

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Web-додаток для пошуку дистанційної роботи»

Здобувачки групи ІТ-91 Павловської Олени Тимурівни.
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

(підпис)

Олена ПАВЛОВСЬКА
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доцент Юлія ПАРФЕНЕНКО

(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології
проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Павловська Олена Тимурівна

1 Тема роботи Web-додаток для пошуку дистанційної роботи .

керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент ,

затверджені наказом по університету від «29» травня 2023 р. №0588-VI

2 Строк подання студентом роботи «07» червня 2023 р.

3 Вхідні дані до роботи текстові та графічні матеріали для наповнення web-додатку .

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку .

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) дослідження продуктів-аналогів, вимоги до додатку, діаграма IDEF0, діаграма декомпозиції, діаграма використання web-додатку, діаграма структурного моделювання, архітектура web-додатку, логічна модель бази даних, таблиці бази даних, зображення сторінок web-додатку .

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 07.11.2022

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі проєкту	01.03.2023 - 07.03.2023	
2	Аналіз додатків-аналогів	08.03.2023 - 14.03.2023	
3	Складання технічного завдання	15.03.2023 - 31.03.2023	
4	Проектування web-додатку	01.04.2023 - 15.04.2023	
5	Розробка web-додатку	16.04.2023 - 05.05.2023	
6	Тестування web-додатку	16.05.2023 - 12.05.2023	
7	Завантаження на хостинг	13.05.2023 - 14.05.2023	
8	Оформлення документації	15.05.2023 - 05.06.2023	

Студент

(підпис)

Олена ПАВЛОВСЬКА

Керівник роботи

(підпис)

к.т.н., доц. Юлія ПАРФЕНЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток для пошуку дистанційної роботи».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 15 найменувань, чотирьох додатків. Загальний обсяг роботи – 90 сторінок, у тому числі 55 сторінок основного тексту, 2 сторінки списку використаних джерел, 33 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку для пошуку дистанційної роботи.

У першому розділі було здійснено аналіз предметної області, огляд останніх досліджень та публікацій, розглянуті та порівняні між собою аналогічні web-додатки. Також була сформульована постановка задачі проекту.

У другому розділі проведено структурно-функціональне моделювання, визначено варіанти використання web-додатку та спроектовано базу даних. У результаті було змодельовано такі діаграми як: контекстна діаграма IDEF0 та її декомпозиції, діаграма варіантів використання.

У третьому розділі була сформована архітектура системи. Також була розглянута робота з програмним продуктом та проведено тестування web-додатку.

Ключові слова: web, web-додаток, розробка, база даних, ASP.NET, C#, web-додаток, дистанційна робота.

Зміст

ВСТУП.....	6
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.2 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів-аналогів.....	12
1.3 Постановка задачі.....	17
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКУ	19
2.1 Функціональне моделювання web-додатку в IDEF0	19
2.2 Проектування інформаційної системи	21
2.3 Проектування бази даних.....	24
3 РОЗРОБКА WEB-ДОДАТКУ	28
3.1 Архітектура web-додатку	28
3.2 Розробка дизайну web-додатку	30
3.4 Демонстрація роботи web-додатку.....	32
3.5 Тестування web-додатку.....	47
ВИСНОВОК	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	55
ДОДАТОК А	57
ДОДАТОК Б.....	67
ДОДАТОК В	76

ВСТУП

На сучасному ринку праці, який постійно змінюється, шукачі роботи стикаються з проблемами під час пошуку можливостей працевлаштування. Однією з найбільших проблем є пошук роботи, яка відповідає їхнім навичкам, досвіду та кар'єрним прагненням. Крім того, в останні роки набула популярності саме віддалена робота. Однією з головних причин є технологічний прогрес, який полегшив віддалену роботу. Завдяки наявності високошвидкісного інтернету, хмарних обчислень та інструментів для відеоконференцій, віддалені працівники можуть спілкуватися та співпрацювати зі своїми колегами та роботодавцями в режимі реального часу з будь-якої точки світу.

Ще однією причиною поширення віддаленої роботи є зростаючий попит на баланс між роботою та особистим життям. Дистанційна робота дозволяє працівникам мати більшу гнучкість і контроль над своїм робочим графіком, що може допомогти їм краще керувати своїм особистим і професійним життям. Крім того, віддалена робота може заощадити час і гроші на поїздки на роботу та інші витрати, пов'язані з роботою в традиційному офісі.

Пандемія COVID-19, що триває, також прискорила впровадження віддаленої роботи. Оскільки багато організацій були змушені закрити свої фізичні робочі місця, щоб запобігти поширенню вірусу, віддалена робота стала єдиною життєздатною можливістю для багатьох підприємств продовжувати свою діяльність. Пандемія також призвела до зміни ставлення до віддаленої роботи: багато працівників і роботодавців тепер розглядають її як довгострокову або постійну форму роботи.

Війна, що триває в Україні, має значний вплив на економіку країни та ринок праці. Конфлікт торкнувся багатьох підприємств і галузей, деякі з них були змушені припинити свою діяльність або переїхати в безпечніші райони. У такій ситуації віддалена робота стає все більш актуальною як засіб

підтримання економічної активності та забезпечення можливостей працевлаштування для тих, хто постраждав від конфлікту.

Метою кваліфікаційної роботи бакалавра є розробка web-додатку для дистанційного пошуку роботи, який відповідає потребам шукачів на цьому мінливому ринку праці.

Для досягнення мети проекту необхідно виконати наступні задачі:

- визначити актуальність роботи, цільову аудиторію та дослідити предметну область, обрати технології для розробки;
- провести аналіз аналогів платформ пошуку дистанційної роботи і виділити їх переваги та недоліки;
- спроектувати модель та структуру додатку;
- створити прототип додатку;
- розробити функціонал додатку та виконати його тестування;

Отже, розробка web-додатку для дистанційного пошуку роботи є важливим кроком на шляху до задоволення потреб шукачів роботи на мінливому ринку праці. Цей web-додаток забезпечить платформу для зв'язку між шукачами роботи та роботодавцями, що в кінцевому підсумку призведе до більш ефективного та успішного пошуку роботи. Завдяки використанню найкращих практик у розробці програмного забезпечення, цей web-додаток стане надійним і зручним інструментом для шукачів роботи по всьому світу.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.2 Огляд останніх досліджень і публікацій

За останні кілька років з'являється все більше досліджень і публікацій на тему дистанційної роботи, що відображає зростаючий інтерес до цієї форми організації праці. Деякі з ключових висновків останніх досліджень і публікацій приведені в таблиці 1.1.

Таблиця 1.1 – Дослідження щодо дистанційної роботи

Дослідження	Ким було проведене	Результати
Дистанційна робота може покращити баланс між роботою та особистим життям	International Journal of Human Resource Management [1]	Результатом дослідження є те, що дистанційна робота пов'язана з вищим рівнем балансу між роботою та особистим життям і задоволеністю роботою серед працівників. Дослідження також показало, що дистанційна робота сприяє утриманню працівників і зменшує кількість прогулів.
Дистанційна робота може призвести до підвищення продуктивності	Journal of Business and Psychology [2]	Дослідження показало, що дистанційна робота пов'язана з вищим рівнем задоволеності роботою та залученості до неї, що, в свою чергу, призводить до підвищення продуктивності. Дослідження також показало, що дистанційна робота особливо ефективна для працівників, які

		мають високу мотивацію та самоспрямованість.
Дистанційна робота може зменшити стрес від поїздок на роботу та витрати на неї	Journal of Business and Psychology [2]	Дослідження показало, що дистанційна робота пов'язана зі зменшенням стресу та витрат на поїздки на роботу, що може покращити загальний стан здоров'я та знизити рівень плинності кадрів.
Дистанційна робота може створювати проблеми з комунікацією	Journal of Business and Psychology [2]	Дослідження показало, що дистанційна робота може створювати проблеми з комунікацією та призводити до відчуття ізоляції серед віддалених працівників. У дослідженні пропонується, щоб менеджери надавали пріоритет комунікації та співпраці між віддаленими командами для подолання цих проблем.

Отже, нещодавні дослідження та публікації про дистанційну роботу висвітлили потенційні переваги та виклики цієї форми роботи. Хоча дистанційна робота може покращити баланс між роботою та особистим життям, підвищити продуктивність, зменшити стрес і витрати на поїздки на роботу, вона також може створювати проблеми з комунікацією і залежати від організаційної культури. Щоб отримати максимальну віддачу від дистанційної роботи, компаніям слід надавати пріоритет комунікації та співпраці, створювати сприятливу культуру, а також надавати ресурси та підтримку віддаленим працівникам.

Проаналізувавши подібні роботи, можна віділити технології створення web-додатків для пошуку роботи. Наприклад, в роботі «Online Job Search Application» [3] було використано такі технології:

- Visual Studio 2017 як середовище розробки має вбудовані візуальні інструменти, які полегшили створення і редагування web-інтерфейсу;
- у даній роботі використовували MSSQL Server як базу даних, яка виконує функцію збереження інформації про шукачів роботи, компанію, вакансії, та забезпечує високу доступність і надійність даних web-додатку;
- використана web-платформа ASP.NET дозволила розробникам швидко створити інтерактивні та візуально привабливі користувацькі інтерфейси;
- PHP допоміг створити функціональний web-додаток з динамічним контентом і можливістю взаємодії з користувачем;
- C# допоміг валідувати форми реєстрації/авторизації, виконати асинхронні запити до сервера і реагувати на події, такі як клік мишею або натискання клавіші.

У таблиці 1.2 представлені інформаційні технології, які зазвичай використовуються для пошуку роботи.

Таблиця 1.2 – Список технологій для пошуку дистанційної роботи

Технології	Використання
Web-додатки	Ці платформи дозволяють користувачам уточнити свій пошук на основі таких критеріїв, як галузь, місцезнаходження, рівень досвіду та тип роботи. Такі відомі платформи, як Robota.ua, Work.ua, djinni, DOU підтримують великі бази даних вакансій, надаючи такі можливості, як завантаження резюме, сповіщення про вакансії та пряме подання заявок.
Нетворкінг-платформи	Такі платформи, як LinkedIn, дають можливість створювати профілі, які демонструють їхні навички, досвід і досягнення. Користувачі можуть спілкуватися з

	колегами, рекрутерами, розширюючи свою професійну мережу та підвищуючи видимість для потенційних роботодавців.
Соціальні мережі	Щоб розширювати свою професійну мережу, стежити за впливовими людьми в галузі та спілкуватися з потенційними роботодавцями можна використовувати соціальні мережі, такі як Twitter і Facebook. Платформи соціальних мереж також надають можливості для пошуку роботи через спеціальні групи, сторінки компаній та оголошення про вакансії, якими діляться рекрутери.
Онлайн-портфоліо	Дозволяють шукачам роботи, особливо у творчих галузях, демонструвати свої роботи за допомогою онлайн-портфоліо. Такі платформи, як Behance, GitHub, Dribbble і конструктори персональних сайтів, пропонують зручні інтерфейси і шаблони для створення візуально привабливих портфоліо. Шукачі роботи можуть додавати посилання на свої портфоліо в резюме або онлайн-профілі, щоб роботодавці могли легко переглянути зразки їхніх робіт.

Використання інформаційних технологій у пошуку дистанційної роботи революціонізувало цей процес, зробивши його більш ефективним, доступним і пристосованим до індивідуальних уподобань. Шукачі роботи можуть використовувати ці технологічні інструменти та платформи, щоб збільшити свої шанси знайти підходящу роботу.

1.2 Аналіз програмних продуктів-аналогів

Починаючи реалізацію створення сайту з пошуку дистанційної роботи, треба розуміти ризики та те, що вже існує десятки сайтів за такою тематикою. Тому необхідно аналізувати вигляд та функціональність їх web-додатків. В той же час треба орієнтуватись на сучасні тенденції у дизайні та розробці, щоб користувачам було приємно, зрозуміло та комфортно користуватись програмним забезпеченням. Для визначення вимог майбутнього програмного продукту було проведено дослідження існуючих аналогів web-додатків, а саме: «DOU», «Djinni» та «Robota.ua».

Головна сторінка сайту «DOU» (рис. 1.1) представлена як стрічка новин, що не є привабливим, адже велика кількість тексту заважає сприймати інформацію. На мою думку, краще б було представити перелік можливостей, які надає цей сайт, а новини винести на окрему сторінку.

Справа на сторінці представлені компанії високого рівня, які співпрацюють з DOU, що показує їхню надійність та безпеку сайту.

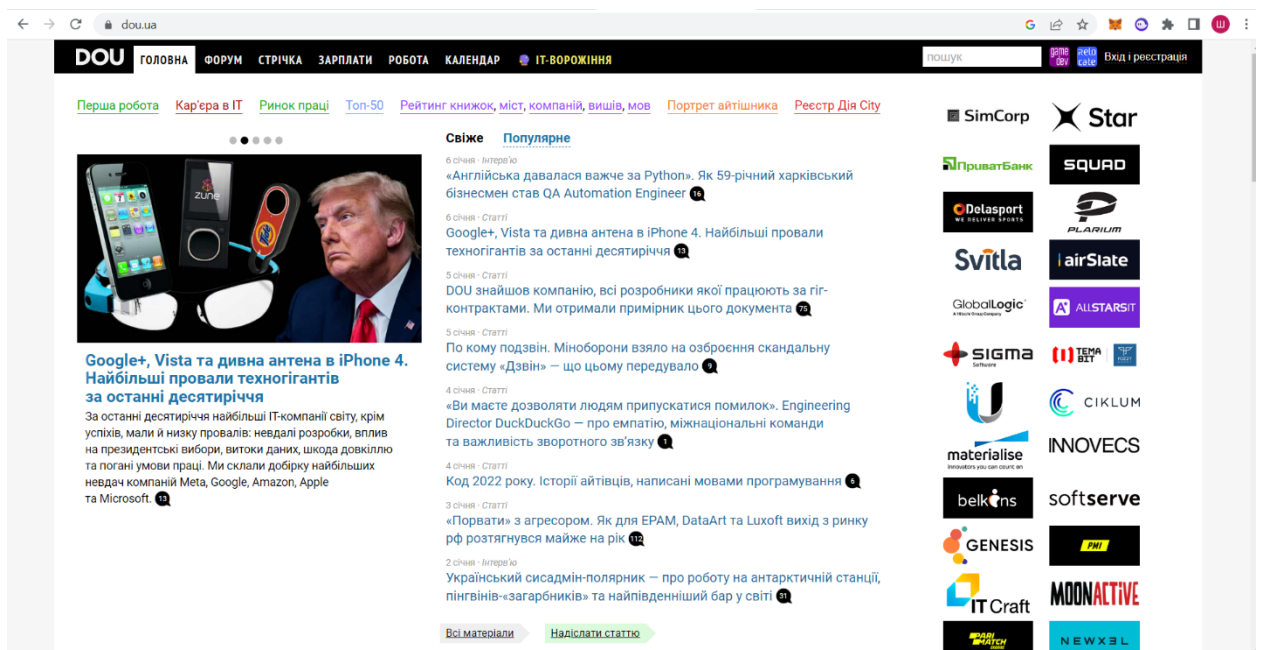


Рисунок 1.1 – Головна сторінка сайту «DOU»

На сайті представлений корисний розділ, такий як перегляд заробітної плати за вибраною спеціалізацією (рис. 1.2). Також тут можна обрати місто, досвід роботи, домен, тип компаній, освіту та рівень володіння англійською мовою, що допоможе відсортувати зарплати за обраними фільтрами.

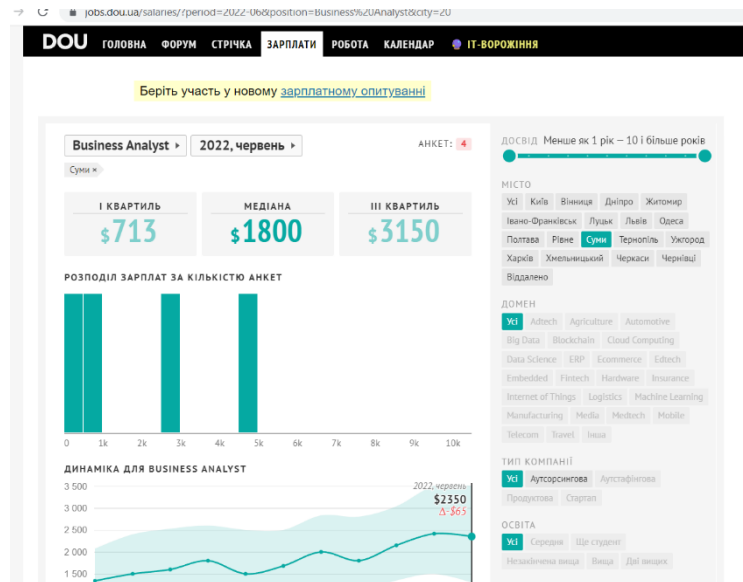


Рисунок 1.2 – Сторінка «Зарплата» сайту «DOU»

Сторінка з вакансіями (рис 1.3) також виглядає непривабливо, вона містить багато тексту, який не стилізований і зливається між собою.

The screenshot shows the 'Jobs' page on the DOU website. It includes a search bar with filters for categories, location, and company type. Below the search bar, there are sections for 'Java', 'Python', 'Ruby', 'iOS/macOS', and 'Android' jobs, each with a list of job titles and company logos. The right side of the page features a grid of logos for various companies.

Рисунок 1.3 – Сторінка «Вакансії» сайту «DOU»

Головна сторінка «Djinni» (рис 1.4) виконана у мінімалістичному стилі, користувачу відразу пропонується знайти роботу для кандидатів чи знайти розробників для компанії.

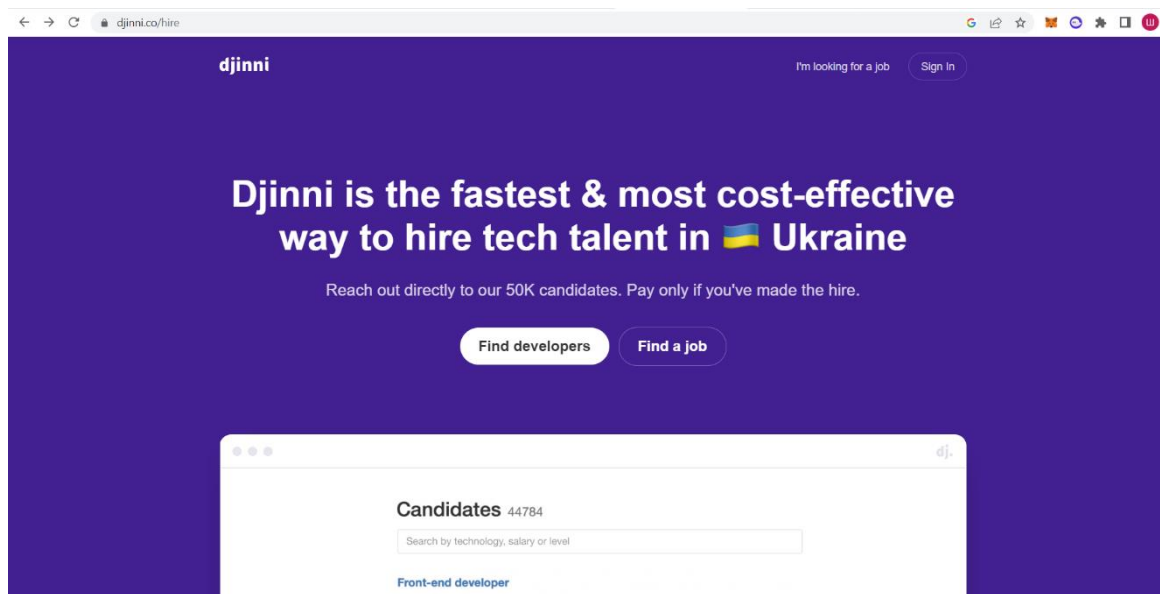


Рисунок 1.4 – Головна сторінка сайту «Djinni»

Сторінка вакансій (рис 1.5) як і в попередньому прикладі не дуже стилізована, через те, що вакансії не виділені блоками, вони зливаються між собою.

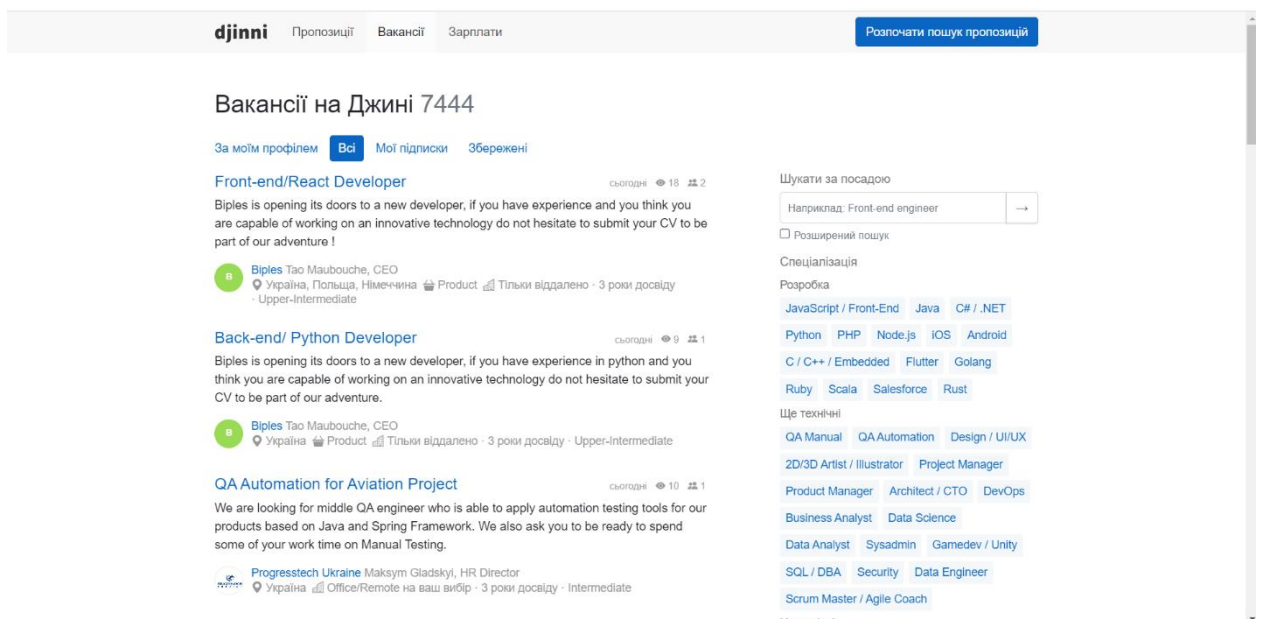


Рисунок 1.5 – Сторінка вакансій на «Djinni»

Також сайт має розділ зарплат (рис. 1.6), в якому можна обрати фільтри по досвіду роботи, рівню володіння англійської мови та міста.

