

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проєкту (роботи)

магістра

(ступінь вищої освіти)

на тему ВДОСКОНАЛЕНА СИСТЕМА ПОШУКУ РОБОТИ

Виконав: студент 2 курсу, групи КНТ-622м
спеціальності _____

123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Спеціалізовані комп'ютерні системи

ПАВЕЛКО О.О.

(ПРИЗВИЩЕ та ініціали)

Керівник КУДЕРМЕТОВ Р.К.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЄФИМЕНКО М.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти магістерський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)

Освітня програма (спеціалізація) «Спеціалізовані комп'ютерні системи»
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.

“12” грудня 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ПАВЕЛКО Олександра Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Вдосконалена система пошуку роботи
 керівник проєкту (роботи) к. т. н., зав. каф., КУДЕРМЕТОВ Равіль Камілович
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від “24” жовтня 2023 р. № 400

2. Строк подання студентом проєкту (роботи) 02 грудня 2023 року

3. Вихідні дані до проєкту (роботи) наявні системи пошуку роботи, які доступні за адресами: work.ua, rabota.ua, jooble.org, dou.ua.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

1) Аналіз предметної області;

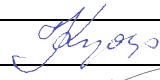
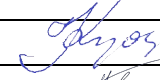
2) Аналіз програмних засобів розробки системи;

3) Розробка вебзастосунку;

4) Експлуатація та тестування системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	КУДЕРМЕТОВ Р. К., зав.каф.		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання 01.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз систем аналогів, їх переваг та недоліків	10.10.2023 р.	
2	Визначення вимог для створення системи	15.10.2023 р.	
3	Вибір засобів розробки системи	20.10.2023 р.	
4	Визначення модулів системи	05.10.2023 р.	
5	Розробка модулів системи	20.11.2023 р.	
6	Тестування програми	22.11.2023 р.	
6	Оформлення отриманих результатів у ПЗ	01.12.2023 р.	

Студент(ка)



(підпис)

Олександр ПАВЕЛКО

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)



(підпис)

Равіль КУДЕРМЕТОВ

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 83 с., 18 табл., 54 рис.

ВАКАНСІЯ, ВЕБЗАСТОСУНОК, ВІДГУК, ГРАФІК, КАНДИДАТ,
ПОШУК РОБОТИ, РЕЗЮМЕ, РОБОТОДАВЕЦЬ, ТЕСТУВАННЯ.

Об'єктом дослідження є системи для пошуку роботи та працівників.

Предметом дослідження є способи удосконалення функціональних можливостей систем для пошуку роботи та працівників

Метою роботи є удосконалення процесу пошуку роботи за рахунок створення системи, що надасть можливість компаніям зменшити час на пошук працівників, а потенційним співробітникам – швидше отримувати інформацію про наявні вакансії.

Робота складається із 4 частин.

У першій частині проведено аналіз предметної області, визначено вимоги до реалізації вдосконаленої системи.

У другій частині проведено аналіз та обрано засоби розробки системи. Для розробки серверної частини вебзастосунку обрано фреймворк Django, СУБД MySQL, візуальної частини – бібліотеку Bootstrap, мову шаблонів DTL, бібліотеку Charts.js та мову програмування JavaScript. В якості архітектурного шаблону вебзастосунку було обрано шаблон MTV.

У третій частині спроектовано основні модулі системи, описано їх взаємодію та надано опис безпосередньо реалізації системи обраними програмними засобами.

У четвертій частині наведено інтерфейс розробленої системи згідно вимог до вдосконаленої системи пошуку роботи, проведено її тестування, проаналізовано отримані результати.

ABSTRACT

Explanatory note to the master's work: 83 pp., 18 tables, 54 figures.

JOB, WEBSITE, FEEDBACK, SCHEDULE, CANDIDATE, JOB SEARCH, RESUME, EMPLOYER, TESTING.

The object of research is systems for searching for work and employees.

The subject of research is methods of improving the functionality of systems for job and employee search

The purpose of the work is to improve the job search process by creating a system that will enable companies to reduce the time it takes to search for employees, and potential employees to receive information about available vacancies faster.

The work consists of 4 parts.

In the first part, the analysis of the subject area was carried out, and the requirements for the implementation of the improved system were determined.

In the second part, the analysis was carried out and the means of system development were selected. The Django framework, the MySQL DBMS, the Bootstrap library, the DTL template language, the Charts.js library, and the JavaScript programming language were chosen for the development of the server part of the web application. The MTV template was chosen as the architectural template of the web application.

In the third part, the main modules of the system are designed, their interaction is described, and a description of the direct implementation of the system with selected software tools is provided.

The fourth part presents the interface of the developed system according to the requirements for an improved job search system, its testing was carried out, and the obtained results were analyzed.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області	9
1.1 Огляд ринку праці в Україні.....	9
1.2 Динаміка безробіття та зайнятості	10
1.3 Складнощі у працевлаштуванні	12
1.4 Аналогічні сучасні системи	14
1.5 Визначення вимог для створення системи.....	21
1.6 Висновки до розділу 1	22
2 Аналіз програмних засобів розробки системи	22
2.1 Програмні засоби для розробки серверної частини вебзастосунку.....	23
2.2 Програмні засоби для розробки візуальної частини вебзастосунку.....	28
2.3 Вибір архітектури вебзастосунку	31
2.4 Аналіз сховища даних.....	33
2.5 Інструментарій для написання програмного коду.....	34
2.6 Висновки до розділу 2.....	35
3 Розробка вебзастосунку	36
3.1 Визначення модулів системи.....	36
3.2 Розробка модулів системи	39
3.3 Висновки до розділу 3.....	54
4 Експлуатація та тестування системи.....	55
4.1 Виконання програми	56
4.2 Тестування програми	75
4.3 Висновки до розділу 4.....	80
Висновки.....	81
Перелік джерел посилання	82

ВСТУП

У сучасному світі, коли наша повсякденна діяльність змінилася завдяки появі та розвитку Інтернету, процес пошуку роботи також зазнав змін. Щодня, безліч осіб в Україні намагаються знайти нові перспективи в професійній діяльності або зробити перший крок у світі працевлаштування.

У добу постійного розвитку технологій, коли весь світ доступний в мережі, створення вдосконаленої системи для пошуку роботи стає необхідністю, оскільки вона зменшує час на пошук вакансій та кандидатів. Вона допомагає як роботодавцям, так і потенційним співробітникам простіше знаходити одне одного, заощаджуючи час. Така система стає не лише вікном в безмежний світ можливостей розвитку себе і своїх кар'єрних перспектив, але й потужним інструментом для тих, хто шукає нове місце роботи або чекає ідеального кандидата для своєї компанії.

Вдосконалена система для пошуку роботи може стати корисним інструментом для обох сторін – роботодавців і робітників, об'єднавши в собі всі корисні функції систем аналогів. Окрім базових функцій систем аналогів, ця платформа надасть можливість для зручної взаємодії роботодавця з кандидатом, можливість отримувати сповіщення про нові вакансії, можливість переглядати детальні графіки значень заробітної плати, кількості та категорій вакансій а також можливість пройти тестування для оцінки рівня певних знань кандидатів.

Таким чином, розробка вдосконаленої системи для пошуку роботи стає не лише актуальною, але й надзвичайно важливою складовою досягнення професійних та кар'єрних цілей.

Метою роботи є удосконалення процесу пошуку роботи за рахунок створення системи, що надасть можливість компаніям зменшити час на пошук працівників, а потенційним співробітникам – швидше отримувати інформацію про наявні вакансії.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести дослідження найпопулярніших вебзастосунків для пошуку роботи в Україні та аналіз їх переваг та недоліків;
- визначити вимоги до вдосконаленої системи пошуку роботи;
- розробити архітектуру системи та алгоритм її функціонування;
- обрати програмне забезпечення для створення системи;
- реалізувати систему із дотриманням зазначених вимог;
- протестувати систему та проаналізувати отримані результати.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд ринку праці в Україні

Останнім часом помітне певне погіршення ситуації на ринку праці в Україні, оскільки спостерігається зниження кількості вакансій, збільшення конкуренції серед кандидатів без досвіду роботи, зменшення цінності диплому вищої освіти при пошуку роботи, зростання кількості безробітних та постійно зростаючу кількість переселенців. В результаті ринок праці останнім часом сформував парадокс, оскільки з одної сторони є велика кількість кандидатів, а з іншої – низька кількість вакансій. Такий стан справ обумовлюється високим рівнем безробіття, наявністю великої кількості кандидатів без досвіду роботи, дефіцитом висококваліфікованих кадрів та низкою кількістю вакансій. Всі ці фактори утворюють велику кількість безробітних, що негативно впливає на ринок праці в Україні [1].

На сьогоднішній день ринок праці в Україні зустрічає безліч актуальних викликів і проблем, які впливають на якість життя та кар'єрний розвиток населення. Для повного розуміння детальної ситуації на ринку праці та розробки необхідних рішень, спрямованих на покращення ситуації, потрібно провести детальний аналіз цих викликів [2].

Одною із головних проблем є рівень безробіття, на який впливають різні фактори, такі як технологічний розвиток, відсутність економічної стабільності, незатребуваність спеціалістів з дипломом вищої освіти, зникнення певних професій та загалом еволюція ринку праці [3].

Іншою значущою трудністю можна відзначити міграцію працездатного населення за кордон, що може критично позначитись на кількості робочих місць та рівні заробітної плати в країні. Відсутність або зменшення кількості висококваліфікованих спеціалістів може зупинити роботу цілих підприємств та компаній.

Слід також виділити іншу важливу проблему, що полягає у складності отримання першої роботи. Основними факторами отримання першої вакансії є досвід роботи та освіта. Отримавши диплом вищої освіти, випускники університетів зазвичай не мають досвіду роботи за своєю спеціальністю, що ускладнює пошук першої роботи. Хтось знаходить першу роботу через декілька місяців, а комусь для цього треба декілька років, що збільшує кількість тимчасово безробітного населення.

1.2 Динаміка безробіття та зайнятості

Одним з важливих факторів ринкової економіки будь-якої країни є безробіття. Кожна країна світу стикається з безробіттям і намагається зменшити його вплив на якість життя громадян, або ж звести його рівень до нуля, що зазвичай неможливо.

Незважаючи на велику кількість досліджень можливих причин зростання рівня безробіття, складну ситуацію в країні, дана проблема все одно потребує особливої уваги та детального вивчення.

Незалежно від ситуації в країні, стан безробіття завжди можна охарактеризувати декількома головними критеріями, а саме наявністю пропозицій на ринку праці, мотивацією людини працювати та здатністю знайти роботу. Вчені поділяють безробіття на фрикційне, що базується на процесі пошуку або очікування вакансії, тобто відсутністю роботи в найближчому майбутньому, та структурне, що може з'явитися в країні під впливом технологічних або соціальних факторів, наприклад якщо з'являється попит на нову професію, яка раніше не існувала або якщо зникає попит на професію, яка існувала досить довго та була важливою складовою економіки країни. Обидва види безробіття приводять до соціального напруження в суспільстві, негативно

впливають на якість життя населення та несуть ризики для економіки. Наслідком безробіття стає міграція населення за кордон, завдяки чому країна втрачає найкращу робочу силу, а саме висококваліфікованих спеціалістів. Останнім часом ситуація з міграцією тільки погіршується, країну продовжують залишати сотні тисяч людей, що може створити серйозні проблеми всередині країни та обмежити потенціал людей до спроможності зниження рівня безробіття [4].

Хоча безробіття і є серйозною проблемою, воно присутнє не тільки в Україні, а й в будь-якій іншій країні світу, оскільки має не лише негативні сторони, а й власні переваги. Серед переваг безробіття можна виділити наступні:

- підсилення активності громадян до власного вдосконалення, збільшення продуктивності роботи, покращення її якості;

- збільшення конкуренції на ринку вакансій;

- зростання значимості робочого місця серед конкурентів;

- активізація змін в певних сферах шляхом їх реформування;

Серед негативних факторів безробіття слід виділити наступні:

- загострення фінансових проблем та зниження якості рівня життя безробітного населення;

- зниження рівня інвестицій в країну;

- посилення напруження в суспільстві;

- збільшення бажання міграції за кордон у населення;

- необхідність виділення додаткових бюджетних коштів на підтримку безробітного населення [5].

Останнім часом показник безробіття в Україні різко зріс та став максимальним за останні 30 років. Міжнародна організація праці зазначає, що рівень безробіття економічно спроможного населення по всіх областях України в 2023 р. становить близько 26.5%, що на 15-17 відсотків більше ніж у 2022 р. Згідно з даними Державної служби статистики, станом на сьогоднішній день серед країн Європи в Україні найбільший показник безробіття. [6].

1.3 Складнощі у працевлаштуванні

Чимало шукачів роботи постійно стикаються з труднощами в процесі пошуку роботи. Перше з чим стикаються шукачі роботи це вибір ресурсу для пошуку роботи. Лише 10-20 років тому людство використовувало газети та журнали, де публікували статті про різну роботу та вакансії. З появою інтернету цей спосіб став застарілим та неефективним, але насправді досить багато людей досі їм користуються. Такий спосіб пошуку роботи сильно обмежує інформацію про наявні вакансії, тому більшість людей зараз використовує вебзастосунки для пошуку роботи. Всі вони мають свої позитивні сторони і недоліки. Для кожної галузі кількість вакансій може різнитися, наприклад для ІТ-сфери найбільш доцільно використовувати вебзастосунок «Doou.ua», оскільки він спеціалізується саме на цій сфері. Але найефективнішим способом пошуку роботи є використання якомога більшої кількості таких систем.

В сучасних реаліях використання якомога більшої кількості вебзастосунків хоч і збільшує шанси на успіх, але не гарантує його. Причин для цього може бути багато, починаючи від високих потреб, вказаних у вакансії, завершуючи високою конкуренцією.

Відомо, що основними факторами успішного працевлаштування є освіта, досвід роботи, особисті якості та спеціальні навички, але щодня вимоги до кандидатів збільшуються, а кількість досвідених працівників зменшуються, що породжує високу конкуренцію серед кандидатів без досвіду.

Гарним прикладом збільшення конкуренції серед шукачів роботи є ІТ-сфера. Останнім часом досить сильно помітно збільшення вимог до потенційних робітників, високу конкуренцію на одне вакантне місце, а кількість заявок може досягати сотні відгуків, що ускладнює процес працевлаштування для роботодавців та збільшує конкуренцію серед шукачів. Висока конкуренція також

може сприяти зменшенню заробітної плати, оскільки цінність спеціалістів сильно зменшується [7].

Наслідком високої конкуренції є підвищення якості кандидатів, їх мотивації до самовдосконалення, готовність працювати більше та якісніше а також бажання виділитися серед конкурентів. Основним фактором успішного працевлаштування для кандидата є резюме, а для роботодавця ефективно залучення співробітника до своєї компанії визначає вакансія [8].

Важливу роль у подоланні всіх перелічених вище труднощів грає резюме. Резюме визначає сильні сторони кандидата та зазвичай має в собі перелік попередніх місць роботи, що надає роботодавцю певне уявлення про кандидата, а також служить основним аргументом для запрошення на співбесіду. Саме тому резюме повинно бути правильно складено, оскільки роботодавцю важливо за мінімальну кількість часу зрозуміти якомога більше про кандидата.

Існують хронологічні та функціональні резюме. Хронологічні резюме містять список місць роботи та перелік професійних обов'язків у хронологічному порядку. Його слід використовувати кандидатам, в яких є суттєвий досвід роботи та якщо можна прослідкувати кар'єрний зріст у певній професійній сфері. В функціональних резюме увага приділяється професійним навикам та досягненням. Такий формат демонструє саме опис обов'язків, які виконувалися на певній роботі, а не перелік місць де працював кандидат. Такі резюме слід використовувати коли кандидат прийняв рішення про зміну професійної діяльності [9].

Роботодавці відчують схожі проблеми з кандидатам, оскільки вони конкурують з іншими компаніями за якісних співробітників, але вони знаходяться в більш вигідному становищі ніж кандидати, оскільки конкуренція серед кандидатів вища ніж серед компаній. Правильно створена вакансія може пригорнути увагу якомога більшої кількості потенціальних співробітників, що дозволить обрати кращого, або знизити заробітну плату на вакантне місце та таким чином зменшити витрати компанії.

1.4 Аналогічні сучасні системи

На сьогоднішній день в Україні існує багато вебзастосунків для пошуку роботи, серед яких найпопулярнішими є «Work.ua», «Rabota.ua», «Jooble.org» та «Dou.ua». Всі вони мають приємний інтерфейс, підтримують базовий функціонал, такий як створення, пошук за фільтрами, можливість перегляду вакансій та кандидатів, але все ж таки відрізняються деякими функціями. Тому критично важливим стало проведення аналізу переваг та недоліків цих вебзастосунків, щоб виявити сильні сторони кожного з них та сформулювати вимоги до вдосконаленої системи пошуку роботи [10].

«Work.ua» останнім часом є найпопулярнішим вебзастосунком для пошуку вакансій та кандидатів для роботи по всій Україні. Він має простий, дружній для користувача інтерфейс та майже всі необхідні функції для професійного зростання компанії, або людини, що шукає роботу. Ця система надає вакансії в різних категоріях в майже кожному регіоні країни. Роботодавці можуть зареєструвати свою компанію, після чого розміщувати оголошення про вакансії та шукати відповідних кандидатів. Користувачі можуть зареєструватись, після чого їм буде надана можливість завантажувати резюме та підписуватись на повідомлення про нові вакансії. Крім того, сайт надає корисну статистику про поточний стан ринку праці та поради щодо успішного пошуку роботи, а також цікаві статті відомих успішних людей.

Вебзастосунок максимально простий у використанні, оскільки для пошуку вакансій навіть не потрібно мати власний обліковий запис, достатньо ввести назву посади, що цікавить користувача в пошукове поле, після чого система знайде відповідні вакансії. «Work.ua» пропонує широкий вибір фільтрів для більш точного пошуку вакансій, що робить його зручним інструментом для шукачів. На головній сторінці відразу чекає добірка різних пропозицій від

найперспективніших компаній – спортивних брендів, медичних фірм, закладів харчування та мобільних операторів.

На рис. 1.1 наведено інтерфейс вебзастосунку «Work.ua».

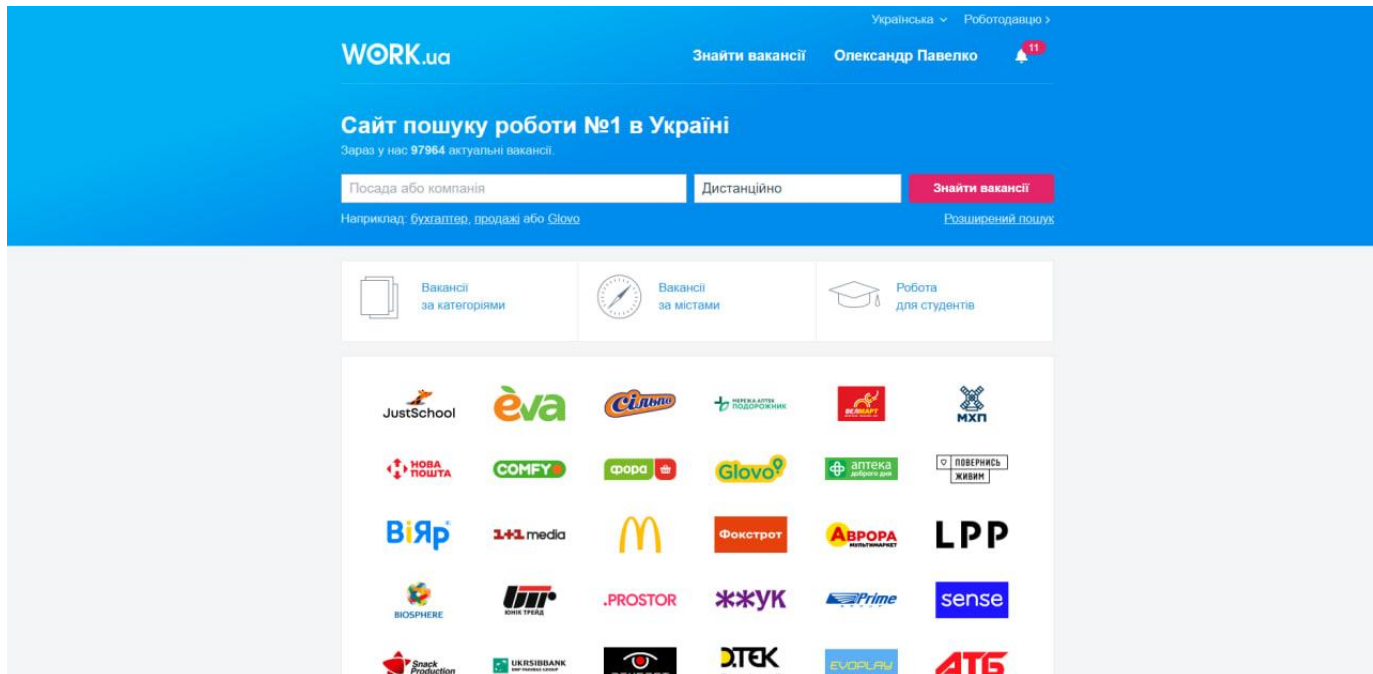


Рисунок 1.1 – Головна сторінка вебзастосунку «Work.ua»

Позитивні риси «Work.ua»:

- підтримка базового функціоналу;
- можливість отримувати сповіщення про нові вакансії на електронну пошту для підписаних на компанію користувачів, що дозволяє їм слідкувати за новими вакансіями без постійного входу на сайт;
- наявність статистики та графіків заробітної плати, кількості та категорій вакансій, що дозволяє детально проаналізувати ринок праці в певний момент часу;
- наявність зручної взаємодії між роботодавцем та потенційним працівником.

Недоліки «Work.ua»:

– відсутність можливості пройти тестування або опитування для кандидата з метою визначення рівня його певних знань.

Наступним аналогом є «Rabota.ua». Сайт відрізняється зручною системою пошуку роботи як за назвами спеціальностей, так і за посадами. Зареєстровані користувачі можуть залишати резюме, підписуватись на корисні розсилки, читати статті різних експертів та успішних підприємців. Роботодавці також можуть розміщувати вакансії для своєї компанії та переглядати резюме потенційних співробітників. Для зареєстрованих користувачів доступна особиста панель з інформацією про збережені вакансії та резюме.

«Rabota.ua» також проводить регулярні дослідження ринку праці, які можуть бути корисними як роботодавцям, так і шукачам роботи. Ці дослідження включають аналіз заробітних плат, вимог до кваліфікації, тренди на ринку праці і прогнози. Для роботодавців це означає доступ до актуальних даних про зарплати та конкурентоспроможність вакансій, що допомагає їм привертати талановитих працівників. Кандидати можуть користуватися цією інформацією для визначення справедливих очікувань щодо заробітної плати і кар'єрного зростання.

На рис. 1.2 наведено інтерфейс вебзастосунку «Rabota.ua».

