

Міністерство освіти і науки України

**Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ВЕБПЛАТФОРМИ ДЛЯ ОРЕНДИ КВАРТИР З
ІНТЕГРАЦІЄЮ ШТУЧНОГО ІНТЕЛЕКТУ ЯК ПОМІЧНИКА ДЛЯ
КОРИСТУВАЧА З ВИКОРИСТАННЯМ REACT, PYTHON DJANGO ТА
SQLITE**

**DEVELOPMENT OF A WEB PLATFORM FOR RENTING APARTMENTS
WITH THE INTEGRATION OF ARTIFICIAL INTELLIGENCE AS A USER
ASSISTANT USING REACT, PYTHON DJANGO AND SQLITE**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-44
Федіна Ірина Миколаївна

Керівник:
Потейчук Михайло Іванович

Кваліфікаційну роботу
допущено до захисту
«__» _____ 20__ р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«__» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Федіні Ірині Миколаївні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка вебплатформи для оренди квартир з інтеграцією штучного інтелекту як помічника для користувача з використанням React, Python Django та SQLite»

Керівник роботи: асистент Потейчук М. І.

затверджені наказом закладу вищої освіти від «31» грудня 2025 р. № 541/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «29» травня 2026 р.

3. Вихідні дані до роботи: React, Python Django, SQLite, методичні вказівки до виконання кваліфікаційної роботи бакалавра

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): вступ, аналіз предметної області, постановка завдань, специфікація вимог, проектування системи, вибір технологій і алгоритмів, реалізація вебдодатку, тестування та налагодження, розробка бази даних, захист системи, документація для користувача, висновки

5. Перелік графічного матеріалу: 10 рисунків, 2 таблиці, 14 лістингів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	<i>Потейчук М. І.</i>		
Специфікація вимог до розробленої системи	<i>Потейчук М. І.</i>		
Розробка об'єкта проектування	<i>Потейчук М. І.</i>		
Нормоконтроль	<i>Суринович О. М.</i>		
Гарант ОП	<i>Ліщина Н. М.</i>		
Показник запозичень тексту	_____ %		
Академічна доброчесність	<i>Потейчук М. І.</i>		

7. Дата видачі завдання «03» лютого 2026 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 14.02.2026 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 03.03.2026 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 14.03.2026 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 14.04.2026 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 20.04.2026 р.	
6	Тестування та налагодження об'єкта проектування	до 04.05.2026 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 29.05.2026 р.	

Здобувач вищої освіти _____

Ірина ФЕДІНА

Керівник кваліфікаційної роботи _____

Михайло ПОТЕЙЧУК

АНОТАЦІЯ

Федіна І. М. Розробка вебплатформи для оренди квартир з інтеграцією штучного інтелекту як помічника для користувача з використанням React, Python Django та SQLite. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності Інженерія програмного забезпечення. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота бакалавра складається з вступу, 3 розділів, висновків і списку використаних джерел.

У роботі досліджено сучасний стан вебплатформ для оренди житла, проаналізовано особливості функціонування онлайн-сервісів у сфері нерухомості та визначено основні проблеми сучасних систем. У процесі виконання роботи було розроблено вебплатформу для оренди квартир із використанням React, Django REST Framework та SQLite, реалізовано систему авторизації користувачів, каталог оголошень, механізм пошуку та фільтрації житла, REST API та адміністративну панель. Також у системі реалізовано базові механізми захисту інформації, проведено тестування програмного продукту та підготовлено документацію для його встановлення і використання.

Ключові слова: вебплатформа, оренда квартир, React, Django REST Framework, SQLite, REST API, frontend, backend, інформаційна система, веброзробка, штучний інтелект.

ABSTRACT

Fedina Iryna. Development of a Web Platform for Renting Apartments With the Integration of Artificial Intelligence as a User Assistant Using React, Python Django, and SQLite. Manuscript.

Bachelor's qualification thesis of the Educational and Professional Program "Software Engineering" in the specialty 121 "Software Engineering". Lutsk National Technical University. Lutsk, 2026.

The bachelor's qualification work consists of an introduction, 3 sections, conclusions and a list of used sources.

The thesis examines the current state of web platforms for housing rental, analyzes the features of online real estate services, and identifies the main problems of modern systems. During the development process, a web platform for apartment rental using React, Django REST Framework, and SQLite was created, including user authentication, a listings catalog, housing search and filtering mechanisms, a REST API, and an administrative panel. The system also implements basic information security mechanisms, software testing, and documentation for installation and use of the platform.

Keywords: web platform, apartment rental, React, Django REST Framework, SQLite, REST API, frontend, backend, information system, web development, artificial intelligence.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ВЕБПЛАТФОРМ ДЛЯ ОРЕНДИ КВАРТИР ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	10
1.1 Аналіз сучасного стану проблеми	10
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	16
Висновки до розділу 1	17
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	18
2.1 Аналіз, визначення вимог до програмного забезпечення та проєктування системи	18
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	24
Висновки до розділу 2	27
РОЗДІЛ 3 РОЗРОБКА ВЕБПЛАТФОРМИ ДЛЯ ОРЕНДИ КВАРТИР	28
3.1 Практична реалізація об'єкта проєктування	28
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	35
3.3 Розробка бази даних.....	38
3.4 Захист інформаційно-комп'ютерної системи.....	41
3.5 Розробка документації для програмного продукту	43
Висновки до розділу 3	46
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій та стрімке зростання популярності цифрових сервісів сприяють активному впровадженню вебплатформ у різні сфери діяльності людини. Однією з таких сфер є ринок нерухомості, зокрема оренда житла, де використання вебтехнологій значно спрощує процес взаємодії між власниками квартир та потенційними орендарями. В умовах постійного зростання кількості онлайн-сервісів особливої актуальності набуває створення зручних, функціональних та інтелектуальних платформ, здатних забезпечити швидкий пошук житла, автоматизацію процесу обробки інформації та надання користувачам персоналізованих рекомендацій.

Нині існує значна кількість вебплатформ для оренди квартир, серед яких Airbnb, Booking, OLX, Dom.gia та інші. Такі системи дозволяють переглядати оголошення, здійснювати пошук житла за певними параметрами, взаємодіяти з орендодавцями та переглядати відгуки користувачів. Проте значна частина існуючих платформ має низку недоліків, серед яких перевантаженість інтерфейсу, недостатній рівень персоналізації, складність пошуку релевантних пропозицій, відсутність інтелектуальної підтримки користувача та обмежені можливості автоматизації рекомендацій. Також актуальною проблемою залишається забезпечення швидкої обробки запитів, захисту персональних даних користувачів та адаптації вебресурсів до мобільних пристроїв.

Світові тенденції розвитку вебтехнологій свідчать про активне використання штучного інтелекту у вебсистемах для автоматизації процесів аналізу даних, персоналізації контенту та покращення взаємодії з користувачем. Сучасні програмні рішення дедалі частіше інтегрують інтелектуальних помічників, чат-ботів та рекомендаційні системи, які дозволяють значно підвищити ефективність роботи сервісу та якість обслуговування користувачів. Також важливими напрямками розвитку є використання SPA-технологій, адаптивного дизайну, REST API та сучасних фреймворків для створення високопродуктивних вебзастосунків.

Актуальність теми кваліфікаційної роботи бакалавра полягає у необхідності створення сучасної вебплатформи для оренди квартир із використанням інтелектуальних механізмів підтримки користувача, що дозволить оптимізувати процес пошуку житла, підвищити швидкість обробки інформації та забезпечити зручну взаємодію між користувачами системи. Інтеграція штучного інтелекту у вебплатформу дозволить реалізувати функції рекомендацій, автоматизованого підбору житла та інтелектуального консультування користувачів.

Метою кваліфікаційної роботи бакалавра є розробка вебплатформи для оренди квартир з інтеграцією штучного інтелекту як помічника для користувача з використанням React, Python Django та SQLite.

Для досягнення поставленої мети у роботі необхідно вирішити такі завдання:

- 1) провести аналіз сучасного стану вебплатформ для оренди житла та визначити основні проблеми предметної області;
- 2) сформулювати постановку завдання на кваліфікаційну роботу бакалавра та визначити основні функціональні можливості програмного продукту;
- 3) виконати аналіз, визначення вимог до програмного забезпечення та спроектувати структуру інформаційної системи;
- 4) обґрунтувати вибір засобів, методів і алгоритмів реалізації вебплатформи;
- 5) здійснити практичну реалізацію вебплатформи для оренди квартир із використанням React, Django REST Framework та REST API;
- 6) провести тестування та налагодження інформаційно-комп'ютерної системи;
- 7) розробити структуру бази даних для зберігання інформації про користувачів, оголошення та медіафайли;
- 8) реалізувати засоби захисту інформаційно-комп'ютерної системи, включаючи механізми автентифікації та авторизації користувачів;

9) розробити документацію для програмного продукту та описати особливості його використання.

Об'єктом дослідження є процес організації онлайн-оренди житла за допомогою вебтехнологій.

Предметом дослідження є методи та засоби розробки вебплатформи для оренди квартир із використанням технологій штучного інтелекту.

У процесі виконання кваліфікаційної роботи бакалавра використано сучасні засоби веброзробки та програмування, зокрема бібліотеку React для створення клієнтської частини системи, фреймворк Django для реалізації серверної логіки та систему керування базами даних SQLite для зберігання інформації. Для реалізації окремих функцій взаємодії між компонентами системи використовуються REST API, засоби маршрутизації та механізми автентифікації користувачів.

Практичне значення роботи полягає у створенні сучасної вебплатформи для оренди квартир, яка може бути використана як основа для подальшого розвитку сервісів у сфері нерухомості, автоматизації процесу пошуку житла та інтеграції інтелектуальних рекомендаційних систем.

Кваліфікаційна робота бакалавра пов'язана із сучасними дослідженнями у сфері веброзробки, інформаційних систем, електронних сервісів та інтеграції штучного інтелекту у програмні продукти. Отримані результати можуть бути використані для подальшого вдосконалення інформаційних систем у сфері онлайн-оренди житла та створення інтелектуальних вебзастосунків.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ ВЕБПЛАТФОРМ ДЛЯ ОРЕНДИ КВАРТИР І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації суспільства вебплатформи стали невід’ємною частиною багатьох сфер діяльності людини. Одним із найбільш динамічних напрямів є сфера онлайн-оренди нерухомості, де вебтехнології забезпечують автоматизацію процесів пошуку житла, взаємодії між орендодавцями та орендарями, а також обробки великого обсягу інформації. Зростання попиту на дистанційні сервіси, мобільність користувачів та необхідність швидкого отримання актуальної інформації сприяють активному розвитку вебплатформ для оренди квартир.

Нині існує значна кількість популярних сервісів, призначених для пошуку та оренди житла. Серед найбільш відомих платформ можна виділити Airbnb, Booking, OLX Нерухомість, Dom.ria та Flatfy. Дані системи надають користувачам можливість переглядати оголошення, здійснювати фільтрацію квартир за різними параметрами, переглядати фотографії, контактувати з власниками житла та залишати відгуки. Завдяки використанню сучасних вебтехнологій такі сервіси забезпечують швидкий доступ до великої кількості інформації та значно спрощують процес пошуку житла.

