Кваліфікаційна робота викладена на 102 сторінках друкованого тексту, містить 39 рисунків, 11 таблиць, список використаних джерел із 11 найменуваннями та 5 додатків.

ANNOTATION

This qualification work is dedicated to the development of a web-based job search service for users in Ukraine. The goal of the work is to create efficient and convenient software for job search that will help reduce the time and effort spent in finding employment.

An analysis of the practical task of software engineering for the development of the "Fast-Work" job search web service was conducted. The architectural aspects of the web service and the selection of technologies for implementation were considered. A web service project was developed, and the job search web service was implemented.

The obtained results will recommend the developed web service as an effective job search tool in Ukraine. The web service will help increase the efficiency of the job search process and contribute to mutually beneficial connections in the labor market.

The diploma thesis is presented in 102 printed text format, including 39 drawings, 11 tables, a list of sources used of 11 items, and 5 appendices.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST-WORK»	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення з розробки веб-сервісів пошуку роботи.....	10
1.2 Аналіз існуючих аналогів веб-сервісів пошуку роботи.....	12
1.3 Аналіз методів та інструментів розробки веб-сервісів	19
1.4 Постановка задачі на розробку веб-сервісу пошуку роботи «Fast-Work».....	21
2 РОЗРОБКА ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST-WORK»	24
2.1 Аналіз вимог до програмного забезпечення.....	24
2.1.1 Побудова моделі варіантів використання	25
2.1.2 Специфікація варіантів використання	27
2.2 Архітектура програмного забезпечення	30
2.2.1 Розробка діаграми компонентів.....	31
2.2.2 Розробка діаграми розміщення.....	32
2.3 Проект програмного забезпечення.....	33
2.3.1 Ескізний проект програмного забезпечення.....	33
2.3.2 Діаграма діяльності	34
2.3.3 Інформаційна модель	38
2.3.4 Розробка ескізу інтерфейсу	38
2.3.5 Технічний проект програмного забезпечення	41
2.3.6 Діаграма послідовності	45

2.3.7 Логічна модель даних	47
2.3.8 Робочий проект програмного забезпечення	48
2.3.9 Вибір та обґрунтування інструментів розробки.	48
2.3.10 Фізична модель бази даних.....	49
2.3.11 Реалізація програмного забезпечення	52
2.3.12 Тестування програмного забезпечення	53
3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST- WORK».....	60
4 ОХОРОНА ПРАЦІ	65
4.1 Основні положення	65
4.2 Аналіз шкідливих та небезпечних факторів на робочому місці	66
4.3 Заходи щодо зменшення впливу шкідливих факторів роботи за ПК	67
ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ.....	71
ДОДАТОК Б КОД ПРОГРАМИ.....	76
ДОДАТОК В ОПИС ПРОГРАМИ.....	87
ДОДАТОК Г ІНСТРУКЦІЯ КОРИСТУВАЧА.....	90
ДОДАТОК Д ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ.....	101

ВСТУП

У сучасному світі пошук роботи є дуже складним та часто тривожним процесом для багатьох людей. Раніше, до появи інтернету та веб-сайтів пошуку роботи, люди зазвичай шукали потрібні вакансії у різноманітних джерелах інформації. Люди дізнавалися про доступну роботу з журналів та газет, ефективним також були рекомендації від знайомих та рідних, інші ж використовували прямі дзвінки роботодавцям. Пізніше почали створювати агентства з працевлаштування, які допомагали знаходити контакти з роботодавцями та пропонували вакансії, які відповідали кваліфікаціям та потребам людей. Всі ці способи вимагали витрачення багато часу та зусиль.

Інтернет зробив це більш доступним та значно простішим. Для вирішення проблем з працевлаштуванням було створено багато веб-сайтів пошуку вакансій. Вони допомагають користувачам знайти відповідну роботу за різними критеріями, такими як галузь, місто, регіон, тип зайнятості, мінімальної та максимальної заробітної плати. Додатки стали популярними через те, що пошук можна робити у будь якому місці та в будь який час.

Вони є безкоштовними для користувачів та можуть мати різний рівень деталізації інформації про вакансії до повного циклу пошуку роботи, включаючи підготовку до співбесід та підтримку після працевлаштування.

Загалом вебсайти складаються з інформації про доступні вакансії у певному регіоні, включаючи компанії та роботодавців. За час появи інтернету було вже створено багато видів програмного забезпечення, але всі вони мати свої певні недоліки. До них відноситься низька якість вакансій, які не оновлюються або опубліковані сумнівними роботодавцями, що може викликати розчарування користувачів. Деякі мають складний та незрозумілий інтерфейс.

Мета кваліфікаційної роботи полягає в тому, що необхідно вирішити задачу інженерії програмного забезпечення з проектування веб сервісу пошуку роботи, який значно полегшить людям працевлаштуватися за допомогою розробленого

програмного забезпечення.

Відмінність сервісу пошуку роботи полягає в розробці простого та багато функціонального інтерфейсу.

Для того щоб розробити програмне забезпечення необхідно вирішити наступні завдання:

- 1) Провести аналіз практичної задачі інженерії програмного забезпечення з розробки веб-сервісів пошуку роботи;
- 2) Провести аналіз існуючих веб сайтів пошуку роботи;
- 3) Провести аналіз інструментів розробки веб сервісів;
- 4) Провести постановку задачі на розробку веб-сервісу пошуку роботи «Fast-Work»;
- 5) Провести аналіз вимог до програмного забезпечення;
- 6) Спроекувати архітектуру програмного забезпечення;
- 7) Розробити користувальницький інтерфейс (frontend);
- 8) Розробити функціонал (backend);
- 9) Протестувати розроблене програмне забезпечення.

Об'єктом роботи є процес розробки програмного забезпечення веб-сервісу пошуку роботи «Fast-Work».

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST-WORK»

1.1 Аналіз практичної задачі інженерії програмного забезпечення з розробки веб-сервісів пошуку роботи

Однією з практичних задач інженерії програмного забезпечення є розробка веб-сервісів. Веб-сервіс можна позначити як програмний модуль, призначений для виконання певного набору завдань. Структурні елементи веб-сервісу знаходяться в мережі і можуть бути викликані відповідним чином. При виклику веб-сервіс зможе надавати функціональні можливості клієнту [1]. Одним із важливих сервісів є веб-сервіс пошуку роботи.

Проблема пошуку роботи є актуальною для багатьох людей, особливо для тих, хто тільки починає свою кар'єру або хто шукає нові можливості розвитку.

Одним із головним нюансом є велика кількість людей на ринку праці, через це підвищується конкуренція, необхідно мати багато знань для успішного проходження співбесіди та працевлаштування. Для того щоб легко знайти вакансію необхідно ефективно скласти резюме, яке буде описувати повністю ваші можливості.

Резюме – це короткий документ, який містить інформацію про досвід роботи, навички, освіту та інші відомості, що можуть бути важливими для роботодавця при відборі кандидатів на роботу.

Інформаційні технології розвиваються з кожним днем, щоденно у мережі інтернет з'являється багато програм, веб-сервіси. Для людей, які шукають працевлаштування і були створені сервіси пошуку роботи.

Сервіс пошуку роботи – це онлайн-сервіс, який допомагає користувачам знайти відповідні вакансії та подати заявку на роботу. Зазвичай ці сервіси працюють як інтернет-платформи, на яких роботодавці можуть розміщувати

оголошення про вакансії, а потенційні працівники можуть створювати свої профілі.

Практична задача веб-сервісу пошуку роботи «Fast-Work» полягає в зменшенні часу та зусиль, витрачених на пошук роботи.

Вакансія – це посада, яка відкрита для зайняття у певній компанії або організації. Вона може містити в собі інформацію про потрібні навички, вимоги до кандидата, відповідальність та умови праці, зокрема заробітну плату та графік роботи.

Кандидати, які відповідають вимогам посади, зазвичай надсилають свої резюме, а вже після успішного перегляду їх запрошують для співбесіди. На якій зазвичай запитують різноманітну інформацію: досвід роботи, скільки часу кандидат працював на певній посаді, навички, які зможуть стати у нагоді, комунікативність, як добре кандидат спілкується з колегами по роботі, мотивація, чому саме він повинен працевлаштуватися, доступність, чи зможе працюватися у певний час, або мати динамічний графік, очікування, яку заробітну плату хоче отримувати.

Веб-сервіси можуть надавати користувачам інструменти для полегшення процесу пошуку роботи такі як фільтри за ключовими словами, місцем розташування, заробітною платою тощо. Крім того, вони можуть розміщати корисну інформацію про ринок праці, зокрема, про потрібність на ринку роботодавців, середню заробітну зарплатню на різних посадах, особливості робочих місць тощо.

Основні переваги веб додатків – це можливість створювати та редагувати резюме, швидко та зручно мати доступ до великої кількості вакансій, можливість отримання нотифікацій при появі нової актуальної інформації.

Отже, ця предметна область є корисною в наш час, тому було вирішено розробити програмне забезпечення веб сервісу пошуку роботи «Fast-Work», щоб автоматизувати процес створення резюме для працівників, створення резюме для роботодавців, пошуку вакансій та ефективного управління ними.

1.2 Аналіз існуючих аналогів веб-сервісів пошуку роботи

Розглянемо та проаналізуємо основні переваги та недоліки існуючих на даний момент онлайн сервісів пошуку роботи.

Одними із найпопулярнішими онлайн сервісів пошуку роботи є наступні:

- «Robota-UA»
- «Work-UA»

Robota UA – це веб-сайт пошуку роботи в Україні, який надає можливість безкоштовно розміщувати резюме та шукати вакансії в різних сферах діяльності. Сайт пропонує різноманітні інструменти для пошуку роботи, такі як фільтри, що дозволяють обирати роботу за категорією, регіоном, заробітною платою та іншими параметрами [2]. Інтерфейс головної сторінки представлений на рисунку 1.1

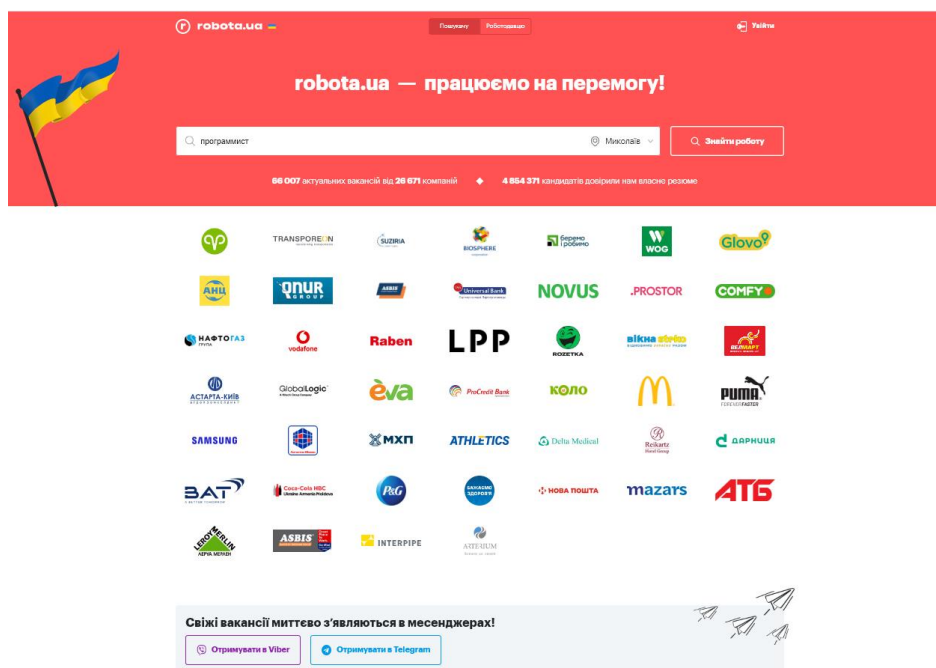


Рисунок 1.1 – Головна сторінка вебсайту «Robota-UA»

Перелік можливостей, які надає даний вебсайт.

- Реєстрація та авторизація роботодавця (рисунок 1.2);
- Реєстрація та авторизація робітника;

- Створення та публікація резюме робітником (рисунок 1.3);
- Перегляд персональної сторінки кандидата;
- Пошук вакансій за фільтрами (рисунок 1.4);
- Створення вакансії роботодавцем та розміщення;
- Перегляд компаній (рисунок 1.5).

The image shows a web form titled "Створіть акаунт роботодавця" (Create Employer Account). At the top, there are two tabs: "пошукачам" (for job seekers) and "роботодавцям" (for employers), with the latter being active. The form contains two input fields: "Email для входу" (Email for login) with an envelope icon and "Пароль для входу" (Password for login) with a key icon and a toggle eye icon. Below these fields is a prominent red button labeled "Зареєструватись" (Register). Under the button, there is a line of small text: "Натискаючи «Зареєструватись», ви підтверджуєте свою згоду з Умовами користування сайтом" (By clicking 'Register', you confirm your agreement with the site's Terms of Use), followed by links to "roboota.ua" and "Політикою конфіденційності" (Privacy Policy). Below this is a link "Увійти в існуючий акаунт" (Log in to existing account). At the bottom, there is a red button with a person icon and the text "Створити акаунт пошукача" (Create job seeker account).

Рисунок 1.2 – Сторінка реєстрації та авторизації роботодавця

← До списку резюме

Особиста інформація *

Чопенко Микита

👤 22 роки

📍 Миколаїв

✎ Редагувати

Додайте фотографію — так резюме стане більш персоналізованим і викличе більше довіри у роботодавця

Контактна інформація *

📞 +38 (095) 219 - 33 - 05 ✓

✉ chopenko.nikita.konkurs@gmail.com

✎ Редагувати

Додайте посилання на месенджери — чим більше зручних каналів для комунікації, тим простіше роботодавцю зв'язатися з вами.

Бажана посада *

➕ Додати

Ким ви хочете працювати та яку зарплату отримувати

Досвід роботи

Немає досвіду роботи ☐

➕ Додати

Освіта *

Admiral Makarov National University of Shipbuilding

📍 121 "Інженерія програмного забезпечення", вища

📅 Николаєв, 2023

✎

Рисунок 1.3— Сторінка створення та публікації резюме робітником

robota.ua 🇺🇦

👍 Рекомендовані 📁 Мої резюме ⭐ Обрані

🔍 Пошук по професії, компанії, навичку

📍 Вся Україна 🔍

👤 Микита Чопенко

Робота в Україні

66 027 вакансій

Гаряча

Менеджер з продажу

25 000 — 50 000 ₴ + % + бонуси

ПРОМІНВЕСТ-ТПК, ТОВ Київ

Бонуси / премії Безкоштовне навчання Офіс біля метро Офіс в центрі

👤 Відгукнутись ☆ 🗨

PRC PHARMA

Гаряча

Фінансовий аналітик

PRO PHARMA Київ

👤 Відгукнутись ☆ 🗨

Рисунок 1.4 – Сторінка пошуку вакансій

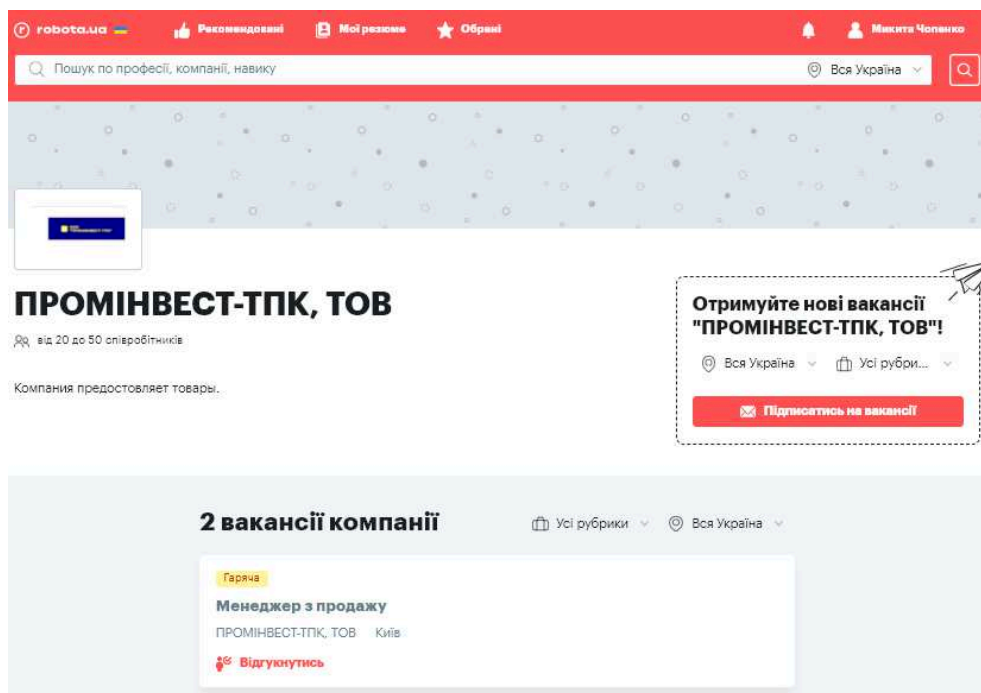


Рисунок 1.5 – Сторінка перегляду компанії

Головні переваги вебсайту пошуку роботи «Robota-UA» [2]

- 1) Можливість шукати вакансії за допомогою різних фільтрів. Вони допомагають шукачам точно шукати роботу, які відповідають критеріям.
- 2) Можливість конструювати та публікувати власне резюме. Вебсайт дає змогу додавати інформацію про кваліфікації, досвід роботи, освіту, ключові навички. Головним плюсом є те, що можна створювати декілька резюме та відправляти на обрану вакансію.
- 3) Нотифікації. Додаток дає змогу підписатися на розсилку повідомлень, якщо знайдеться вакансія ви отримаєте повідомлення на електронну адресу.

Недоліків під час користуванням сервісу не було знайдено. Вебсайт має багатий функціонал та зручний користувацький інтерфейс. Він заслуговує високий рейтинг серед сайтів пошуку роботи.

Work-UA – це популярна українська онлайн платформа, де роботодавці можуть публікувати оголошення про вакансії, а працівники можуть шукати та подавати заявки на робочі місця. Це один з найбільших веб-сайтів пошуку роботи в Україні та пропонує широкий спектр вакансій в різних галузях [2].

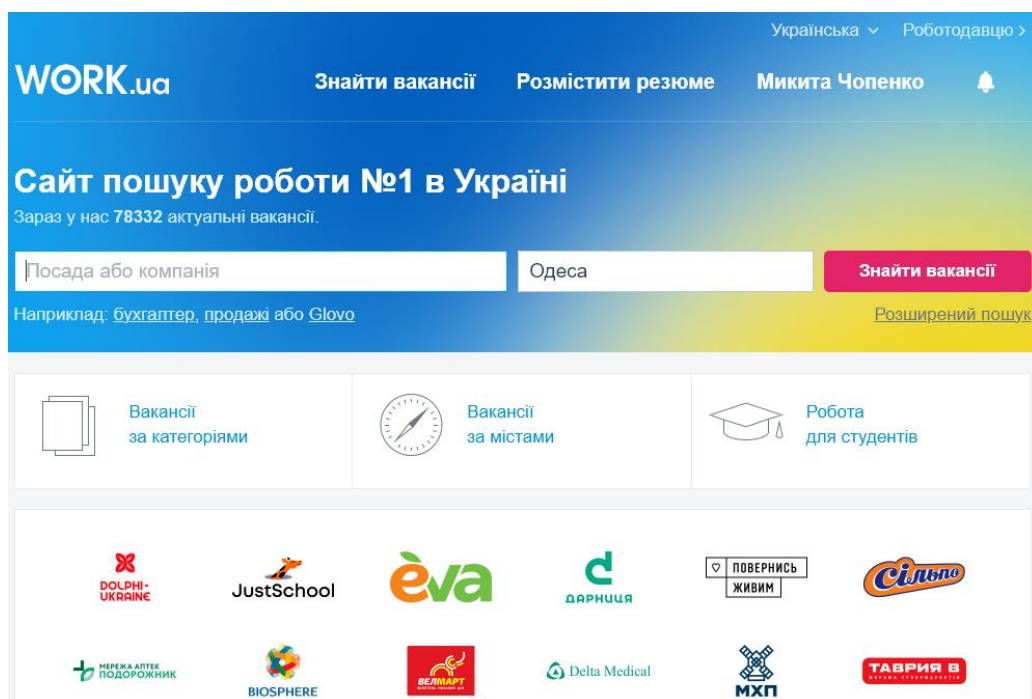


Рисунок 1.6 – Головна сторінка вебсайту «Work-UA»

Перелік можливостей, які надає даний вебсайт:[3]

- Реєстрація та вхід для шукача
- Реєстрація та від для роботодавця;
- Створювання резюме (рисунок 1.7);
- Пошук вакансій за допомогою вибраного фільтру (рисунок 1,8);
- Створення та публікування вакансії для роботодавця;
- Перегляд статей та новин, які пов'язані з пошуком роботи (рисунок 1.9);
- Відгук до відповідної вакансії.

WORK.ua

⚡ — — — — —


 Створюйте резюме українською. Активна громадянська позиція — це ваша перевага у боротьбі за роботу мрії.

Візитка ⓘ

Ваше ім'я та прізвище*:

Посада, на якій ви хочете працювати*:

Де ви хочете працювати*:

[Додати ще](#)

Дата народження:

-

-

-

[Далі >](#)

Рисунок 1.7 – Сторінка створення резюме для шукача

WORK.ua

[Знайти вакансії](#)
[Розмістити резюме](#)
Микита Чопенко
🔔



Робота у Миколаєві

Зараз у нас **888** актуальних вакансій.

Знайти

[Розширений пошук](#)

888 вакансій у Миколаєві за 30 днів, за датою

[Підписатися на пошук](#)

Торговий представник з авто

Ставка + КРІ

Сузір'я, ГК · VIP · Миколаїв

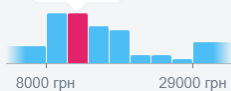
Повна зайнятість. Досвід роботи від 1 року.

Група компаній «Сузір'я» — національна компанія з виробництва зоотоварів, яка понад 30 років є надійним... >

Зберегти
 Не показувати
Гаряча

Середня зарплата у Миколаєві

14000 грн



8000 грн 29000 грн

[Докладніше >](#)

Шукаєте роботу у Миколаєві?

[Розмістіть резюме](#), щоб роботодавці могли запропонувати вам роботу.