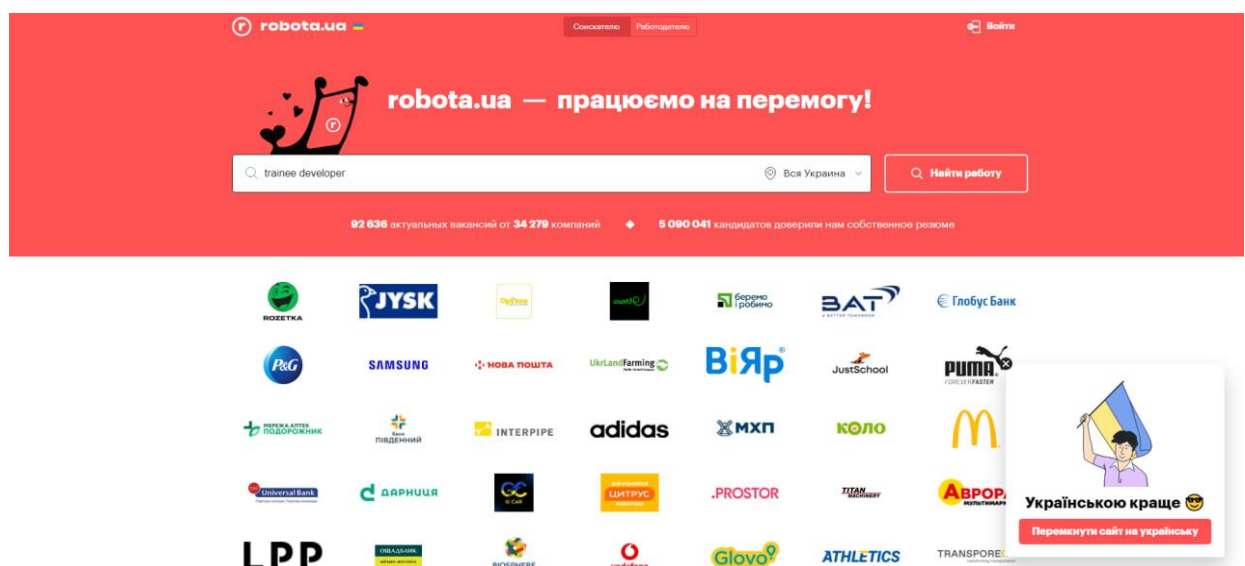


Рисунок 1.2 – Головна сторінка вебзастосунку «Rabota.ua»

Позитивні риси «Rabota.ua»:

- підтримка базового функціоналу;
- можливість отримувати сповіщення про нові вакансії на електронну пошту;
- наявність статистики та різних графіків;
- можливість пройти різного роду опитування та тестування перед відгуком на вакансію, що дозволяє роботодавцю оцінити рівень знань та мотивацію кандидата ще до першого спілкування з ним.

Недоліки «Rabota.ua»:

- відсутність зручної взаємодії кандидата та роботодавця, що може призвести до втрати можливості швидко обговорювати деталі вакансії та вимоги, що в свою чергу може призвести до неефективного оброблення заявок і затримок у процесі найму.

«Jooble.org» є ще одним аналогом, який відрізняється від попередніх тим, що збирає інформацію про вакансії з різних сайтів і джерел в інтернеті, і надає результати пошуку на своїй платформі. Він не надає власних оголошень про вакансії, на відміну від «Work.ua» та «Rabota.ua», які розміщують вакансії безпосередньо на своїх платформах.

«Jooble.org» не обмежується Україною та працює у різних країнах, що робить його більш міжнародним ресурсом по відношенню до аналогів, що були описані вище, які в свою чергу фокусуються на вакансіях в Україні. Це особливо корисно для тих, хто розглядає можливості працевлаштування за кордоном або для компаній, які шукають міжнародних кандидатів. Оскільки останнім часом дистанційна робота набирає обертів, це надає можливість кандидатам з будь-якого куточку світу працювати на компанії з інших країн, вакансії яких можна знайти у даному вебзастосунку.

Вебзастосунок приваблює розумною пошуковою системою, яка фільтрує вакансії за посадою, геолокацією, та зарплатою, а також дозволяє отримувати сповіщення про нові вакансії, що відповідають цим запитам. Це допомагає

користувачам слідкувати за робочими місцями, які відповідають їхнім інтересам, без необхідності постійного відвідування сайту.

Хоча «Jooble.org» має багато переваг та особливостей, слід пам'ятати, що він залежить від даних, наданих іншими сайтами для пошуку роботи, тому точність та актуальність вакансій може коливатися. Також, для деяких функцій, таких як взаємодія між кандидатом та роботодавцем, тестування кандидатів та детальна статистика, користувачам скоріш за все потрібно буде відвідати конкретний сайт з вакансією, оскільки сам «Jooble.org» не підтримує дані функції на своїй платформі.

На рис. 1.3 наведено інтерфейс вебзастосунку «Jooble.org».

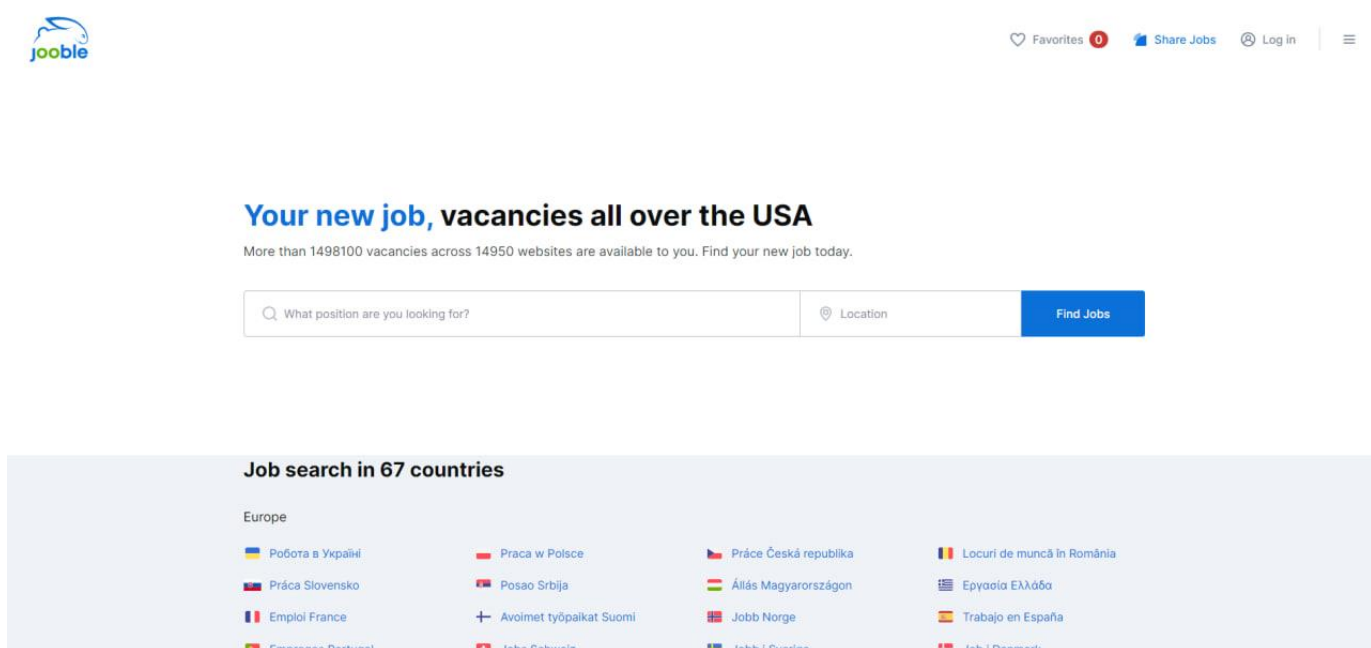


Рисунок 1.3 – Головна сторінка вебзастосунку «Jooble.org»

Позитивні риси «Jooble.org»:

- підтримка базового функціоналу;
- велика кількість вакансій, яка пояснюється тим, що вебзастосунок знаходить вакансії на інших популярних сайтах для пошуку роботи, які надають вакансії на своїх платформах;

- наявність міжнародних вакансій, що дозволяє користувачам знайти роботу за кордоном, або працювати в дистанційному режимі;

- можливість отримувати сповіщення про нові вакансії за вказаними раніше параметрами на електронну пошту.

Недоліки «Jooble.org»:

- відсутність чату або можливості відгукнутись на вакансію для взаємодії між роботодавцем та потенційним кандидатом, що зменшує шанси на успішну співпрацю обох сторін;

- відсутність можливості пройти тестування або опитування для кандидата з метою визначення рівня його певних знань;

- відсутність статистичних даних та графіків, що відображають динаміку зростання та зменшення кількості вакансій та середньої заробітної плати, що обмежує розуміння реального стану справ на ринку праці.

«Dou.ua» сильно відрізняється від інших аналогів, оскільки він надає інформацію про вакансії, пов'язані саме з програмуванням, розробкою, дизайном, тестуванням та іншими аспектами інформаційних технологій. Це робить його ідеальним ресурсом як для професіоналів у сфері інформаційних технологій, так і для компаній, які шукають кваліфікованих співробітників.

«Dou.ua» забезпечує можливість розміщення резюме та пошуку вакансій, та його цільова аудиторія оцінює можливість отримання інформації про робочі місця, пов'язані з технологіями та програмуванням.

«Dou.ua» містить різноманітні розділи, включаючи форум для обговорення питань і обміну досвідом між користувачами, доступ до новин і статей, наданий журналістам і редакціям, інформацію про роботодавців, вакансії і відгуки про роботу в ІТ-компаніях, а також календар подій та заходів у сфері ІТ.

На рис. 1.4 наведено інтерфейс вебзастосунку «Dou.ua».

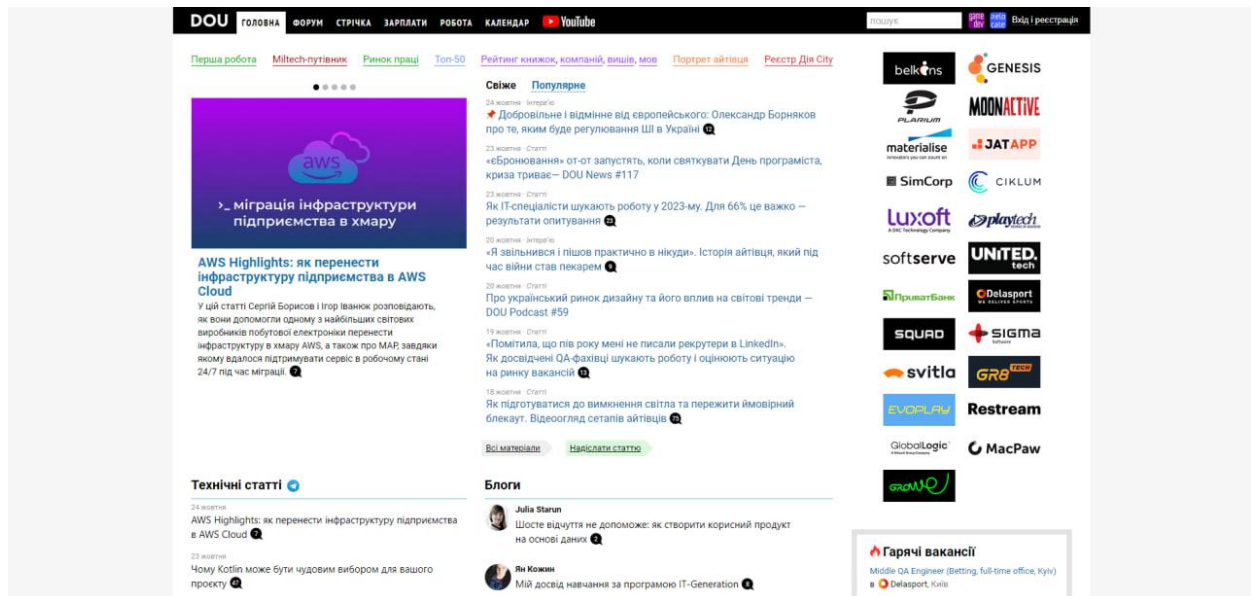


Рисунок 1.4 – Головна сторінка вебзастосунку «Dou.ua»

Позитивні риси «Dou.ua»:

- підтримка базового функціоналу;
- багато вакансій у сфері IT і стартапів, що робить даний ресурс особливо корисним для фахівців цих сфер;
- наявність форуму як альтернативи для взаємодії потенційного кандидата та роботодавця;
- багато цікавих новин, технічних статей та календарю подій заходів у сфері IT, що може підвищити рівень знань кандидата та бути корисним при пошуку першої роботи.

Недоліки «Dou.ua»:

- основна частина вакансій спрямована на фахівців з IT сфери та стартапів, що анулює користь даного вебзастосунку для фахівців інших сфер;
- відсутність можливості пройти тестування або опитування для кандидата з метою визначення рівня його певних знань;
- відсутність статистичних даних та графіків, що відображають реальний стан справ на ринку праці.

Вище було проаналізовано різні вебзастосунки, які надають користувачам широкий спектр можливостей для знаходження бажаної роботи. Проте, варто

звернути увагу на якісні характеристики кожного з них, для того, щоб виявити їх переваги та недоліки. В табл. 1.1 наведено порівняння популярних програмних продуктів.

Таблиця 1.1 – Порівняння аналогічних систем

Критерій	«Work.ua»	«Rabota.ua»	«Jooble.org»	«Dou.ua»
Виконання базового функціоналу	Так	Так	Так	Так
Отримання сповіщень про нові вакансії на електронну пошту	Так	Так	Так	Ні
Наявність зручної взаємодії кандидата та роботодавця	Так	Ні	Ні	Так
Наявність статистики та графіків про вакансії та кандидатів	Так	Так	Ні	Ні
Можливість пройти тестування для оцінки рівня знань кандидата	Ні	Так	Ні	Ні

1.5 Визначення вимог для створення системи

При аналізі популярних аналогічних вебзастосунків для пошуку роботи було виявлено переваги та недоліки кожного з них, а також визначено вимоги до реалізації вдосконаленої системи, а саме:

- виконання базового функціоналу;
- отримання сповіщень про нові вакансії на електронну пошту;
- наявність зручної взаємодії кандидата та роботодавця у вигляді можливості відгукуватися на вакансію;
- наявність статистики та графіків;
- можливість пройти тестування для оцінки рівня знань кандидата.

1.6 Висновки до розділу 1

В цьому розділі було здійснено дослідження предметної області, виявлені труднощі при працевлаштуванні, обґрунтована важливість правильного складання резюме. Проаналізовано популярні аналоги, які реалізовані у вигляді вебзастосунку, а також визначено переваги та недоліки кожного з них. На підставі цього аналізу визначено вимоги до вдосконаленої системи пошуку роботи, які будуть впроваджені при розробці відповідного програмного забезпечення, зокрема, виконання базового функціоналу, отримання сповіщень про нові вакансії на електронну пошту, наявність зручної взаємодії кандидата та роботодавця у вигляді відгуку на вакансію, наявність статистики та графіків, а також можливість пройти тестування для оцінки рівня знань кандидата.

2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ СИСТЕМИ

В процесі аналізу популярних вебзастосунків для пошуку роботи були визначені переваги та недоліки кожного з них, а також встановлені вимоги до реалізації вдосконаленої системи. Наступним етапом є визначення концепції розроблюваної системи та вибір інструментів для її створення.

З метою полегшення користування програмним забезпеченням для пошуку роботи, необхідно реалізувати його у вигляді вебзастосунку. Це надасть можливість користувачам працювати з системою на різних пристроях та ділитися посиланням на вакансії незалежно від їх місцезнаходження та типу пристрою.

2.1 Програмні засоби для розробки серверної частини вебзастосунку

Оскільки програмне забезпечення буде реалізовано у вигляді вебзастосунку, корисним буде використання ORM (Object-Relational Mapping). Це технологія, яка дозволяє взаємодіяти з реляційною БД (базою даних) через об'єктно-орієнтований код. Використання ORM-фреймворка для створення вебзастосунку має кілька важливих переваг і може бути необхідним з кількох причин:

- ORM-фреймворки надають можливість розробникам працювати з базою даних на найвищому абстрактному рівні, використовуючи об'єктно-орієнтований підхід. Це спрощує процес розробки та зменшує кількість коду;
- ORM-фреймворки можуть надавати абстракцію, яка дозволяє взаємодіяти з різними системами управління базами даних (СУБД) без змін в програмному коді;
- ORM-фреймворки надають можливість безпечно взаємодіяти з БД, мінімізуючи ризик SQL-ін'єкцій та інших безпекових загроз;
- ORM дозволяє спільно використовувати моделі даних між різними частинами додатка, що полегшує роботу з даними та підвищує продуктивність розробки;
- ORM-фреймворки можуть автоматизувати багато рутинних завдань, таких як створення запитів до бази даних і перетворення даних на об'єкти, що дозволяє розробникам фокусуватися на бізнес-логіці застосунку [11].

Вибір фреймворку для розробки вебзастосунку для пошуку роботи здійснювався на вимогах до вдосконаленої системи пошуку роботи, згідно з поставленим завданням. Було обрано три найпопулярніші варіанти, які підтримують використання ORM-фреймворків, а саме Django, ASP.NET та Laravel. Нижче було виконано аналіз цих фреймворків для розробки функціоналу системи.

Розглянемо цю структуру детальніше:

- таблиця `auth_user` зберігає дані про користувачів, такі як ім'я, пароль, та інші атрибути. Пароль завжди зберігається у вигляді хешу, завдяки чому зловмисники не зможуть заволодіти ним маючи доступ до бази даних;
- таблиця `auth_user_groups` встановлює відносини між користувачами та групами, визначаючи до яких груп належить кожен користувач;
- таблиця `auth_group` містить інформацію про групи користувачів;
- таблиця `auth_group_permissions` містить інформацію про дозволи груп користувачів;
- таблиця `auth_permission` містить права доступу, які можуть бути надані користувачам або групам користувачів;
- таблиця `auth_user_user_permissions` визначає конкретні дозволи, які надаються окремим користувачам.

Слід зазначити, що в інших фреймворках вбудовані таблиці реалізовані схожим чином.

Laravel також має високорівневий ORM, який називається Eloquent. Laravel має зручний синтаксис та шаблонізатор Blade. Його розширюваність підтримується багатьма пакетами та активною спільнотою. Laravel забезпечує вбудовану систему управління ролями через модель Eloquent, яка дозволяє визначати та керувати доступом користувачів.

ASP.NET використовує Entity Framework для роботи з базою даних. Можна використовувати Code First підхід, де моделі класів перетворюються в таблиці в базі даних за допомогою міграцій. Entity Framework надає можливість вручну визначати схему бази даних або використовувати автоматичну генерацію.

Розглянемо типи зв'язків в БД для кожного фреймворку:

- Django використовує `ForeignKey`, `OneToOneField`, `ManyToManyField` для визначення зв'язків між моделями;
- Laravel використовує Eloquent-зв'язки, такі як `hasOne`, `hasMany`, `belongsTo`, `belongsToMany`;

– ASP.NET застосовує відносини типу One-to-One, One-to-Many, та Many-to-Many за допомогою звичайних або атрибутів Fluent API.

Розглянемо переваги Django в контексті роботи з базою даних:

– Django надає простий та зручний інтерфейс для взаємодії з базою даних, спрощуючи весь процес від створення таблиць до виконання запитів;

– Django ORM забезпечує гнучкість у визначенні відносин та можливість вручну керувати структурою бази даних, що дає розробникам більше контролю;

– вбудована таблиця адміністратора в Django спрощує управління даними та дозволяє здійснювати CRUD-операції без додаткового коду;

– Django пропонує готові інструменти для автоматизації створення бази даних, що робить розробку швидшою та менш витратною.

В підсумку, Django забезпечує високорівневий та потужний інтерфейс для роботи з базою даних, що робить його прекрасним вибором для швидкої та ефективної розробки вебзастосунків.

Виконання базового функціоналу включає реєстрацію користувачів, автентифікацію за роллю, додавання даних кандидата, реєстрацію компанії роботодавця, додавання, видалення та редагування вакансій роботодавцем, перегляд та пошук вакансій за фільтрами. Django є відмінним вибором для виконання базового функціоналу. Використання вбудованої системи автентифікації та Django ORM робить реєстрацію користувачів, авторизацію, та управління даними (додавання кандидатів, реєстрація роботодавця та вакансій) простим та зручним. Django Admin забезпечує потужний інструмент для створення адміністративної панелі та управління вмістом вебзастосунка за допомогою адміністратора системи.

Laravel також є високопродуктивним фреймворком для виконання такого функціоналу. Вбудована система автентифікації Laravel і Eloquent ORM полегшують реалізацію реєстрації, авторизації та управління вакансіями та кандидатами. Відсутність вбудованого адміністративного інтерфейсу може вимагати використання сторонніх пакетів.

ASP.NET також є потужним інструментом для розробки вебзастосунків з реєстрацією користувачів, авторизацією та управлінням вакансіями та кандидатами. Використання системи Identity для автентифікації та Entity Framework для взаємодії з базою даних робить його гнучким у реалізації подібного функціоналу. Велика кількість інструментів та ресурсів Microsoft дозволяє ефективно взаємодіяти з різними аспектами розробки.

Для пошуку вакансій та кандидатів за фільтрами Django має вбудовані інструменти у вигляді функцій для зручної фільтрації за параметрами. Laravel та ASP.NET не мають власних реалізацій для пошуку та фільтрації за параметрами, але дозволяють створювати складні запити до бази даних та фільтрувати дані за заданими критеріями.

Реалізація даного функціоналу може бути досягнута в будь-якому з цих фреймворків, але Django та Laravel надають більш готові рішення та спрощують розробку.

Для створення та відправки сповіщень про нові вакансії Django, Laravel та ASP.NET надають можливості відправки сповіщень через різні канали.

Для відображення статистики та графіків заробітної плати, кількості та категорій вакансій Django, Laravel та ASP.NET будуть однаково ефективними, оскільки підтримують різні бібліотеки для створення графіків і діаграм.

Для можливості відгуку кандидатом на вакансію всі фреймворки дозволяють створювати форми, де кандидат зможе завантажити свій файл резюме та супровідний лист, а для тестування знань кандидата можна створювати форми з запитаннями, які потім будуть порівняні системою з правильними відповідями, таким чином надавши можливість тестування знань кандидатів.

В табл. 2.1 наведено порівняння фреймворків для реалізації модулів розроблюваної системи.

Таблиця 2.1 – Порівняння фреймворків для реалізації функціональних вимог розроблюваної системи

Вимога	Django	ASP.NET	Laravel
База даних	ORM, СУБД	ORM, СУБД	ORM, СУБД
Виконання базового функціоналу	Вбудований інтерфейс	Сторонні рішення	Вбудований інтерфейс
Пошук вакансій за фільтрами	Вбудовані інструменти	Складні запити до бази даних	Складні запити до бази даних
Отримання сповіщень про нові вакансії на електронну пошту	Відправка через SMTP та інші канали	Відправка через SMTP та інші канали	Відправка через SMTP та інші канали
Можливість кандидата відгукуватися на вакансію та пройти тестування	Форми	Форми	Форми
Статистика та графіки	Використання бібліотек	Використання бібліотек	Використання бібліотек

Порівнявши фреймворки для реалізації модулів розроблюваної системи, враховуючи вказані вимоги, було обрано саме фреймворк Django, оскільки він має ряд переваг, а саме:

- вбудований інтерфейс адміністрування, що спрощує керування моделями даних та правами доступу;
- готові рішення для реєстрації та автентифікації користувачів;
- наявність вбудованих реалізацій для фільтрації даних [12].

