

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

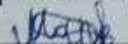
КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

САЙТ ДЛЯ ПОШУКУ РОБОТИ ДЛЯ
СТУДЕНТІВ ТА ВИПУСКНИКІВ ЛНТУ
JOB SEARCH SITE FOR STUDENTS AND
GRADUATES OF LNTU

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

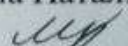
освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗс-31
Ярчак Марко Анатолійович


(підпис)

Керівник:
асистент
Вознюк Анастасія Вадимівна


(підпис)

Кваліфікаційну роботу
допущено до захисту
«1» 06 2023 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«08» 02 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Ярчаку Маркові Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Сайт для пошуку роботи для студентів та випускників ЛНТУ

Керівник роботи: асистент Вознюк Анастасія Вадимівна

затверджені наказом закладу вищої освіти від «23» грудня 2022 р. № 961-01-02

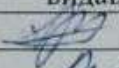
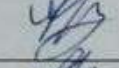
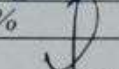
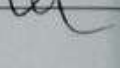
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «8» червня 2023 р.

3. Вихідні дані до роботи: програмне забезпечення, вимоги до розробки програмного забезпечення, функціональні та не функціональні вимоги до програмного засобу, Технології програмування клієнтської частини – HTML, SCSS, JavaScript, Vue, Vuetify, Firebase, розробка бази даних.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів розробки, опис функціонального наповнення об'єкта проєктування, розробка дизайну, розробка додатку.

5. Перелік графічного матеріалу: 28 рис.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Вознюк А.В.		
Специфікація вимог до розробленої системи	Вознюк А.В.		
Розробка об'єкта проєктування	Вознюк А.В.		
Нормоконтроль	Вознюк А.В.		
Гарант ОП	Ліщина Н.М.		
Показник запозичень тексту		2,24%	
Академічна доброчесність	Яциук А.А.		

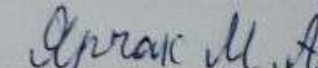
7. Дата видачі завдання «2» лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Прим.
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	05.02.2023 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2023 р.	
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	10.03.2023 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	17.04.2023 р.	
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2023 р.	
6	Тестування та налагодження об'єкта проєктування	01.06.2023 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	08.06.2023 р.	

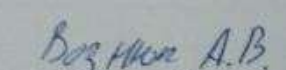
Здобувач вищої освіти


(підпис)


(прізвище, ініціали)

Керівник кваліфікаційної роботи


(підпис)


(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Факультет комп'ютерних та інформаційних технологій
(назва)

Кафедра інженерії програмного забезпечення
(назва)

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач освіти Ярчак Марко Анатолійович
(ПІБ)

група ІПЗс-31
(шифр)

Тема кваліфікаційної роботи:

Сайт для пошуку роботи для студентів та випускників ЛНТУ

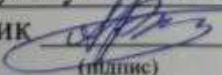
(повна назва згідно наказу)

Керівник: асистент Вознюк Анастасія Вадимівна
(науковий ступінь, вчене звання, посада, ПІБ)

Актуальність теми: сайти для пошуку робіт надають зручний та ефективний
Спосіб зв'язку між шукачами роботи та роботодавцями. Уміння знайти швидко
відповідну вакансію та встановити контакт із потенційним роботодавцем.
Об'єкт дослідження: є вебсайт, який розроблявся для забезпечення студентів
та випускників ЛНТУ, що допоможе знайти роботу.

Характеристика теоретичного рівня роботи, наявності самостійних розробок і
практичної значущості роботи: ця робота характеризується високим рівнем
теоретичних знань, що дозволяють зрозуміти основні принципи веброзробки.
Поєднання самостійних розробок з реалізацією програмного продукту
підтверджує його вміння використовувати теоретичні знання на практиці.
Результати дослідження мають велику значущість.
Зауваження та недоліки: ця робота представляє собою альтернативу наявним
рішенням проблеми, проте варто зазначити, що дослідження демонструє
навички дипломанта.

Загальний висновок: робота була виконана на достатньому рівні. Всі завдання
успішно та повноцінно виконано. Кваліфікаційна робота відповідає вимогам
та заслуговує на високу оцінку.

Керівник  (підпис) (Вознюк Анастасія Вадимівна)
(ПІБ)

«8» червня 2023 р.

АНОТАЦІЇ

Ярчак М.А. Сайт для пошуку роботи для студентів та випускників ЛНТУ.
Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності Інженерія програмного забезпечення. Луцький національний технічний університет. Луцьк, 2023.57 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Метою роботи є розробка інноваційного сайту для пошуку роботи, спеціально призначеного для студентів та випускників

У першому розділі досліджено сучасний стан проблеми із сайтами для працевлаштування, розглянуто актуальні сайти для пошуку робіт.

У другому розділі визначено вимоги для розроблювального сайту, описано особливості обраного стеку технологій.

У третьому розділі продемонстровано практичну реалізацію об'єкта проєктування, опис розробленого сайту та приклад роботи з сайтом.

Ключові слова: сайт, пошук роботи, JavaScript, Vue.js.

ANNOTATION

Yarchak M.A. Job search site for students and graduates of LNTU. Manuscript.

Bachelor's qualifying work of the OP "Software Engineering" specialty Software Engineering. Lutsk National Technical University. Lutsk, 2023. 57 p.

The qualification work consists of an introduction, three sections, conclusions, a list of used sources, and appendices.

The purpose of the work is to develop an innovative job search site, specially designed for students and graduates

In the first chapter, the current state of the problem with employment sites was investigated, current job search sites were considered.

The second section defines the requirements for the development site, describes the features of the selected technology stack.

The third section demonstrates the practical implementation of the design object, a description of the developed site and an example of working with the site.

Keywords: site, job search, JavaScript, Vue.js.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ САЙТІВ ДЛЯ ПОШУКУ РОБОТИ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	10
1.1 Аналіз сучасного стану проблеми	10
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	17
Висновки до розділу 1	18
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	19
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення.....	19
2.1.1 Функціональні та нефункціональні вимоги	20
2.1.2 Моделі елементів програмного додатку у нотації UML.....	21
2.1.3 Створення UX/UI-дизайну	24
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	26
Висновки до розділу 2	32
РОЗДІЛ 3 РОЗРОБКА САЙТУ ДЛЯ ПОШУКУ РОБОТИ ДЛЯ СТУДЕНТІВ ТА ВИПУСКНИКІВ ЛНТУ	32
3.1 Практична реалізація об'єкта проєктування	33
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	50
3.3 Розробка документації для програмного продукту	52
Висновки до розділу 3	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТКИ.....	60

ВСТУП

Актуальність теми роботи є важливим інструментом як для тих, хто шукає роботу, так і для роботодавців на сучасному ринку праці. Ці вебсайти надають платформу для роботодавців для розміщення вакансій, а для шукачів – для пошуку та подання заявок на ці вакансії.

З появою Інтернету за останні роки кількість сайтів пошуку роботи зросла в геометричній прогресії, з незліченною кількістю варіантів, доступних для шукачів роботи. Ці сайти варіюються від великих загальних сайтів пошуку роботи до спеціальних сайтів, зосереджених на конкретних галузях, типах робіт або демографічних показниках.

Шукачі роботи можуть використовувати ці сайти, щоб шукати потенційних роботодавців, дізнаватися про вакансії та подавати заявки онлайн. Ці сайти також надають такі ресурси, як шаблони резюме та супровідних листів, поради щодо співбесіди та інформацію про зарплату.

Для роботодавців сайти пошуку роботи надають величезну кількість потенційних кандидатів, що дозволяє їм розширити коло вакансій. Роботодавці також можуть використовувати ці сайти для перегляду резюме, спілкування з кандидатами та керування процесом найму.

Однак через велику кількість доступних сайтів пошуку роботи шукачам і роботодавцям може бути важко орієнтуватися та вибирати найефективніші варіанти. Деякі сайти пошуку роботи стягують з роботодавців плату за публікацію списків вакансій, тоді як інші можуть мати обмежені або застарілі оголошення про роботу.

Незважаючи на ці проблеми, сайти пошуку роботи залишаються важливим інструментом у процесі пошуку роботи. Завдяки своїй зручності, доступності та широкому охопленню ці сайти пропонують шукачам роботи та роботодавцям цінний ресурс для зв'язку та пошуку потрібної пари.

Мета дослідження є розробка інноваційного сайту для пошуку роботи, спеціально призначеного для студентів та випускників. Програмне забезпечення має надавати комфортний і швидкий доступ до вакансій, ефективний пошук та їх фільтрацію, а також можливість додавання резюме та взаємодії з роботодавцями. Головною метою дослідження є створення цілісної та функціональної платформи, що задовольняє потреби студентів і випускників у пошуку роботи та сприяє їхньому професійному розвитку.

Об'єкт дослідження є вебсайт, який розробляється для забезпечення студентів та випускників можливістю ефективного пошуку роботи. Вебсайт має на меті збирати та відображати різноманітні вакансії, надавати зручний інтерфейс для пошуку та фільтрації вакансій за різними критеріями. Крім того, вебсайт надає можливість користувачам створювати та редагувати свої резюме, встановлювати спілкування з потенційними роботодавцями та отримувати оновлення про нові вакансії шляхом підписки на розсилку.

Предметом дослідження є програмне втілення сайту для пошуку роботи, що базується на використанні Vue.js, Firebase та Vuetify. Основна мета дослідження полягає в розробці та вдосконаленні функціоналу сайту, забезпечуючи користувачам зручність та ефективність процесу пошуку вакансій та можливості додавання резюме. в рамках дослідження були проведені роботи щодо інтеграції з Firebase для збереження користувацьких даних, розробки фільтрів, аутентифікації та захисту інформації з метою забезпечення безпеки та зручності використання сайту.

Для досягнення мети дослідження були поставлені наступні завдання:

- розробка сайту має забезпечувати легкий та зручний пошук вакансій для користувачів;
- сайт повинен містити можливість додавання резюме користувачами та їхнього редагування;
- реалізувати функціонування різноманітних фільтрів та інструментів для пошуку вакансій, що забезпечують швидкий та точний пошук;

- розроблюваний сайт повинен бути масштабованим та легким для розширення функціоналу в майбутньому;
- адаптувати дизайн сайту під різні пристрої, що забезпечить зручний перегляд сайту на різних пристроях;
- забезпечити безпеку даних користувачів, включаючи захист від атак зловмисників та незаконного доступу до даних;
- реалізувати рекомендаційні системи, які забезпечують користувачам персоналізований підбір вакансій;
- забезпечити можливість підписки на розсилку нових вакансій за певними критеріями;
- забезпечити можливість комунікації між роботодавцями та кандидатами для подальшої співпраці.

РОЗДІЛ 1

АНАЛІЗ САЙТІВ ДЛЯ ПОШУКУ РОБОТИ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

З розвитком ринку праці з'являються нові виклики та можливості як для шукачів роботи, так і для роботодавців. Відтак, використання сайтів пошуку роботи стає все популярнішим, оскільки вони надають зручний та ефективний спосіб з'єднати шукачів роботи з потенційними роботодавцями. Проте, разом із зростаючою популярністю, сайти пошуку роботи стикаються з проблемою поширених шахрайських оголошень про роботу, що можуть ввести шукачів в оману та призвести до втрати часу, доходу і навіть крадіжки особистих даних.

Ще однією проблемою сайтів пошуку роботи є величезна кількість доступних варіантів. З такою кількістю різних сайтів і платформ на вибір, шукачам роботи може бути важко визначити, які з них є законними та ефективними. Крім того, деякі сайти можуть містити обмежені списки вакансій або обслуговувати певну галузь, що може не підходити для всіх шукачів роботи.

Крім того, хоча сайти пошуку роботи пропонують велику кількість ресурсів для шукачів, вони не завжди можуть надавати точну чи актуальну інформацію. Наприклад, інформація про зарплату, наведена на цих сайтах, може не відображати поточні ринкові тенденції чи регіональні відмінності, що може ввести в оману шукачів роботи.

З боку роботодавця також є проблеми з використанням сайтів пошуку роботи. Однією з проблем є велика кількість заявок, які можуть отримати роботодавці, що може ускладнити ефективний відбір кандидатів і пошук того, хто їм підходить. Крім того, деякі сайти пошуку роботи можуть вимагати плату за розміщення списків вакансій або доступ до баз даних кандидатів, що може коштувати дорого роботодавцям.

Загалом, незважаючи на те, що сайти пошуку роботи пропонують багато переваг шукачам роботи та роботодавцям, є також значні проблеми та недоліки, які слід враховувати. Для того, щоб ефективно орієнтуватися в процесі пошуку роботи, шукачам роботи та роботодавцям важливо ретельно оцінювати доступні варіанти та пам'ятати про потенційні ризики та обмеження використання сайтів пошуку роботи.

Для розробки сайту, спочатку потрібно провести аналіз серед аналогів, щоб уникнути недоліків, яких допустили конкуренти. Для аналізу існуючих аналогів було обрано наступні:

- jooble.org.ua;
- rabota.ua;
- dou.ua;
- work.ua.

Jooble – популярна в Україні пошукова система, яка дозволяє шукачам в одному місці шукати вакансії, розміщені на різних дошках вакансій та корпоративних сайтах. Сайт був заснований у 2006 році і з тих пір став провідним агрегатом пошуку роботи в Україні [2].

Jooble.org.ua має на меті спростити процес пошуку роботи як для шукачів, так і для роботодавців. Вебсайт збирає списки вакансій із широкого кола джерел і представляє їх у зручному для користувача форматі, що полегшує шукачам роботу пошук відповідних оголошень про роботу. Сайт також пропонує низку функцій, таких як фільтри пошуку, сповіщення електронною поштою та конструктор резюме, щоб допомогти шукачам роботи ефективніше керувати пошуком роботи.

За даними SimilarWeb, станом на квітень 2023 року Jooble.org.ua має приблизно 10,4 мільйона відвідувань на місяць, що робить його одним із найпопулярніших сайтів пошуку роботи в Україні. Сайт доступний українською мовою та пропонує оголошення про роботу в різних галузях, включаючи IT, продажі, маркетинг, фінанси тощо [3].

Переваги використання Jooble [4]:

- широкий вибір оголошень про роботу – Jooble.org.ua збирає списки вакансій з різних джерел, що робить його комплексною системою пошуку роботи для шукачів;
- зручний інтерфейс. Вебсайт простий у використанні та навігації, доступні різноманітні пошукові фільтри та параметри сортування;
- сповіщення про роботу. Шукачі роботи можуть налаштувати сповіщення електронною поштою, щоб отримувати сповіщення про публікацію нових оголошень про роботу, які відповідають критеріям пошуку;
- конструктор резюме. Jooble пропонує безкоштовний інструмент для створення резюме, який допоможе шукачам роботи створити професійні резюме;
- економія часу. Використання jooble.org.ua може заощадити час шукачів роботи, усуваючи необхідність відвідувати численні дошки оголошень і вебсайти компаній, щоб знайти вакансії.

Недоліки використання Jooble [4]:

- якість списків вакансій. Якість оголошень про вакансії на Jooble.org.ua може відрізнятися, оскільки сайт збирає списки вакансій із різних джерел, деякі з яких можуть бути нелегітимними;
- обмежена інформація. У деяких оголошеннях про роботу на Jooble.org.ua може бути недостатньо інформації про вимоги до роботи, обов'язки чи зарплату;
- конкуренція з іншими претендентами. Оскільки Jooble.org.ua є популярною системою пошуку роботи, шукачі роботи можуть зіткнутися з конкуренцією з боку інших претендентів, які також подали заявку на те саме оголошення про роботу;
- не всі оголошення про вакансії є актуальними. Деякі оголошення про вакансії на Jooble можуть бути неактуальними, оскільки сайт покладається на різні джерела списків вакансій, деякі з яких можуть не регулярно оновлювати свої оголошення про вакансії.

Jooble – популярна система пошуку роботи в Україні, яка пропонує шукачам зручний та ефективний спосіб пошуку вакансій у різних джерелах. Хоча є деякі недоліки використання сайту, такі як якість списків вакансій і конкуренція з іншими претендентами, переваги використання Jooble.org.ua, такі як широкий вибір оголошень про роботу та зручний інтерфейс, роблять його цінний ресурс для тих, хто шукає роботу [5]. На рисунку 1.1 зображено головну сторінку Jooble, де є пошук вакансій і вибір за містом проживання.