Рисунок 1.8 – Сторінка пошуку вакансій за допомогою фільтрів

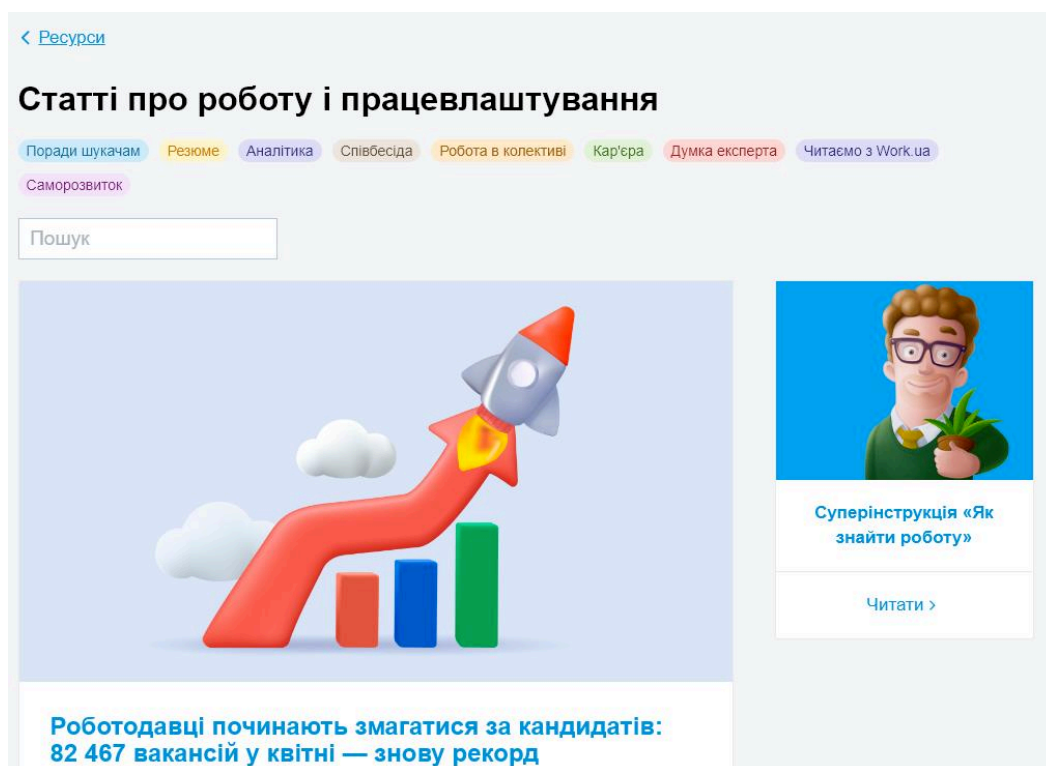


Рисунок 1.9 – Сторінка перегляду статей та новин

Переваги вебсайту пошуку роботи: [3]

- Широкий вибір вакансій;
- Зручний пошук вакансії за допомогою різноманітних фільтрів (категорія, вид зайнятості, зарплатах;
- Можливість створювати власне резюме, за допомогою зручного конструктора;
- Перегляд новин, статей для працевлаштування;
- Збереження вакансій, які сподобалися;
- Перегляд рекомендованих вакансій в особистому кабінеті;

Недоліки сайту:

- Невелика кількість вакансій;
- Немає можливості отримувати нотифікацію, коли буде створена нова вакансія, яка підходить до вашого резюме.

Можна зробити висновок, що сучасні вебсайти пошуку роботи дуже подібні за функціоналом. Загалом, вони мають дуже подібний користувацький

інтерфейс та багатий функціонал. Шукачі та роботодавця за допомогою конструкторів можуть легко задовільнити свої потреби.

1.3 Аналіз методів та інструментів розробки веб-сервісів

Існує багато різних методів розробки, і вибір конкретного методу залежить від багатьох факторів, таких як розмір проекту, вимоги до функціональності, строк та бюджет.

Waterfall – цей метод є традиційним і популярним методом розробки, в якому кожен етап проекту проходить від початку до кінця перед тим, як перейти до наступного етапу. Цей метод вимагає, щоб всі етапи проекту були повністю завершені, перш ніж почати наступний етап, що може призвести до затримок у розробці, якщо щось піде не так. Однак, цей метод дозволяє зручно керувати проектом і контролювати його витрати [5].

Agile – базується на ітераційному підході, в якому розробка проводиться у невеликих ітераціях або спринтах, які зазвичай тривають від одного до чотирьох тижнів. Кожен спринт складається зі збору вимог, планування, розробки, тестування та випуску. Agile дозволяє розробникам більш ефективно відповідати на зміни вимог до проекту і підтримувати більш тісний контакт з клієнтом, але вимагає від розробників високої кваліфікації та відкритості до змін [5].

При створенні вебсайтів можна використовувати конструктори. Конструктор – це інструмент, який дозволяє швидко та легко створювати вебсайти без необхідності програмування. Він має вбудований інтерфейс, що дозволяє користувачам створювати веб додатки шляхом вибору та редагування готових шаблонів, вибору функцій та параметрів, що потрібні для їх проекту. Можна визначити такі переваги при користуванні. Це простота, допомагає початківцям створювати власні сторінки, економія часу та коштів, готові шаблони та функції.

Але для того щоб створювати великі додатки необхідно розділити

реалізацію веб-сервісу на дві частини: backend та frontend.

Backend (або back-end) – це частина веб-додатка, яка відповідає за логіку обробки даних, забезпечення взаємодії з базами даних та іншими зовнішніми сервісами. Backend складається з програмного забезпечення, яке запускається на сервері та взаємодіє з клієнтськими додатками (frontend), оброблює запити від користувачів, зберігає дані та повертає результати виконання запитів у вигляді HTML, JSON, XML або інших форматах [6].

Frontend (або front-end) – це частина веб-додатка, яка відповідає за відображення інтерфейсу користувача та взаємодію з ним. Frontend складається з програмного забезпечення, яке запускається у браузері користувача та відповідає за відображення сторінок, взаємодію з користувачем та передачу даних на сервер (за допомогою запитів до backend) [6].

Майже кожен веб додаток, який є в інтернеті, використовує мову програмування JavaScript. Це високорівнева мова програмування, яка використовується для створення динамічного контенту на веб-сторінках. Для додавання зовнішнього вигляду, стилів використовують CSS. CSS (Cascading Style Sheets) – це мова опису стилів, що використовується для оформлення та візуального оформлення веб-сторінок. CSS дозволяє розділити опис структури веб-сторінки від опису її вигляду.

Щоб реалізувати backend найчастіше використовують такі мови програмування:

- JavaScript;
- PHP;
- Python.

Це дозволяє створювати функціонал до додатків, наприклад: реєстрація, авторизація, форма зворотного зв'язку, електронні нотифікації.

Після детального аналізу, було вирішено при розробці програмного забезпечення сервісу пошуку роботи «Fast-Work» реалізувати окремо backend та frontend.

1.4 Постановка задачі на розробку веб-сервісу пошуку роботи «Fast-Work»

На основі аналізу предметної області поставлена задача на розробку веб-сервісу пошуку роботи «Fast-Work».

Необхідно розробити програмне забезпечення веб-сервісу пошуку роботи для автоматизації розміщення резюме та вакансій.

Метою розробки програмного забезпечення є:

- Надання зручного та ефективного інструменту для знаходження робочих місць, які відповідають вимогам шукача;
- Можливість створення профілю зі своїм досвідом, навичками, освітою та іншою інформацією, яка може зацікавити потенційного роботодавця;
- Надання можливості перегляду вакансій та компаній, які можуть бути зацікавлені в співпраці;
- Надання корисних порад та ресурсів для людей, які шукають роботу, що можуть допомогти їм підвищити свої шанси на успіх.

Після проведення аналізу предметної області та дослідження існуючих онлайн сервісів пошуку роботи, було встановлено, що вони є досить популярними на ринку програмного забезпечення. Проте не всі з них надають повного функціоналу для пошуку та публікування резюме.

1) Вимоги до функціональних характеристик:

Повинні бути реалізовані наступні функції у програмному забезпеченні веб-сервісу пошуку роботи «Fast-Work».

Адміністратор:

- Реєстрація та авторизація;
- Додавання та редагування вакансій;
- Видалення вакансій;
- Додавання категорій;

- Додавання новин;
- Вихід з особистого кабінету.

Шукач:

- Створення та публікація резюме;
- Перегляд доступних вакансій;
- Відгук на вакансію;
- Перегляд вакансій відгуку;
- Вихід з особистого кабінету;
- Перегляд новин.

2) Вимоги до вхідних та вихідних характеристик:

Вхідні дані – це дані, які вводяться в програму або систему для обробки або збереження. Текстові дані, які користувач додає на сайті зберігаються за допомогою бази даних.

Вхідними даними також можуть бути файли, які розміщені на веб-сервісі, зазвичай вони зберігаються на основному сервері або на облачному.

Вхідними даними є:

- Інформація при реєстрації шукача роботи;
- Інформація про додавання резюме;
- Дані про вакансію;
- Інформація про компанію, яка додає вакансію;
- Дані про новини, які розміщені на веб-сервісі;
- Дані про категорії.

Вихідні дані – це дані, що отримуються після обробки вхідних даних в рамках деякого процесу або програми. Наприклад, вихідні дані можуть бути: таблицями, графіками, зображеннями, текстовими документами тощо. Частина вихідних даних складається з вхідних, якщо програмне забезпечення

інформаційне.

Вихідними даними є:

- Список сформованих вакансій компаній;
- Сформоване резюме шукача роботи;
- Сформований список користувачів;
- Дані про список вакансій;
- Зображення;
- Новини працевлаштування;
- Категорії для пошуку роботи.

3) Вимоги до складу й параметрів технічних засобів:

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплекції:

- CPU: Intel® Celeron® – 2.16 МГц;
- RAM: 2 GB;
- HDD: 20 GB вільного місця;
- Пропускна здатність 100мб/с.

4) Вимоги до інформаційної та програмної сумісності:

Програмне забезпечення підтримує усі сучасні операційні системи:

- Windows 7 та вище;
- Linux 20.04 та вище.

Список веб браузерів, на яких працює веб-сервіс:

- Chrome (114.0.5735.110 та вище);
- Mozilla Firefox (114.0 та вище).

2 РОЗРОБКА ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST-WORK»

2.1 Аналіз вимог до програмного забезпечення

Вимоги до програмного забезпечення веб-сервісу пошуку роботи можна поділити на функціональні та нефункціональні.

Функціональні вимоги визначають, які функції та можливості повинно мати програмне забезпечення. Вони описують що саме програма повинна робити та які дії вона має виконувати для задоволення потреб користувачів. Функціональні вимоги вказують на конкретні функції, які мають бути реалізовані.

Нефункціональні вимоги визначають характеристики, якості та обмеження, яким повинно відповідати програмне забезпечення. Вони описують вимоги до продуктивності, безпеки доступності, масштабованості.

Функціональні вимоги до програмного забезпечення веб-сервісу пошуку роботи викладені в пункті «Постановка задачі».

Нефункціональні вимоги до програмного забезпечення веб-сервісу пошуку роботи «Fast-Work»

Продуктивність:

- Середній час відповіді на запит менше 1 секунди;
- Максимальний час завантаження сторінок не повинен перевищувати 3 секунди.

Безпека:

- Весь обмін даними повинен передаватися за допомогою протоколу HTTPS;

- Користувачі повинні мати можливість створювати облікові записи;

Доступність:

- Сервіс повинен бути доступний для користувачів 24 години за добу;
- У разі виникнення технічних проблем та непередбачуваних ситуацій;
- Система повинна швидко відновлюватися.

Масштабованість:

- Система повинна мати можливість масштабуватись горизонтально для підтримки зростаючої кількості користувачів та великого обсягу;

Інтерфейс:

- Користувальницький інтерфейс повинен бути інтуїтивно зрозумілим, простим у користуванні та привабливим для користувачів різного рівня технічної підтримки;

- Система повинна бути адаптивною та підтримувати відображення на різних пристроях.

2.1.1 Побудова моделі варіантів використання

У процесі аналізу предметної області було виділено 3 сутності:

- Шукач;
- Адміністратор;
- Гість.

Шукач роботи – це особа, яка активно займається пошуком роботи або стажування, має певний рівень кваліфікації та досвіду роботи і бажає знайти роботу, яка відповідає його потребам та можливостям.

Адміністратор сервісу пошуку роботи – це особа або команда фахівців, яка відповідає за управління та підтримку роботи веб-сайту пошуку роботи

Гість – це може бути особа, яка займається пошуком роботи або звичайний відвідувач сайту в пошуках нової інформації.

Провівши аналіз функцій та вимог програмного забезпечення було виділено такі варіанти використання:

Адміністратор:

- Реєстрація та авторизація;

- Додавання та редагування вакансій;
- Видалення вакансій;
- Додавання категорій;
- Додавання новин;
- Вихід з особистого кабінету.

Шукач:

- Реєстрація та авторизація;
- Створення та публікація резюме;
- Перегляд доступних вакансій;
- Відгук на вакансію;
- Перегляд вакансій відгуку;
- Вихід з особистого кабінету;
- Перегляд новин.

Гість:

- Реєстрація та авторизація;
- Перегляд доступних вакансій;
- Перегляд новин.

Діаграма варіантів використання зображена на рисунку 2.1

Таблиця 2.1 – Варіанти використання та їх короткий опис

Код	Актор	Назва	Формулювання
1	2	3	4
A1	Адміністратор	Авторизація	Дозволяє авторизуватися на сайті
A2	Адміністратор	Вихід з кабінету	Вихід з особистого кабінету
A3	Адміністратор	Редагування вакансії	Дозволяє редагувати інформацію про вакансію
A4	Адміністратор	Додавання категорій	Дозволяє додати категорію для вакансії
A5	Адміністратор	Додавання компаній	Дозволяє додати компанію
A6	Адміністратор	Додавання новин	Дозволяє додати новини на сайт
K1	Шукач	Авторизація	Дозволяє авторизуватися на сайті
K2	Шукач	Вихід з кабінету	Вихід з особистого кабінету
K3	Шукач	Створення резюме	Створення резюме та публікація на веб сайті
K4	Шукач	Відгук на вакансію	Перегляд відгуків для вакансій
K5	Шукач	Редагування резюме	Дозволяє редагувати резюме та публікувати на сайті
K6	Шукач	Перегляд новин	Дозволяє переглядати новини на сайті
Г1	Гість	Перегляд вакансій	Дозволяє переглядати вакансії
Г2	Гість	Перегляд новин	Дозволяє переглядати новини на сайті
Г3	Гість	Реєстрація на сайті	Дозволяє зареєструватися на сайті

Проведемо конкретизацію основних варіантів використання.

Г3 – Реєстрація

Основна діюча особа: Гість

Інші учасники прецеденту: відсутні зв'язки з іншими варіантами.

Передумови, необхідні для ініціювання прецеденту: клієнт перейшов на сторінку реєстрації

Короткий опис

Даний варіант використання дозволяє зареєструватися на вебсайті.

Г1 – Перегляд вакансій

Основні діючі особи: Шукач, Гість

Інші учасники прецеденту: відсутні зв'язки з іншими варіантами.

Передумови, необхідні для ініціювання прецеденту: клієнт відкрив сторінку з доступними вакансіями.

Короткий опис

Даний варіант використання дозволяє шукачу та гостю переглядати доступні вакансії .

К4 – Відгук на вакансію

Основна діюча особа: Шукач

Інші учасники прецеденту: відсутні зв'язки з іншими варіантами

Передумови, необхідні для ініціювання прецеденту: клієнт відкрив сторінку з вакансіями або вакансію та натиснув кнопку відгукнутися.

Короткий опис

Даний варіант використання дозволяє шукачу відгукнутися на вибрану вакансію, а саме відправити своє резюме.

К5 – Редагування резюме

Основна діюча особа: Шукач

Інші учасники прецеденту: відсутні зв'язки з іншими варіантами.

Короткий опис

Даний варіант використання дозволяє шукачу редагувати особисту інформацію у резюме.

Авторизація акторів:

1) Адміністратор

Адміністратор має свій власний обліковий запис. Завдяки цьому адміністратор має можливість виконувати глобальні функції.

2) Гість

Гість може авторизуватися до сайту, для цього йому треба вказати логін та пароль, який був вказаний при реєстрації.

Після авторизації гість отримає змогу використовувати більше функціоналу.

2.2 Архітектура програмного забезпечення

Архітектура програмного забезпечення показує взаємодії компонентів інтерфейсу з внутрішніми процесами програми. Головна ідея архітектури полягає в спрощенні процесу сприйняття системи. Архітектура програмного забезпечення дозволяє створити чітку структуру, за якою зручно працювати розробникам.

Архітектура програмного забезпечення вирішує низку важливих завдань:

- Визначає структуру додатку;
- Визначає поведінку і взаємодію елементів;
- Дозволяє зрозуміти логічні взаємозв'язки додатку;
- Дає можливість коректно вести документацію та розписувати функціонал [7].

Для розробки програмного забезпечення найчастіше застосовують шаблони. Вони дозволяють вирішувати однотипні завдання швидшим способом, що позитивно впливає на час розробки та спрощує багато процесів.

Для розробки серверної частини програмного забезпечення веб-сервісу пошуку роботи «Fast-Work» було вибрано шаблон проектування модель-перегляд-контролер. Цей шаблон ділиться на 3 компоненти. Модель містить дані про сутності програми. Перегляд відображає дані, які були отримані з контролерів. Контролер відповідає за надання даних, які присутні у моделі. Серверна частина розбита на декілька компонентів. Контролери обробляють запити від клієнтів, моделі представляють структуру даних у додатку, наприклад: вакансії, користувачі, компанії, резюме і так далі.

Для розробки клієнтської частини було вибрано шаблон проектування

Модель-Вид. Структура клієнтської частини складається з мутацій, стейту та дій. Дії дозволяють виконувати певну логіку, наприклад: отримання вакансій, вхід в особистий кабінет, пошук вакансій тощо. Мутації дають можливість змінювати дані нашого стейту, вони зазвичай викликаються в діях. Стейт дозволяє зберігати дані програмного забезпечення під час роботи.

Частиною архітектури сучасного програмного забезпечення є база даних. Вона дозволяє зберігати певну інформацію, наприклад: дані про вакансію, користувачів, резюме та іншу інформацію.

Архітектура програмного забезпечення веб-сервісу пошуку роботи «Fast-Work» об'єднує різні компоненти для забезпечення ефективної та надійної роботи системи. Вона включає в собі клієнтський інтерфейс, серверну частину та базу даних.

2.2.1 Розробка діаграми компонентів

У UML діаграма компонентів зображує, як компоненти з'єднані між собою для формування великих компонентів або програмних систем. Він ілюструє архітектуру програмних компонентів та залежності між ними. Ці компоненти програмного забезпечення, включаючи компоненти часу виконання, виконувані компоненти, також компоненти вихідного коду [8]. Діаграма компонентів зображена на рисунку 2.2.

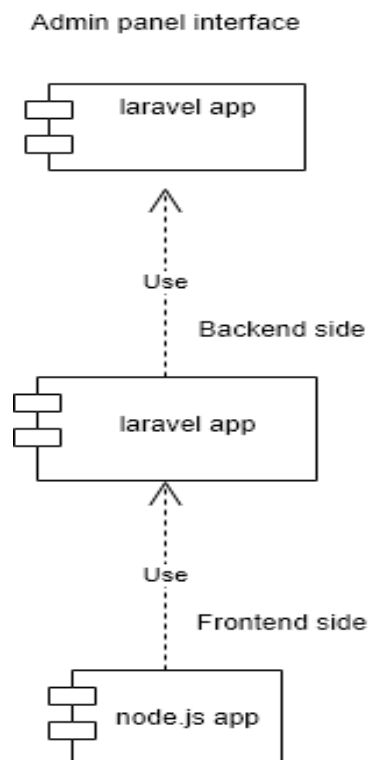


Рисунок 2.2 – Діаграма компонентів веб-сервісу пошуку роботи «Fast-Work»

2.2.2 Розробка діаграми розміщення

Діаграма розміщення програмного забезпечення (Software deployment diagram) – це тип діаграми, що використовується в області програмного забезпечення для відображення того, як програмні компоненти взаємодіють між собою та з апаратним забезпеченням в процесі розгортання системи.

Ця діаграма показує фізичну структуру системи, її компоненти, а також залежності між цими компонентами. Кожен компонент системи може бути представлений у вигляді віртуальної машини або фізичного сервера. Діаграма розміщення дозволяє визначити, на яких компонентах буде запущена система, які компоненти будуть взаємодіяти між собою, а також які сервіси будуть доступні користувачам.

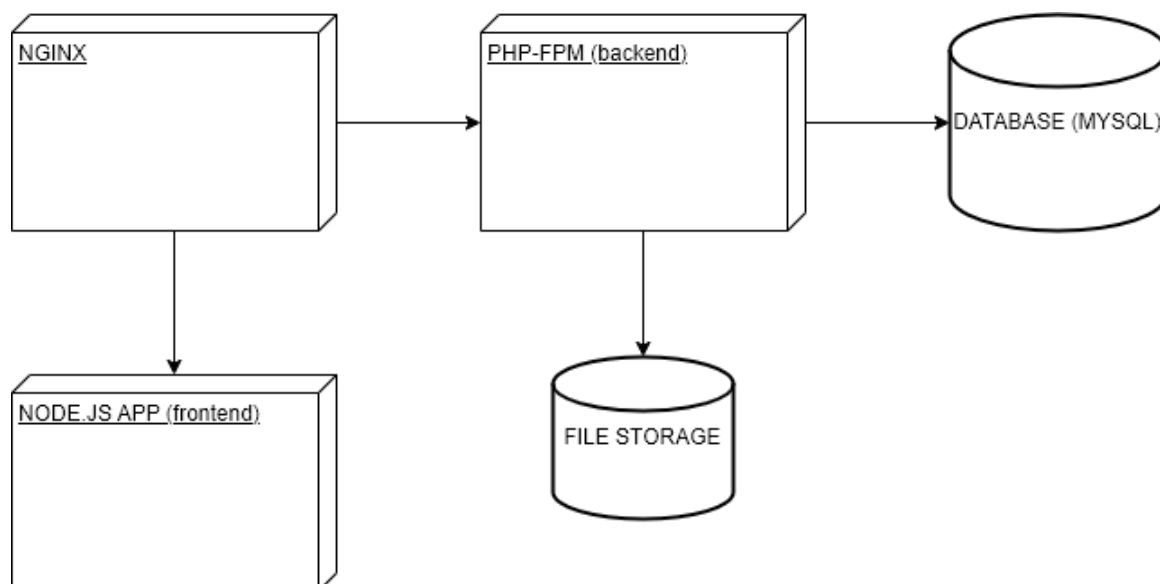


Рисунок 2.3 – Діаграма розміщення веб-сервісу пошуку роботи «Fast-Work»

2.3 Проект програмного забезпечення

2.3.1 Ескізний проект програмного забезпечення

Ескізний проект – це перша стадія розробки програмного забезпечення, на якій створюється загальне уявлення про функціональність майбутньої системи. Для створення ескізного проекту використовуються спеціальні методи і інструменти, такі як мова моделювання UML та об’єктно-орієнтована методологія. У рамках ескізного проекту створюються моделі варіантів використання, діаграми діяльності та прототип інтерфейсу користувача, що дозволяє зрозуміти функціональні можливості майбутньої системи та її взаємодію з користувачем.

На основі проведеного аналізу вимог був розроблений ескізний проект, який використовує спеціальну мову моделювання UML. Для проектування програмного забезпечення на даному етапі використовувалась об’єктно-орієнтована методологія, і були створені наступні діаграми:

- Діаграма діяльності;
- Діаграма класів.

2.3.2 Діаграма діяльності

Діаграма діяльності дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів.

Діаграми діяльності є аналогом блок-схеми будь-якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребрами – переходи між діями.

Діаграми діяльності доцільно використовувати для аналізу: - змісту сценаріїв застосування проектованої системи; - взаємодії потоків робіт різних сценаріїв; – виконання сценаріїв у багатопроцесорних обчислювальних середовищах. Ці діаграми широко використовуються в описі поведінки, що включає велику кількість паралельних процесів. Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції. Таким чином, діаграму діяльності можна вважати приватним випадком діаграми станів.

Основним напрямком використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно надати алгоритми їх виконання [9].