Платформа Airbnb (рис. 1.1) є одним із найбільш популярних міжнародних сервісів короткострокової оренди житла [2]. Основною перевагою цієї системи є зручний інтерфейс, наявність системи рейтингів та відгуків, інтеграція онлайн-бронювання та підтримка великої кількості країн. Також сервіс використовує елементи персоналізації рекомендацій для користувачів. Проте недоліками Airbnb є високі комісії, складність для нових користувачів та обмежена доступність деяких функцій без реєстрації.

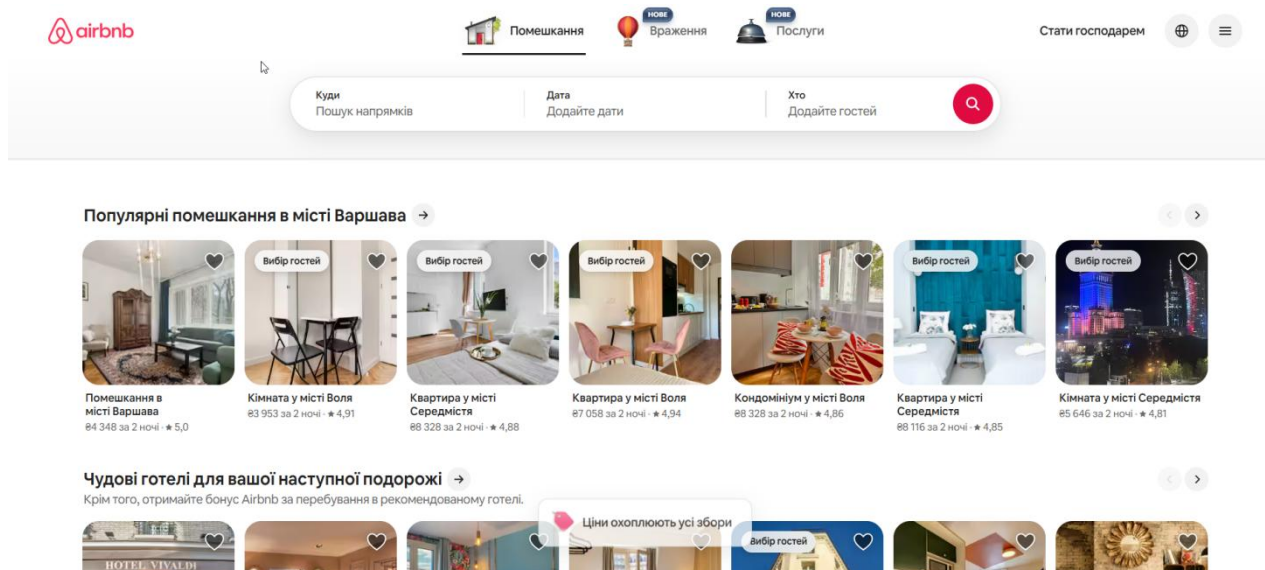


Рисунок 1.1 – Платформа Airbnb

Сервіс Booking орієнтований переважно на бронювання готелів та апартаментів (рис. 1.2), однак також підтримує функції оренди квартир [3]. Перевагами системи є швидкий пошук, детальна інформація про житло, адаптивний інтерфейс та високий рівень довіри користувачів. Недоліками є складність взаємодії між власниками житла та клієнтами, а також перевантаженість інтерфейсу великою кількістю інформації.

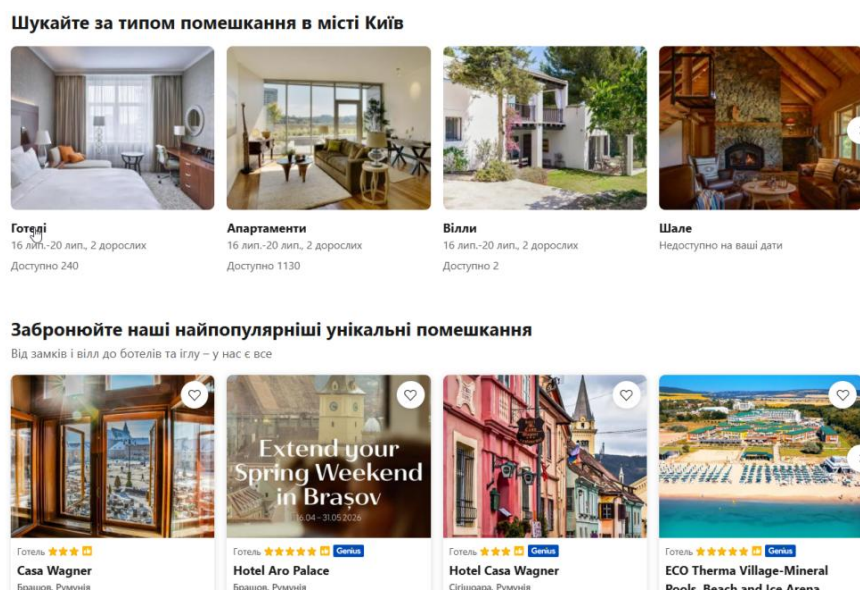


Рисунок 1.2 – Сервіс Booking

Українські платформи OLX Недухомість [1] та Dom.rіa [4] є популярними сервісами для пошуку довгострокової оренди квартир. Їх основними перевагами є велика база оголошень, можливість сортування за ціною, місцем розташування та іншими параметрами. Однак такі системи мають певні недоліки, серед яких недостатній рівень перевірки оголошень, наявність неактуальної інформації, відсутність інтелектуальних механізмів підбору житла та обмежені можливості автоматизованої підтримки користувачів.

Платформа Flatfy [7] є агрегатором оголошень нерухомості, який об'єднує інформацію з різних вебресурсів та надає користувачам можливість швидкого пошуку житла за заданими параметрами. Основною перевагою сервісу є велика кількість доступних оголошень, зручна система фільтрації та інтеграція картографічних сервісів для відображення розташування об'єктів нерухомості. Також система забезпечує швидкий доступ до пропозицій із різних джерел без необхідності окремого пошуку на кожному сайті. Водночас недоліками Flatfy є наявність дубльованих або неактуальних оголошень, обмежені можливості взаємодії між користувачами та відсутність інтелектуального помічника для автоматизованого підбору житла.

Проведений аналіз аналогів (табл. 1.1) показав, що більшість існуючих вебплатформ забезпечують базовий функціонал пошуку та перегляду квартир, однак мають низку недоліків. Основними проблемами сучасних систем є недостатній рівень персоналізації, відсутність інтелектуального помічника, складність пошуку релевантних пропозицій та недостатня автоматизація взаємодії з користувачами. Саме тому доцільним є створення нової вебплатформи для оренди квартир із інтеграцією штучного інтелекту, яка забезпечить інтелектуальний підбір житла, автоматизовані рекомендації та більш зручну взаємодію між користувачами системи.

Таблиця 1.1 – Порівняльна таблиця аналогів вебплатформ для оренди квартир

Назва платформи	Основне призначення	Переваги	Недоліки	Наявність III-функцій
Airbnb	Короткострокова оренда житла	Зручний інтерфейс, система рейтингів, онлайн-бронювання, велика база житла	Високі комісії, складна система верифікації, обмежений функціонал без реєстрації	Часткова персоналізація рекомендацій
Booking.com	Бронювання готелів та квартир	Швидкий пошук, детальні описи, адаптивний дизайн, міжнародна підтримка	Перевантаженість інтерфейсу, обмежена комунікація між користувачами	Базові рекомендації
OLX	Оренда та продаж нерухомості	Велика кількість оголошень, простий пошук, популярність серед користувачів	Наявність фейкових оголошень, відсутність перевірки даних, низький рівень персоналізації	Відсутні
DIM.RIA	Купівля та оренда житла	Перевірені оголошення, фільтрація квартир, зручна навігація	Обмежені функції автоматизації, відсутність інтелектуального помічника	Відсутні
Flatfy	Агрегація оголошень з різних сервісів	Велика база пропозицій, швидкий пошук, зручна карта	Частина оголошень неактуальна, обмежений функціонал взаємодії	Відсутні

Аналіз сучасних наукових джерел свідчить про активний розвиток цифрових платформ у сфері оренди нерухомості та впровадження інтелектуальних технологій у процес взаємодії між користувачами. У дослідженні Singh, Kumar, Shrivastava, Wadhwa та Shrimal [17] розглянуто еволюцію систем оренди квартир і апартаментів, а також проаналізовано тенденції переходу від традиційних методів пошуку житла до сучасних вебплатформ. Автори зазначають, що сучасні системи оренди нерухомості

повинні забезпечувати швидкий пошук інформації, автоматизацію процесів взаємодії між орендарями та власниками житла, підтримку фільтрації даних та зручний користувацький інтерфейс. Особлива увага приділяється необхідності використання адаптивних вебтехнологій та централізованих баз даних для підвищення ефективності роботи онлайн-сервісів.

У роботі Pino Gatica досліджується процес цифровізації інформаційних носіїв та сервісів у сфері оренди житла. Автор підкреслює, що сучасні вебсистеми повинні забезпечувати швидкий доступ до інформації, автоматичну обробку даних та підтримку дистанційної взаємодії між користувачами. У дослідженні наголошується на важливості інтеграції цифрових технологій у сферу нерухомості, оскільки це дозволяє значно спростити процес пошуку житла, скоротити час обробки запитів та підвищити рівень інформаційного обслуговування користувачів. Також автор відзначає необхідність забезпечення безпеки персональних даних та надійності інформаційних систем.

Особливу увагу сучасні дослідники приділяють використанню штучного інтелекту у вебплатформах для оренди житла. У праці Pallavi, Rishika, Sree, Ruthwik та Yashaswini [12] представлено платформу SmartRent – інтелектуальну систему оренди житла, яка використовує алгоритми штучного інтелекту для автоматизованого підбору нерухомості. Автори зазначають, що використання AI-технологій дозволяє аналізувати запити користувачів, формувати персоналізовані рекомендації та значно підвищувати точність пошуку житла. Також у роботі розглядається використання чат-ботів та інтелектуальних помічників для автоматизації консультаційної підтримки користувачів.

Проведений аналіз літературних джерел показав, що сучасні тенденції розвитку вебплатформ оренди нерухомості орієнтовані на використання адаптивних вебтехнологій, автоматизацію процесів обробки даних та інтеграцію елементів штучного інтелекту. Водночас значна кількість існуючих систем все ще має певні недоліки, серед яких недостатній рівень персоналізації, складність взаємодії з користувачами, перевантаженість інтерфейсів та відсутність інтелектуального механізму підбору житла. Саме тому актуальним є створення

вебплатформи для оренди квартир із використанням сучасних вебтехнологій та інтеграцією інформаційного помічника, що забезпечить більш ефективний та зручний пошук житла для користувачів системи.

У процесі аналізу предметної області було проведено огляд сучасних технологій веброзробки. Для створення клієнтської частини вебплатформи доцільним є використання бібліотеки React, яка забезпечує високу швидкодію інтерфейсу, компонентний підхід до розробки та підтримку SPA-застосунків. Для реалізації серверної частини системи ефективним рішенням є використання фреймворку Django, який надає готові механізми маршрутизації, роботи з базами даних, автентифікації користувачів та створення REST API. Для зберігання інформації може використовуватись система SQLite, яка є компактною, швидкою та зручною для невеликих і середніх вебпроектів.