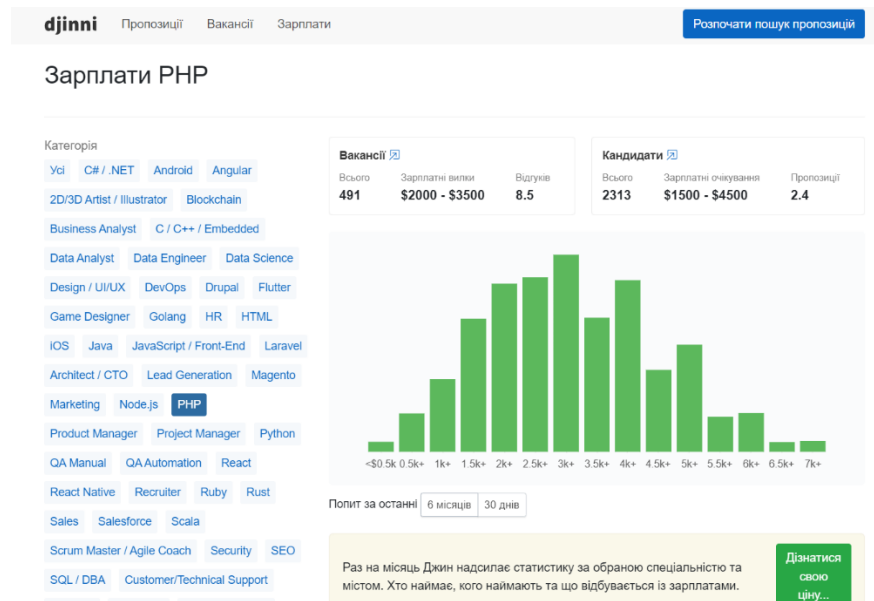


Рисунок 1.6 – Сторінка зарплат на Djinni

На головній сторінці сайту Robota.ua (рис. 1.7) відразу пропонується знайти вакансію, яка цікавить користувача та представлені компанії, які співпрацюють з Robota.ua. Сайт виконаний в мінімалістичному стилі, використовуючи червоний колір як основний.

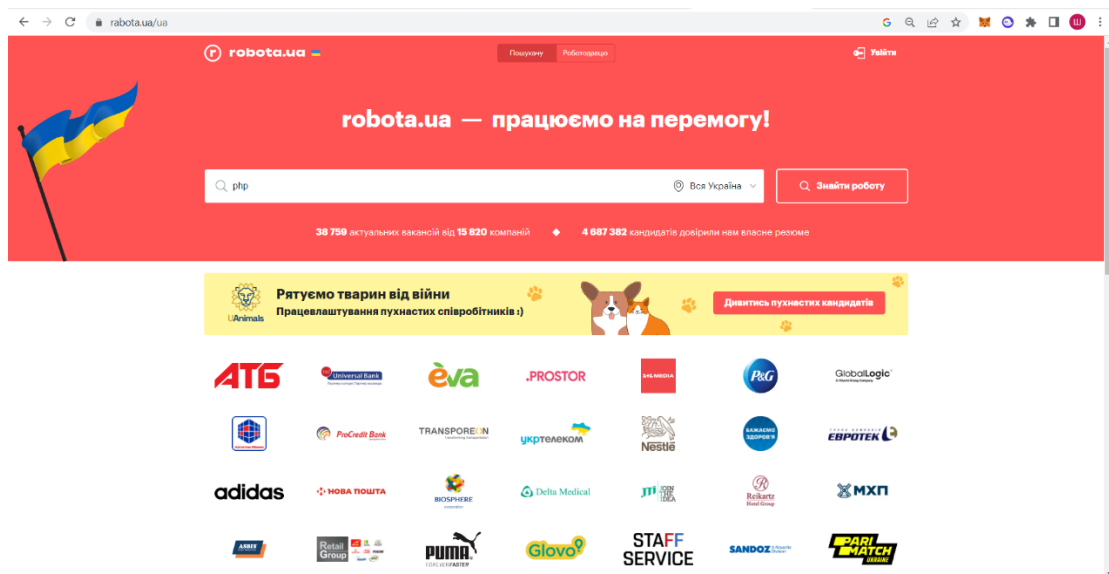


Рисунок 1.7 – Головна сторінка сайту «Robota.ua»

На рисунку 1.8 представлена сторінка вакансій. Пошук необхідної вакансії виконується легко, присутні фільтри та легкий функціонал. Переглянувши пропозиції можна відразу обрати ті, які зацікавили, або відхилити, щоб вони більше не попадалися.

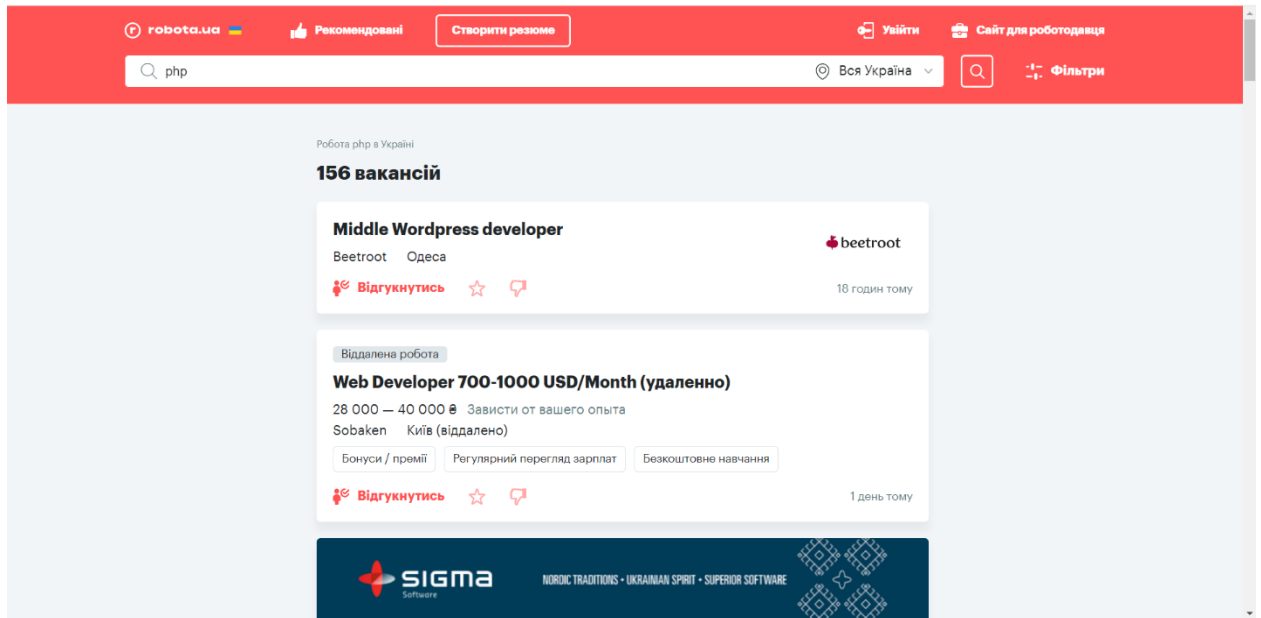


Рисунок – Сторінка вакансій на сайті «Robota.ua»

Після детального аналізу аналогів Інтернет-каталогів було визначено їх переваги і недоліки. Його результати представлені в таблиці 1.2.

Таблиця 1.2 – Порівняльна таблиця характеристик аналогів web-додатків

Характеристика/ Сайт	DOU	Djinni	Robota.ua
Сучасний дизайн	-	-	+
Зручний інтерфейс	-	-	+
Адаптивність	+	+	+
Функціональність	+	+	+
Навігація	+	+	+
Реєстрація	+	+	+

Дані з таблиці 1.4 надають змогу під час аналізу звернути увагу на функціональні доповнення, які можна втілити у власну розробку, і недоліки, яких необхідно уникати. Даний програмний продукт повинен мати сучасний дизайн, зручну навігацію та інтерфейс, адаптивність web-сторінок під різні види пристроїв та браузерів.

1.3 Постановка задачі

Метою даної роботи є розробка web-додатку для пошуку дистанційної роботи, який дозволить оптимізувати процес пошуку роботи по категоріям, залежно від запиту кандидата.

Web-додаток розроблений таким чином, аби полегшити процес пошуку роботи як для роботодавців, так і для шукачів. Шукачі роботи можуть створювати профілі, які демонструють їхні навички та досвід, шукати вакансії відповідно до своїх уподобань та подавати заявки на роботу безпосередньо через web-додаток. Роботодавці, в свою чергу, можуть розміщувати списки вакансій компанії, шукати кваліфікованих кандидатів та керувати заявками на роботу через web-додаток

Основні вимоги до створюваного програмного продукту є наступними:

- створити модуль реєстрації/авторизації, обираючи ролі роботодавця або кандидата.
- створити модуль для роботодавця, що дає можливість публікувати наявні вакансії компанії, писати вимоги щодо кандидатів та мати можливість зв'язатися з ними.
- створити модуль для кандидата з можливістю створювати власне портфоліо, переглянути всі доступні вакансії на сайті, сортувати за категоріями, надсилати відгук на зацікавлену вакансію та мати можливість зв'язатися з роботодавцем певної компанії.

Реалізація web-додатку відбувається з використанням:

- Figma – для створення прототипу web-додатку;
- HTML 5 – мова розмітки web-сторінок;
- CSS 3 – каскадні таблиці стилів для опису зовнішнього вигляду;
- C# – мова програмування;
- MSSQL – база даних для збереження інформації;
- BootStrap – набір шаблонів CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу;
- ASP.NET – платформа для створення web-додатку.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКУ

2.1 Функціональне моделювання web-додатку в IDEF0

IDEF0 – це техніка моделювання, яка використовується для представлення та аналізу функцій і діяльності системи або процесу. Вона надає графічну нотацію для опису потоку входів, виходів, елементів управління та механізмів, що беруть участь у досягненні конкретних цілей.

В IDEF0 система або процес декомпозується на ієрархічні рівні функцій, де кожна функція представляє певну дію або перетворення. Ці функції зображуються у вигляді блоків, а стрілки вказують на потік входів і виходів між функціями. Основна мета IDEF0 - забезпечити чітке розуміння того, як функції пов'язані між собою і сприяють досягненню загальних цілей системи або процесу.

Контекстна діаграма процесу пошуку дистанційної роботи за допомогою web-додатку в нотації IDEF0 представлена на рисунку 2.1.

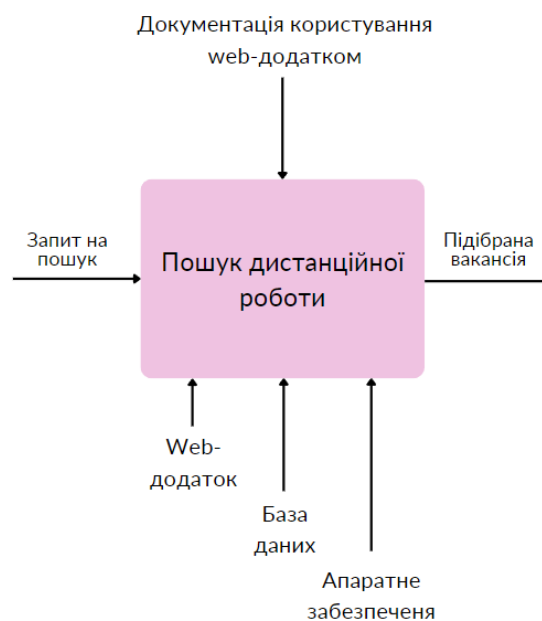


Рисунок 2.1 – Контекстна діаграма

Наступним кроком було побудовано IDEF0-деталізацію (рис. 2.2) – це розширення техніки моделювання IDEF0, що дозволяє більш детально аналізувати та документувати функції системи. Вона забезпечує більш глибокий рівень деталізації та специфічності у відображенні взаємодій, входів, виходів та елементів керування в системі або процесі. В цілому, IDEF0-деталізація підвищує потужність і точність техніки моделювання IDEF0, фіксуючи більш дрібні деталі функцій системи. Це дозволяє проводити більш ретельний аналіз системи і сприяє ефективній оптимізації та вдосконаленню.

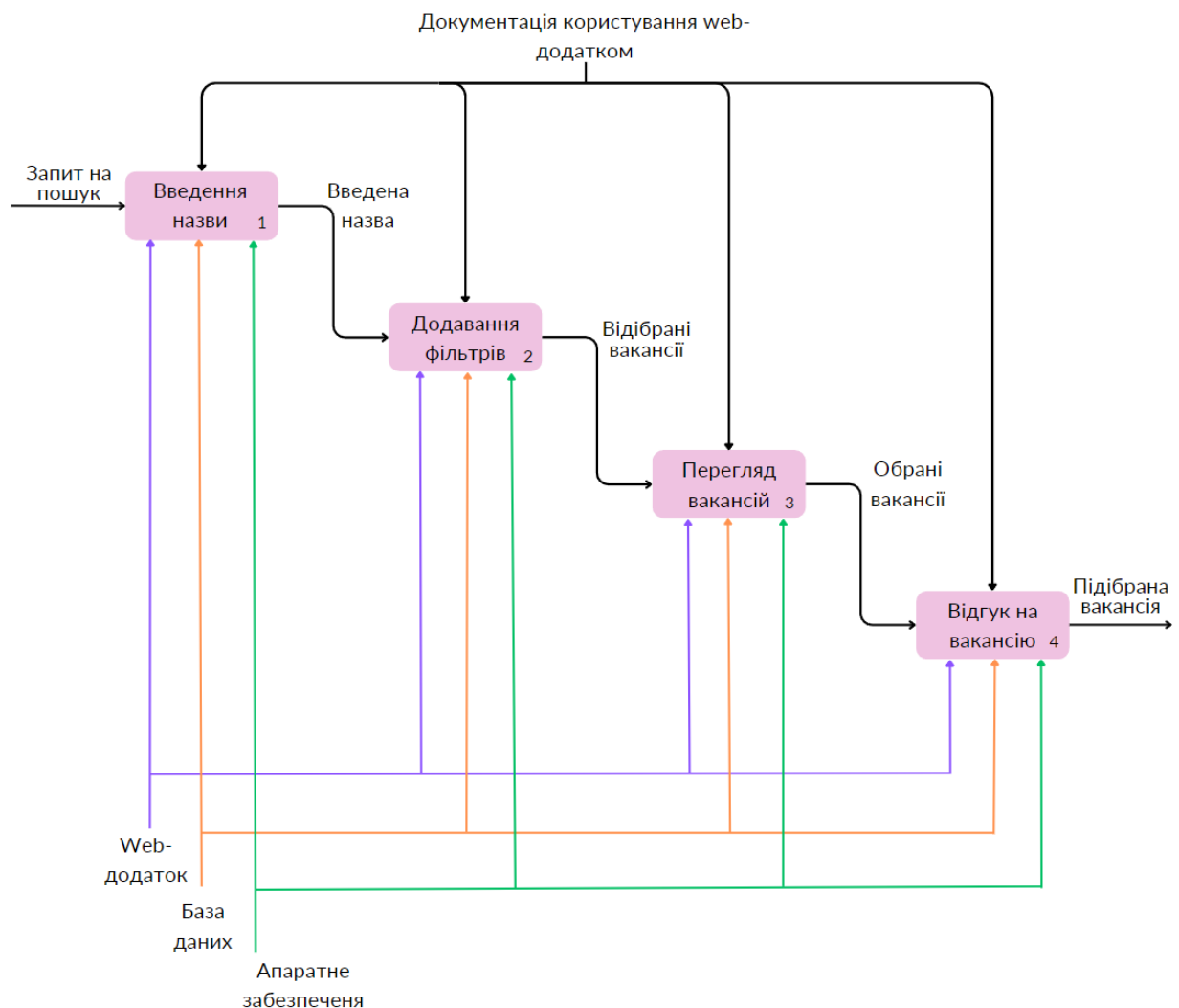


Рисунок 2.2 – Діаграма декомпозиції першого рівня в нотації IDEF0

2.2 Проектування інформаційної системи

2.2.1 Діаграма варіантів використання web-додатку

Діаграма варіантів використання – це візуальне представлення, яке відображає взаємодію між акторами (користувачами або зовнішніми системами) і системою, що розглядається. Це потужний інструмент, який використовується при розробці програмного забезпечення для відображення функціональних вимог до системи з точки зору користувача.

На діаграмі варіантів використання функціональність системи представлена варіантами використання, які є конкретними діями або завданнями, які користувачі виконують з системою. Актори зображуються у вигляді паличок і представляють ролі, які взаємодіють з системою. Варіанти використання з'єднуються з акторами лініями, що ілюструють взаємозв'язки та взаємодію між ними.

Основна мета діаграми варіантів використання – надати загальний огляд функціональності системи та зовнішніх сутностей, які з нею взаємодіють. Вона допомагає ідентифікувати різні варіанти використання і залучених до них суб'єктів, а також взаємозв'язки і залежності між ними. Діаграми варіантів використання також допомагають визначити сферу застосування системи, її межі та потенційні взаємодії між різними частинами системи.

На рисунку 2.3 представлена діаграма варіантів використання в нотації UML.



Рисунок 2.3 – Діаграма варіантів використання

У системі присутні 3 актори:

- Шукач роботи – зареєстрований користувач, що може переглядати наявні вакансії та відгукуватися на них;
- Роботодавець – зареєстрований користувач, що має можливість додавати, редагувати та видаляти вакансії, а також спілкуватися з кандидатами, що відгукнулися;
- База даних, яка надає доступ до даних.

2.2.2 Діаграма структурного моделювання

Структурне моделювання допомагає представити потік і взаємодію між базами даних і програмами. У ньому описані основні операції, які можуть бути виконаними на web-ресурсі з використанням бази даних.

На рисунку 2.4 представлена діаграма структурного моделювання роботи з вакансіями.

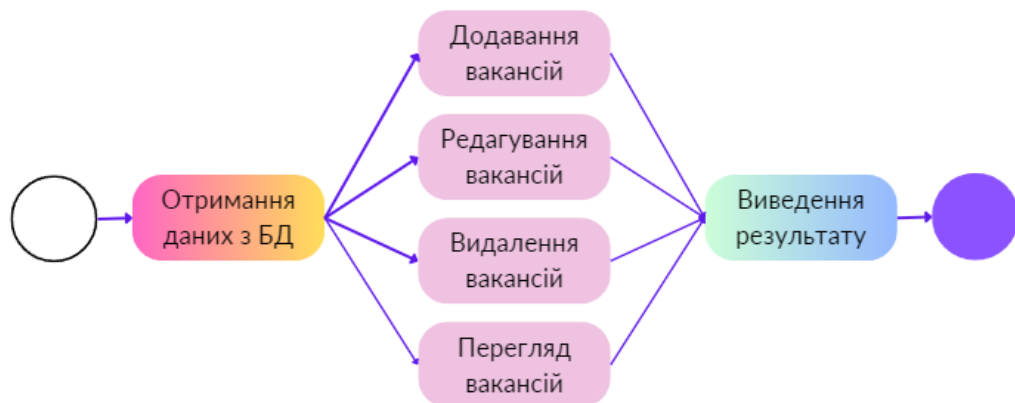


Рисунок 2.4 – Діаграма структурного моделювання роботи з вакансіями

Спочатку отримуються дані з бази даних. Після чого над ними можна виконувати такі дії:

- додавання вакансій;
- редагування вакансій;
- видалення вакансій;
- перегляд вакансій.

2.3 Проектування бази даних

Логічна модель бази даних – це представлення структури та організації даних без урахування деталей фізичної реалізації або конкретних систем управління базами даних. Вона зосереджена на визначенні зв'язків між різними сутностями та їхніми атрибутами. Логічна модель допомагає зрозуміти структуру даних в базі даних, що забезпечує ефективне проектування і розробку бази даних. Вона слугує планом для створення фізичної бази даних, спрямовуючи реалізацію та забезпечуючи цілісність і узгодженість даних.

Для побудови логічної моделі були створені такі таблиці даних:

- AspNetUsers – зберігає дані про зареєстрованих користувачів;
- Vacancies – зберігає дані про наявні вакансії;
- Categories – зберігає дані про категорії вакансій;
- Messages – зберігає дані повідомлень.

На рисунку 2.5 – 2.9 представлені таблиці створеної бази даних.

	Id	UserId	FirstName	LastName	DateRegister	DateBirthday	Role	Phone	Email	EmailConfirmed	PasswordHash	SecurityStamp	PhoneNumber	PhoneNum...
	Se...	1	Олена	Павловсь...	31.05.2023 1...	31.05.2023 1...	Employer	NULL	o_pavlovska...	False	AGQONHINO...	a8b25ec0-61...	NULL	False
	bff...	2	Наталія	Сидоренко	21.05.2023 1...	11.02.2002 0...	RegisterUser	NULL	natali_sid@...	False	Afabo8UhgS/...	04276f46-a56...	NULL	False
	N...	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

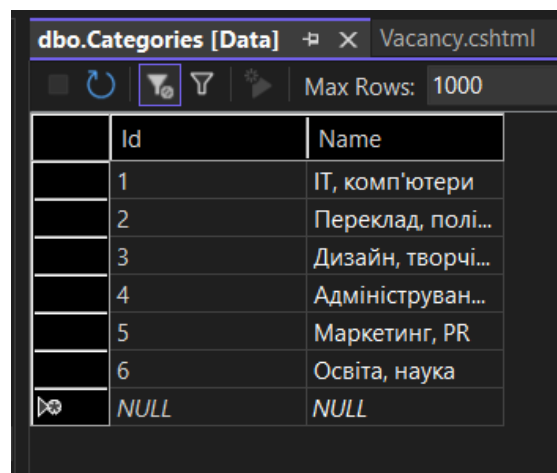
Рисунок 2.5 – Таблиця користувачів

TwoFactorE...	Lockout...	LockoutEn...	AccessFa...	UserName	Photo	Experience	Skills	City
False	NULL	True	0	o_pavlovska@g...	NULL	NULL	NULL	Суми
False	NULL	True	0	natali_sid@gma...	04143548_kristi...	2 роки	C#, JavaScript, P...	Київ
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2.6 – Продовження таблиці користувачів

Таблиця користувачів містить:

- Id – id користувача;
- FirstName – ім'я користувача;
- LastName – прізвище користувача;
- DateRegister – дата реєстрації;
- Role – роль користувача (роботодавець, шукач роботи);
- Phone – номер телефону;
- Email – електронна адреса;
- PasswordHash – зашифрований пароль;
- Photo – фото користувача;
- Experience – досвід роботи;
- Skills – навички роботи;
- City – місто проживання.



Id	Name
1	ІТ, комп'ютери
2	Переклад, полі...
3	Дизайн, творчі...
4	Адмініструван...
5	Маркетинг, PR
6	Освіта, наука
NULL	NULL

Рисунок 2.7 – Таблиця категорій

Таблиця категорій містить:

- Id – id категорії;
- Name – назву категорії.