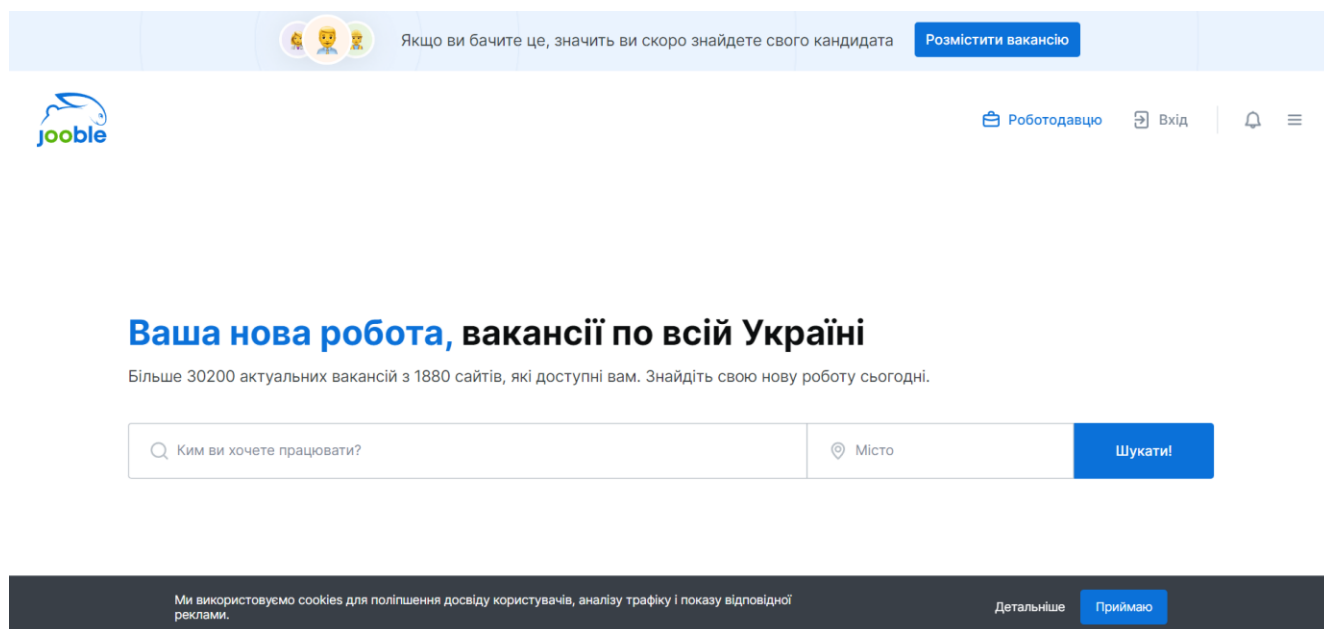


Рисунок 1.1 – Головна сторінка Jooble [6]

Work.ua – один із найпопулярніших сайтів пошуку роботи в Україні, який надає шукачам широкий вибір оголошень про роботу в різних галузях та регіонах. Вебсайт був запущений у 2007 році і з того часу став провідною онлайн-платформою для пошуку роботи та підбору персоналу в Україні [7].

Work.ua орієнтований як на шукачів роботи, так і на роботодавців. Сайт пропонує ряд функцій, які допоможуть шукачам роботи знайти відповідні вакансії, включаючи пошукову систему з різними фільтрами, сповіщення про роботу та калькулятор зарплати. Роботодавці також можуть використовувати платформу для розміщення вакансій, пошуку в базі даних резюме та керування процесом найму.

За даними SimilarWeb, станом на квітень 2023 року Work.ua має приблизно 9,9 мільйона відвідувань на місяць, що робить його одним із найбільш відвідуваних сайтів пошуку роботи в Україні. Сайт на українській мові та пропонує оголошення про роботу в різних галузях, таких як ІТ, фінанси, маркетинг, продажі тощо [8].

Переваги використання Work.ua:

- широкий вибір оголошень про роботу – Work.ua пропонує вакансії в різних галузях і місцях, що робить його комплексним сайтом пошуку роботи для шукачів;
- зручний інтерфейс. Вебсайт простий у навігації та використанні, доступні різноманітні пошукові фільтри та параметри сортування;
- розширена пошукова система. На сайті є потужна пошукова система з розширеними фільтрами, що дозволяє шукачам роботи швидко знаходити відповідні вакансії;
- сповіщення про роботу. Шукачі роботи можуть налаштувати сповіщення електронною поштою, щоб отримувати сповіщення про публікацію нових оголошень про роботу, які відповідають критеріям пошуку;
- калькулятор зарплати. За допомогою якого шукачі роботи можуть оцінити свою потенційну зарплату на основі своїх навичок, освіти та досвіду.

Недоліки використання Work.ua [9]:

- обмежена інформація. У деяких оголошеннях про роботу на Work.ua може бути недостатньо інформації про вимоги до роботи, обов'язки чи зарплату;
- конкуренція з іншими претендентами. Work.ua є популярним сайтом пошуку роботи, тому шукачі роботи можуть зіткнутися з конкуренцією з боку інших претендентів, які також подали заявку на те саме оголошення про роботу;
- шахрайство з роботою. Як і на будь-якому іншому сайті пошуку роботи, на Work.ua можуть бути шахрайства з роботою, тому шукачам роботи слід бути обережними, подаючись на вакансії;
- потрібно створити профіль. Щоб подати заявку на вакансії на Work.ua, шукачам роботи потрібно створити профіль на вебсайті, що може зайняти час і зусилля.

Work.ua є цінним ресурсом для шукачів роботи в Україні, що пропонує широкий вибір оголошень про роботу та розширені функції пошуку. Незважаючи на деякі недоліки, такі як обмежена інформація та конкуренція з боку інших претендентів, переваги використання Work.ua, такі як зручний інтерфейс і інструмент калькулятора зарплати (рис. 1.2), роблять його цінним інструментом для шукачів роботи, які шукають можливості працевлаштування в Україні.

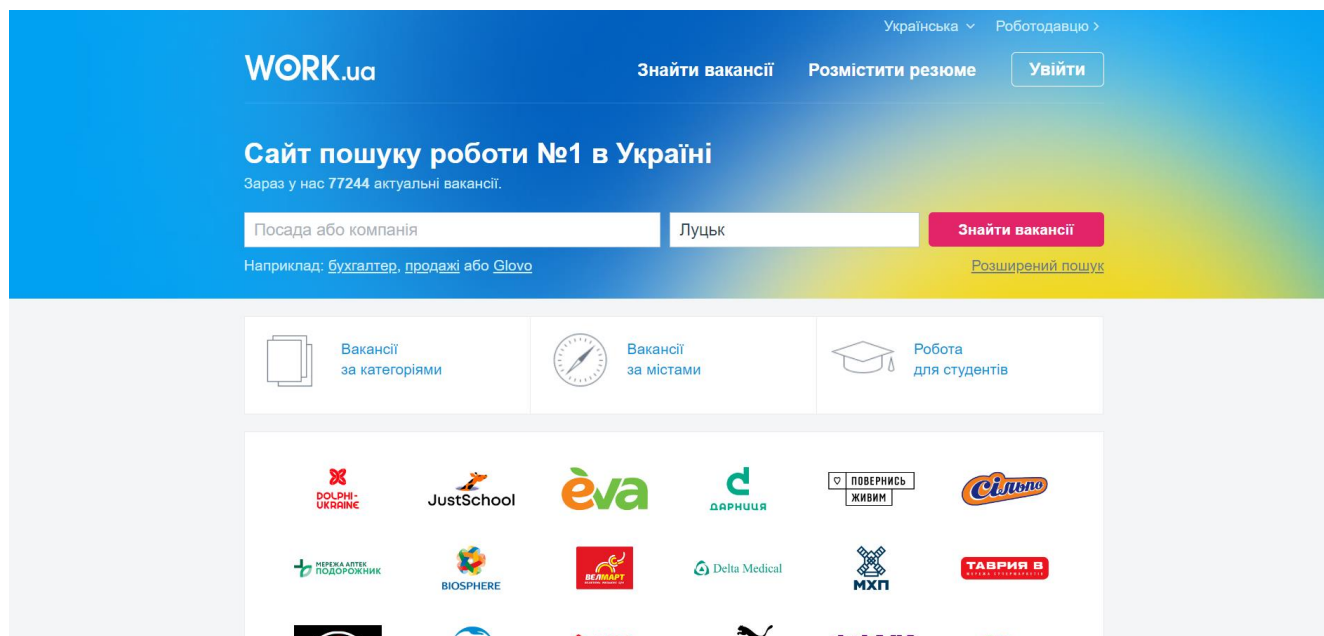


Рисунок 1.2 – Головна сторінка Work.ua [10]

DOU – популярна онлайн-платформа для IT-фахівців та шукачів роботи в Україні. Вебсайт був запущений у 2005 році і з тих пір став провідним сайтом для пошуку роботи в IT-сфері [11]. Він надає оголошення про роботу в різних галузях, але зосереджується в основному на IT, що робить його ідеальною платформою для тих, хто шукає роботу в цій галузі.

DOU орієнтований як на шукачів роботи в IT, так і на роботодавців. Він пропонує ряд функцій, які допомагають шукачам роботи знайти відповідні вакансії, включаючи пошукову систему з різними фільтрами, сповіщення про роботу та базу даних резюме. Роботодавці також можуть використовувати платформу для розміщення вакансій, пошуку в базі даних резюме та керування процесом найму.

За даними SimilarWeb, станом на квітень 2023 року DOU має приблизно 1,5 мільйона відвідувань на місяць. Сайт пропонує оголошення про роботу в різних містах, таких як Київ, Львів, Харків тощо [12].

Переваги використання DOU:

- зосередження на вакансіях у сфері IT. DOU – це платформа, спеціально створена для IT-фахівців і шукачів роботи, що робить її ідеальним місцем для пошуку вакансій, пов'язаних із IT;
- зручний інтерфейс. Вебсайт простий у навігації та використанні, доступні різноманітні пошукові фільтри та параметри сортування;
- розширена пошукова система – на сайті є потужна пошукова система з розширеними фільтрами, що дозволяє шукачам роботи швидко знаходити відповідні вакансії;
- сповіщення про роботу - шукачі роботи можуть налаштувати сповіщення електронною поштою, щоб отримувати сповіщення про публікацію нових оголошень про роботу, які відповідають критеріям пошуку;
- база резюме. DOU надає шукачам можливість завантажити своє резюме на сайт, яке зможуть переглянути потенційні роботодавці.

Недоліки використання DOU.ua:

- обмежені вакансії. Хоча DOU є популярною платформою для розміщення вакансій, пов'язаних із IT, на ній може бути не так багато вакансій в інших галузях;
- конкуренція з боку інших претендентів - DOU.ua є популярним сайтом пошуку роботи для IT-фахівців, тому шукачі роботи можуть зіткнутися з конкуренцією з боку інших претендентів, які також подали заявку на те саме оголошення про роботу;
- необхідно створити профіль. Щоб подати заявку на вакансії на DOU, шукачам необхідно створити обліковий запис на вебсайті, що може зайняти час і зусилля;
- обмежені мовні можливості.

Зробивши висновок, можна сказати, що DOU є цінним ресурсом для IT-фахівців та шукачів роботи в Україні, що пропонує низку функцій, які допоможуть шукачам знайти відповідну вакансію. Незважаючи на певні недоліки, такі як обмеженість оголошень про вакансії в інших галузях та конкуренція з боку інших претендентів, переваги використання DOU, такі як зосередженість на IT-вакансіях і зручний інтерфейс, роблять його цінним інструментом для шукачів роботи шукає можливості працевлаштування в Україні. На рисунку 1.3 відображено головну сторінку, де виведені актуальні новини у сфері IT.

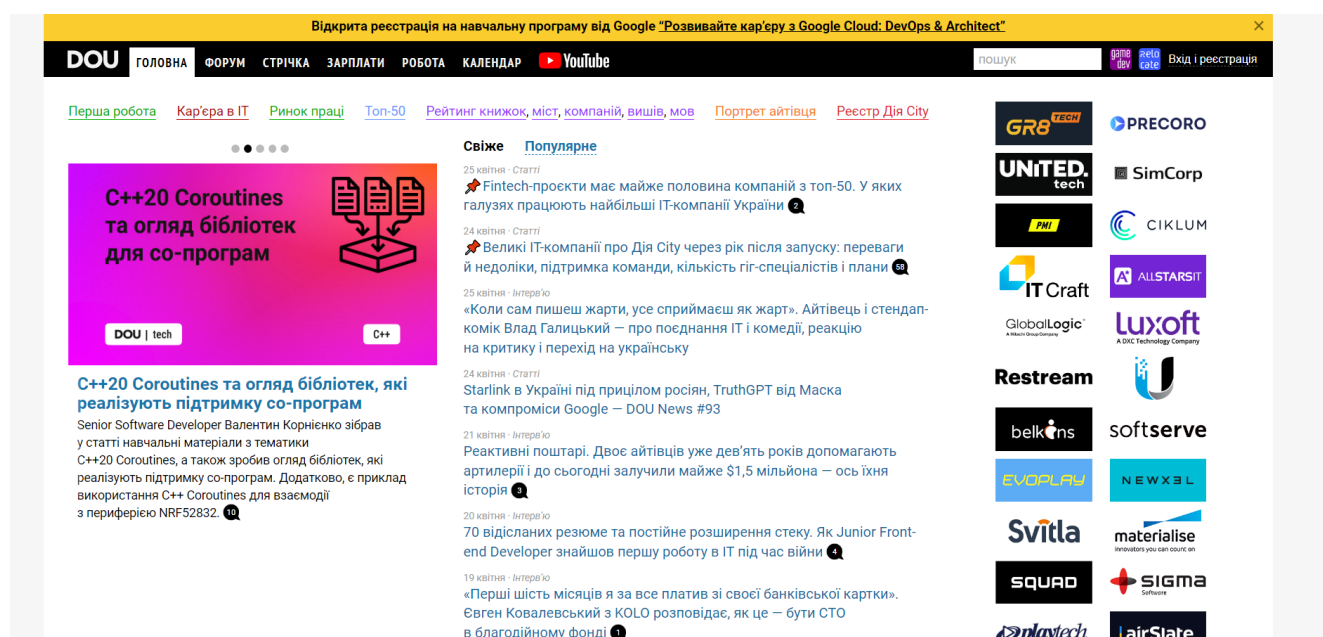


Рисунок 1.3 – Головна сторінка DOU [13]

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Завданнями даної роботи є:

- розробка сайту має забезпечувати легкий та зручний пошук вакансій для користувачів;
- сайт повинен містити можливість додавання резюме користувачами та їхнього редагування;

- наявність різноманітних фільтрів та інструментів для пошуку вакансій, що забезпечують швидкий та точний пошук;
- розроблюваний сайт повинен бути масштабованим та легким для розширення функціоналу в майбутньому;
- наявність адаптивного дизайну, що забезпечує зручний перегляд сайту на різних пристроях;
- забезпечення безпеки даних користувачів, включаючи захист від атак зловмисників та незаконного доступу до даних;
- наявність рекомендаційних систем, які забезпечують користувачам персоналізований підбір вакансій;
- можливість підписки на розсилку нових вакансій за певними критеріями;
- наявність функції спілкування між роботодавцями та кандидатами для подальшої співпраці.

Дотримання цих вимог дозволить створити сайт пошуку роботи, який забезпечить користувачам зручний та швидкий пошук вакансій, персоналізований вибір та максимальний захист даних.

Висновки до розділу 1

Підсумовуючи, вебсайти пошуку роботи стали важливим інструментом для шукачів роботи та роботодавців в Україні. Jooble.org.ua, Work.ua, rabota.ua та dou.ua є одними з популярних сайтів пошуку роботи в Україні, які пропонують ряд функцій і послуг, щоб допомогти користувачам знайти відповідну роботу та працівників.

Вебсайти пошуку роботи стали важливим інструментом для шукачів роботи та роботодавців в Україні. Jooble.org.ua, work.ua та dou.ua є одними з популярних сайтів пошуку роботи в Україні, які пропонують ряд функцій і послуг, щоб допомогти користувачам знайти відповідну роботу та працівників.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення

Для проведення аналізу та визначення вимог до розроблюваного ПЗ необхідно:

- вивчити функціональні вимоги до сайту пошуку роботи, які мають найбільший пріоритет для користувачів;
- оцінити роботу схожих сайтів пошуку роботи, зокрема, їхні можливості, переваги та недоліки;
- визначити ключові функції та особливості, які має містити розроблюваний сайт пошуку роботи;
- розглянути технічні вимоги до розробки сайту з використанням Vue.js, включаючи вимоги до швидкодії та безпеки даних;
- встановити системні вимоги до розробки та визначити необхідні інструменти та технології, необхідні для реалізації проєкту;
- вивчити вимоги до масштабування та розширення сайту в майбутньому.

Керуючись ціллю та предметну областю додатку, можна виявити такі вимоги:

- система реєстрації та авторизації користувачів, що дозволяє користувачам створювати облікові записи та керувати ними;
- можливість розміщувати та керувати резюме в обліковому записі користувача;
- зберігати цікаві вакансії в обліковому записі користувача;
- розширені пошукові фільтри, що дозволяють користувачам шукати вакансії за різними критеріями, такими як місцезнаходження, галузь, тип роботи, зарплата та рівень досвіду;
- класифікація вакансій для полегшення навігації та організації;

- сумісність з різними браузерами та пристроями, що забезпечує доступність для всіх користувачів;
- зручний та інтуїтивно зрозумілий інтерфейс, що полегшує користувачам навігацію та використання вебсайту.
- сумісність із різними форматами файлів, такими як PDF і Word, для завантаження резюме та опису роботи;
- інтеграція з платформами соціальних мереж для обміну списками вакансій і підвищення видимості вебсайту;
- регулярне оновлення та обслуговування для забезпечення функціональності та безпеки вебсайту.