2.2 Програмні засоби для розробки візуальної частини вебзастосунку

Візуальна частина програми складається з взаємопов'язаних HTML (HyperText Markup Language) вебсторінок, за допомогою яких користувачу надано інтерфейс для взаємодії з системою. Для зручнішого користування системою було використано різні технології, які покращують зовнішній вигляд вебсторінок. Ці технології будуть розглянуті нижче.

2.2.1 Bootstrap

Bootstrap – це бібліотека, яка надає готові класи стилів, які можуть бути використані прямо в коді HTML. Для використання цих класів достатньо вказати лише назву цього класу в відповідному тезі HTML. Такий підхід дозволяє дуже швидко покращити зовнішній вигляд вебсторінок, що може бути зручно для розробників, для яких пріоритетом є саме функціонал системи, а також для тих, хто хоче заощадити час на зовнішньому вигляді вебзастосунку [13].

Кожна вебсторінка складається з різних елементів, але основні елементи включають наступні:

- `<html>` обгортає весь вміст сторінки і вказує на початок і кінець HTML-документа;
- `<head>` містить теги заголовку, підключених стилів, скриптів та інші;
- `<title>` визначає заголовок вебсторінки, який відображається на вкладці браузера;
- `<body>` містить весь вміст, який буде відображено на сторінці, такий як тексти, зображення, посилання і інші елементи;
- `<header>` містить заголовок або верхню частину сторінки, містить логотип, назву сайту тощо;
- `<nav>` використовується для створення навігаційного меню, яке може містити посилання на інші сторінки або розділи сайту;
- `<body>` містить основний вміст сторінки, такий як текст, зображення, відео, тощо;
- `<footer>` вказує на кінець сторінки і може містити контактну інформацію, посилання на соціальні мережі, авторські права і інше.

Для всіх цих елементів Bootstrap пропонує готові класи стилів, такі як `nav`, `nav-link`, `container`, `footer`, `text-center` та інші, тому вони будуть використані при розробці візуальної частини програми [14].

2.2.2 Django Template Language

Django надає можливість використання серверної розмітки у вигляді шаблонів, мова яких називається Django Template Language (DTL) і може бути використана для відображення даних переданих з серверної частини застосунку на сторінку. Вона надає спеціальний синтаксис, завдяки якому в HTML-кодї можна користуватися різними конструкціями, такими як цикли, масиви та змінні, які були передані з серверної частини застосунку. Цими змінними можна користуватись для виводу їх значень. Наприклад, `{{ variable }}` вставитиме значення змінної. DTL також надає можливості для логічних операцій, використовуючи такі умовні оператори як `if`, `else`, `elif`, які дозволяють гнучко керувати відображенням контенту в залежності від різних умов в шаблонах [15].

2.2.4 Charts.js

Для відображення графіків, які будуть містити дані про вакансії та кандидатів було обрано бібліотеку Charts.js. Вона здатна створювати та динамічно відображати інтерактивні графіки та діаграми на вебсторінках. Ця бібліотека надає простий і зручний інтерфейс для створення різноманітних графічних візуалізацій. Можна виділити наступні особливості Charts.js:

- Charts.js дуже легка для використання і включає в себе простий API, що дозволяє швидко створювати графіки та діаграми без глибокого розуміння складних технічних деталей;
- Charts.js підтримує різноманітні типи графіків, такі як лінійні графіки, стовпчасті графіки, кругові діаграми, точкові графіки і інші;
- можливість підсвічування даних при наведенні, а також анімації при оновленні даних;
- багато опцій для налаштування вигляду графіків, включаючи кольори, шрифти, розміри, події та інше;
- графіки можуть легко адаптуватися до розмірів контейнера, що полегшує їх використання в різних макетах та відображеннях;

– Charts.js має велику активну спільноту, а також власну документацію і приклади, що полегшують вивчення і використання бібліотеки.

2.2.4 JavaScript

JavaScript – це мова програмування, яка буде використана для керування поведінкою вебсторінок, а також для взаємодії з бібліотекою Charts.js, яка є потужним інструментом для створення та відображення графіків. JavaScript легко інтегрується з іншими технологіями користувацького інтерфейсу, такими як HTML, CSS та Bootstrap. За допомогою функцій JavaScript будуть створені та налаштовані графіки, а також відображені в правильному форматі дані [16].

2.3 Вибір архітектури вебзастосунку

Класичним архітектурним шаблоном, що використовується в фреймворку Django для побудови вебзастосунків є MTV (Model-Template-View). MTV є варіацією архітектури MVC (Model-View-Controller), тому основна його роль полягає в розподіленні обов’язків між різними компонентами, що спрощує розробку вебзастосунків та забезпечує більш логічну організацію коду [17].

На рис. 2.2 зображена архітектура Django MTV вебзастосунку.

Основними компонентами MTV-архітектури в Django можна виділити наступні:

– модель (Model) відповідає за роботу вебзастосунку з базою даних та відображає певну сутність (користувач, вакансія тощо). Для всіх моделей Django за замовченням містить базові функції, такі як CRUD (створення, читання, оновлення, видалення) операції. Модель в Django використовує ORM для взаємодії з базою даних, що дозволяє використовувати об’єкти для роботи з даними;

– шаблон (Template) визначає вигляд вебсторінок та відображення даних користувачам. Django використовує спеціальну мову шаблонізації, яка дозволяє вбудовувати дані, які будуть передані з моделей та відображення в HTML код сторінок. Шаблони допомагають відокремити логіку від візуальної частини програми, що сприяє більш чистому та підтримуваному коду;

– відображення (View) визначає логіку додатку та контролює взаємодію з користувачем. Воно приймає HTTP-запити, взаємодіє з моделями та вибирає відповідні шаблони для відображення результатів.

Таким чином, обираючи MTV архітектуру для вебзастосунку з використанням Django, розробники забезпечують собі зручну та ефективну платформу для розробки, що дозволяє швидко створювати складні вебзастосунки та забезпечує легкість їх управління та розширення у майбутньому [18].

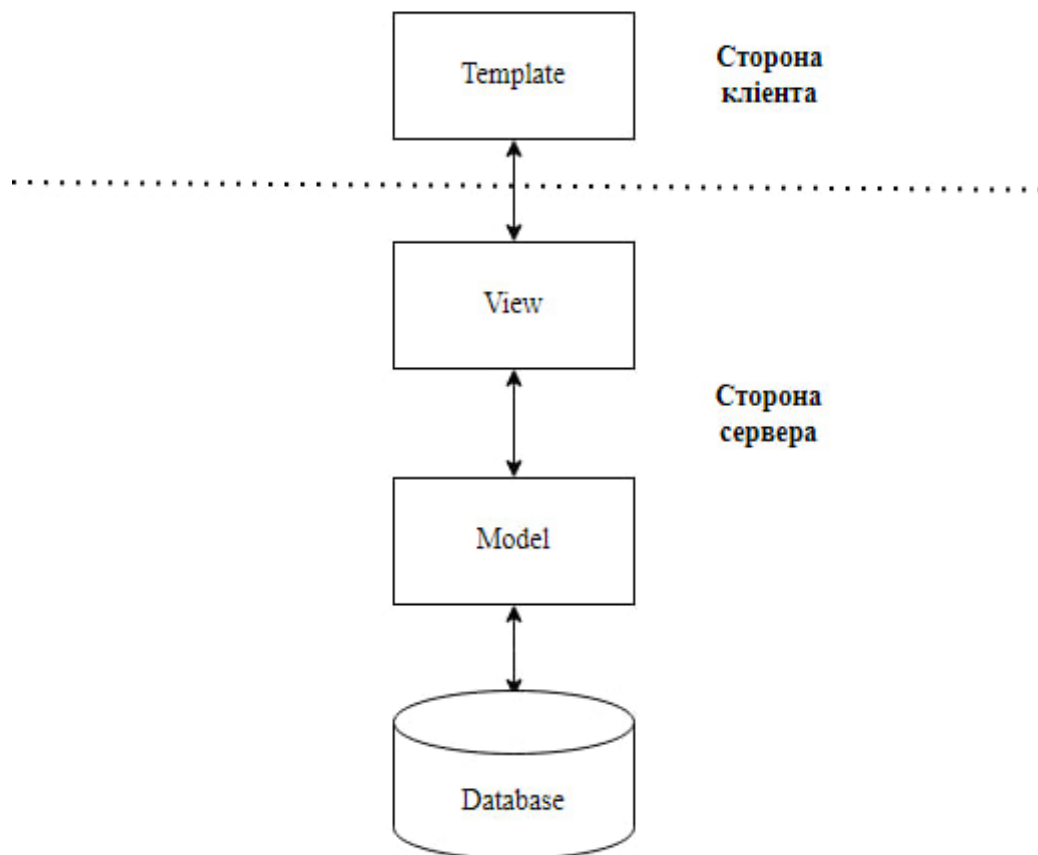


Рисунок 2.2 – Архітектура Django MTV вебзастосунку

2.4 Аналіз сховища даних

Для ефективної роботи вебзастосунку та забезпечення оптимальної організації даних, виникає необхідність у СУБД. При виборі СУБД, перш за все, необхідно визначити її тип. Оскільки розроблюваний вебзастосунок передбачає наявність зв'язків між моделями даних та організацію фіксованих таблиць, було прийнято рішення використовувати реляційні СУБД. Зокрема, при розробці на фреймворку Django найбільш популярними реляційними СУБД є PostgreSQL, MySQL та SQLite [19].

В табл. 2.2 наведено порівняння реляційних СУБД для розроблюваного вебзастосунку.

Таблиця 2.2 – Порівняння реляційних СУБД

Параметр порівняння	PostgreSQL	MySQL	SQLite
Структура даних	Гнучка	Гнучка	Спрощена
Тип бази даних	Серверна	Серверна	Файлова
Мова запитів	SQL	SQL	SQL
Швидкодія	Добра	Висока	Середня
Запити за індексами	Хороша підтримка	Хороша підтримка	Обмежена підтримка

При виборі бази даних варто зазначити, що MySQL та PostgreSQL є серверними базами даних, що означає, що вони взаємодіють із клієнтськими додатками через мережу. Ці бази даних можуть бути розгорнуті як окремі служби або сервери, що працюють незалежно від основного вебзастосунку. Цей підхід дозволяє розділити обробку даних і запитів між вебзастосунком та базою даних, сприяючи високій продуктивності та масштабованості.

З іншого боку, SQLite є вбудованою базою даних, яка зберігає дані в одному локальному файлі. Вона працює в межах того ж самого процесу, що й додаток, і не вимагає окремого серверу. Хоча цей підхід може бути зручним для

простих або вбудованих застосунків, йому може бути важко впоратися з великою кількістю запитів або великими обсягами даних.

Розгортання MySQL або PostgreSQL як окремих серверів сприяє покращенню безпеки та забезпеченню ефективного використання ресурсів. Також цей підхід полегшує процеси резервного копіювання та відновлення даних, що робить систему більш надійною та готовою до вирішення можливих проблем.

Оскільки MySQL має високу швидкодію, підтримку зв'язків між даними, та має гнучку структуру даних, було обрано саме цю СУБД.

2.5 Інструментарій для написання програмного коду

Для розробки вебзастосунків на Django використовуються різні IDE (інтегровані середовища розробки) та редактори коду, такі як PyCharm, Visual Studio Code та Visual Studio IDE.

Редактори коду, як Visual Studio Code (VS Code), зазвичай дуже легкі та швидкі і підтримують розширення для додавання функціональності, але спрямовані на просте редагування коду на відміну від IDE, які в свою чергу надають повноцінні інструменти для розробки, такі як інтегровані дебагери, підказки під час розробки, автоматичне створення проєктів та багато іншого. Вони надають інструменти для керування проєктами, роботи з системами контролю версій, відлагодженням, автоматичним завершенням коду, вбудованими консолями та іншими корисними речами. В табл. 2.3 наведено порівняння інструментів для написання коду [20].

За результатами, наданими у таблиці 2.3 видно, що PyCharm має більше функцій редагування коду, має вбудований вебсервер та надає можливості для розширення. Тому для розробки проєкту було обрано саме його.

Таблиця 2.3 – Порівняння інструментів для написання коду

Критерії	PyCharm	Visual Studio Code	Visual Studio
Підтримка Django	Так	Так	Так
Функції редагування	Велика кількість функцій	Основні функції	Велика кількість функцій
Вбудований дебагер	Так	Так	Так
Наявність вебсервера	Вбудований	Тільки командний рядок для запуску Django сервера	Вбудований
Підтримка від розширень	Так	Ні	Так

2.6 Висновки до розділу 2

В даному розділі було проаналізовано та обґрунтовано вибір засобів розробки відповідно до функціональних вимог системи. Обрано фреймворк Django, оскільки він має потужну ORM, яка надає простий та зручний інтерфейс для взаємодії з базою даних, містить вбудовані інструменти для пошуку і фільтрації вакансій, систему авторизації, адміністративну панель та управління ролями, а також надає розробникам можливість швидкого створення вебзастосунків. Для розробки візуальної частини вебзастосунку обрано бібліотеку Bootstrap для створення стилів, мову шаблонів DTL для відображення даних та бібліотеку Charts.js для відображення графіків. В якості архітектурного шаблону вебзастосунку було обрано шаблон MTV, оскільки він забезпечує зручну та ефективну платформу для розробки, що дозволяє швидко створювати складні вебзастосунки та забезпечує легкість їх управління та розширення у майбутньому. Для зберігання даних було обрано СУБД MySQL, оскільки вона надає всі необхідні інструменти, має високу швидкодію, підтримку зв'язків між даними, та гнучку структуру даних. Було обрано середовище розробки PyCharm для створення програмного коду, оскільки воно має багато функцій редагування коду, вбудований вебсервер та можливості для розширення.

3 РОЗРОБКА ВЕБЗАСТОСУНКУ

3.1 Визначення модулів системи

Виходячи з вимог до розробки вдосконаленої системи пошуку роботи було прийнято рішення поділити розробку на наступні модулі:

- модуль бази даних;
- модуль основного функціоналу;
- модуль створення та відправки сповіщень про нові вакансії;
- модуль взаємодії між кандидатом та роботодавцем;
- модуль статистики та графіків.

Модуль «База даних» буде включати в себе різні таблиці, такі як: вакансія, компанія, користувач, роботодавець, кандидат, відгук, місто, контакти, запитання, відповідь, тестування кандидату. Для цих таблиць має бути реалізовано основні операції взаємодії з даними, а саме додавання, зчитування, видалення та змінення.

Модуль «Основний функціонал» містить наступні можливості для користувачів:

- реєстрація облікового запису та можливість входу в систему залежно від ролі особистого запису. Реєстрація та авторизація передбачена тільки для кандидатів та роботодавців. Адміністратор створюється при першому запуску застосунку та має окрему сторінку для авторизації в якості адміністратора, яка вже реалізована фреймворком;
- адміністрування сайтом включає в себе створення адміністраторського облікового запису та надання йому повного доступу до системи, включаючи можливість додавання інших адміністраторів;
- можливість реєстрації компанії для роботодавця та створення, редагування і видалення вакансій до неї;

- можливість кандидатів скористатися пошуком за назвою вакансії та фільтрами, а також переглядати вакансії та пов'язані з ними компанії;

- додавання додаткових даних про кандидатів, таких як місто проживання, рівень досвіду, освіта, бажана вакансія та бажана заробітна плата.

Модуль «Створення та відправка сповіщень про нові вакансії» призначений для відправки на електронну пошту кандидата сповіщень про нові вакансії, які відповідають його вказаним критеріям для бажаної вакансії.

Модуль «Взаємодія між співробітником та роботодавцем» надає можливість кандидатам реагувати на вакансії, відправляючи своє резюме та супровідний лист. Роботодавець може переглядати резюме, відправляти відповіді та тестувати кандидатів.

Модуль «Статистика та графіки зарплат та вакансій» призначений для графічного відображення зміни кількості вакансій, їх категорій та медіани зарплат.

Структура модулів програми зображена на рис. 3.1.

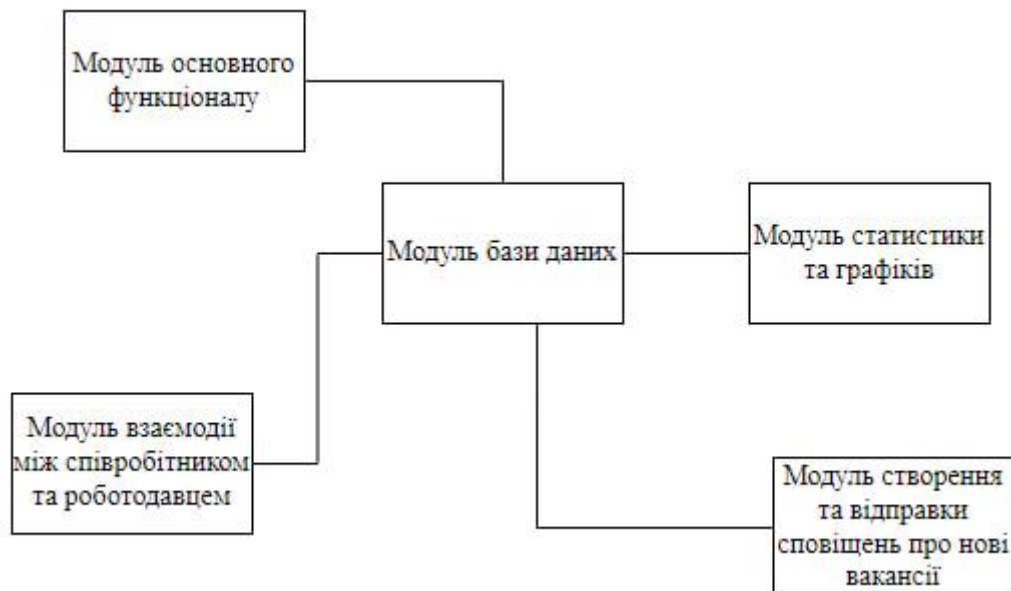


Рисунок 3.1 – Структурна схема модулів системи

3.2 Розробка модулів системи

3.2.1 Модуль бази даних

Необхідною складовою системи є база даних, оскільки проєкт передбачає взаємодію між різними ролями користувачів та зберігання їх даних. База даних буде відповідальною за ефективне зберігання та швидкий доступ до даних для оптимального функціонування системи та взаємодії між її різними компонентами. На рис. 3.3 зображено схему БД.

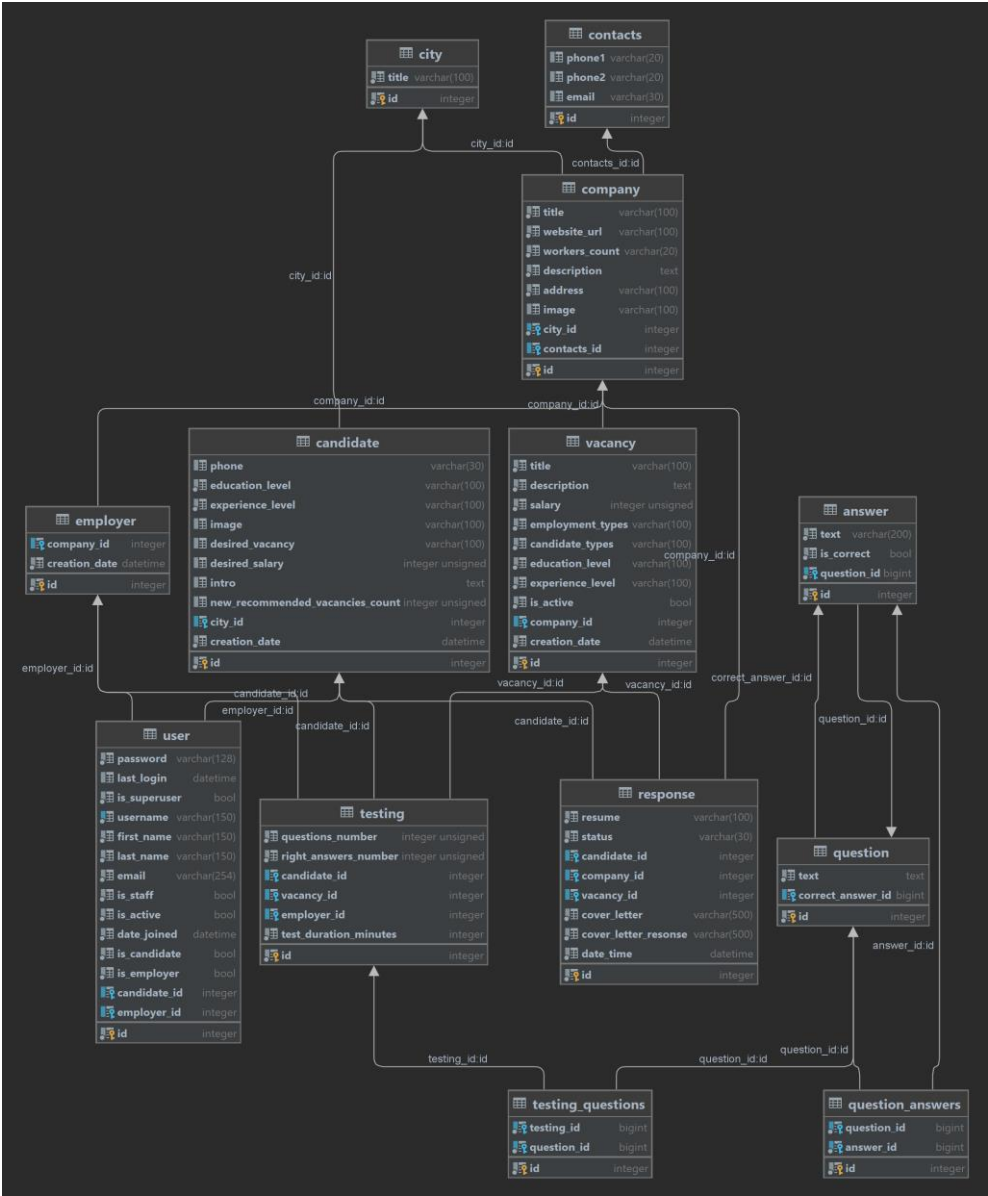


Рисунок 3.3 – Схема бази даних

Таблиця City визначає місто, яке може бути пов'язане з компанією або кандидатом (табл. 3.1).

Таблиця 3.1 – Таблиця City

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
title	CharField(100)	Назва міста

Таблиця Contacts містить конкретну інформацію про контактні дані компанії (табл. 3.2).

Таблиця 3.2 – Таблиця Contacts

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
phone1	CharField(20)	Номер телефону
phone2	CharField(20)	Додатковий номер телефону (не обов'язково)
email	CharField(30)	Електронна пошта роботодавця або людини, яка відповідальна за компанію

Таблиця Company містить конкретну інформацію про компанію роботодавця (табл. 3.3).

Таблиця 3.3 – Таблиця Company

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
title	CharField(100)	Назва компанії
website_url	CharField(100)	Адреса вебсайту компанії
workers_count	CharField(20)	Кількість працівників
description	TextField(1500)	Опис компанії
city	ForeignKey(City)	Місто, до якого належить компанія
address	CharField(100)	Адреса компанії
image	ImageField	Логотип компанії
contacts	OneToOneField(Contacts)	Контактна інформація компанії

Таблиця Vacancy містить інформацію про вакансії роботодавця, які були додані до його компанії (табл. 3.4).