Також у межах аналізу було виконано патентний пошук та огляд існуючих програмних рішень у сфері електронних сервісів оренди житла. Результати аналізу показали, що більшість існуючих систем орієнтовані переважно на базовий функціонал перегляду оголошень і не забезпечують достатнього рівня інтелектуальної взаємодії з користувачем. Значна частина платформ не використовує сучасні можливості штучного інтелекту для автоматизованого підбору житла та персоналізації сервісу.

Отже, проведений аналіз сучасного стану проблеми показав, що існуючі вебплатформи для оренди квартир мають низку функціональних та організаційних недоліків, які потребують вдосконалення. З огляду на це доцільним є створення нової вебплатформи для оренди квартир із інтеграцією штучного інтелекту як помічника користувача. Використання сучасних технологій React, Python Django та SQLite дозволить створити функціональну, зручну та масштабовану систему, яка забезпечить ефективний пошук житла, персоналізацію рекомендацій та покращення взаємодії між користувачами вебплатформи.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою кваліфікаційної роботи бакалавра є розробка сучасної вебплатформи для оренди квартир із використанням технологій React, Django REST Framework та SQLite, яка забезпечить зручний пошук житла, взаємодію між орендарями та орендодавцями, а також реалізацію інтелектуального помічника для автоматизованого підбору житла відповідно до потреб користувача.

У процесі виконання кваліфікаційної роботи необхідно створити програмний продукт, що поєднуватиме функціональність каталогу оголошень, системи авторизації та управління обліковими записами, модулів фільтрації та пошуку житла, а також інформаційно-довідкового помічника. Особлива увага приділяється забезпеченню зручності користування системою, адаптивності інтерфейсу та безпечної взаємодії між клієнтською і серверною частинами застосунку.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно вирішити такі завдання відповідно до структури дослідження:

- 1) провести аналіз сучасного стану вебплатформ для оренди житла, визначити їх переваги, недоліки та актуальні проблеми предметної області;
- 2) сформулювати постановку завдання кваліфікаційної роботи, визначити основні функціональні вимоги та напрями реалізації програмного продукту;
- 3) виконати аналіз і визначення вимог до програмного забезпечення, а також спроектувати архітектуру інформаційної системи;
- 4) обґрунтувати вибір засобів розробки, методів та алгоритмів реалізації вебплатформи для оренди квартир;
- 5) реалізувати клієнтську та серверну частини вебплатформи з використанням React, Django REST Framework та REST API;
- 6) провести тестування та налагодження інформаційно-комп'ютерної системи для перевірки коректності роботи функціональних модулів;

7) розробити структуру бази даних та реалізувати механізми зберігання інформації про користувачів, оголошення та медіафайли;

8) реалізувати засоби захисту інформаційно-комп'ютерної системи, включаючи автентифікацію, авторизацію та розмежування прав доступу користувачів;

9) розробити документацію для програмного продукту, яка міститиме опис структури системи, основних функцій та особливостей використання вебплатформи.

У результаті виконання кваліфікаційної роботи бакалавра має бути створена сучасна вебплатформа для оренди квартир, яка забезпечує зручний пошук житла, підтримує роботу з оголошеннями, реалізує механізми авторизації користувачів та надає можливість використання інтелектуального помічника для підбору житла.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи бакалавра було проаналізовано новітній стан вебплатформ для оренди житла та досліджено основні підходи до організації онлайн-сервісів у сфері нерухомості. У результаті аналізу встановлено основні недоліки існуючих систем, серед яких недостатня персоналізація, обмежені можливості фільтрації та відсутність інтелектуальних механізмів підбору житла. Проведене дослідження підтвердило доцільність розробки власної вебплатформи з використанням React та Django, яка поєднуватиме адаптивний інтерфейс, систему управління оголошеннями, REST API, механізми авторизації користувачів та інформаційного помічника для автоматизованого пошуку квартир. Отримані результати стали основою для формування вимог до системи та подальшого проєктування програмного забезпечення.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до програмного забезпечення та проєктування системи

У процесі розробки сучасних вебзастосунків важливе значення має правильне визначення вимог до програмного забезпечення та проєктування структури системи ще на початкових етапах створення програмного продукту. Саме етап аналізу вимог дозволяє визначити основні функціональні можливості системи, сформувати архітектуру програмного забезпечення та забезпечити ефективну взаємодію між окремими компонентами застосунку.

Розроблювана вебплатформа для оренди квартир призначена для автоматизації процесу пошуку житла, публікації оголошень та взаємодії між орендарями й орендодавцями. Система забезпечує користувачам можливість швидкого пошуку житла за заданими параметрами, перегляду інформації про об'єкти нерухомості, створення власних оголошень, а також використання інформаційного помічника для підбору житла на основі текстових запитів.

Для моделювання програмної системи було обрано багаторівневу клієнт-серверну архітектуру, яка передбачає розділення frontend та backend частин застосунку. Такий підхід забезпечує гнучкість розробки, спрощує масштабування системи та дозволяє незалежно розвивати інтерфейс користувача і серверну логіку. Клієнтська частина реалізована за допомогою бібліотеки React, що дозволяє створювати інтерактивний адаптивний інтерфейс користувача. Серверна частина побудована на основі Django REST Framework, який забезпечує створення REST API для обміну даними між сервером та клієнтом.

Під час визначення вимог до системи було використано функціональний підхід до аналізу програмного забезпечення. Основна увага приділялась потребам кінцевих користувачів системи орендарів, орендодавців та адміністратора. На основі аналізу предметної області було сформовано перелік основних функцій, які повинна підтримувати вебплатформа.

До основних функціональних вимог системи належать:

- 1) реєстрація та авторизація користувачів;
- 2) підтримка різних ролей користувачів;
- 3) створення, редагування та видалення оголошень;
- 4) завантаження фотографій об'єктів нерухомості;
- 5) перегляд детальної інформації про житло;
- 6) реалізація пошуку та фільтрації оголошень;
- 7) підтримка роботи інформаційного помічника;
- 8) адміністрування користувачів та оголошень;
- 9) забезпечення захисту даних та контролю доступу.

Оскільки система працює у мережі Інтернет, важливе значення мають нефункціональні вимоги до програмного забезпечення. До них належать швидкодія системи, адаптивність інтерфейсу, надійність збереження інформації, безпечна передача даних та зручність користування вебплатформою.

Для реалізації системи використовується REST-архітектура, відповідно до якої frontend частина надсилає HTTP-запити до backend API. Серверна частина обробляє запити користувачів, виконує роботу з базою даних та повертає результати у форматі JSON. Авторизація користувачів реалізована за допомогою токенів доступу, що забезпечує захищену взаємодію між клієнтом і сервером.

У процесі проєктування системи було сформовано структуру основних програмних модулів. Архітектура системи включає:

- 1) модуль автентифікації та авторизації;
- 2) модуль управління користувачами;
- 3) модуль роботи з оголошеннями;
- 4) модуль завантаження зображень;
- 5) модуль пошуку та фільтрації;
- 6) модуль інформаційного помічника;
- 7) адміністративний модуль.

Користувачами системи є три основні категорії:

- 1) орендар – користувач, який переглядає оголошення та виконує пошук житла;
- 2) орендодавець – користувач, який створює та редагує власні оголошення;
- 3) адміністратор – користувач, який здійснює контроль за роботою системи через адміністративну панель Django.

Для опису взаємодії користувачів із системою доцільно використати UML-модель прецедентів. Основними прецедентами системи є:

- 1) реєстрація користувача;
- 2) вхід до системи;
- 3) перегляд каталогу оголошень;
- 4) пошук житла за параметрами;
- 5) перегляд детальної інформації про житло;
- 6) створення оголошення;
- 7) редагування оголошення;
- 8) використання інформаційного помічника;
- 9) адміністрування системи.

У системі реалізовано UX/UI-дизайн, орієнтований на простоту та зручність користування. Інтерфейс користувача побудований із використанням компонентного підходу React та стилізації Tailwind CSS. Основні сторінки системи містять навігаційне меню, форми введення інформації, сторінки перегляду оголошень, сторінку авторизації, сторінку реєстрації, модуль пошуку та сторінку роботи помічника.

Інформаційні потоки системи побудовані за схемою «клієнт – сервер – база даних». Користувач через вебінтерфейс надсилає запит до REST API, серверна частина обробляє запит, звертається до бази даних SQLite та повертає відповідь клієнтській частині у форматі JSON. Такий підхід забезпечує централізоване управління даними та спрощує інтеграцію окремих компонентів системи.

Для формалізації структури та логіки роботи вебплатформи у процесі проєктування системи було використано UML-модельювання. UML-діаграми дозволяють наочно відобразити взаємодію користувачів із системою, структуру програмних компонентів та взаємозв'язки між основними сутностями програмного забезпечення. Використання UML забезпечує спрощення процесу проєктування, полегшує аналіз архітектури системи та дозволяє більш детально описати функціональні можливості вебплатформи. Діаграма прецедентів (рис. 2.1) описує взаємодію користувачів із вебплатформою та відображає основні функціональні можливості системи.

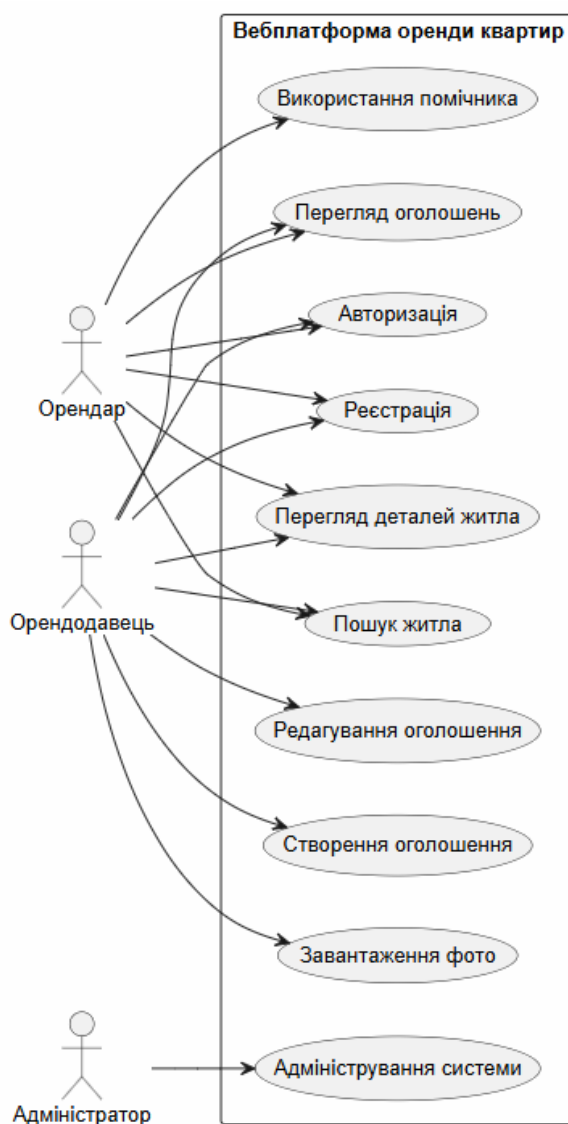


Рисунок 2.1 – UML-діаграма прецедентів

У розробленій системі передбачено три основні типи користувачів, орендар, орендодавець та адміністратор. Орендар може переглядати оголошення, використовувати систему пошуку та взаємодіяти з інформаційним помічником. Орендодавець додатково має можливість створювати та редагувати власні оголошення. Адміністратор здійснює контроль користувачів та інформаційних ресурсів системи через адміністративну панель Django.