Діаграми діяльності основних варіантів використання наведенні на рисунках 2.4 – 2.6

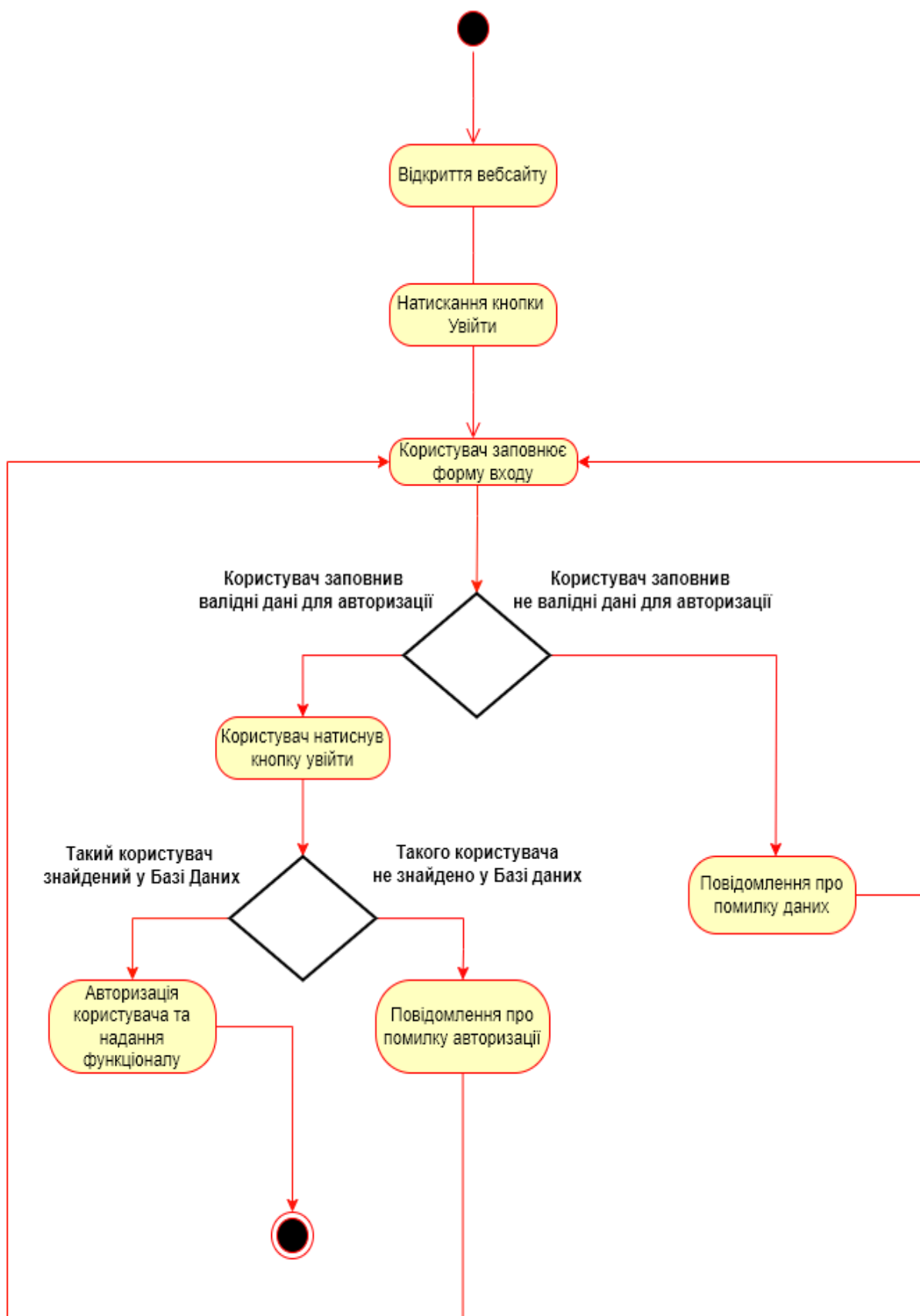


Рисунок 2.4 – Діаграма діяльності використання «Авторизація»

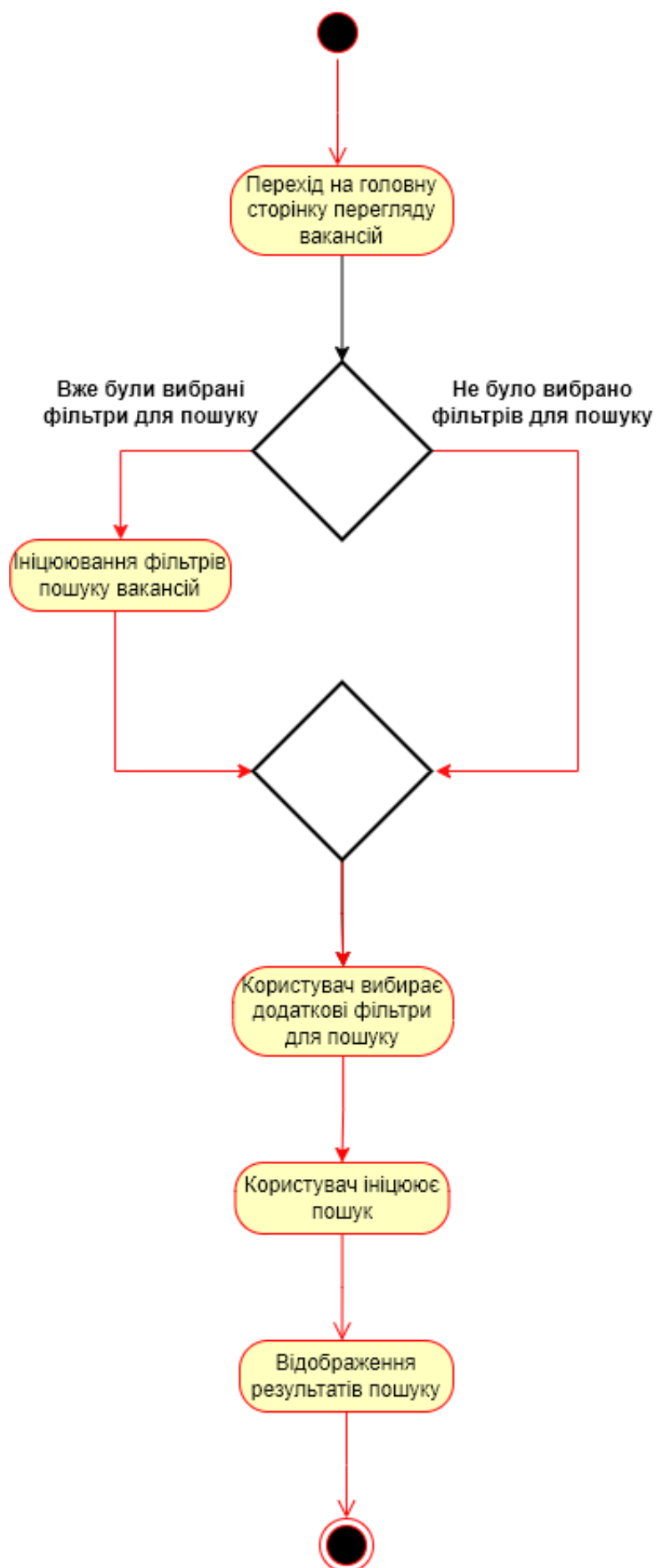


Рисунок 2.5 – Діаграма діяльності використання «Пошук вакансій»

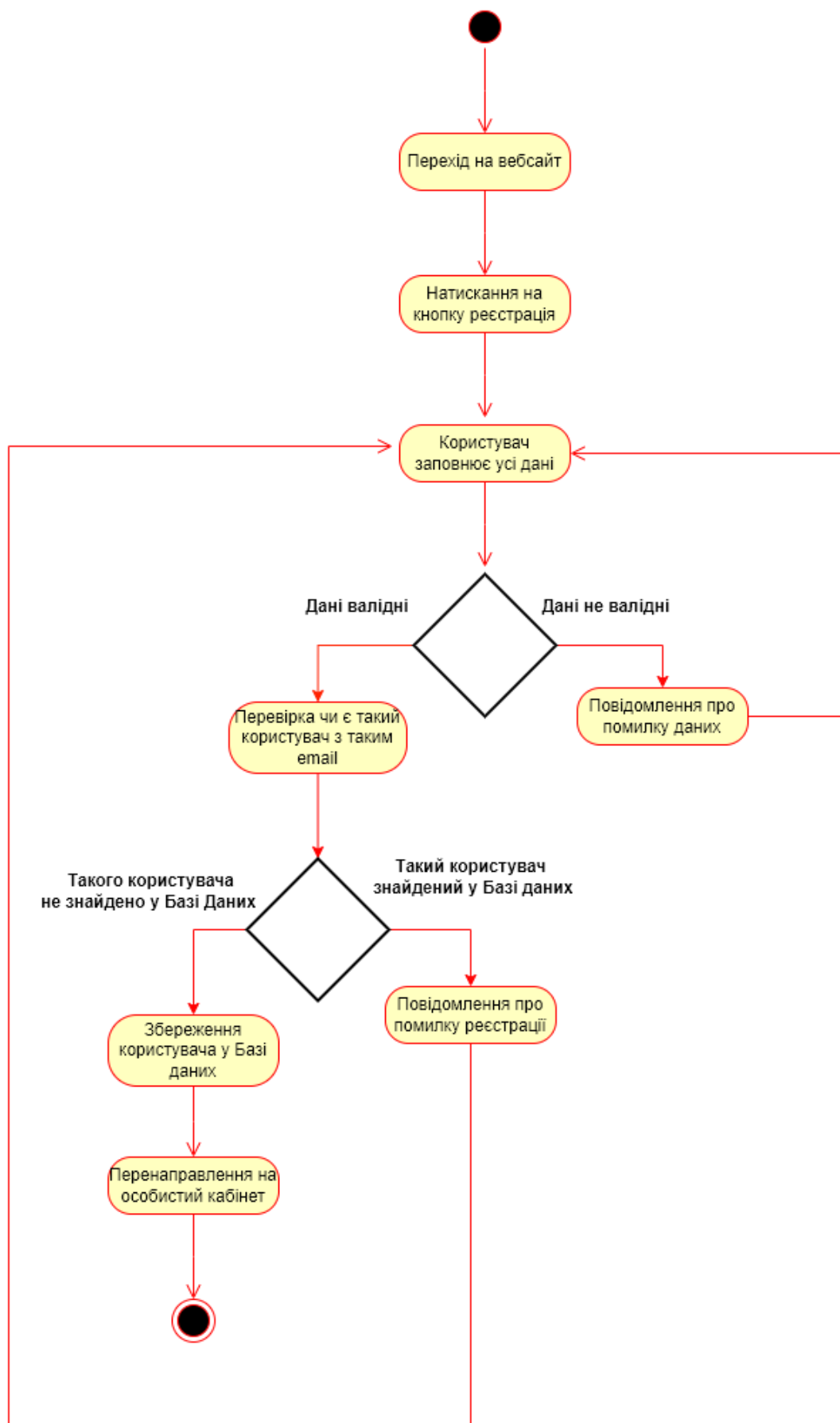


Рисунок 2.6 – Діаграма діяльності використання «Реєстрація»

2.3.3 Інформаційна модель

Інформаційна модель програмного забезпечення – це абстрактна модель, яка описує структуру та організацію даних, що використовуються в програмному забезпеченні. Вона визначає сутності і взаємозв'язки між ними. Інформаційна модель зображена на рисунку 2.7.

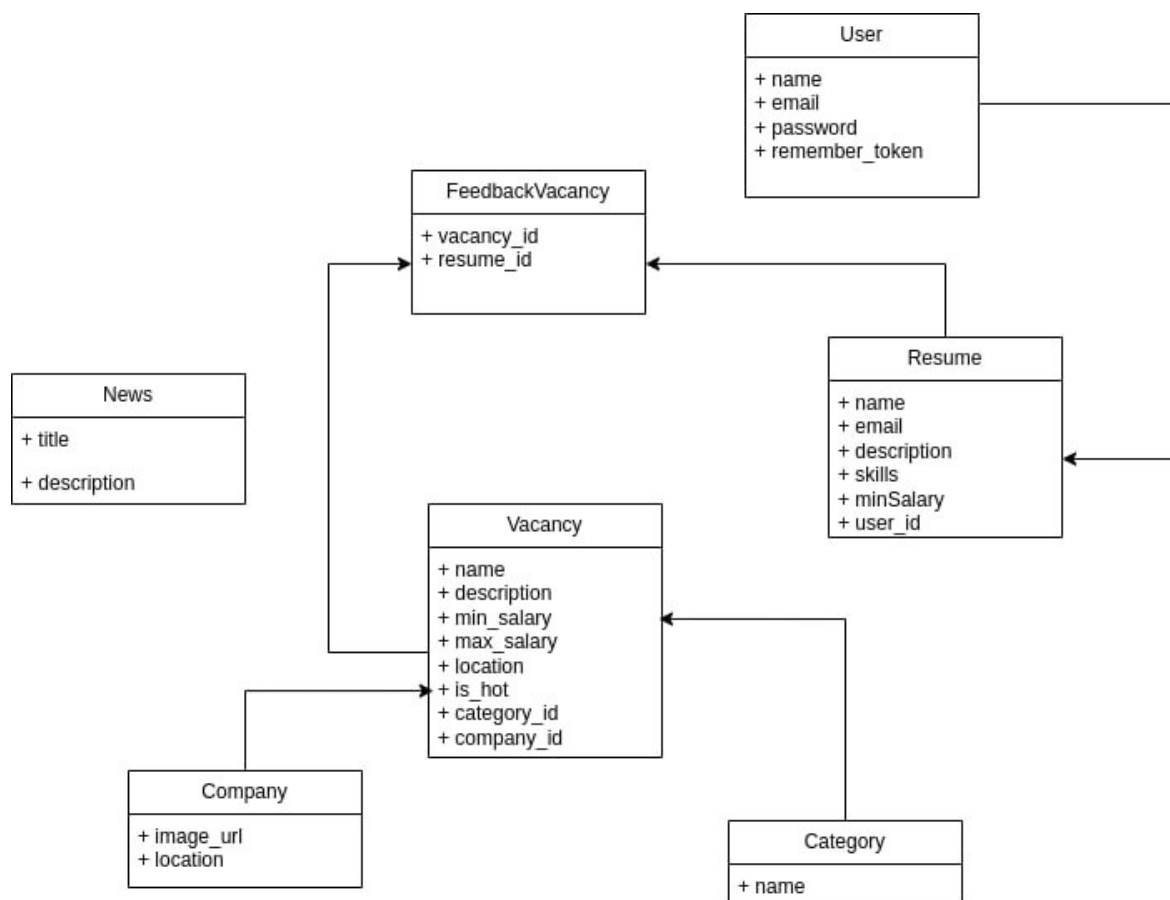


Рисунок 2.7 – Інформаційна модель веб-сервісу пошуку роботи «Fast-Work»

2.3.4 Розробка ескізу інтерфейсу

Прототип інтерфейсу користувача – це зразок майбутнього ІТ– продукту, який використовується для тестування та демонстрації програмного забезпечення. Він часто створюється на ранніх етапах розробки додатків. Для створення користувацького інтерфейсу використовують різноманітні інструменти.

Наприклад: Sketch, Figma, InVision або паперові прототипи.

Мета прототипу полягає в тому, щоб продемонструвати усі майбутні можливості програмного забезпечення, знайти недоліки та вдосконалити їх. Зразок може бути інтерактивним у вигляді анімацій, який буде ілюструвати переходи між сторінками та взаємодію з інтерфейсом. Все це дозволяє отримати реальні відгуки від користувачів та покрити майже всі вимоги користувачів до майбутнього продукту.

Майже будь-який вебсайт складається з різних сторінок, які служать для представлення контенту додатку. Наведемо деякі приклади типів сторінок:

- Головна сторінка (Home page), на якій зазвичай міститься інформація про продукт, компанію, послуги
- Про нас (About us), на цій сторінці розповідається про компанію, про послуги які вона надає і так далі.
- Блог (Blog), може бути окремою сторінкою або розділом на веб-сайті, де публікуються статті, новини, вакансії.

Можна виділити такі переваги та мінуси при створенні прототипу інтерфейсу користувача.

Переваги:

- Прототип дозволяє команді проекту та користувачам зрозуміти та уточнити усі вимоги вебсайту. Можна обговорити, зібрати фідбеки реальних користувачів та прийняти рішення щодо функцій.
- Прототипування дозволяє зменшити час на реалізацію програмного продукту.

Недоліки:

- Створення прототипу займає певний час та зусиль. Це може затримати процес розробки програмного забезпечення.
- Фінанси. Якщо проект великий, то створення різних прототипів може потребувати багато коштів.

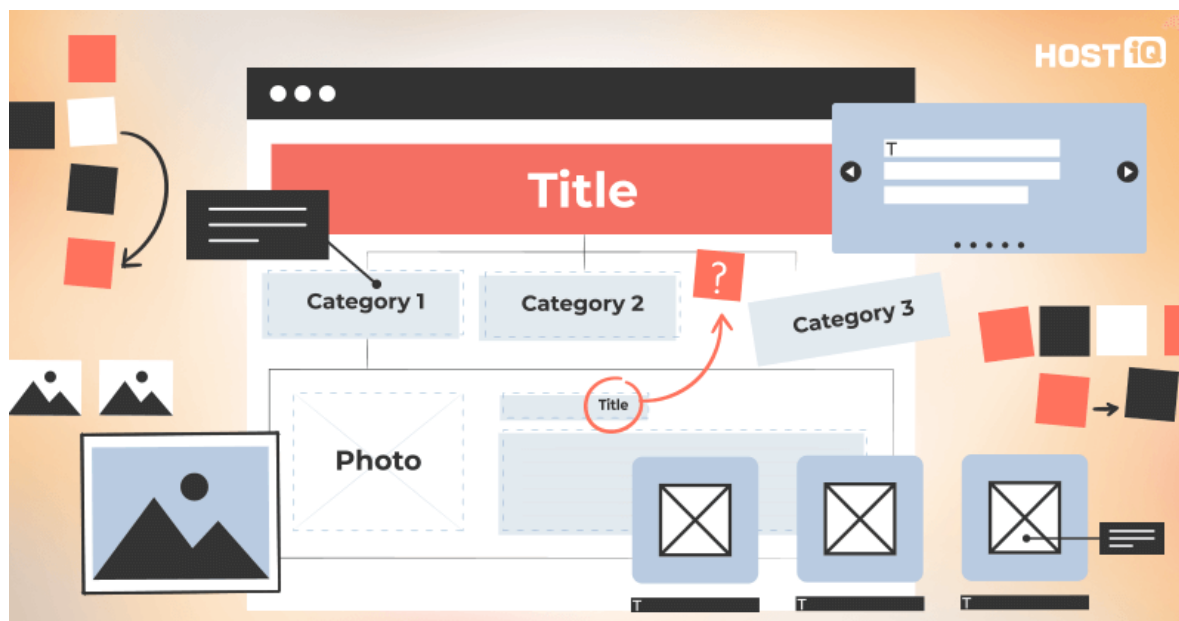


Рисунок 2.8 – Прототип майбутнього веб-сервісу

Прототип головної сторінки зображений на рисунку 2.9.

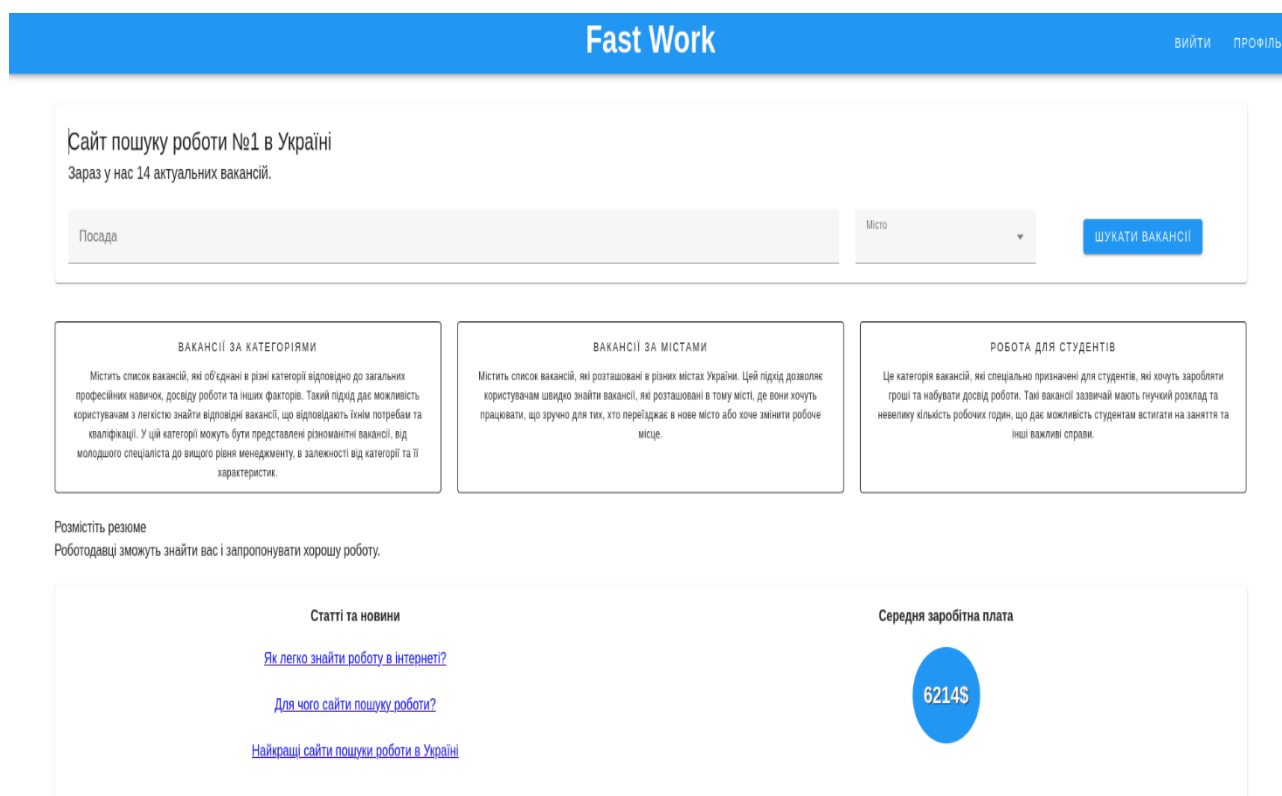


Рисунок 2.9 – Прототип головної сторінки веб-сервісу

Прототип сторінки пошуку вакансій зображений на рисунку 2.10

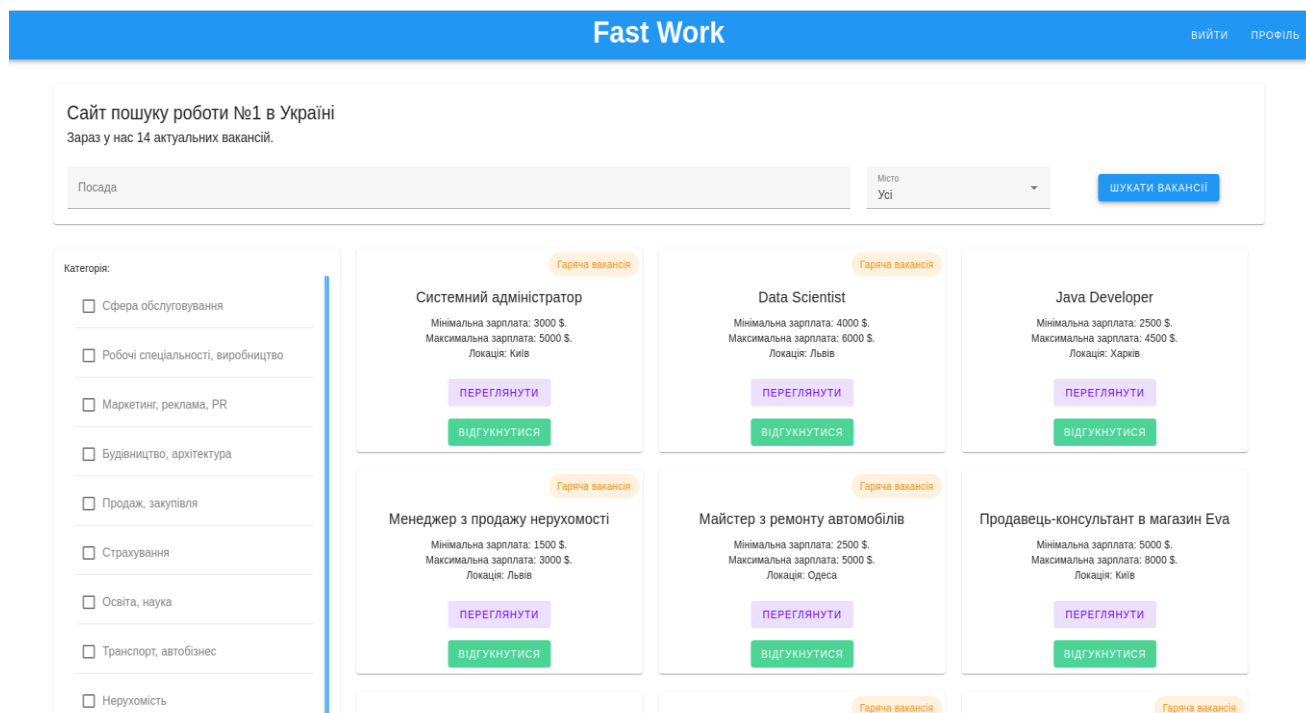


Рисунок 2.10 – Прототип сторінки пошуку вакансій

Шукач, коли йому сподобається деяка вакансія зможе відгукнутися на неї. Сторінки відгуків зображені на рисунках.