Таблиця 3.4 – Таблиця Vacancy

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
title	CharField(100)	Назва вакансії
description	CharField(1500)	Опис вакансії
creation_date	DateTimeField	Дата створення вакансії
salary	PositiveIntegerField	Зарплата (грн/міс.)
employment_types	CharField(100)	Тип зайнятості
candidate_types	CharField(100)	Типи кандидатів
education_level	CharField(100)	Освіта
experience_level	CharField(100)	Досвід
company	ForeignKey(Company)	Компанія, яка пропонує вакансію
is_active	BooleanField	Активна вакансія

Таблиця Candidate містить інформацію про кандидатів, які шукають роботу (табл. 3.5).

Таблиця 3.5 – Таблиця Candidate

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
phone	CharField(30)	Номер телефону
creation_date	DateTimeField	Дата реєстрації
education_level	CharField(100)	Освіта
experience_level	CharField(100)	Рівень досвіду
image	ImageField	Фото кандидата
desired_vacancy	CharField(100)	Бажана вакансія
desired_salary	PositiveIntegerField	Бажана зарплата (грн/міс.)
intro	TextField(2000)	Інформація про себе
city	ForeignKey(City)	Місто, де знаходиться кандидат
new_recommended_vacancies_count	PositiveIntegerField	Кількість нових рекомендованих вакансій

Таблиця Response зберігає відгуки кандидатів на вакансії, відповіді на них та інше (табл. 3.6).

Таблиця 3.6 – Таблиця Response

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
candidate	ForeignKey(Candidate)	Кандидат, на якого надійшов відгук
vacancy	ForeignKey(Vacancy)	Вакансія, на яку надійшов відгук
date_time	DateTimeField	Дата відгуку
company	ForeignKey(Company)	Компанія, яка залишила відгук
cover_letter	CharField(500)	Супровідний лист кандидата
cover_letter_resonse	CharField(500)	Відповідь роботодавця кандидату
resume	FileField	Резюме кандидата
status	CharField(30)	Статус відгуку

Таблиця Employer містить інформацію про роботодавців, які можуть створити компанію для подальшого керування її вакансіями (табл. 3.7).

Таблиця 3.7 – Таблиця Employer

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
creation_date	DateTimeField	Дата реєстрації
company	ForeignKey(Company)	Компанія роботодавця (якщо зареєстровано)

Таблиця Testing містить інформацію про тестування кандидатів (табл. 3.8).

Таблиця 3.8 – Таблиця Testing

Поле	Тип даних	Опис
employer	ForeignKey(Employer)	Роботодавець
candidate	ForeignKey(Candidate)	Кандидат
vacancy	ForeignKey(Vacancy)	Вакансія
questions_number	PositiveIntegerField	Кількість питань
right_answers_number	PositiveIntegerField	Кількість правильних відповідей
questions	ManyToManyField(Question)	Питання для тестування
test_duration_minutes	PositiveIntegerField	Максимальна тривалість тестування, хвилин

Таблиця User створюється за допомогою відповідного класу User, який успадковується від базового класу AbstractUser і переймає частину його полів (табл. 3.9).

Таблиця 3.9 – Таблиця User

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
username	CharField(150)	Ім'я користувача
email	CharField(128)	Пароль
password	EmailField	Електронна пошта
first_name	CharField(30)	Ім'я
last_name	CharField(150)	Прізвище
is_active	BooleanField	Чи активний користувач
is_staff	BooleanField	Має доступ до адмін-панелі
date_djoined	DateTimeField	Дата реєстрації
is_candidate	BooleanField	Чи є користувач кандидатом
is_employer	BooleanField	Чи є користувач роботодавцем
candidate	OneToOneField(Candidate)	Профіль кандидата (якщо is_candidate=True)
employer	OneToOneField(Employer)	Профіль роботодавця (якщо is_employer=True)

Таблиця Testing_Questions містить запитання до конкретного тестування (табл. 3.10).

Таблиця 3.10 – Таблиця Testing_Questions

Поле	Тип даних	Опис
id	TextField(200)	Унікальний ідентифікатор
question	ForeignKey(Answer)	Запитання до тестування
testing	ManyToManyField(Answer)	Тестування

Таблиця Answer містить відповіді на відповідні запитання з конкретного тестування (табл. 3.11). Таблиця Question містить усі запитання для кожного тестування (табл. 3.12), а таблиця Question_Answers (табл. 3.13) – відповідно відповіді на запитання.

Таблиця 3.11 – Таблиця Answer

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
question	OneToOneField(Question)	Зв'язане питання
text	CharField(200)	Текст відповіді
is_correct	BooleanField	Чи є відповідь правильною

Таблиця 3.12 – Таблиця Question

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
text	TextField(200)	Текст запитання
correct_answer	ForeignKey(Answer)	Правильна відповідь
answers	ManyToManyField(Answer)	Відповіді на питання

Таблиця 3.13 – Таблиця Question_Answers

Поле	Тип даних	Опис
id	AutoField	Унікальний ідентифікатор
question	ForeignKey(Question)	Запитання
answer	ForeignKey(Answer)	Правильна відповідь

3.2.2. Модуль базового функціоналу

Модуль базового функціоналу має містити наступні можливості:

- адміністрування сайтом;
- створення CRUD (Create, Read, Update, Delete) операцій, які надають можливість перегляду, додавання, редагування та видалення для моделей системи;
- реєстрація та аутентифікація користувачів;
- пошук вакансій та кандидатів за фільтрами.

Для адміністрування сайтом Django надає вбудовану адміністративну панель Django Admin, яка може бути створена, налаштована та розширена. Django автоматично генерує адміністративний інтерфейс на основі моделей вебзастосунку, що дозволяє легко додавати, редагувати та видаляти дані без необхідності писати власний код для адміністративного інтерфейсу.

Першим кроком для реалізації можливості адміністрування вебзастосунком необхідно створити самого адміністратора, який буде керувати

системою, маючи до неї повний доступ. В Django це можна зробити за допомогою команди `python manage.py createsuperuser`. Після введення команди треба заповнити відповідні поля, а саме ім'я користувача, його електронна адреса, пароль та повторення пароля, при чому пароль повинен бути достатньо складним, бо інакше адміністратора не буде створено. На рис. 3.4 наведено процедуру створення адміністратора.

```
Microsoft Windows [Version 10.0.19044.1645]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

(worksearcher) D:\University\4\python\worksearcher>python manage.py createsuperuser
Username (leave blank to use 'alex'): admin
Email address: admin_worksearcher@gmail.com
Password:
Password (again):
Superuser created successfully.
```

Рисунок 3.4 – Створення адміністратора

Для того, щоб переконатись що адміністратора було успішно створено і подивитись який саме тепер функціонал йому тепер доступний, треба запустити вебзастосунок командою `python manage.py runserver`, після чого відкрити вкладку браузера та ввести `localhost:8080/admin`, після чого відкриється сторінка авторизації для адміністратора, де потрібно вказати ім'я користувача та його пароль, які було створено раніше. На рис. 3.5 наведено сторінку авторизації для адміністратора.

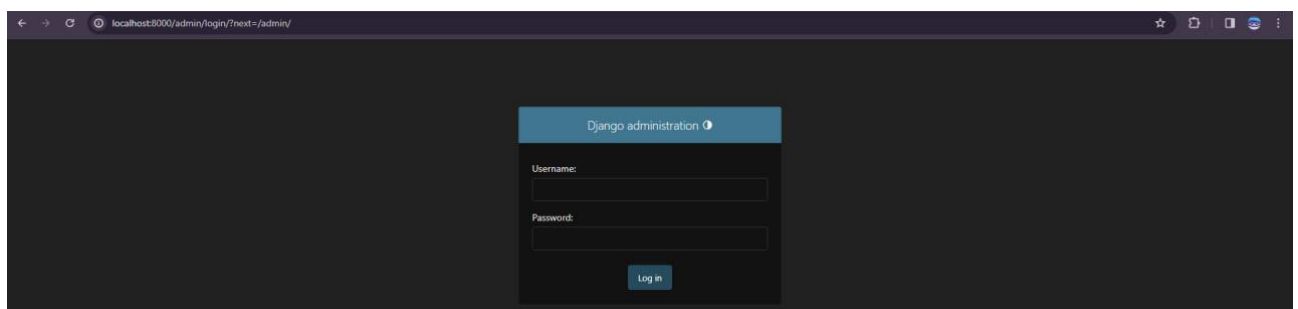


Рисунок 3.5 – Сторінка входу для адміністратора

При спробі неправильно ввести ім'я користувача або пароль, буде відображено помилку та доступ до системи не буде надано (рис. 3.6).

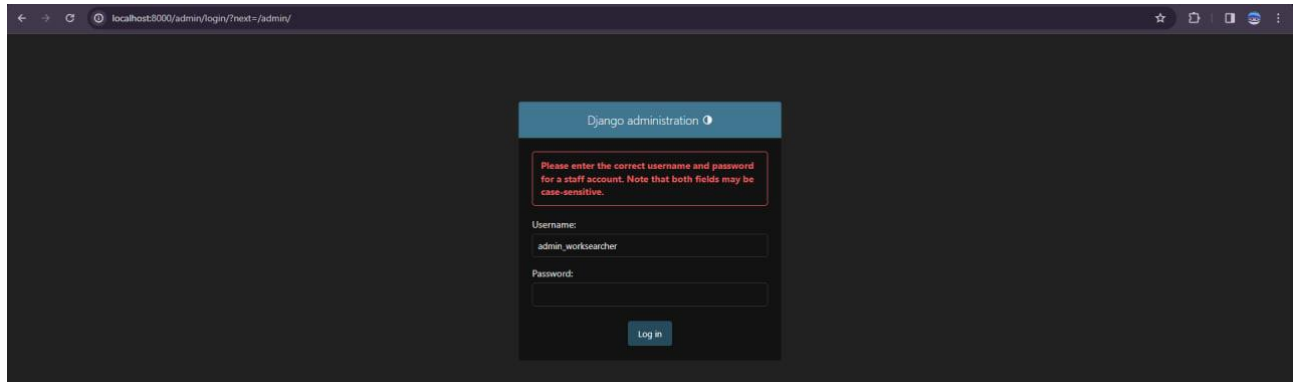


Рисунок 3.6 – Спроба входу адміністратора при неправильному імені користувача чи паролі

Якщо ім'я користувача та пароль було введено правильно система надасть доступ до адміністративної панелі (рис. 3.7). Можна побачити, що жодних моделей, які були створені в базі даних, окрім стандартних (групи користувачів та користувачі) в адміністративній панелі немає, але для стандартних моделей вже доступний функціонал їх створення (кнопка Add) та редагування (Change).

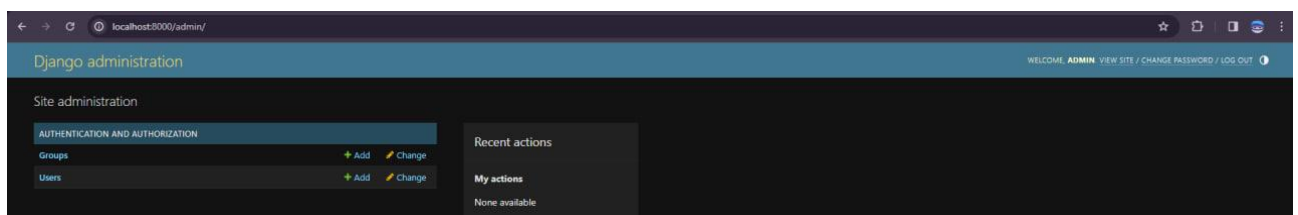


Рисунок 3.7 – Адміністративна панель

Django містить вбудовані можливості для зміни мови адміністративної панелі та загалом всього застосунку, що дуже зручно, оскільки не потрібно створювати та наповнювати файли локалізації, в Django достатньо змінити лише один параметр, а саме `LANGUAGE_CODE`, вказавши йому відповідний код мови. Django підтримує більшість популярних мов, включаючи українську, тому

в відповідний параметр було встановлено код української мови (uk-UA), після чого адміністративна панель буде відображена українською мовою (рис. 3.8).

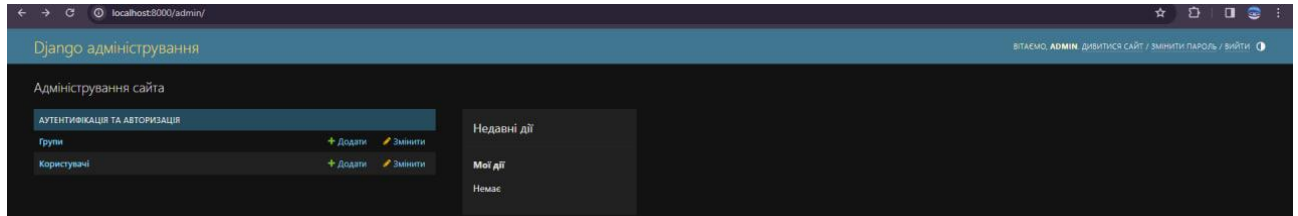


Рисунок 3.8 – Локалізована українською мовою адміністративна панель

Для відображення в адміністративній панелі всіх моделей, які були створені в базі даних та можливості керування ними адміністратором необхідно визначити класи для кожної моделі та зареєструвати їх за допомогою функції `admin.site.register()` в класі `admin.py` проєкта Django, після чого для кожної моделі треба визначити клас, який розширює `admin.ModelAdmin`. Для того, щоб визначити, яка інформація буде відображатися та фільтруватися у списку моделей в адміністративній панелі, треба визначити ці класи в файлі `admin.py` і задати їх параметри, такі як `list_display`, `search_fields`, `list_filter` а також `verbose_name`. Після цього адміністративна панель буде містити потрібні моделі та їх поля (рис. 3.9).

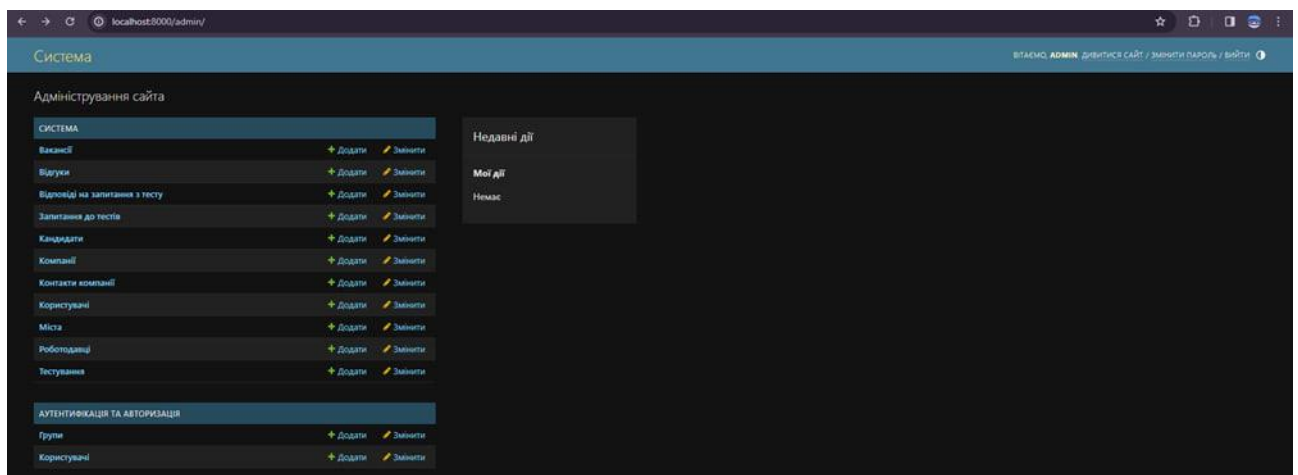


Рисунок 3.9 – Адміністративна панель з усіма моделями

Тепер, коли всі моделі відображені для адміністратора, він має повний доступ до них, та може переглядати, створювати нові екземпляри цих моделей, редагувати та видаляти їх. Створення та налаштування адміністративної панелі завершено.

Наступною вимогою до базового функціоналу системи є створення CRUD операцій для моделей системи, завдяки чому ними буде легко керувати на рівні бази даних. В Django є наступні класи, які входять до групи класів, які спрощують реалізацію операцій CRUD для моделей:

- `CreateView`, який використовується для створення нових об'єктів моделі. Він автоматично генерує форму для введення даних та обробляє її відправку, додаючи новий запис до бази даних;

- `DetailView` дозволяє відображати деталі конкретного об'єкту моделі. Він використовується для відображення інформації про один об'єкт, забезпечуючи сторінку з докладним переглядом;

- `UpdateView` дозволяє користувачам редагувати існуючі об'єкти моделі. Він автоматично виводить форму з поточними значеннями та оновлює запис в базі даних після відправлення форми;

- `DeleteView`, який використовується для видалення об'єктів моделі. Він підтверджує бажання користувача видалити об'єкт та здійснює відповідну операцію в базі даних;

- `ListView`, який дозволяє відображати список всіх об'єктів моделі на одній сторінці. Він надає зручний спосіб перегляду всіх записів та перехід до їх детального перегляду.

Дані класи будуть використані при розробці CRUD операцій для вакансій, компаній роботодавців та кандидатів.

Однією з вимог до базового функціоналу системи є реалізація реєстрації та автентифікації для користувачів. Django дозволяє легко реалізувати систему реєстрації та автентифікації для користувачів у вебзастосунку. При реєстрації, користувачеві буде запропоновано заповнити необхідні дані, такі як логін,

електронна пошта, ім'я, прізвище, пароль та підтвердження пароллю. Django забезпечує захист паролів, оскільки вони не зберігаються у відкритому вигляді, а замість цього зберігаються у вигляді хешів.

Після успішної реєстрації, користувач може використовувати введений логін та пароль для входу на сайт. Під час входу було надано можливість користувачеві обрати роль, таку як кандидат чи роботодавець. Ця система ролей дозволяє надати відповідний функціонал залежно від потреб користувача.

Роль «кандидат» надає доступ до функціоналу, пов'язаного з пошуком вакансій, можливістю відгукнутись на вакансії та керуванням особистим профілем. З іншого боку, роль «роботодавець» відкриває можливості створення та публікації вакансій, перегляду резюме кандидатів і управління власною компанією а також тестування знань кандидата.

Така архітектура дозволяє ефективно використовувати функціонал Django для зручної та безпечної реєстрації та авторизації користувачів у вебзастосунку з урахуванням їхніх ролей та потреб.

Останньою вимогою до базового функціоналу є пошук вакансій та кандидатів за фільтрами. Django надає потужні засоби для фільтрації даних, що дозволяє легко реалізувати пошук вакансій та кандидатів за різними параметрами. Для цього використовується функціонал Django QuerySet та можливості фільтрації. Для реалізації пошуку вакансій за фільтрами, буде використано фільтрацію за полями моделі Vacancy, такими як місто, бажана зарплата, тип зайнятості, освіта, рівень досвіду та іншими. Використовуючи Q-об'єкти, можна об'єднати кілька умов для складніших запитань. Для фільтрації кандидатів можна використовувати аналогічний підхід, фільтруючи кандидатів за параметрами, такими як рівень освіти, досвід роботи, бажана вакансія та іншими.

Зокрема, Django також надає можливість використовувати фільтри в адміністративній панелі для зручності редагування та відображення даних.

Такий підхід дозволяє створювати потужні та гнучкі системи пошуку, які враховують різноманітні потреби користувачів, будь то роботодавець, який шукає кандидатів, або кандидат, який шукає відповідні вакансії.

3.2.3 Модуль створення та відправки сповіщень на пошту

Функція отримання сповіщень на електронну пошту про нові вакансії стає допоміжним інструментом для кандидатів, спрямованим на оптимізацію та полегшення їх пошукового процесу. Завдяки цій функції, кандидат зможе отримувати інформацію про нові можливості безпосередньо на свою пошту, що виключає необхідність постійного моніторингу різних вебзастосунків. Кожне сповіщення буде враховувати індивідуальні побажання кандидата до вакансії, забезпечуючи отримання сповіщень про лише ті вакансії, які будуть відповідати його очікуванням.

Було прийнято рішення встановити один чіткий часовий інтервал, коли система буде надсилати оповіщення на пошту кандидата, тому що такий підхід має наступні переваги:

- можливість уникнути надмірної кількості повідомлень, що потрапляють на пошту кандидата;
- кандидат може спокійно розраховувати на отримання важливих повідомлень лише в конкретний часовий проміжок, що полегшує його планування та зосередженість на інших завданнях;
- конкретний час відправлення сповіщень може допомогти кандидатові створити режим перевірки пошти, що робить взаємодію з системою більш усвідомленою та зосередженою;
- одноразове відправлення сповіщень у визначений час буде більш ефективним, оскільки воно зменшить навантаження на сервер та оптимізує використання електронної пошти.

В результаті ця практика сприяє покращенню користувацького досвіду та взаємодії з системою, забезпечуючи більш зручний та ефективний спосіб

отримання важливої інформації для кандидатів а також оптимізацію використання електронної пошти.

Для реалізації відправки сповіщень на електронну пошту треба спочатку обрати бібліотеку для роботи з електронною поштою. Всі бібліотеки використовують протокол SMTP (Simple Mail Transfer Protocol) для того, щоб надсилати електронні листи через електронну пошту. SMTP – це протокол, який використовується для надсилання електронних листів між серверами електронної пошти. В табл. 3.14 наведено порівняння бібліотек для роботи з електронною поштою.

Таблиця 3.14 – Порівняння бібліотек для роботи з електронною поштою

Критерій	django.core.mail	django-anymail	Flask-Mail	yagmail	smtplib
Зручність використання	Висока	Висока	Висока	Середня	Висока
Документація	Хороша	Хороша	Хороша	Задовільна	Задовільна
Функціональність	Середня	Висока	Середня	Висока	Середня
Розширюваність	Обмежена	Висока	Обмежена	Висока	Висока
Вбудована підтримка Django	Так	Ні	Ні	Так	Так
Основна причина використання	Вбудована підтримка Django	Розширена функціональність	Висока гнучкість, повний контроль	Зручний інтерфейс, висока гнучкість	Вбудована підтримка Django

Для системи, яка передбачає щоденну розсилку сповіщень про нові вакансії кандидатам у чіткий часовий інтервал, використання smtplib може бути найбільш відповідним варіантом з наступних причин:

- smtplib надає повний контроль над процесом відправки електронних листів. Можна легко налаштувати індивідуальні листи для кожного кандидата та відправляти їх у визначений час;

- використання smtplib дозволяє працювати безпосередньо з SMTP-серверами, полегшуючи управління відправкою та сприяючи швидкості доставки листів;

- використання `smtp`lib вважається відносно простим та прозорим. Це полегшує процес налаштування та забезпечує стабільну роботу відправлення сповіщень;

- `smtp`lib є частиною стандартної бібліотеки Python, тому він підтримує використання Django;

- `smtp`lib надає повний контроль над вмістом листа, дозволяючи динамічно генерувати та персоналізувати сповіщення для кожного кандидата.