2.1.1 Функціональні та нефункціональні вимоги

Функціональні вимоги:

- система реєстрації та авторизації користувачів, що дозволяє користувачам створювати облікові записи та керувати ними;
- можливість розміщувати та керувати резюме в обліковому записі користувача;
- зберігати цікаві вакансії в обліковому записі користувача;
- розширені пошукові фільтри, що дозволяють користувачам шукати вакансії за різними критеріями, такими як місцезнаходження, галузь, тип роботи, зарплата та рівень досвіду;
- класифікація вакансій для полегшення навігації та організації;
- система керування заявками на роботу, що дозволяє користувачам подавати заявки на роботу та відстежувати статус своїх заявок;
- система розміщення та керування вакансіями для роботодавців, що дозволяє їм створювати та керувати списками вакансій на вебсайті;
- інтеграція з платформами соціальних мереж для обміну списками вакансій і підвищення видимості сайту;
- система сповіщень електронною поштою, надаючи користувачам оновлення про вакансії та статуси заявок.

Нефункціональні вимоги:

- зручність використання. Сайт повинен мати зручний та інтуїтивно зрозумілий інтерфейс, що полегшує користувачам навігацію та використання вебсайту;
- ефективність. Має бути швидким і чуйним, надаючи користувачам швидкий доступ до списків вакансій і результатів пошуку;
- надійність. Має бути надійним, з мінімальними простоями та помилками, забезпечуючи користувачам доступ до списків вакансій і подання заявок на роботу без перерв;
- безпека. Сайт повинен бути безпечним, захищати дані користувачів і запобігати несанкціонованому доступу або витоку даних.
- сумісність. Повинен бути сумісний з різними браузерами та пристроями, забезпечуючи доступність для всіх користувачів;
- масштабованість. Має бути масштабованим, здатним обробляти зростаючий трафік і активність користувачів без проблем із продуктивністю;
- ремонтпридатність. Вебсайт має бути простим в обслуговуванні та оновленні, забезпечуючи регулярне оновлення та технічне обслуговування для забезпечення функціональності та безпеки вебсайту.

2.1.2 Моделі елементів програмного додатку у нотації UML

Для того, щоб побудувати діаграми потрібно визначити інтерфейси та структури даних, які будуть використовуватися для обміну інформацією між компонентами. На діаграмі послідовності, рисунок 2.1, користувач взаємодіє з вебсайтом пошуку роботи, відвідуючи сайт, вводячи критерії пошуку та натискаючи на оголошення про роботу. Потім пошукова система повертає результати пошуку на вебсайт, які відображаються користувачеві. Потім користувач подає заявку на роботу, заповнюючи форму заявки та надсилаючи її через сайт. Потім він відображає повідомлення про підтвердження для користувача.

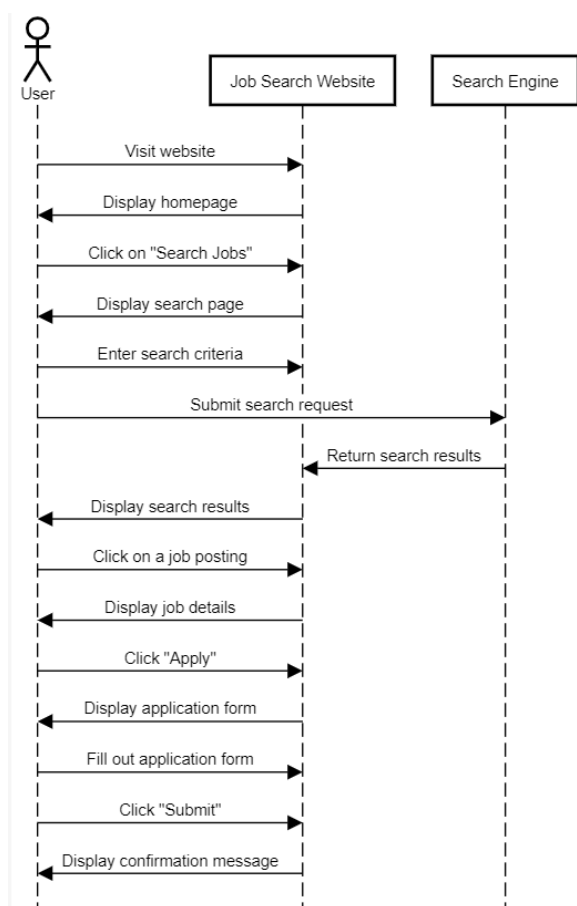


Рисунок 2.1 – Діаграма послідовностей взаємодії користувача з сайтом

Джерело: розроблено автором

Перегляд списків вакансій: користувачі можуть переглядати списки вакансій на вебсайті, які відображаються на основі різних критеріїв, таких як місцезнаходження, галузь, тип роботи, зарплата та рівень досвіду.

Пошук списків вакансій: користувачі можуть шукати списки вакансій за допомогою фільтрів розширеного пошуку, які дозволяють фільтрувати за різними критеріями, включаючи місцезнаходження, галузь, тип роботи, зарплату та рівень досвіду.

Зберегти списки вакансій: користувачі можуть зберігати списки вакансій, які їх цікавлять, у своєму обліковому записі для подальшого використання.

Подати заявку на роботу: користувачі можуть подати заявку на отримання вакансій безпосередньо на вебсайті, що передбачає заповнення форми заявки на роботу та подання свого резюме.

Створення облікових записів роботодавця та керування ними: роботодавці можуть створювати облікові записи та керувати ними, що включає створення та керування списками вакансій, перегляд заявок і зв'язок із потенційними кандидатами.

Загалом діаграма варіантів використання показує основну взаємодію між користувачем і вебсайтом пошуку роботи, включаючи перегляд і пошук списків вакансій, збереження списків вакансій, що цікавлять, подання заявок на роботу та створення облікових записів роботодавця та керування ними.

Діаграма варіантів використання, рисунок 2.2, показує різні взаємодії між користувачем і вебсайтом пошуку роботи. А саме сценарій використання сайту при пошуку вакансій, коли користувач відвідує сайт з цієї ціллю. Також передбачено сценарій створення персонального профілю для релевантних пропозицій користувачу за певними критеріями пошуку.

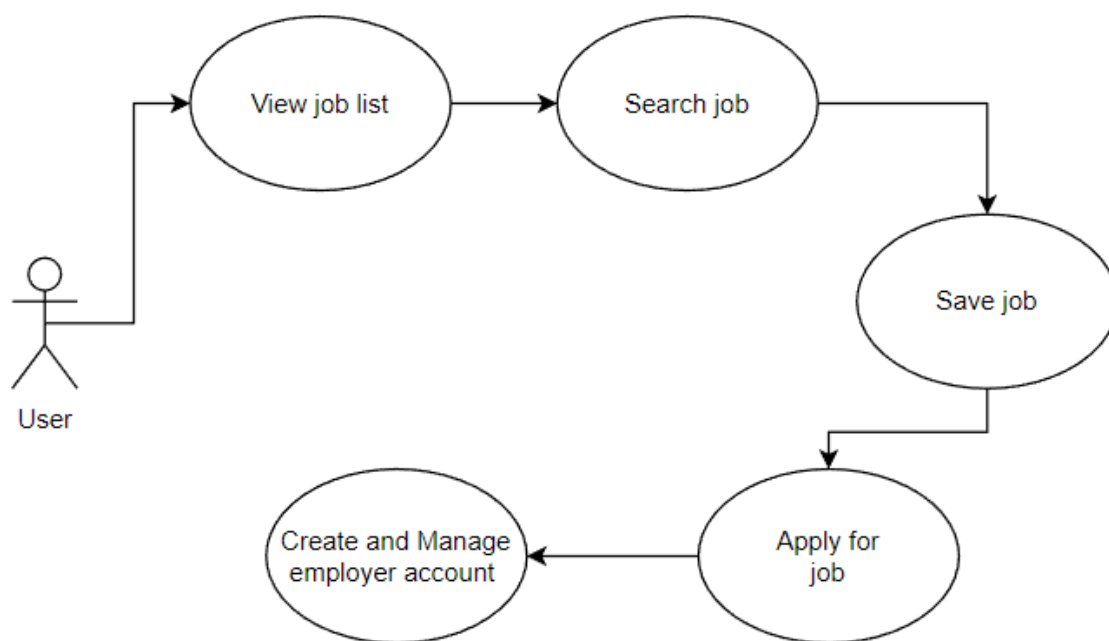


Рисунок 2.2 – Діаграма варіантів використання

Джерело: розроблено автором

2.1.3 Створення UX/UI-дизайну

Щоб створити дизайн UX/UI для вебсайту пошуку роботи, необхідно врахувати кілька факторів:

- дослідження користувачів: дизайн UX/UI має ґрунтуватися на глибокому розумінні потреб, уподобань і поведінки цільової аудиторії. Цього можна досягти шляхом дослідження користувачів, включаючи інтерв'ю, опитування та тестування зручності використання;

- інформаційна архітектура: інформаційна архітектура вебсайту повинна бути розроблена таким чином, щоб надавати користувачам легкий доступ до списків вакансій і пошукових фільтрів, а також до функцій керування обліковими записами користувачів. Це передбачає розробку навігації вебсайту, організації вмісту та потоків користувачів;

- дизайн інтерфейсу користувача. UI користувача має бути візуально привабливим, із чітким і послідовним макетом, типографікою, кольоровою схемою та зображеннями. Дизайн інтерфейсу користувача також має віддавати перевагу зручності використання, гарантуючи, що вебсайт простий у використанні, навігації та розумінні;

- дизайн взаємодії. Він має бути інтуїтивно зрозумілим, із чітким і зрозумілим відгуком щодо дій користувача, таких як подання заявки на роботу чи збереження списку вакансій. Це передбачає розробку компонентів інтерфейсу користувача, таких як кнопки, форми та меню;

- створення прототипів і тестування. Слід використовувати для вдосконалення дизайну UX/UI, гарантуючи, що він відповідає потребам цільової аудиторії та забезпечує безперебійну взаємодію з користувачем. це передбачає створення інтерактивних прототипів і проведення тестування зручності використання з цільовими користувачами.

Загалом, для створення успішного дизайну UX/UI вебсайту пошуку роботи вимагає глибокого розуміння потреб цільової аудиторії, а також орієнтованого на

користувача підходу до дизайну, який надає перевагу зручності використання, доступності та візуальній привабливості. Дизайн зображено на рисунках 2.3 – 2.5.

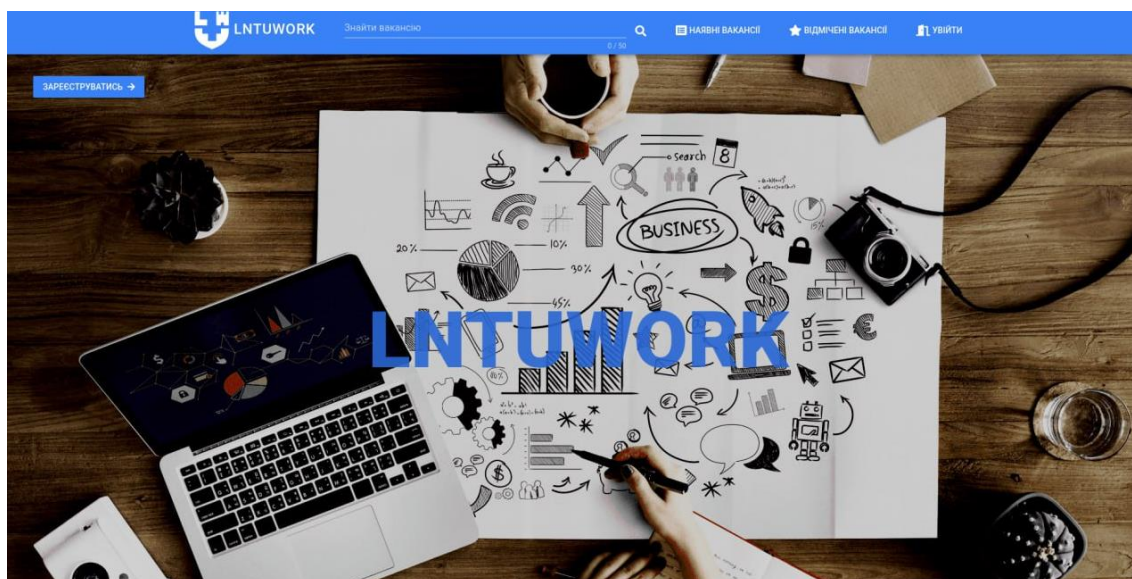


Рисунок 2.3 – Головна сторінка сайту

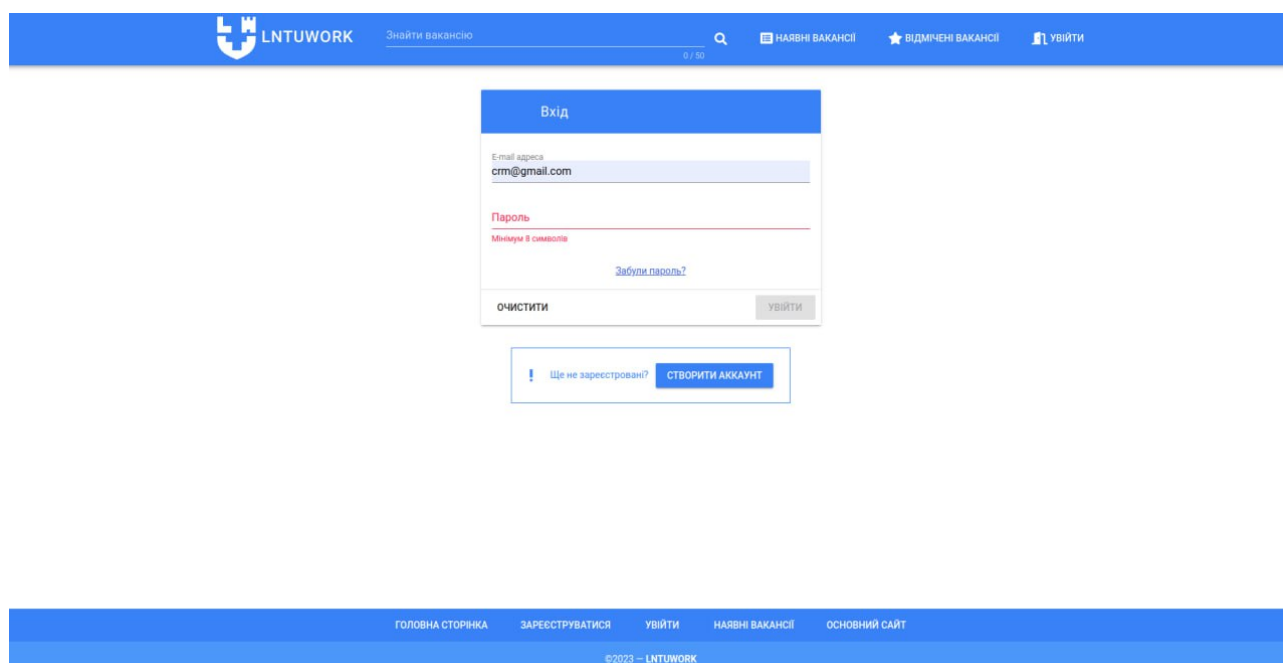


Рисунок 2.4 – Сторінка авторизації

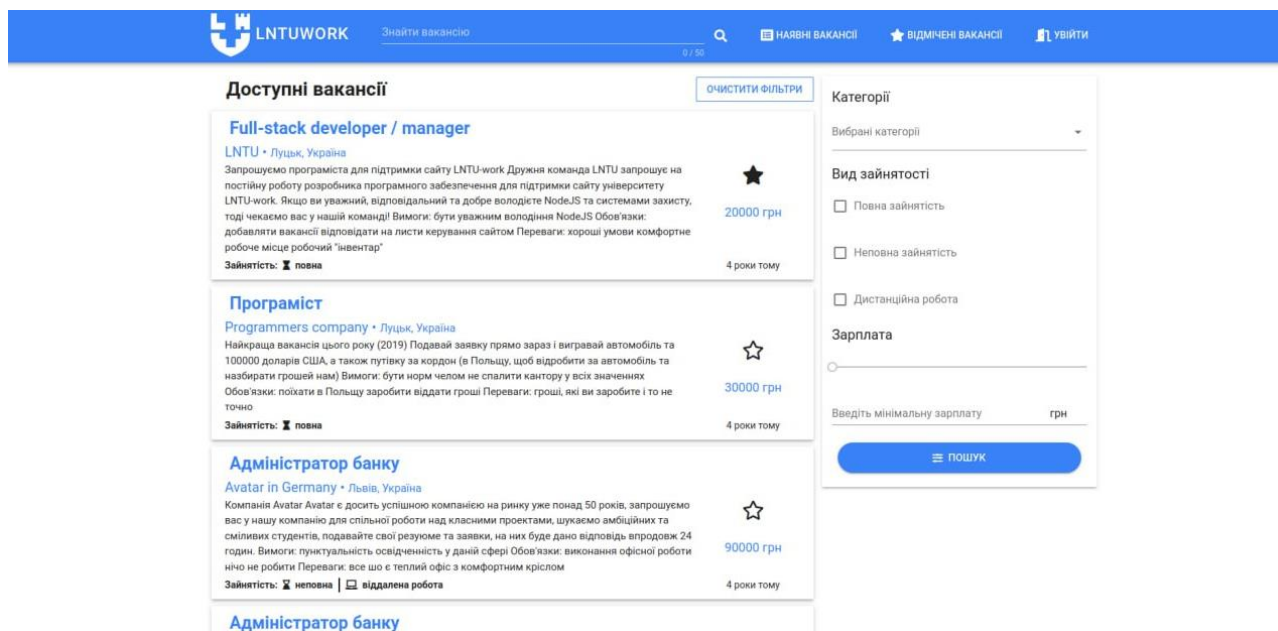


Рисунок 2.5 – Сторінка вакансій

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Вибір правильних технологій для вебсайту пошуку роботи залежить від низки факторів, у тому числі функціональності та функцій, які потрібні сайту, потреб у продуктивності та масштабованості, досвіду команди розробників і знайомства з різними технологіями, а також бюджету та графіку для демонструвати.