Прототип інтерфейсу є важливим етапом в процесі розробки будь-якого програмного продукту або веб-сайту. Він дозволяє команді проекту оцінити і визначити вигляд та функціональність інтерфейсу до його фінальної реалізації. Після оцінки та тестування прототипу можна виявити і виправити потенційні проблеми та недоліки, забезпечити зручну та ефективну взаємодію з користувачем і забезпечити високу якість кінцевого продукту.

2.3.5 Технічний проєкт програмного забезпечення

Базуючись на ескізному проєкті, було розроблено технічний проєкт для програмного забезпечення веб-сервісу. У процесі розробки були створено діаграму класів, специфікацію класів та логічну модель даних.

- vacancies – метод повертає вакансії, які прив'язані до категорії

Назва: Company

Атрибути:

- name

Відповідальність класу: Відповідає за роботу з компаніями та приймає участь у роботі в контролерах.

Функції, які надає клас:

- vacancy() – метод повертає вакансію, які прив'язані до компанії;
- resume() – метод повертає резюме, які прив'язані до компанії;

Назва: Vacancy

Атрибути:

- name;
- description;
- min_salary;
- max_salary;
- location;
- is_hot;
- category_id;
- company_id;

Відповідальність класу: Відповідає за доступ вакансій та приймає участь у контролерах.

Функції, які надає клас:

- category() – метод повертає категорію, яка прив'язана до вакансії;
- company() – метод повертає компанію, яка прив'язана до вакансії;
- feedbackVacancy() – метод повертає фідбек до вакансії;

Назва: Resume

Відповідальність класу: Відповідає за доступ класу та приймає участь у контролерах:

Атрибути:

- name

- email
- description
- skills
- minSalary
- user_id

Функції, які надає клас:

- feedbackResume() – метод повертає фідбек для резюме;

Назва: AuthController

Відповідальність класу: Відповідає за вхід та реєстрацію у систему;

Функції, які надає клас:

- register() – реєстрація для користувача;
- login() – вхід для користувача;
- me() – інформація про користувача;

Назва: CategoryController

Відповідальність класу: Відповідає за доступ категорій

Функції, які надає клас:

- index() – список категорій

Назва: CompanyController

Відповідальність класу: Відповідає за доступ компаній

Функції, які надає клас:

- index() – перегляд компаній
- get() – перегляд компанії по id

Назва: NewsController

Відповідальність класу: Відповідає за доступ новин

Функції, які надає клас:

- index() – перегляд новин
- get() – перегляд новини по id

Назва: ResumeController

Відповідальність класу: Відповідає за створення резюме, оновлення та перегляд

Функції, які надає клас:

- store() – збереження резюме;
- update() – оновлення резюме;
- index() – перегляд резюме;

Назва: StatisticController

Відповідальність класу: Відповідає за перегляд статистики

Функції, які надає класу:

- get() – отримування резюме;

Назва: VacancyController

Відповідальність класу: Відповідає за створення вакансії, перегляд локацій вакансії, фідбеки

Функції, які надає клас:

- index() – отримування вакансії;
- get() – отримування специфічної вакансії;
- locations() – отримування локацій для вакансії;
- feedback() – отримування відгуку для вакансії;

2.3.6 Діаграма послідовності

Діаграма послідовності – різновид діаграми UML. Діаграма послідовності відображає взаємодію об'єктів впорядкованих за часом. Зокрема такі діаграми відображають задіяні об'єкти та послідовність надісланих повідомлень.

Діаграма послідовності для варіанту використання Пошук вакансій та Авторизація представлено на рисунку 2.12 та 2.13.

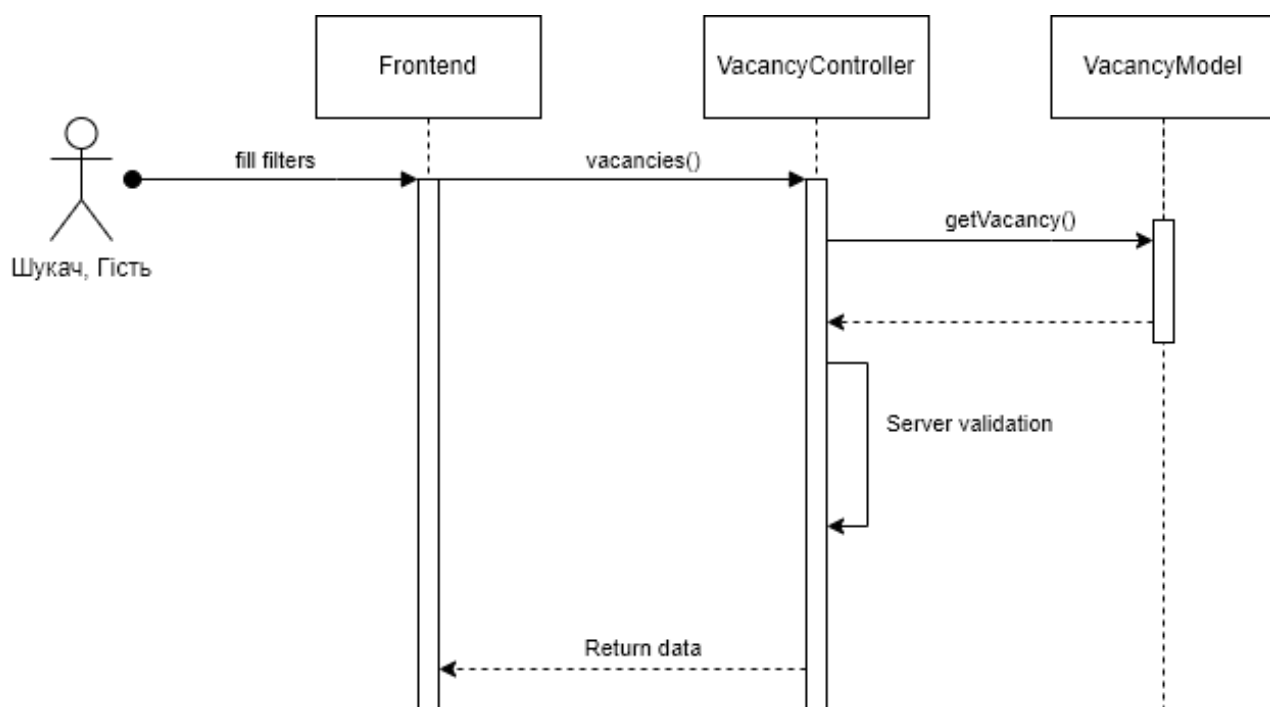


Рисунок 2.12 – Динамічна модель варіантів використання «Пошук вакансій»

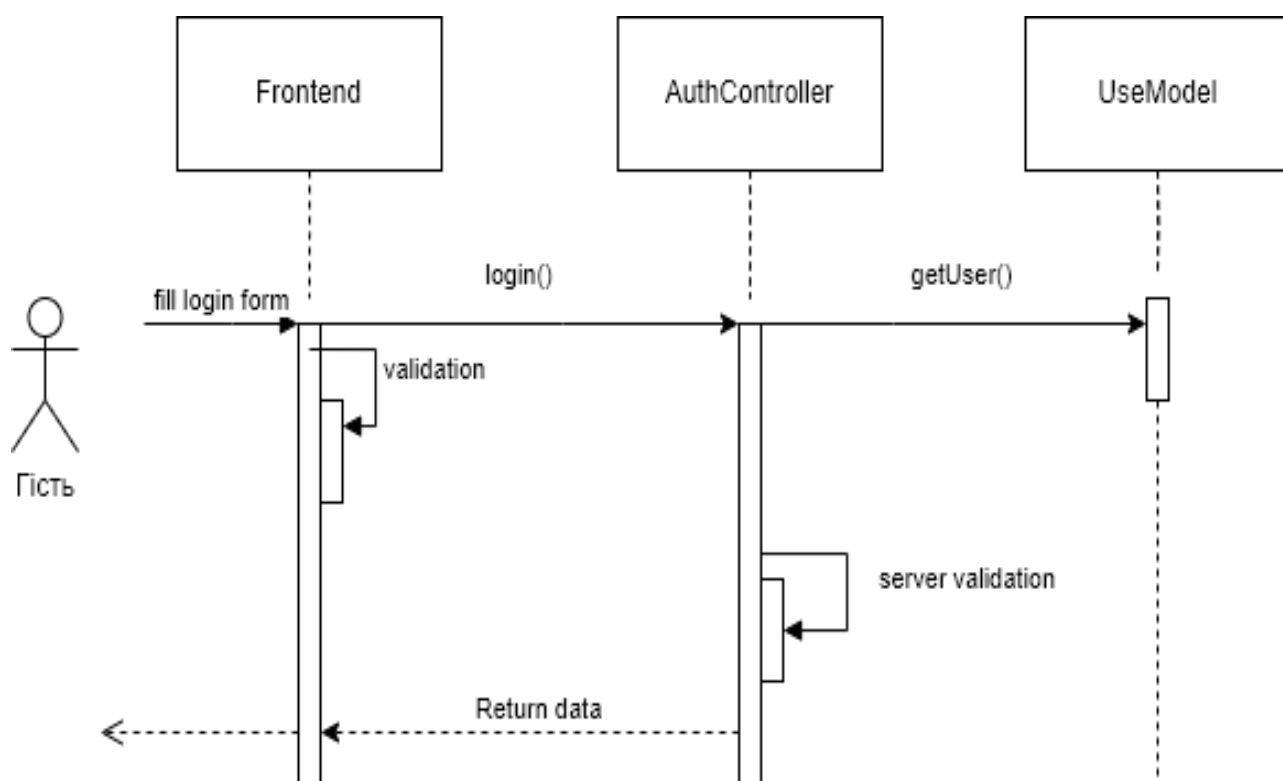


Рисунок 2.13 – Динамічна модель варіантів використання «Авторизація»

2.3.7 Логічна модель даних

Логічна модель даних – це модель даних конкретної предметної області, виражено незалежно від конкретного продукту керування базами даних або технології зберігання (фізична модель даних).

Логічну модель даних представлено на рисунку 2.14.

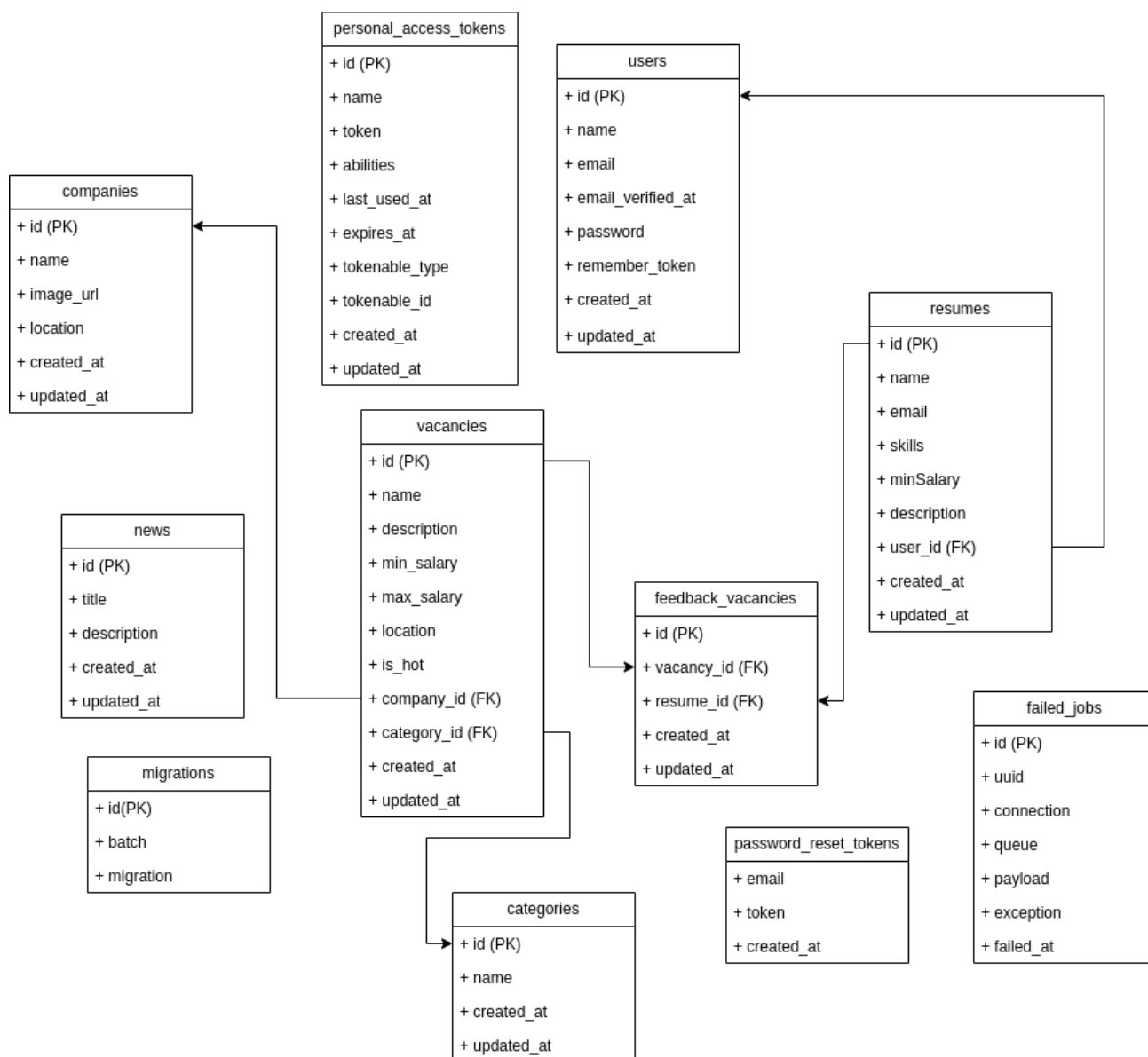


Рисунок 2.14 – Логічна модель бази даних веб-сервісу пошуку роботи «Fast-Work»

2.3.8 Робочий проект програмного забезпечення

Після аналізу вимог до програмного забезпечення, було вибрано повний набір інструментів для розробки веб-сервісу. Цей розділ присвячений опису переваг та недоліків вибраних мов програмування та СУБД. Проведено тестування та випробування програмного забезпечення.

2.3.9 Вибір та обґрунтування інструментів розробки.

Перед початком розробки програмного забезпечення необхідно вибрати мови програмування. У попередніх розділах було проаналізовано, що краще додаток розділити на дві частини: backend та frontend.

У якості backend частини було вибрано Laravel. Laravel – це відкритий фреймворк для веб-розробки на мові програмування PHP. Він заснований на принципах модульності та масштабованості, має зрозумілий синтаксис та широкий спектр функцій, що воляє швидко розробляти складні веб-додатки.

Laravel має вбудовану систему маршрутизації, ORM, шаблонізатор Blade, підтримує сесії та аутентифікацію користувачів, а також має велику кількість розширень та пакетів. Фреймворк пропонує конфігурування та налаштування за допомогою файлів конфігурації, що дозволяє з легкістю налаштувати проект під вимоги проекту. Laravel добре підходить для розробки веб-додатків будь-якої складності, від простого блогу до складних систем управління контентом або електронної комерції. Пропонує швидку розробку, стабільність та зручність у роботі для розробників.

Однією з головних переваг є його активна спільнота, яка постійно доповнює фреймворк новими функціями та пакетами. Фреймворк також має документацію високої якості, що дозволяє швидко засвоїти його та приступити до розробки веб-додатків. Також він реалізує шаблон MVC.

Сучасні веб додатки найчастіше використовують SPA. SPA означає «Single

Page Application» (односторінкова додаток). Це веб-додаток, який взаємодіє з користувачем, не перезавантажуючи сторінку. Всі необхідні ресурси, такі як HTML, CSS та JavaScript, завантажуються разом зі сторінкою при першому відкритті і потім використовуються динамічно за допомогою AJAX запитів до сервера. SPA надає користувачам більш інтерактивний та зручний інтерфейс, знижує час завантаження сторінок та робить додаток більш ефективним у використанні. Широко використовуються в сучасних веб-додатках та веб-сайтах, зокрема в онлайн-магазинах, соціальних мережах та інших веб-додатках з багатим функціоналом.

Найкращим фреймворком для реалізації веб інтерфейсу є Vue.js. Vue.js – це прогресивна JavaScript бібліотека для створення користувацьких інтерфейсів. Вона дозволяє легко та ефективно створювати одно сторінкові додатки (Single Page Applications) та інші складні інтерфейси на стороні клієнта.

Вона є дуже популярною для веб-розробки та надає можливість розробникам створювати динамічні та інтерактивні інтерфейси з використанням компонентного підходу. Бібліотека має легкий та інтуїтивно зрозумілий синтаксис, що дозволяє швидко вивчити її та розробляти з її допомогою як малі, так і складні додатки. Крім того, Vue.js має багатий екосистему, що складається з різноманітних доповнень та інструментів, що дозволяє зробити розробку веб-додатків ще більш простою та продуктивною.

Отже, вибрані технології сучасні, мають велику документацію, зарекомендували себе на практиці, що допоможе швидко та якісно програмне забезпечення веб-сервісу пошуку роботи «Fast-Work».

2.3.10 Фізична модель бази даних

Фізична модель бази даних визначає спосіб фізичного зберігання даних на комп'ютері. Вона описує структуру даних, способи їх збереження, індексування та взаємозв'язки між таблицями

Таблиця 2.2 – Структура таблиці News

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	title		VARCHAR(255)
3	description		LONGTEXT
4	created_at		TIMESTAMP
5	updated_at		TIMESTAMP

Таблиця 2.3 – Структура таблиці Companies

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	name		VARCHAR(255)
3	image_url		VARCHAR(255)
4	location		VARCHAR(255)
5	created_at		TIMESTAMP
6	updated_at		TIMESTAMP

Таблиця 2.4 – Структура таблиці Categories

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	name		VARCHAR(255)
3	created_at		TIMESTAMP
4	updated_at		TIMESTAMP

Таблиця 2.5 – Структура таблиці Vacancies

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	name		VARCHAR(255)
3	description		VARCHAR(255)
4	min_salary		double
5	max_salary		double
6	location		VARCHAR(255)
7	is_hot		tinyint(1)
8	company_id	FK	bigint(20) unsigned
9	category_id	FK	bigint(20) unsigned
10	created_at		TIMESTAMP
11	updated_at		TIMESTAMP

Таблиця 2.6 – Структура таблиці Resumes

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	email		VARCHAR(255)
3	name		VARCHAR(255)
4	skills		VARCHAR(255)
5	minSalary		Double
6	description		VARCHAR(255)
7	user_id		bigint(20) unsigned
8	created_at		TIMESTAMP
9	updated_at		TIMESTAMP

Таблиця 2.7 – Структура таблиці Feedback_vacancies

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	vacancy_id	FK	bigint(20) unsigned
3	resume_id	FK	bigint(20) unsigned
4	created_at		TIMESTAMP
5	updated_at		TIMESTAMP

Таблиця 2.8 – Структура таблиці Users

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	name		VARCHAR(255)
3	email		VARCHAR(255)
4	email_verified_at		TIMESTAMP
5	password		VARCHAR(255)
6	remember_token		VARCHAR(100)
7	created_at		TIMESTAMP
8	updated_at		TIMESTAMP

Таблиця 2.9 – Структура таблиці Password_reset_tokens

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	email		VARCHAR(255)
3	token		VARCHAR(255)
4	created_at		TIMESTAMP
5	updated_at		TIMESTAMP

Таблиця 2.10 – Структура таблиці Migrations

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	batch		int(11)
3	migration		VARCHAR(255)

Таблиця 2.11 – Структура таблиці Personal Access Tokens

Номер	Назва поля	Ключ	Тип Даних
1	id	PK	bigint(20) unsigned
2	name		VARCHAR(255)
3	token		VARCHAR(64)
4	abilities		TEXT
5	tokenable_type		VARCHAR(255)
6	tokenable_id		bigint(20) unsigned
7	last_used_at		TIMESTAMP
8	expires_at		TIMESTAMP
9	created_at		TIMESTAMP
10	updated_at		TIMESTAMP

2.3.11 Реалізація програмного забезпечення

Для зручної реалізації програмного забезпечення веб-сервісу пошуку роботи «Fast-Work» було використано інтегроване середовище розробки (IDE), яке призначене спеціально для роботи з мовами програмування PHP, HTML, CSS, JS. Один із головних плюсів це можливість покращувати продуктивність розробки на мові програмування PHP.

Під час реалізації програмного забезпечення було використано серверний фреймворк Laravel, щоб обробляти запити користувачів, взаємодіяти з базою даних та забезпечувати логіку додатку. Vue.js використано для побудови динамічного інтерфейсу користувача, який дозволяє швидко оновлювати інформацію на сторінці без перезавантаження.

2.3.12 Тестування програмного забезпечення

При тестуванні програмного забезпечення був вибраний метод чорної скриньки.

Метод чорної скриньки (Black Box Testing) – це підхід до тестування програмного забезпечення, при якому тестувальник не має доступу до внутрішньої структури, логіки та реалізації програми. Він базується на аналізі вхідних та вихідних даних програми, без знання про її внутрішній механізм роботи.