3.2.4 Модуль взаємодії між кандидатом та роботодавцем

Для ефективної роботи вебзастосунку необхідно забезпечити взаємодію кандидатів та роботодавців. Було прийнято додати в цей розділ наступний функціонал:

- можливість кандидатам реагувати на вакансії, відправляючи своє резюме та супровідний лист, після чого роботодавець зможе переглядати резюме та відправляти відповідь на нього;

- можливість роботодавцем додати тестування до вакансії для кандидатів з метою виявлення їх рівня знань. Після проходження кандидатом тестування, роботодавець побачить його результат та прийме рішення чи підходить йому цей кандидат.

Для реалізації можливості відгуку кандидатом на вакансію буде створено кнопку «Відгукнутися» на сторінці вакансії, яка відкриє сторінку з формою де кандидат зможе завантажити свій файл резюме та супровідний лист. Для роботодавця буде реалізовано сторінку перегляду відгуків на кожну вакансію його компанії, де після перегляду даних кандидата можна буде відправити йому відповідь у вигляді текстового повідомлення або надіслати йому тест для перевірки його знань. На сторінці перегляду відгуку на вакансію можна буде змінювати статус кандидата. Будуть доступні наступні статуси кандидата: «розглядається», «прийнято» та «відхилено». У випадку обрання статусу «прийнято», контактні дані обох сторін стануть доступні для перегляду один одному.

Можливість тестування кандидатом буде реалізована наступним чином: роботодавець може додати тест до вакансії, після чого на сторінці відгуків за бажанням зможе надіслати його кандидату, який відгукнувся на вакансію. Тест створюється роботодавцем шляхом заповнення відповідної форми, де роботодавець визначає час, за який може бути пройдено тестування, запитання, варіанти відповідей та правильну відповідь на кожне запитання. Кандидат може пройти тестування поки вакансія активна. Проходження тестування буде відбуватися на відповідній сторінці шляхом вибору натискання кандидатом на варіант відповіді, який на думку кандидата є правильним. Після закінчення тестування система порахує кількість правильних відповідей та відправить результат роботодавцю.

3.2.5 Модуль графіків і статистики

З точки зору кандидата та роботодавця, наявність статистики та графіків на сайті пошуку роботи є надзвичайно важливою функцією. Для кандидата це може бути корисно з наступних причин:

- статистика заробітної плати надає кандидату можливість оцінити ринкові стандарти та реальні очікування щодо оплати праці в конкретній галузі чи регіоні;

- графіки кількості та категорій вакансій допомагають кандидату розробити стратегію ефективного пошуку, визначити сектори з підвищеним попитом та підготувати засоби для конкурентних галузей.

Для роботодавця статистика буде реалізована у вигляді графіків, які будуть відображати рівень освіти кандидатів та їх рівень досвіду роботи, завдяки чому роботодавець зможе швидко та зручно знаходити кандидатів. Графік дозволяє роботодавцю сконцентруватися на кандидатах із певним рівнем кваліфікації та освіти, завдяки чому зможе обирати тих, хто відповідає конкретним вимогам вакансії.

Для створення графіків буде використано бібліотеку Charts.js для відображення графіку а також бібліотеку statistics для підрахунку медіанної

зарплати. Для створення даних, які будуть використані та відображені у графіках буде створено такі функції з наступним функціоналом:

- `get_vacancy_count_data()` формує дані для графіку, який відображає кількість вакансій для кожного міста після чого дані сортуються та обрізаються до перших п'яти міст з найбільшою кількістю вакансій;

- `get_categories_data()` створює дані для графіку, який відображає відсоток вакансій для кожної категорії відносно загальної кількості вакансій після чого дані також сортуються та обрізаються до перших п'яти категорій;

- `get_salary_data()` формує дані для графіку, що показує медіанну зарплату для п'яти вакансій з найвищою зарплатою. Обчислення медіани буде виконуватися за допомогою бібліотеки `statistics`;

- `get_candidates_edu_data()` створює дані для графіку, який відображає відсоток кандидатів за рівнем освіти відносно загальної кількості кандидатів, що мають обрану бажану вакансію;

- `get_candidates_exp_data()` формує дані для графіку, який відображає відсоток кандидатів за рівнем досвіду відносно загальної кількості кандидатів, що мають обрану бажану вакансію.

Ці функції отримують дані з моделей Django (таких як `City`, `Category`, `Vacancy`, `Candidate`, тощо), обчислюють необхідні показники та формують відповідні дані для графіків у форматі JSON, які будуть використані для візуалізації за допомогою бібліотеки `Charts.js`.

3.3 Висновки до розділу 3

В даному розділі було визначено модулі системи, які будуть реалізовані. Модуль бази даних, який буде містити саму базу даних, яка буде складатися з таких таблиць, як `City`, `Contacts`, `Company`, `Vacancy`, `Candidate`, `Response`,

Employer, User, Testing, Testing_Questions, Answer, Question та Question_Answers. Модуль базового функціоналу буде містити можливості адміністрування сайтом, створення CRUD операцій для моделей системи, реєстрацію та автентифікацію користувачів, а також пошук вакансій та кандидатів за фільтрами. Модуль створення та відправки сповіщень на пошту буде містити можливості для відправки сповіщень кандидату на електронну пошту про нові вакансії. Модуль взаємодії кандидата та роботодавця буде містити можливості для відгуку кандидатом на вакансію і тестування його знань. Модуль графіків і статистики буде містити можливості для графічного відображення рівня заробітної плати і кількості та категорій вакансій для кандидатів і рівня досвіду та освіти кандидата для роботодавця.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

Головним призначенням системи є удосконалення процесу пошуку роботи за рахунок створення системи, що надасть можливість компаніям зменшити час на пошук працівників, а потенційним співробітникам – швидше отримувати інформацію про наявні вакансії.

Для повноцінної роботи системи користувач має бути зареєстрованим та авторизованим за роллю. Існують наступні ролі: адміністратор, кандидат та роботодавець. Адміністратор має повний доступ до системи, може керувати нею та створювати інших адміністраторів. Кандидат може шукати вакансії за фільтрами, переглядати конкретну вакансію та відгукуватися на неї, надіславши своє резюме та супровідний лист роботодавцю, а також переглядати контакти компанії роботодавця у випадку якщо він схвалив кандидата. Роботодавець може зареєструвати власну компанію, додавати та редагувати її дані, додавати, видаляти та редагувати вакансії до неї, переглядати відгуки на вакансії,

схвалювати кандидатів а також за бажанням надсилати їм тести до вакансії для виявлення знань кандидатів, а також відповідати на відгуки кандидатів, переглядати контактні дані кандидатів, які були схвалені ним.

4.1 Виконання програми

Щоб розпочати використання вебзастосунок, необхідно спочатку запустити його на вебсервері. Після цього вебзастосунок буде доступний за певною адресою, за якою користувачі зможуть звертатися до нього. Вебзастосунок було запущено на локальному комп'ютері за стандартною адресою вебзастосунків Django, а саме 127.0.0.1:8000. Після запуску вебзастосунку треба перейти за цією адресою в адресному рядку браузера, після чого відкриється головна сторінка вебзастосунку.

На головній сторінці (рис. 4.1) користувач може побачити наступні блоки:

- навігаційна панель, яка містить логотип сайту, посилання на головну сторінку, посилання на розміщення вакансії, посилання на реєстрацію та вхід;
- пошук вакансій, який складається з форми, де користувач може ввести запит, обрати місто та натиснути «Знайти» для пошуку вакансій. Зверху над цією формою виведено кількість знайдених вакансій.
- список вакансій, які було знайдено. Якщо користувач скористався фільтрами для пошуку, будуть присутні тільки ті вакансії, які відповідають цим фільтрам. У списку вакансій наведено спрощену версію вакансій, які мають посилання на їх повноцінну версію та посилання на компанію, за якою вони зареєстровані. Варто відзначити, що в випадку, коли користувач не увійшов в систему в якості кандидата, він не матиме можливості відгукуватися на вакансії;

– пошук за фільтрами, який містить різні чекбокси, дозволяючи користувачу шукати вакансії за категоріями, видом зайнятості, зарплатою та параметрами «без досвіду», «пенсіонер», «переселенець» та «студент».

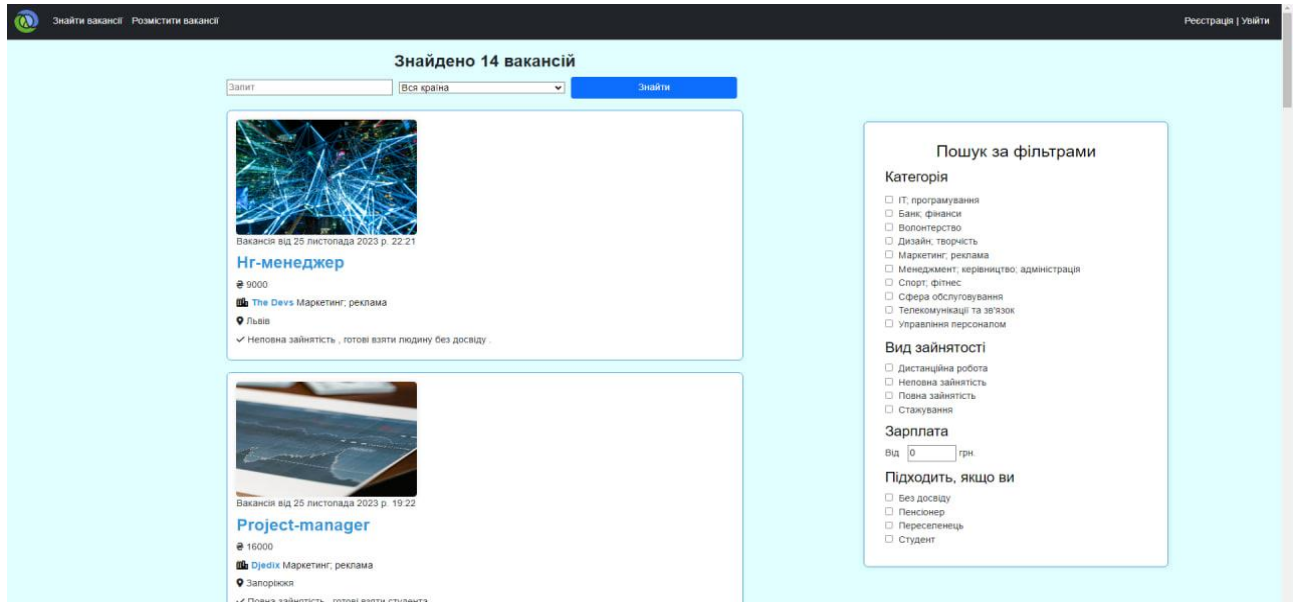


Рисунок 4.1 – Головна сторінка

Для повноцінної роботи з вебзастосунком потрібно зареєструватись та увійти до системи. На рис. 4.2 наведено сторінку реєстрації користувача. При реєстрації необхідно заповнити такі поля, як логін, електронна пошта, ім'я, прізвище, пароль та підтвердження пароля, після чого натиснути кнопку реєстрації.

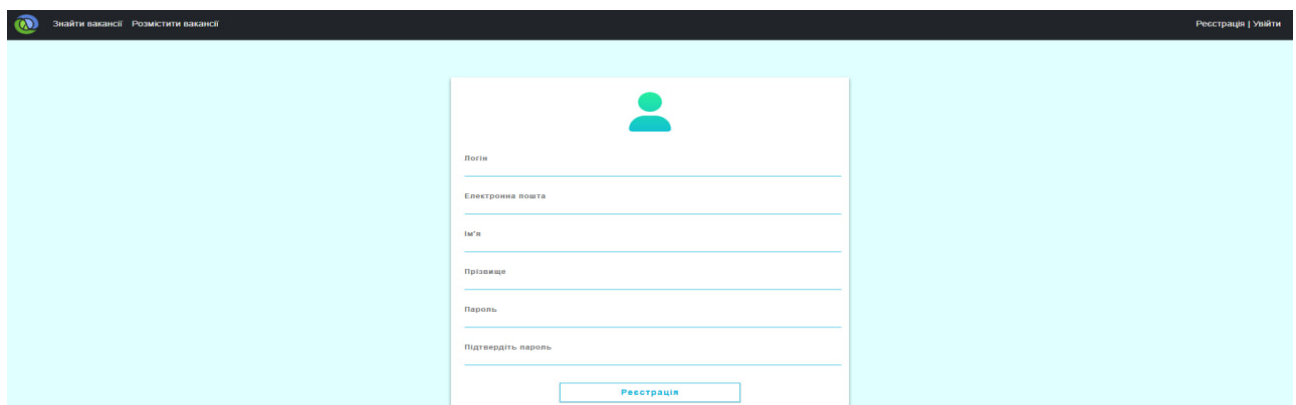


Рисунок 4.2 – Реєстрація користувача

Якщо всі поля було коректно заповнено, користувача буде перенаправлено на сторінку входу до системи (рис. 4.3). Користувач може увійти до системи як кандидат, або роботодавець, ввівши свій логін, пароль та натиснувши відповідну кнопку. Відповідно до натиснутої кнопки (вхід як кандидат, або як роботодавець) буде надано відповідний функціонал.

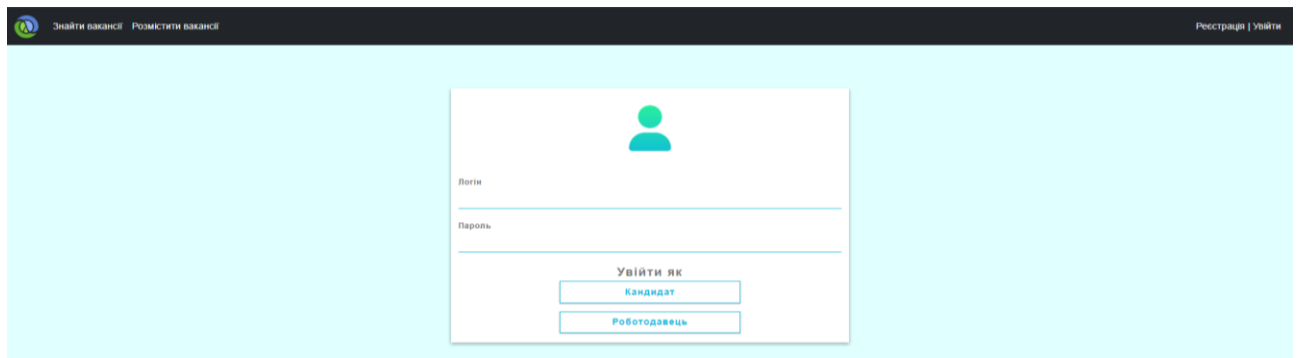


Рисунок 4.3 – Вхід до системи

Якщо користувач увійшов до системи як роботодавець, система одразу перенаправить його на сторінку реєстрації його компанії для подальшого керування нею (рис. 4.4). Для реєстрації компанії потрібно заповнити такі поля, як назва компанії, адреса її вебсайту, кількість працівників, опис компанії, місто, фізичну адресу компанії, логотип та категорію надання послуг.

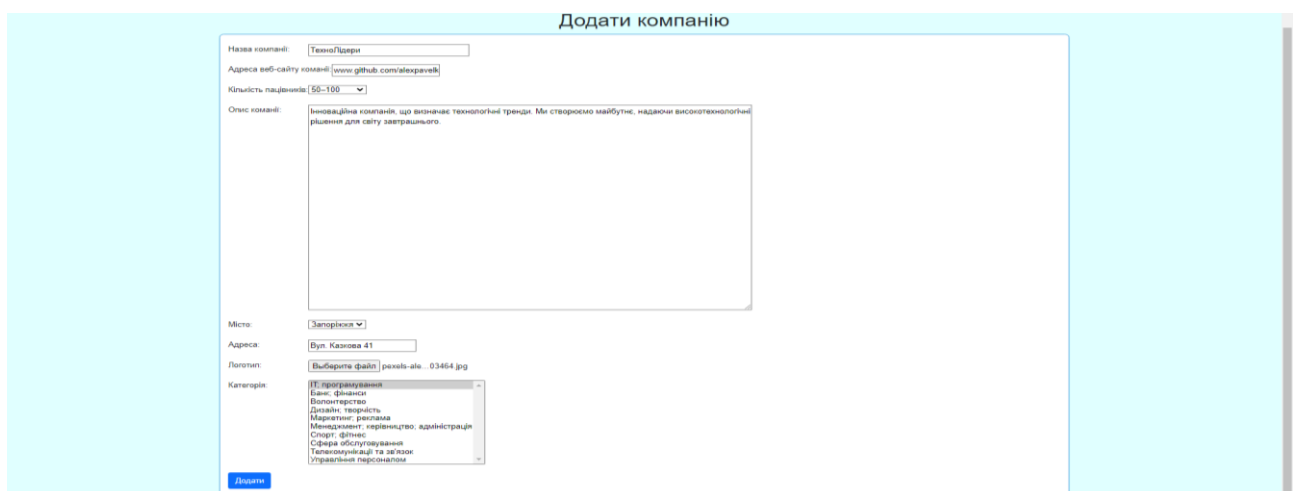


Рисунок 4.4 – Реєстрація компанії роботодавця

Після успішної реєстрації компанії, користувача буде перенаправлено на сторінку перегляду його компанії (рис. 4.5), де він зможе додавати контактні дані, редагувати опис компанії та додавати вакансії.

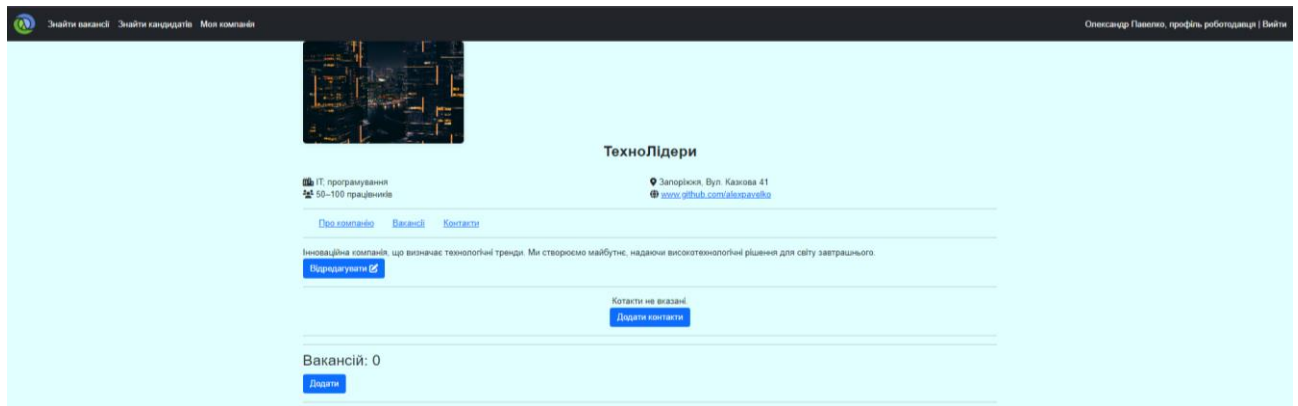


Рисунок 4.5 – Сторінка перегляду компанії роботодавця

Якщо користувач натисне кнопку додавання контактів на сторінці перегляду своєї компанії, він потрапить на сторінку додавання контактних даних для компанії (рис. 4.6), де може вказати свою контактну інформацію, або контактну інформацію людини, яка буде відповідати за комунікацію з кандидатами. До контактних даних компанії належить номер телефону, додатковий номер телефону, який є не обов'язковим та email. При натисканні кнопки редагування опису компанії відкривається сторінка з формою реєстрації компанії (рис. 4.4), яка вже заповнена даними компанії, де можна відредагувати будь-яке поле форми, включаючи опис.

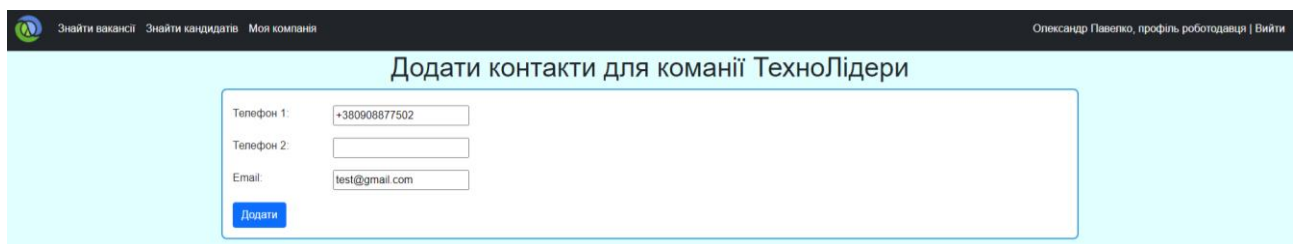


Рисунок 4.6 – Сторінка додавання даних для компанії роботодавця

Після додавання контактних даних для компанії, на сторінці перегляду компанії роботодавець може додавати вакансії до своєї компанії на сторінці додавання вакансії (рис. 4.7), де потрібно заповнити такі поля, як назва вакансії, опис вакансії, зарплата кандидата, тип зайнятості, типи кандидатів, освіта, досвід, категорія та обрати чи буде ця вакансія активною, натиснувши на відповідний чекбокс. Якщо чекбокс не було натиснуто, вакансія буде вважатися не активною, і вона не буде відображатися в списку вакансій.

The screenshot shows a web application interface for adding a job. The title is 'Додати вакансію'. The form contains the following fields and options:

- Назва вакансії:** A text input field containing 'Junior Python Developer'.
- Опис вакансії:** A large text area containing a description in Ukrainian: 'Виконає до вакансії: 1+ рік досвіду розробки (коммерційні або власні проекти). Впевнені знання мови Python. Розуміння RESTful API. Досвід роботи з фреймворками (FastAPI, Flask, Django). Базовий досвід роботи з реляційними (PostgreSQL, MySQL) та NoSQL (Cassandra) базами даних. Розуміння роботи з Git.'
- Зарплата (гривні):** A text input field containing '11000'.
- Тип зайнятості:** A dropdown menu with options: 'Дистанційна робота', 'Неповна зайнятість', 'Повна зайнятість', and 'Стандартна'.
- Типи кандидатів:** A dropdown menu with options: 'Всі досвід', 'Початковий', 'Посередній', and 'Старший'.
- Освіта:** A dropdown menu with the option 'Вакансія'.
- Досвід:** A dropdown menu with the option 'Після'.
- Категорія:** A dropdown menu with options: 'IT-розробка', 'Бач. бізнес', 'Бухгалтерство', 'Дизайн, творчість', 'Маркетинг, реклама', 'Менеджмент, керівництво, адміністрація', 'Служба безпеки', 'Сфера обслуговування', 'Телекомунікації та зв'язок', and 'Управління персоналом'.
- Активна вакансія:** A checkbox that is currently checked.
- Додати:** A blue button at the bottom left of the form.

Рисунок 4.7 – Сторінка додавання вакансії

Після додавання вакансії, система одразу звірить, чи немає таких кандидатів, у яких критерії бажаної вакансії такі самі, або схожі на параметри тієї, що була щойно додана. Якщо таких кандидатів було знайдено, система автоматично їх повідомить у встановлений проміжок часу повідомленням на електронну пошту (рис. 4.8). Таким проміжком було обрано опівдні. Таким чином усі кандидати отримають листа на електронну пошту про нові вакансії лише раз на день, за умови, що були додані нові вакансії, які відповідають критеріям бажаних вакансій кандидатів. Надалі планується розробка відключення надсилання сповіщень на електронну пошту, якщо кандидату це не

потрібно, а також встановлення іншого тимчасового проміжку, який буде зручним кандидатам.