	Id	SenderId	RecipientId	DateMessage	TextMessage	Title	IsReviewed
	1011	2	1	04.06.2023 15:2...	Доброго дня! ...	Новий відгук н...	False
	1012	2	1	04.06.2023 15:2...	Доброго дня! ...	Новий відгук н...	False
	1013	2	1	04.06.2023 15:2...	Доброго дня! ...	Новий відгук н...	False
	1014	2	1	04.06.2023 15:2...	Доброго дня! ...	Новий відгук н...	False
	1015	2	1	04.06.2023 15:5...	Доброго дня! ...	Новий відгук н...	False
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2.8 – Таблиця повідомлень

Таблиця повідомлень містить:

- Id – id повідомлення;
- SenderId – id відправника;
- RecipientId – id отримувача;
- DateMessage – дата відправлення;
- TextMessage – текст повідомлення;
- Title – заголовок листа;
- IsReviewed – чи переглянуте повідомлення.

	Id	JobTitle	JobDescription	JobPayment	CompanyName	Experience	EmployerId	CategoryId	Photo	DatePublish
	1002	Контент менедж...	У зв'язку з роз...	25 000 грн	Word Factory	Від 2 років	1	5	word.png	22.05.2023 17:5...
	1008	Senior C# Devel...	We are looking ...	6000\$	Sigma Software	4 роки	1	1	sigm.png	31.05.2023 0:00:...
	1012	Викладач Python...	Ми одна із най...	1500\$	IT School	2 роки	1	1	itschool.png	31.05.2023 14:0...
	1014	Копірайтер	Ми рекрутинго...	1000\$	Recruiter Team	2 роки	1	2	recruiter.png	31.05.2023 14:1...
	1015	Blockchain Dev...	We are currentl...	4000\$	CryptoDyor	4 years	1	1	crypto.png	04.06.2023 11:4...
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Таблиця 2.9 – Таблиця вакансій

Таблиця вакансій містить:

- Id – id вакансії;
- JobTitle – назва вакансії;
- JobDescription – опис вакансії;
- JobPayment – заробітня плата;
- CompanyName – назва компанії;

- Experience – бажаний досвід роботи;
- EmployerId – id користувача, що додав вакансію;
- CategoryId – id категорії, в якій розміщена вакансія;
- Photo – логотип компанії;
- DatePublish – дата публікації.

На рисунку 2.5 представлена логічна модель бази даних, побудована на основі створених таблиць.

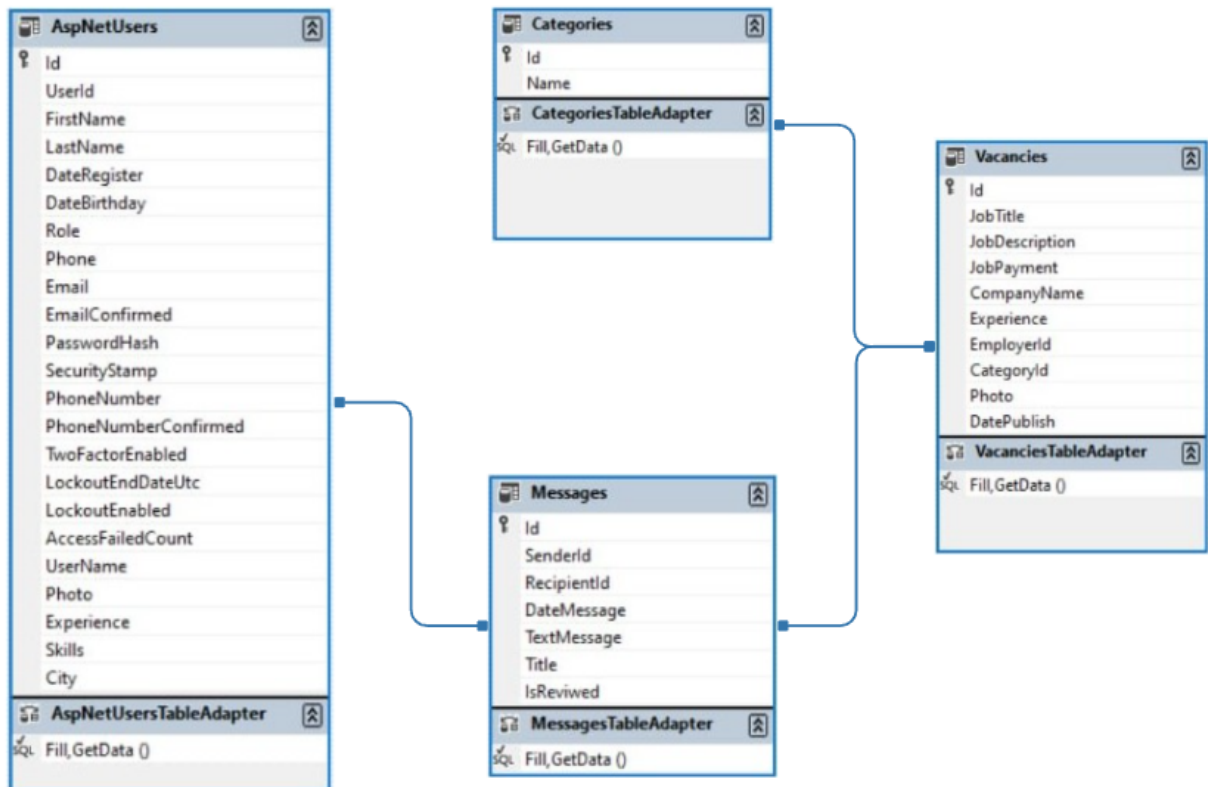


Рисунок 2.5 – Логічна модель бази даних

3 РОЗРОБКА WEB-ДОДАТКУ

3.1 Архітектура web-додатку

Для розробки бази даних була обрана трирівнева архітектура. Вона сприяє модульності, оскільки кожен рівень можна розробляти, підтримувати та масштабувати незалежно від інших. Це підвищує гнучкість і масштабованість, дозволяючи вносити зміни в один рівень, не впливаючи на інші. Крім того, вона підтримує кращу організацію коду, багаторазове використання та розподіл обов'язків.

На рисунку 2.6 представлена архітектура web-додатку.

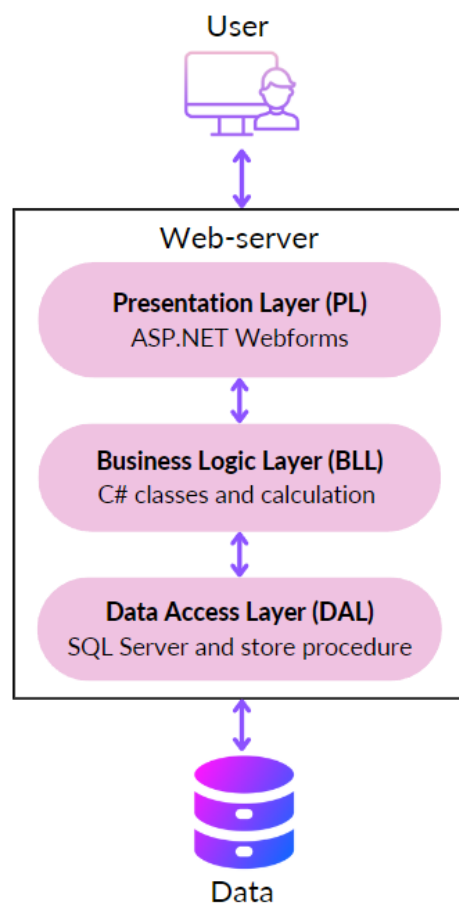


Рисунок 2.6 – Архітектура web-додатку

Рівень представлення (Presentation Layer): цей рівень відповідає за відображення інтерфейсу користувача та обробку введення від користувача. Він забезпечує взаємодію з користувачем і відображення даних для користувача.

Рівень логіки бізнес-процесів (Business Logic Layer): цей рівень відповідає за обробку бізнес-логіки та логічних операцій програми. Він включає компоненти, які обробляють дані, здійснюють розрахунки, взаємодіють з базою даних.

Рівень доступу до даних (Data Access Layer): цей рівень відповідає за збереження та доступ до даних, які використовуються програмою.

3.2 Розробка дизайну web-додатку

Для розробки шаблону дизайну web-додатку було використано графічний редактор Figma. Web-додаток був виконаний у темно-сірих кольорах. Серед користувачів web-додатку можна виділити неавторизованого користувача, роботодавця та шукача роботи. У кожного з них різні права доступу, тому головна сторінка також має різний вигляд.

На рисунку 3.1 представлена головна сторінка сайту неавторизованого користувача. Він має можливість лише переглядати наявні вакансії.

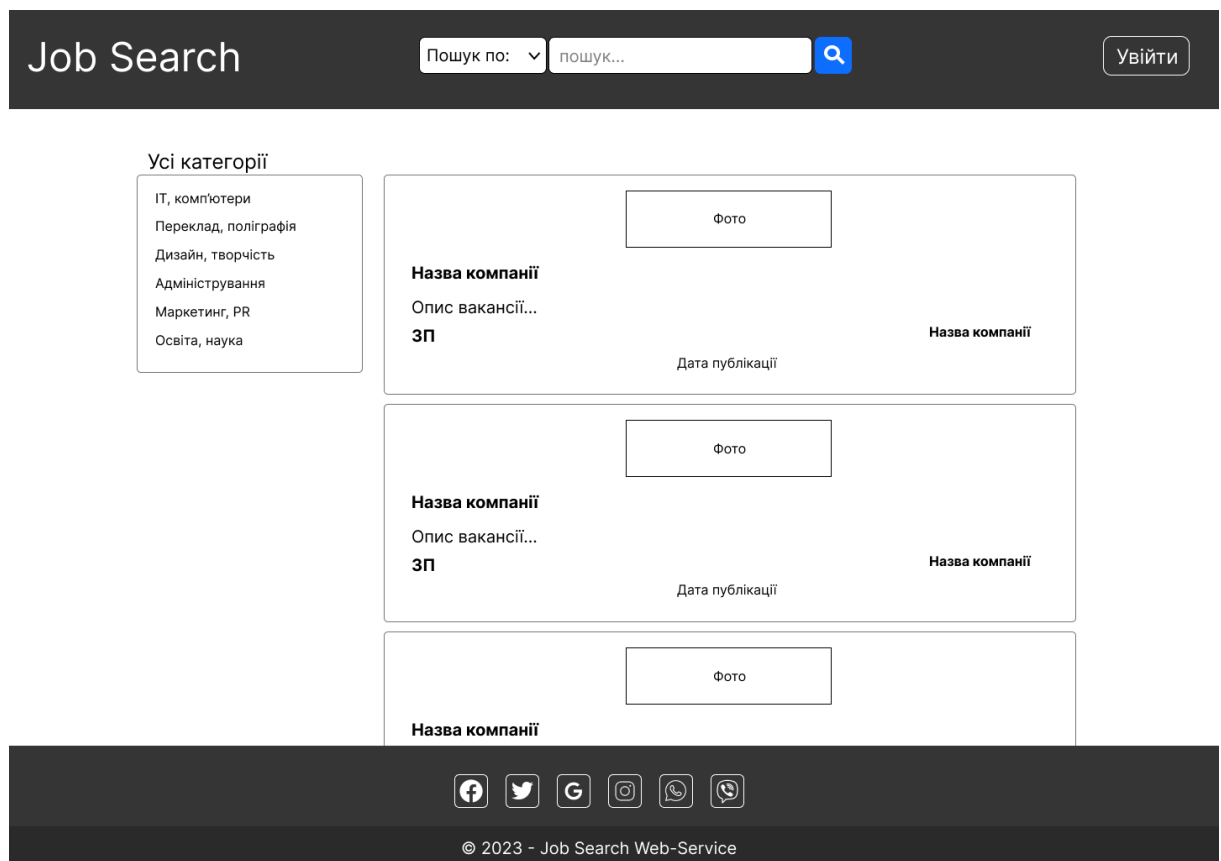


Рисунок 3.1 – Головна сторінка незареєстрованого користувача

На рисунку 3.2 представлена головна сторінка шукача роботи. Окрім перегляду вакансій, він також має можливість відгукнутися на зацікавлені варіанти та переглянути вхідні повідомлення.

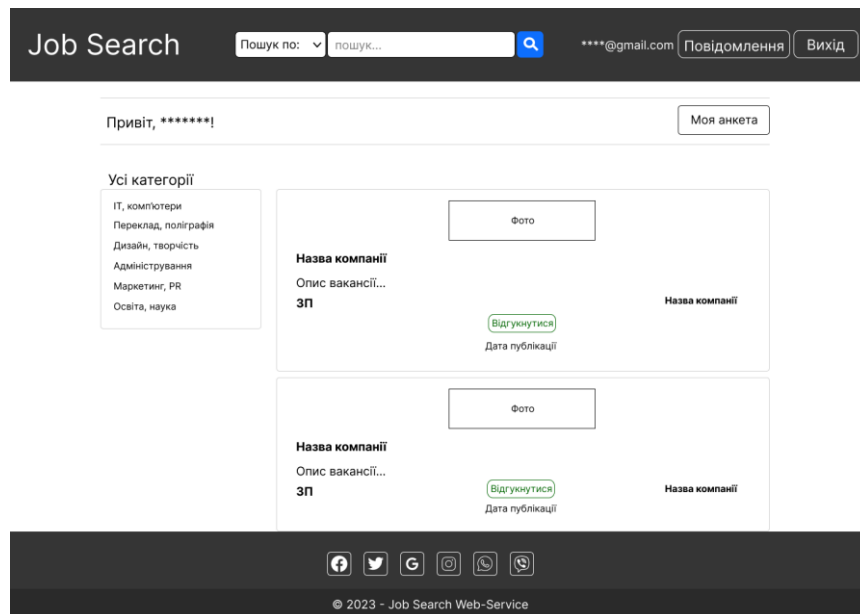


Рисунок 3.2 – Головна сторінка шукача роботи

На рисунку 3.3 представлена головна сторінка роботодавця. Він має такі ж функції, як і шукач роботи, а також має можливість додавати, редагувати та видаляти власні вакансії.

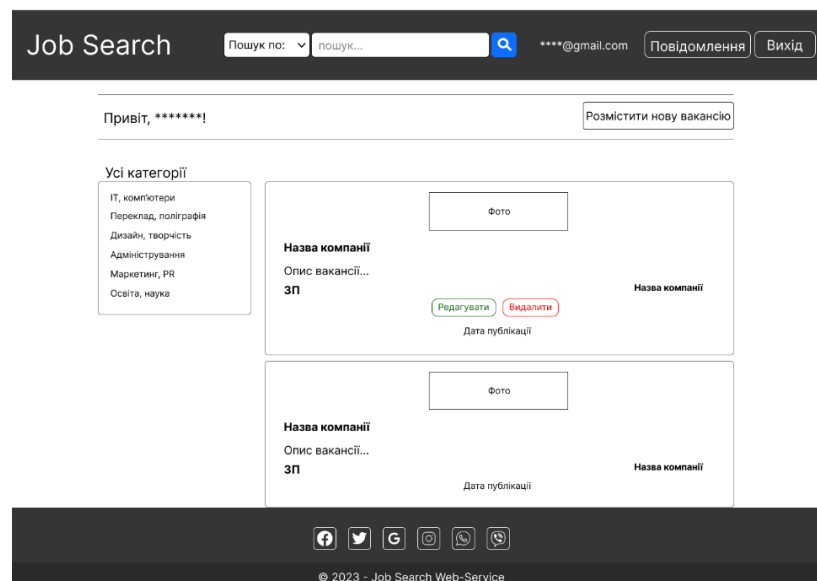


Рисунок 3.3 – Головна сторінка роботодавця

3.4 Демонстрація роботи web-додатку

Потрапляючи на головну сторінку сайту, неавторизований користувач бачить перед собою такі області:

- верхню частину сторінки, що містить логотип, рядок пошуку та кнопку «Увійти»;
- основну частину сторінки, що містить категорії вакансій та перелік самих вакансій;
- нижню частину сторінки, що містить посилання на соціальні мережі.

На рисунках 3.4-3.5 зображена головна сторінка неавторизованого користувача.

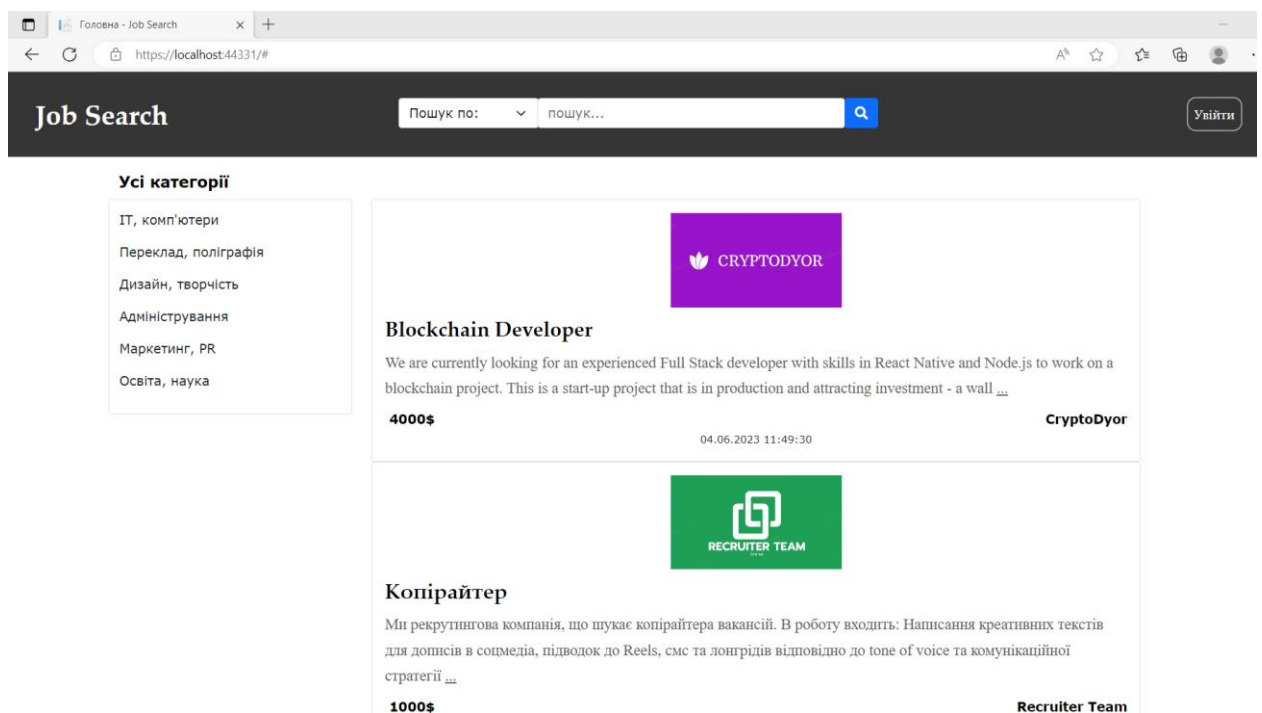


Рисунок 3.4 – Головна сторінка неавторизованого користувача

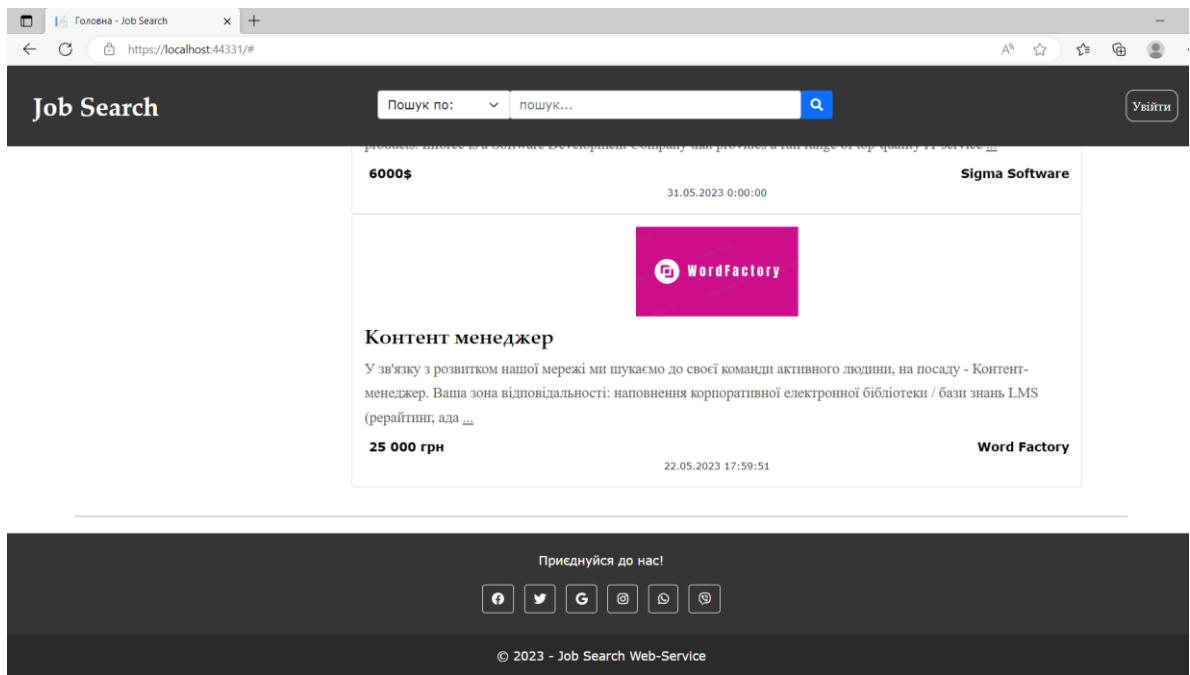


Рисунок 3.5 – Продовження головної сторінки неавторизованого користувача

Неавторизований користувач може переглядати повний опис наявних вакансій, натиснувши по контейнеру з зацікавленою вакансією. На рисунку 3.6 представлена сторінка з повним описом вакансії. Тут користувач може побачити логотип компанії, дізнатися деталі роботи, оплати та актуальність вакансії за датою публікації.