Мета проведення тестування це перевірка усіх функцій розробленого програмного забезпечення сервісу пошуку роботи «Fast-Work»

Тестування варіантів використання:

Перший варіантом використання, який буде протестований «Авторизація».

Вхідними даними є:

- Логін гостя (Електронна адреса)
- Пароль гостя

Всього було проведено 3 тести.

Тести:

- 1) Авторизація гостя з помилкою

Щоб увійти до особистого профілю необхідно на головній сторінці натиснути кнопку «Увійти». Очікуваним результатом повинно бути відкриття сторінки входу на вебсайт та при введенні неправильного паролю від особистого запису показати, що пароль не вірний. Результат зображений на рисунку 2.15

The image shows a login form titled "Fast Work" with a blue header. The form itself is titled "Увійти" (Login). It contains two input fields: "Електронна адреса" (Email address) with the value "o.ivanov@email.com" and "Пароль" (Password) with masked characters "*****". A red error message "Incorrect password" is displayed below the password field. A blue button labeled "УВІЙТИ" (Login) is at the bottom.

Рисунок 2.15 – Авторизація з помилкою

2) Авторизація гостя з помилкою

Якщо гість не заповнив обов'язкові поля для авторизації, то очікуваним результатом повинні бути помилки.

The image shows the same "Fast Work" login form, but with validation errors. The "Електронна адреса" field has a red error message "email field mustn't be empty". The "Пароль" field has a red error message "password is a required field". The blue "УВІЙТИ" button remains at the bottom.

Рисунок 2.16 – Авторизація з помилкою

3) Авторизація користувача з успішними даними

Якщо гість ввів коректні дані від особистого профілю, то повинна відкритися особиста сторінка з інформацією.

The screenshot shows the 'Fast Work' header in a blue bar. Below it, the text 'Ваше резюме: 😊' is centered. The resume form consists of several fields:

- Ім'я:** Nikita1
- Електронна адреса:** nikita@gmail.com
- Skills:** js sql
- Мінімальна зарплата:** 1212121
- Опис:** test

At the bottom of the form is a purple button labeled 'ОНОВИТИ'.

Рисунок 2.17 – Особиста сторінка шукача «Резюме»

The screenshot shows the 'Fast Work' header in a blue bar. Below it, the text 'Ваші відгуки: 😊' is centered. The page has two tabs: 'RESUME' and 'FEEDBACKS', with 'FEEDBACKS' being the active tab. The feedback section displays a list of reviews with the following details:

- Title:** Data Scientist
Company name: ПриватБанк
[ПЕРЕГЛЯНУТИ ВАКНСІЮ](#)
- Title:** Менеджер з продажу нерухомості
Company name: Рута
[ПЕРЕГЛЯНУТИ ВАКНСІЮ](#)
- Title:** Продавець-консультант в магазин Eva
Company name: Eva, лінія магазинів
[ПЕРЕГЛЯНУТИ ВАКНСІЮ](#)
- Title:** Системний адміністратор
Company name: ПриватБанк
[ПЕРЕГЛЯНУТИ ВАКНСІЮ](#)

At the bottom, a partially visible review for 'Майстер з ремонту автомобілів' is shown.

Рисунок 2.18 – Особиста сторінка шукача «Ваші відгуки»

В ході тестування блоку «Авторизація» помилок не було знайдено.

Другий варіантом використання , який буде протестований «Пошук вакансій»

Вхідними даними є:

- Назва вакансії;
- Категорії вакансії;
- Мінімальна та максимальна заробітна плата;
- Місто.

Всього було проведено 2 тести.

Тести:

1) Пошук вакансій без вибраних фільтрів за всіма містами. Очікуваним результатом є відображення списку всіх вакансій. Результат зображений на рисунку.

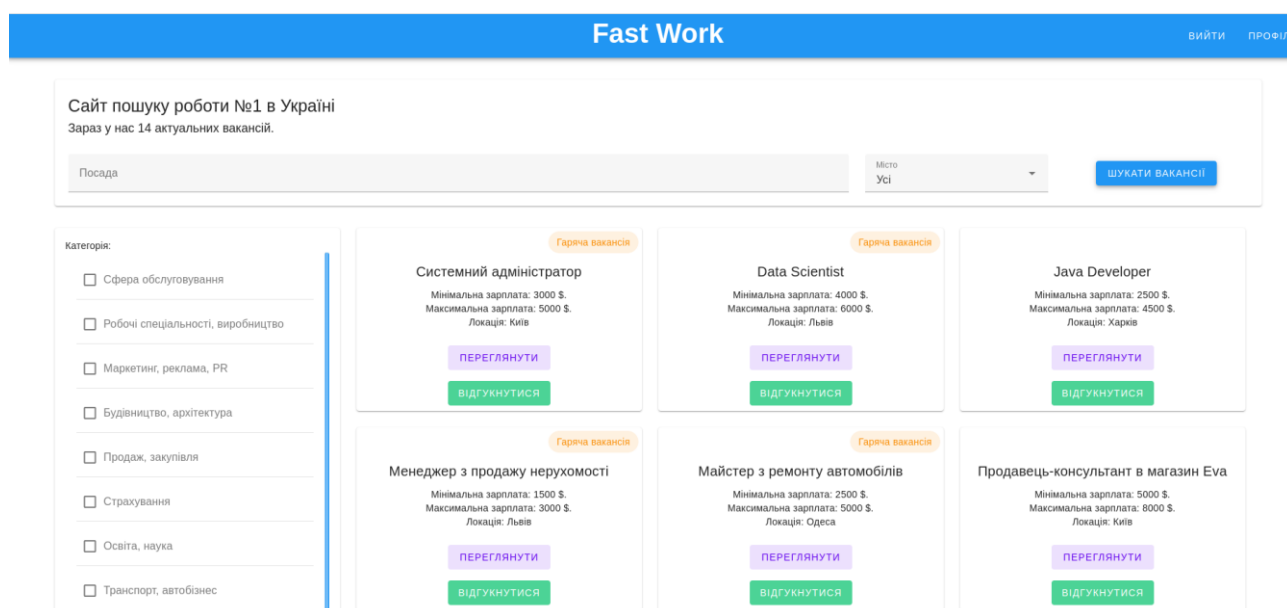


Рисунок 2.19 – Список вакансій

2) Пошук вакансій з вибраними фільтрами категорія та посада. Очікуваним результатом є відображення вакансій, які відповідають фільтром.

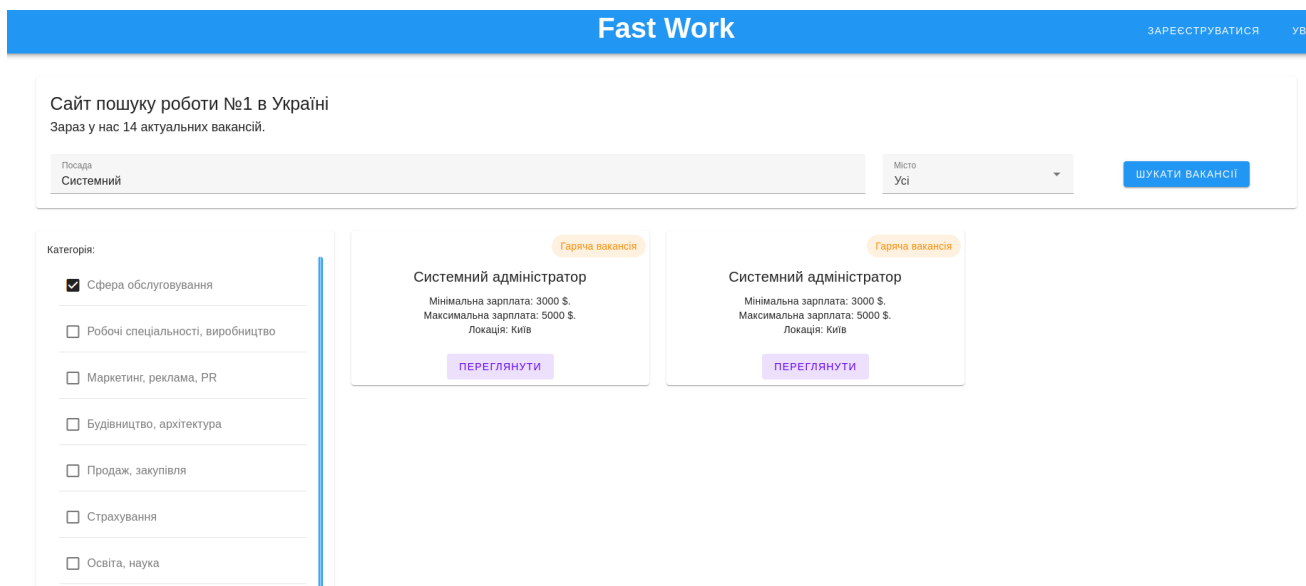


Рисунок 2.20 – Список вакансій з вибраними фільтрами

В ході тестування блоку «Пошук вакансій» помилок не було знайдено.

Третім варіантом використання є «Відгук шукача на вакансію».

Вхідними даними є:

- Вакансія;
- Резюме шукача

Всього було проведено 3 тести.

Тести:

- 1) Відгук на вакансію зі списку вакансій.

Для того щоб відгукнутися на вакансію необхідно спочатку відкрити сторінку з вакансіями. Очікуваним результатом повинен бути відгук на вакансію зі спливаючим повідомленням. Результат зображений на рисунку.

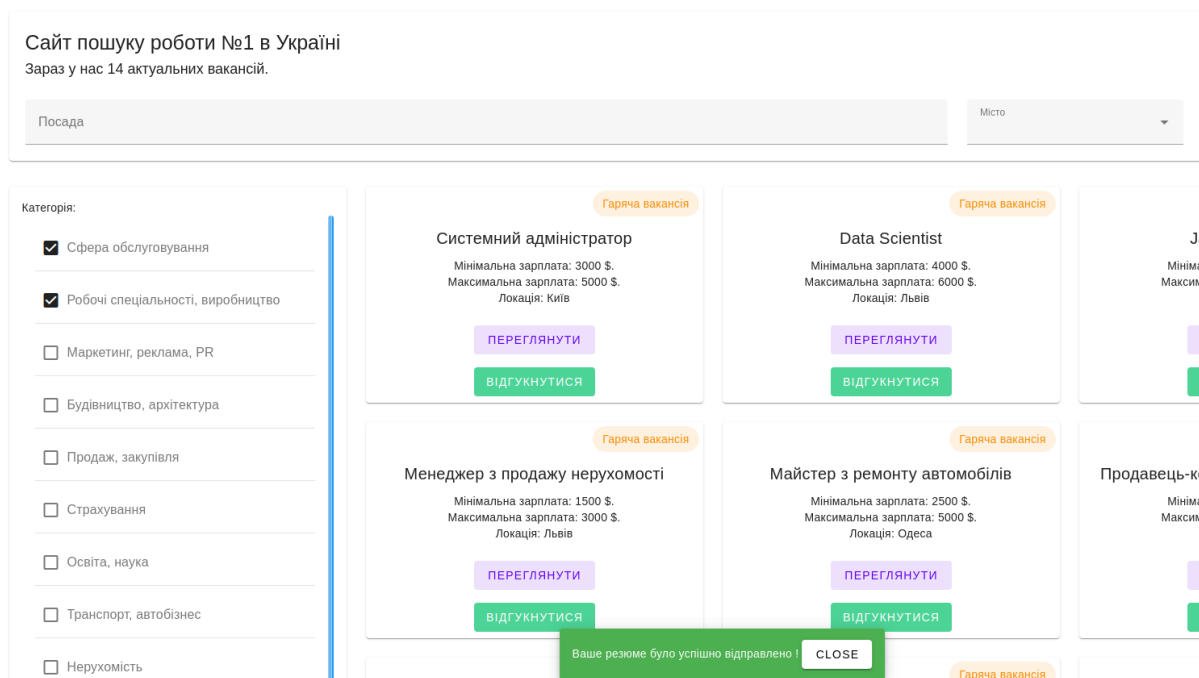


Рисунок 2.21 – Відгук на вакансію

2) Відгук на вакансію зі сторінки вакансії.

Для того щоб відгукнутися зі сторінки вакансії необхідно вибрати відповідну та натиснути кнопку «відгукнутися». Очікуваним результатом повинно бути спливаюче повідомлення на вакансію.

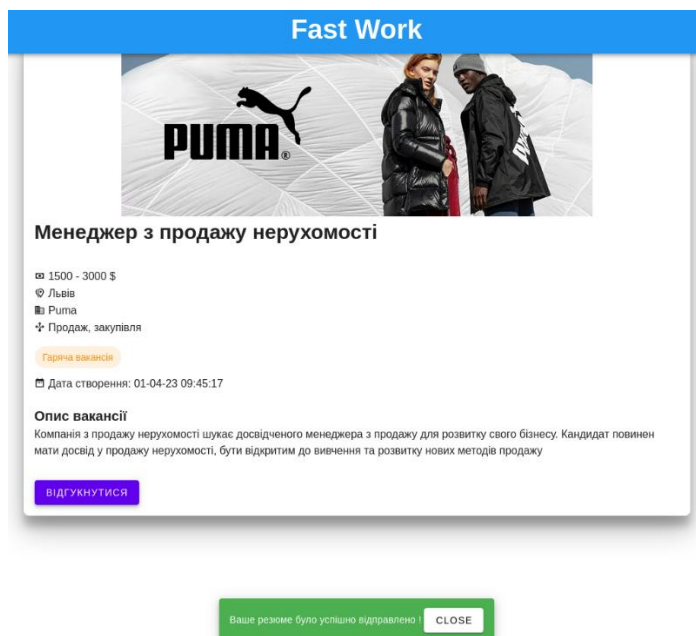


Рисунок 2.22 – Перегляд відповідної вакансії

Для того щоб переглянути свої відгуки на вакансії необхідно зайти на сторінку особистого профілю та натиснути на відгуки. Очікуваним результатом

повинно бути відображення всіх відгуків.

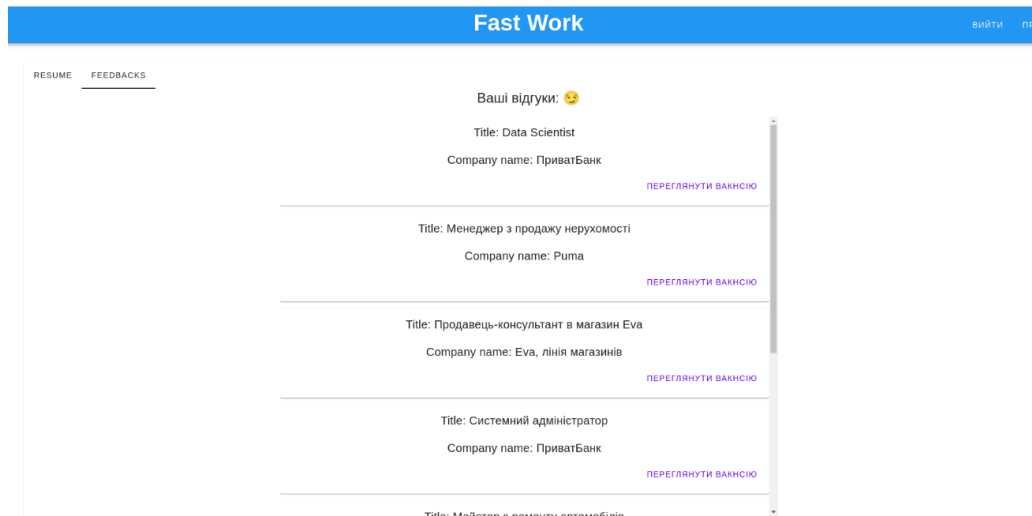


Рисунок 2.23 – Перегляд відгуків на вакансії

В ході тестування блоку «Відгук шукача на вакансію» помилок не було знайдено.

Отже, після тестування усіх варіантів використання не було знайдено жодних помилок.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ВЕБ-СЕРВІСУ ПОШУКУ РОБОТИ «FAST-WORK»

В результаті розробки було створено веб-сервіс пошуку роботи «Fast-Work». При розробці веб-сервісу було використано мову програмування PHP та JavaScript. Серверна частина реалізована за допомогою фреймворку Laravel, клієнтська частина на Vue.js. Було створено технічне завдання для веб-сервісу пошуку роботи «Fast-Work, яке знаходиться у Додатку А.

Частина тексту реалізованої програми представлено у Додатку Б.

Опис програми знаходиться у Додатку В.

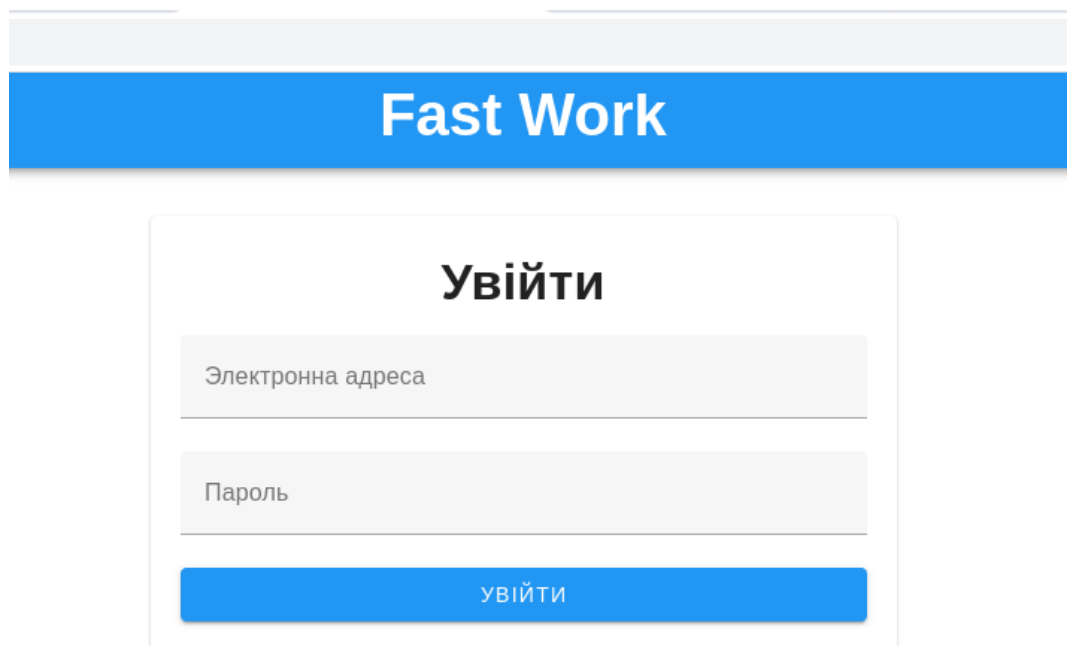
Інструкція користувача, яка описує як працювати з додатком знаходиться у Додатку Г.

Програма та методика випробувань знаходиться у Додатку Д.

Усі задачі які були поставлені виконані у повному обсязі, програмне забезпечення повністю функціонує.

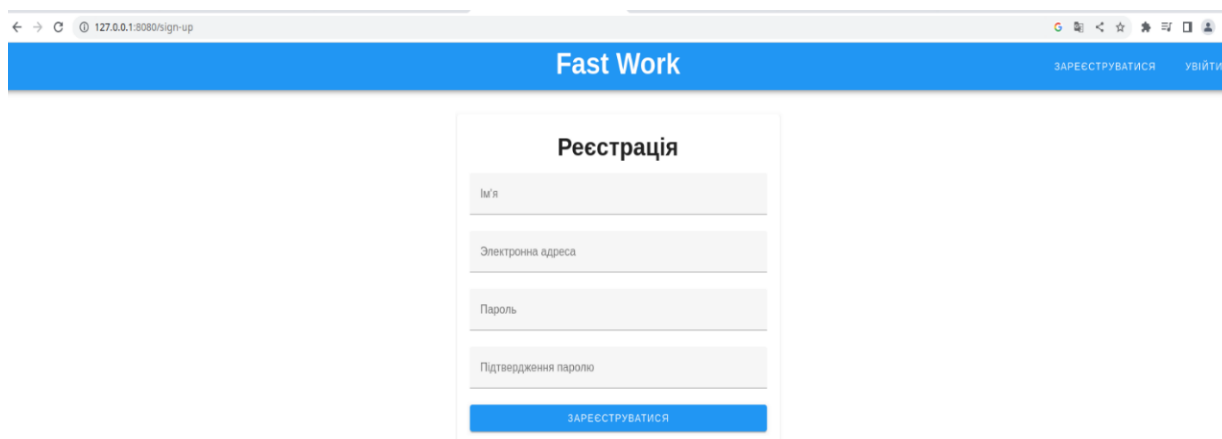
Результати розробки веб-сервісу пошуку роботи «Fast-Work» зображені на рисунках 3.1-3.7.

Щоб отримати доступ до усіх функцій необхідно зареєструватися та увійти до додатку. Процес входу та реєстрації було вирішено зробити окремими сторінками



The image shows a login form for a service called "Fast Work". At the top, there is a blue header bar with the text "Fast Work" in white. Below this, the form is centered and has a title "Увійти" (Login) in bold black text. There are two input fields: "Електронна адреса" (Email address) and "Пароль" (Password). Below these fields is a blue button with the text "УВІЙТИ" (Login) in white.

Рисунок 3.1 – Форма входу до веб-сервісу



The image shows a registration form for a service called "Fast Work". At the top, there is a blue header bar with the text "Fast Work" in white. To the right of the header, there are two links: "ЗАРЕЄСТРУВАТИСЯ" (Register) and "УВІЙТИ" (Login). Below the header, the form is centered and has a title "Реєстрація" (Registration) in bold black text. There are four input fields: "Ім'я" (Name), "Електронна адреса" (Email address), "Пароль" (Password), and "Підтвердження паролю" (Confirm password). Below these fields is a blue button with the text "ЗАРЕЄСТРУВАТИСЯ" (Register) in white.

Рисунок 3.2 – Форма реєстрації на веб-сервісі

Коли шукач зареєструється чи увійде до веб-сервісу його автоматично перенаправить на сторінку перегляду його особистого профілю для додавання особистого резюме. Процес заповнення резюме зображений на рисунку 3.3.

Ваше резюме: 😊

Ім'я Nikita Chopenko
Електронна адреса webnikitchopenko@gmail.com
Skills php, js, sql, docker
Мінімальна зарплата 1000
Опис my description

[ОНОВИТИ](#)

Рисунок 3.3 – Створення резюме

Головною функцією розробленого додатку є можливість шукати вакансії за допомогою фільтрів. Пошук доступних вакансій зображено на рисунку 3.4.

Fast Work Вийти Профіль

Сайт пошуку роботи №1 в Україні
Зараз у нас 14 актуальних вакансій.

Посада

Місто
Усі

[ШУКАТИ ВАКАНСІЇ](#)

Категорія:

- ☐ Сфера обслуговування
- ☐ Робочі спеціальності, виробництво
- ☐ Маркетинг, реклама, PR
- ☐ Будівництво, архітектура
- ☐ Продаж, закупівля
- ☐ Страхування
- ☐ Освіта, наука
- ☐ Транспорт, автобізнес
- ☐ Недружність

Гарна вакансія Системний адміністратор Мінімальна зарплата: 3000 \$. Максимальна зарплата: 5000 \$. Локація: Київ ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ	Гарна вакансія Data Scientist Мінімальна зарплата: 4000 \$. Максимальна зарплата: 6000 \$. Локація: Львів ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ	Гарна вакансія Java Developer Мінімальна зарплата: 2500 \$. Максимальна зарплата: 4500 \$. Локація: Харків ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ
Гарна вакансія Менеджер з продажу нерухомості Мінімальна зарплата: 1500 \$. Максимальна зарплата: 3000 \$. Локація: Львів ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ	Гарна вакансія Майстер з ремонту автомобілів Мінімальна зарплата: 2500 \$. Максимальна зарплата: 5000 \$. Локація: Одеса ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ	Гарна вакансія Продавець-консультант в магазин Eva Мінімальна зарплата: 5000 \$. Максимальна зарплата: 8000 \$. Локація: Київ ПЕРЕГЛЯНУТИ ВІДГУКНУТИСЯ

Рисунок 3.4 – Пошук вакансій

Шукач, коли йому сподобається деяка вакансія зможе відгукнутися на неї. Сторінки відгуків зображені на рисунках 3.5 – 3.6.