Ось, що потрібно аби враховувати при виборі технологій для вебсайту пошуку роботи.

UI: інтерфейс вебсайту відповідатиме за представлення інтерфейсу користувача та взаємодії з користувачем. До популярних інтерфейсних технологій належать HTML, CSS і JavaScript, а також фреймворки. Конкретні вимоги до вебсайту, наприклад потребу в оновленнях у реальному часі або складну взаємодію з користувачем.

Серверна частина вебсайту обслуговуватиме логіку, зберігання даних і керування БД. До популярних серверних технологій належать PHP, Python, Ruby та Node.js, а також такі фреймворки, як Laravel, Django, Ruby on Rails та Express.js.

Технології баз даних: вебсайт пошуку роботи, потребуватиме зберігання значної кількості даних, таких як оголошення про роботу, облікові записи користувачів і матеріали для вакансій.

Вибір використання Firebase як рішення для зберігання даних для розробленого вебсайту базувався на кількох факторах. По-перше, Firebase надає комплексну та масштабовану серверну інфраструктуру, яка спеціально розроблена для веб та мобільних додатків. Він пропонує низку послуг, включаючи базу даних у реальному часі, автентифікацію, зберігання файлів і хостинг, які необхідні для створення динамічної та інтерактивної платформи пошуку роботи.

По-друге, функція бази даних Firebase у реальному часі забезпечує безперебійну синхронізацію даних між клієнтом і сервером, забезпечуючи миттєве оновлення та співпрацю в реальному часі [14]. Це особливо вигідно для вебсайту пошуку роботи, де списки вакансій і профілі користувачів мають постійно оновлюватися та мати доступ до них одночасно для кількох користувачів.

Крім того, служба автентифікації Firebase забезпечує безпечний і надійний механізм автентифікації користувачів, гарантуючи захист даних і облікових даних користувача. Це вкрай важливо для збереження конфіденційності та безпеки інформації користувачів на платформі пошуку роботи [15].

Рішення вибрати Firebase замість AWS Amplify як рішення для зберігання даних для розробленого вебсайту базувалося на кількох міркуваннях.

По-перше, Firebase пропонує широкий набір функцій, спеціально розроблених для розробки веб-додатків і мобільних додатків. Він забезпечує повну інтеграцію різних служб, таких як база даних у реальному часі, автентифікація, хостинг і зберігання, які необхідні для створення платформи пошуку роботи. Універсальний характер Firebase спрощує процес розробки та зменшує потребу в управлінні багатьма службами від різних постачальників.

По-друге, функція бази даних Firebase у реальному часі дозволяє миттєво синхронізувати дані між клієнтом і сервером, забезпечуючи оновлення в реальному часі та співпрацю. Це особливо вигідно для вебсайту пошуку роботи, де своєчасні оновлення та сповіщення мають вирішальне значення. Можливості Firebase у режимі реального часу забезпечують безперебійну та швидку роботу користувачів.

Крім того, Firebase пропонує зручне та інтуїтивно зрозуміле середовище розробки з добре задокументованим API і великою документацією. Це полегшує розробникам вивчення та впровадження Firebase у свої проєкти, скорочуючи час і зусилля на розробку.

Хостинг і розгортання: потрібно враховувати потреби вебсайту в хостингу та розгортанні, балансуванні навантаження та автоматичному масштабуванні. Серед технологій хостингу та розгортання – Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform і Heroku і Firebase.

Безпека та продуктивність. Це включає використання сертифікатів SSL для безпечного зв'язку, впровадження передових методів безпеки, таких як хешування паролів і перевірка введених даних, а також оптимізацію продуктивності за допомогою таких методів, як кешування та мережі доставки вмісту (CDN).

Загалом, вибір правильних технологій для вебсайту пошуку роботи має вирішальне значення для забезпечення того, щоб сайт був функціональним, ефективним і безпечним.

Vue.js є популярною платформою JavaScript з відкритим кодом, яка використовується для створення інтерфейсів користувача та односторінкових програм (SPA). Він був створений Еваном Ю в 2014 році і здобув значну популярність серед розробників завдяки своїй простоті, гнучкості та зручності використання [16].

Для розробки сайту було вибрано фреймворк Vue.js. Ось деякі основні функції та переваги використання Vue.js для розробки вебсайту пошуку роботи:

- реактивне зв'язування даних. Vue.js використовує підхід реактивного зв'язування даних, що означає, що зміни, внесені до даних програми, автоматично

відображаються в інтерфейсі користувача. Це полегшує керування складними інтерфейсами користувача та синхронізацію даних із станом програми [17];

- компонентна архітектура. Фреймворк базується на компонентній архітектурі, що означає, що складні інтерфейси користувача можна розбити на менші багаторазово використовувані компоненти. Це полегшує керування кодовою базою та підтримує узгодженість у програмі;

- простий у використанні. Vue.js. Він має простий API і документацію, що дозволяє розробникам швидко розпочати роботу;

- висока продуктивність. Vue.js розроблений як високопродуктивний, з невеликим розміром і високою швидкістю рендерингу. Він використовує віртуальну DOM (об'єктну модель документа), яка допомагає оптимізувати процес візуалізації та покращити продуктивність;

- велика та активна спільнота: Vue.js має велику та активну спільноту розробників, що означає, що існує багато ресурсів та інструментів, доступних для навчання та розвитку. Це також означає, що існує широкий спектр плагінів і компонентів, доступних для розширення функціональності фреймворку.

Таким чином, Vue.js – це потужний фреймворк JavaScript (рис. 2.6), який можна використовувати для створення високоякісних інтерфейсів користувача та SPA. Його простота, гнучкість і продуктивність роблять його чудовим вибором для розробки вебсайту пошуку роботи.



Рисунок 2.6 – Логотип Vue.js [17]

Для розробки UI компонентів, стилізації сайту було вибрано vuetify бібліотеку. Вона була розроблена спеціально під вибраний фреймоврк.

Vuetify – популярна бібліотека інтерфейсу користувача, створена спеціально для програм Vue.js. Він розроблений, щоб надати розробникам набір повторно використовуваних компонентів, які можна використовувати для швидкого створення красивих і гнучких інтерфейсів користувача [18].

Ось деякі основні функції та переваги використання Vuetify для розробки вебсайту пошуку роботи:

- Vuetify створено з використанням принципів матеріального дизайну Google, який забезпечує послідовний і знайомий вигляд усіх компонентів. Це полегшує користувачам навігацію та взаємодію з програмою;

- компоненти Vuetify створені для адаптації до різних розмірів екрана та роздільної здатності. Це важливо для того, щоб вебсайт був доступним і придатним для використання на різних пристроях;

- простий у використанні та налаштуванні. Vuetify надає широкий спектр готових компонентів, які можна легко налаштувати відповідно до дизайну та

бренду вебсайту. Він також містить вичерпну документацію та набір інструментів, які полегшують розробникам розпочати роботу та налаштувати компоненти за потреби;

- велика та активна спільнота. Vuetify має велику та активну спільноту розробників, що означає, що існує багато ресурсів та інструментів, доступних для навчання та розвитку. Це також означає, що існує широкий спектр плагінів і компонентів, доступних для розширення функціональності бібліотеки;

- продуктивність. Vuetify розроблений як високопродуктивний, з невеликим розміром і оптимізованою швидкістю візуалізації. Це допомагає гарантувати швидке завантаження вебсайту та забезпечує зручну взаємодію з користувачем.

На рисунку 2.7 зображено головну сторінку Vuetify, яка демонструє багатофункціональний та естетичний дизайн інтерфейсу. Бібліотека надає широкі можливості для створення привабливого та сучасного вигляду веб-додатків, з великою кількістю готових компонентів та стилів. У проекті пошуку роботи, використання Vuetify доцільно, оскільки він дозволяє швидше розробляти інтерфейс, забезпечувати єдність стилів і зручність взаємодії користувача.

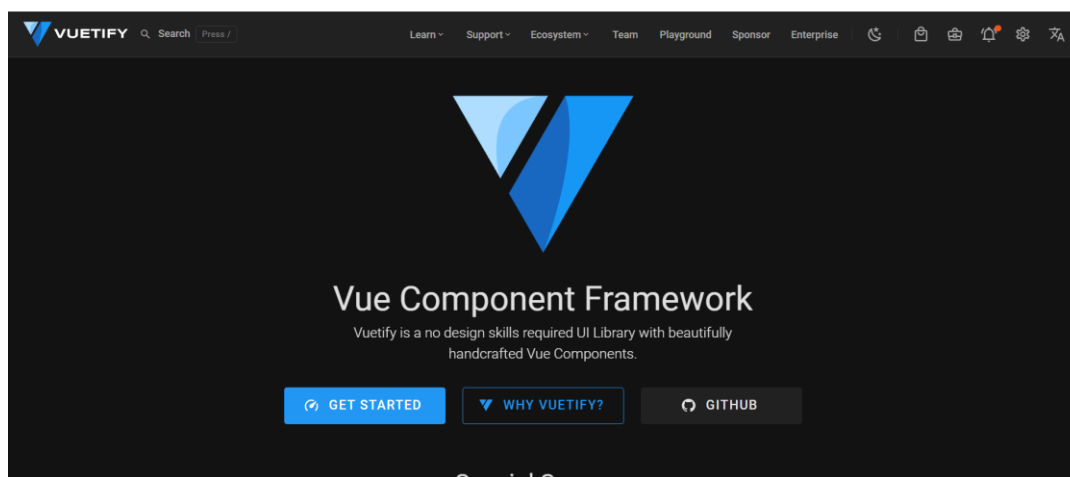


Рисунок 2.7 – Головна сторінка Vuetify

Vuetify – це потужна та гнучка бібліотека інтерфейсу користувача, яку можна використовувати для створення красивих і чутливих інтерфейсів користувача для вебсайту пошуку роботи. Його принципи матеріального дизайну, адаптивний

дизайн, простота використання та продуктивність роблять його чудовим вибором для розробників, які хочуть створити високоякісну взаємодію з користувачем [19].

Висновки до розділу 2

Було досліджено вимоги та специфікації для розробки вебсайту пошуку роботи з використанням фреймворку Vue.js та бібліотеки інтерфейсу користувача Vuetify. Визначив функціональні та нефункціональні вимоги до вебсайту, а також міркування дизайну UX/UI, які слід взяти до уваги. потім перейшов до створення різних діаграм UML, включаючи діаграму варіантів використання, діаграму послідовності та діаграму стану, що допомогло ще більше уточнити вимоги та зрозуміти поведінку системи. Поставлено завдання по виконанню роботи.

Також було описано важливість вибору правильних технологій для проєкту та висвітлили деякі ключові особливості та переваги використання Vue.js і Vuetify для розробки вебсайту. Vue.js було обрано через його простоту, гнучкість і легкість використання, тоді як Vuetify надає багатий набір компонентів і інструментів для створення красивих і чуйних інтерфейсів користувача.

РОЗДІЛ 3

РОЗРОБКА САЙТУ ДЛЯ ПОШУКУ РОБОТИ ДЛЯ СТУДЕНТІВ ТА ВИПУСКНИКІВ ЛНТУ

3.1 Практична реалізація об'єкта проєктування

У практичній реалізації об'єкта проєктування, орієнтуючись на тему сайту пошуку роботи для студентів і випускників, проявляється кілька ключових аспектів. Цей розділ роботи містить опис процесу написання коду та використання певної мови програмування в середовищі розробки програмного забезпечення.

Спершу була розроблена реєстрація користувача, код описано у лістингу 3.1.

Лістинг 3.1 – Реєстрація користувача з Firebase

```

signUpByUser() {
  this.registrationDialogLoading = true;
  firebase
    .auth()
    .createUserWithEmailAndPassword(this.userEmail,
this.userPassword)
    .then(() => {
      this.$store.dispatch("registerUser", {
        name: `${this.name} ${this.surname}`,
        photo: this.userPhoto,
        phone: this.userPhoneNumber,
        email: this.userEmail,
        password: this.userPassword,
      });
    })
    .catch((error) => {
      this.registrationDialogLoading = false;
      if (error.message) {
        switch (error.message) {
          case "The email address is already in use by another
account.": {
            this.errorText =
              "Вказана електронна адреса уже використовується
іншим користувачем.";
            this.snackbar = true;
            break;
          }
          case "A network error (such as timeout, interrupted
connection or unreachable host) has occurred.": {
            this.errorText = "Помилка мережі, повторіть спробу";
            this.snackbar = true;
          }
        }
      }
    })
  }
}

```

Продовження лістингу 3.1

```

        break;
    }
    default: {
        if (error.message) {
            this.errorText = error.message;
            this.snackbar = true;
        }
        break;
    }
}
} else alert("Виникла помилка");
});

```

кінець лістингу 3.1

Наданий фрагмент коду представляє функцію під назвою «signUpByUser()», яка є частиною практичної реалізації сайту пошуку роботи для студентів і випускників. Ця функція відповідає за обробку процесу реєстрації користувачів під час авторизації на сайті. Функціональність коду:

Функція починається із встановлення стану завантаження для діалогового вікна реєстрації, що вказує на те, що процес реєстрації триває.

Він використовує бібліотеку автентифікації Firebase для створення нового облікового запису користувача з указаною адресою електронної пошти та паролем. Метод «createUserWithEmailAndPassword()» викликається з параметрами «this.userEmail» і «this.userPassword».

Якщо створення користувача пройшло успішно, функція надсилає дію під назвою «registerUser» за допомогою Vuex (бібліотека керування станом для Vue.js). Ця дія відповідає за реєстрацію користувача та збереження його даних, таких як ім'я, фото, номер телефону, адреса електронної пошти та пароль.

У разі виникнення помилки під час створення користувача виконується блок «catch()». Він обробляє різні сценарії помилок на основі повідомлення про помилку, отриманого від Firebase.

Якщо повідомлення про помилку вказує на те, що адреса електронної пошти вже використовується іншим обліковим записом, воно встановлює відповідне повідомлення про помилку в «this.errorText» і запускає панель швидкого доступу

(компонент сповіщення), щоб відобразити повідомлення про помилку для користувача.

Якщо повідомлення про помилку вказує на помилку мережі, воно встановлює загальне повідомлення про помилку, пов'язане з проблемами мережі.

Для будь-якого іншого повідомлення про помилку він встановлює повідомлення про помилку в «this.errorText» і запускає панель швидкого доступу для відображення помилки. Якщо конкретного повідомлення про помилку не надано, відображається загальне сповіщення з повідомленням «Виникла помилка». Використовуючи такий функціонал у дипломної роботи, демонструються навички практичного застосування в обробці реєстрації користувачів, інтеграції з автентифікацією Firebase та ефективного керування сценаріями помилок, щоб забезпечити безперебійну роботу користувача на сайті пошуку роботи. На рисунку 3.1 зображено реєстрацію користувача.

The image shows a web registration form for 'LNTUWORK'. The header bar is blue with the site logo, navigation links ('Знайти вакансію', 'НАЯВНІ ВАКАНСІЇ', 'ВІДМІЧЕНІ ВАКАНСІЇ'), and a login button ('Увійти'). The registration form itself is a white box with a blue header 'Реєстрація'. It contains the following fields: 'Ім'я' (0/20), 'Прізвище' (0/20), a photo upload area with a silhouette icon and the text 'Фото (натисніть для завантаження)', 'Номер телефону' (0/10), 'E-mail', 'Введіть пароль' (0), and 'Підтвердіть пароль' (0). At the bottom, there is a checkbox for 'Я даю згоду на обробку персональних даних.' and two buttons: 'ОЧИСТИТИ' and 'СТВОРИТИ АКАУНТ'.

Рисунок 3.1 – Форма реєстрації

На рисунку 3.2 зображено авторизація користувача, якщо користувач відвідує сайт перший раз та зацікавлений у отриманні відповідних вакансій лише для нього, то може створити профіль.