Діаграма класів (рис. 2.2) описує структуру основних сутностей системи та взаємозв'язки між ними. Основними класами у вебплатформі є користувач, профіль користувача, оголошення та фотографії оголошень. Один користувач може мати декілька оголошень, а кожне оголошення може містити декілька фотографій. Також система підтримує розмежування ролей користувачів.

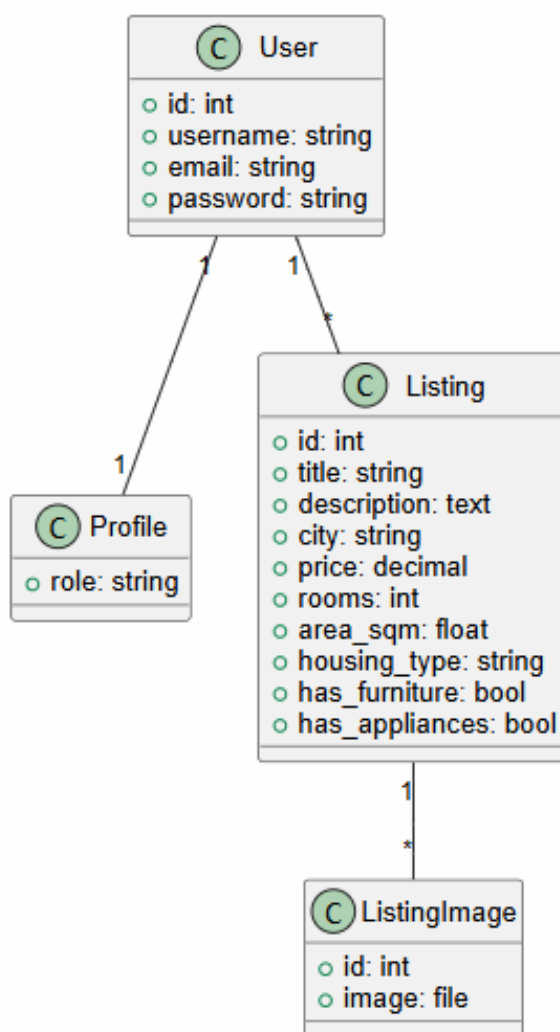


Рисунок 2.2 – UML-діаграма класів

Діаграма компонентів (рис. 2.3) демонструє взаємодію між основними програмними компонентами системи. Клієнтська частина, реалізована на React, взаємодіє із серверною частиною Django REST Framework через REST API. Серверна частина забезпечує обробку HTTP-запитів, авторизацію користувачів та взаємодію із базою даних SQLite.

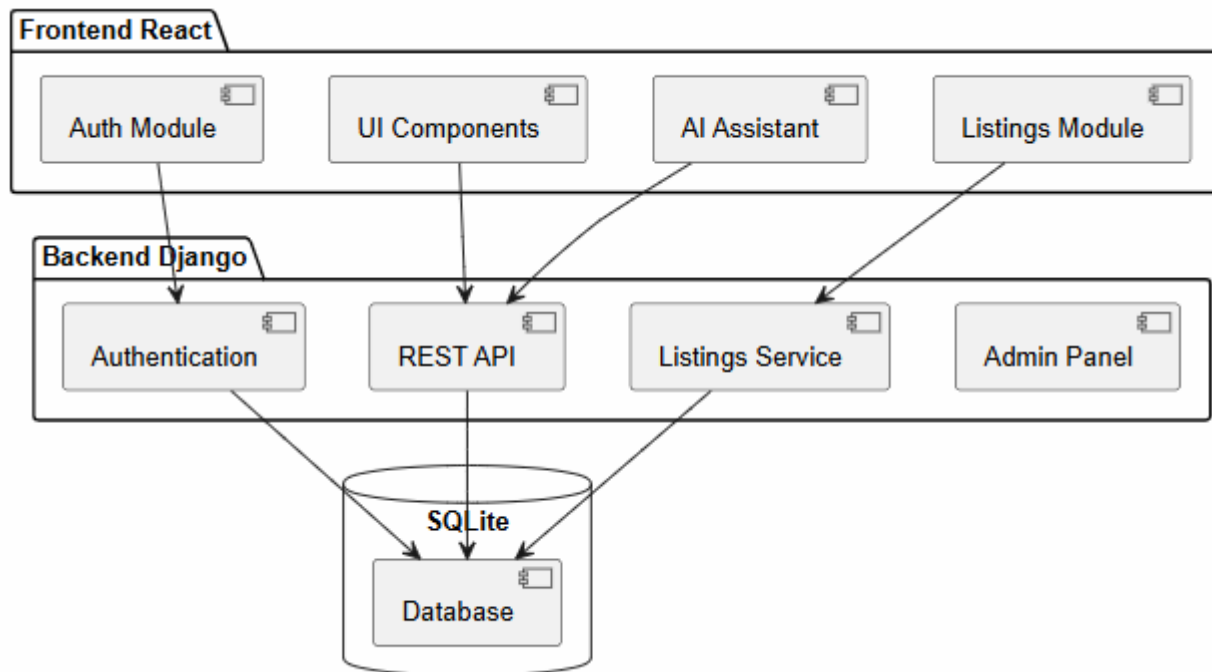


Рисунок 2.3 – UML-діаграма компонентів

Діаграма розгортання (рис. 2.4) описує фізичну структуру системи та взаємодію між її апаратними і програмними компонентами. Користувач працює із системою через веббраузер, який взаємодіє із вебсервером Django через мережу Інтернет за протоколом HTTP/TCP-IP. Серверна частина обробляє запити та працює із базою даних SQLite.

Використання UML-моделювання дозволило формалізувати структуру вебплатформи для оренди квартир, описати взаємодію користувачів із системою та визначити архітектуру програмного забезпечення. Побудовані діаграми спрощують процес подальшої реалізації програмного продукту, забезпечують

кращу організацію програмних компонентів та дозволяють більш детально описати логіку роботи інформаційної системи.

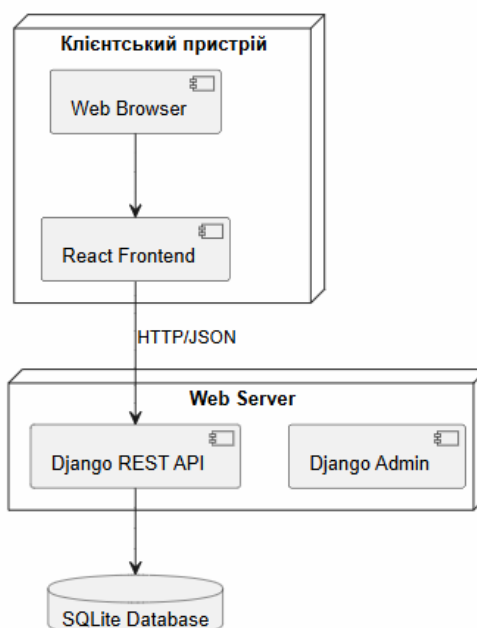


Рисунок 2.4 – UML-діаграма розгортання

У результаті проведеного аналізу було визначено основні вимоги до програмного забезпечення та сформовано архітектуру вебплатформи для оренди квартир. Отримані результати створюють основу для подальшої реалізації функціональних модулів системи, розробки бази даних та програмної реалізації інформаційно-комп'ютерної системи.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Під час розробки сучасних інформаційних систем важливе значення має правильний вибір програмних засобів, методів та алгоритмів реалізації програмного продукту. Від обраних технологій залежить продуктивність системи, зручність розробки, можливість масштабування, безпечність роботи та ефективність подальшого супроводу програмного забезпечення.

Для реалізації вебплатформи оренди квартир було проаналізовано сучасні засоби веброзробки, які використовуються для створення клієнтських і серверних вебзастосунків. Основна увага приділялась технологіям, які забезпечують швидку розробку, підтримку REST API, адаптивний інтерфейс користувача та можливість організації взаємодії між frontend і backend частинами системи.

У результаті проведеного аналізу для реалізації клієнтської частини вебплатформи було обрано бібліотеку React [14]. Цей інструмент є одним із найпоширеніших засобів створення сучасних односторінкових вебзастосунків. Основною перевагою React є компонентний підхід до побудови інтерфейсу, що дозволяє повторно використовувати окремі елементи системи та спрощує підтримку програмного коду. Також React забезпечує високу швидкодію за рахунок використання Virtual DOM та підтримує зручну інтеграцію із REST API.

Для маршрутизації між сторінками застосунку було використано бібліотеку React Router DOM [15]. Вона забезпечує навігацію між окремими сторінками системи без повного перезавантаження вебсторінки, що позитивно впливає на швидкодію та зручність користування системою.

Для стилізації інтерфейсу користувача було використано Tailwind CSS. Цей інструмент дозволяє швидко створювати адаптивний сучасний інтерфейс за допомогою набору готових CSS-класів. Перевагою Tailwind CSS є спрощення процесу створення дизайну та забезпечення єдиного стилю компонентів системи [19].

Серверна частина вебплатформи реалізована за допомогою фреймворку Django та Django REST Framework. Вибір Django обумовлений його високим рівнем безпеки, наявністю вбудованої адміністративної панелі, підтримкою ORM та можливістю швидкого створення вебзастосунків [5]. Django REST Framework забезпечує зручне створення REST API та підтримує серіалізацію даних, систему автентифікації користувачів і механізми контролю доступу [6].

Для зберігання інформації використовується база даних SQLite. Основною перевагою SQLite є простота використання та інтеграції із Django. Для

кваліфікаційної роботи бакалавра даний тип бази даних є достатнім, оскільки не потребує окремого сервера баз даних і забезпечує зручне локальне зберігання інформації [18].

У процесі реалізації системи використовувалась токенна автентифікація користувачів. Після успішного входу до системи сервер генерує токен доступу, який зберігається у локальному сховищі браузера та використовується для виконання авторизованих HTTP-запитів до API. Такий підхід дозволяє забезпечити безпечну взаємодію між клієнтською та серверною частинами системи.

Для обміну даними між frontend та backend частинами системи використовується REST API, яке забезпечує взаємодію між клієнтською та серверною частинами вебплатформи [16]. Передача інформації здійснюється у форматі JSON за допомогою HTTP-запитів [8]. У процесі роботи системи використовуються різні типи HTTP-методів, GET-запити застосовуються для отримання інформації з сервера, POST – для створення нових записів у базі даних, PUT та PATCH – для редагування існуючих даних, а DELETE використовується для видалення інформації із системи [9].