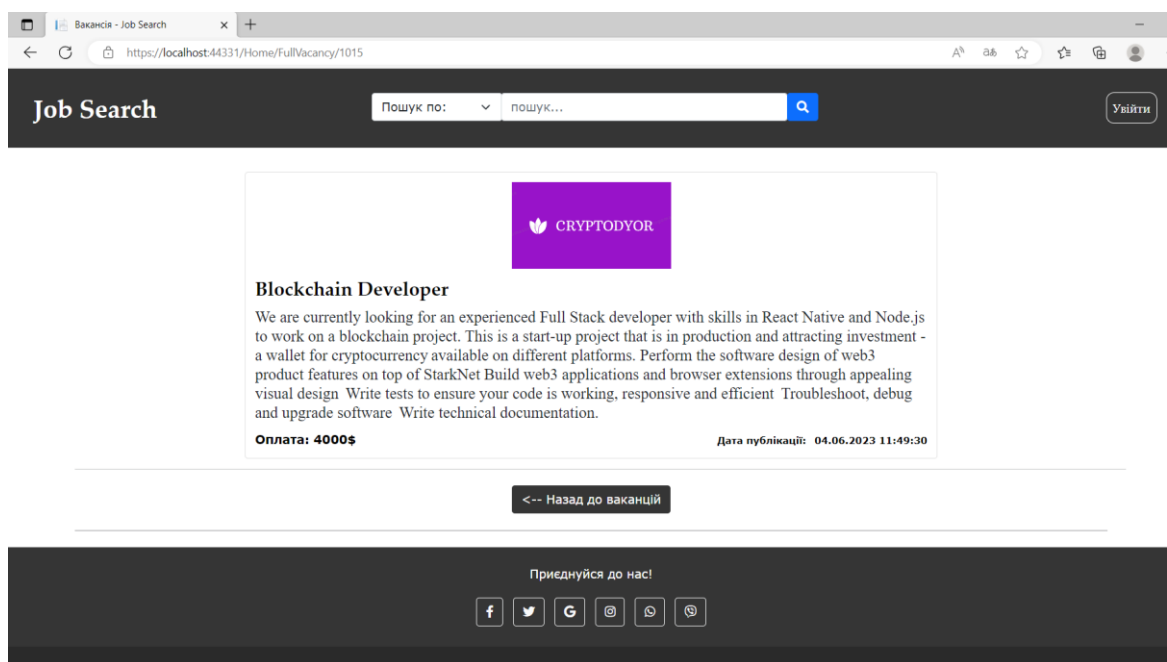


Рисунок 3.6 – Сторінка повного опису вакансії

Також, у неавторизованого користувача є можливість шукати вакансію за її назвою або компанією, що її опублікувала. Для цього необхідно обрати відповідне поле у випадаючому списку та ввести у рядок пошуку назву.

На рисунках 3.7-3.8 представлена робота пошуку за назвою вакансії.

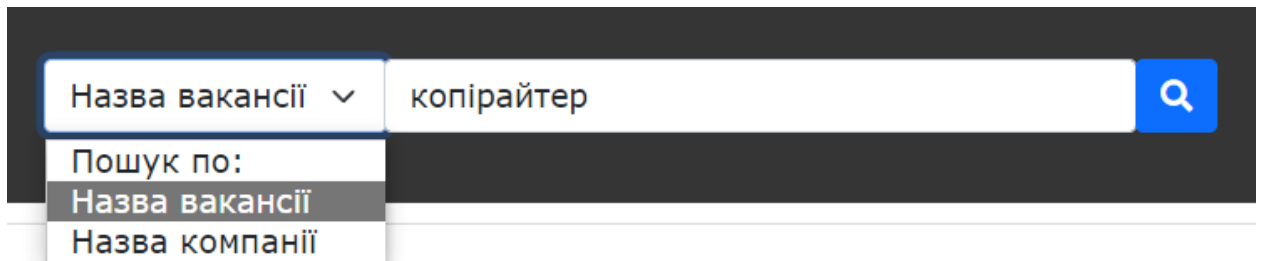


Рисунок 3.7 – Рядок пошуку за назвою вакансії

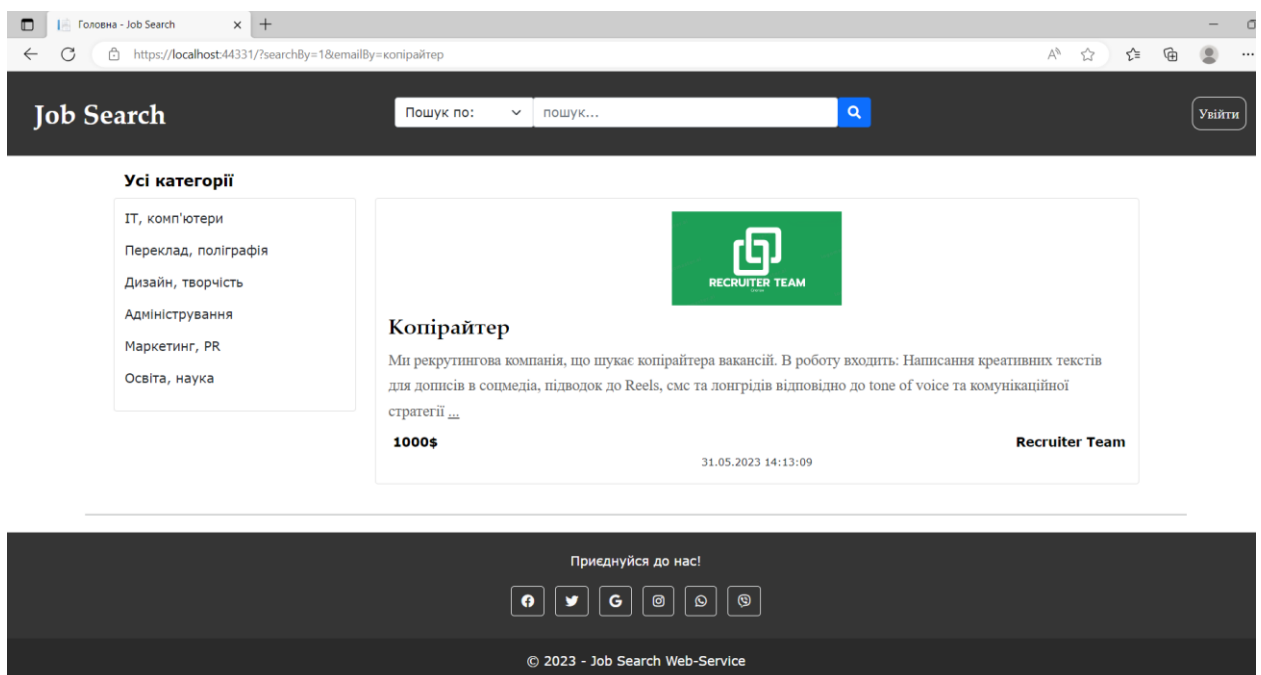


Рисунок 3.8 – Результат пошуку за назвою вакансії

На рисунках 3.9-3.10 представлена робота пошуку за назвою компанії.



Рисунок 3.9 – Рядок пошуку за назвою компанії

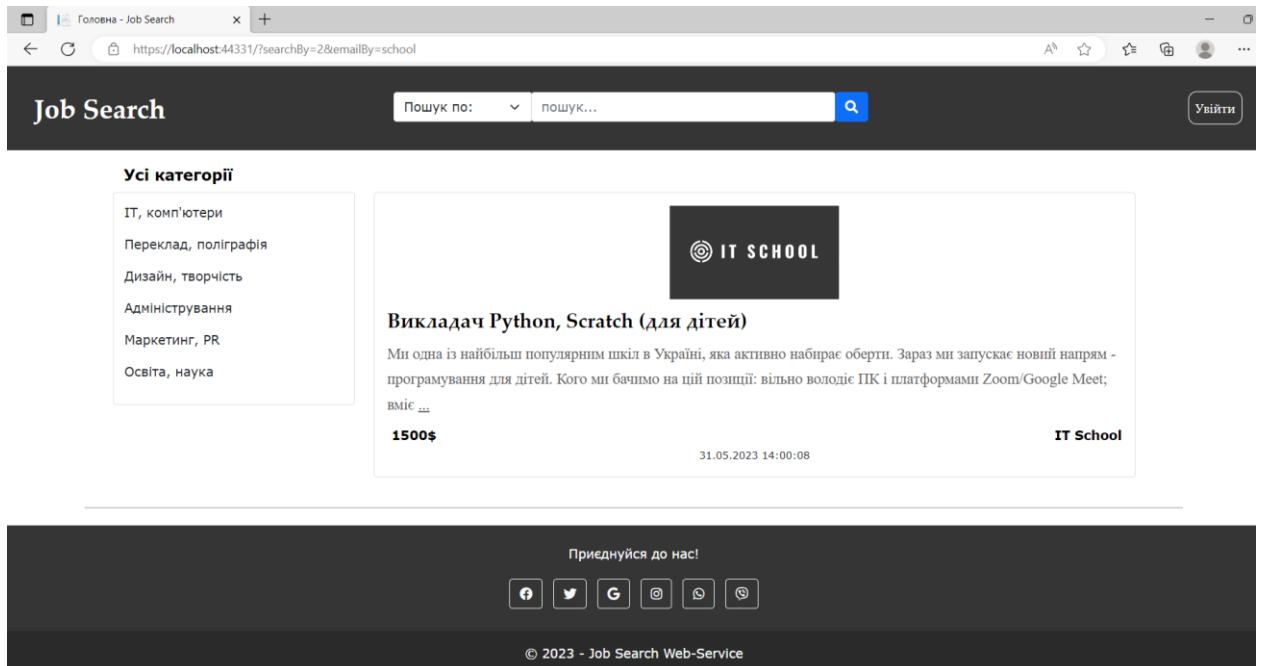


Рисунок 3.10 – Результати пошуку за назвою компанії

Також є можливість фільтрувати вакансії за категоріями. Для цього необхідно серед наявного списку обрати категорію та натиснути на неї. На рисунках 3.11-3.12 представлено фільтрування вакансій за категоріями.

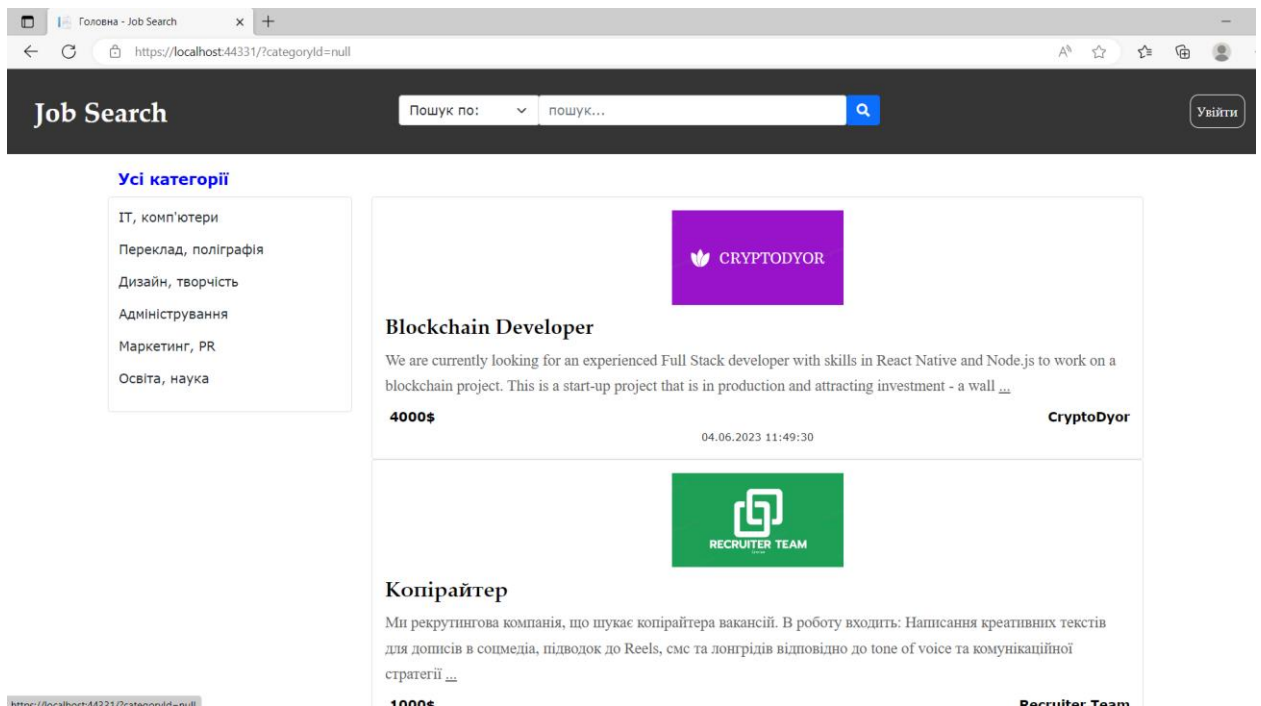


Рисунок 3.11 – Перегляд усіх категорій

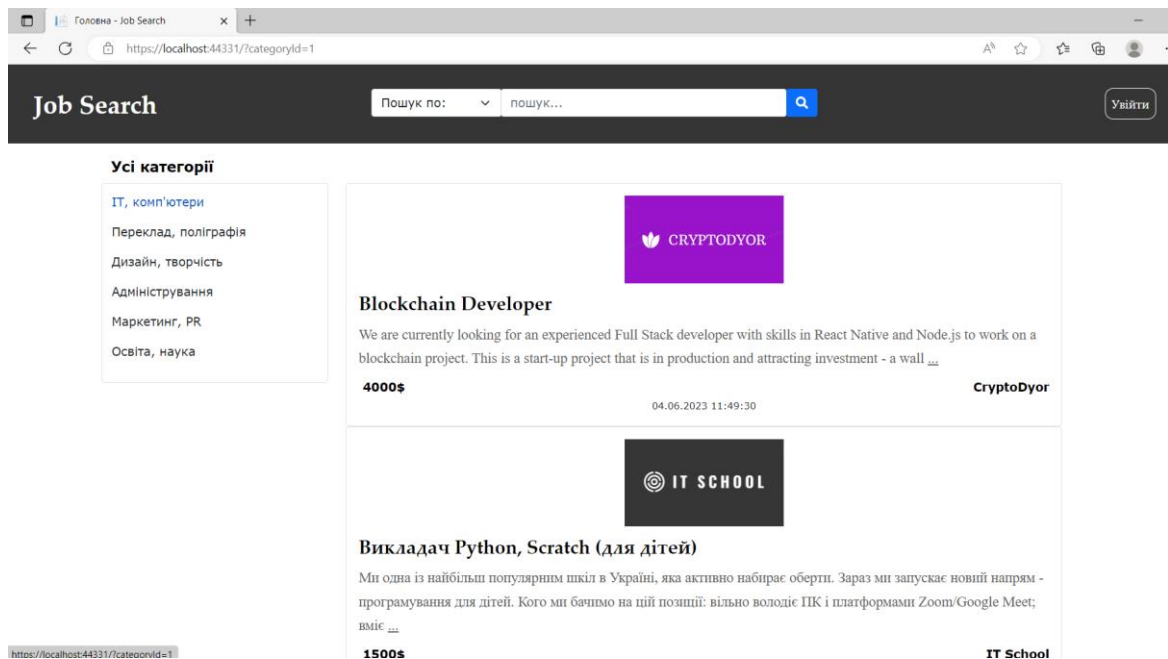


Рисунок 3.12 – Перегляд вакансій в категорії «ІТ, комп'ютери»

Для того, щоб збільшити можливості користування сайтом користувачу необхідно авторизуватися або зареєструватися.

На рисунку 3.13 зображена форма авторизації. Якщо користувач вже має акаунт, то йому необхідно ввести пошту до якої прив'язаний акаунт та пароль, що був придуманий при реєстрації. Після введення даних натиснути кнопку «Увійти».

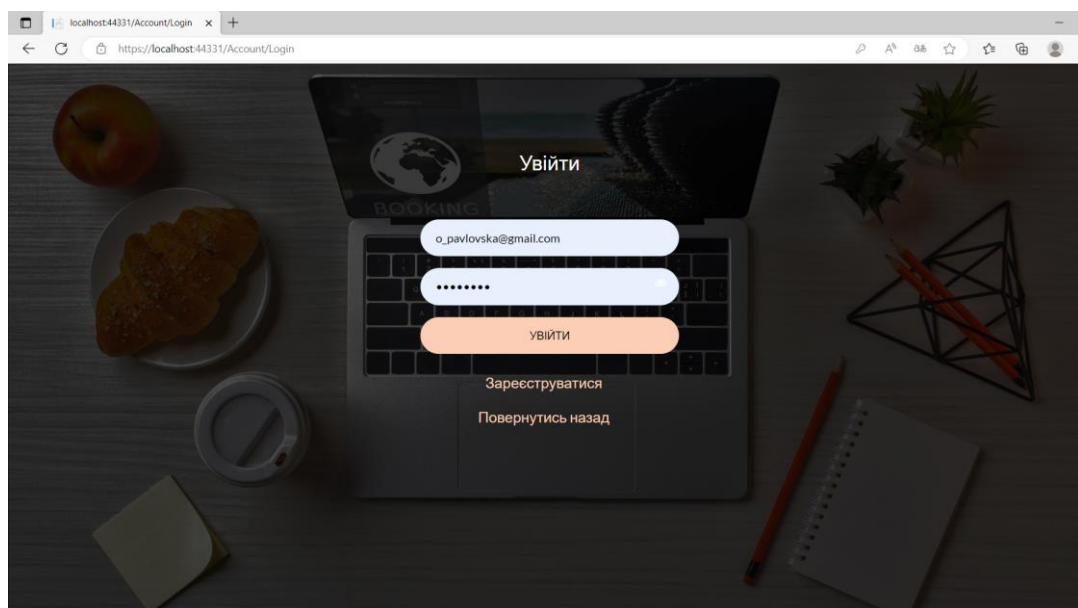


Рисунок 3.13 – Форма авторизації

Якщо ж користувач ще не має акаунту, то необхідно натиснути на кнопку «Зареєструватися», що веде на сторінку з формою реєстрації (рис. 3.14). У цій формі необхідно заповнити усі свої дані, такі як: ім'я, прізвище, електронну адресу, обрати роль з випадаючого списку (рис. 3.15) під якою буде здійснено вхід (роботодавець або шукач роботи), місто, пароль та підтвердження паролю. Після заповненні натиснути на кнопку «Зареєструватися».

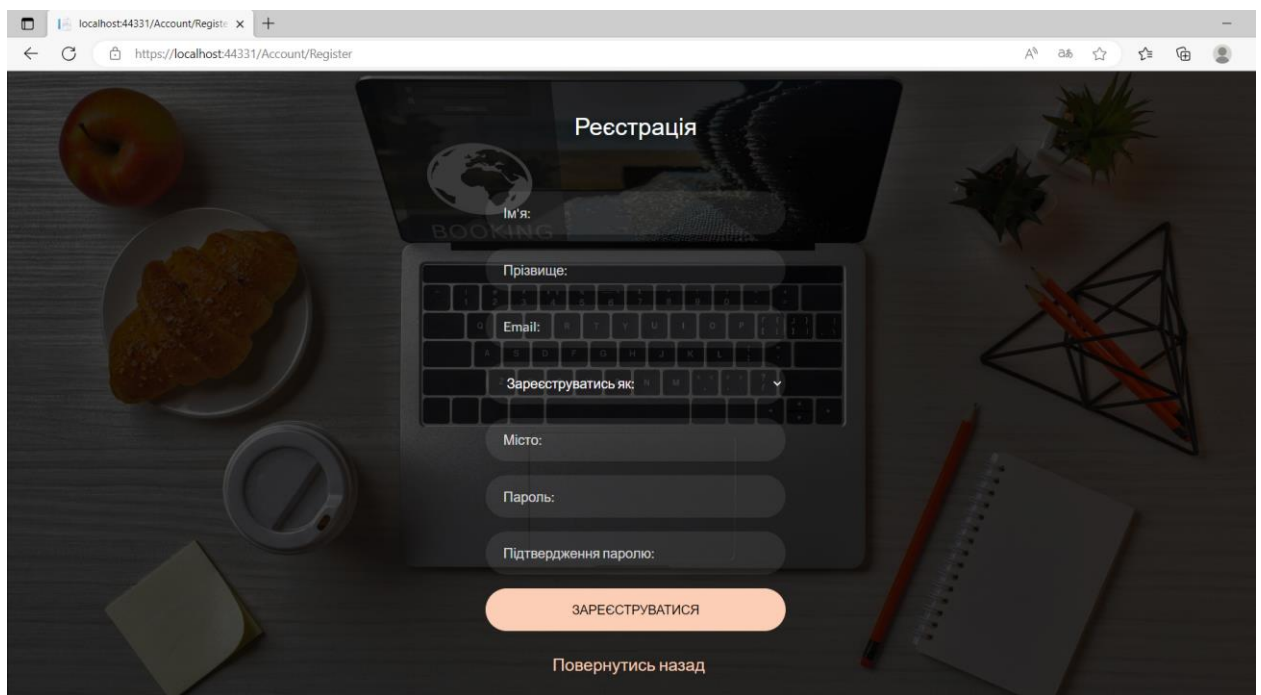


Рисунок 3.14 – Форма реєстрації

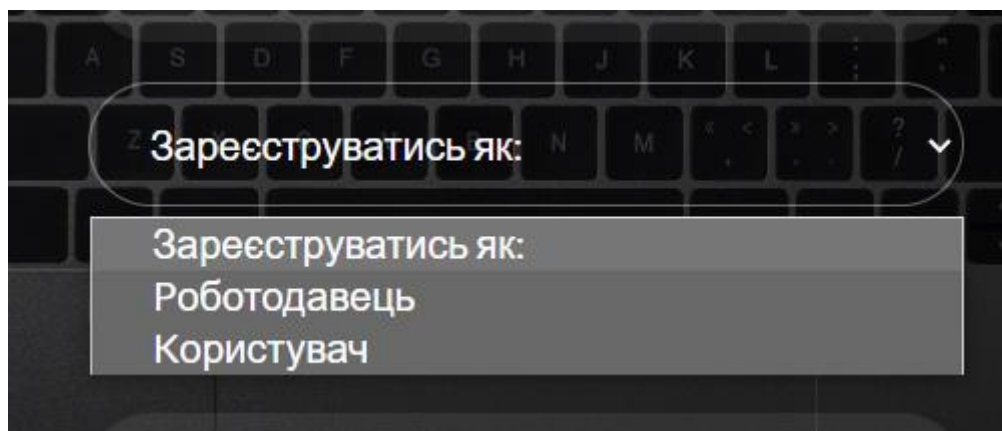


Рисунок 3.15 – Перелік ролей для реєстрації

Зареєструвавшись як роботодавець, користувач потрапляє на головну сторінку (рис. 3.16), на якій має можливість розмістити нову вакансію, редагувати та видалити наявні.