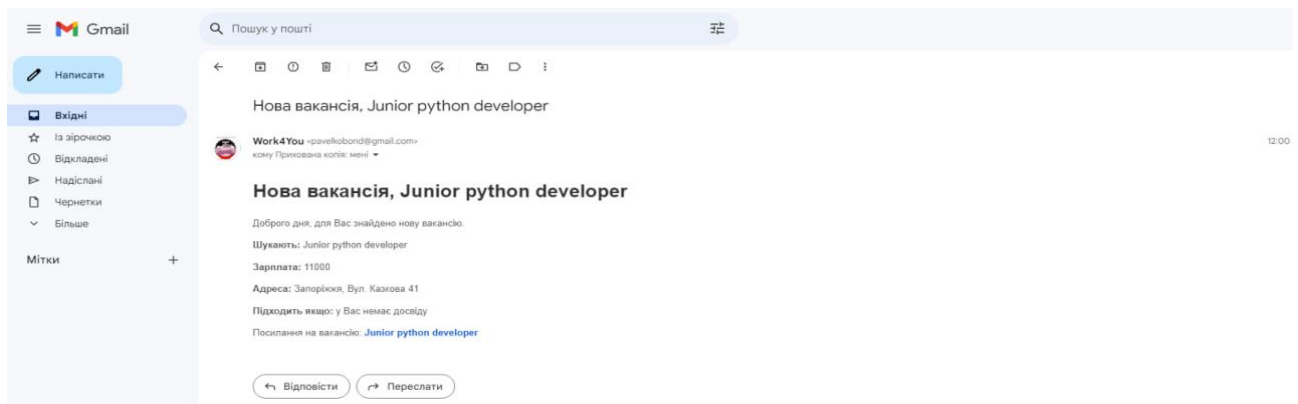


Рисунок 4.8 – Сповідження кандидата про нову вакансію

Після того, як роботодавець додав вакансію до своєї компанії, на сторінці перегляду його компанії (рис. 4.9) в списку вакансій для кожної з них йому буде надано наступні кнопки для керування нею:

- кнопка «Додавання тестування» відповідає за додавання тестування до вакансії;
- кнопка «Відгуки» відповідає за перегляд відгуків кандидатів на вакансію;
- кнопка «Редагування» при натисканні на неї перенаправить на сторінку з формою додавання вакансії (рис. 4.7), яка буде заповнена даними;
- кнопка «Видалити» при натисканні на неї видаляє вакансію;
- кнопка із зображенням ока відповідає за зміну активності вакансії та є чекбоксом. При натисканні на неї, зображення ока буде перекреслено, що вказує на неактивність вакансії. У цьому випадку вакансія не буде відображатися в списку вакансій для кандидатів.

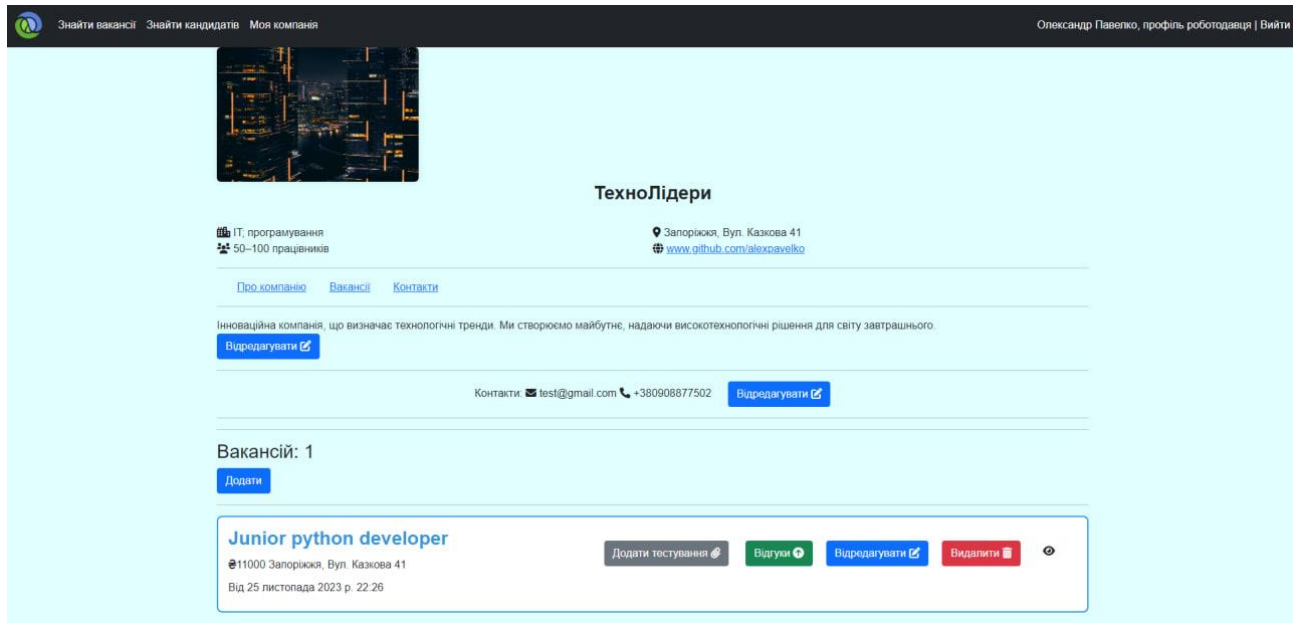


Рисунок 4.9 – Сторінка перегляду компанії роботодавця після додавання вакансії

Якщо роботодавець на сторінці перегляду його компанії натисне кнопку «Додавання тестування», система перенаправить його на сторінку з формою додавання тестування до вакансії (рис. 4.10). Роботодавець може додавати запитання, відповіді на них та встановлювати правильну відповідь та максимально дозволений час, за який може бути пройдено тест з моменту його початку кандидатом. Максимально можна додати двадцять запитань та чотири варіанти відповіді для кожного з них. Правильна відповідь на запитання може бути лише одна.

При натисканні на кнопку «Відгуки», система перенаправить роботодавця на сторінку перегляду відгуків кандидатів на вакансію (рис. 4.11), де наведено інформацію про всі відгуки кандидатів у вигляді таблиці, в якій містяться такі дані про кандидатів, як посилання на його профіль, дата відгуку, його файл резюме (який можна завантажити та подивитися), статус відгуку, можливі дії та оцінки з тестувань, якщо вони проводились. Існують такі статуси відгуків, як «розглядається», «прийнято» та «відхилено». Роботодавець може обрати статус кандидата згідно своїх уподобань. Якщо було обрано статус «прийнято»,

контактні дані сторін стануть доступні для перегляду обом сторонам. В стовпці можливих дій роботодавець може провести тестування кандидата, відправивши йому тест, який був попередньо доданий до вакансії. Після проходження тесту кандидатом в таблиці відгуків у відповідному стовпці з'явиться результат тестування. Цей результат може бути важливим фактором для роботодавця, оскільки він відображає рівень знань кандидата. Кандидат може пройти тестування лише один раз, після чого кнопка «Провести тестування» на сторінці перегляду відгуків кандидатів стане неактивною для цього кандидата. Кандидат може пройти тестування якщо вакансія активна та не видалена. Також в стовпці можливих дій роботодавець може відповісти кандидату текстовим повідомленням (рис. 4.12), де може вказати подальші дії для кандидата або причину відмови. Відповідь може бути надіслана роботодавцем лише один раз, після чого зможе тільки переглядати її. Після надсилання відповіді роботодавцем, кандидат зможе побачити її.

Знайти вакансії Знайти кандидатів Моя компанія Олександр Павелко, профіль роботодавця Вийти

Додавання тестування до вакансії

Junior Python Developer

Запитання 1:

Що робить функція lambda в Python?

Варіант відповіді 1:

Створює анонімну функцію

Варіант відповіді 2:

Визначає глобальну змінну

Варіант відповіді 3:

Реалізує умовні вирази

Додати варіант відповіді

Правильна відповідь: 1

Запитання 2:

Які типи даних є immutable в Python?

Варіант відповіді 1:

list та set

Варіант відповіді 2:

tuple та int

Рисунок 4.10 – Сторінка додавання тестування до вакансії

Відгуки на вакансію Junior Python Developer					
Кандидат	Дата відгуку	Резюме	Статус	Дії	Оцінка з тестування
Костенко Іван	26.11.2023	kostenko.pdf	відхилено	Провести тестування Відправити відповідь	11/20
Петренко Марія	26.11.2023	petrenko.pdf	прийнято	Провести тестування Відправити відповідь	18/20
Іванов Петро	27.11.2023	ivanov.pdf	розглядається	Провести тестування Відправити відповідь	Тестування не проводилось
Сидоренко Олена	27.11.2023	sidorenko.pdf	розглядається	Провести тестування Відправити відповідь	Тестування не проводилось
Михайленко Володимир	27.11.2023	mihailenko.pdf	відхилено	Провести тестування Відправити відповідь	Тестування не проводилось

Рисунок 4.11 – Сторінка перегляду відгуків кандидатів на вакансію

Знайти вакансії

Знайти кандидатів

Моя компанія

Олександр Павленко, профіль роботодавця | Вийти

Відповідь кандидату [Петренко Марія](#) щодо вакансії [Junior Python Developer](#)

Супровідний лист кандидата

26.11.2023

ТехноПідери

Шановний Павленко Олександр!

Я - Петренко Марія, висококваліфікований кандидат на посаду Junior Python Developer, із великим інтересом подаю свою заяву на вакансію, яку ви оголосили.

Чому саме я:

- Маю вищу освіту в галузі інформаційних технологій, а також успішний досвід роботи з Python.
- Маю глибокі знання Python та його екосистеми, включаючи Django та Flask. Маю досвід в розробці зручних та ефективних веб-додатків.
- Здатна працювати в команді, ефективно спілкуватися та ділитися ідеями, шукаю можливості для постійного вдосконалення.

Чому я хочу приєднатися до вашої команди:

Ваше повідомлення кандидату

Ваше повідомлення кандидату...

Відправити відповідь

Рисунок 4.12 – Сторінка надання роботодавцем відповіді на відгук кандидата

Окрім перерахованого вище функціоналу, у навігаційній панелі є посилання «Знайти вакансії». Якщо роботодавець перейде за ним, система запропонує йому авторизуватися в якості кандидата. Також у навігаційній панелі роботодавцям доступний пошук кандидатів за допомогою посилання «Знайти кандидатів». Це посилання відкриває сторінку пошуку кандидатів (рис. 4.13), яка

схожа на сторінку перегляду вакансій, але замість вакансій тут наведена інформація про кандидатів. Ця сторінка містить наступні елементи:

- блок пошуку, який складається із запити, вибору міста та кнопки знайти;
- блок списку кандидатів, які були знайдені згідно з фільтрами до пошуку;
- блок пошуку за фільтрами, який містить фільтри освіти, досвіду та бажаної заробітної плати кандидата;
- графік, який в процентному співвідношенні показує яка у кандидатів освіта. Блок в якому знаходиться цей графік містить посилання на сторінку графіків (рис. 4.14), яка містить інформацію про кандидатів.

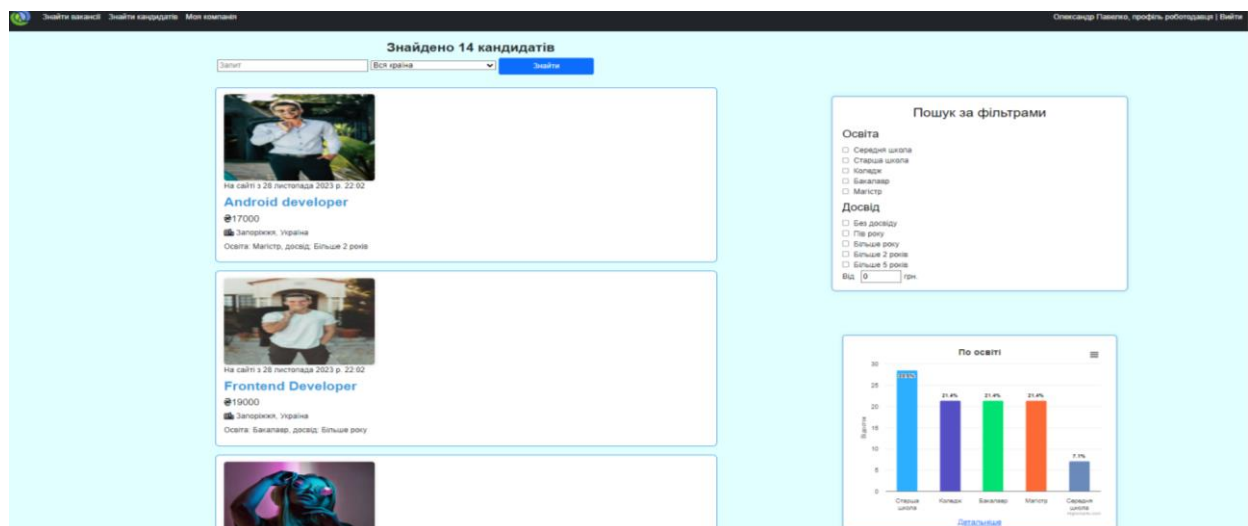


Рисунок 4.13 – Сторінка пошуку кандидатів

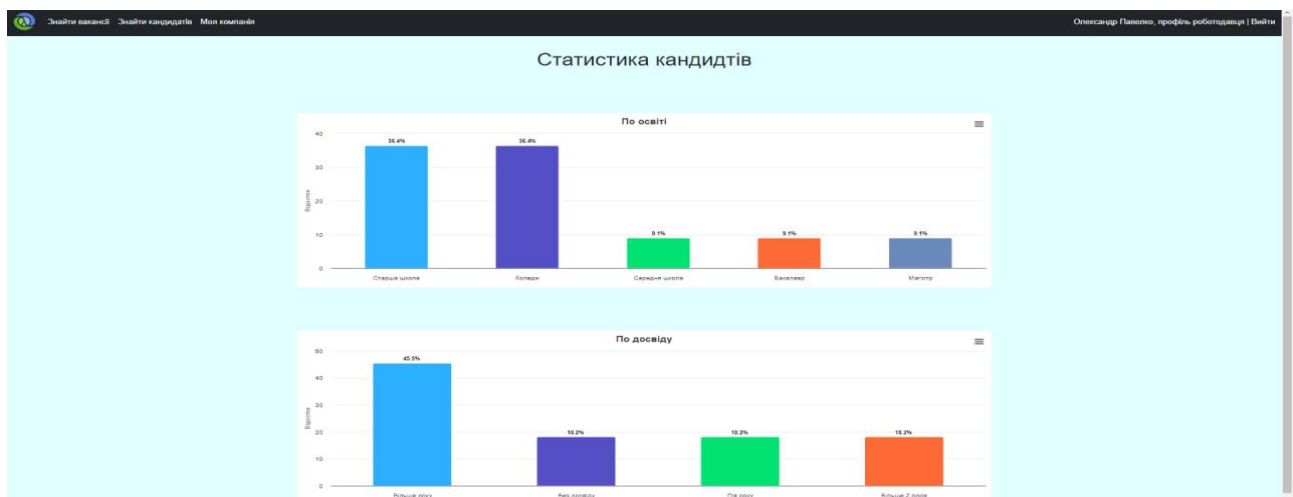


Рисунок 4.14 – Сторінка графіків з інформацією про кандидатів

Роботодавець може подивитись детальні дані кандидата, який зацікавив його, натиснувши на сторінці пошуку кандидатів (рис. 4.13), на посилання бажаної вакансії кандидата, яке перенаправить роботодавця на сторінку перегляду детальної інформації про кандидата. Якщо цей кандидат ще не відгукувався на вакансії роботодавця та не був схвалений ним, його контактна інформація буде недоступна (рис. 4.15).



Рисунок 4.15 – Сторінка детальної інформації кандидата

Якщо кандидат вже відгукувався на вакансії роботодавця та був схвалений ним, роботодавцю на сторінці перегляду детальної інформації про кандидата буде доступна його контактна інформація (рис. 4.16).

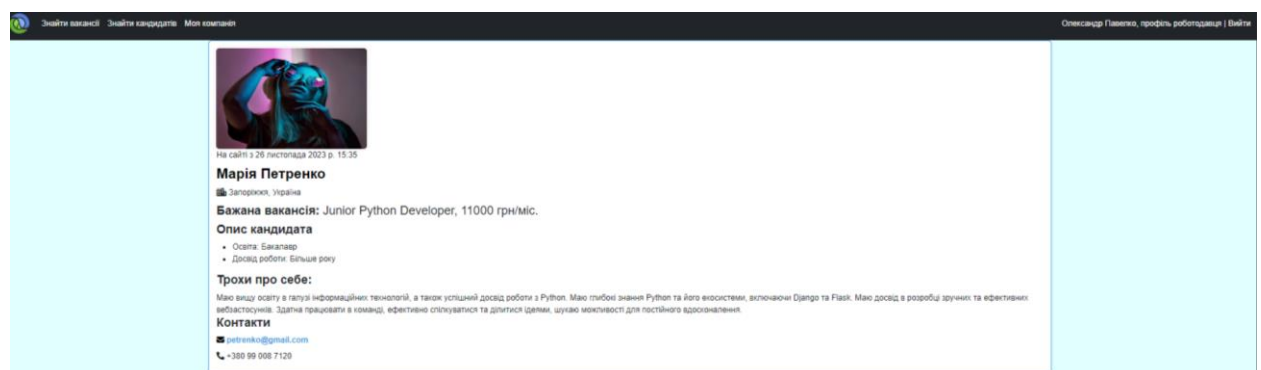


Рисунок 4.16 – Сторінка детальної інформації кандидата з доступною контактною інформацією

Вище було наведено роботу роботодавця з системою. Розглянемо роботу кандидата з системою. Після входу в систему користувачем в якості кандидата,

його буде перенаправлено на сторінку додавання даних кандидата (рис. 4.17). На цій сторінці потрібно заповнити форму з такими полями, як номер телефону, освіта, досвід, фото, бажана вакансія, бажана заробітна плата, невеликий опис своїх навичок (поле «Трохи про себе») та місто проживання. Заповнення цих даних дозволить роботодавцям дізнатися більше про кандидата.

Рисунок 4.17 – Сторінка додавання даних кандидата

Після додавання своїх даних, кандидата буде перенаправлено на сторінку перегляду його профіля (рис. 4.18). При використанні цього профіля в якості кандидата в навігаційній панелі на будь якій сторінці буде доступні такі посилання, як «Знайти вакансії» та «Мої відгуки», а також список, що розкривається при натисканні на своє ім'я та прізвище. В цьому списку будуть доступні такі посилання як «Переглянути мій профіль», яке перенаправить користувача на сторінку перегляду свого профілю (рис. 4.18), «Редагувати профіль», яке відкриє сторінку додавання даних кандидата (рис. 4.17), яка вже буде заповнена даними, та «Вийти», яке дозволить користувачу вийти зі свого профілю кандидата та увійти як роботодавець, або увійти в інший обліковий запис на сторінці входу в систему (рис. 4.3).



Рисунок 4.18 – Сторінка перегляду даних кандидата

Якщо кандидат натисне на посилання «Знайти вакансії», його буде перенаправлено на сторінку пошуку вакансій (рис. 4.19). Після входження в систему користувача в якості кандидата він матиме доступ не лише до функціоналу пошуку вакансій (рис. 4.1), але також до можливості відгукуватися на них.

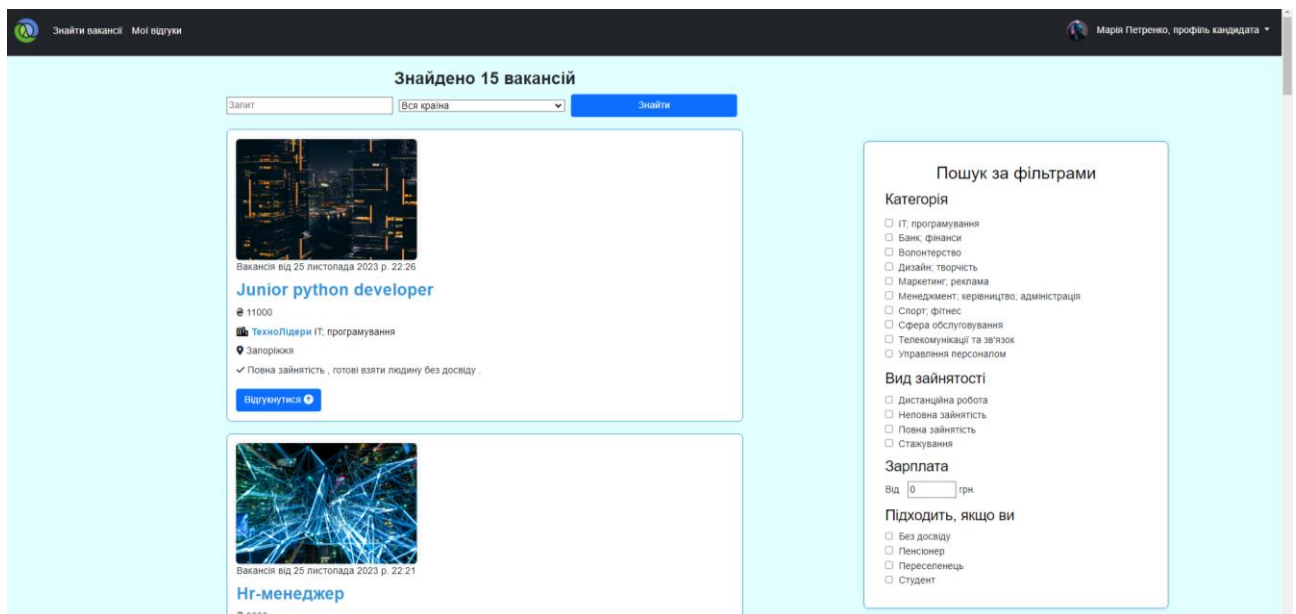


Рисунок 4.19 – Сторінка перегляду вакансій з можливістю відгукуватися на них

На цій сторінці нижче зображено графік (рис. 4.20), який відображає медіанну зарплату для п'яти вакансій з найвищою зарплатою та містить посилання на сторінку детальної статистики щодо вакансій (рис. 4.21, 4.22).