Рисунок 3.2 – Вхід у профіль користувача

Наведений фрагмент коду, лістинг 3.2, представляє асинхронну функцію під назвою «registerUser», яка є частиною практичної реалізації сайту пошуку роботи для студентів і випускників. Ця функція відповідає за реєстрацію користувача та збереження його інформації в базі даних. Опис функціоналу даної функції наступний:

- функція отримує об’єкт «payload», що містить такі властивості, як «name», «phone», «email», «password» і «photo». Для отримання цих значень використовується деструктурування об’єктів;

- функція починається із внесення мутації під назвою «OPEN_REGISTER» зі значенням «true» для оновлення стану. Ймовірно, ця мутація контролює видимість пов’язаного з реєстрацією компонента або діалогового вікна;

- потім створюється посилання на сховище Firebase і вказується місце для зберігання фотографії профілю користувача. Властивість «photo.imageFireName» використовується для створення шляху зберігання;

- функція завантажує фотографію користувача в сховище Firebase, викликаючи «putString()» в об’єкті «fileStorageRef». Він використовує

«photo.imageUrl», як джерело даних, яке вважається URL адресою даних, що представляє зображення;

- усередині блоку «try-catch» функція отримує URL адресу завантаження завантаженого зображення з Firebase Storage за допомогою «getDownloadURL()» у відповідному посиланні на сховище;

- функція оновлює профіль поточного користувача в автентифікації Firebase, викликаючи «updateProfile()» на «firebase.auth().currentUser». Він встановлює «displayName» на ім'я користувача, а «photoURL» – на отриману URLадресу зображення;

- далі функція генерує унікальний ідентифікатор для користувача, фіксує поточну дату за допомогою бібліотеки Moment.js і зберігає інформацію про користувача в базі даних Firestore. Дані зберігаються в колекції «користувачі» зі згенерованим ідентифікатором як ідентифікатором документа;

- функція вносить мутацію під назвою «GLOBAL_USER» з об'єктом «payload» для оновлення стану. Ймовірно, ця мутація встановлює глобальний об'єкт користувача, роблячи його доступним у всій програмі;

- код мови автентифікації Firebase встановлено на «uk» (українська), щоб вказати мову для підтвердження електронної пошти;

- функція надсилає поточному користувачеві посилання для підтвердження електронної пошти за допомогою «sendEmailVerification()» на «firebase.auth().currentUser»;

- якщо під час процесу реєстрації виникає будь-яка помилка, вона виводиться на консолі за допомогою «console.log(err)».

Лістинг 3.2 – Діспатч для реєстрації

```
async registerUser(context, payload) {
  const { name, phone, email, password, photo } = payload;
  await context.commit(OPEN_REGISTER, true);
  const fileStorageRef =
    firebase.storage().ref().child(`profilePhotos/${photo.imageFireName}
`);
  await fileStorageRef.putString(photo.imageUrl, 'data_url');
```

Продовження лістингу 3.2

```

    try {
      const imageUrl = await
firebase.storage().ref().child(`profilePhotos/${photo.imageFireName}
`).getDownloadURL();
      await firebase.auth().currentUser.updateProfile({ displayName:
name, photoURL: imageUrl });
      const uniqueId = uniqueString().substr(0, 24);
      const currentDate = moment().format();
      await
firebase.firestore().collection('users').doc(uniqueId).set({
        id: uniqueId,
        registrationDate: currentDate,
        name,
        phone,
        email,
        password,
        photo: {
          imageName: photo.imageName,
          imageFireName: photo.imageFireName,
          imageUrl,
        },
      });
      await context.commit(GLOBAL_USER, payload);
      firebase.auth().languageCode = 'uk';
      await firebase.auth().currentUser.sendEmailVerification();
    } catch (err) {
      console.log(err);
    }
  }

```

кінець лістингу 3.2

У лістингу 3.3 продемонстровано функцію для завантаження вакансій. Фрагмент коду вище представляє функцію під назвою «loadVacancies()», яка відповідає за отримання та фільтрацію вакансій із бази даних Firestore. Його функціональність:

- функція отримує колекцію «вакансій» з БД Firestore і впорядковує їх за полем «placementDate» у порядку спадання;
- отримавши знімок колекції, функція очищає масиви «availableVacancies» і «allAvailableVacancies», щоб забезпечити свіжий набір даних;
- він перебирає кожен документ у колекції та додає його дані до масивів «availableVacancies» і «allAvailableVacancies»;

- якщо в URL-адресі є параметр запиту під назвою «SearchString», функція викликає метод «filterVacanciesByParameters()», який, ймовірно, виконує додаткову фільтрацію на основі параметрів пошуку;
- якщо в URL-адресі є параметр запиту під назвою «categories», функція фільтрує масив «availableVacancies» на основі вказаної категорії. Якщо значення параметра є рядком, воно відповідає вакансіям тієї самої категорії. Якщо це масив, він зіставляє вакансії з будь-якою категорією, наявною в масиві;
- якщо в URL-адресі є параметр запиту під назвою «employment», функція фільтрує масив «availableVacancies» на основі вказаного типу зайнятості. Він обробляє різні сценарії залежно від типу та довжини значення параметра;
- якщо в URL-адресі є параметр запиту під назвою «isRemote», функція фільтрує масив «availableVacancies», щоб включити лише віддалені вакансії, якщо значення параметра має значення «true»;
- якщо в URL-адресі є параметр запиту під назвою «minSalary», функція фільтрує масив «availableVacancies», щоб включити лише вакансії із зарплатою, що перевищує або дорівнює вказаній мінімальній зарплаті;
- функція встановлює значення «false» для змінної «activeLoadingSpin», щоб вказати, що процес завантаження завершено.

Лістинг 3.3 – Завантаження вакансій

```
loadVacancies() {
  firebase
    .firestore()
    .collection("vacancies")
    .orderBy("placementDate", "desc")
    .onSnapshot((collection) => {
      this.availableVacancies = [];
      this.allAvailableVacancies = [];
      collection.forEach((doc) => {
        this.availableVacancies.push(doc.data());
        this.allAvailableVacancies.push(doc.data());
      });
      if (this.$route.query.SearchString !== undefined) {
        this.filterVacanciesByParameters();
      }
      if (this.$route.query.categories !== undefined) {
```

Продовження лістингу 3.3

```

        if (typeof this.$route.query.categories === "string") {
            this.availableVacancies =
this.availableVacancies.filter(
                (vac) => vac.category ===
this.$route.query.categories);
        } else {
            this.availableVacancies =
this.availableVacancies.filter((vac) =>
                this.$route.query.categories.some((c) => c ===
vac.category)
            );
        }
    }
    if (this.$route.query.employment !== undefined) {
        if (typeof this.$route.query.employment === "string") {
            this.availableVacancies =
this.availableVacancies.filter((v) =>
                v.employment.some((e) => e ===
this.$route.query.employment)
            );
        } else if (this.$route.query.employment.length === 1) {
            this.availableVacancies =
this.availableVacancies.filter((v) =>
                v.employment.some((e) => e ===
this.$route.query.employment[0])
            );
        } else if (this.$route.query.employment.length === 2) {
            this.availableVacancies =
this.availableVacancies.filter(
                (v) => v.employment.length === 2);
        }
    }
    if (this.$route.query.isRemote !== undefined) {
        if (this.$route.query.isRemote) {
            this.availableVacancies =
this.availableVacancies.filter(
                (vac) => vac.isRemote
            );
        }
    }
    if (this.$route.query.minSalary !== undefined) {
        this.availableVacancies =
this.availableVacancies.filter(
            (vac) => vac.salary >= this.$route.query.minSalary
        );
    }
    this.activeLoadingSpin = false; this.page = 1;
}

```

кінець лістингу 3.3

На рисунку 3.3 продемонстровано результат функції для завантаження вакансій.

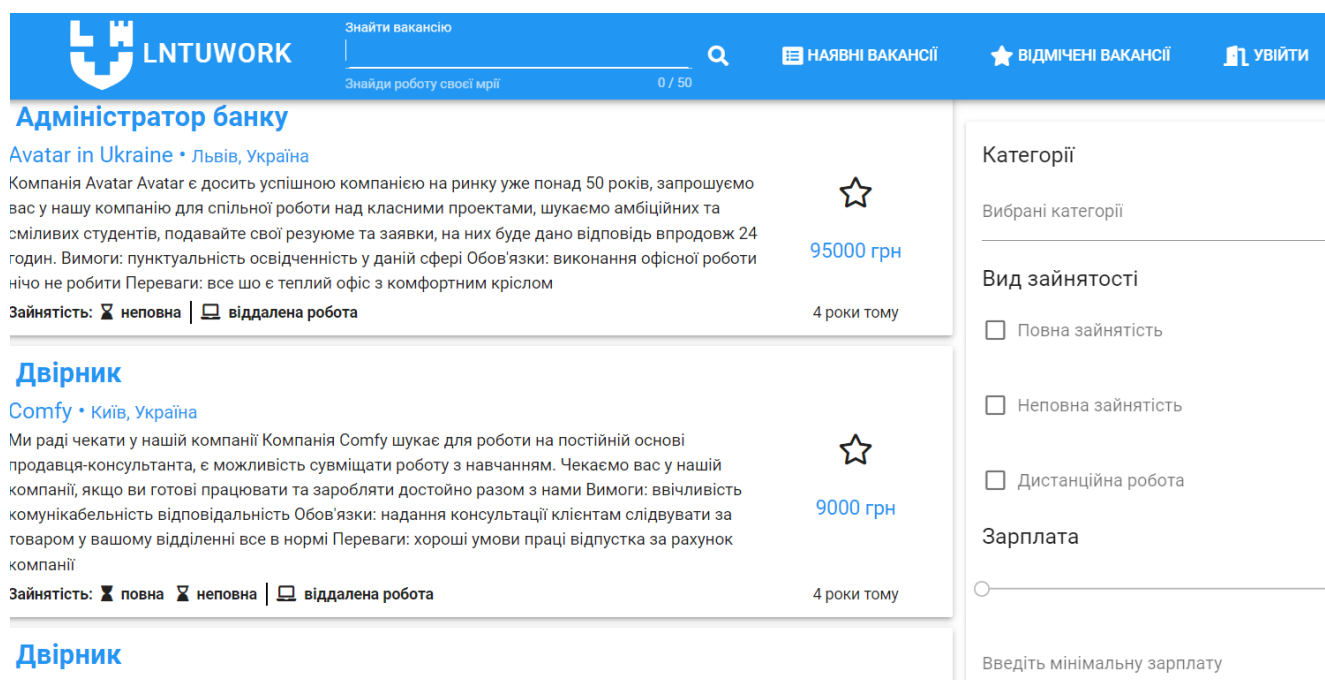


Рисунок 3.3 – Завантаження вакансій

На лістингу 3.4 продемонстровано функцію «`filterVacanciesByParameters()`», яка відповідає за фільтрацію доступних вакансій за заданими параметрами. Почнемо з розбору коду.

Функція отримує доступ до списку доступних вакансій «`this.availableVacancies`» і застосовує на ньому метод «`filter()`», який приймає умову для фільтрації. Умова передається у вигляді стрілкової функції з одним параметром «`vac`», що представляє кожну окрему вакансію.

Умова включає в себе порівняння декількох полів вакансії зі значенням, отриманим з URL-параметру «`SearchString`», який передається у «`this.$route.query.SearchString`».

Перша частина умови перевіряє, чи містить поле «`position`» назву посади, що задовольняє введеному рядку пошуку.

Друга частина умови перевіряє, чи містить поле «`company.name`» назву компанії, яка задовольняє введеному рядку пошуку.

Третя частина умови перевіряє, чи містить поле «location» місцезнаходження вакансії, яке задовольняє введеному рядку пошуку.

Четверта та п'ята частини умови перевіряють, чи містять заголовок «description.header» та текст «description.text» опису вакансії введений рядок пошуку.

Якщо яка-небудь з цих умов відповідає, то вакансія залишається у списку «this.availableVacancies». Якщо жодна з умов не відповідає, вакансія виключається зі списку. Таким чином, цей код забезпечує функціональність фільтрації вакансій за декількома параметрами, що вводяться користувачем, та забезпечує відображення лише тих вакансій, які задовольняють критеріям пошуку.

Лістинг 3.4 – Фільтрація вакансій

```
filterVacanciesByParameters() {
    this.availableVacancies = this.availableVacancies.filter(
        (vac) =>
            vac.position
                .toLowerCase()
                .search(this.$route.query.SearchString.toLowerCase())
            !== -1 ||
            vac.company.name
                .toLowerCase()
                .search(this.$route.query.SearchString.toLowerCase())
            !== -1 ||
            vac.location
                .toLowerCase()
                .search(this.$route.query.SearchString.toLowerCase())
            !== -1 ||
            vac.description.header
                .toLowerCase()
                .search(this.$route.query.SearchString.toLowerCase())
            !== -1 ||
            vac.description.text
                .toLowerCase()
                .search(this.$route.query.SearchString.toLowerCase())
            !== -1
        );
    },
```

кінець лістингу 3.4

На рисунку 3.4 відображено результат відфільтрованих вакансій по категорії та зарплаті.

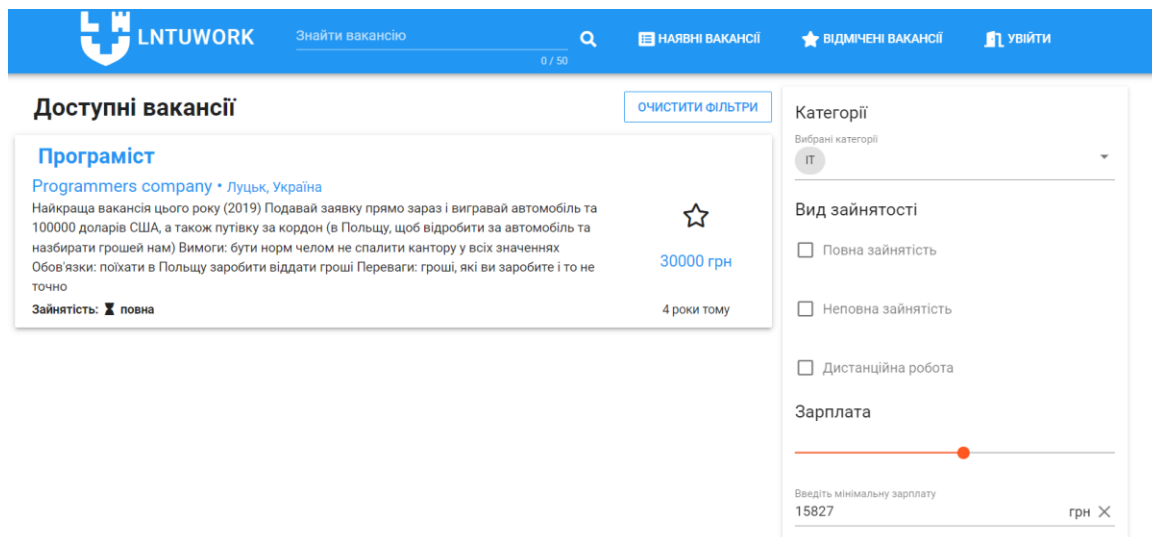


Рисунок 3.4 – Відфільтровані вакансії

Після знайдених, бажаних вакансій можна перейти по ним для перегляду детальної інформації про конкретну роботу (рис. 3.5). Описується посада, зарплата, опис, вимоги до кандидата та можна подати резюме. Щоб подати резюме, потрібно увійти в профіль або зареєструватись, якщо немає облікового запису.

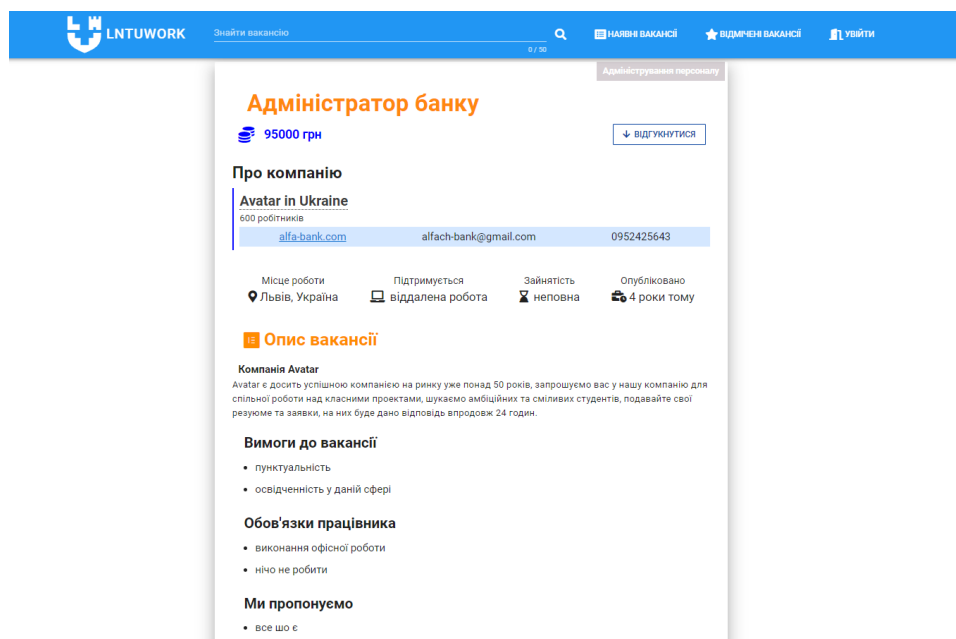


Рисунок 3.5 – Сторінка вакансії

На рисунку 3.6 показано повідомлення для входу, коли користувач натиснув, щоб подати резюме, якщо не авторизований. А якщо не існує профілю, то можна створити.