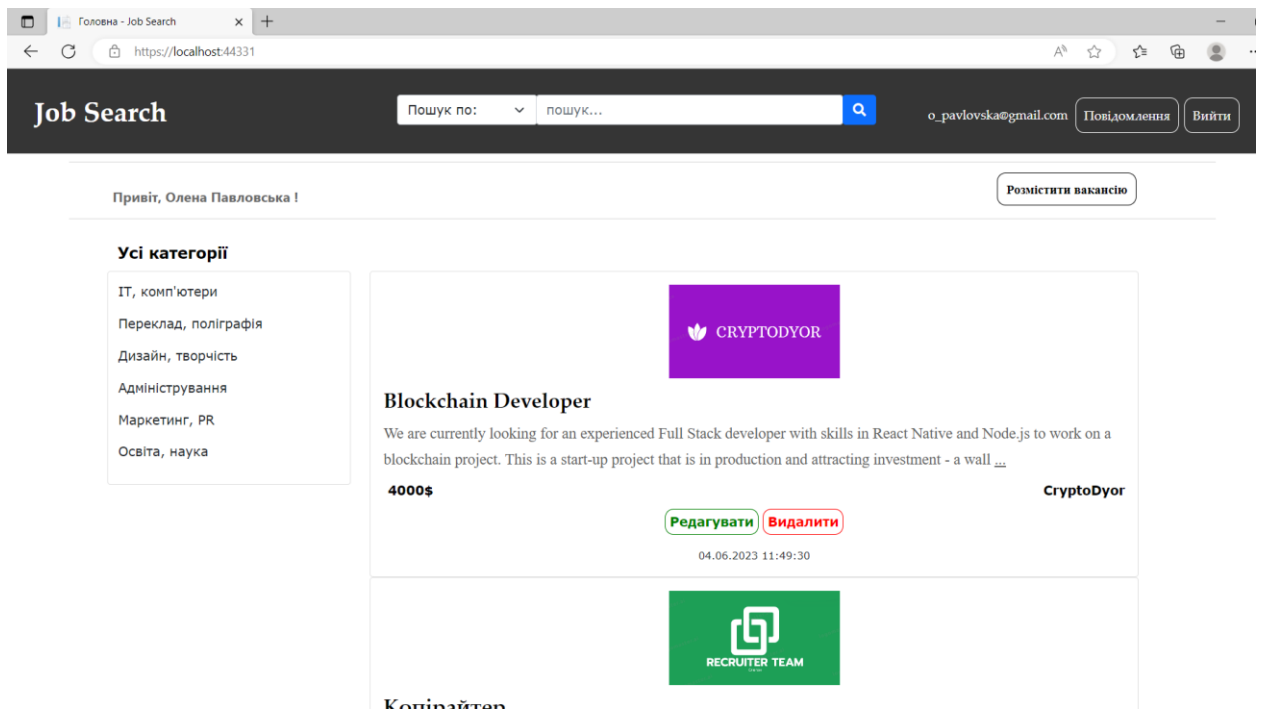
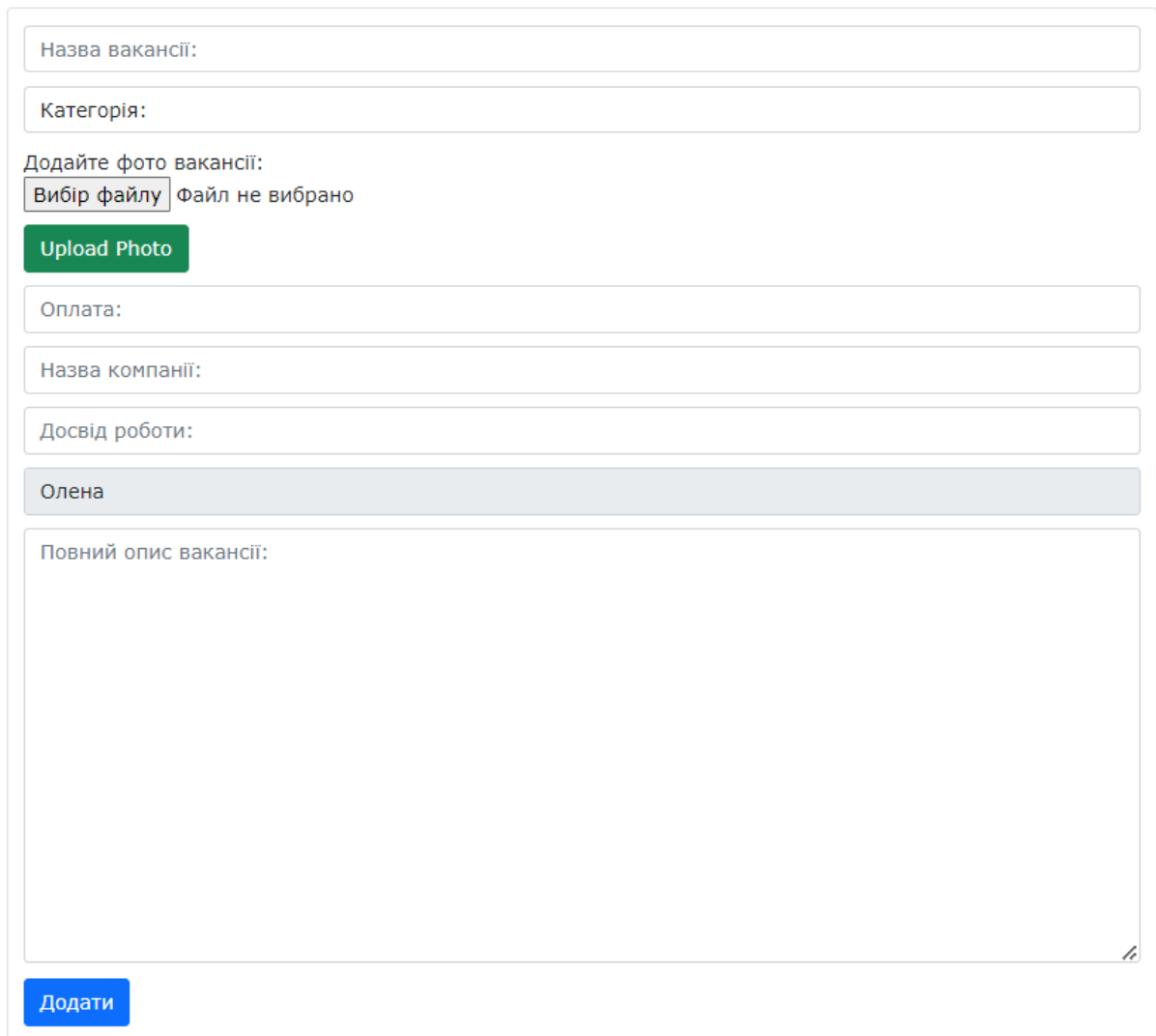


Рисунок 3.16 – Головна сторінка роботодавця

Для того, щоб додати нову вакансію, необхідно натиснути на кнопку «Розмістити вакансію», що веде на сторінку з формою додавання. В цій формі необхідно заповнити всі наявні поля, такі як назва вакансії, категорію, в якій вона буде розміщена, фото-логотип компанії, заробітну плату, назву компанії, досвід роботи для даної посади, та максимально детально описати вимоги до кандидата. Далі треба натиснути кнопку «Додати», після чого нова вакансія з'явиться в списку усіх вакансій.

На рисунку 3.17 представлена форма для додавання нової вакансії.

Додати нову вакансію



Назва вакансії:

Категорія:

Додайте фото вакансії:

Вибір файлу Файл не вибрано

Upload Photo

Оплата:

Назва компанії:

Досвід роботи:

Олена

Повний опис вакансії:

Додати

Рисунок 3.17 – Форма додавання нової вакансії

Для редагування опублікованої вакансії необхідно знайти вакансію, що треба редагувати, та натиснути поруч на кнопку «Редагувати», що веде на сторінку з формою редагування. У формі є можливість обрати поле для редагування, замінити в ньому текст/фото та натиснути «Оновити». Оновлена інформація з’явиться на сторінці вакансії.

На рисунку 3.18 представлена форма редагування вакансії.

Редагувати вакансію

Blockchain Developer

ІТ, комп'ютери

Додайте фото вакансії:

Вибір файлу Файл не вибрано

Upload Photo

4000\$

CryptoDyor

4 years

Олена

We are currently looking for an experienced Full Stack developer with skills in React Native and Node.js to work on a blockchain project.

This is a start-up project that is in production and attracting investment - a wallet for cryptocurrency available on different platforms.

Perform the software design of web3 product features on top of StarkNet

Build web3 applications and browser extensions through appealing visual design

Write tests to ensure your code is working, responsive and efficient

Troubleshoot, debug and upgrade software

Write technical documentation.

Оновити

Рисунок 3.18 – Форма редагування вакансій

Для того, щоб змінити фотографію компанії, треба натиснути на «Вибір файлу», обрати на пристрої фото та натиснути «Upload Photo», після чого буде повідомлення про успішне завантаження.

На рисунках 3.19-3.21 представлений алгоритм додавання фото.

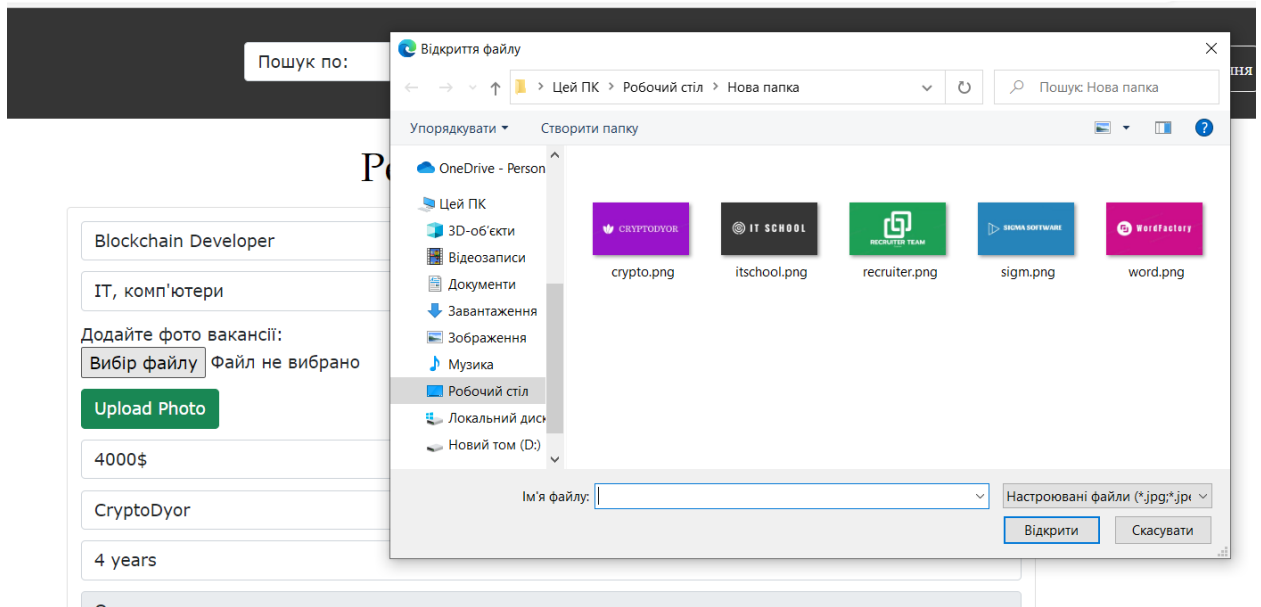


Рисунок 3.19 – Вибір файлу на пристрої

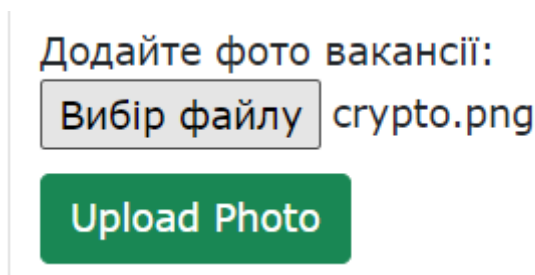


Рисунок 3.20 – Зміна «Файл не вибрано» на назву обраного фото

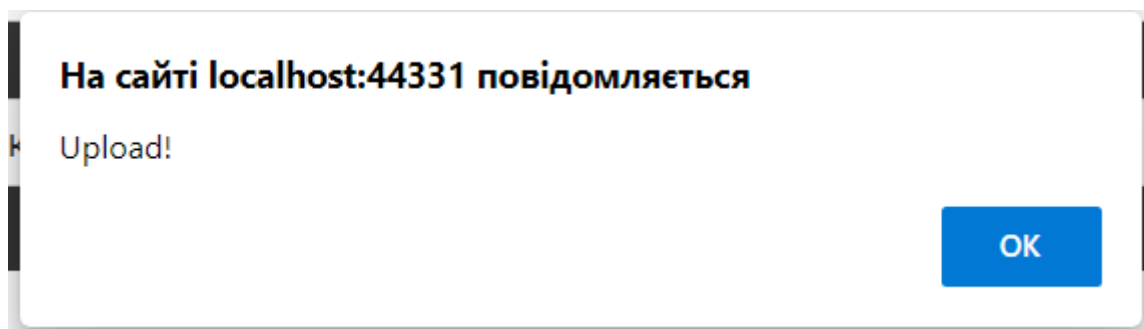


Рисунок 3.21 – Повідомлення про успішну заміну фото

Для того, щоб видалити наявну вакансію, необхідно знайти ту, яку треба видалити, та поруч з нею натиснути на «Видалити». Після цього користувача буде переміщено на сторінку з видаленням та запитано, чи дійсно він хоче

видалити відповідну вакансію. Далі необхідно просто натиснути на «Видалити» і відповідні зміни можна буде побачити на головній сторінці.

На рисунку 3.22 зображена сторінка видалення.

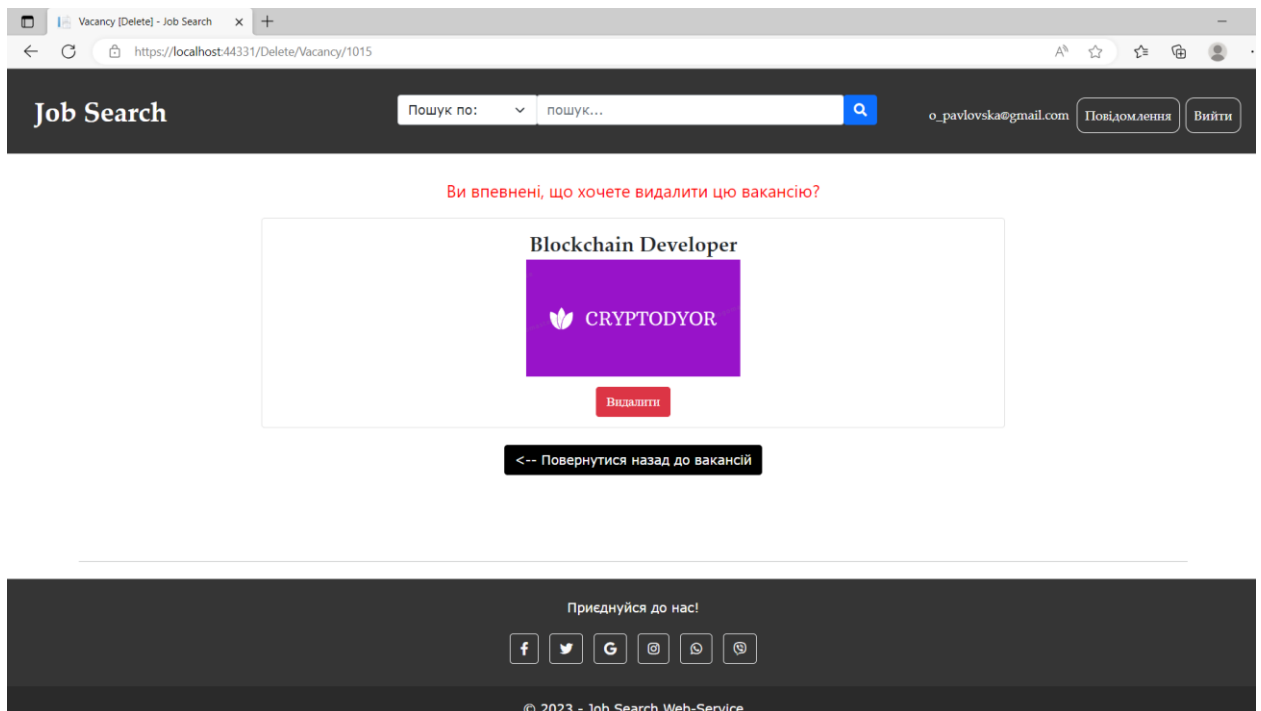


Рисунок 3.22 – Сторінка видалення вакансії

Викладаючи вакансію, роботодавець може отримувати відгуки на неї безпосередньо у web-додатку. Для перегляду відкликів йому потрібно перейти у вкладку «Повідомлення», що закріплена у шапці кожної web-сторінки. На цій сторінці видно усі повідомлення, включаючи адресу відправника, заголовок листа та час коли воно було відправлено.

Сторінка вхідних повідомлень зображена на рисунку 3.23.

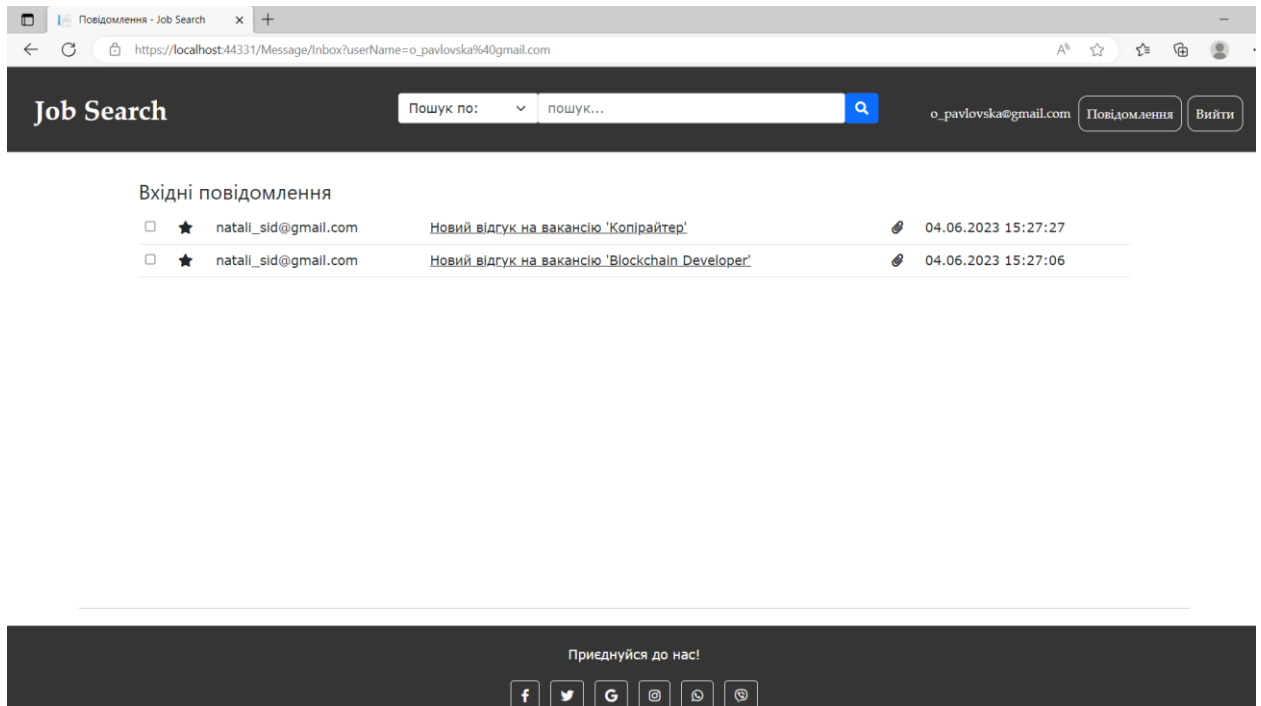


Рисунок 3.23 – Сторінка вхідних повідомлень

Для того, щоб переглянути повне повідомлення, необхідно натиснути на заголовок листа, після чого відкриється сторінка з описом кандидата та його досвідом роботи.

На рисунку 3.24 показано сторінку відкритого повідомлення.

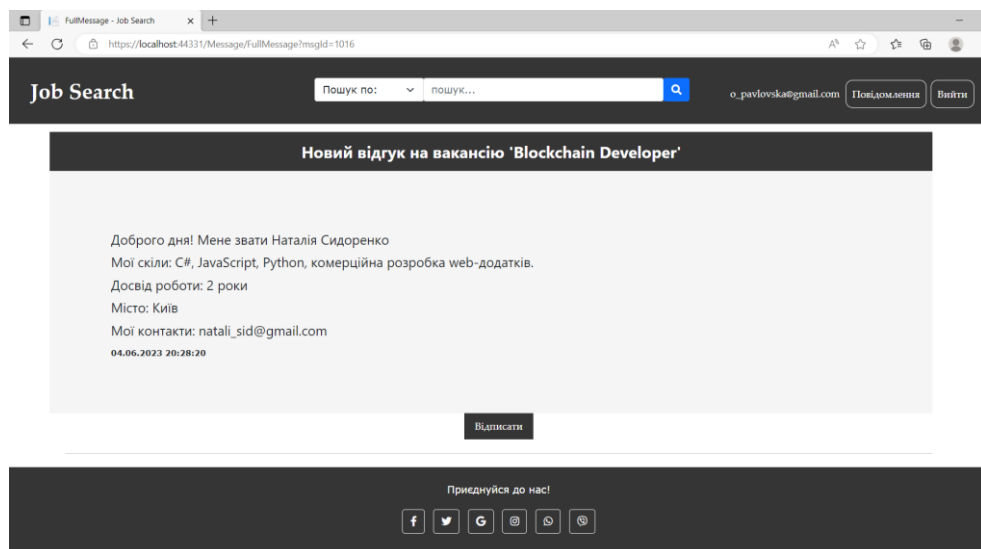


Рисунок 3.24 – Сторінка повідомлення

Щоб дати відповідь на повідомлення кандидата, потрібно натиснути на «Відписати», після чого з'явиться вікно, в якому буде автоматично вписані

електронні адреси відправника та отримувача, тема листа, яку можна змінити, та поле для тексту повідомлення. Після написання тексту необхідно натиснути «Надіслати», в наслідок чого лис буде відправлений отримувачу.

На рисунку 3.25 представлена форма для написання повідомлення.

Рисунок 3.25 – Форма для відправки повідомлення

Далі розглянемо сторінку користувача, який авторизувався, як шукач роботи. На головній сторінці (рис 3.26) у нього з'являється кнопка «Моя анкета» та «Відгукнутися».

Рисунок 3.26 – Головна сторінка шукача роботи

Для перегляду або редагування своїх даних треба натиснути на «Моя анкета», після цього відкриється сторінка з формою для заповнення особистої інформації (рис. 3.27). Тут є можливість змінити фото профілю, а також вказати: прізвище, ім'я, досвід роботи, місто проживання, електронну адресу, номер мобільного телефону, дату народження та професійні навички. Для редагування треба просто заповнити відповідні поля та натиснути на кнопку «Зберегти».

Саме ці дані будуть використовуватися і відправлятися разом з відгуком на вакансію.