У процесі розробки вебплатформи було використано компонентний метод побудови програмної системи. Кожен функціональний елемент реалізований як окремий компонент або модуль. Такий підхід дозволяє спростити підтримку програмного забезпечення, підвищити повторне використання коду та забезпечити модульність системи.

Для реалізації пошуку оголошень у вебплатформі застосовано алгоритм фільтрації даних за параметрами, які задає користувач. Пошук житла здійснюється з урахуванням міста розташування об'єкта, вартості оренди, кількості кімнат, типу житла, наявності меблів та побутової техніки. Також система підтримує текстовий пошук за описом оголошення, що дозволяє користувачу швидше знаходити необхідні варіанти житла відповідно до власних потреб.

Алгоритм роботи пошуку передбачає отримання параметрів із форми фільтрації, формування HTTP-запиту до REST API та обробку результатів серверною частиною системи. Після отримання запиту сервер виконує фільтрацію записів у базі даних та повертає список відповідних оголошень клієнтській частині.

Для роботи інформаційного помічника використовується алгоритм аналізу текстового запиту користувача. Помічник обробляє введений текст, визначає основні параметри пошуку та формує рекомендації щодо доступних оголошень із каталогу системи. Це дозволяє спростити процес пошуку житла та зробити взаємодію користувача із системою більш зручною.

Отже, у результаті аналізу було обрано сучасні програмні засоби та методи реалізації вебплатформи для оренди квартир. Використання React, Django REST Framework, SQLite та REST API дозволяє створити сучасну інформаційно-комп'ютерну систему із зручним інтерфейсом користувача, ефективною взаємодією між компонентами та підтримкою основних функціональних можливостей вебплатформи.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи проведено аналіз вимог до програмного забезпечення та виконано проєктування вебплатформи для оренди квартир. Визначено основні функціональні й нефункціональні вимоги системи, а також обґрунтовано вибір технологій React, Django REST Framework і SQLite для реалізації застосунку.

Також описано архітектуру системи, взаємодію frontend і backend частин через REST API та основні функціональні можливості вебплатформи. Отримані результати стали основою для подальшої реалізації та тестування інформаційної системи.

РОЗДІЛ 3

РОЗРОБКА ВЕБПЛАТФОРМИ ДЛЯ ОРЕНДИ КВАРТИР

3.1 Практична реалізація об'єкта проєктування

Практична реалізація вебплатформи для оренди квартир здійснювалася із використанням сучасних засобів веброзробки та клієнт-серверної архітектури. У процесі створення програмного забезпечення було використано frontend-частину на основі бібліотеки React та backend-частину на основі Django REST Framework. Такий підхід дозволив розділити логіку роботи інтерфейсу користувача та серверної обробки даних, що забезпечило масштабованість, зручність супроводу та ефективність роботи системи. Розробка програмного забезпечення виконувалася у середовищі Visual Studio Code [20]. Для керування залежностями frontend-частини використовувався пакетний менеджер npm [11], а backend-система працювала у віртуальному середовищі Python. Структура проєкту була розділена на окремі директорії frontend та backend, що дозволило логічно відокремити клієнтську та серверну частини застосунку.

Backend-частина системи реалізована мовою Python [13] із використанням фреймворку Django. Основною точкою запуску серверної частини є файл manage.py (ліст. 3.1), який забезпечує виконання адміністративних команд Django, запуск сервера розробки, створення міграцій та управління базою даних. У програмному коді використовується встановлення конфігурації середовища через параметр DJANGO_SETTINGS_MODULE та підключення механізму execute_from_command_line для виконання команд системи. Серверна частина системи містить окремі модулі accounts, listings та core. Модуль accounts відповідає за авторизацію, реєстрацію та управління профілями користувачів. Модуль listings реалізує логіку роботи з оголошеннями, фотографіями житла та пошуком нерухомості. У модулі також реалізовано сервіс chat_service.py, який використовується для роботи інформаційного помічника. Структура backend-проєкту включає моделі, серіалізатори, представлення та маршрути API.

Лістинг 3.1 – Файл запуску серверної частини Django-застосунку (manage.py)

```
#!/usr/bin/env python
import os
import sys
def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'config.settings')
    from django.core.management import execute_from_command_line
    execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

кінець лістингу 3.1

Frontend-частина реалізована мовою JavaScript із використанням бібліотеки React. Для маршрутизації між сторінками системи використано бібліотеку React Router (ліст. 3.2). Основні маршрути вебплатформи визначаються у файлі App.jsx.

Лістинг 3.2 – Реалізація маршрутизації сторінок у React-застосунку

```
<Route path="/" element={<MainLayout />}>
  <Route index element={<HomePage />} />
  <Route path="listings" element={<ListingsPage />} />
  <Route path="listings/:listingId" element={<ListingDetailPage
/>} />
  <Route path="my-listings" element={<MyListingsPage />} />
  <Route path="ai-assistant" element={<AiAssistantPage />} />
  <Route path="login" element={<LoginPage />} />
  <Route path="register" element={<RegisterPage />} />
</Route>
```

кінець лістингу 3.2

Наведений фрагмент коду демонструє реалізацію системи маршрутизації вебзастосунку. Кожен маршрут відповідає окремій сторінці системи, зокрема головній сторінці, сторінці оголошень, формі авторизації, особистому кабінету орендодавця та сторінці інформаційного помічника. Для реалізації системи автентифікації користувачів використано механізм токенів. Після успішного входу користувача токен зберігається у localStorage браузера (ліст. 3.3), що

дозволяє підтримувати авторизований стан користувача під час роботи із системою.

Лістинг 3.3 – Реалізація механізму збереження токена авторизації користувача

```
const TOKEN_KEY = "apartment_rent_token";
export function getStoredToken()
{
    return localStorage.getItem(TOKEN_KEY);
}
export function setStoredToken(token)
{
    if (token) localStorage.setItem(TOKEN_KEY, token);
    else localStorage.removeItem(TOKEN_KEY);
}
```

кінець лістингу 3.3

Для обміну даними між frontend та backend частинами системи використовується REST API. Передача інформації здійснюється у форматі JSON через HTTP-запити. У системі реалізовано отримання даних про оголошення, створення нових записів, редагування та видалення інформації, а також авторизацію користувачів. Важливою складовою вебплатформи є сторінка пошуку оголошень, реалізована у компоненті ListingsPage.jsx. Для пошуку квартир використовується система фільтрації за параметрами користувача. Алгоритм пошуку враховує місто, вартість оренди, кількість кімнат, тип житла, наявність меблів та побутової техніки. Отримані результати відображаються у вигляді карток оголошень. Для реалізації пошуку квартир у системі використовується набір параметрів фільтрації, реалізований за допомогою React state (ліст. 3.4). Наведений фрагмент коду демонструє створення структури фільтрів для реалізації пошуку житла. Значення параметрів зберігаються у стані React-компонента та використовуються для формування HTTP-запитів до серверної частини системи.

Лістинг 3.4 – Структура параметрів фільтрації оголошень у системі пошуку