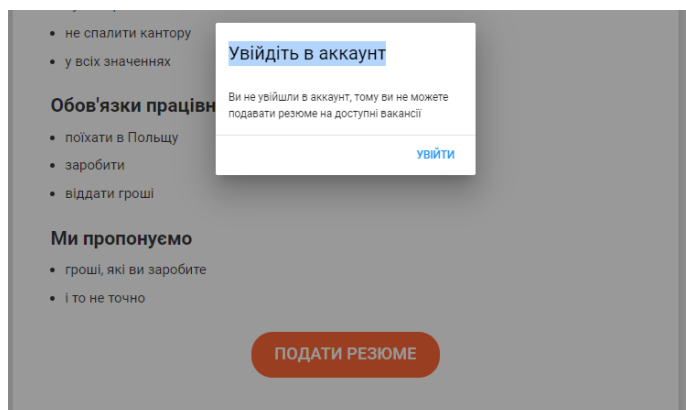


Рисунок 3.6 – Повідомлення для входу у профіль

На рисунку 3.7 показано, як заповняти форму реєстрації. Потрібно вписати: ім'я, прізвище, фото, номер телефону, імейл, пароль та його підтвердження.

Реєстрація

Ім'я

Яна

3 / 20

Прізвище

Цист

4 / 20

Фото (натисніть для завантаження)

li-ion-battery-18650-rechargeable-n1PQw2F-600.jpg

Номер телефону

(099) 234 - 3321

10 / 10

E-mail

yana00@gmail.com

Введіть пароль

12345678

8

Підтвердіть пароль

.....

8

☒ Я даю згоду на обробку персональних даних.

ОЧИСТИТИ

СТВОРИТИ АККАУНТ

Рисунок 3.7 – Реєстрація профілю

Після завершення реєстрації доступний профіль, де можна створити резюме. Потрібно ввести ПІБ, дату народження, місто проживання, адресу, посаду бажану, вид зайнятості, інформацію про освіту, період навчання, додаткову інформацію для роботодавця (рис. 3.8).

Рисунок 3.8 – Створити резюме

Також тепер можна подавати резюме на бажані вакансії. Якщо список резюме порожній, потрібно створити (рис. 3.9).

Рисунок 3.9 – Подача резюме на вакансію

Ще є важливою частиною цієї розробки – це завантаження нових вакансій, керування користувачами, вакансіями, відгуками, технічна підтримка, опублікованими резюме (рис. 3.10). Для всіх цих речей призначений адмін.

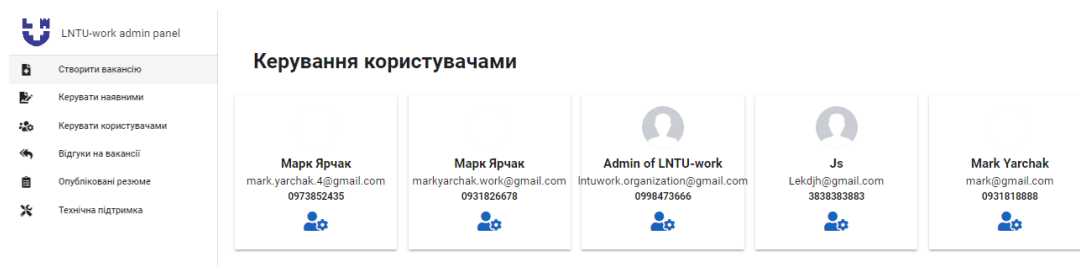


Рисунок 3.10 – Сторінка адміна

На рисунку 3.11 зображено вкладку зі створенням нової вакансії.

The screenshot shows the 'Створити вакансію' (Create vacancy) form. It includes a sidebar with the same menu items as in Figure 3.10. The main content area has a title 'Створити вакансію' and a blue notification bar that reads: 'Рекомендовано заповнити всі поля для кращого відображення вакансії після публікування на сайті' (Recommended to fill all fields for better vacancy display after publishing on the site). The form fields are:

- Посада (Job title)
- Введіть мінімальну зарплату (Enter minimum salary) with a 'грн' (UAH) unit indicator.
- Категорія (Category) with a dropdown arrow.
- Про компанію (About company) section:
 - Назва (Name)
 - Кількість працівників (Number of employees) with a 'робітників' (employees) unit indicator.
- Контакти (Contacts) section:
 - Доменне ім'я сайту (без 'https://') (Domain name of the site (without 'https://'))
 - Номер телефону (Phone number)
 - Електронна адреса (Email address)

At the bottom left, there is a 'Вийти з аккаунта' (Logout) link.

Рисунок 3.11 – Вкладка зі створенням вакансії

Заповнивши форму, рекомендується перевірити достовірність інформації про вакансію, тоді можна опублікувати (рис. 3.12).

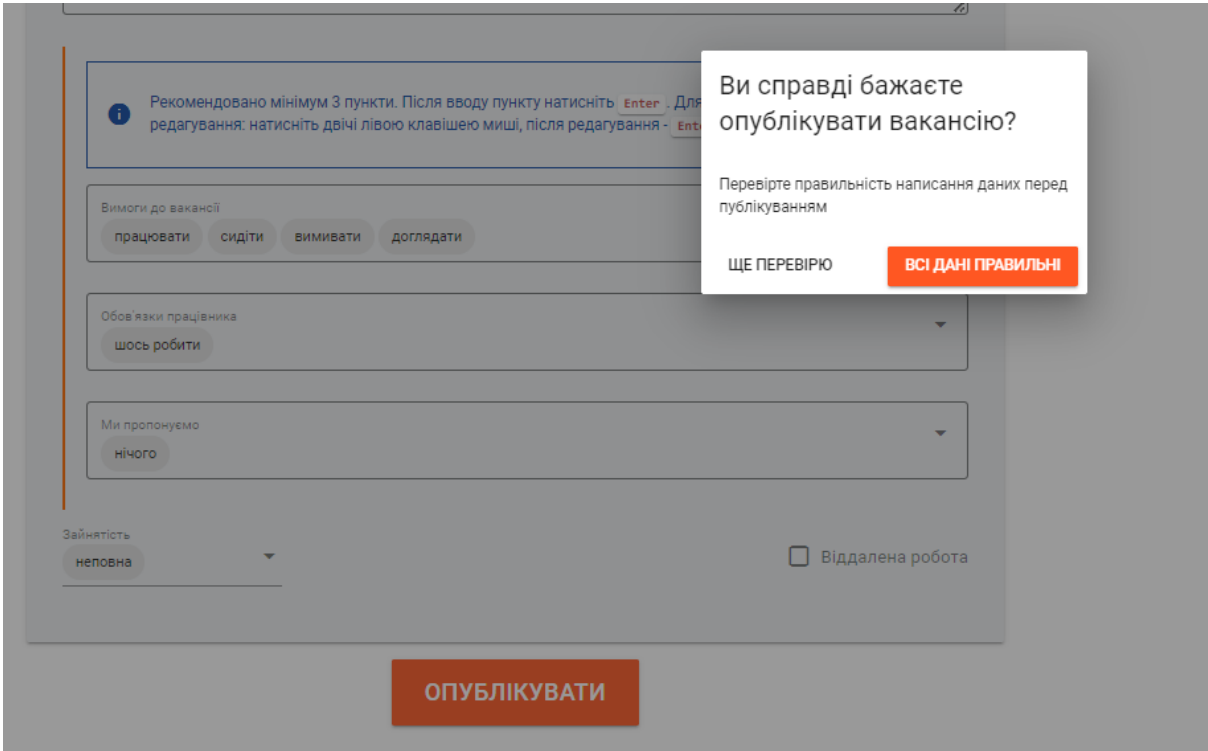


Рисунок 3.12 – Опублікування вакансії

Всіма наявними вакансіями можна керувати. Передбачається наступний функціонал: редагування, видалення (рис. 3.13).

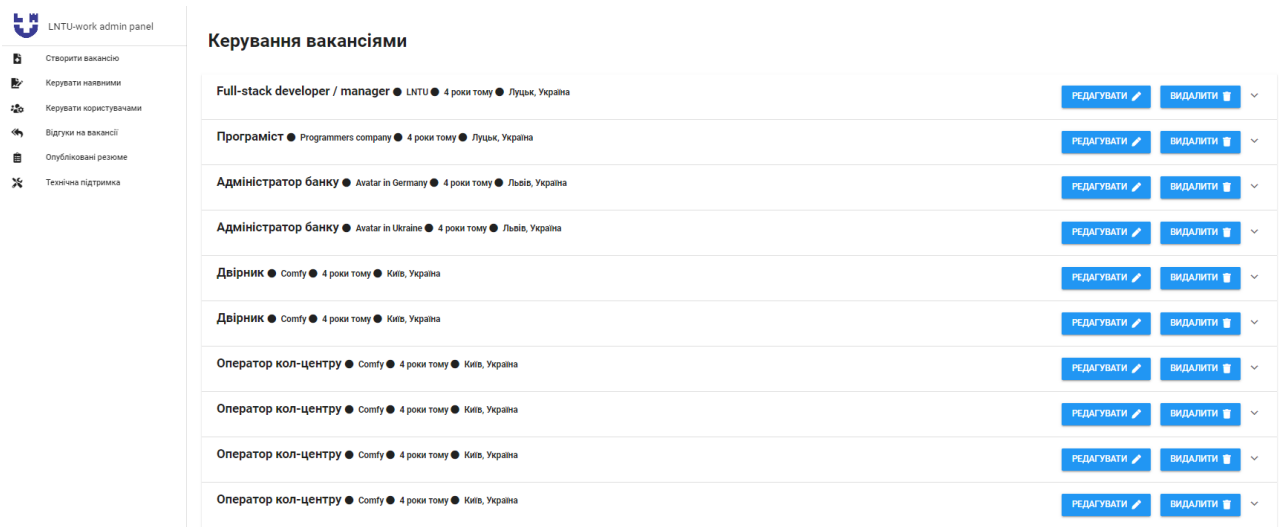


Рисунок 3.13 – Керування вакансіями

На рисунку 3.14 продемонстровано видалення вакансії. Для цього потрібно натиснути кнопку «Видалити» та підтвердити дію.

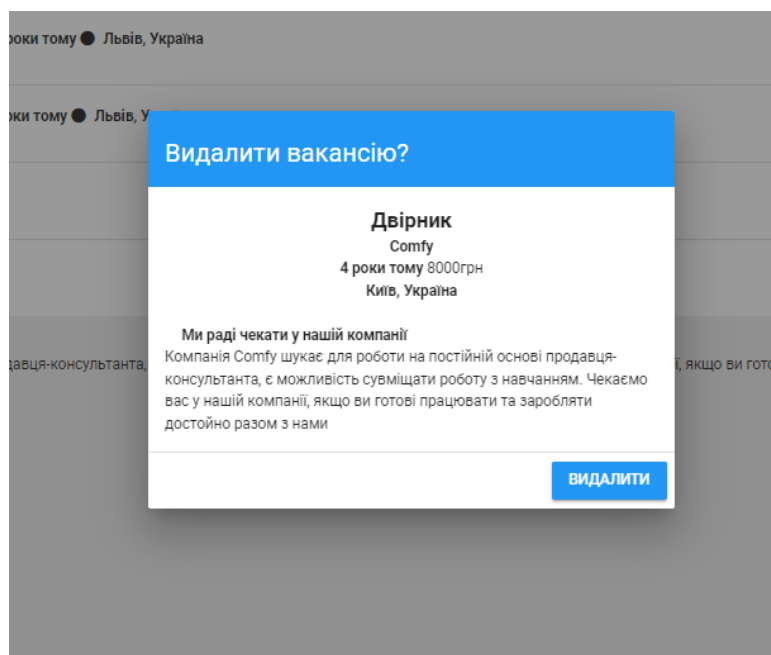


Рисунок 3.14 – Видалення вакансії

Також можна переглядати опубліковані резюме (рис. 3.15).

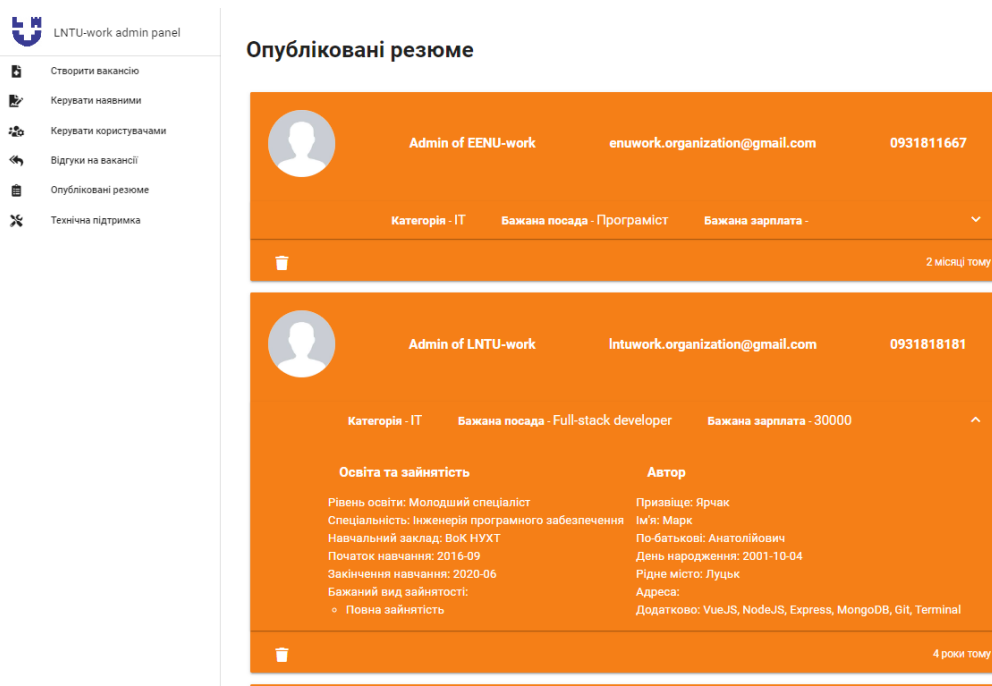


Рисунок 3.15 – Опубліковані резюме

Перегляд відгуків на вакансії можна побачити на вкладці «Відгуки на вакансії», що зображено на рисунку 3.16.

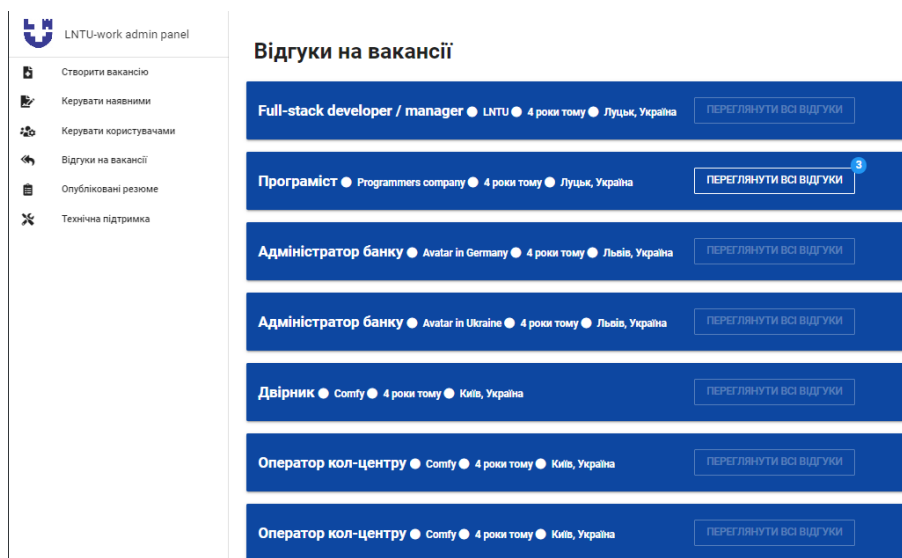


Рисунок 3.16 – Перегляд відгуків

Якщо є якісь відгуки, то можна натиснути на кнопку та переглянути усі відгуки (рис. 3.17).