The screenshot shows a web browser window with the URL `https://localhost:44331/Home/UserProfile`. The page header includes the text 'Job Search' and a search bar. The user's email is `natali_sid@gmail.com`. The profile section is divided into two main parts: 'Резюме' (Resume) and 'Деталі облікового запису' (Account Details). The 'Резюме' section features a circular profile picture of a woman, with a file upload button labeled 'Вибір файлу' and a 'Завантажте своє фото' button. Below the photo, it specifies 'JPG or PNG no larger than 5 MB'. The 'Деталі облікового запису' section contains several input fields: 'Прізвище' (Last name) with 'Наталія', 'Ім'я' (First name) with 'Сидоренко', 'Досвід роботи' (Work experience) with '2 роки', 'Місто' (City) with 'Київ', 'Ел. пошта' (Email) with 'natali_sid@gmail.com', 'Моб.' (Mobile) with an empty field, and 'Дата народження' (Date of birth) with a date picker showing 'дд.мм.рррр'. A 'Навички' (Skills) text area contains the text 'C#, JavaScript, Python, комерційна розробка web-додатків.'. A blue 'Зберегти' (Save) button is located at the bottom of the form.

Рисунок 3.27 – Анкета користувача

Щоб надіслати відгук на зацікавлену вакансію, треба натиснути на кнопку «Відгукнутися» поруч з текстом вакансії, після чого з'явиться вікно про успішно відправлений відгук (рис. 3.28).

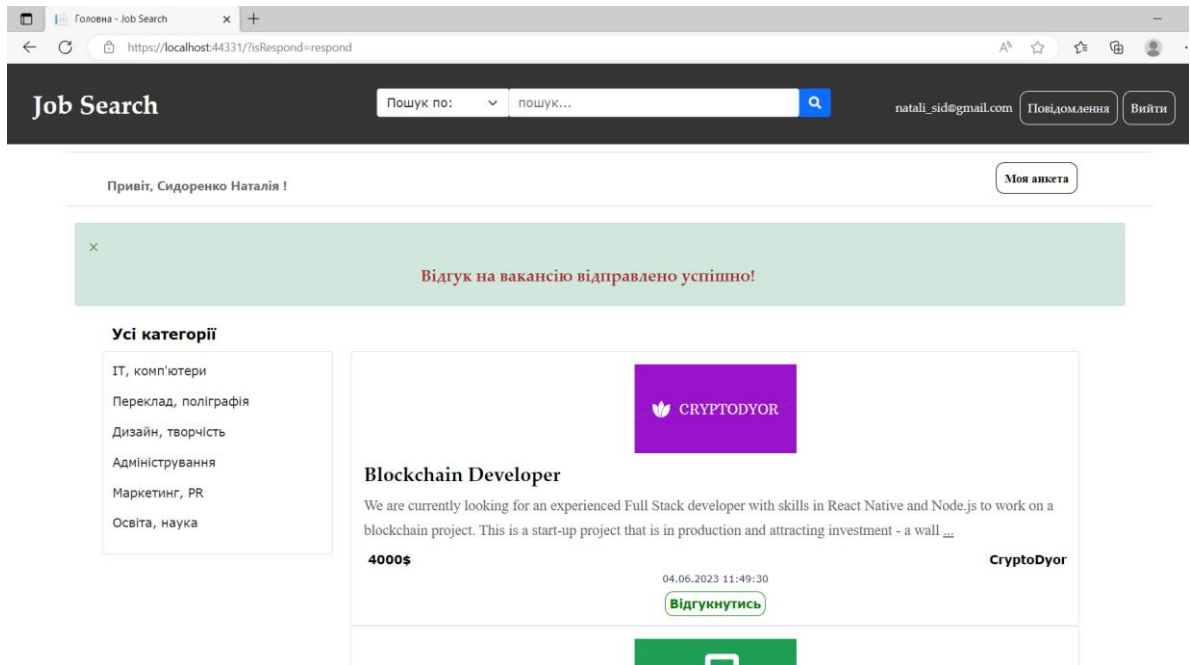


Рисунок 3.28 – Сповіднення про успішний відгук

3.5 Тестування web-додатку

Тестування web-додатків – це важливий крок у життєвому циклі розробки програмного забезпечення. Воно включає в себе оцінку функціональності, продуктивності, безпеки та зручності використання додатку, щоб переконатися, що він відповідає бажаним вимогам.

Першим кроком було перевірено клікабельність усіх кнопок та можливість переміщення по ним на відповідні сторінки. Таким чином, було перевірено:

Кнопки наявні у всіх користувачів:

- Логотип – натиснувши на нього, користувач потрапляє на головну сторінку;
- Пошук – шукає наявні вакансії по запити.
- Увійти – веде на сторінку авторизації;
- Зареєструватися – веде на сторінку реєстрації;
- Вийти – виходить з аканта користувача;
- Іконки соціальних мереж – ведуть на сторінки відповідних мереж.

Сторінка повідомлення:

- Повідомлення – веде на сторінку вхідних повідомлень;
- Відписати – з'являється вікно для написання повідомлення;
- Надіслати – відправляє листа отримувачу.

Сторінка анкети користувача, що авторизований як шукач роботи:

- Моя анкета – веде на сторінку з особистими даними;
- Вибір файлу – відкриває сховище пристрою для вибору фото;
- Завантажте своє фото – змінює фото на обране;
- Зберегти – зберігає оновлені дані.

Головна сторінка шукача роботи:

- Відгукнутись – відправляє відгук на вакансію роботодавцю.

Сторінка додавання вакансії для користувача, що авторизований як роботодавець:

- Розмістити вакансію – веде на сторінку з формою додавання вакансії;
- Upload Photo – завантажує нове фото-логотип компанії;
- Додати – зберігає вакансію в базі та розміщує її на головній сторінці.

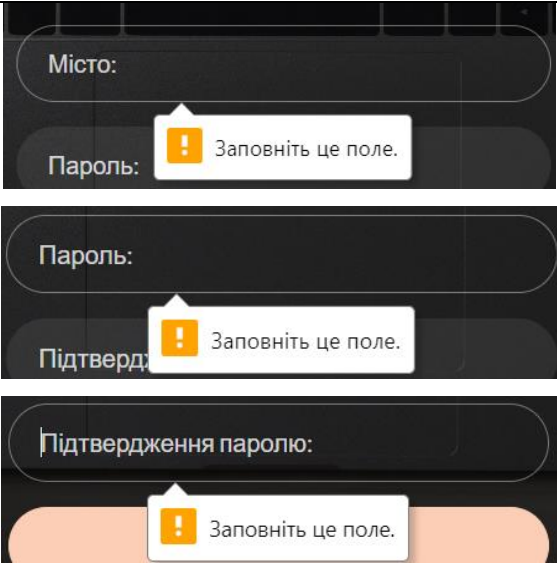
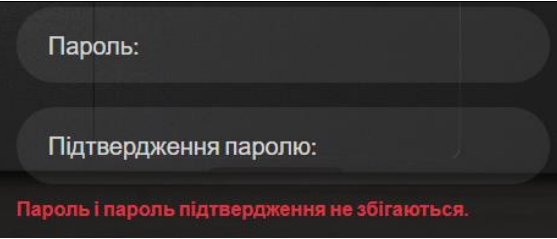
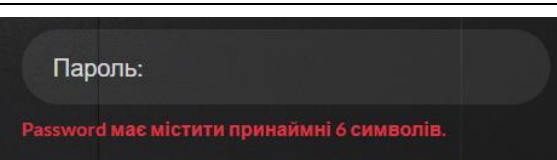
Головна сторінка роботодавця:

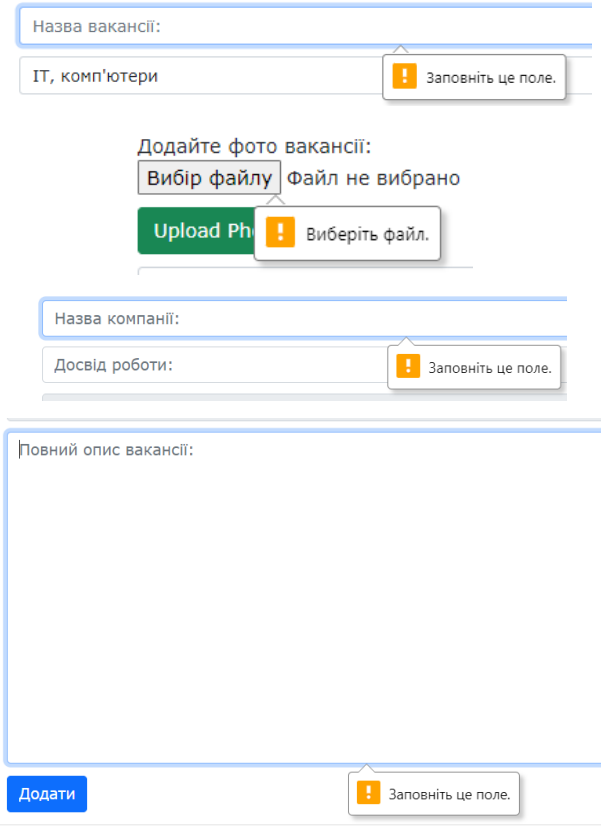
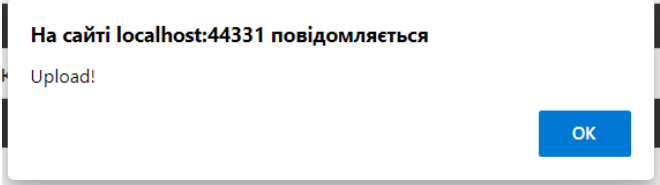
- Редагувати – веде на форму з редагуванням;
- Видалити – веде на сторінку з видаленням.

Усі наведені вище кнопки працюють коректно. Далі проведемо тестування форм та наявність перевірки введених даних. Результати тестування представлені у таблиці 3.1.

Таблиця 3.1 – Тестування форм

№	Назва	Очікуваний результат	Фактичний результат	Чи пройдена перевірка
1	Не заповнені поля даних у формі авторизації	Повідомлення про незаповнені поля.		Так
2	Не заповнені поля у формі реєстрації	Повідомлення про незаповнені поля.		Так

№	Назва	Очікуваний результат	Фактичний результат	Чи пройдена перевірка
2	Не заповнені поля у формі реєстрації	Повідомлення про незаповнені поля.	 <p>The first screenshot shows the 'Місто:' field with a validation message 'Заповніть це поле.' (Fill this field). The second screenshot shows the 'Пароль:' field with the same message. The third screenshot shows the 'Підтвердження паролю:' field with the same message.</p>	Так
3	Не співпадіння паролю та його підтвердження	Повідомлення, що паролі не збігаються	 <p>The screenshot shows the 'Пароль:' and 'Підтвердження паролю:' fields. A red error message at the bottom states: 'Пароль і пароль підтвердження не збігаються.' (Password and password confirmation do not match).</p>	Так
4	Пароль містить менше 6 символів	Повідомлення про те, що пароль має містити більше 6 символів	 <p>The screenshot shows the 'Пароль:' field. A red error message at the bottom states: 'Password має містити принаймні 6 символів.' (Password must contain at least 6 characters).</p>	Так

№	Назва	Очікуваний результат	Фактичний результат	Чи пройдена перевірка
5	Не заповнені поля у формі додавання вакансії	Повідомлення про незаповнені поля	 <p>Назва вакансії: IT, комп'ютери ! Заповніть це поле.</p> <p>Додайте фото вакансії: Вибір файлу Файл не вибрано Upload Photo ! Виберіть файл.</p> <p>Назва компанії: Досвід роботи: ! Заповніть це поле.</p> <p>Повний опис вакансії: ! Заповніть це поле.</p> <p>Додати</p>	Так
6	Успішне завантаження фото	Повідомлення про успішне завантаження	 <p>На сайті localhost:44331 повідомляється Upload! OK</p>	Так

№	Назва	Очікуваний результат	Фактичний результат	Чи пройдена перевірка
7	Відсутність вакансій за запитом	Повідомлення про відсутність вакансії	За даним запитом вакансії відсутні!	Так
8	Відгук на вакансію	Повідомлення про успішне відправлення відгуку	Відгук на вакансію відправлено успішно!	Так

За результатами тестування можна зробити висновок, що програмний продукт працює правильно та надає користувачу супровідні повідомлення.

ВИСНОВОК

У сучасному світі, де технології займають провідне місце в нашому повсякденному житті, Інтернет став важливим інструментом для шукачів роботи в їхньому прагненні працевлаштуватися. Завдяки поширенню веб-сайтів з пошуку роботи, шукачі роботи тепер можуть отримати доступ до списків вакансій з різних джерел всього за кілька кліків. Це зробило процес пошуку роботи набагато більш керованим та ефективним.

Метою створеного веб-додатку було полегшення процесу пошуку роботи як для роботодавців, так і для шукачів. Роботодавці можуть розмішувати свої вакансії та шукати кваліфікованих кандидатів, а шукачі роботи можуть створити профіль, який демонструє їхні навички та досвід, і подавати заявки на роботу, яка відповідає їхній кваліфікації.

Платформа дистанційного пошуку роботи була реалізована за допомогою веб-інструментів та забезпечує визначений набір функціональних можливостей, таких як: сторінка для реєстрації та авторизації з вибором ролі роботодавця або кандидата; сторінка роботодавця, що дає змогу публікувати наявні вакансії компанії, писати вимоги щодо кандидатів та має можливість зв'язатися з ними; сторінка кандидата, де він має змогу створювати власне портфоліо, переглянути всі доступні вакансії, сортувати за категоріями, надсилати відгук на зацікавлену вакансію та має можливість зв'язатися з роботодавцем.

Під час виконання дипломного проектування були проаналізовані дослідження та публікації щодо користі дистанційної роботи та платформ, завдяки яким можна її знайти. Також були досліджені аналоги створеного продукту та виділені функції, які можна використовувати у власній розробці. Була проведена деталізація проєкту методом SMART, створено WBS-діаграму, OBS-структуру та календарний графік у вигляді діаграми Ганта. Також, аби

Щоб зменшити негативний вплив ризиків на проєкт, було проведено планування заходів по їхньому усуненню.

Отже, розроблений web-додаток є корисним інструментом як для роботодавців, так і для шукачів роботи, забезпечуючи зручний спосіб пошуку та заповнення вакансій. Він має надійні заходи безпеки для захисту даних користувачів і запобігання шахрайству. Крім того, він має ефективну систему перевірки вакансій і кандидатів, щоб гарантувати, що на сайті розміщуються лише легальні вакансії та кваліфіковані кандидати.

Результати роботи були оприлюднені на науково-технічній конференції ІМА-2023. Тези доповіді розміщені у додатку В.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The International Journal of Human Resource Management [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tandfonline.com/journals/rijh20> (Дата звернення: 17.04.23)
2. Journal of Business and Psychology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.springer.com/journal/10869> (Дата звернення: 17.04.23)
3. Online Job Search Application [Електронний ресурс] – Режим доступу до ресурсу: https://www.academia.edu/48863918/Online_Job_Search_Application (Дата звернення: 18.04.23)
4. Контроль на відстані: плюси та мінуси дистанційної роботи [Електронний ресурс] – Режим доступу до ресурсу: <https://thepage.ua/ua/economy/kontrol-na-vidstani-plyusi-ta-minusi-distancijnoyi-roboti> (Дата звернення: 19.04.23)
5. Achievements and challenges for work/life balance strategies in Australian organizations [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tandfonline.com/doi/full/10.1080/0958519042000295966> (Дата звернення: 19.04.23)
6. Канали пошуку роботи по всій Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://ukrainian.city/roboata/kanaly-poshuku> (Дата звернення: 20.04.23)
7. Кращі сайти і телеграм канали для пошуку роботи в Україні 2023 [Електронний ресурс] – Режим доступу до ресурсу: <https://bizmag.com.ua/krashchi-saity-i-telehram-kanaly-dlia-poshuku-roboty-v-ukraini/> (Дата звернення: 20.04.23)
8. Що таке LinkedIn і чи потрібен вам профіль у цій соцмережі? [Електронний ресурс] – Режим доступу до ресурсу: <https://happymonday.ua/shho-take-linkedin-i-navishho-vin-potriben> (Дата звернення: 20.04.23)

9. Розробка web-додатків, мобільних додатків та порталів [Електронний ресурс] – Режим доступу до ресурсу: <https://ittel.com.ua/informacijnitexnologiyi/rozrobka-mobilnih-dodatkiw/> (Дата звернення: 24.04.23)
10. Web-додаток "DOU" [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/> (Дата звернення: 25.04.23)
11. Web-додаток "roboata.ua" [Електронний ресурс] – Режим доступу до ресурсу: <https://rabota.ua/ua> (Дата звернення: 27.04.23)
12. Web-додаток "djinni" [Електронний ресурс] – Режим доступу до ресурсу: <https://djinni.co/> (Дата звернення: 28.04.23)
13. Огляд робочих структур проекту [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/uk-ua/dynamics365/project-operations/prod-pma/work-breakdown-structures> (Дата звернення: 08.05.23)
14. Структура web-додатку [Електронний ресурс] – Доступ до ресурсу: https://stud.com.ua/97612/informatika/struktura_dodatku (Дата звернення: 10.05.23)
15. Загладько Є. Що таке діаграма Ганта та як нею правильно користуватися? [Електронний ресурс] / Євгеній Загладько – Режим доступу до ресурсу: <https://nachasi.com/creative/2020/09/03/gantt-chart/> (Дата звернення: 15.05.23)

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку

Web-додатку з пошуку дистанційної роботи

Суми 2023

1. Призначення й мета web-додатку

1.1 Призначення web-додатку

Web-додаток призначений для пошуку дистанційної роботи.

1.2 Мета створення web-додатку

Метою проєкту є створення web-додатку для пошуку дистанційної роботи, який дозволить оптимізувати процес пошуку роботи по категоріям, залежно від спеціальності кандидата. Web-додаток дасть можливість швидко знайти необхідну вакансію не виходячи з дому.

1.3 Цільова аудиторія

Web-додаток є корисним інструментом як для роботодавців, так і для шукачів роботи, забезпечуючи зручний спосіб пошуку та заповнення вакансій

2 Вимоги до проєкту

2.1 Вимоги до проєкту в цілому

2.1.1 Вимоги до структури й функціонування

Web-додаток повинен бути стійким до витоків інформації, злому та інших вторинних ефектів. Web-додаток повинен працювати на різних платформах, браузерях та пристроях. Також він повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2 Вимоги до персоналу

Адміністратор web-додатку не повинен мати особливих технічних навичок для роботи з web-додатком та його підтримкою. Головне – це базове знання користування персональним комп'ютером задля додавання та видалення даних в панелі адміністратора.

2.1.3 Вимоги до збереження інформації

Уся інформація web-додатку буде зберігатися у базі даних, реалізованій засобами системи управління базами даних MSSQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Незареєстровані користувачі зможуть лише переглядати наявні вакансії. Для того, щоб відгукнутися на зацікавлену вакансію користувачу необхідно пройти звичайну реєстрацію. Для роботодавця, що хоче опублікувати нові вакансії компанії необхідно пройти реєстрацію від імені організації.

Адміністратор має необмежений доступ до даних з правами перегляду, додавання та видалення даних. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Роботодавець має право додавати нові вакансії та спілкуватися в чаті з шукачем роботи.

Людина, що шукає роботу, має право переглядати наявні вакансії, відкликатися на них та спілкуватися з роботодавцем у чаті.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель.

Перелік сторінок web-додатку наступний:

- «Головна» сторінка містить навігаційне меню, яке закріплене на кожній сторінці web-додатку та перелік нещодавно доданих вакансій;
- на сторінці «Вхід» розміщена форма входу та авторизації користувача;
- на сторінці «Мій акаунт» у шукача роботи є форма з власними даними, та вкладка з вподобаними вакансіями; роботодавець має форму з інформацією про компанію;
- сторінка «Повідомлення» містить вхідні повідомлень між роботодавцем та шукачем роботи;

2.2.2 Навігаційне меню

Для зручної навігації повинно бути створене меню, що забезпечить швидке переміщення користувача по всім доступним сторінкам web-сайту. Меню буде складатися з: Головна сторінка (з переліком вакансій), Вхід/Авторизація та Особистий кабінет користувача. Меню має бути закріплене і розташовуватися у шапці на кожній сторінці.

2.2.3 Наповнення web-додатку

Наповнення web-додатку контентом буде відбуватися через акаунт роботодавців компанії. Усі додані вакансії мають міститися у базі даних.

2.2.4 Дизайн web-додатку

Дизайн web-додатку має бути з простим та лаконічним інтерфейсом: Головна сторінка повинна містити всі необхідні функції для пошуку роботи, такі як фільтри, категорії і пошукову строку.

Дизайн буде виконаний в темно сірих тонах для створення простого та елегантного вигляду. Колірна схема зображено на рисунку А.1.

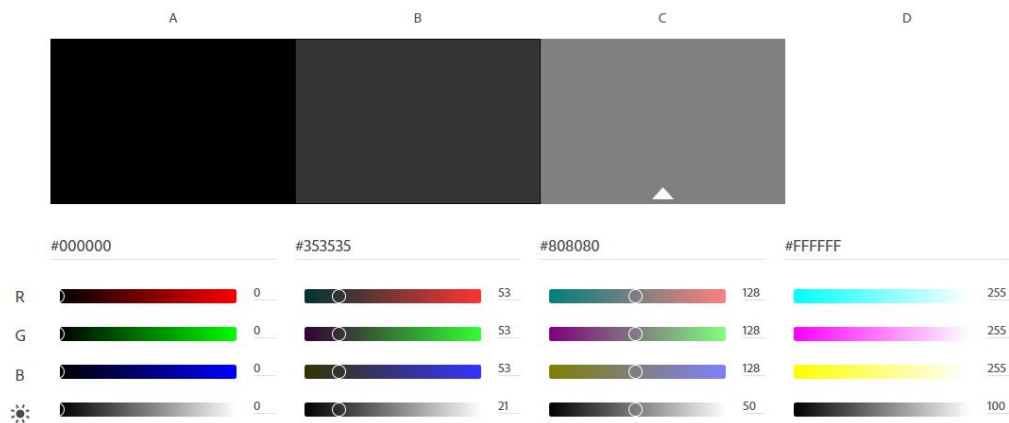


Рисунок А.2 – Колірна схема web-додатку

Макет головної сторінки майбутнього web-додатку зображено на рисунку А.2



Рисунок А.2 – Макет головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.