```
const [filters, setFilters] = useState
({
  search: "",
  city: "",
  min_price: "",
  max_price: "",
  min_rooms: "",
  max_rooms: "",
  housing_type: "",
  has_furniture: "",
  has_appliances: "",
});
```

кінець лістингу 3.4

Особливу увагу під час реалізації програмного забезпечення було приділено створенню адаптивного інтерфейсу користувача. Для стилізації вебзастосунку використовувався Tailwind CSS, який забезпечив створення сучасного інтерфейсу з підтримкою адаптації до різних розмірів екранів. Головна сторінка системи реалізує інформаційні блоки, навігаційне меню, каталог оголошень та інтеграцію інформаційного помічника. Також у системі реалізовано адміністративну панель Django, яка дозволяє адміністратору здійснювати модерацію оголошень, керувати користувачами та контролювати роботу вебплатформи. Адміністративний інтерфейс забезпечує централізоване управління інформаційною системою та спрощує супровід програмного забезпечення.

На рисунку 3.1 зображено сторінку реєстрації користувача у вебплатформі для оренди квартир. Інтерфейс сторінки реалізовано із використанням React та стилізовано за допомогою Tailwind CSS. Форма реєстрації містить поля для введення логіна, електронної пошти, пароля та підтвердження пароля. Також користувач має можливість обрати тип акаунта, орендар або орендодавець. Після заповнення форми дані передаються до backend-частини системи для створення нового облікового запису та подальшої авторизації користувача.

Рисунок 3.1 – Сторінка реєстрації користувача вебплатформи

На рисунку 3.2 представлено головну сторінку розробленої вебплатформи.

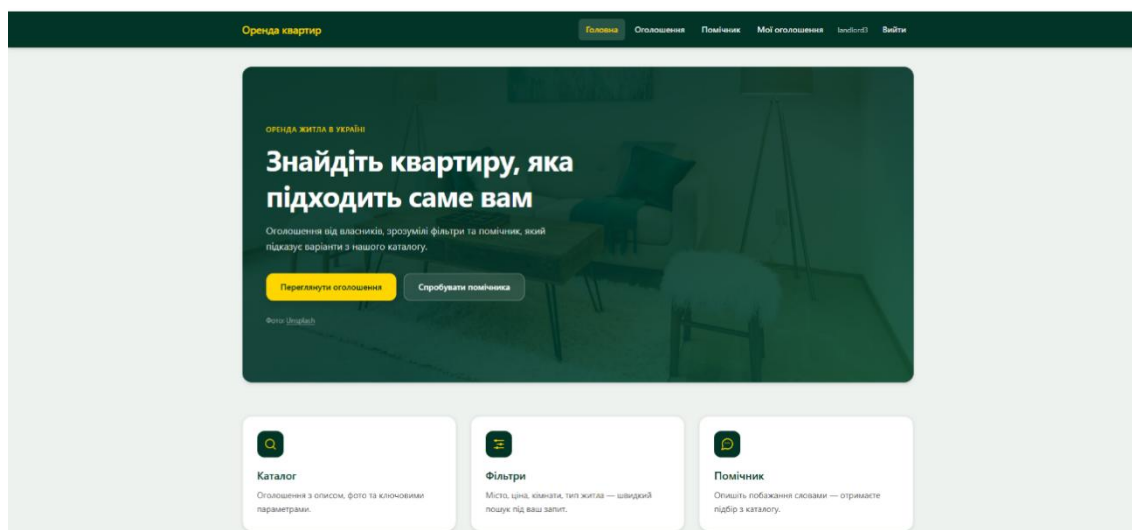


Рисунок 3.2 – Головна сторінка вебплатформи для оренди квартир

Сторінка містить навігаційне меню, інформаційний банер, короткий опис функціональних можливостей системи та кнопки переходу до каталогу оголошень і AI-помічника. Нижче розміщено блоки з описом основних функцій системи, серед яких каталог оголошень, система фільтрації та інтелектуальний

помічник. Інтерфейс реалізовано відповідно до сучасних принципів UX/UI-дизайну та адаптовано для зручної роботи користувачів. На рисунку 3.3 показано сторінку перегляду оголошень вебплатформи.

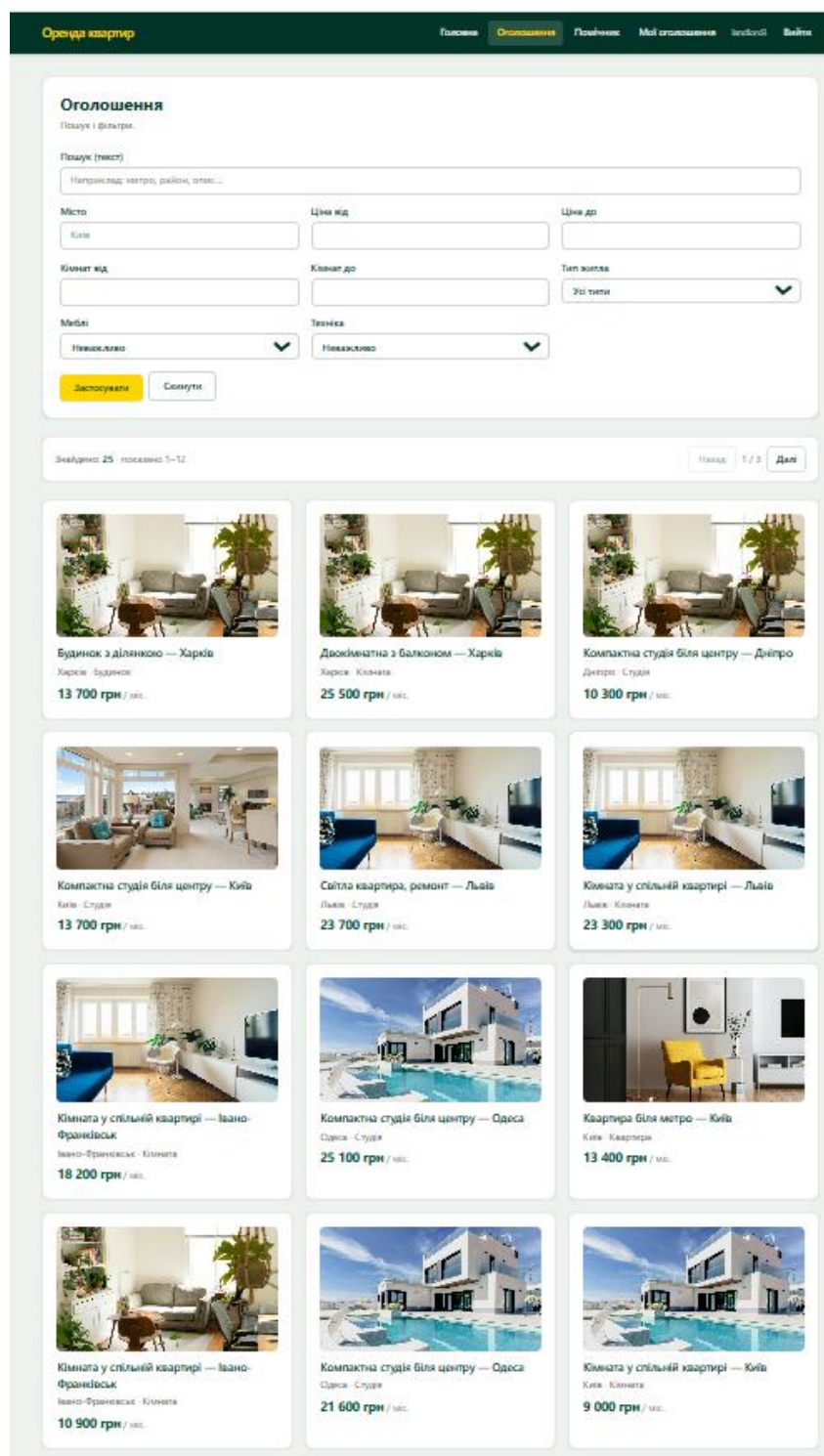


Рисунок 3.3 – Сторінка каталогу оголошень із системою фільтрації

У верхній частині сторінки реалізовано систему пошуку та фільтрації житла за різними параметрами, зокрема містом, ціною, кількістю кімнат, типом житла, наявністю меблів та техніки. Нижче відображаються картки оголошень із фотографіями квартир, коротким описом та вартістю оренди. Також реалізовано пагінацію для переходу між сторінками результатів пошуку. Дана сторінка забезпечує швидкий та зручний пошук житла відповідно до потреб користувача.

На рисунку 3.4 представлено адміністративну панель системи, реалізовану за допомогою Django Admin. Адміністративний інтерфейс забезпечує можливість перегляду, редагування, додавання та видалення оголошень, а також керування користувачами та іншими об'єктами системи. У таблиці відображаються основні параметри оголошень, серед яких назва, місто, ціна, кількість кімнат, статус оголошення та власник. Використання Django Admin дозволило значно спростити процес адміністрування вебплатформи та забезпечити централізоване керування інформацією системи.

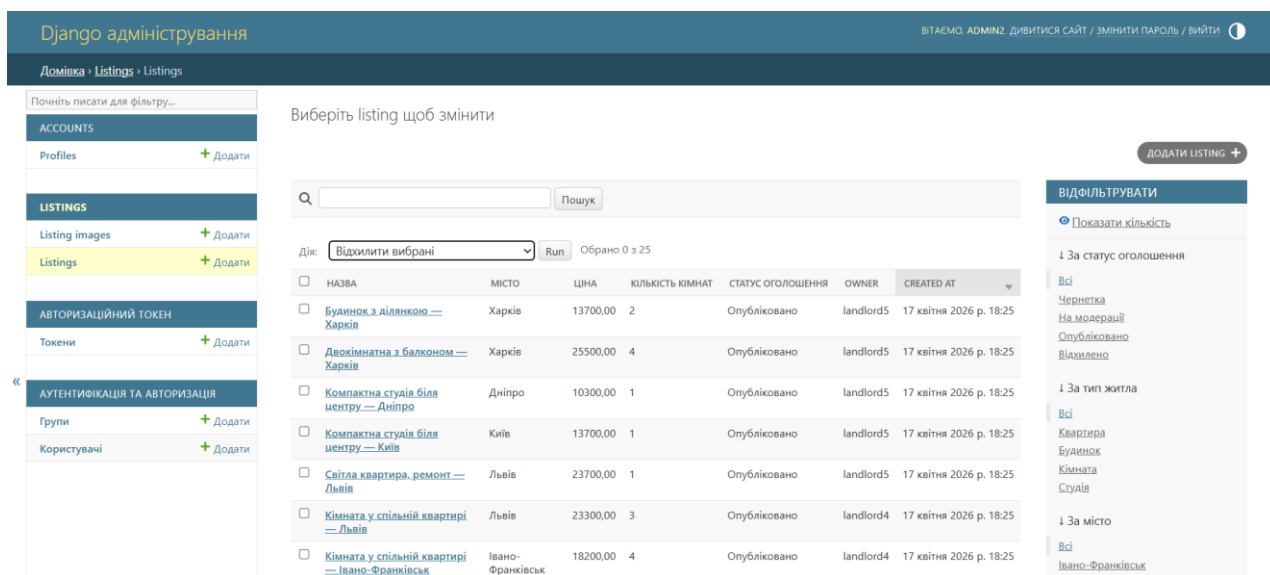


Рисунок 3.4 – Адміністративна панель керування оголошеннями Django

Отже, у процесі практичної реалізації об'єкта проектування було створено повнофункціональну вебплатформу для оренди квартир із підтримкою авторизації користувачів, публікації оголошень, системи пошуку та фільтрації житла, інформаційного помічника та адміністративної панелі управління.

Використання сучасних технологій веброзробки забезпечило високу гнучкість, зручність використання та можливість подальшого розширення функціональних можливостей системи.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Після завершення основної реалізації вебплатформи для оренди квартир було проведено тестування та налагодження інформаційно-комп'ютерної системи. Основною метою тестування була перевірка коректності роботи програмного забезпечення, виявлення можливих помилок та забезпечення стабільної роботи системи в різних умовах використання. У процесі тестування перевірялася взаємодія між frontend та backend частинами застосунку, правильність обробки даних, функціонування REST API та коректність роботи інтерфейсу користувача.

Для тестування системи використовувалися методи функціонального, інтеграційного та ручного тестування. Функціональне тестування застосовувалося для перевірки роботи окремих модулів системи, зокрема авторизації, реєстрації користувачів, створення та редагування оголошень, роботи пошуку та фільтрації житла. Інтеграційне тестування дозволило перевірити коректність взаємодії між клієнтською та серверною частинами системи, а також правильність передачі даних через REST API. Ручне тестування використовувалося для перевірки роботи користувацького інтерфейсу та адаптивності вебплатформи на різних розмірах екранів.

У процесі тестування було перевірено роботу механізму авторизації користувачів. Після введення правильних облікових даних система успішно здійснювала автентифікацію користувача та надавала доступ до функціональних можливостей відповідно до ролі користувача. Також було перевірено обробку помилкових даних під час входу в систему. У разі неправильного логіна або пароля система виводила повідомлення про помилку без порушення роботи застосунку.

Окрему увагу приділено тестуванню системи пошуку та фільтрації оголошень. Перевірялася коректність роботи фільтрів за містом, ціною, кількістю кімнат, типом житла, наявністю меблів та техніки. Результати тестування показали, що система правильно формує HTTP-запити до серверної частини та відображає релевантні результати відповідно до параметрів пошуку користувача. Було проведено тестування сторінки перегляду детальної інформації про оголошення. Перевірено коректність завантаження фотографій, відображення характеристик житла, опису, контактної інформації та статусу оголошення. Також здійснювалася перевірка роботи сторінки у випадку відсутності оголошення або помилки завантаження даних.

Для налагодження програмного забезпечення використовувалися вбудовані засоби браузера Google Chrome DevTools, консоль розробника Visual Studio Code та механізми логування Django. Під час тестування було виявлено та усунуто ряд помилок, пов'язаних із передачею параметрів фільтрації, обробкою HTTP-запитів та відображенням інформації на frontend-частині системи. Зокрема, було виправлено помилки обробки невалідних даних під час авторизації користувачів, усунуто проблеми із завантаженням зображень та оптимізовано механізм оновлення списку оголошень після застосування фільтрів.

Також було проведено перевірку роботи інформаційного помічника, який реалізує підбір житла за текстовим запитом користувача. Тестування показало, що система коректно аналізує введений текст та формує відповідний список рекомендованих оголошень із каталогу. У процесі тестування також було визначено мінімальні системні вимоги для роботи інформаційної системи. Для коректного функціонування вебплатформи необхідний персональний комп'ютер або мобільний пристрій із доступом до мережі Інтернет та сучасним веббраузером, який підтримує JavaScript і технології HTML5 та CSS3. Серверна частина системи потребує встановленого Python версії 3.11 або новішої, Django, Django REST Framework та Node.js [10] для роботи frontend-частини застосунку.

Рекомендованими параметрами для стабільної роботи системи є операційна система Windows 10 або новіша, не менше 8 ГБ оперативної пам'яті,

ефективний багатоядерний процесор та стабільне мережеве підключення. Для розробки та супроводу програмного забезпечення рекомендовано використовувати середовище Visual Studio Code та сучасні браузері Google Chrome або Mozilla Firefox. Результати тестування наведені в таблиці 3.1.

Таблиця 3.1 – Результати тестування функціональних можливостей вебплатформи

№	Функціональний модуль	Тестова дія	Очікуваний результат	Результат тестування
1	Реєстрація користувача	Введення коректних даних під час реєстрації	Створення нового акаунта та автоматична авторизація	Успішно
2	Авторизація користувача	Введення правильного логіна та пароля	Вхід у систему та отримання доступу до функцій користувача	Успішно
3	Перевірка помилкового входу	Введення неправильного пароля	Відображення повідомлення про помилку	Успішно
4	Перегляд каталогу	Відкриття сторінки оголошень	Завантаження списку квартир із бази даних	Успішно
5	Фільтрація оголошень	Застосування фільтрів пошуку	Відображення релевантних результатів	Успішно
6	Перегляд детальної інформації	Відкриття сторінки конкретного оголошення	Відображення повної інформації про житло	Успішно
7	Додавання оголошення	Створення нового оголошення орендодавцем	Збереження інформації у базі даних	Успішно
8	Редагування оголошення	Зміна параметрів існуючого оголошення	Оновлення даних у системі	Успішно
9	Завантаження фотографій	Додавання зображень до оголошення	Коректне збереження та відображення фото	Успішно
10	Робота інформаційного помічника	Введення текстового запиту користувача	Відображення рекомендованих оголошень	Успішно
11	Адміністративна панель	Вхід адміністратора до Django Admin	Отримання доступу до керування системою	Успішно
12	Вихід із системи	Натискання кнопки «Вийти»	Видалення токена та завершення сесії	Успішно

Результати тестування показали, що вебплатформа стабільно функціонує, забезпечує коректну взаємодію між користувачами та серверною частиною системи, а також успішно реалізує основні функціональні можливості. Виявлені

під час тестування помилки були усунені, що дозволило підвищити надійність, зручність використання та продуктивність інформаційно-комп'ютерної системи.

3.3 Розробка бази даних

Однією з важливих складових вебплатформи для оренди квартир є база даних, яка забезпечує збереження, обробку та швидкий доступ до інформації про користувачів, оголошення, фотографії житла та результати взаємодії із системою. У розробленій інформаційній системі база даних реалізована із використанням SQLite, яка інтегрована у фреймворк Django та забезпечує зручну роботу із даними під час розробки й тестування вебзастосунку.

Файл бази даних `db.sqlite3` автоматично створюється у `backend`-частині проєкту після виконання міграцій Django. Для взаємодії з базою даних використовувався ORM-механізм Django, який дозволяє працювати з таблицями у вигляді Python-моделей без необхідності ручного написання великої кількості SQL-запитів. Такий підхід спрощує розробку програмного забезпечення, забезпечує цілісність даних та дозволяє швидко виконувати операції створення, редагування та пошуку інформації.

У системі реалізовано декілька основних таблиць бази даних, які відповідають окремим сутностям предметної області. Основними таблицями є таблиця користувачів, таблиця оголошень та таблиця фотографій житла. Таблиця користувачів містить інформацію про логін, електронну пошту, роль користувача та параметри авторизації. Таблиця оголошень зберігає інформацію про житло, зокрема назву, опис, місто, адресу, вартість оренди, кількість кімнат, площу, тип житла та контактну інформацію власника. Таблиця фотографій забезпечує збереження шляхів до зображень, пов'язаних із конкретними оголошеннями.

Для створення структури бази даних використовувалися Django-моделі. Після опису моделей виконувалися міграції за допомогою команд

makemigrations та migrate, які автоматично формували необхідні таблиці у базі даних (ліст. 3.5).

Лістинг 3.5 – Виконання міграцій для створення структури бази даних Django