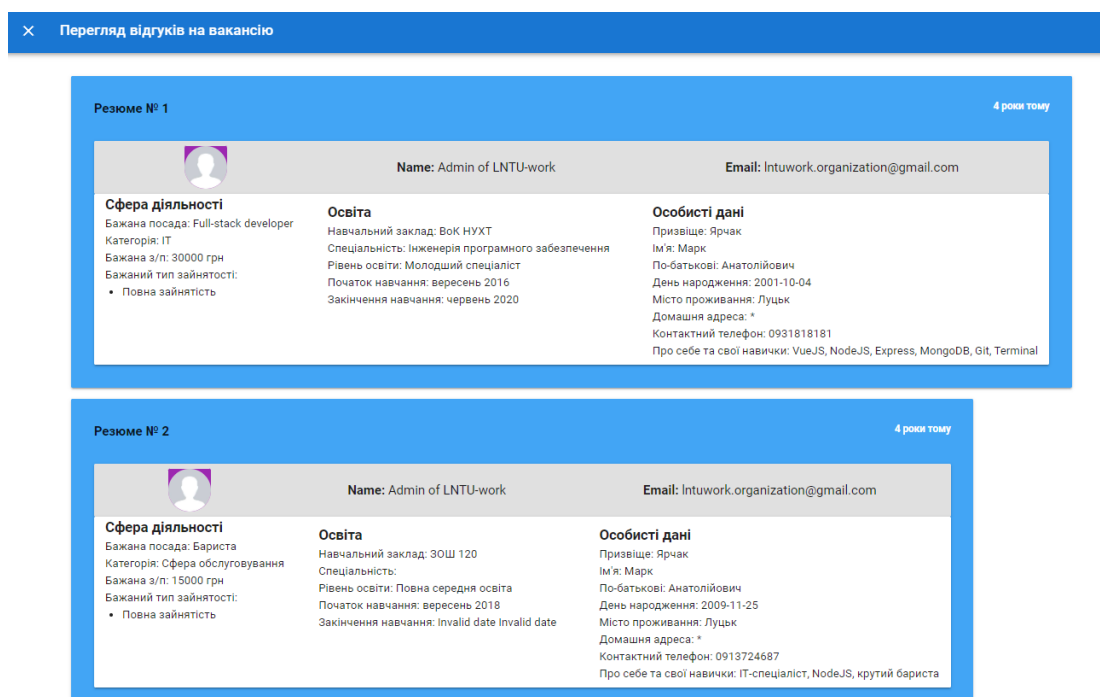


Рисунок 3.17 – Перегляд відгуків на конкретну вакансію

Також наявна технічна підтримка, якщо є якісь проблеми (рис. 3.18), питання від користувачів то їх можна переглянути на вкладці «Технічна підтримка».

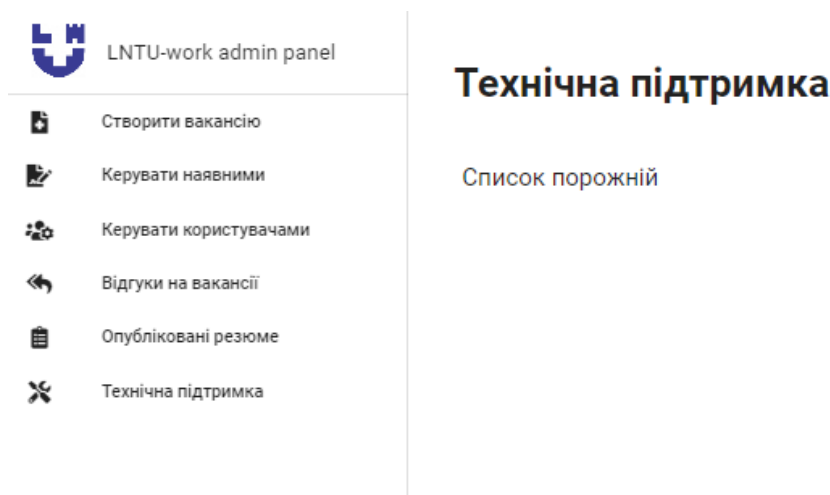


Рисунок 3.18 – Технічна підтримка

3.2 Тестування та налагодження інформаційно-комп'ютерної системи.

Було проведено тестування та налаштування інформаційно-комп'ютерної системи для вебсайту пошуку роботи за допомогою Vue.js, отримав цінний досвід у процесі тестування та налагодження. Спершу варто розглянути важливість тестування, чому все ж варто робити такі дії під час розробки будь-якого ПЗ.

Тестування відіграє вирішальну роль у розробці програмного забезпечення, гарантуючи якість, функціональність і надійність системи. Для вебсайту пошуку роботи ретельне тестування мало важливе значення для забезпечення бездоганної взаємодії з користувачем, перевірки реалізованих функцій, а також виявлення та усунення будь-яких проблем або дефектів. Це допомогло підвищити загальну продуктивність і зручність використання вебсайту.

Використані методи тестування під час розробки сайту.

Модульне тестування використовувалося для тестування окремих компонентів і функцій вебсайту. Це включало тестування компонентів Vue.js, забезпечення їх належного відтворення, функціональності та потоку даних. Модульні тести були написані з використанням тестових фреймворків, таких як

Jest, що дозволяє перевіряти очікувані результати порівняно з наданими вхідними даними.

Інтеграційне тестування було проведено для перевірки взаємодії та інтеграції різних компонентів і модулів на вебсайті. Він був зосереджений на тестуванні плавного зв'язку та координації між різними частинами системи, такими як автентифікація, підключення до бази даних і компоненти інтерфейсу користувача.

Тестування E2E було проведено для моделювання реальних сценаріїв користувача та перевірки функціональності всієї системи. Це передбачало автоматизацію дій користувача та тестування всього процесу пошуку роботи. Такі інструменти, як Cypress, використовувалися для проведення тестів E2E, гарантуючи, що вебсайт відповідає очікуванням користувачів і безперебійно працює в різних браузерах і пристроях.

Варто також, описати проблеми, що виникли під час тестування. Під час етапу тестування виникло кілька проблем, які вимагали ретельного аналізу та вирішення проблем, а саме:

- сумісність із різними браузерами. Забезпечення сумісності вебсайту в кількох браузерах викликало проблеми через відмінності в механізмах відтворення та підтримуваних функціях. Щоб досягти узгодженої поведінки та зовнішнього вигляду в таких популярних браузерах, як Chrome, Firefox і Safari, були необхідні масштабні тестування та налаштування;

- тестування адаптивного дизайну. Зі збільшенням використання мобільних пристроїв вебсайт пошуку роботи мав бути адаптивним і доступним на екранах різних розмірів. Тестування швидкості реагування та адаптація інтерфейсу користувача до різних пристроїв вимагали прискіпливої уваги до деталей і налаштувань для забезпечення оптимальної взаємодії з користувачем.

Застосовані методи налагодження. Налагодження було ключовим аспектом процесу розробки, дозволяючи ідентифікувати та вирішувати проблеми в системі. Для налагодження інформаційно-комп'ютерної системи використовувалися різні техніки:

- журнал консолі. Використання операторів `console.log` зіграло важливу роль у відстеженні потоку даних, визначенні значень змінних і виявленні будь-якої несподіваної поведінки під час виконання. Ця техніка допомогла звузити коло причин помилок і зрозуміти процес виконання коду;

- огляд та перевірка коду. Критичний аналіз кодової бази та проведення перевірок дозволили виявити логічні помилки, потенційні помилки та області, які потребують оптимізації.

Використання інструментів розробника браузера, таких як Vue.js devtools і консоль браузера, дало цінну інформацію про внутрішній стан програми, ієрархію компонентів і журнали помилок. Ці інструменти полегшили процес налагодження, визначивши конкретні проблемні області та допомагаючи у вирішенні проблем.

Тестування та налагодження інформаційно-комп'ютерної системи вебсайту пошуку роботи з використанням Vue.js були важливими кроками для забезпечення її функціональності, надійності та зручності використання. За допомогою методів ретельного тестування, таких як модульне тестування, інтеграційне тестування та наскрізне тестування було виправлено помилки. Роботу було оцінено, виявлено недоліки та внесено покращення. Крім того, для діагностики та вирішення проблем у системі було застосовано ефективні методи налагодження, такі як журнал консолі, перегляд коду та використання інструментів налагодження. Загалом цей досвід дав цінну інформацію про важливість тестування та налагодження в розробці програмного забезпечення та надав мені необхідні навички для створення високоякісних і надійних веб-додатків у майбутньому.

3.3 Розробка документації для програмного продукту

Мінімальні системні вимоги:

- операційна система: Windows 7 або новіша версія, macOS 10.10 або новіша версія, Linux (Ubuntu, Fedora або подібні дистрибутиви);

- веб-браузер: Google Chrome (остання версія), Mozilla Firefox (остання версія), Safari (остання версія), Microsoft Edge (остання версія);

– підключення до Інтернету: високошвидкісне підключення до Інтернету для оптимальної роботи.

Довідкові матеріали:

Посібник користувача, який охоплює всі функції та функції вебсайту пошуку роботи. Він містить покрокові інструкції щодо створення облікового запису, оновлення профілів користувачів, пошуку роботи, подання заявок на вакансії та керування сповіщеннями про роботу.

Поширені запитання: добірка поширених запитань і відповідей на них, що стосуються типових запитів, пов'язаних із використанням вебсайту, керуванням обліковим записом, процесом подання заявки на роботу та вирішенням поширених проблем.

Посібник користувача: вебсайт пошуку роботи для студентів і випускників.

Створення облікового запису:

- відвідайте домашню сторінку вебсайту;
- натисніть кнопку «Зареєструватися»;
- заповніть реєстраційну форму, вказавши свою особисту інформацію, включаючи ім'я, адресу електронної пошти та пароль;
- погодьтеся з умовами;
- натисніть кнопку «Зареєструватися»;
- підтвердьте свою електронну адресу, перейшовши за посиланням для підтвердження, надісланим на вашу зареєстровану електронну адресу.

Вхід в систему:

- перейдіть на домашню сторінку вебсайту;
- натисніть на кнопку «Увійти»;
- введіть зареєстровану адресу електронної пошти та пароль;
- натисніть на кнопку «Увійти».

Вивчення можливостей роботи:

- використовуйте панель пошуку на домашній сторінці, щоб ввести ключові слова, пов'язані з бажаною посадою, компанією чи місцем розташування;

- застосуйте такі фільтри, як категорія, місцезнаходження, діапазон зарплати та тип роботи, щоб звузити пошук;

- перегляньте відображений список вакансій;

- натисніть на список вакансій, щоб переглянути детальну інформацію, включно з компанією, вимогами до роботи та інструкціями щодо застосування.

Збереження та подання заявок на роботу:

- щоб зберегти вакансію для використання в майбутньому, натисніть кнопку «Зберегти» або «Додати в закладки», пов'язану зі списком вакансій;

- щоб подати заявку на роботу, дотримуйтесь інструкцій щодо подання заявки, наданих роботодавцем, які можуть включати подання резюме, супровідного листа або форми заяви;

- деякі списки вакансій можуть спрямовувати вас на зовнішній вебсайт або електронну адресу для подання заявки.

Керування вашим профілем:

- увійдіть у свій профіль, клацнувши ім'я користувача або зображення профілю;

- відредагуйте інформацію свого профілю, включаючи освіту, навички, досвід роботи та контактні дані;

- завантажте резюме у власний профіль, щоб легко подати заявку;

- налаштуйте параметри роботи та сповіщення, щоб отримувати сповіщення про відповідні вакансії;

Комунікація та мережа:

- використовуйте систему обміну повідомленнями, щоб спілкуватися з роботодавцями щодо можливостей роботи чи запитів;

- підключайтеся та спілкуйтеся з іншими користувачами, досліджуючи їхні профілі та надсилаючи запити на підключення.

Усунення несправностей і підтримка:

- якщо користувач зіткнувся з будь-якими технічними проблемами або має запитання, то звертається до розділу поширених запитань, щоб дізнатися про типові запитання та їх вирішення;

- щоб отримати додаткову допомогу, зверніться до служби підтримки клієнтів, використовуючи надану контактну інформацію.

Вихід:

- натисніть своє ім'я користувача або зображення профілю;
- виберіть опцію «Вийти», щоб безпечно вийти зі свого облікового запису.

Висновки до розділу 3

В цьому розділі було описано код про реєстрацію, вакансії та їх фільтрацію. Також було продемонстровано посібник користувача, що потрібно робити на сайті, покрокові дії.

ВИСНОВКИ

За результатами проведеного дослідження варто зробити наступні висновки: перш за все, розробка сайту була успішно виконана, відповідно до поставлених завдань. Сайт забезпечує легкий та зручний пошук вакансій для користувачів, завдяки наявності різноманітних фільтрів та інструментів, що забезпечують швидкий та точний пошук. Крім того, користувачам надається можливість додавати та редагувати свої резюме.

Поставлені завдання було виконано.

Розроблений сайт який є масштабованим та легким для розширення функціоналу в майбутньому, що дозволить забезпечити зростання та вдосконалення платформи залежно від потреб користувачів.

На сайті реалізована можливість створення резюме користувачем, що дозволяє зберігати у зручному форматі особисті та контактні дані кандидата. Також це спрощує для роботодавця відбір кандидата на відкриту вакансію з подальшим офером.

Присутні фільтри вакансій за критеріями, які спрощують процес пошуку вакансій кандидатами. Фільтри дозволяють максимально ефективно здійснювати пошук вакансій кандидатами.

Адаптивний дизайн сайту гарантує зручний перегляд і використання на різних пристроях, забезпечуючи максимальну сумісність та зручність для користувачів. Дизайн коректно функціонує на мобільних пристроях, планшетах, а також на ПК з розширеннями 4K. Сайт стабільно працює на попередніх версіях браузера. Медіа адаптовані під Retina дисплеї.

Важливим аспектом є забезпечення безпеки даних користувачів. Розроблена система включає захист від хакерських атак та незаконного доступу до даних, забезпечуючи конфіденційність та цілісність інформації. Використана технологія Firebase забезпечує максимальний захист особистих даних користувачів.

Додатково, реалізована функція підписки на розсилку нових вакансій за певними критеріями сприяє оперативному отриманню актуальної інформації та полегшує процес пошуку роботи.

Створені зручні умови для комунікації між роботодавцем та кандидатом, що покращує потенційну співпрацю.

Загалом, розроблений сайт для пошуку роботи для студентів та випускників відповідає всім поставленим завданням, забезпечуючи зручний, швидкий та безпечний пошук вакансій, сприяючи успішному розвитку кар'єри користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Н. М. Ліщина, А. А. Ящук, О. М. Суринович. Методичні вказівки до виконання кваліфікаційної роботи бакалавра: метод. рек.. Луцьк: ЛНТУ, 2022. 56 с.
2. Про компанію Jooble URL: <https://ua.jooble.org/info/about> (дата звернення: 01.05.2023).
3. Market Share, Revenue and Traffic Analytics URL: <https://www.similarweb.com/website/jooble.org/#traffic> (дата звернення: 01.05.2023).
4. Customer service reviews about Jooble URL: <https://uk.trustpilot.com/review/jooble.org?page=3> (дата звернення: 01.05.2023).
5. Your new job, vacancies all over URL: <https://ua.jooble.org> (дата звернення: 03.05.2023).
6. Jooble is a Ukranian product IT company URL: <https://happymonday.ua/en/company/jooble> (дата звернення: 07.05.2023).
7. Work.ua – сайт пошуку роботи №1 в Україні | Про нас URL: <https://www.work.ua/about-us/> (дата звернення: 07.05.2023).
8. Work.ua | overview, statistics, about company URL: <https://www.similarweb.com/website/work.ua/#overview> (дата звернення: 08.05.2023).
9. Які переваги й недоліки пошуку роботи через інтернет-ресурси? URL: <https://www.kadrovik.ua/novyny/yaki-perevagy-j-nedoliky-poshuku-roboty-cherez-internet-resursy> (дата звернення: 10.05.2023).
10. Головна сторінка work.ua URL: <https://www.work.ua> (дата звернення: 12.05.2023).
11. Про Dou URL: <https://www.linkedin.com/company/developers.org.ua> (дата звернення: 13.05.2023).
12. Dou | overview URL: <https://www.similarweb.com/website/dou.ua/#ranking> (дата звернення: 13.05.2023).
13. Головна сторінка DOU URL: <https://dou.ua> (дата звернення: 14.05.2023).

14. Що таке Firebase? Повна історія. URL: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> (дата звернення: 16.05.2023).

15. Правила безпеки Firebase для URL-адреси хмарного сховища URL: <https://firebase.google.com/docs/storage/security?hl=en> (дата звернення: 16.05.2023).

16. Vue.js документація URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 16.05.2023).

17. Top 10 Practical Benefits of Vue.js for Web Development URL: <https://www.techmagic.co/blog/benefits-of-vuejs/> (дата звернення: 18.05.2023).

18. Why should to choose vuetify? URL: <https://vuetifyjs.com/en/introduction/why-vuetify/> (дата звернення: 18.05.2023).

19. Choosing the Right Front-End Framework for Your Vue App URL: <https://medium.com/the-web-tub/choosing-the-right-front-end-framework-for-your-vue-app-4448bac12ce7> (дата звернення: 19.05.2023).

20. Підручник з архітектури та дизайну програмного забезпечення URL: https://www.tutorialspoint.com/software_architecture_design/ (дата звернення: 20.05.2023).

21. How to build and deploy a Vue.js app with Cloud Firestore and Vuefire by Timothy Olaleke. URL: <https://blog.logrocket.com/how-to-build-deploy-vue-js-app-with-cloud-firestore-vuefire/> (дата звернення: 21.05.2023).

22. Work ua | overview, statistics, about company URL: <https://www.similarweb.com/website/work.ua/#geography> (дата звернення: 19.05.2023).

23. Web Development 101: A Comprehensive Guide to Building Modern Websites. Blog post. Medium. URL: <https://medium.com/codechef-vit/web-development-101-with-codechefvit-4ec369e86f68> (дата звернення: 20.05.2023).