Рисунок А.2 – Система навігації

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Пошук вакансій	Шукач роботи
UN-02	Фільтрування вакансій	Шукач роботи
UN-03	Перегляд вакансій	Шукач роботи

UN-04	Перегляд опису вакансії	Шукач роботи
UN-05	Додавання даних	Роботодавець
UN-06	Редагування даних	Роботодавець
UN-07	Видалення даних	Роботодавець
UN-08	Листування	Роботодавець, шукач роботи

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувачів та персоналу закладу було визначено наступні вимоги:

- модуль реєстрації/авторизації, обираючи ролі роботодавця або шукача роботи;
- пошук наявних вакансій;
- сортування публікацій за категоріями;
- перегляд опису наявних вакансій;
- чат між роботодавцем та шукачем роботи;
- публікування нових вакансій компанії, написання вимог щодо шукачів роботи;
- редагування та видалення наявних вакансій компанії;

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Модуль авторизації/реєстрації	М	Надає можливість роботодавцям та шукачам

			роботи авторизуватися або зареєструватися
SR-02	База даних з вакансіями	M	Відповідає за заповнення web-додатку контентом
SR-03	База з особистими даними користувачів	M	Відповідає за збереження даних зареєстрованих користувачів
SR-04	Модуль пошуку вакансій за фільтрами	M	Відповідає за пошук вакансій на сайті за такими фільтрами як: спеціальність, досвід, тип зайнятості, заробітна плата
SR-05	Модуль листування	S	Надає можливість зв'язатися шукачу роботи та роботодавцю за допомогою чату

Умовні позначення в таблиці А.2:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проєкту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- Figma – для створення прототипу web-додатку

- HTML 5 – мова розмітки web-сторінок
- CSS 3 – каскадні таблиці стилів для опису зовнішнього вигляду
- C# – мова програмування
- MSSQL – база даних для збереження інформації
- BootStrap – набір шаблонів CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу
- ASP.NET – платформа для створення web-додатку.

2.4.2 Вимоги до лінгвістичного забезпечення

Весь текст у web-додатку має бути написаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Для забезпечення стабільної роботи web-додатку операційна система має бути:

- Windows XP і вище
- Mac OS X 10.0.4 і вище
- Linux.

Для web-браузерів необхідно мати такі версії:

- Chrome 2 і вище
- Opera 9.5 і вище
- Internet Explorer 7.0 і вище
- Firefox 3.5 і вище
- Safari 3.2.1 і вище.

3 Склад і зміст робіт зі створення web-додатку

Детальний опис етапів створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web- додатку

№	Склад і зміст робіт	Строк розробки
1	Розробка шаблону web-додатку	2 дні
2	Задання верстки сторінок web-додатку	10 днів
3	Розробка модулів web-додатку	10 днів
4	Розробка бази даних	5 днів
5	Beta-тестування	5 днів
6	Alpha-тестування	5 днів
7	Розміщення на хостингу	1 день
8	Перевірка працездатності	2 дні
9	Написання супровідної документації	3 дні
10	Реліз web-додатку	1 день
	Загальна тривалість робіт	44 дні

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Для того, щоб web-додатком могли користуватися роботодавці та люди, що в пошуку роботи, необхідно розмістити його у мережі Інтернет, тому для цього треба придбати доменне ім'я та місце на хостингу. На хостинг переноситься web-додаток і наповнення бази даних з подальшою їх доробкою. Для коректного переносу web-додатку на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ.

ДОДАТОК Б

Продуктом дипломного проекту є web-додаток для пошуку дистанційної роботи. Проєкт має на меті створення web-додатку для пошуку дистанційної роботи, що дозволить кандидатам ефективно оптимізувати процес пошуку відповідно до їхньої спеціалізації та категорії.

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити web-додаток для пошуку дистанційної роботи, що має оптимізувати процесу пошуку роботи відповідно до спеціалізації та категорії користувача.
Measurable (вимірювана)	Успіх проєкту буде вимірюватися збільшенням щонайменше на 20% кількості успішних працевлаштованих на віддалену роботу за допомогою додатку протягом перших шести місяців після запуску додатку.
Achievable (досяжна)	Реалізація продукту здійснюється за допомогою CMS WordPress.
Relevant (реалістична)	Для реалізації наявні технічні навички в роботі з CMS WordPress.
Time-framed (обмежена у часі)	Проєкт повинен бути завершений згідно з календарним планом.

WBS-діаграма – це ієрархічне візуальне представлення, яке демонструє обсяг, завдання та результати проєкту. Розбиваючи проєкт на керовані компоненти, вона дозволяє покращити планування, організацію та розуміння структури проєкту.

Починаючи з найвищого рівня, який представляє весь проєкт, WBS-діаграма згодом ділить його на підрівні, кожен з яких представляє менші завдання або компоненти. Така ієрархічна структура дає чітке і систематичне уявлення про те, як розбита робота над проєктом. WBS-діаграма представлена на рисунку Б.1.

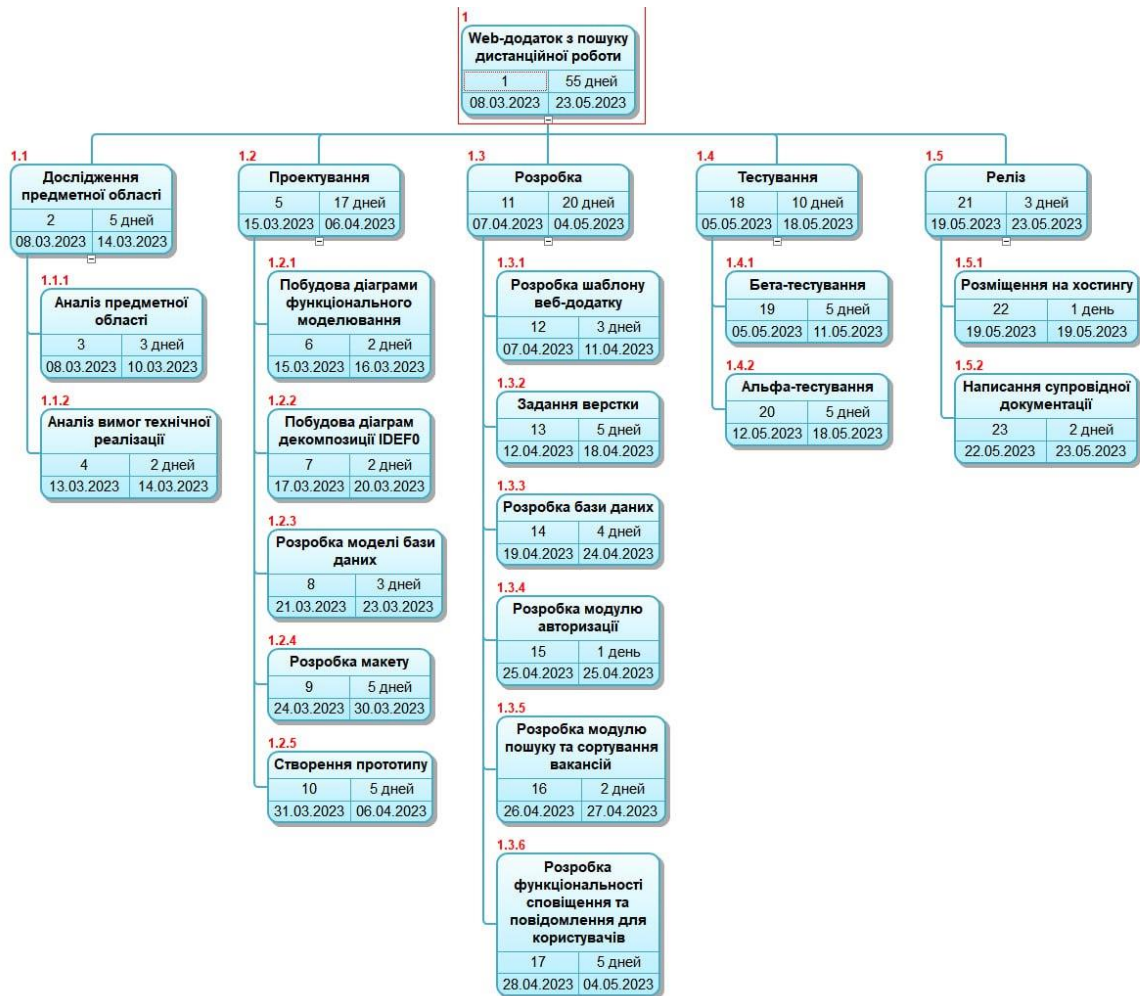


Рисунок Б.1 – WBS-структура робіт проєкту

Організаційна структура (OBS) – це ієрархічне зображення, яке демонструє організаційну структуру проєкту. Організаційна структура для планування проєкту зображена на рисунку Б.2. Вона виділяє осіб або відділи, які беруть участь у проєкті, та ілюструє їхні стосунки і лінії підпорядкування. Забезпечуючи чітке розуміння організаційної структури проєкту, OBS допомагає визначити ролі, обов'язки та канали комунікації всередині проєктної команди. Список виконавців проєкту представлені в таблиці Б.2.

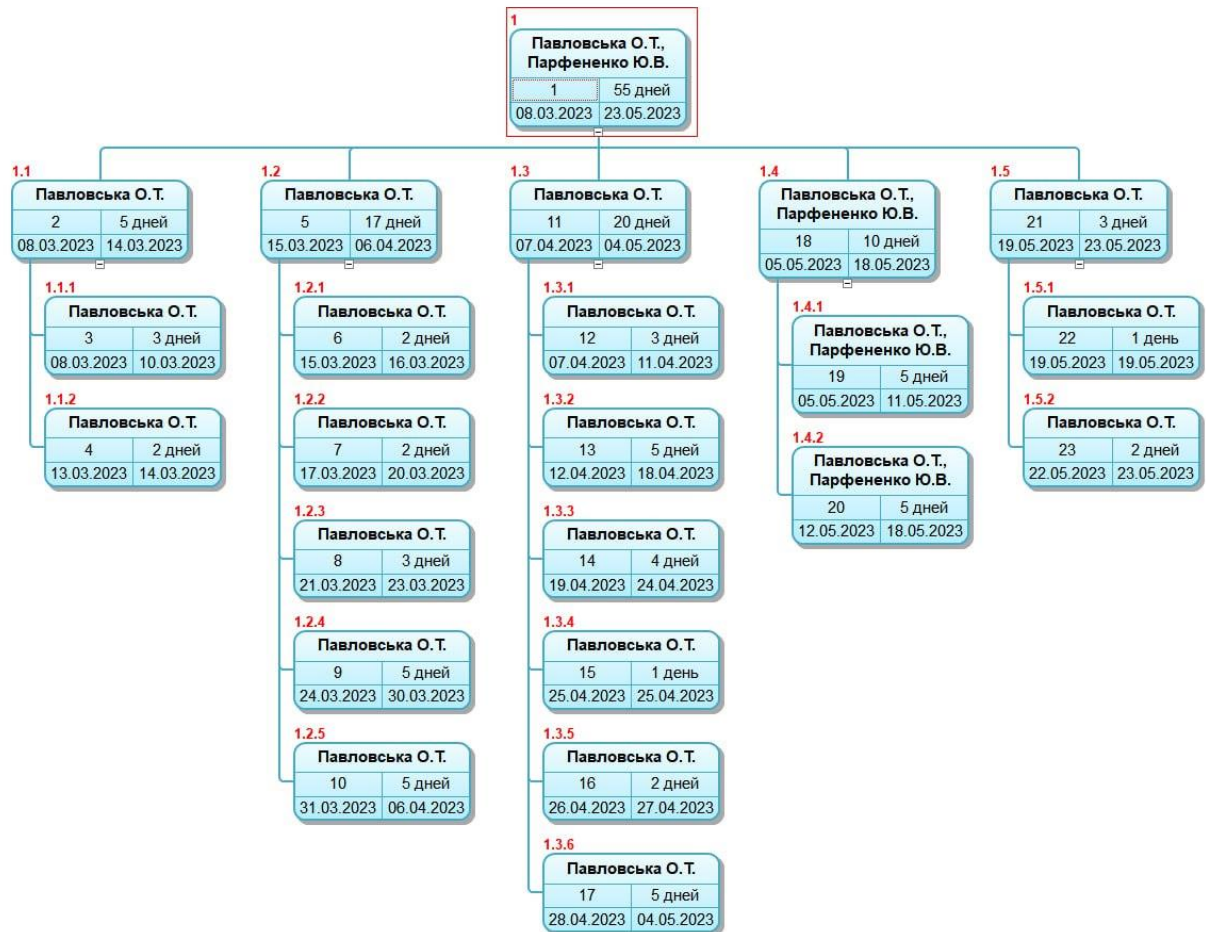


Рисунок Б.2 – OBS-структура проекту

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Павловська О.Т.	Займається front-end та back-end розробкою.
Проектувальник	Павловська О.Т.	Займається проектуванням бази даних та розробкою структури web-додатку.
Тестувальник	Павловська О.Т.	Займається тестуванням функціоналу web-додатку.
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту.
Менеджер проекту	Павловська О.Т.	Контролює вчасне виконання завдань, розподіл ресурсів та завдань між учасниками. Виконує процес збору та аналізу даних.

Наступним етапом є створення календарного графіка. Діаграми Ганта є важливою складовою планування проекту. Цей графік використовується для візуалізації послідовності та тривалості виконання робіт у проекті.

Процес створення календарного графіка розпочинається з ідентифікації всіх завдань та активностей, які необхідно виконати в проекті. Ці завдання потім розбиваються на більш дрібні етапи або підзавдання. Далі визначаються послідовність виконання цих етапів та встановлюються залежності між ними.

Кожне підзавдання визначається тривалістю, яка може бути виміряна у днях, тижнях або інших одиницях часу. На основі цих тривалостей та залежностей між завданнями створюється календарний графік, який показує розклад виконання робіт протягом усього проекту. Діаграма Ганта та список робіт зображені на рисунку Б.3-Б.4.

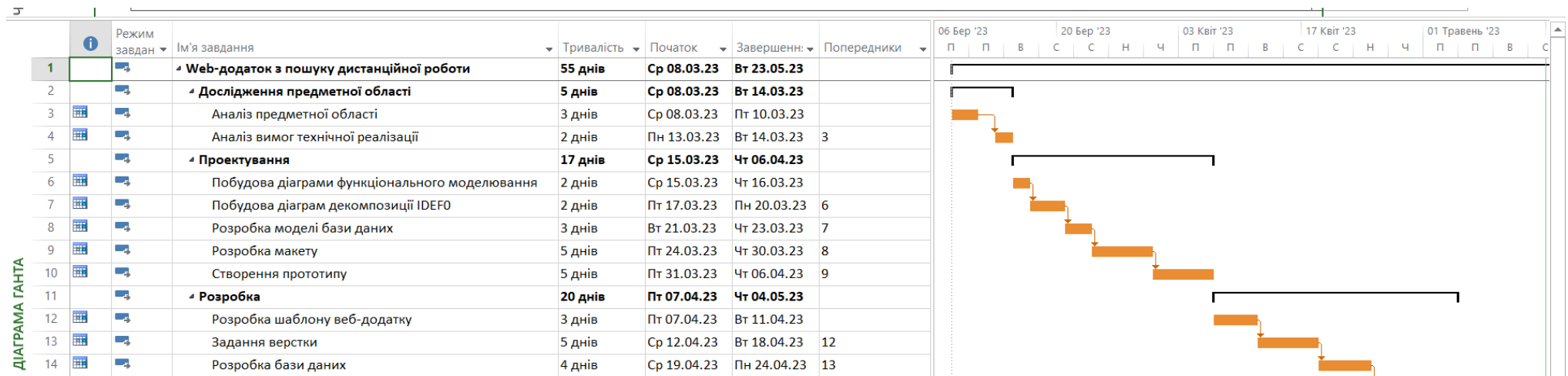


Рисунок Б.3 – Діаграма Ганта

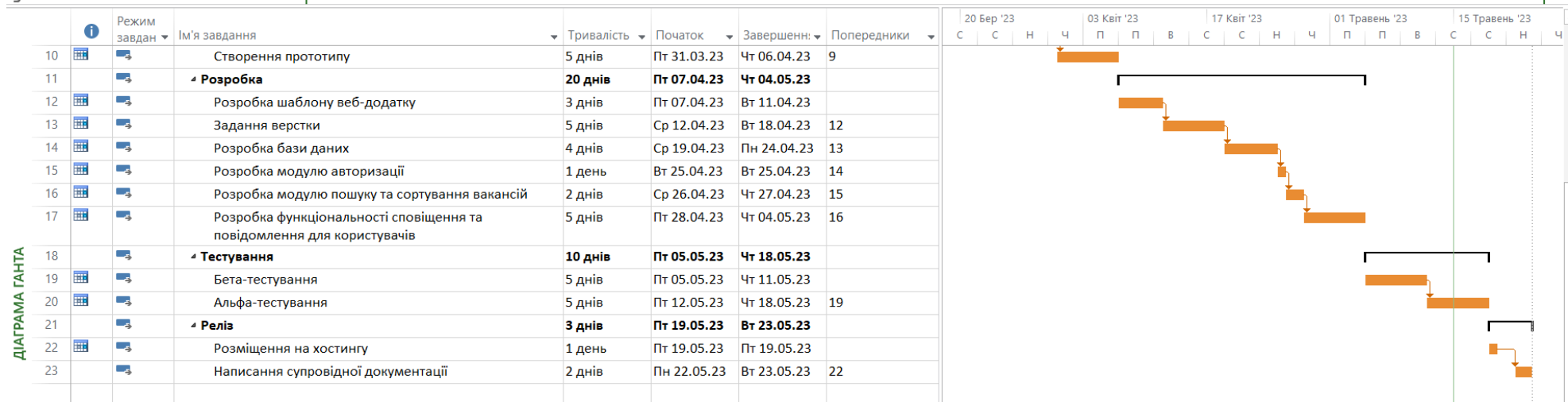


Рисунок Б.3 – Продовження діаграми Ганта

Управління ризиками проекту є ключовим процесом, спрямованим на виявлення, аналіз, оцінку та керування потенційними ризиками, які можуть вплинути на успішність проекту. Його головна мета полягає у зменшенні негативного впливу ризиків та максимізації використання можливостей для досягнення поставлених цілей проекту. Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту представлена в таблиці Б.3.

Таблиця Б.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

На основі оцінки будемо матрицю ймовірності виникнення ризиків та впливу ризику, що представлена в таблиці Б.4.

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

Таблиця Б.4. – Матриця ймовірності виникнення ризиків та впливу ризику

Ймовірність ризику	Вплив загрози		
	Малий	Середній	Високий
	1	2	3
3			RS_3
2		RS_1, RS_2	RS_4, RS_6, RS_8
1		RS_5	RS_7

У таблиці Б.5 наведено класифікацію ризиків за рівнем, враховуючи отримане значення індексу у таблиці Б.4.

Таблиця Б.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять
1	Прийнятні	$1 \leq R \leq 2$	5
2	Виправдані	$3 < R \leq 5$	1, 2, 7
3	Недопустимі	$6 < R \leq 9$	3, 4, 6, 8

Таблиця Б.6 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив	Ранг ризику	План А	Тип стратегії	План Б
RS_1	Відкритий	Нечітке завдання на розробку	Середня	Середній	4	Провести детальні консультації із замовником з метою обговорення всіх видів вимог	Попередження	Якщо виявлено відхилення деяких характеристик продукту від заявлених вимог, необхідно точно визначити недоліки та здійснити відповідні коригування для виправлення ситуації
RS_2	Відкритий	Зміна вимог	Середня	Середній	4	На початкових етапах необхідно досягти згоди щодо всіх питань з метою зменшення потреби у внесенні змін під час розробки	Пом'якшення	Робити переоцінку проєкту кожного разу, коли змінюються вимоги

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив	Ранг ризику	План А	Тип стратегії	План Б
RS_3	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Здійснити оцінку актуальності найважливіших процесів та завдань. Визначити пріоритети для виконання робіт. Ретельно дотримуватися календарного плану	Пом'якшення	Обговорити можливі варіанти внесення змін до термінів реалізації замовником
RS_4	Відкритий	Низька кваліфікація розробників	Середня	Висока	6	Підвищити кваліфікацію співробітників за допомогою курсів	Пом'якшення	Знайти більш кваліфікованих співробітників, що допоможуть в розробці

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив	Ранг ризику	План А	Тип стратегії	План Б
RS_5	Відкритий	Неправильна оцінка в масштабі проекту	Низька	Середній	2	Провести детальний аналіз масштабів проекту на основі додаткових джерел інформації	Пом'якшення	Перебудова стратегії реалізації проекту
RS_6	Відкритий	Відсутність електроенергії	Середня	Високий	6	Знайти приміщення з наявною електроенергією	Ухилення	Купити генератори
RS_7	Відкритий	Відсутність резервних копій даних	Низька	Високий	3	Встановити автоматичне збереження даних	Попередження	Здійснювати створення резервних копій даних після завершення кожного етапу
RS_8	Відкритий	Збої в роботі програмного забезпечення	Середня	Високий	6	Забезпечити наявність резервних програмних засобів	Попередження	Виконати заміну програмного забезпечення

ДОДАТОК В

СЕКЦІЯ 2: Інформаційні технології проєктування

ІМА :: 2023

Web-додаток з пошуку дистанційної роботи

Павловська О.Т., студент ІТ-91, Парфененко Ю.В., доцент
Сумський державний університет, м. Суми, Україна

Зараз, коли все більше компаній переходять на роботу з дому та дистанційний формат, пошук роботи в Інтернеті стає все більш популярним. Актуальність розроблення додатку з пошуку дистанційної роботи полягає в тому, що він надає платформу для шукачів роботи та роботодавців, які можуть спілкуватися у віддалений та зручний спосіб.

Мета цього web-додатку – полегшити процес пошуку роботи як для роботодавців, так і для шукачів. Роботодавці можуть розміщувати свої вакансії та шукати кваліфікованих кандидатів, а шукачі роботи можуть створити профіль, який демонструє їхні навички та досвід, і подавати заявки на роботу, яка відповідає їхній кваліфікації.

Платформа дистанційного пошуку роботи була реалізована за допомогою web-інструментів та забезпечує визначений набір функціональних можливостей, таких як: сторінка для реєстрації та авторизації з вибором ролі рекрутера або кандидата; сторінка рекрутера, що дає змогу публікувати наявні вакансії компанії, писати вимоги щодо кандидатів та має можливість зв'язатися з ними; сторінка кандидата, де він має змогу створювати власне портфоліо, переглянути всі доступні вакансії, сортувати за категоріями, надсилати відгук на зацікавлену вакансію та має можливість зв'язатися з рекрутером.

Дизайн платформи є адаптивним для його перегляду з більшості девайсів та браузерів. Для реалізації проєкту було використано такі інструменти, як HTML, CSS, JS, React JS, MongoDB. Категорії користувачів web-додатку можуть бути такими: роботодавці, які шукають кандидатів на свої вакансії; особи, хто шукає можливості працевлаштування, адміністратори: ті, хто керує та перевіряє вакансії.

Отже, розроблений web-додаток є корисним інструментом як для роботодавців, так і для шукачів роботи, забезпечуючи зручний спосіб пошуку та заповнення вакансій. Він має надійні заходи безпеки для захисту даних користувачів і запобігання шахрайству. Крім того, він має ефективну систему перевірки вакансій і кандидатів, щоб гарантувати, що на сайті розміщуються лише легальні вакансії та кваліфіковані кандидати.