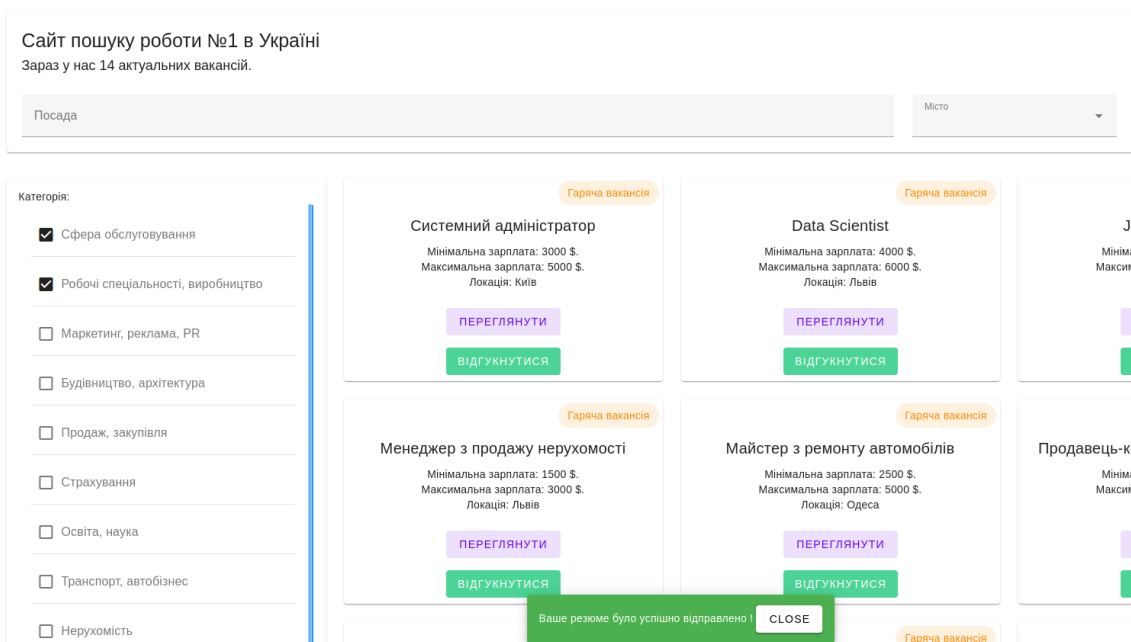


Рисунок 3.5 – Процес відгуку на вакансії

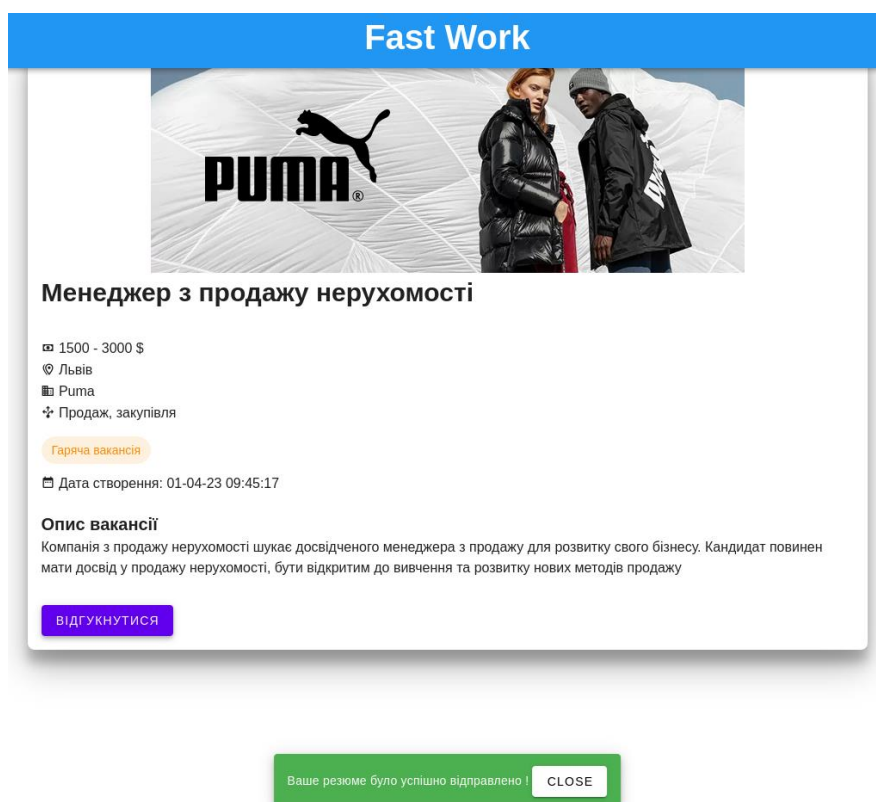


Рисунок 3.6 – Процес відгуків зі сторінки перегляду вакансій.

Користувач щоб дізнатися актуальні новини може перейти на сторінку їх перегляду.

Fast Work

Як легко знайти роботу в інтернеті?

Існує кілька способів знайти роботу в Інтернеті. Ось декілька корисних порад:

1. Використовуйте сайти пошуку роботи: Сайти, такі як LinkedIn, Indeed, Glassdoor, Monster і CareerBuilder, пропонують велику кількість вакансій, які ви можете переглянути та подати заявку. Використовуйте фільтри пошуку, щоб знайти роботу, яка відповідає вашим потребам та кваліфікації.
 2. Слідкуйте за соціальними медіа: Багато компаній розміщують вакансії на своїх сторінках в соціальних мережах, таких як Facebook, Twitter та LinkedIn. Ви можете підписатися на сторінки цих компаній та отримувати оновлення щодо вакансій.
 3. Відвідайте веб-сайти компаній: Багато компаній розміщують вакансії на своїх веб-сайтах. Якщо ви знаєте, в якій компанії ви хочете працювати, перевірте їх веб-сайт на наявність вакансій.
 4. Використовуйте пошукові системи: Google та інші пошукові системи можуть допомогти вам знайти вакансії в різних компаніях та розмістити резюме на спеціальних сайтах.
 5. Використовуйте мережі для пошуку роботи: Існують мережі для пошуку роботи, такі як ZipRecruiter, які допомагають знайти роботу, яка відповідає вашим потребам та кваліфікації.
- Не забувайте, що знайти роботу може зайняти час, іноді довго. Вам може знадобитися зробити кілька спроб, перш ніж знайти ту роботу, яка вам підходить. Тримайтеся та не впадайте у відчай.

Рисунок 3.7— Сторінка перегляду новин

4 ОХОРОНА ПРАЦІ

4.1 Основні положення

Охорона праці – як галузь людської діяльності – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини у процесі її трудової діяльності.

Основною метою охорони праці є створення безпечних умов трудової діяльності людини, забезпечення її високої та ефективної працездатності.

Охорона праці як соціально-технічна дисципліна вивчає теоретичні та практичні питання безпеки праці, запобігання виробничому травматизму, професійним захворюванням і отруєнням, аваріям (катастрофам), пожежам і вибухам на виробництві. Вона вивчається з метою формування у майбутніх фахівців необхідного рівня знань та умінь з правових й організаційних питань охорони та гігієни праці, виробничої санітарії, техніки безпеки, а також активної позиції щодо практичної реалізації головного принципу Конституції України – пріоритетності охорони життя та здоров'я працівників відносно результатів виробничої діяльності [10].

Роботодавець – власник підприємства, установи, організації або уповноважений ним орган, незалежно від форм власності, виду діяльності, господарювання, і фізична особа, яка використовує найману працю [10].

Працівник – особа, яка працює на підприємстві, в організації, установі та виконує обов'язки або функції згідно з трудовим договором (контрактом) [10].

Усі права працівника на безпечні умови праці захищаються законом України “Про охорону праці” [10].

4.2 Аналіз шкідливих та небезпечних факторів на робочому місці

Комп'ютерна техніка у наш час знайшла широке використання у діловодстві, промисловості, науці, навчальному процесі, але при недотриманні вимог безпеки, вона може завдавати значної шкоди працюючим.

Особливістю негативного впливу комп'ютерних технологій на працездатність і здоров'я людини є комплексна одночасна дія декількох шкідливих факторів, при значній інтенсивності яких відбувається накопичення і акумулювання їх впливу, що викликає суттєві зміни в організмі людини, розлад функцій окремих органів і систем.

До основних негативних факторів належать: випромінювання різних діапазонів електромагнітного спектру (рентгенівське та оптичне випромінювання). Робота комп'ютера призводить до зміни фонові концентрації іонів повітря. Так, приблизно через 5 хвилин роботи монітору концентрація легких негативних іонів знижується в 5–10 разів (фонове значення цього показнику становить 350–620 іонів/см³), а через 3 години роботи їх концентрація наближається до нуля.

Знижується також концентрація середніх і тяжких негативних іонів, натомість концентрація позитивного заряджених іонів різко зростає, що дуже негативно відбивається на газообміні в легенях, загальному почутті людини.

Значна кількість позитивних іонів, особливо тяжких, призводить до підвищення артеріального тиску, тахікардії, прояву болю в області серця, затрудненню дихання, прискоренню швидкості осідання еритроцитів, розладу функцій центральної нервової системи (дратівливість, головний біль, порушення сну, тонусу м'язів і інше), порушення травлення. Оптимальним рівнем аероіонізації на робочому місці рахується вміст легких іонів від 150 до 5000 в 1 см³, повітря.

Нормалізувати іонний склад повітря виробничої зони можна різними способами: механічна вентиляція, застосування іонізаторів, заземлених захисних екранів тощо.

Доза рентгенівського випромінювання перед екраном монітора на відстані 50 см від його поверхні є безпечною, вона не досягає межі допустимого рівня (50 мкР/год), але не вивчена дія цих променів у поєднанні з іншими, які генеруються комп'ютером на людину, що не дозволяє говорити відносно безпечної їх дії.

До психологічних і поведінкових розладів відносяться: агресивність, фрустрація, нервозність, дратівливість, порушення сну, швидкий розвиток втоми тощо.

Зміни на соматичному, фізіологічному, психологічному і поведінковому рівнях працівників на комп'ютерах пов'язані з високим навантаженням при виконанні завдань, високою емоційально-психологічною напруженістю та дією негативних виробничих факторів [11].

4.3 Заходи щодо зменшення впливу шкідливих факторів роботи за ПК

Організація робочого місця користувача монітору повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам, характеру та особливостям трудової діяльності. Площа одного робочого місця повинна бути не менше 6 м², а обсяг – не менше 20 м³.

При розміщенні робочих місць необхідно дотримуватись таких вимог:

- природне світло повинно падати збоку, переважно зліва;
- відстань від робочого місця до стін зі світловим прорізами повинна складати не менше 1 м;
- відстань між бічними поверхнями монітора має бути не меншою за 1,2 м;
- іншого не повинна бути меншою 2,5 м, а прохід між рядами робочих місць – не меншим одного метра.

Висота робочої поверхні столу для монітору має бути в межах 68-80 см, а ширина повинна забезпечувати можливість використання операцій у зоні

досяжності моторного поля (рекомендовані розміри столу: висота — 72,5 см, ширина — 60-140 см, глибина — 80-100 см).

Робоче сидіння (сидіння, стілець, крісло) працівника на обчислювальній техніці повинно бути підйомно-поворотним, плоским, спереду закругленим, а для усунення статичного напруження м'язів рук улаштоване стаціонарними або змінними підлокітниками.

Монітор та клавіатура мають розташовуватися на оптимальній відстані від очей працівника, але не ближче 60 см, з урахуванням розміру алфавітно-цифрових знаків та символів.

При потребі високої концентрації уваги під час виконання робіт з високим рівнем напруженості суміщені робочі місця з комп'ютером необхідно відділяти одне від одного перегородками висотою 1,5-2 м.

Організація робочого місця для комп'ютера:

- достатній простір для людини-оператора;
- вільну досяжність органів ручного управління в зоні моторного поля: відстань по висоті — до 133 см, по глибині — 40-50 см;
- розташування екрана монітора в робочій зоні, яке забезпечувало б зручність зорового спостереження у вертикальній площині під кутом плюс-мінус 30° від лінії зору оператора;
- можливість повертання екрана монітора навколо горизонтальної та вертикальної осі [11].

ВИСНОВКИ

У кваліфікаційній роботі було успішно вирішено задачу інженерії програмного забезпечення – створення веб-сервісу пошуку роботи «Fast-Work».

Під час розробки програмного забезпечення було проаналізовано існуючі додатки для пошуку роботи. Було виявлено їх недоліки та переваги. Для створення веб-сервісу використовувалися сучасні об'єктно-орієнтовані мови програмування PHP та JavaScript. Середовищем для розробки було вибрано JetBrains PhpStorm. Реалізоване програмне забезпечення складається з frontend та backend частини. В ході реалізації веб-сервісу було використано сучасні технології, для backend – Laravel, для frontend – Vue.js. Ці інструменти дають можливість створювати потужні сервіси та легко масштабуватися у майбутньому при необхідності.

В ході проектування веб-сервісу пошуку роботи «Fast-Work» було створено діаграми, які показують взаємодію між компонентами програмного забезпечення.

Після розробки веб-сервісу frontend та backend успішно пройшли випробування на помилки. Веб-сервіс пошуку роботи «Fast-Work» повністю відповідає функціональним вимогам та поставленому завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке веб-сервіси: поняття, принципи роботи, переваги і недоліки. URL: <https://what.com.ua/sho-take-veb-servisi-poniatti/2/> (дата звернення 05.05.2023)
2. robota.ua – працюємо на перемогу! URL: <https://rabota.ua/ua> (дата звернення 10.05.23)
3. Сайт сервісу пошуку роботи work.ua. URL: <https://www.work.ua/> (дата звернення 05.05.23)
4. Про сучасні технології для backend розробки. URL: <https://highload.today/category/back-end/> (дата звернення 10.04.23)
5. Про Agile та Waterfall URL: <https://worksection.com/ua/blog/waterfall-vs-agile.html> (дата звернення 15.05.2023)
6. Розробка з боку frontend – що це таке і чим відрізняється від backend? URL: <https://dan-it.com.ua/uk/blog/rozrobka-z-boku-front-end-shho-ce-take-i-chim-vidriznjaietsja-vid-back-end/> (дата звернення 20.05.2023)
7. Про архітектуру програмного забезпечення URL: <https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya> (дата звернення 16.05.2023)
8. Про діаграму компонентів. URL: https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20151203140326/204841/index.html (дата звернення 18.05.2023)
9. Діаграма діяльності. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-78-diahrama-dijalnosti.pdf> (дата звернення 19.05.2023)
10. Про охорону праці: Закон України від 21.11.2002 р. № 229-IV. Дата оновлення: 04.02.2021. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text> (дата звернення: 20.05.2023)
11. Про охорону праці. URL: https://library.udpu.edu.ua/library_files/423594.pdf (дата звернення 20.05.2023)

ДОДАТОК А ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва розроблюваного проекту: «Веб-сервіс пошуку роботи Fast-Work».
Галузь застосування – рекрутинг, працевлаштування, платформа з пошуку роботи.

1 Підстава для розробки

Розробка програмного забезпечення виконується на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

2 Призначення розробки

2.1 Експлуатаційне призначення

Експлуатаційне призначення веб-сервісу пошуку роботи полягає у забезпеченні функціонування та наданні послуг працівникам і роботодавцям.

2.2 Функціональне призначення

Функціональним призначенням веб-сервісу пошуку роботи полягає в можливості автоматизованого пошуку роботи для працівників, а також розміщення вакансій компаніями.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

Програмне забезпечення повинно виконувати наступні функції:

Адміністратор:

- Реєстрація та авторизація;
- Додавання та редагування вакансій;
- Видалення вакансій;
- Додавання категорій;

- Додавання новин;
- Вихід з особистого кабінету.

Шукач:

- Створення та публікація резюме;
- Перегляд доступних вакансій;
- Відгук на вакансію;
- Перегляд вакансій відгуку;
- Вихід з особистого кабінету;
- Перегляд новин.

3.1.2 Вимоги до організації вхідних та вихідних даних

Дані зберігаються в файлі БД з довільним доступом типу .sql.

Введення даних здійснюється за допомогою пристроїв введення даних (клавіатури та миші). Дані або вводяться вручну, або обираються із представлених списків.

Виведення даних здійснюється за допомогою пристроїв виведення даних (монітор та принтер).

3.2 Вимоги до надійності

Програмне забезпечення повинно бути доступним у будь-який час. Сервіс повинен забезпечувати надійний захист персональних даних користувачів.

Програмне забезпечення повинно бути здатним ефективно працювати зі зростанням числа користувачів та обсягу даних.

Розроблений сервіс повинен мати системи моніторингу та виявлення проблем, які допомагають вчасно виявляти та усувати несправності.

3.3 Вимоги до умов експлуатації

Для використання програмного забезпечення необхідно мати базові навички роботи з комп'ютером. Веб-сервіс повинен бути розміщений на надійному

швидкому хостингу, забезпечувати достатній рівень пропускної здатності та ресурсів для виконання запитів користувачів.

3.4 Вимоги до складу та параметрів технічних засобів

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплектації:

- CPU: Intel(R) Celeron(R) – 2.16 МГц;
- RAM: 2 GB;
- HDD: 20 GB вільного місця;
- Пропускна здатність 100мб/с.

3.5 Вимоги до інформаційної та програмної сумісності

Для функціонування програмного забезпечення необхідно використовувати операційну систему Windows 7 не нижче 7.

Необхідна наявність встановленого PHP 8.0.

3.6 Вимоги до маркування та пакування

Програмне забезпечення буде складатися з двох електронних архівів та мати наступні ім'я backend.zip, frontend.zip.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- текст програми;
- опис програми;
- інструкцію користувача;
- програму і методику випробувань ПЗ.

5 Стадії та етапи розробки

Стадії та етапи розробки представлені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки програмного забезпечення

Технічне завдання	Обумовлення необхідності розробки	Постановка задачі. Збір початкових матеріалів. Вибір і обумовлення критеріїв ефективності та якості розроблюваного ПЗ	24.03.2023 -31.03.2022
	Розробка та затвердження технічного завдання	Визначення вимог до ПЗ. Узгодження і затвердження технічного завдання.	01.04.2023 -07.04.2022
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних. Уточнення методів рішення задачі. Розробка загального опису алгоритму розв'язання задачі	08.04.2023 -14.04.2023
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту	10.04.2023 -15.04.2023
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних. Розробка алгоритму рішення задачі. Розробка структури ПЗ	20.04.2023 -01.05.2023
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту	02.05.2023 -06.05.2023
Робочий проект	Розробка ПЗ	Програмування та налагодження програми	06.05.2023 -12.05.2023
	Розробка програмної документації	Розробка програмної документації відповідно до вимог ГОСТ 19.101-77	12.05.2022 -26.05.2022
	Випробування ПЗ	Розробка, узгодження та затвердження порядку і методики випробувань. Проведення попередніх випробувань. Коригування ПЗ і програмної документації за результатами випробувань.	26.05.2023- 29.05.2023

6 Порядок контролю та прийому

Кожна частина програмного забезпечення проходить контроль та оцінку на кожному етапі з урахування вимог, які визначені в технічному завданні. Це забезпечує відповідність функціональності та архітектури програми встановленим критеріям. Для кожної стадії розробки програмного забезпечення повинні бути створені звіти, документація та інші матеріали, які описують хід робіт та

досягнуті результати.

Прийом програмного виробу відбувається згідно з запланованою програмою та методикою випробувань. Результати прийому документуються за допомогою протоколу проведення випробувань. У разі виявлення помилок під час прийому складається акт про знайдені помилки, який підписуються представниками замовника та розробника і затверджується керівником організації-замовника та організації-розробника. Після виправлення помилок повторно проводиться перевірка, і замовник повідомляє про готовність продукту.