```
python manage.py makemigrations  
python manage.py migrate
```

кінець лістингу 3.5

Наведені команди дозволяють створити SQL-структуру бази даних відповідно до описаних моделей Django. У процесі виконання міграцій автоматично формуються таблиці, зв'язки між ними та необхідні службові поля. Для отримання інформації з бази даних використовувалися ORM-запити Django. Наприклад, для отримання списку всіх оголошень застосовується наступний метод (ліст. 3.6).

Лістинг 3.6 – ORM-запит Django для отримання списку всіх оголошень

```
Listing.objects.all()
```

кінець лістингу 3.6

Для фільтрації оголошень за містом або іншими параметрами використовуються ORM-запити із умовами (ліст. 3.7).

Лістинг 3.7 – ORM-запит Django для фільтрації оголошень за містом

```
Listing.objects.filter(city="Київ")
```

кінець лістингу 3.7

Також у системі використовувалися SQL-запити, які автоматично генеруються Django ORM під час виконання операцій із базою даних. Наприклад, отримання всіх записів із таблиці оголошень відповідає наступному SQL-запиту (ліст. 3.8).

Лістинг 3.8 – SQL-запит для отримання всіх записів таблиці оголошень

```
SELECT * FROM listings_listing;
```

кінець лістингу 3.8

Для вибірки оголошень за певним містом може використовуватися наступний запит (ліст. 3.9).

Лістинг 3.9 – SQL-запит для вибірки оголошень за заданим містом

```
SELECT * FROM listings_listing  
WHERE city = 'Київ';
```

кінець лістингу 3.9

У процесі реалізації бази даних було забезпечено зв'язки між таблицями за допомогою зовнішніх ключів. Зокрема, кожне оголошення пов'язане із конкретним користувачем-орендодавцем, а фотографії – із відповідним оголошенням. Такий підхід дозволяє забезпечити цілісність інформації та правильну організацію структури даних. Для адміністрування бази даних використовувалася вбудована адміністративна панель Django Admin. Вона дозволяє переглядати записи таблиць, додавати нові дані, редагувати оголошення, управляти користувачами та здійснювати модерацію контенту через вебінтерфейс без необхідності прямого редагування бази даних.

Під час проектування бази даних було враховано необхідність забезпечення швидкого доступу до інформації та підтримки фільтрації даних за різними параметрами. Це дозволило реалізувати ефективний пошук житла та забезпечити стабільну роботу вебплатформи навіть при значній кількості оголошень. Отже, у процесі розробки бази даних було створено структуру зберігання інформації, яка забезпечує ефективну роботу вебплатформи для оренди квартир, підтримує взаємозв'язки між основними сутностями системи та дозволяє здійснювати швидку обробку й пошук даних.

3.4 Захист інформаційно-комп'ютерної системи

У процесі розробки вебплатформи для оренди квартир важливу увагу було приділено забезпеченню захисту інформації, безпечній взаємодії користувачів із системою та обмеженню несанкціонованого доступу до даних. Оскільки система працює з персональною інформацією користувачів, даними авторизації, оголошеннями та контактною інформацією, реалізація механізмів інформаційної безпеки є важливою складовою програмного продукту.

У розробленій системі реалізовано механізми автентифікації та авторизації користувачів. Автентифікація використовується для перевірки особи користувача під час входу до системи, тоді як авторизація забезпечує розмежування прав доступу до функціональних можливостей вебплатформи. Неавторизовані користувачі мають доступ лише до перегляду оголошень, тоді як авторизовані користувачі можуть створювати власні оголошення, редагувати інформацію та взаємодіяти із системою від імені власного акаунта. Для реалізації системи автентифікації використовувалися стандартні засоби Django та механізм токенів доступу. Після успішної авторизації користувач отримує спеціальний токен, який зберігається у frontend-частині системи та використовується під час виконання HTTP-запитів до REST API. Це дозволяє перевіряти права доступу користувача при кожному зверненні до серверної частини. Фрагмент реалізації збереження токена користувача наведено у лістингу 3.10.

Лістинг 3.10 – Реалізація механізму збереження токена авторизації

```
const TOKEN_KEY = "apartment_rent_token";

export function getStoredToken() {
  return localStorage.getItem(TOKEN_KEY);
}

export function setStoredToken(token) {
  if (token) localStorage.setItem(TOKEN_KEY, token);
  else localStorage.removeItem(TOKEN_KEY);
}
```

кінець лістингу 3.10

Під час реєстрації нового користувача система виконує перевірку введених даних та створює обліковий запис із використанням засобів Django ORM. Паролі користувачів не зберігаються у відкритому вигляді. Для забезпечення безпеки використовується алгоритм хешування паролів, реалізований у фреймворку Django. Перед збереженням пароль проходить процедуру криптографічного перетворення, у результаті чого в базі даних зберігається лише хеш-значення.

Алгоритм хешування дозволяє уникнути компрометації паролів навіть у випадку отримання доступу до бази даних сторонніми особами. Django автоматично використовує сучасні криптографічні алгоритми хешування та механізми сольового захисту (salt), що підвищує рівень безпеки системи.

Процес авторизації користувача у системі можна представити у вигляді такої послідовності дій:

- 1) користувач вводить логін та пароль;
- 2) frontend надсилає POST-запит до backend-сервера;
- 3) django виконує перевірку облікових даних;
- 4) у разі успішної автентифікації формується токен доступу;
- 5) токен передається frontend-частині;
- 6) frontend зберігає токен у localStorage;
- 7) під час подальших запитів токен передається серверу для перевірки

прав доступу.

Для обміну інформацією між клієнтською та серверною частинами застосовується REST API. Передача даних здійснюється у форматі JSON через HTTP-запити. Використання серверної архітектури Django REST Framework дозволяє контролювати доступ до API та обмежувати виконання певних операцій лише авторизованими користувачами.

Важливим елементом захисту є також перевірка та валідація введених даних. На frontend-рівні реалізовано контроль правильності введення інформації у формах, а на backend-рівні додатково застосовуються serializer-класи Django REST Framework. Це дозволяє уникнути помилкового введення даних,

підвищити стабільність роботи системи та зменшити ризик передачі некоректної інформації до бази даних.

Для захисту інформаційної системи від несанкціонованого доступу також застосовується розмежування функціональних можливостей між різними категоріями користувачів. Частина API-запитів доступна лише після проходження авторизації. Таким чином забезпечується контроль доступу до операцій створення, редагування та видалення оголошень.

Оскільки система розроблена як вебплатформа з використанням React та Django, додатковий рівень захисту забезпечується розділенням frontend та backend частин. Користувач не має прямого доступу до серверної логіки або структури бази даних, що зменшує ризик втручання у внутрішню роботу програмного забезпечення.

Отже, у процесі розробки інформаційно-комп'ютерної системи було реалізовано комплекс базових механізмів захисту даних, які забезпечують безпечну автентифікацію користувачів, захист облікових записів, контроль доступу до функціональних можливостей системи та безпечний обмін інформацією між компонентами вебплатформи.

3.5 Розробка документації для програмного продукту

У процесі розробки вебплатформи для оренди квартир було підготовлено технічну та користувацьку документацію, необхідну для встановлення, налаштування та використання програмного продукту. Наявність документації дозволяє спростити процес розгортання системи, полегшує її подальше супроводження та забезпечує зручність роботи для кінцевих користувачів і адміністраторів системи.

Розроблений програмний продукт являє собою клієнт-серверну вебсистему, яка складається з frontend-частини, реалізованої за допомогою React, та backend-частини, створеної із використанням Django REST Framework. Для

збереження даних використовується база даних SQLite, яка інтегрована у серверну частину системи.

Для стабільної роботи програмного продукту необхідно забезпечити відповідність мінімальним системним вимогам. Серверна частина системи потребує встановлення Python версії 3.10 або вище, Node.js версії 18 або новішої, а також менеджера пакетів `npm`. Для запуску системи необхідно мати не менше 4 ГБ оперативної пам'яті, близько 2 ГБ вільного місця на диску та сучасний веббраузер із підтримкою JavaScript. Рекомендованими браузерами є Google Chrome, Mozilla Firefox та Microsoft Edge.

Перед запуском системи необхідно виконати встановлення серверної частини програмного продукту. Для цього здійснюється клонування або копіювання проєкту, після чого виконується встановлення необхідних бібліотек Python за допомогою менеджера пакетів `pip`. Далі необхідно перейти до каталогу backend-проєкту та виконати команди створення міграцій і структури бази даних (ліст. 3.11).

Лістинг 3.11 – Команди створення структури бази даних