ДОДАТОК Г

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ

Layout.cshtml

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - Job Search</title>
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"
/>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
integrity="sha512-
iBBXm8fW90+nuLcSKlbrPcLa0OT92xO1BIsZ+ywDWZCvqsWgccV3gFoRBv0z+8dLJgyAHIhR35VZc2oM/gI
lw==" crossorigin="anonymous" />
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
  <link href="~/Content/prodList.css" rel="stylesheet" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.3/font/bootstrap-icons.css" rel="stylesheet" />
</head>
<body>
  <div class="col-md-pull-2">
    <header>
      <nav id="nav" class="navbar fixed-top navbar-expand-lg navbar-light bg-inverse-info"
style="background-color: #353535; border: 2px solid transparent; border-radius: 0px; padding: 9px ">
        <div class="container-fluid" style="">
          @Html.ActionLink("Job Search", "Index", "Home", new { area = "" }, new { @class = "navbar-brand",
id = "hompng" })
          <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarCollapse" style="text-align:center">
            <div class="navbar-nav">
              @using (Html.BeginForm("Index", "Home", FormMethod.Get))
              {
                <div class="col input-group" style="margin:20px 0 0 245px">
                  <div class="form-outline" style="">
                    <select id="searchBy" name="searchBy" class="form-select">
                      <option value="0" selected="selected">Пошук по:</option>
                      <option value="1">
                        Назва вакансії
                      </option>
                      <option value="2">
                        Назва компанії
                      </option>
                    </select>
                  </div>
                  <div class="form-outline">
                    <input type="text" class="form-control search" style="width:360px" id="emailBy"
name="emailBy" placeholder="пошук..." onfocus="this.placeholder = "" onblur="this.placeholder = 'search'" />
                    <label class="form-label" for="search"></label>
                  </div>
                </div>
              }
            </div>
          </div>
        </div>
      </nav>
    </header>
  </div>

```

```

        <button style="" type="submit" class="btn btn-primary">
            <i class="fas fa-search"></i>
        </button>
    </div>
</div>
}
</div>
<div class="navbar-nav ms-auto">
    @Html.Partial("_LoginPartial")
</div>
</div>
</div>
</nav>

</header>
</div>
<div class="container body-content">
    @RenderBody()
    <hr />
</div>
<!-- Footer -->
<footer class="text-center text-white"
    style="background-color: #353535; opacity: 1">
    <!-- Grid container -->
    <div class="container p-4 pb-0">
        <!-- Section: Social media -->
        <section class="mb-4 text-center">
            <p>Приеднуйся до нас!</p>
            <!-- Facebook -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="http://www.facebook.com"
                target="_blank"
                role="button">
                <i class="fab fa-facebook-f"></i>
            </a>
            <!-- Twitter -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="#"
                role="button">
                <i class="fab fa-twitter"></i>
            </a>
            <!-- Google -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="mailto:aaa@gmail.com" target="_blank"
                role="button">
                <i class="fab fa-google"></i>
            </a>
            <!-- Instagram -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="" target="_blank"
                role="button">
                <i class="fab fa-instagram"></i>
            </a>
            <!-- LinkedIn -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="https://wa.me/" target="_blank"
                role="button">
                <i class="fab fa-whatsapp"></i>
            </a>
            <!-- Github -->
            <a class="btn btn-outline-light btn-floating m-1"
                href="viber://public?id=" target="_blank"

```

```

        role="button">
        <i class="fab fa-viber"></i>
    </a>
</section>
</div>
<div class="text-center p-3"
    style="background-color: rgba(0, 0, 0, 0.2)">
    <a class="text-white" style="text-decoration:none" href="#">
        &copy; @DateTime.Now.Year - Job Search Web-Service
    </a>
</div>
</footer>
<!-- End Footer -->
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>

```

Index.cshtml

```

@model IEnumerable<JobSearch.BLL.Dto.VacancyDto>
@{
    ViewBag.Title = "Головна";
    var role = ViewBag.Role;
}
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" />
<div class="row" style="font-family: 'Times New Roman'; font-weight: bold; text-align: left; margin-top: 108px;
color: #696969">
    @if (ViewBag.UserId != 0)
    {
        <hr />
        <div class="col" style="margin-left: 40px">
            <h6 style="font-weight: bold; margin-bottom: 15px">
                @ViewBag.WelcomeStr
            </h6>
        </div>
        if (role == "RegisterUser")
        {
            <div class="col" style="text-align: right; padding-right: 70px">
                @Html.ActionLink("Моя анкета", "UserProfile", "Home", null, new { id = "profile" })
            </div>
        }
        else
        {
            <div class="col" style="text-align:right; padding-right: 70px">
                @Html.ActionLink("Розмістити вакансію", "Vacancy", "Create", new { emplEmail = ViewBag.MyEmail
}, new { id = "profile" })
            </div>
        }
        <hr width: 5px/>
    }
</div>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script>
    var emailBy = document.getElementById("emailBy");
    var e = document.getElementById("searchBy");
    var value = e.value;
    var searchBy = e.options[e.selectedIndex].text;
    var search = { srchBy: searchBy, mailBy: searchBy };
    $.ajax({
        contentType: 'application/json; charset=utf-8',

```

```

        dataType: 'json',
        type: 'GET',
        url: '/Index/Index',
        data: search,
        success: function (result) {
            alert(result);
        }
    });
</script>
@if (ViewBag.IsRespond == true)
{
    <div style="" class="alert alert-success alert-dismissible">
        <a href="~/Home/CloseAlerts" class="close" data-dismiss="alert" style="text-decoration: none;text-align:right;
font-size: large; color: #82822f"
        aria-label="close">&times;</a>
        <h4 style="text-align:center;color:brown;font-family:'Palatino Linotype';font-weight:bold">
            @ViewBag.ModalMsg
        </h4>
    </div>
}
@using (Html.BeginForm())
{
    <div class="row container">
        <div class="row" style="width: 100%; margin: 7px">
            <div class="col">
                @Html.ActionLink("Yci kareropii", "Index", "Home", new { categoryId = "null" }, new { id = "allCateg"
            })
        </div>
    </div>
    <div class="row clearfix" style="width: 100%; margin: auto">
        <div style="border-radius:5px" class="col-md-3">
            <div class="card single_post" style="padding:12px">
                @foreach (var category in ViewBag.Categories as List<JobSearch.BLL.Dto.CategoryDto>)
                {
                    var c = category.Id;
                    <p>
                        @Html.ActionLink(category.Name, "Index", "Home", new { categoryId = c }, new { style = "text-
decoration:none" })
                    </p>
                }
            </div>
        </div>
        <div class="col-md-9" style="margin-bottom: 15px">
            @foreach (var vacancy in Model)
            {
                string Path = "~/Images/Vacancies/" + vacancy.Photo;
                <div class="card single_post" style="padding:15px">
                    <div class="body" style="overflow:auto">
                        <div style="text-align:center">
                            
                        </div>
                        <h3>
                            <a style="text-decoration: none; font-weight: bold; color: black; font-family: 'Palatino Linotype'
"
                                href="/Home/FullVacancy?id=@vacancy.Id">
                                @vacancy.JobTitle
                            </a>
                        </h3>
                        @{
                            if (vacancy.JobDescription.Count() < 222)
                            {

```



```

        @Html.DropDownListFor(p => p.Role, ViewBag.Role as SelectList, "Зареєструватись як:",
new { @class = "form-control", id = "select-placeholder", required = "required" })
        @Html.ValidationMessageFor(model => model.Role, "", new { @class = "text-danger", style
= "font-size:small;font-weight:bold" })
        </div>
        <div class="form-group">
            @Html.TextBoxFor(p => p.City, new { placeholder = "Місто:", @class = "form-control", id =
"select-placeholder", required = "required" })
            @Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger", style
= "font-size:small;font-weight:bold" })
        </div>
        <div class="form-group">
            @Html.PasswordFor(p => p.Password, new { placeholder = "Пароль:", id = "password-field",
@class = "form-control", style = "color:white", required = "required", pattern = "^(?=.*[a-z])(?=.*[A-
Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{6,}$" })
            @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger",
style = "font-size:small;font-weight:bold" })
        </div>
        <div class="form-group">
            @Html.PasswordFor(p => p.ConfirmPassword, new { placeholder = "Підтвердження
паролю:", @class = "form-control", style = "color:white", required = "required" })
            @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-
danger", style = "font-size:small;font-weight:bold" })
        </div>
        <div class="form-group">
            <button type="submit" class="form-control btn btn-primary submit px-
3">Зареєструватися</button>
        </div>
    </form>

    </div>
    </div>
    </div>
    </div>
    <div class="row" style="text-align:center">
        <div class="col">
            @Html.ActionLink("Повернутись назад", "Index", "Home", new { area = "" }, new { @class = "navbar-
brand", id = "ret" })
        </div>
    </div>
</section>
<script src="@Url.Content("~/Content/AccountForm/js/jquery.min.js")"></script>
<script src="@Url.Content("~/Content/AccountForm/js/popper.js")"></script>
<script src="@Url.Content("~/Content/AccountForm/js/bootstrap.min.js")"></script>
<script src="@Url.Content("~/Content/AccountForm/js/main.js")"></script>
}
</body>
</html>

```

UserProfile.cshtml

```

@model JobSearch.PL.Models.ApplicationUser
@{
    ViewBag.Title = "Анкета користувача";
    var user = Model;
    string photoPath = "~/Images/Users/" + user.Photo;
}
<div class="container-xl px-4 mt-4">
    <!-- Account page navigation-->
    <nav class="nav nav-borders">

```

```

    <a class="nav-link active ms-0" href="https://www.bootdey.com/snippets/view/bs5-edit-profile-account-
details" target="_blank">Profile</a>
    <a class="nav-link" href="https://www.bootdey.com/snippets/view/bs5-profile-billing-page"
target="_blank">Billing</a>
    <a class="nav-link" href="https://www.bootdey.com/snippets/view/bs5-profile-security-page"
target="_blank">Security</a>
    <a class="nav-link" href="https://www.bootdey.com/snippets/view/bs5-edit-notifications-page"
target="_blank">Notifications</a>
</nav>
@if (ViewBag.IsRespond == true)
{
    <div style="margin-top:35px" class="alert alert-success alert-dismissible">
        <a href="~/Home/CloseAlerts" class="close" data-dismiss="alert" style="text-decoration: none;text-
align:right; font-size: large; color: #82822f"
        aria-label="close">&times;</a>
        <h4 style="text-align:center;color:brown;font-family:'Palatino Linotype';font-weight:bold">
            Дані успішно оновлені!
        </h4>
    </div>
}
<br />
<hr class="mt-0 mb-4">
<div class="row">
    <div class="col-xl-4">
        <div class="card mb-4 mb-xl-0">
            <div y class="card-header">Резюме</div>
            <div class="card-body text-center">
                @if (user.Photo == null)
                {
                    
                }
                else
                {
                    
                }
                <div class="small font-italic text-muted mb-4">JPG or PNG no larger than 5 MB</div>
                <div class="row" style="text-align:center">
                    <div class="col">
                        <input id="fileinput" type="file" accept=".jpg, .jpeg, .png, .icon, .ico, .gif">
                        <br />
                        <input class="btn btn-primary" type="button" style="margin-top:6px" id="btnUpload"
value="Завантажте своє фото" />
                    </div>
                    @Html.ValidationMessageFor(model => model.Photo, "", new { @class = "text-danger" })
                </div>
            </div>
        </div>
    </div>
    <div class="col-xl-8">
        <!-- Account details card-->
        <div class="card mb-4">
            <div class="card-header">Деталі облікового запису</div>
            <div class="card-body">
                @using (Html.BeginForm())
                {
                    <!-- Form Row-->
                    <div class="row gx-3 mb-3">
                        <!-- Form Group (first name)-->
                        <div class="col-md-6">
                            <label class="small mb-1" for="inputFirstName">Прізвище</label>

```

```

        @Html.TextBoxFor(p => p.FirstName, new { @class = "form-control" })
    </div>
    <!-- Form Group (last name)-->
    <div class="col-md-6">
        <label class="small mb-1" for="inputLastName">Ім'я</label>
        @Html.TextBoxFor(p => p.LastName, new { @class = "form-control" })
    </div>
</div>
<!-- Form Row -->
<div class="row gx-3 mb-3">
    <!-- Form Group (experiance)-->
    <div class="col-md-6">
        <label class="small mb-1" for="inputOrgName">Досвід роботи</label>
        @Html.TextBoxFor(p => p.Experience, new { @class = "form-control" })
    </div>
    <!-- Form Group (location)-->
    <div class="col-md-6">
        <label class="small mb-1" for="inputLocation">Місто</label>
        @Html.TextBoxFor(p => p.City, new { @class = "form-control" })
    </div>
</div>
<!-- Form Group (email address)-->
<div class="mb-3">
    <label class="small mb-1" for="inputEmailAddress">Ел. пошта</label>
    @Html.TextBoxFor(p => p.Email, new { @class = "form-control" })
</div>
<!-- Form Row-->
<div class="row gx-3 mb-3">
    <!-- Form Group (phone number)-->
    <div class="col-md-6">
        <label class="small mb-1" for="inputPhone">Моб.</label>
        @Html.TextBoxFor(p => p.PhoneNumber, new { @class = "form-control" })
    </div>
    <!-- Form Group (birthday)-->
    <div class="col-md-6">
        <label class="small mb-1" for="inputBirthday">Дата народження</label>
        @Html.EditorFor(p => p.DateBirthday, new { htmlAttributes = new { @class = "form-control" }
    </div>
</div>
<!-- Form Group (skills) -->
<div class="mb-3">
    <label class="small mb-1" for="inputEmailAddress">Навички</label>
    @Html.TextAreaFor(p => p.Skills, new { @class = "form-control", rows = "12" })
</div>
<!-- Save changes button-->
<button class="btn btn-primary" type="submit">Зберегти</button>
    }
</div>
</div>
</div>
</div>
</div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $('#btnUpload').click(function () {
            // Checking whether FormData is available in browser
            if (window.FormData !== undefined) {
                var fileUpload = $("#fileinput").get(0);
                var files = fileUpload.files;
                // Create FormData object

```

```

var fileData = new FormData();
// Looping over all files and add it to FormData object
for (var i = 0; i < files.length; i++) {
    fileData.append(files[i].name, files[i]);
}
$.ajax({
    url: '/Home/PhotoUpload',
    type: "POST",
    contentType: false,
    processData: false,
    data: fileData,
    success: function (result) {
        alert(result);
    },
    error: function (err) {
        alert(err.statusText);
    }
});
} else {
    alert("FormData is not supported.");
}
});
});
</script>

```

Create Vacancy.cshtml

```

@model JobSearch.BLL.Dto.VacancyDto
@{
    ViewBag.Title = "Vacancy";
    var employer = ViewBag.Employer as JobSearch.PL.Models.ApplicationUser;
}
@using (Html.BeginForm())
{
    <div class="container">
        <div class="row clearfix" style="width:70%;margin:auto">
            <h1 style="margin-top:120px;text-align:center;font-family:'Palatino Linotype';color:black">
                Додати нову вакансію
            </h1>
            <div class="col">
                <div class="card single_post" style="padding:12px">
                    <div class="body">
                        <form class="row clearfix">
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => t.JobTitle, new { placeholder = "Назва ваканції:", @class = "form-control", required = "required" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.DropDownListFor(p => p.CategoryId, ViewBag.Categories as SelectList, "Категорія:", new { @class = "form-control", id = "select-placeholder" })
                                @Html.ValidationMessageFor(model => model.CategoryId, "", new { @class = "text-danger", style = "font-size:small;font-weight:bold" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.Label("Додайте фото ваканції:", htmlAttributes: new { id = "lbl", @class = "control-label col-md-6" })
                                <div class="col-md-10" style = "margin-bottom: 10px">
                                    <input style = "margin-bottom: 10px" id="fileinput" type="file" accept=".jpg, .jpeg, .png, .icon, .ico, .gif" required = "required">
                                    <br />
                                    <input type="button" id="btnUpload" value="Upload Photo" class="btn btn-success form-control-file" />
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        @Html.ValidationMessageFor(model => model.Photo, "", new { @class = "text-danger" })
    </div>
</div>
<div class="form-group" style = "margin-bottom: 10px">
    @Html.TextBoxFor(t => t.JobPayment, new { placeholder = "Оплата:", @class = "form-control"
})
    </div>
<div class="form-group" style = "margin-bottom: 10px">
    @Html.TextBoxFor(t => t.CompanyName, new { placeholder = "Назва компанії:", @class =
"form-control", required = "required" })
    </div>
<div class="form-group" style = "margin-bottom: 10px">
    @Html.TextBoxFor(t => t.Experience, new { placeholder = "Досвід роботи:", @class = "form-
control" })
    </div>
<div class="form-group" style = "margin-bottom: 10px">
    @Html.TextBoxFor(t => employer.FirstName, new { placeholder = "Роботодавець:", @class =
"form-control", @readonly = "true" })
    </div>
<div class="form-group" style = "margin-bottom: 10px">
    @Html.TextAreaFor(t => t.JobDescription, new { placeholder = "Повний опис вакансії:",
@class = "form-control", rows = "14", required = "required" })
    </div>
<div style="margin-top: 12px">
    <button type="submit" class="btn btn-block btn-primary">
        Додати
    </button>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
}
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $('#btnUpload').click(function () {

            // Checking whether FormData is available in browser
            if (window.FormData !== undefined) {

                var fileUpload = $("#fileinput").get(0);
                var files = fileUpload.files;

                // Create FormData object
                var fileData = new FormData();

                // Looping over all files and add it to FormData object
                for (var i = 0; i < files.length; i++) {
                    fileData.append(files[i].name, files[i]);
                }
                $.ajax({
                    url: '/Create/PhotoUpload',
                    type: "POST",
                    contentType: false,
                    processData: false,
                    data: fileData,
                    success: function (result) {
                        alert(result);
                    },
                },

```

```

        error: function (err) {
            alert(err.statusText);
        }
    });
} else {
    alert("FormData is not supported.");
}
});
});
</script>

```

Delete Vacancy.cshtml

```

@model JobSearch.BLL.Dto.VacancyDto
@{
    ViewBag.Title = "Vacancy [Delete]";
}
@using (Html.BeginForm())
{
    var vacancy = Model;
    @Html.AntiForgeryToken()
    <div class="container">
        <div class="row clearfix" style="width:70%;margin:auto">
            <h4 style="text-align:center;color:red; margin-top: 130px; margin-bottom: 20px">
                Ви впевнені, що хочете видалити цю вакансію?
            </h4>
            <div class="col">
                <div class="card single_post" style="padding:12px">
                    <div class="body" style="font-family:'Times New Roman';font-weight:bold;text-align:center">
                        <form class="row clearfix">
                            <div class="form-group">
                                @Html.HiddenFor(t => vacancy.Id)
                            </div>
                            <div class="form-group" style="font-weight: bold; font-family:'Palatino Linotype';text-align:center; font-size: x-large">
                                @Html.DisplayFor(t => vacancy.JobTitle)
                            </div>
                            <div class="form-group" style="text-align:center">
                                
                                </div>
                                <div style="margin-top:12px; text-align:center">
                                    <button type="submit" class="btn btn-block btn-danger">Видалити</button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="row" style="text-align:center; margin-top: 20px; margin-bottom: 100px">
        <div class="col">
            @Html.ActionLink("<-- Повернутися назад до вакансій", "Index", "Home", null,
                new { @class = "btn btn-default", style = "background-color:black;color:white" })
        </div>
    </div>
</div>

```

Edir Vacancy.cshtml

```

@model JobSearch.BLL.Dto.VacancyDto
@{
    ViewBag.Title = "Оновлення";
}

```



```

    var employer = ViewBag.Employer as JobSearch.PL.Models.ApplicationUser;
}
@using (Html.BeginForm())
{
    <div class="container">
        <div class="row clearfix" style="width:70%;margin: auto">
            <h1 style="margin-top: 120px; margin-bottom: 15px; text-align:center; font-family: 'Palatino Linotype'; color:
black">
                Редагувати вакансію
            </h1>
            <div class="col">
                <div class="card single_post" style="padding:12px">
                    <div class="body">
                        <form class="row clearfix">
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => t.JobTitle, new { placeholder = "Назва вакансії:", @class = "form-
control" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.DropDownListFor(p => p.CategoryId, ViewBag.Categories as SelectList, "Категорія:",
new { @class = "form-control", id = "select-placeholder" })
                                @Html.ValidationMessageFor(model => model.CategoryId, "", new { @class = "text-danger",
style = "font-size:small;font-weight:bold" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.Label("Додайте фото вакансії:", htmlAttributes: new { id = "lbl", @class = "control-
label col-md-6" })
                                <div class="col-md-10" style = "margin-bottom: 10px">
                                    <input style = "margin-bottom: 10px" id="fileinput" type="file" accept=".jpg, .jpeg, .png, .icon,
.ico, .gif">
                                    <br />
                                    <input type="button" id="btnUpload" value="Upload Photo" class="btn btn-success form-
control-file" />
                                    @Html.ValidationMessageFor(model => model.Photo, "", new { @class = "text-danger" })
                                </div>
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => t.JobPayment, new { placeholder = "Оплата:", @class = "form-control"
})
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => t.CompanyName, new { placeholder = "Назва компанії:", @class =
"form-control" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => t.Experience, new { placeholder = "Досвід роботи:", @class = "form-
control" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextBoxFor(t => employer.FirstName, new { placeholder = "Роботодавець:", @class =
"form-control", @readonly = "true" })
                            </div>
                            <div class="form-group" style = "margin-bottom: 10px">
                                @Html.TextAreaFor(t => t.JobDescription, new { placeholder = "Повний опис вакансії:",
@class = "form-control", rows = "14" })
                            </div>

                            <div style="margin-top:12px">
                                <button type="submit" class="btn btn-block btn-primary">
                                    Оновити
                                </button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
}

```

```

        </form>
    </div>
</div>
</div>
</div>
</div>
}
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
    $(document).ready(function () {
        $('#btnUpload').click(function () {

            // Checking whether FormData is available in browser
            if (window.FormData !== undefined) {

                var fileUpload = $("#fileinput").get(0);
                var files = fileUpload.files;

                // Create FormData object
                var fileData = new FormData();

                // Looping over all files and add it to FormData object
                for (var i = 0; i < files.length; i++) {
                    fileData.append(files[i].name, files[i]);
                }
                $.ajax({
                    url: '/Create/PhotoUpload',
                    type: "POST",
                    contentType: false,
                    processData: false,
                    data: fileData,
                    success: function (result) {
                        alert(result);
                    },
                    error: function (err) {
                        alert(err.statusText);
                    }
                });
            } else {
                alert("FormData is not supported.");
            }
        });
    });
</script>

```