Рисунок 4.20 – Графік медіанної зарплати найпопулярніших вакансій з найвищою зарплатою

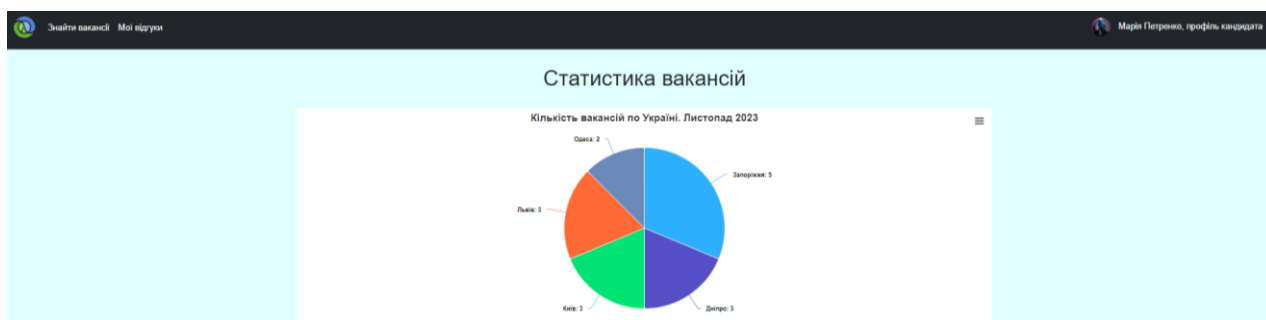


Рисунок 4.21 – Графік відображення кількості вакансій по містах

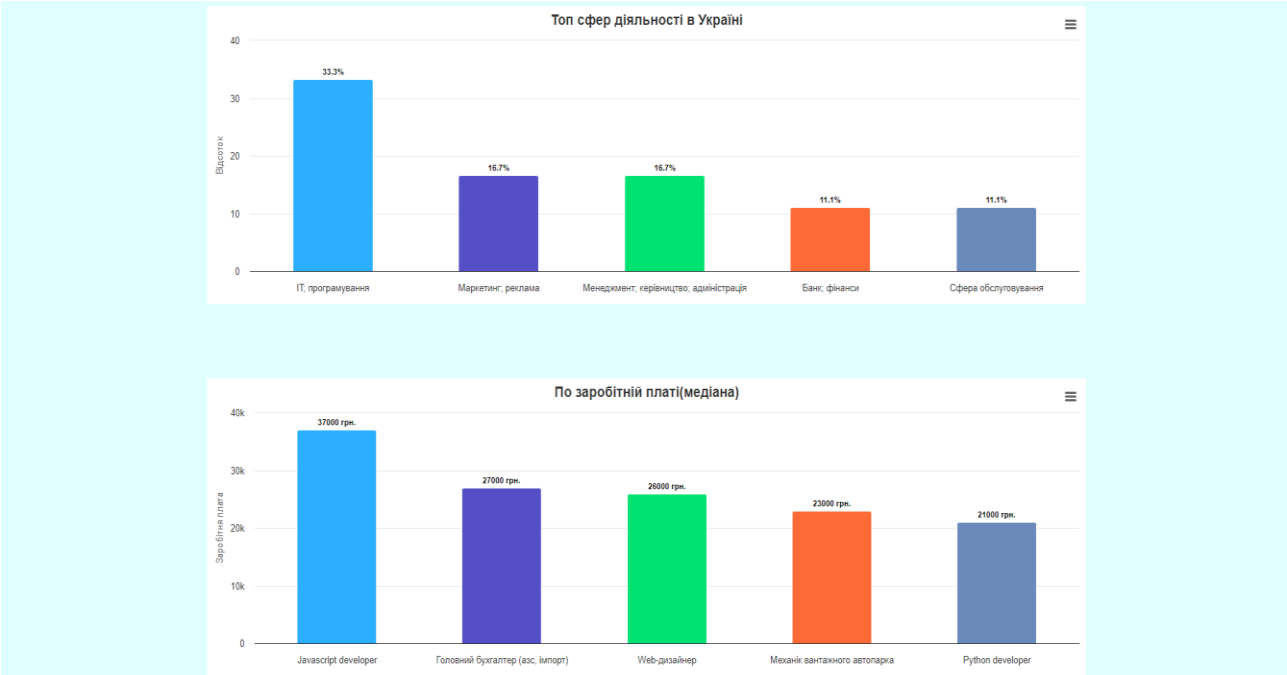


Рисунок 4.22 – Графіки відображення найпопулярніших сфер діяльності та медіанної зарплати найпопулярніших вакансій з найвищою зарплатою

Кандидат може скористатися пошуком вакансій за фільтрами на головній сторінці (рис. 4.23). Фільтри допомагають кандидату шукати вакансії за категоріями, видом зайнятості, зарплатою та параметрами «без досвіду», «пенсіонер», «переселенець» та «студент».

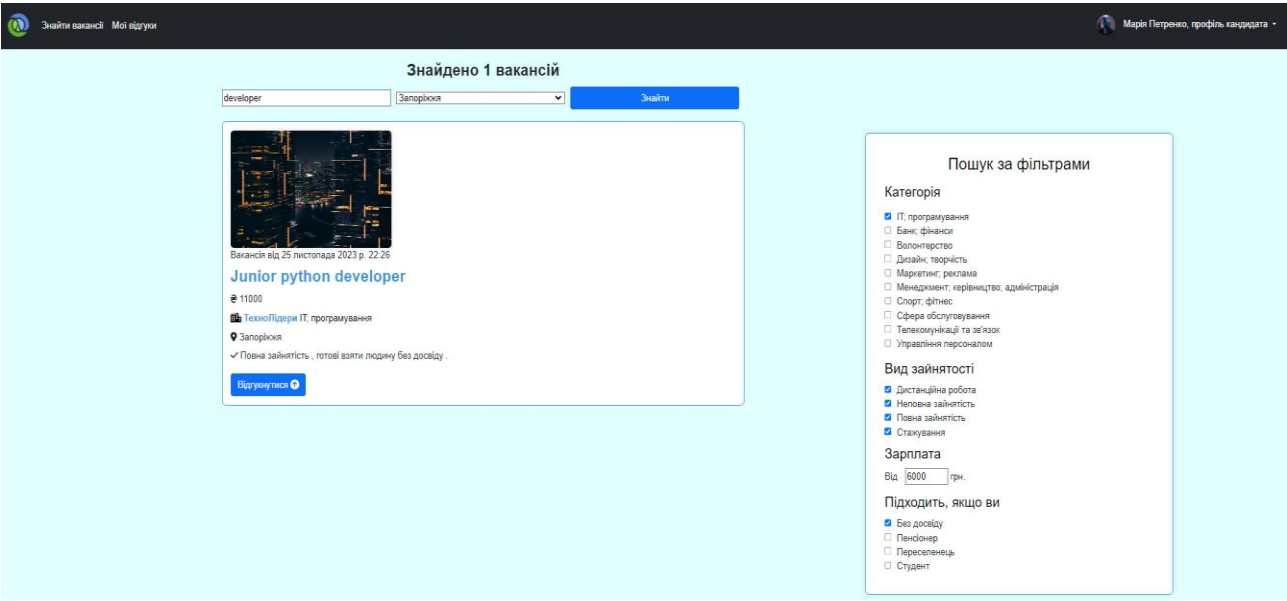


Рисунок 4.23 – Пошук вакансій за фільтрами

Після пошуку вакансій, якщо кандидата зацікавила вакансія, він може подивитися інформацію про компанію, до якої належить вакансія (рис. 4.24), натиснувши на посилання з назвою компанії, або подивитися більш детальну інформацію про вакансію (рис. 4.25, 4.26), натиснувши на посилання з назвою вакансії. Варто зазначити, що кожна вакансія містить контакти компанії, які будуть відображені кандидату тільки в разі якщо він відгукнувся на будь-яку вакансію компанії, пов'язану з цією вакансією, після чого був схвалений роботодавцем.

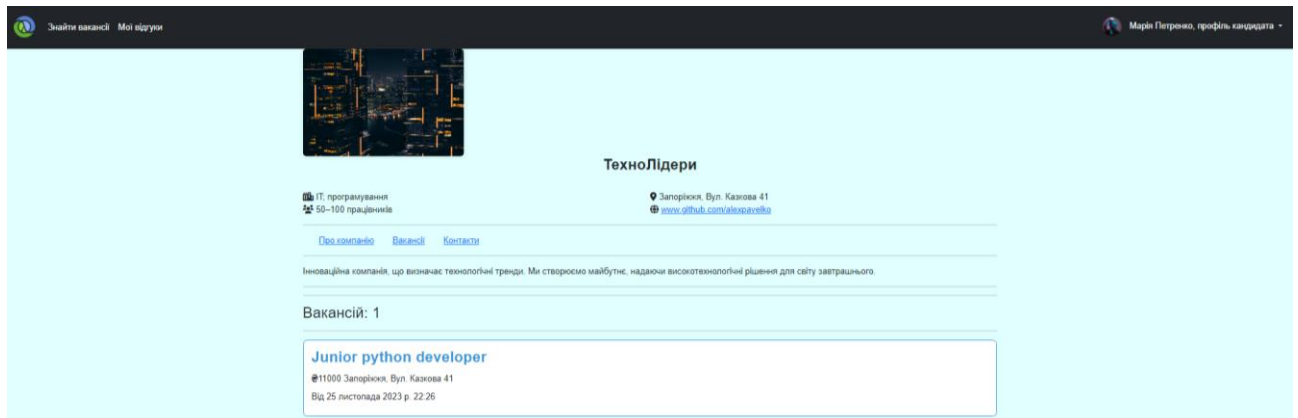


Рисунок 4.24 – Компанія, до якої відноситься вакансія

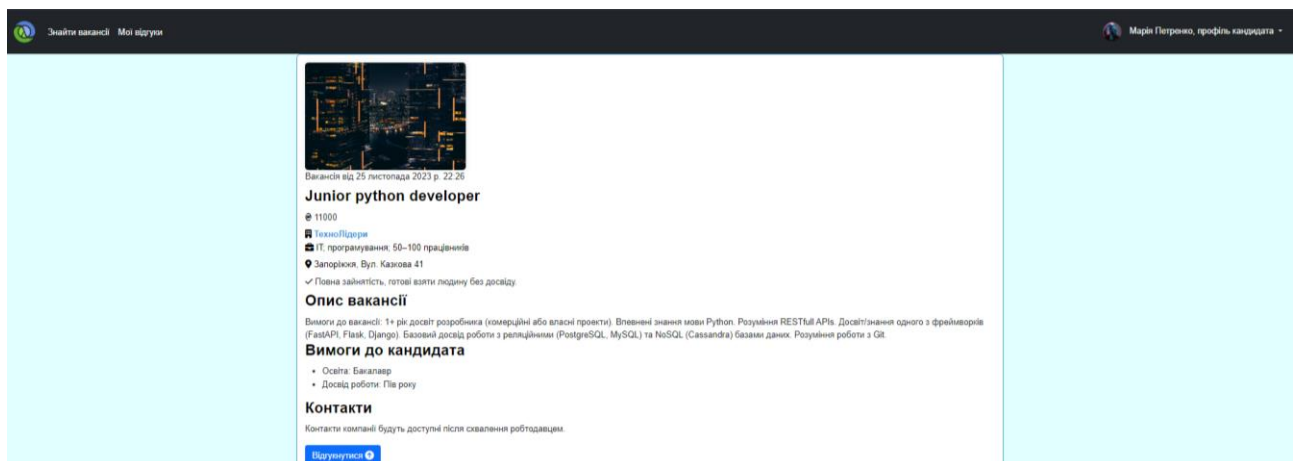


Рисунок 4.25 – Перегляд детальної інформації про вакансію кандидатом, який не схвалений роботодавцем

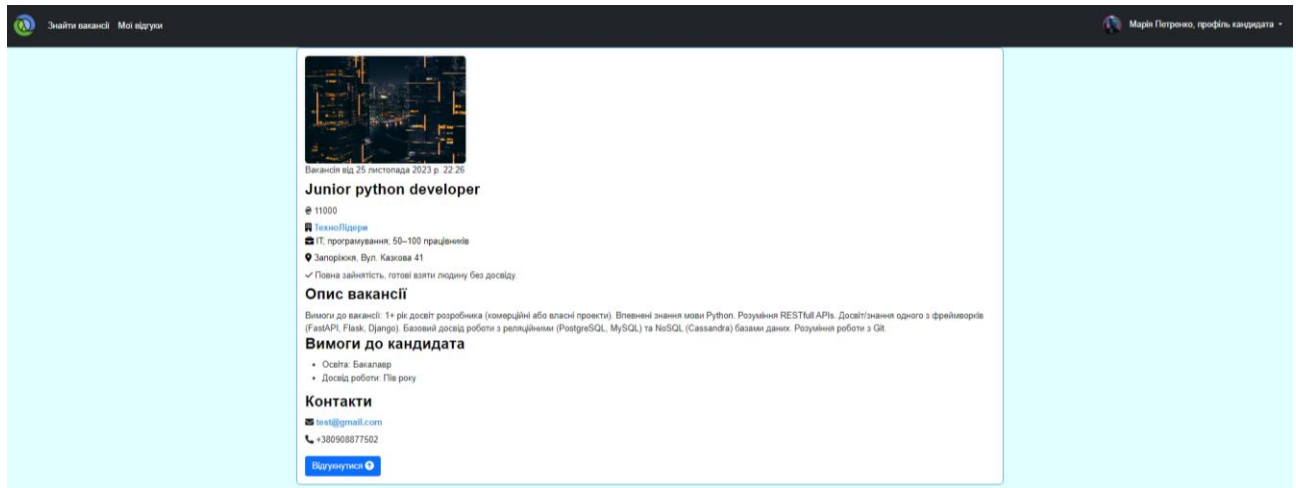


Рисунок 4.26 – Перегляд детальної інформації про вакансію кандидатом, який схвалений роботодавцем

Якщо кандидат натисне кнопку «Відгукнутися» на сторінці перегляду вакансії, система перенаправить його на сторінку з формою відгуку на вакансію (рис. 4.27), де йому потрібно надати супровідний лист та прикріпити файл резюме (файл резюме має бути у форматі .docx, .doc або .pdf). Після цього він має очікувати на відповідь роботодавця, яка відобразиться на цій сторінці у блоці «Відповідь роботодавця», якщо роботодавець її відправить.

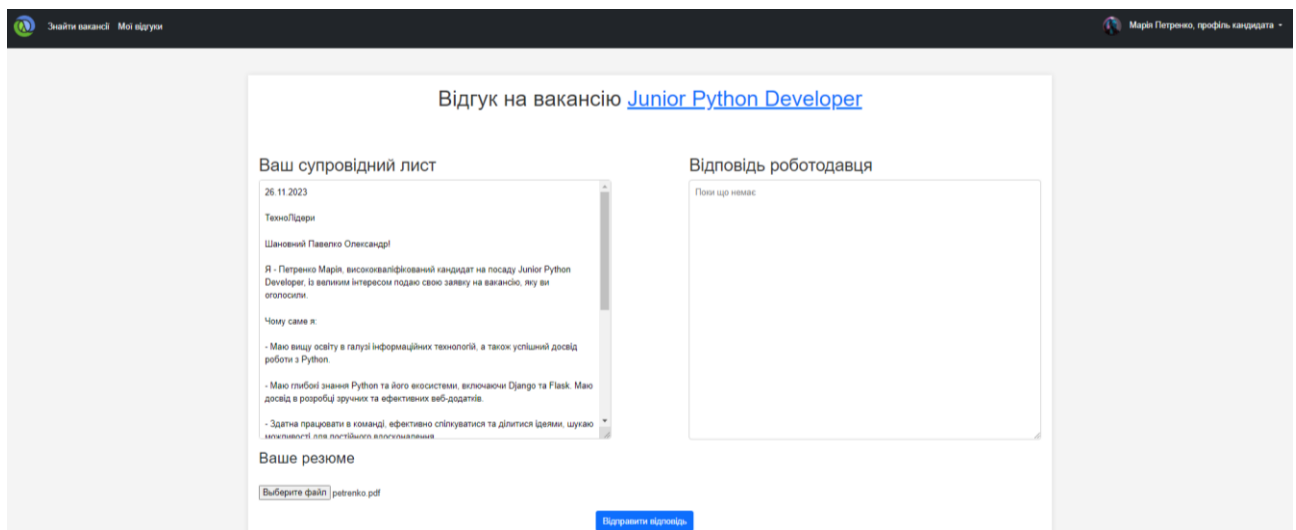
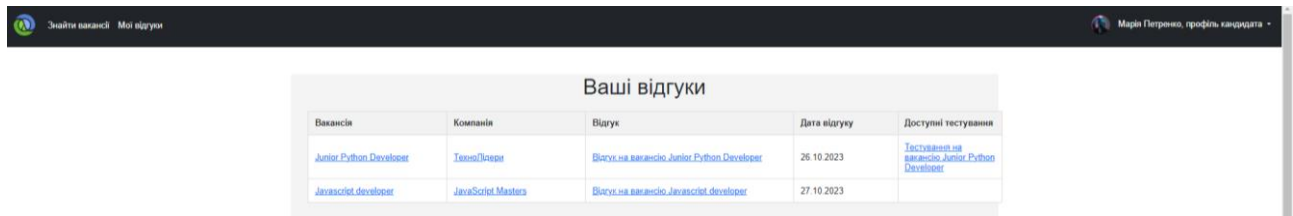


Рисунок 4.27 – Відправка відгуку на вакансію

Кандидат може переглядати всі свої відгуки за допомогою посилання «Мої відгуки» в навігаційній панелі. На рис. 4.28 наведено список відгуків кандидата на вакансії.



Вакансія	Компанія	Відгук	Дата відгуку	Доступні тестування
Junior Python Developer	Tosens/Океан	Відгук на вакансію Junior Python Developer	26.10.2023	Тестування на вакансію Junior Python Developer
JavaScript developer	JavaScript Masters	Відгук на вакансію JavaScript developer	27.10.2023	

Рисунок 4.28 – Список відгуків кандидата на вакансії

Список відгуків кандидата на вакансії представлений у вигляді таблиці, яка відображає наступні дані щодо відгуків на вакансії:

- посилання на вакансію, на яку відгукнувся кандидат;
- посилання на компанію, за якою закріплена вакансія;
- посилання на відгук, яке відкриває сторінку відгуку, яка містить супровідний лист та резюме кандидата;
- дата відгуку;
- посилання на тестування, якщо роботодавець додав його для цього кандидата.

Якщо кандидат перейде по посиланню на тестування, він потрапить на сторінку проходження тестування (рис. 4.29), де йому потрібно обрати правильні відповіді на запитання за встановлений проміжок часу, який встановив роботодавець при створенні тесту. Результат тестування зможе побачити тільки роботодавець на сторінці відгуків кандидатів на його вакансії.

Знайти вакансії Мої відгуки

Марія Петренко, профіль кандидата

Проходження тестування до вакансії Junior Python Developer

Залишилось часу 27:45

- Що робить функція lambda в Python?
 - ☒ a) Створює анонімну функцію
 - ☐ b) Визначає глобальну змінну
 - ☐ c) Реалізує умовні вирази
 Ваша відповідь: a)
- Які типи даних є immutable в Python?
 - ☐ a) list та set
 - ☐ b) tuple та int
 - ☒ c) dict та str
 - ☐ d) float та bool
 Ваша відповідь: c)
- Що таке рекурсія в програмуванні?
 - ☒ a) Одна з колекційних структур даних
 - ☐ b) Процес, коли функція викликає саму себе
 - ☐ c) Спосіб сортування даних
 - ☐ d) Вбудована функція для зчитування файлів
 Ваша відповідь: a)
- Якість коду можна підвищити за допомогою PEP 8. Що це таке?

Рисунок 4.29 – Проходження тестування кандидатом

Для керування системою передбачено роботу адміністратора, який має повний доступ до системи та її даних. Для зручного керування системою, адміністративну панель було поділено на блоки. Зліва можна побачити всі сутності, які присутні в проєкті. Якщо натиснути на будь-яку з них, в центральному блоку можна буде побачити всі екземпляри цієї сутності, а зверху будуть доступні дії з ними. Блок праворуч сторінки містить можливості для фільтрування за параметрами. На рис. 4.30 та рис. 4.31 наведено приклади роботи адміністратора з адміністративною панеллю.

Django адміністрування

Вітаємо, BOOF, дивіться сайт / змінити пароль / вийти

Домівка - WorkYouApp - Вакансії

Виберіть Вакансії щоб змінити

Дік.	Вибрати обрані Вакансії	КОМПАНІЯ	ДАТА ДОДАВАННЯ	ЗАРПЛАТА (грн/міс.)
56	Junior python developer	ТехноЛідери	25 листопада 2023 р. 22:26	11000
49	ІТ-менеджер	The Devs	25 листопада 2023 р. 22:21	9000
48	Project manager	DjediX	25 листопада 2023 р. 19:22	16000
47	Фейсф	123212	22 листопада 2023 р. 16:45	1111
13	Web-дизайнер	Tech	02 серпня 2022 р. 14:12	26000
7	Механік вантажного автопарка	AutoCar	02 серпня 2022 р. 00:50	23000
9	Продавець-консультант	Tech	02 серпня 2022 р. 00:50	12000
10	Проектний менеджер	Tech	02 серпня 2022 р. 00:50	19000
12	Project-менеджер	JavaScript Masters	02 серпня 2022 р. 00:50	12000
11	JavaScript developer	JavaScript Masters	02 серпня 2022 р. 00:50	37000
8	Головний бухгалтер (ах, бухгалтер)	AutoCar	02 серпня 2022 р. 00:50	27000
6	Project manager	SkyNet	01 серпня 2022 р. 23:50	13000
5	С/я розробник	SkyNet	01 серпня 2022 р. 23:50	17000
3	Java developer	CompX	01 серпня 2022 р. 22:47	19000
2	Python developer	CompX	01 серпня 2022 р. 22:47	21000
1	Рекрутер, ІТ-менеджер	CompX	01 серпня 2022 р. 22:32	14000

16 Вакансій

ВІДФИЛЬТРУВАТИ

За Зарплатою (грн/міс.)