ДОДАТОК Б КОД ПРОГРАМИ

Нижче наведені файли моделей backend:

Category.php

```
<?php

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use CrudTrait;
    use HasFactory;

    protected $fillable = [
        'name'
    ];

    public function vacancies()
    {
        return $this->hasMany(Vacancy::class);
    }
}
```

Company.php

```
<?php

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Company extends Model
{
    use CrudTrait;
    use HasFactory;
```



```

protected $fillable = [
    'name',
    'image_url',
    'location'
];

public function vacancies()
{
    return $this->hasMany(Vacancy::class);
}
}

FeedbackVacancy.php
<?php

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class FeedbackVacancy extends Model
{
    use CrudTrait;
    use HasFactory;

    protected $with = [
        'vacancy',
        'resume'
    ];

    protected $fillable = [
        'vacancy_id',
        'resume_id'
    ];

    public function vacancy()
    {
        return $this->belongsTo(Vacancy::class);
    }

    public function resume()
    {
        return $this->belongsTo(Resume::class);
    }
}

News.php

<?php

```

```

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class News extends Model
{
    use CrudTrait;
    use HasFactory;

    protected $fillable = [
        'title',
        'description'
    ];
}

```

Resume.php

```
<?php
```

```

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Resume extends Model
{
    use CrudTrait;
    use HasFactory;

    protected $fillable = [
        'name',
        'email',
        'description',
        'skills',
        'minSalary',
        'user_id'
    ];

    public function feedbackResume() {
        return $this->hasMany(FeedbackVacancy::class);
    }
}

```

User.php

```
<?php
```

```

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;

```

```

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use CrudTrait;
    use HasApiTokens, HasFactory, Notifiable;

    protected $with = [
        'resume'
    ];

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    public function resume() {
        return $this->hasOne(Resume::class);
    }
}

```

Vacancy.php

```

<?php

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Builder;

class Vacancy extends Model
{
    use CrudTrait;
    use HasFactory;

    protected $fillable = [
        'name',
        'description',
        'min_salary',
        'max_salary',
        'location',
        'is_hot',
        'category_id',
        'company_id'
    ];

    public $casts = [
        'is_hot' => 'boolean'
    ];

    public function category()
    {
        return $this->belongsTo(Category::class);
    }

    public function company()
    {
        return $this->belongsTo(Company::class);
    }

    public function scopeSalaryBetween(Builder $query, $start, $end): Builder
    {
        return $query->where('min_salary', '>=', $start)
            ->where('max_salary', '<=', $end);
    }

    public function scopeSalaryStart(Builder $query, $start): Builder
    {
        return $query->where('min_salary', '>=', $start);
    }

    public function scopeSalaryEnd(Builder $query, $end) : Builder
    {

```

```

        return $query->where('max_salary', '<=', $send);
    }

    public function feedbackVacancy()
    {
        return $this->hasMany(FeedbackVacancy::class);
    }
}

```

Нижче наведені файли контролерів backend:

AuthController.php

```
<?php
```

```

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Http\Requests\SignInRequest;
use App\Http\Requests\SignUpRequest;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class AuthController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth:sanctum')->except(['register', 'login']);
    }

    public function register(SignUpRequest $request)
    {
        $user = $request->validated();

        $user['password'] = Hash::make($user['password']);

        return User::create($user);
    }

    public function login(SignInRequest $request)
    {
        $user = $request->validated();

        $resultUser = User::where('email', $user['email'])->first();

        if (Auth::attempt($user)) {
            return $request->user();
        } else {
            return response()->json(['message' => 'Unauthorized'], 400);
        }
    }
}

```

```

    public function me()
    {
        return Auth::user();
    }

    public function logout()
    {
        Auth::guard('web')->logout();
    }
}

```

CategoryController.php

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
use App\Models\Category;
```

```
class CategoryController extends Controller
{
    public function index()
    {
        return Category::all();
    }
}

```

CompanyController.php

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
use App\Models\Company;
```

```
class CompanyController extends Controller
{
    public function index()
    {
        return Company::all();
    }

    public function get($id)
    {
        return Company::where('id', $id)->firstOrFail();
    }
}

```

NewsController.php

```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\News;

class NewsController extends Controller
{
    public function index()
    {
        return News::all();
    }

    public function get($id)
    {
        return News::where('id', $id)->first();
    }
}
```

ResumeController.php

```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Http\Requests\ResumeRequest;
use App\Http\Requests\UpdateResumeRequest;
use App\Models\Resume;
use Illuminate\Support\Facades\Auth;

class ResumeController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth:sanctum');
    }

    public function store(ResumeRequest $request)
    {
        $resume = Resume::where('user_id', Auth::user()->id)->first();

        if($resume) {
            return response()->json(['message' => 'Your resume has been already created!'], 400);
        }

        return Resume::create(array_merge($request->validated(), ['user_id' => Auth::user()->id]));
    }

    public function update(UpdateResumeRequest $request)
    {

```

```

        $updatedResume = $request->validated();

        $resume = Resume::where('user_id', Auth::user()->id)->firstOrFail();

        $resume->update($updatedResume);

        $resume = Resume::where('user_id', Auth::user()->id)->first();

        return response()->json($resume);
    }

    public function index()
    {
        $resume = Resume::where('user_id', Auth::user()->id)->first();

        return [
            'resume' => $resume??null
        ];
    }
}

```

StatisticController.php

```

<?php

namespace App\Http\Controllers\Api;

use App\Models\Vacancy;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class StatisticController extends Controller
{
    public function get()
    {
        return [
            'countVacancies' => Vacancy::all()->count(),
            'averageSalary' => (int) Vacancy::avg('max_salary')
        ];
    }
}

```

VacancyController.php

```

<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Http\Requests\ResumeFeedbackRequest;
use App\Models\FeedbackVacancy;

```



```

use App\Models\Resume;
use App\Models\Vacancy;
use Illuminate\Support\Facades\Auth;
use Spatie\QueryBuilder\AllowedFilter;
use Spatie\QueryBuilder\QueryBuilder;

class VacancyController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth:sanctum')->except(['index', 'get', 'locations']);
    }

    public function index()
    {
        return QueryBuilder::for(Vacancy::class)
            ->allowedFilters([
                'name',
                AllowedFilter::exact('location')->ignore('Yci'),
                AllowedFilter::exact('category_id'),
                AllowedFilter::exact('is_hot'),
                AllowedFilter::scope('salary_between'),
                AllowedFilter::scope('salary_start'),
                AllowedFilter::scope('salary_end'),
            ])
            ->with(['category', 'company'])
            ->get();
    }

    public function get($id)
    {
        return Vacancy::where('id', $id)->with(['category', 'company'])->firstOrFail();
    }

    public function locations()
    {
        return Vacancy::select('location')
            ->orderBy('location', 'asc')
            ->distinct()
            ->get()
            ->map(function ($value) {
                return $value->location;
            })->push("Yci");
    }

    public function feedback(ResumeFeedbackRequest $request)
    {
        $feedback = $request->validated();

        Resume::where([
            'id' => $feedback['resume_id'],

```

```

        'user_id' => Auth::user()->id
    ]->firstOrFail();

    $feedbackVacancy = FeedbackVacancy::where([
        'resume_id' => $feedback['resume_id'],
        'vacancy_id' => $feedback['vacancy_id']
    ]->first();

    if(!$feedbackVacancy) {
        FeedbackVacancy::create($feedback);
    }

    return FeedbackVacancy::where('id', $feedbackVacancy->id)->first();
}

public function myFeedbacks()
{
    $resume = Auth::user()->resume ? Auth::user()->resume->id: null;

    if($resume) {
        return FeedbackVacancy::where('resume_id', $resume)->with('vacancy', 'vacancy.company')-
>get();
    }

    return [];
}
}

```

Нижче наведені файли компонентів для frontend

Pages/Home
vCategories.vue

```

<template>
    <v-row class="mt-7">
        <v-col sm="4" v-for="category in categories" :key="category.title">
            <vCategoryCard :category="category" />
        </v-col>
    </v-row>
</template>

<script>
import VcategoryCard from "@pages/home/vCategoryCard.vue";

export default {
    components: {VcategoryCard},
    props:["categories"],
    name: "vCategories"
}
</script>

<style scoped lang="scss">
</style>

```

ДОДАТОК В ОПИС ПРОГРАМИ

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: Програмне забезпечення веб-сервісу пошуку роботи «Fast-Work».

1.2 Програмне забезпечення, необхідне для функціонування програми

Програмне забезпечення складається з серверної(backend) та клієнтської(frontend) частини.

1.2.1 Серверна частина

Для оптимальної роботи серверної частини система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплектації:

- CPU: Intel(R) Celeron(R) – 2.16 МГц;
- RAM: 4 GB;
- HDD: 20 GB вільного місця;

1.2.2 Клієнтська частина

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплектації:

- CPU: Intel(R) Celeron(R) – 2.16 МГц;
- RAM: 2 GB;
- HDD: 20 GB вільного місця;

При розробці програмного забезпечення, для серверної частини (backend) було використано фреймворк Laravel, реалізований на мові програмування PHP та

СУБД MYSQL. Для реалізації клієнтської частини було використано Vue.js.

2 Функціональне призначення

2.1 Класи розв’язуваних завдань та призначення програми

Призначенням розроблюваного веб-сервісу є полегшення процесу пошуку роботи.

Програмне забезпечення повинно реалізовувати такі функції:

- Авторизація;
- Реєстрація;
- Пошук вакансій;
- Додавання резюме;
- Відгук на вакансію;
- Перегляд новин;
- Додавання новин;
- Додавання вакансій;
- Вихід з особистого кабінету.

3 Опис логічної структури

3.1 Використовувані методи

При розробці програмного забезпечення веб-сервісу пошуку роботи «Fast-Work» було використано об’єктно-орієнтовані методи програмування.

3.2 Алгоритм програми

Алгоритм роботи програми:

- Клієнт ініціює запит до сервера, відправляючи HTTP запит. Запит містить метод (GET, POST, PUT, DELETE тощо) та URL-адресу сервера, до якого клієнт хоче отримати доступ;
- Сервер отримує запит від клієнта і аналізує його;
- Сервер виконує необхідні операції для обробки запиту;
- Якщо необхідно, сервер може відправити запит до бази даних або інших зовнішніх сервісів для отримання додаткової інформації або виконання дій;
- Сервер формує відповідь на запит;
- Сервер відправляє відповідь клієнту, який отримує її і аналізує.

3.3 Зв'язок програми з іншими програмами

Розроблене програмне забезпечення взаємодіє з базою даних MySQL.

4 Виклик та загрузка програмного забезпечення

4.1 Спосіб виклику програми

Веб-сервіс пошуку роботи «Fast-Work» викликається після будь-якого HTTP запиту, проксі-сервер NGINX викликає php скрипти за допомогою php-fpm.

4.2 Вхідні точки програми

Вхідними точками до програмного забезпечення (backend) є routes. Вони дозволяють створювати маршрути через які можна в майбутньому відправляти запити.

ДОДАТОК Г ІНСТРУКЦІЯ КОРИСТУВАЧА

Щоб почати працювати з програмним забезпечення необхідно перейти до головної сторінки веб-сервісу.

Користувач побачить головну сторінку веб-сервісу, зображену на рисунках Г.1 - Г.2.

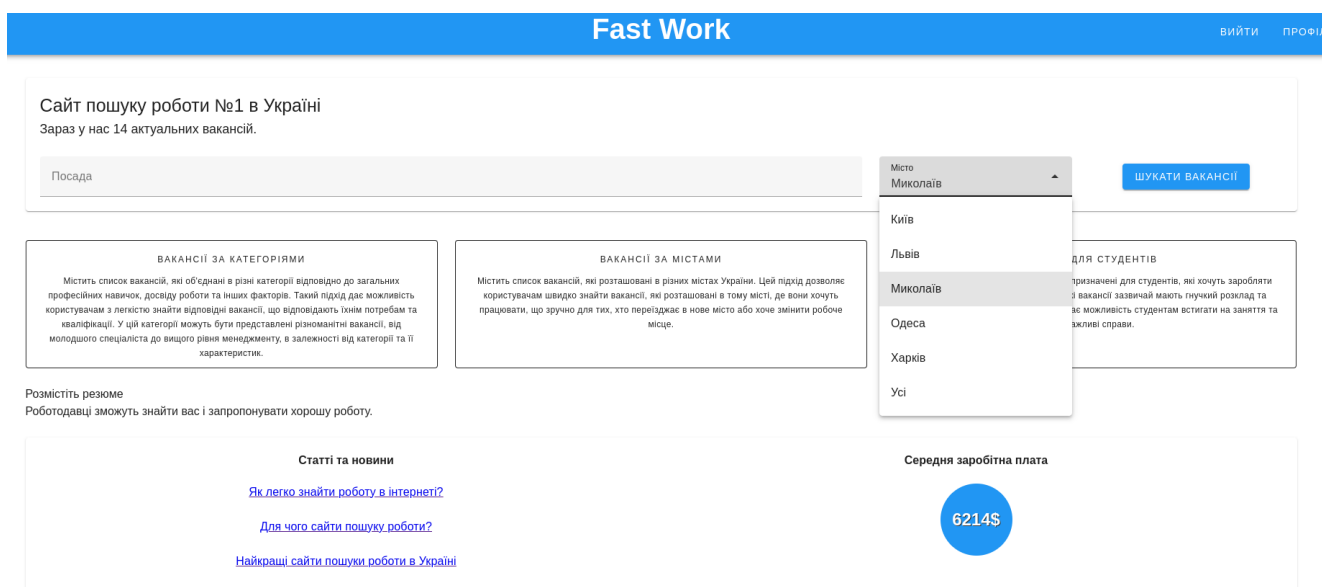


Рисунок Г.1 – Головна сторінка веб-сервісу

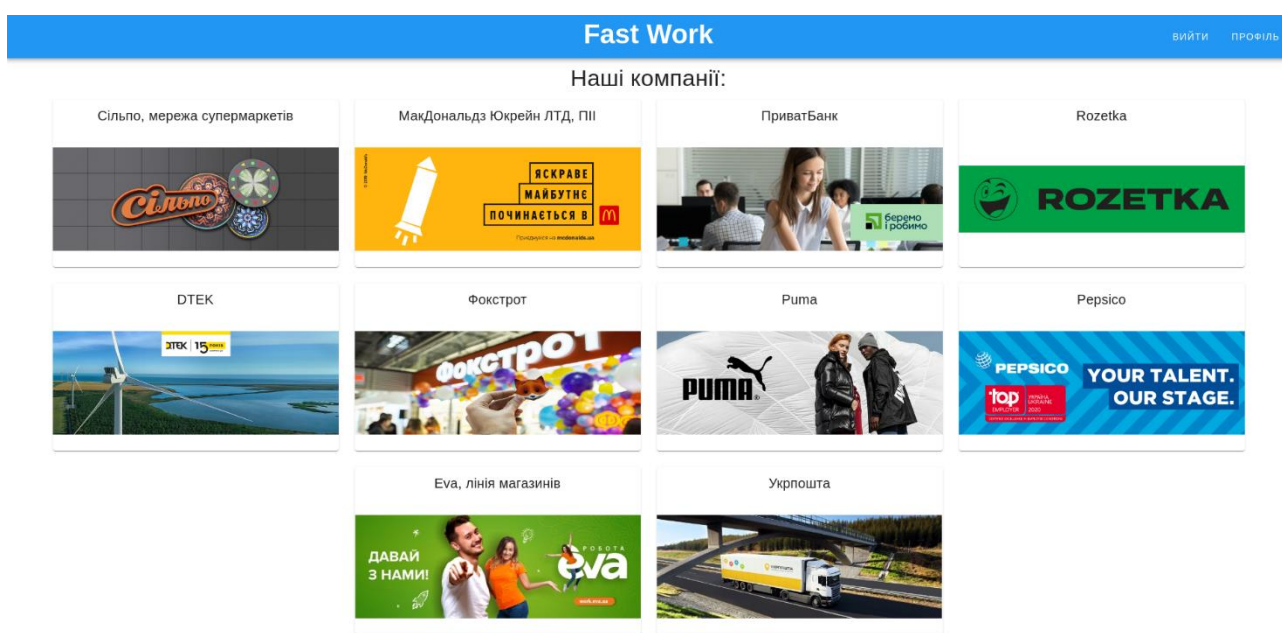
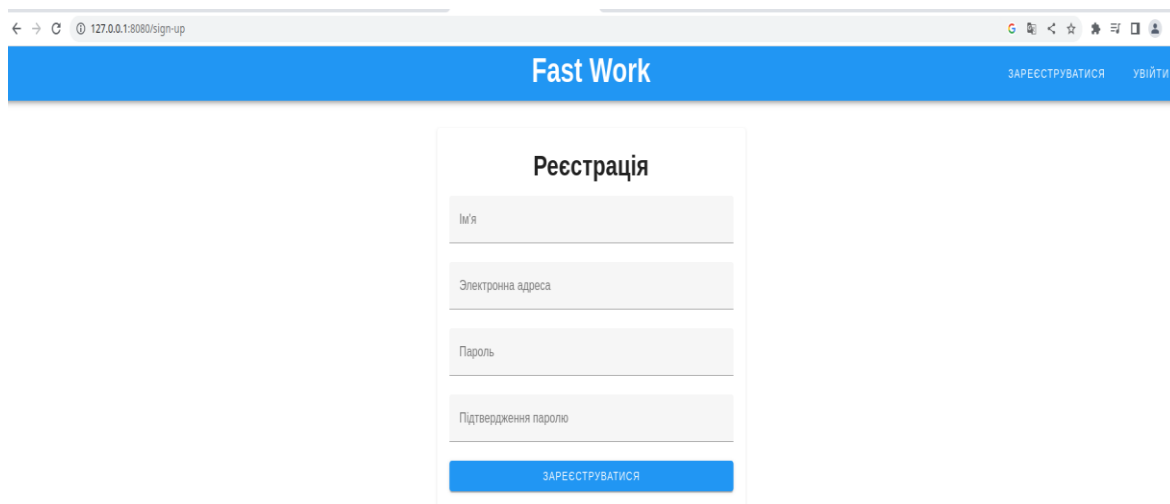


Рисунок Г.2 – Головна сторінка веб-сервісу перегляду компаній

На цих сторінках міститься інформація про веб-сервіс, новини. Користувач може зареєструватися та авторизуватися з головної сторінки

Щоб зареєструватися або увійти необхідно натиснути на кнопки Реєстрація (Рисунок Г.3) або Увійти (рисунок Г.4).



← → 127.0.0.1:8080/sign-up

Fast Work ЗАРЕЄСТРУВАТИСЯ УВІЙТИ

Реєстрація

Ім'я

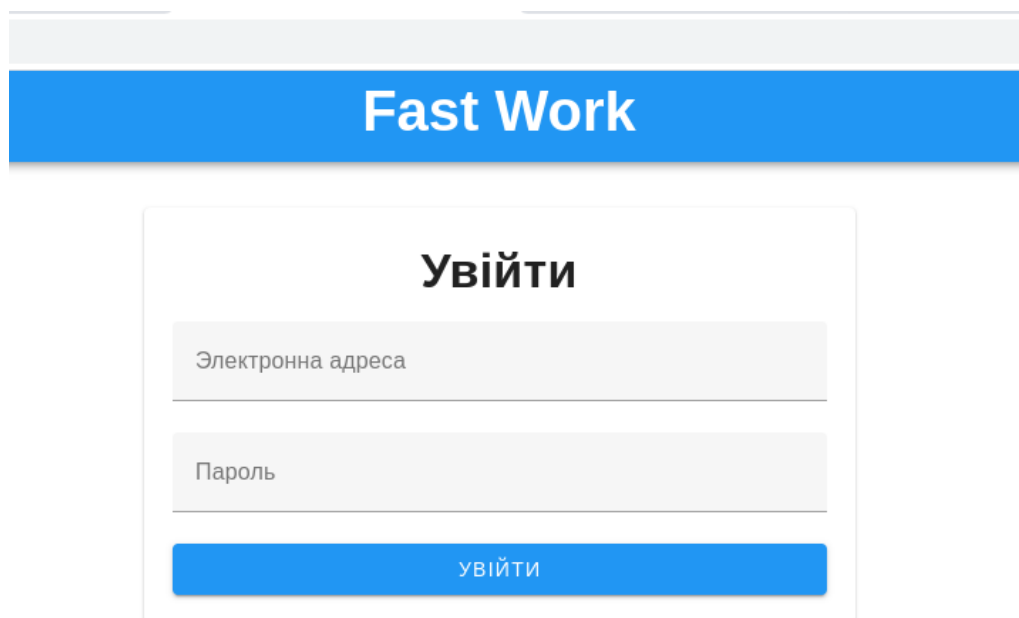
Електронна адреса

Пароль

Підтвердження паролю

ЗАРЕЄСТРУВАТИСЯ

Рисунок Г.3 – Сторінка Реєстрація



Fast Work

Увійти

Електронна адреса

Пароль

УВІЙТИ

Рисунок Г.4 – Сторінка Увійти

Для того щоб заповнити форми «Увійти» та «Реєстрація» необхідно скористатися клавіатурою та натиснути на контрольні кнопки ініціалізації дії.

Усі введенні дані повинні відповідати правилам.

- Електронна адреса повинна бути унікальною, це означає що може бути тільки один користувач з таким логіном;
- Пароль повинен мати мінімальну кількість символів.

Після успішної авторизації або реєстрації користувача перенаправить на сторінку заповнення особистого профілю (див. рис. Г.5).

Ваше резюме: 😊

Ім'я	Nikita Chopenko
Електронна адреса	webnikitchopenko@gmail.com
Skills	php, js, sql, docker
Мінімальна зарплата	1000
Опис	my description

Оновити

Рисунок Г.5 – Сторінка заповнення резюме

Сторінка перегляду резюме складається з текстових полів, які необхідно заповнити та потім натиснути клавішу оновити.

Головною функцією розробленого додатку є можливість шукати доступні резюме. Користувач може перейти на головну сторінку та заповнити поле «Посада» та натиснути на кнопку Шукати. Далі відкриється сторінка вакансій. (Рисунок Г.6)

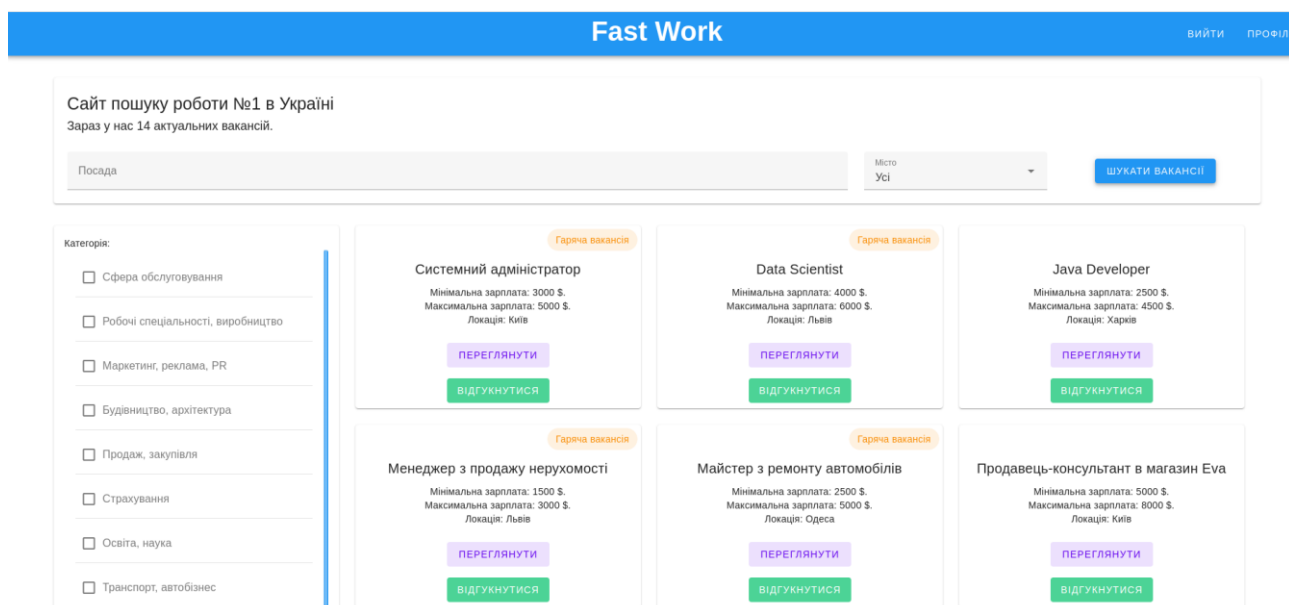


Рисунок Г.6 – Сторінка перегляду вакансій

Зліва можна побачити блок з фільтрами: «Категорія» та «Мінімальна заробітна плата». Для того щоб вибрати фільтр потрібно натиснути на будь-який пункт зі списку. (Рисунок Г.7)

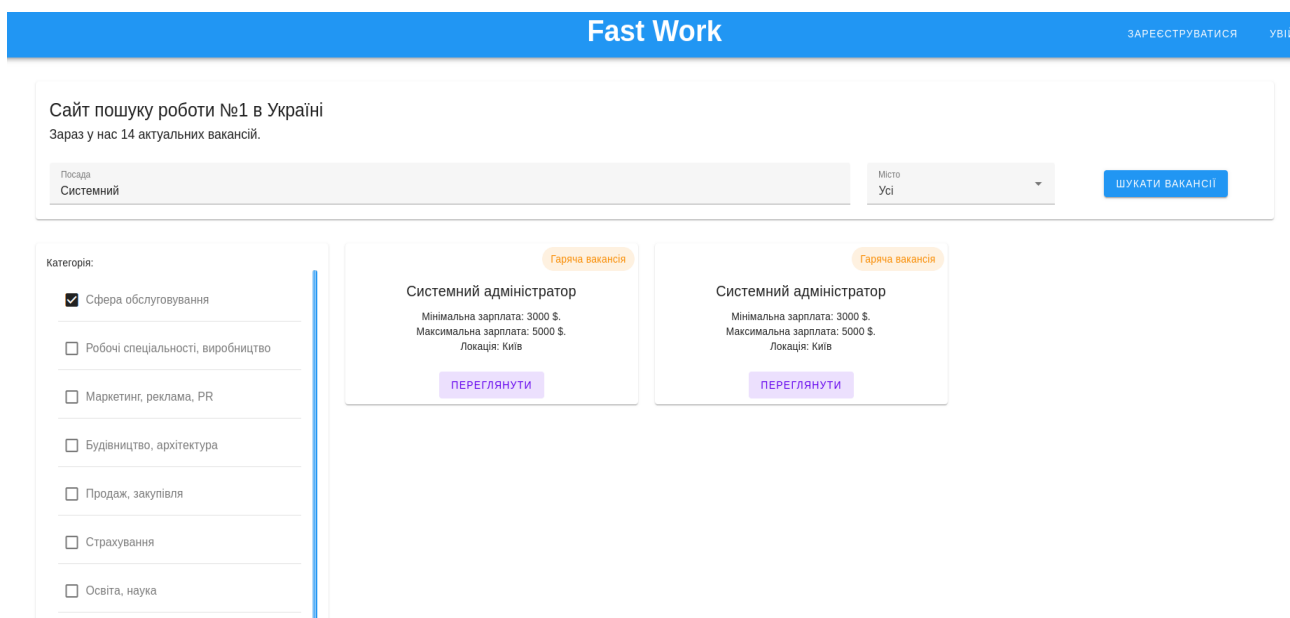


Рисунок Г.7 – Пошук вакансій

Додатковий фільтром є також мінімальна заробітна плата, яка допомагає шукати вакансії за вибраним діапазоном. (Рисунок Г.8)