24. Vue JS Firebase Firestore: Build A CRUD App with Authentication by Raja Tamil. URL: <https://softauthor.com/vue-firestore-crud-app-authentication/> (дата звернення: 21.05.2023).

ДОДАТКИ

ДОДАТОК А

Компонент «VacancyPage»

```

<template xmlns:v-slot="http://www.w3.org/1999/XSL/Transform">
  <div id="vacancy-page">
    <div class="vacancy-page">
      <div class="merge-box__top-plus-description">
        <div
          class="vacancy-page__top-string"
          :style="
            $vuetify.breakpoint.xsOnly || $isMobile()
              ? ''
              : 'margin: 0 7px; height: 50px'
          ">
        </div>
        <div class="top-sting__string-left-box">
          <div class="string-left-box__go-back">
            <v-btn
              flat
              color="amber darken-3"
              @click="
                manageRoute.before.fullPath !== '/'
                  ? $router.go(-1)
                  : $router.push({ name: 'AvailableVacancies' })
            ">
            </v-btn>
            <i class="fas fa-chevron-left fa-lg mr-2"></i>
            <span v-if="manageRoute.before.fullPath === '/'">
              {{
                $vuetify.breakpoint.xsOnly || $isMobile()
                  ? "До вакансій"
                  : "перейти до вибору вакансій"
              }}</span>
            </div>
            <span
              v-else-if="
                manageRoute.before.path ===
                  '/workspace/profile/marked-vacancies' ||
                manageRoute.before.path === '/workspace/marked-
vacancies'
            ">
            </span>
            {{
              $vuetify.breakpoint.xsOnly || $isMobile()
                ? "До вакансій"
                : "повернутися до обраних вакансій"
            }}</span>
            </div>
            <span v-else>{{
              $vuetify.breakpoint.xsOnly || $isMobile()

```

```

        ? "До вакансій"
        : "повернутися до вибору вакансій"
    }}</span>
</v-btn>
</div>
</div>
<div class="top-sting__string-right-box">
    <v-btn v-ripple="false" flat color="deep-blue"
@click="markVacancy">
        <span
            v-if="!markedVacancies.some((v) => v ===
currentOpenVacancy.id)"
        >
            {{
                $vuetify.breakpoint.xsOnly || $isMobile()
                ? "Позначити"
                : "Додати до обраних вакансій"
            }}
            <i class="far fa-star fa-lg ml-2"></i>
        </span>
        <span
            v-if="markedVacancies.some((v) => v ===
currentOpenVacancy.id)"
        >
            {{
                $vuetify.breakpoint.xsOnly || $isMobile()
                ? "Зняти позначку"
                : "Видалити з обраних вакансій"
            }}
            <i class="fas fa-star fa-lg ml-2"></i>
        </span>
    </v-btn>
</div>
</div>
<v-card
    elevation="15"
    :style="
        $vuetify.breakpoint.xsOnly || $isMobile()
        ? 'padding: 36px 15px 20px 15px; margin: 10px'
        : 'padding: 40px 30px; margin: 0 15px'
    "
    class="vacancy-page__full-vacancy-description"
>
    <div v-if="currentOpenVacancy.category" class="vacancy-
category-box">
        <div class="vacancy-category">
            <span>{{ currentOpenVacancy.category }}</span>
        </div>
    </div>
    <div v-if="activeLoading" class="loading-box">
        <v-progress-circular
            :size="70"

```

```

        :width="7"
        color="blue"
        indeterminate
      />
    </div>
    <div v-if="!activeLoading" class="just-vacancy-
description">
      <div class="full-vacancy-description__position-box">
        <span
          :style="
            $vuetify.breakpoint.xsOnly || $isMobile()
              ? 'font-size: 30px;'
              : 'font-size: 40px; margin-left: 25px'
          "
          class="position-box__position"
        >{{ currentOpenVacancy.position }}</span>
      >
    </div>
    <div class="full-vacancy-description__salary-date-box">
      <div
        v-if="currentOpenVacancy.salary"
        class="salary-date-box__salary"
      >
        <i class="fas fa-coins mr-3 fa-lg"></i>
        {{ currentOpenVacancy.salary }}
        грн<!--<i class="fas fa-hryvnia ml-1"></i>-->
      </div>
      <div
        v-if="!$isMobile() || !$vuetify.breakpoint.xsOnly"
        class="salary-date-box__give-resume"
      >
        <v-btn
          outline
          color="blue darken-4"
          @click="$vuetify.goTo('#submit-resume')"
        >
          <span
            ><i class="fas fa-arrow-down mr-1"></i>
Відгукнутися</span>
          </v-btn>
        </div>
      </div>
      <div class="full-vacancy-description__about-company-
box">
        <div class="about-company-box__title">
          <span class="box-title">Про компанію</span>
        </div>
        <div class="about-company-box__details">
          <div>
            <span class="company-name">{{
              currentOpenVacancy.company.name
            }}

```

```

    }}</span>
</div>
<div class="company-size">
    {{ currentOpenVacancy.company.size }}
    {{
        currentOpenVacancy.company.size === "1"
        ? "робітник"
        : "робітників"
    }}
</div>
<v-layout
    class="company-contacts"
    :style="
        $vuetify.breakpoint.xsOnly || $isMobile()
        ? 'justify-content: flex-start; padding-left:
7px; padding-right: 10px'
        : 'justify-content: space-around'
    "
    wrap
>
    <div
        :class="
            $vuetify.breakpoint.xsOnly || $isMobile() ?
'mr-5' : null
        "
        class="company-contacts__site"
    >
        <a
            :href="`https://${currentOpenVacancy.company.contacts.site}`"
            target="_blank"
        >
            {{ currentOpenVacancy.company.contacts.site
        }}</a>
    </div>
    <div
        :class="
            $vuetify.breakpoint.xsOnly || $isMobile() ?
'mr-5' : null
        "
        class="company-contacts__email"
    >
        {{ currentOpenVacancy.company.contacts.email }}
    </div>
    <div
        :class="
            $vuetify.breakpoint.xsOnly || $isMobile() ?
'mr-5' : null
        "
        class="company-contacts__phone"
    >

```



```

        ></i>
        {{ currentOpenVacancy.employment[1] }}
    </span>
</div>
</div>
<div class="location-and-employment-box__one-box">
    <div class="one-lit-header">Опубліковано</div>
    <span>
        <i class="fas fa-business-time"></i>
        {{ returnPlacementDate }}</span>
    >
</div>
</v-layout>
<div class="full-vacancy-description__vacancy-
description-block">
    <div class="vacancy-description-block__block-title-
box">
        <span class="block-title-box__title">
            <i class="fab fa-elementor mr-2"></i> Опис
вакансії
        </span>
    </div>
    <div class="vacancy-description-block__full-
description">
        <div class="full-description__body-block">
            <div class="body-block__description-text">
                <div class="description-text__header">
                    <span>{{ currentOpenVacancy.description.header
}}</span>
                </div>
                <div class="description-text__value">
                    <span>{{ currentOpenVacancy.description.text
}}</span>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="full-vacancy-description__requirements-
block">
    <div class="requirements-block__list-title-box list-
title-box">
        <span class="list-title-box__title">Вимоги до
вакансії</span>
    </div>
    <div class="requirements-block__list-body-box">
        <ul>
            <li>
                v-for="(requirement, rk) in
currentOpenVacancy.description
                .requirements"
                :key="rk"
            </li>
        </ul>
    </div>
</div>

```

```

        class="list-title-box__text"
      >
        <span>{{ requirement }}</span>
      </li>
    </ul>
  </div>
</div>
<div class="full-vacancy-description__duties-block">
  <div class="duties-block__list-title-box list-title-
box">
    <span class="list-title-box__title">Обов'язки
працівника</span>
  </div>
  <div class="duties-block__list-body-box">
    <ul>
      <li
        v-for="(duties, dk) in
currentOpenVacancy.description
        .duties"
        :key="dk"
        class="list-title-box__text"
      >
        <span>{{ duties }}</span>
      </li>
    </ul>
  </div>
</div>
<div class="full-vacancy-description__offers-block">
  <div class="offers-block__list-title-box list-title-
box">
    <span class="list-title-box__title">Ми
пропонуємо</span>
  </div>
  <div class="offers-block__list-body-box">
    <ul>
      <li
        v-for="(offer, ofk) in
currentOpenVacancy.description
        .companyOffers"
        :key="ofk"
        class="list-title-box__text"
      >
        <span>{{ offer }}</span>
      </li>
    </ul>
  </div>
</div>
<div class="respond-vacancy-block">
  <div class="respond-vacancy-block__menu-btn-submit">
    <v-btn
      id="submit-resume"
      depressed

```

```

        large
        dark
        round
        color="#FF5722"
        class="submit-button"
        @click="
            currentUser
            ? (bottomResumesSheet = true)
            : (notRegisteredDialog = true)
        "
    >
        Подати резюме
    </v-btn>
</div>
</div>
</div>
</v-card>
</div>
<v-navigation-drawer
    v-if="userAuthState"
    v-model="rightSubscribeBar"
    fixed
    right
    temporary
    width="340"
>
    <VacancyPageRightBar
        @close-subscribe-bar="
            (rightSubscribeBar = false), (subscribedOnSnackbar =
true)
        "
    />
</v-navigation-drawer>
</div>
<v-snackbar
    v-model="subscribedOnSnackbar"
    :timeout="4000"
    multi-line="multi-line"
    top="top"
    color="blue-grey darken-4"
>
    <span style="text-align: center">Зміни збережено</span>
    <v-btn color="white" outline @click="subscribedOnSnackbar =
false">
        Ок
    </v-btn>
</v-snackbar>
<v-snackbar
    v-model="snackbar"
    :timeout="timeout"
    multi-line="multi-line"
    top="top"

```

```

        color="blue-grey darken-4"
    >
    <span style="text-align: center">{{ snackbarText }}</span>
    <v-btn color="white" outline @click="snackbar = false"> Ок
</v-btn>
</v-snackbar>
<v-fab-transition>
    <v-btn
        v-if="userAuthState && userEmailVerified"
        color="blue darken-1"
        dark
        fab
        fixed
        bottom
        right
        @click="subscBtnClick"
    >
        <i class="fas fa-envelope-open-text fa-lg"></i>
    </v-btn>
</v-fab-transition>
<v-dialog v-model="notEmailVarifiedDialog" width="400" hide-
overlay>
    <v-card>
        <v-card-title
            class="headline deep-blue darken-1 white--text"
            primary-title
        >
            <div style="text-align: center">Підтвердіть електронну
адресу</div>
        </v-card-title>

        <v-card-text>
            Ви не можете подавати свої резюме, оскільки вашу
електронну адресу не
            підтверджено.
        </v-card-text>

        <v-divider></v-divider>

        <v-card-actions>
            <v-spacer />
            <v-btn
                color="deep-blue darken-1"
                flat
                @click="notEmailVarifiedDialog = false"
            >
                Гаразд
            </v-btn>
            <v-spacer />
        </v-card-actions>
    </v-card>
</v-dialog>

```

```

<v-dialog v-model="notRegisteredDialog" width="340">
  <v-card>
    <v-card-title
      class="headline deep-blue darken-3 white--text"
      primary-title
    >
      Увійдіть в аккаунт
    </v-card-title>

    <v-card-text>
      Ви не увійшли в аккаунт, тому ви не можете подавати резюме
      на доступні
      вакансії
    </v-card-text>

    <v-divider />

    <v-card-actions>
      <v-spacer />
      <v-btn color="blue" flat @click="$router.push({ name:
'LoginPage' })">
        Увійти
      </v-btn>
    </v-card-actions>
  </v-card>
</v-dialog>
<v-bottom-sheet v-model="bottomResumesSheet">
  <v-list>
    <v-subheader>Виберіть резюме</v-subheader>
    <template v-if="profileResumes[0].desirablePosition">
      <v-list-tile
        v-for="(resume, keyr) in profileResumes"
        :key="keyr"
        @click="prepareResumeToSubmit(resume) "
      >
        <v-list-tile-avatar>
          <v-avatar size="32px" tile>
            <i class="fas fa-file-alt fa-lg"></i>
          </v-avatar>
        </v-list-tile-avatar>
        <v-list-tile-title>
          {{ resume.desirablePosition }}
        </v-list-tile-title>
      </v-list-tile>
    </template>
    <template v-else>
      <v-list-tile @click="bottomResumesSheet = false">
        <v-list-tile-avatar>
          <v-avatar size="32px" tile>
            <i class="far fa-file fa-lg"></i>
          </v-avatar>
        </v-list-tile-avatar>
      </v-list-tile>
    </template>
  </v-list>
</v-bottom-sheet>

```

```

        <v-list-tile-title> Список порожній </v-list-tile-title>
        <v-btn
          outline
          color="blue darken-3"
          @click="$router.push({ name: 'PostResume' })"
        >
          Додати
        </v-btn>
      </v-list-tile>
    </template>
  </v-list>
</v-bottom-sheet>

<v-dialog
  v-model="readyForSubmitResumeDialog"
  transition="dialog-bottom-transition"
  fullscreen
  hide-overlay
>
  <v-card>
    <v-toolbar class="headline deep-blue darken-2" primary-title
dark app>
      <v-btn icon dark @click="readyForSubmitResumeDialog =
false">
        <v-icon>close</v-icon>
      </v-btn>
      <v-toolbar-title>Подати резюме?</v-toolbar-title>
      <v-spacer />
      <v-toolbar-items>
        <v-btn dark flat color="white" @click="submitResume">
          <v-icon style="font-size: 26px">send</v-icon>
        </v-btn>
      </v-toolbar-items>
    </v-toolbar>
    <v-content>
      <v-card-text>
        <v-card-text>
          <v-layout>
            <div
              class="pl-3 pr-3 pb-2"
              style="font-size: 16px; display: flex; flex-wrap:
wrap"
            >
              <div class="mr-5">
                <div style="font-size: 20px">
                  <span><b>Сфера діяльності</b></span>
                </div>
                <div>
                  <div>
                    Бажана посада:
                    {{ readyForSubmitResume.desirablePosition }}
                  </div>

```

```

        <div>Категорія: {{
readyForSubmitResume.category }}</div>
        <div>
            Бажана з/п:
            {{ readyForSubmitResume.desirableSalary }}
грн
        </div>
        <div>
            Бажаний тип зайнятості:
            <ul>
                <li
                    v-for="(
                        oneType, ti
                    ) in
readyForSubmitResume.employmentType"
                    :key="ti"
                >
                    {{ oneType }}
                </li>
            </ul>
        </div>
    </div>
    <div>
        <v-flex style="margin: 10px 60px 0 0">
            <span style="font-size:
20px"><b>Освіта</b></span>
            <div>
                Навчальний заклад:
                {{ readyForSubmitResume.educationInstitution
|| "*" }}
            </div>
            <div>
                Спеціальність: {{
readyForSubmitResume.specialty }}
            </div>
            <div>
                Рівень освіти: {{
readyForSubmitResume.educationLevel }}
            </div>
            <div>
                Початок навчання:
                {{
moment(readyForSubmitResume.educationStartDate)
                    .locale("uk")
                    .format("MMMM") +
                    " " +
moment(readyForSubmitResume.educationStartDate).format(
                    "YYYY"
                )
            }}
        </v-flex>
    </div>

```

```

</div>
<div>
    Закінчення навчання:
    {{

moment(readyForSubmitResume.educationFinishDate)
    .locale("uk")
    .format("MMMM") +
    " " +

moment(readyForSubmitResume.educationFinishDate).format(
    "YYYY"
    )
    }}
</div>
</v-flex>
<v-flex style="margin-top: 10px">
    <span style="font-size: 20px"><b>Особисті
дані</b></span>
    <div>
        <div>Призвіще: {{ readyForSubmitResume.surname
    }}</div>
        <div>Ім'я: {{ readyForSubmitResume.name
    }}</div>
        <div>
            По-батькові: {{
readyForSubmitResume.thirdName }}
        </div>
        <div>
            День народження: {{
readyForSubmitResume.birthdate }}
        </div>
        <div>
            Місто проживання:
            {{ readyForSubmitResume.locationCity || "*" }}
        </div>
        <div>
            Домашня адреса:
            {{ readyForSubmitResume.homeAddress || "*" }}
        </div>
        <div>
            Контактний телефон:
            {{ readyForSubmitResume.phone || "*" }}
        </div>
        <div>
            Про себе та свої навички:
            {{ readyForSubmitResume.moreAboutSelf }}
        </div>
        <div v-
if="readyForSubmitResume.attachedFile.imageName">
            {{ readyForSubmitResume.attachedFile }}

```

```

        </div>
    </v-flex>
</div>
</v-layout>
</v-card-text>
</v-card-text>
</v-content>
</v-card>
</v-dialog>
<v-snackbar
    v-model="submitResumeState"
    :timeout="4000"
    multi-line="multi-line"
    top="top"
    color="blue-grey darken-4"
>
    <span style="text-align: center">{{ submitResumeStateText
}}</span>
    <v-btn color="white" outline @click="submitResumeState =
false">
        Ок
    </v-btn>
</v-snackbar>
</div>
</template>

```