Всі

1111

9000

11000

12000

13000

14000

16000

17000

19000

21000

23000

26000

27000

37000

За Датою додавання

Будь-яка дата

Сьогодні

Останні 7 днів

Цього місяця

Цього року

Рисунок 4.30 – Перегляд адміністратором вакансій

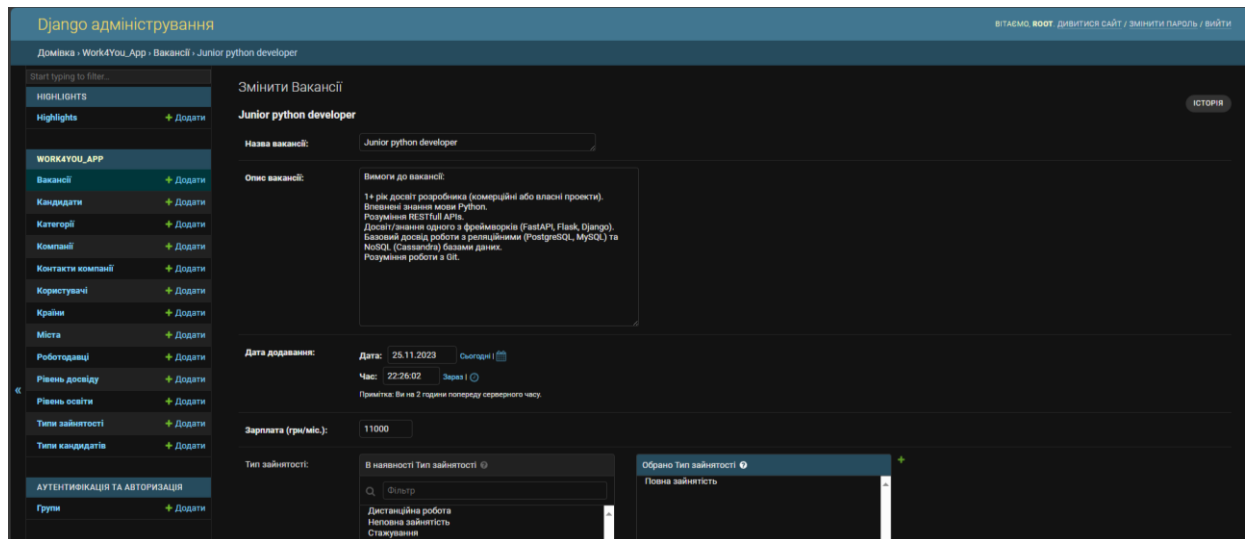


Рисунок 4.31 – Редагування адміністратором даних вакансії

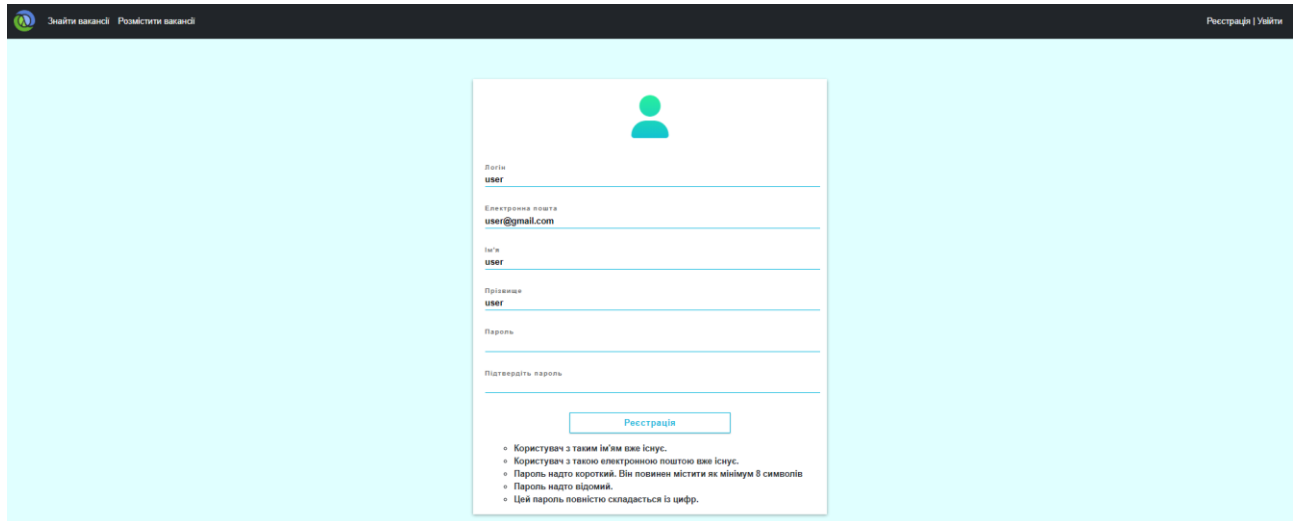
4.2 Тестування програми

Для розробленого вебзастосунку було проведене повноцінне тестування, яке включає повну перевірку роботи системи. Під час перевірки придатності до використання, виявленню, що вебзастосунок має зручний користувацький інтерфейс, всі вказані функціональні вимоги були реалізовані, після чого були проведені тести працездатності реєстрації, входу в систему, валідації форм, пошуку вакансій за фільтрами, додавання файлу резюме, відправки на електронну пошту листів з великою кількістю вакансій, оскільки саме ці функції можуть вплинути на коректну роботу системи.

Робота з системою передбачає роботу з даними, які вводить користувач. Для надійної роботи системи потрібно щоб ці дані були коректними та не створювали ситуації, які можуть призвести до появи помилок та зупинки роботи системи.

Реєстрація користувача включає в себе такі поля, як логін, електронна пошта, ім'я, прізвище, пароль та підтвердження паролю. На рис. 4.32 наведено невдалу спробу реєстрації, оскільки користувач з таким логіном, електронною

поштою вже існує, а пароль занадто легкий, оскільки складається лише з п'яти цифр.



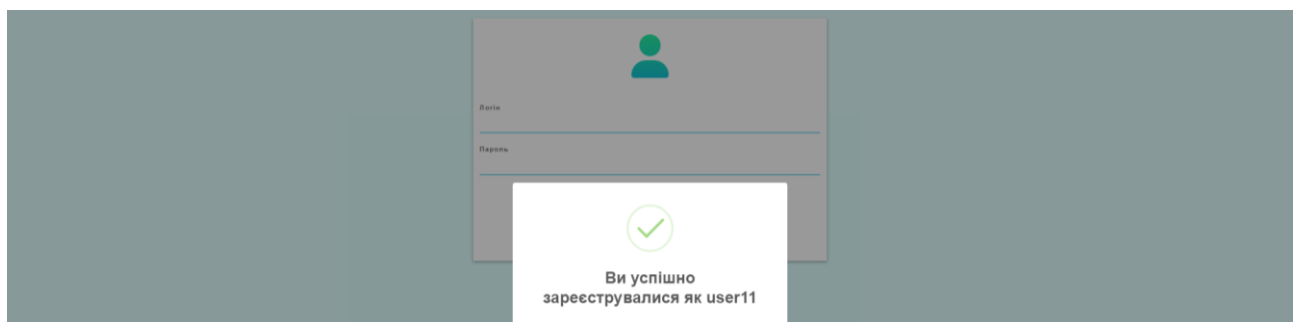
The screenshot shows a registration form on a light blue background. The form has a header with a green user icon. Below the header, there are input fields for 'Логін' (Login) with the value 'user', 'Електронна пошта' (Email) with the value 'user@gmail.com', 'Ім'я' (Name) with the value 'user', 'Прізвище' (Surname) with the value 'user', and 'Пароль' (Password). Below the password field is a 'Підтвердіть пароль' (Confirm password) field. A 'Реєстрація' (Registration) button is located below the form. Below the button, there are four error messages in a list:

- Користувач з таким ім'ям вже існує.
- Користувач з такою електронною поштою вже існує.
- Пароль надто короткий. Він повинен містити як мінімум 8 символів
- Пароль надто відомий.
- Цей пароль повністю складається із цифр.

Рисунок 4.32 – Невдала спроба реєстрації користувача

Логін та електронна пошта мають бути унікальними, оскільки вони дозволяють ідентифікувати конкретного користувача в системі, а пароль має бути достатньо складним, що зменшує ймовірність того, що невідомі особи зможуть легко здобути доступ до облікового запису.

На рис. 4.33 наведено приклад вдалої спроби реєстрації користувача. Всі поля було наповнено валідними даними, після чого була натиснута кнопка «Реєстрація». Дані було перевірено на валідність, і користувача було перенаправлено на сторінку входу в систему, після чого з'явилось віконце про успішну реєстрацію.



The screenshot shows the same registration form as in Figure 4.32, but with a success message displayed in a white box with a green checkmark icon. The message reads: 'Ви успішно зареєструвалися як user11'.

Рисунок 4.33 – Вдала спроба реєстрації користувача

Після реєстрації користувача було здійснено спробу входу в систему з неправильним паролем (рис. 4.34). При спробі ввести неправильний логін та правильний пароль – результат той самий. Варто відзначити, що після невдалої спроби входу в систему, поля логіну та паролю не відображають дані, що були введені, що зменшує шанси невідомих осіб швидко здобути доступ до облікового запису. Можна помітити, що для користувачів, які забули свій пароль, немає можливості його відновлення. Проект буде розвиватися і цей функціонал буде реалізовано пізніше.

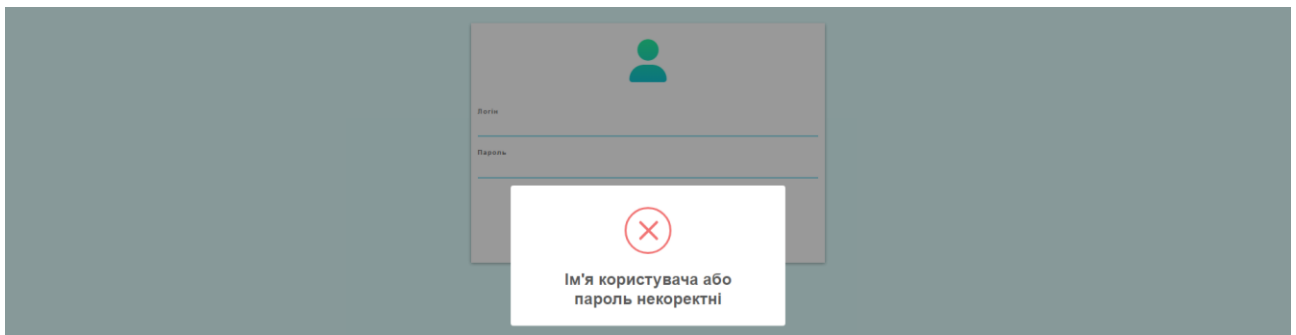


Рисунок 4.34 – Невдала спроба входу в систему користувачем

На рис 4.35 наведено вдалу спробу входу в систему з правильними даними облікового запису.

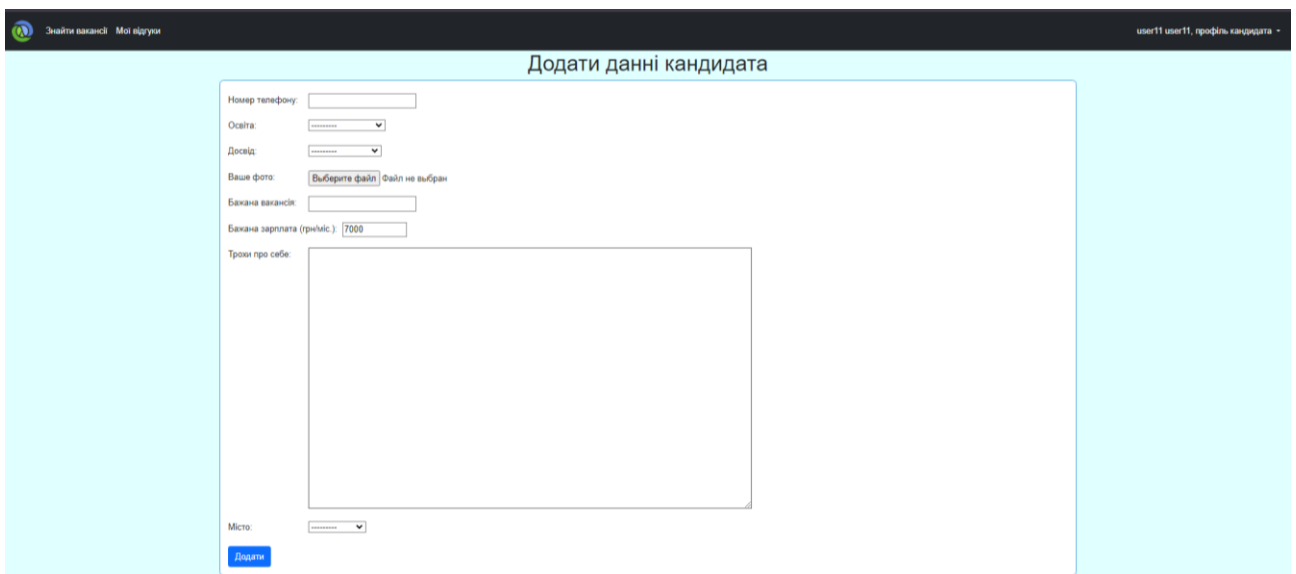


Рисунок 4.35 – Вдала спроба входу в якості кандидата

Користувач увійшов в систему як кандидат, тому одразу потрапив на сторінку додавання своїх даних. Якщо кандидат з введеними даними увійшов би як роботодавець, він потрапив би на сторінку реєстрації своєї компанії.

При додаванні або редагуванні будь-яких даних на будь-якій формі працює валідація даних. Форма не може бути надіслана пустою, або якщо якесь поле було не введено, або було введено некоректно. На рис. 4.36 наведено приклад роботи валідації форми.

Рисунок 4.36 – Робота валідації форми

При пошуку вакансій за фільтрами передбачено ситуацію коли не буде знайдено жодної вакансії (рис. 4.37).

Рисунок 4.37 – Пошук вакансій за фільтрами коли не знайдено вакансії

З точки зору безпеки важливо визначити які самі файли можуть бути відправлені в якості файлу резюме кандидата на сторінці відгуку на вакансію. Файли мають різні розширення, та деякі з них можуть нанести шкоду комп'ютеру роботодавця, якщо він відкриє такі файли. Саме тому було обрано тільки файли з розширенням .docx, doc та .pdf. Ці файли не можуть виконувати шкідливі дії, але можуть відображати текст та зображення, чого буде достатньо для резюме кандидата. На рис. 4.38 наведено спробу додавання файлу резюме, на якому видно які саме файли можуть бути додані.

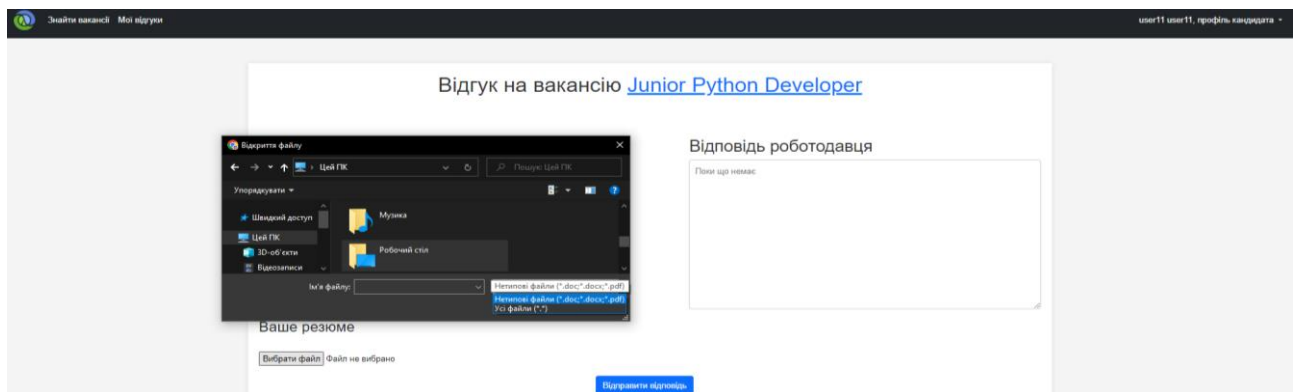


Рисунок 4.38 – Спроба додавання файлу резюме

Відправка сповіщень на електронну пошту грає велику роль та потребує уваги. Система відправляє сповіщення на електронну пошту в один і той самий час кожного дня, а саме опівдні. Якщо з минулого надсилання сповіщень система знайшла лише одну вакансію, то інформацію про неї буде надіслано кандидатам, які визначили її як бажану. Але може скластися ситуація, коли в один день велика кількість роботодавців додала саме таку вакансію, яка визначена у кандидата як бажана. В такому випадку лист зі сповіщенням повинен буде містити в собі занадто багато даних, що збільшить його розмір та сильніше навантажить систему. Саме тому в такому випадку не будуть надаватися детальні дані про вакансії, буде надано тільки число вакансій, яке з'явилося з моменту попереднього сповіщення та посилання, яке перенаправить користувача на

сторінку перегляду вакансій, після чого в списку вакансій спочатку будуть показані нові вакансії. На рис. 4.39 наведено лист з великою кількістю вакансій.

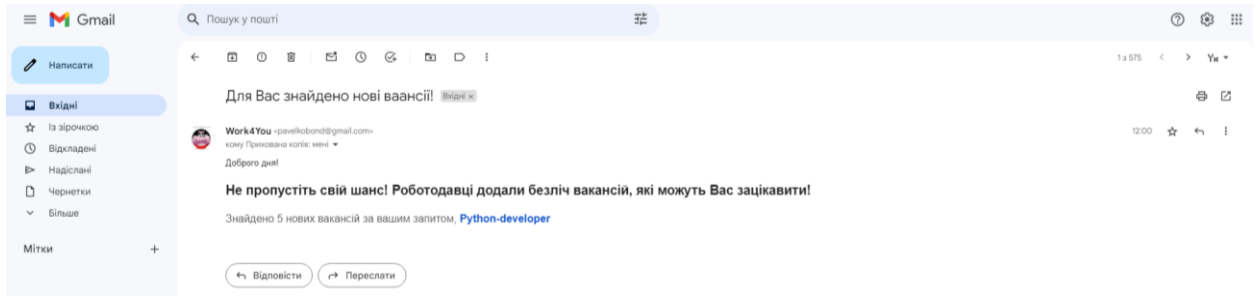


Рисунок 4.39 – Лист з великою кількістю вакансій

4.3 Висновки до розділу 4

В даному розділі було реалізовано вебзастосунок згідно вимог до вдосконаленої системи пошуку роботи та продемонстровано його роботу. Реалізовано основний функціонал, розроблена можливість отримувати сповіщення про нові вакансії на електронну пошту, створена можливість взаємодії кандидата з роботодавцем шляхом відгуку на вакансію, створено графіки, які відображають статистику вакансій та кандидатів та реалізовано можливість пройти тестування для оцінки рівня певних знань кандидата. Визначено функціональні можливості, які можна покращити, а саме відключення надсилання сповіщень на електронну пошту, якщо кандидату це не потрібно, встановлення іншого тимчасового проміжку, який буде зручним кандидатам а також можливість відновлення пароля користувача.

Для розробленого вебзастосунку було проведено повноцінне тестування, яке включає повну перевірку роботи системи. Були проведені тести працездатності реєстрації, входу в систему, валідації форм, пошуку вакансій за фільтрами, додавання файлу резюме, відправки на електронну пошту листів з великою кількістю вакансій.

ВИСНОВКИ

Таким чином, у дипломному проєкті розроблено систему, яка дозволяє удосконалити процесу пошуку роботи. Під час роботи було проаналізовано найпопулярніші аналоги вебзастосунків для пошуку роботи, а саме: «Work.ua», «Rabota.ua», «Dou.ua» та «Jooble.org». Було встановлено, що жодна з цих систем не здатна повністю задовольнити потреби користувачів, тому існує необхідність створення більш вдосконаленої системи пошуку роботи.

У роботі визначено вимоги до вдосконаленої системи пошуку роботи, які полегшать та прискорять процес отримання вакансії для потенційних працівників, а також забезпечать роботодавцям зручні інструменти для знаходження кваліфікованих кандидатів. Такими вимогами стали виконання базового функціоналу, отримання сповіщень про нові вакансії на електронну пошту, наявність зручної взаємодії кандидата та роботодавця у вигляді можливості відгукуватися на вакансію, наявність статистики та графіків та можливість пройти тестування для оцінки рівня знань кандидата.

Розроблено архітектуру системи та алгоритм її функціонування, обрано програмне забезпечення для створення системи. Було зроблено порівняння популярних фреймворків, таких як Django, ASP.NET та Laravel. Для розробки системи використано фреймворк Django, оскільки він має потужну ORM, яка надає простий та зручний інтерфейс для взаємодії з базою даних, містить вбудовані інструменти для пошуку і фільтрації вакансій, систему авторизації, адміністративну панель та управління ролями, а також надає розробникам можливість швидкого створення вебзастосунків. В якості СУБД було обрано MySQL, оскільки вона надає високу швидкодію та може впоратися з великими обсягами даних.

Таким чином, було реалізовано систему із дотриманням зазначених вимог та проведено повну перевірку її працездатності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мантур-Чубата, О., Ваганова, Л. Ринок праці України в умовах воєнного стану. Науковий журнал «Економіка і регіон», 2022, №. 2 (85), с. 81-87.
2. Круглов, В. Державна політика трансформації ринку праці: виклики цифрової епохи. Науковий вісник: Державне управління, 2021, №. 1 (7), с. 140-161.
3. Добуляк, Л.П., Костенко, С.Б. Динаміка рівня безробіття в Україні. The 16 th International scientific and practical conference «Modern scientific research: achievements, innovations and development prospects» (September 11-13, 2022) MDPC Publishing, Berlin, Germany. 2022, с. 249.
4. Грідін, О. Актуальні проблеми, сучасні тенденції та основні напрями подолання безробіття в Україні. Економіка та суспільство, 2022, №. 38, с. 2-6.
5. Гітіс, Т.П., Отоса, І.В. Дослідження сучасного стану рівня безробіття в Україні та можливих шляхів його зниження. Економічний вісник Донбасу, 2022, с. 1-3.
6. Рівень безробіття в Україні. URL: https://elartu.tntu.edu.ua/bitstream/lib/28499/2/SNT_2019_Arkhipova_A-The_level_of_unemployment_332-333.pdf (дата звернення 30.10.2023).
7. Конкуренція шалена. Що чекає українських свічерів та джунів на ІТ-ринку в 2022 році. URL: <https://ain.ua/2022/08/11/shho-chekaye-ukrayinskyh-svicheriv-ta-dzhuniv-na-it-rynku-praczi-2022> (дата звернення 30.10.2023).
8. Новаковська, О. А. Успішне працевлаштування, як початок трудової кар'єри. Всеукраїнська науково-технічна конференція магістрантів і студентів ТДАТУ, 2020, с. 40.
9. Різновиди резюме: який обрати, щоб зацікавити роботодавця. URL: <https://kha.dcz.gov.ua/publikaciya/riznovydy-rezyume-yakyu-obraty-shchob-zacikavyty-robotodavcu> (дата звернення 30.10.2023).

10. Рейтинг сайтів з роботою в Україні. URL: <https://www.unian.ua/economics/other/rejting-saytiv-z-robotoyu-v-ukrajini-12253731.html> (дата звернення 30.10.2023).
11. Jungmann, L. et al. Object-Relational Mapping. Pro Jakarta Persistence in Jakarta EE 10: An In-Depth Guide to Persistence in Enterprise Java Development, 2022, с. 99-153.
12. Павелко О.О., М.Ю. Тягунова. Автоматизований вебсайт для пошуку роботи. Інформаційні технології в освіті та науці : міжнар. наук.-практ. конф., 25 травня 2023 р. : тези доп. Запоріжжя – Мелітополь. С.151-154.
13. Павелко О.О., М.Ю. Тягунова. Вдосконалена система пошуку роботи Shipbuilding & marine infrastructure. № 1 (18). 2023. ISSN 2519-1845. – у друці.
14. Wilson, K. (2023). «Introduction to HTML.» The Absolute Beginner's Guide to HTML and CSS: A Step-by-Step Guide with Examples and Lab Exercises. Berkeley, CA: Apress, pp. 31-39.
15. George, N. Build a Website With Django 3: A complete introduction to Django 3. GNW Independent Publishing, 2020. 264 p.
16. Dale, K. Data Visualization with Python and JavaScript. «O'Reilly Media, Inc.», 2022. 589 p.
17. Phan, K. Fullstack solution for two-sided market business model with containerization (2020). 33 p.
18. Vidal-Silva, C. L., Sánchez-Ortiz, A., Serrano, J., & Rubio, J. M. Academic experience in rapid development of web information systems with Python and Django. Formacion Universitaria. 14 (5). 2021. pp. 85-94.
19. Abdugarimov, D.A. et al. 7 free best open source database management systems. Spectrum Journal of Innovation, Reforms and Development, 2022, T. 7, pp. 40-47.
20. Van Horn II, B.M., Nguyen, Q. Hands-On Application Development with PyCharm: Build applications like a pro with the ultimate Python development tool. Packt Publishing Ltd, 2023. 627 p.