Фільтри пошуку вакансій:

- ☐ Будівництво, архітектура
- ☐ Продаж, закупівля
- ☐ Страхування
- ☐ Освіта, наука
- ☐ Транспорт, автобізнес
- ☐ Нерухомість
- ☐ Фінанси, банк
- ☐ Охорона, безпека
- ☐ Культура, музика, шоу-бізнес
- ☐ IT, комп'ютери, інтернет

Зарплата:

Мінімальна зарплата: 0

Максимальна зарплата: 1000000

Рисунок Г.8 – Фільтри пошуку вакансій

Також можна вибрати фільтр за містом (Рисунок Г.9)

Fast Work ВИЙТИ П

ШУКАТИ ВАКАНСІЇ

Місто

- Київ
- Львів
- Миколаїв
- Одеса
- Харків
- Усі

Data Scientist

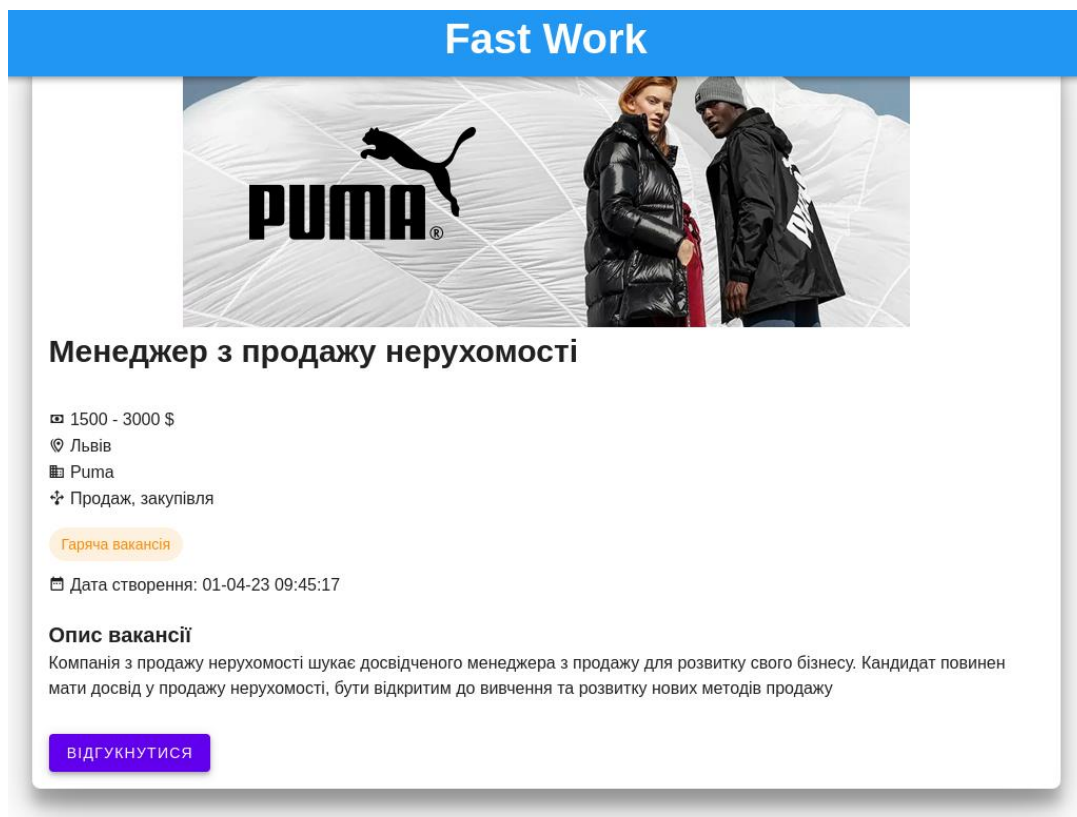
Мінімальна зарплата: 4000 \$
Максимальна зарплата: 6000 \$
Локація: Львів

ПЕРЕГЛЯНУТИ

ВІДГУКНУТИСЯ

Рисунок Г.9 – Фільтр пошуку за містом

Сторінка перегляду вакансій містить тільки основну інформацію. Користувач може натиснути кнопку на «Переглянути вакансію» і відкриється сторінка, яка буде містити усю доступну інформацію. (Рисунок Г.10)



Ваше резюме було успішно відправлено ! CLOSE

Рисунок Г.10 – Сторінка перегляду вакансії

Користувач може відгукнутися на відповідну вакансію, для цього необхідно натиснути на кнопку «Відгукнутися», після цього буде показано вікно з успішною дією. Відгуки, які зробив користувач . (Рисунок Г.11)

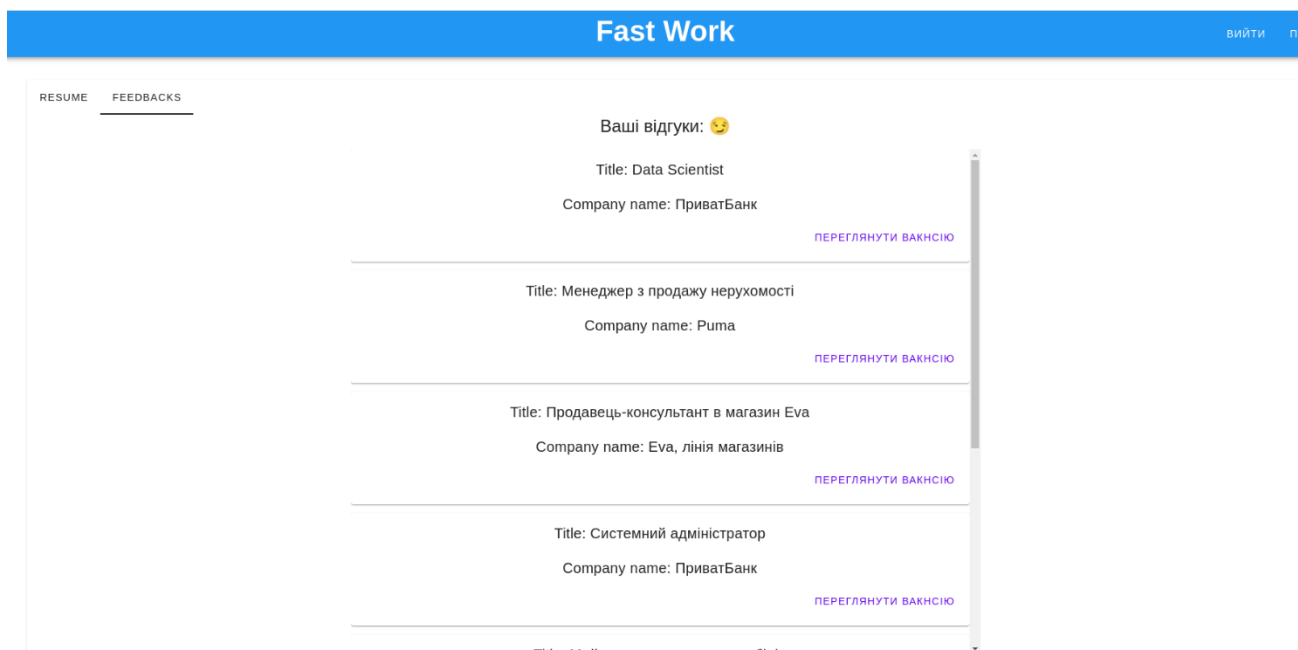


Рисунок Г.11 – Сторінка перегляду відгуків

Усі доступні новини можна переглянути на спеціальній сторінці. (Рисунок Г.12)

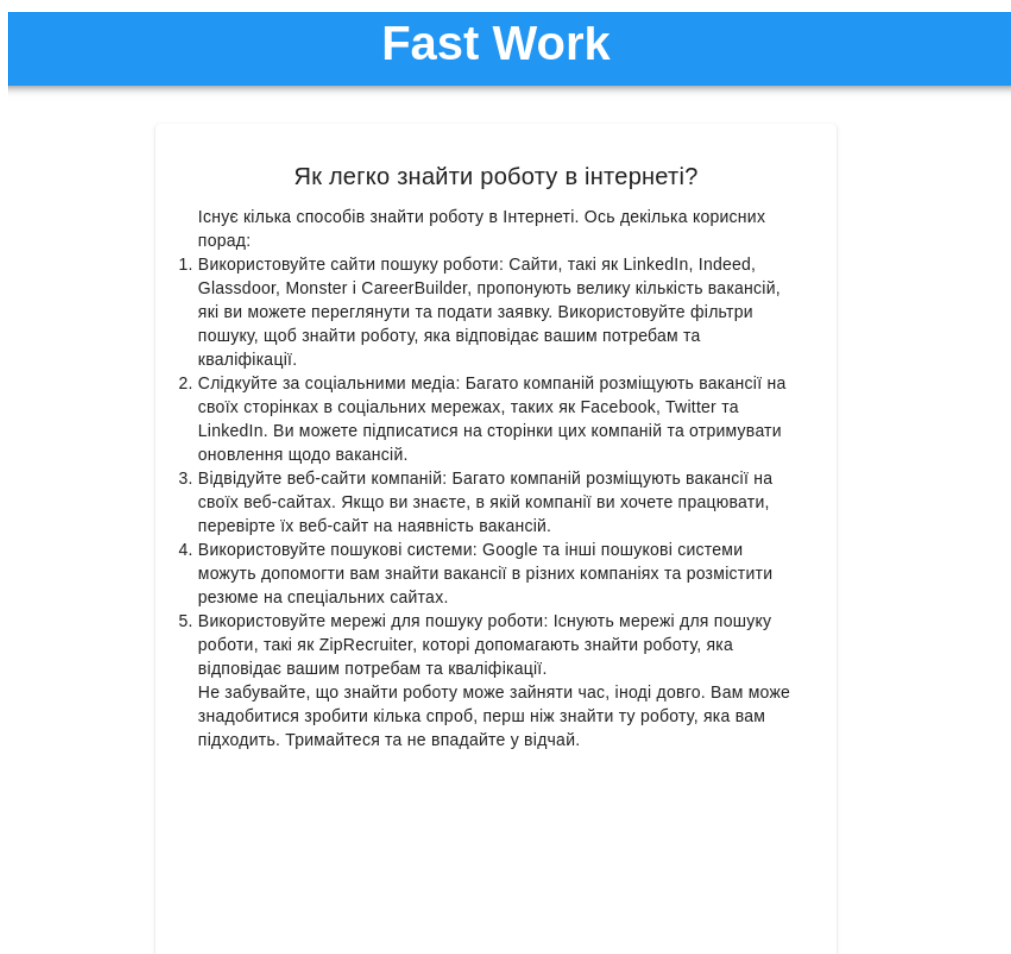
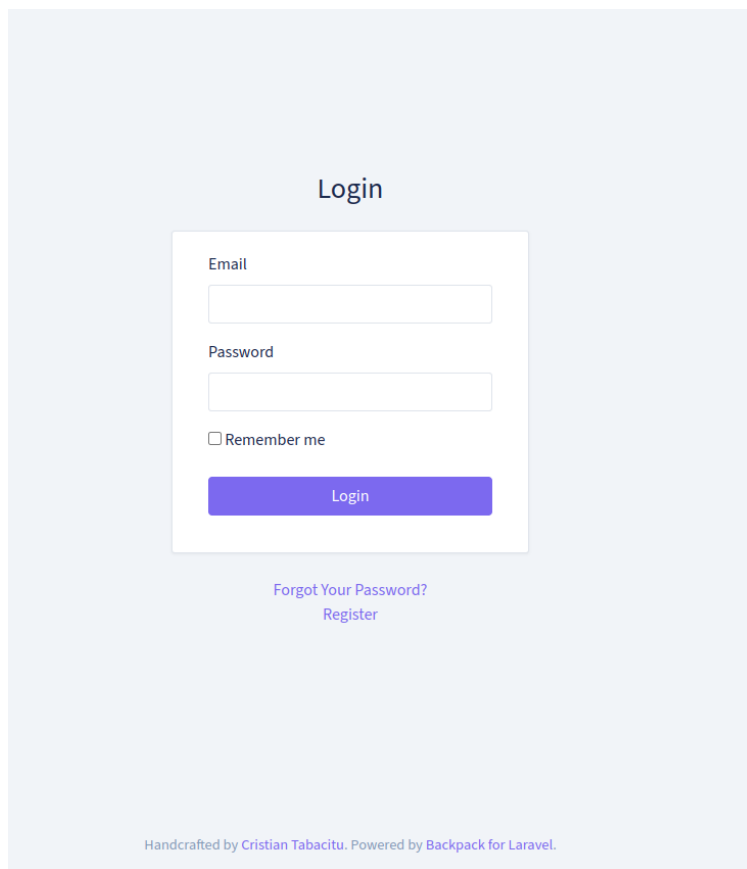


Рисунок Г.12 – Сторінка перегляду новини

Для управління даними, які має веб-сервіс була створена адміністраторська панель. Увійти до неї можна за допомогою логіну та паролю. (Рисунок Г.13)



Login

Email

Password

☐ Remember me

Login

[Forgot Your Password?](#)

[Register](#)

Handcrafted by Cristian Tabacitu. Powered by Backpack for Laravel.

Рисунок Г.13 – Сторінка входу до адміністраторської панелі

Після успішного входу користувач побачить усі сутності програмного забезпечення. (Рисунок Г.14)

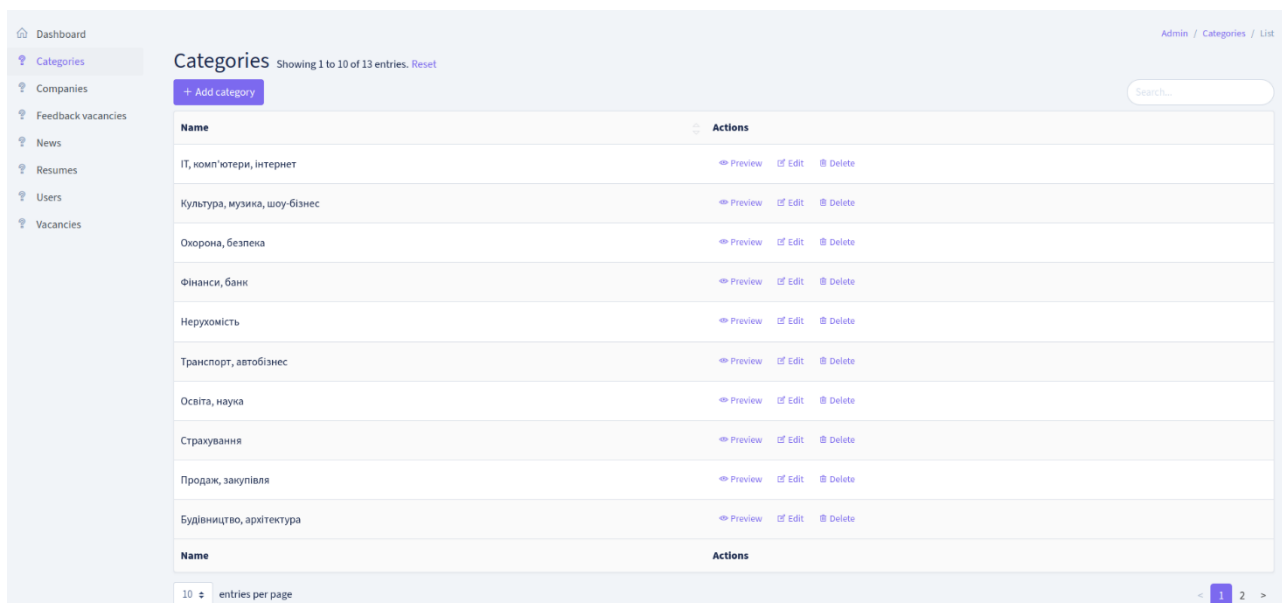


Рисунок Г.14 – Головна сторінка адміністратора

Адміністратор може додавати, оновлювати, видаляти записи за допомогою зручних форм та таблиць. Приклад додавання категорії, компанії, вакансії, зображений на рисунках Г.15-Г.17.

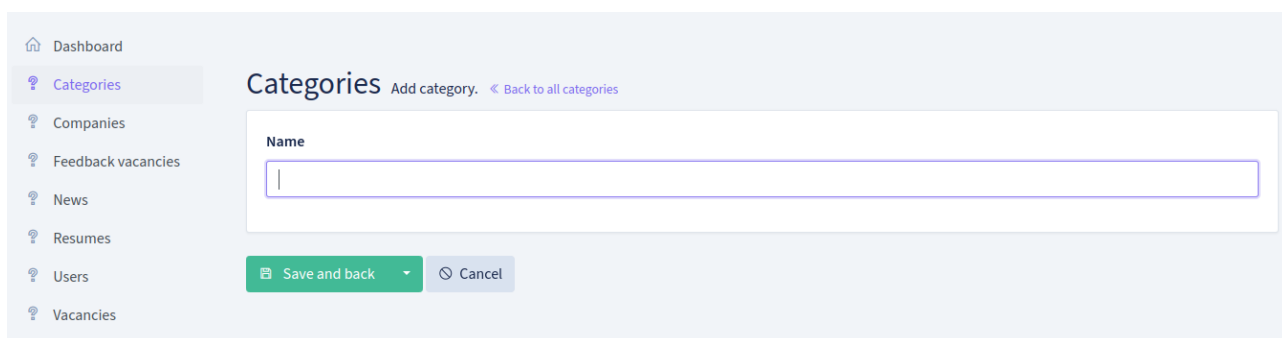


Рисунок Г.15 – Форма додавання категорії

Dashboard

Categories

Companies

Feedback vacancies

News

Resumes

Users

Vacancies

Companies

Add company. [Back to all companies](#)

Name

Image url

Location

Save and back Cancel

Рисунок Г.16 – Форма додавання компанії

Dashboard

Categories

Companies

Feedback vacancies

News

Resumes

Users

Vacancies

Vacancies

Add vacancy. [Back to all vacancies](#)

Name

Description

Min salary

Max salary

Location

☐ Is hot

Category

Сфера обслуговування

Company

Сільпо, мережа супермаркетів

Save and back Cancel

Рисунок Г.17 – Форма додавання вакансії

Також адміністратор може редагувати з сторінок таблиць сутностей.
(Рисунок Г.18)

Vacancies Showing 1 to 10 of 14 entries. [Reset](#)

[+ Add vacancy](#)

Name	Description	Min salary	Max salary	Location	Is hot	Category	Company	Actions
Менеджер з продажу в магазин Eva	Магазин Eva шукає молоду та амб...	8000	12000	Одеса	☐	Сфера обслуговування	Eva, лінія магазинів	Preview Edit Delete
Продавець-консультант в магазин...	Магазин Eva запрошує на роботу п...	5000	8000	Київ	☐	Сфера обслуговування	Eva, лінія магазинів	Preview Edit Delete
Майстер з ремонту автомобілів	Автосервіс шукає досвідченого ма...	2500	5000	Одеса	☒	Сфера обслуговування	Rozetka	Preview Edit Delete
Менеджер з продажу нерухомості	Компанія з продажу нерухомості ш...	1500	3000	Львів	☒	Продаж, закупівля	Puma	Preview Edit Delete
Java Developer	Ми шукаємо Java Developer, який...	2500	4500	Харків	☐	ІТ, комп'ютери, інтернет	Rozetka	Preview Edit Delete
Data Scientist	Ми шукаємо талановитого Data Sci...	4000	6000	Львів	☒	ІТ, комп'ютери, інтернет	ПриватБанк	Preview Edit Delete
Системний адміністратор	Ми шукаємо досвідченого системно...	3000	5000	Київ	☒	ІТ, комп'ютери, інтернет	ПриватБанк	Preview Edit Delete
Менеджер з продажу в магазин Eva	Магазин Eva шукає молоду та амб...	8000	12000	Миколаїв	☐	Сфера обслуговування	Eva, лінія магазинів	Preview Edit Delete
Продавець-консультант в магазин...	Магазин Eva запрошує на роботу п...	5000	8000	Київ	☐	Сфера обслуговування	Eva, лінія магазинів	Preview Edit Delete
Майстер з ремонту автомобілів	Автосервіс шукає досвідченого ма...	2500	5000	Одеса	☒	Сфера обслуговування	Rozetka	Preview Edit Delete
Name	Description	Min salary	Max salary	Location	Is hot	Category	Company	Actions

10 entries per page < 1 2 >

Рисунок Г.18 – Таблиця перегляду вакансій

ДОДАТОК Д ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

1. Об'єкт випробувань

Об'єктом випробувань є веб-сервіс пошуку роботи «Fast-Work».

Предметною областю є пошук роботи.

2. Мета випробувань

Мета проведення випробувань є перевірка готового функціоналу веб-сервісу пошуку роботи «Fast-Work».

3. Вимоги до програми

При проведенні випробувань програмного забезпечення, функціональні характеристики повинні відповідати вимогам, які були викладенні у пункті «Постановка задачі».

4. Вимоги до документації програми

До складу програмної документації входить:

- технічне завдання;
- текст програми;
- опис програми;
- інструкція користувача;
- програма та методика випробувань.

5. Склад та порядок випробувань

5.1 Склад технічних засобів

Система повинна задовольняти вказаним вимогам на комп'ютері наступної мінімальної комплектації:

- Оперативна пам'ять 8Гб;
- Процесор Intel® Pentium Gold G6400 4.00 GHz;
- Відеоадаптери Intel® UHD Graphics 610.

Швидкість передачі даних повинна бути не менше 30Мб/с.

5.2 Склад програмних засобів

До складу програмних засобів, які будуть залучені при випробуванні програмного забезпечення:

- Операційна система Windows та Linux;
- Браузер Chrome (Версія 133);
- Браузер Microsoft Edge (94).

5.3 Порядок проведення випробувань

При проведенні випробування було вибрано метод білої скриньки. Він базується в аналізі вихідного коду, структурі даних та внутрішні процеси для створення тестових сценаріїв та випробування різних шляхів виконання.

Перелік випробувань основних функціональних можливостей:

- Авторизація як шукач;
- Авторизація як адміністратор;
- Пошук вакансій як гість та шукач;
- Додавання резюме;
- Відгук на вакансію.