```
python manage.py makemigrations  
python manage.py migrate
```

кінець лістингу 3.11

Після створення бази даних виконується запуск Django-сервера за допомогою відповідної команди (ліст. 3.12).

Лістинг 3.12 – Запуск серверної частини системи

```
python manage.py runserver
```

кінець лістингу 3.12

Frontend-частина системи запускається окремо за допомогою середовища Node.js. Перед запуском необхідно встановити всі залежності React-проєкту (ліст. 3.13).

Лістинг 3.13 – Встановлення залежностей frontend-частини

```
npm install
```

кінець лістингу 3.13

Після завершення встановлення бібліотек виконується запуск React-додатку (ліст. 3.14).

Лістинг 3.14 – Запуск клієнтської частини системи

```
npm run dev
```

кінець лістингу 3.14

У процесі розгортання системи frontend та backend взаємодіють через REST API. Передача інформації між компонентами системи здійснюється у форматі JSON за допомогою HTTP-запитів. Backend-частина обробляє запити користувачів, взаємодіє з базою даних та повертає результати клієнтській частині. Для зручності користувачів у системі реалізовано інтуїтивно зрозумілий інтерфейс із підтримкою навігації між сторінками. Основними функціональними можливостями системи є перегляд каталогу оголошень, використання фільтрів пошуку, перегляд детальної інформації про житло, реєстрація та авторизація користувачів, а також взаємодія із AI-помічником.

Користувацька документація містить опис основних сторінок вебплатформи та принципів роботи із системою. На головній сторінці користувач може перейти до каталогу оголошень або скористатися AI-помічником. У каталозі реалізовано систему фільтрації житла за містом, ціною, кількістю кімнат, типом житла, наявністю меблів та техніки. Для перегляду детальної інформації про квартиру користувач переходить на сторінку конкретного оголошення, де відображаються фотографії, характеристики житла, опис та контактна інформація.

Для авторизованих користувачів система забезпечує можливість роботи з власними оголошеннями. Авторизація здійснюється шляхом введення логіна та пароля, після чого користувач отримує доступ до додаткових функцій системи.

У процесі створення документації також було описано структуру проєкту та основні компоненти програмного забезпечення. Backend-частина містить модулі `models.py`, `views.py`, `serializers.py` та `urls.py`, які відповідають за роботу з базою даних, реалізацію API та маршрутизацію запитів. Frontend-частина включає React-компоненти сторінок, систему маршрутизації React Router та модулі взаємодії з API.

Отже, у результаті виконання даного етапу було сформовано документацію, яка містить опис системних вимог, інструкції щодо встановлення та запуску програмного продукту, опис функціональних можливостей системи та рекомендації щодо роботи із вебплатформою. Це забезпечує можливість ефективного використання, супроводження та подальшого розвитку програмного продукту.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи бакалавра було реалізовано практичну частину вебплатформи для оренди квартир із використанням React та Django REST Framework. У процесі розробки створено frontend і backend частини системи, реалізовано REST API, систему авторизації користувачів, механізми пошуку та фільтрації оголошень, а також структуру бази даних із використанням ORM Django. Для створення інтерфейсу застосовано React і React Router.

У процесі тестування та налагодження перевірено коректність роботи основних функціональних модулів, взаємодію між клієнтською та серверною частинами, а також механізми обробки даних і авторизації користувачів. Окрему увагу приділено захисту інформації шляхом реалізації автентифікації, авторизації та використання токенів доступу. Також було підготовлено документацію для програмного продукту, що містить опис структури системи та інструкції щодо її використання. У результаті виконання розділу створено повнофункціональну вебплатформу для оренди квартир із сучасним інтерфейсом та підтримкою основних функцій онлайн-оренди житла.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було розроблено вебплатформу для оренди квартир із використанням сучасних вебтехнологій React, Django REST Framework та SQLite. У процесі виконання роботи було проведено аналіз сучасного стану систем онлайн-оренди житла, визначено основні проблеми існуючих вебплатформ та обґрунтовано доцільність створення нової інформаційної системи з інтеграцією елементів штучного інтелекту.

Під час дослідження було проаналізовано сучасні тенденції розвитку вебтехнологій у сфері нерухомості та встановлено, що одним із найперспективніших напрямів є використання інтелектуальних механізмів автоматизованого підбору житла, персоналізації сервісів та інтеграції AI-помічників у вебплатформи. Проведений аналіз існуючих аналогів показав, що більшість сучасних систем забезпечують базовий функціонал перегляду та пошуку оголошень, однак мають недостатній рівень автоматизації взаємодії з користувачами та персоналізації рекомендацій.

У межах виконання кваліфікаційної роботи бакалавра було повністю реалізовано поставлені завдання. Зокрема, проведено аналіз сучасного стану вебплатформ для оренди житла, сформовано вимоги до програмного забезпечення, спроєктовано архітектуру інформаційної системи та обґрунтовано вибір технологій розробки.

Перед процесом розробки було створено повнофункціональну інформаційно-комп'ютерну модель, що забезпечує реєстрацію та авторизацію користувачів, перегляд каталогу оголошень, фільтрацію житла за параметрами, перегляд детальної інформації про квартири, створення та редагування оголошень, завантаження фотографій, а також роботу інформаційного помічника для підбору житла.

Для розробки вебплатформи було обрано сучасні технології веброзробки, React для створення клієнтської частини, Python Django та Django REST Framework для реалізації серверної логіки й REST API, а також SQLite для

зберігання даних. Такий вибір технологій дозволив забезпечити зручну взаємодію користувача із системою, швидку обробку запитів, модульність програмного продукту та можливість його подальшого розвитку.

Реалізовано клієнт-серверну архітектуру із використанням REST API для взаємодії між frontend та backend частинами системи. Для зберігання даних використано SQLite та ORM-механізми Django, що дозволило забезпечити ефективну обробку інформації та зручну взаємодію з базою даних.

Проведене тестування програмного забезпечення підтвердило коректність роботи основних функціональних модулів системи. У процесі тестування було перевірено роботу механізмів авторизації, пошуку та фільтрації оголошень, взаємодію із REST API, роботу бази даних та коректність відображення інформації на frontend-частині застосунку. Виявлені під час тестування недоліки були усунені, що дозволило підвищити стабільність та надійність роботи вебплатформи.

Було створено базу даних для зручного зберігання інформації про користувачів, квартири, оголошення та фото. Вона допомогла впорядкувати дані й забезпечити стабільну роботу основних функцій платформи.

Особливу увагу у кваліфікаційній роботі бакалавра приділено питанням інформаційної безпеки. У системі реалізовано механізми автентифікації та авторизації користувачів, використання токенів доступу, хешування паролів та розмежування прав доступу до функціональних можливостей системи. Це дозволило забезпечити базовий рівень захисту персональних даних та безпечну взаємодію користувачів із вебплатформою.

Також було написано документацію для того, щоб користувач міг легко зрозуміти, як запустити та використовувати вебплатформу. У ній коротко описано основні можливості системи й порядок роботи з її функціями.

Результати кваліфікаційної роботи бакалавра можуть бути використані у сфері веброзробки, інформаційних технологій та створення систем електронних сервісів для нерухомості. Окремі програмні модулі та архітектурні рішення

можуть застосовуватись під час створення інших інформаційних систем із клієнт-серверною архітектурою.

Подальший розвиток програмного продукту може бути пов'язаний із інтеграцією повноцінних алгоритмів штучного інтелекту, реалізацією рекомендаційних систем на основі машинного навчання, впровадженням онлайн-бронювання житла, системи повідомлень між користувачами, інтеграцією картографічних сервісів та розширенням функціональних можливостей адміністративної панелі. Також перспективним напрямом є використання хмарних технологій та більш продуктивних систем управління базами даних для забезпечення масштабованості вебплатформи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сервіс оголошень України – купівля/продаж бу та нових товарів, різноманітні послуги на сайті OLX.ua. URL: <https://www.olx.ua/uk/> (дата звернення: 21.01.2026).
2. Airbnb. Помешкання для відпустки, зруби, будинки на пляжі тощо. Airbnb. URL: <https://www.airbnb.com.ua/> (дата звернення: 15.01.2026).
3. Booking.com. URL: <https://www.booking.com/index.uk.html> (дата звернення: 18.01.2026).
4. DIM.RIA – вся нерухомість України. Продаж і оренда будь-якої нерухомості. DOM.RIA.com. URL: <https://dom.ria.com/uk/> (дата звернення: 26.01.2026).
5. Django – The web framework for perfectionists with deadlines. Django. URL: <https://www.djangoproject.com/> (дата звернення: 05.03.2026).
6. Django REST framework – Web APIs for Django. URL: <https://www.django-rest-framework.org/> (дата звернення: 08.03.2026).
7. Flatfy від ЛУН – вся оренда і продаж нерухомості України. URL: <https://flatfy.ua/uk> (дата звернення: 29.01.2026).
8. JSON Data Interchange Format. URL: <https://www.json.org/json-en.html> (дата звернення: 16.03.2026).
9. MDN Web Docs – HTTP request methods. URL: <https://developer.mozilla.org/> (дата звернення: 19.03.2026).
10. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/> (дата звернення: 23.04.2026).
11. Npm – Home. URL: <https://www.npmjs.com/> (дата звернення: 02.04.2026).
12. Pallavi L., Rishika D., Sree G. B. S., Ruthwik J., Yashaswini J. SmartRent-AI Driven Online House Rental Platform. 2025 5th International Conference on Soft Computing for Security Applications (ICSCSA). IEEE, 2025. P. 346-353.

13. Python Programming Language. URL: <https://www.python.org/> (дата звернення: 09.04.2026).
14. React – JavaScript library for building user interfaces. React. URL: <https://react.dev/> (дата звернення: 17.02.2026).
15. React Router Documentation. React Router. URL: <https://reactrouter.com/> (дата звернення: 21.02.2026).
16. REST API Tutorial. REST API Tutorial. URL: <https://restfulapi.net/> (дата звернення: 13.03.2026).
17. Singh N. T., Kumar S., Shrivastava U., Wadhwa M., Shrimal V. M. A Study of the Evolution of the Real-Estate Flats and Apartment Rental System. 2024 5th International Conference on Smart Electronics and Communication (ICOSEC). IEEE, 2024. P. 1846-1851.
18. SQLite Home Page. URL: <https://www.sqlite.org/> (дата звернення: 11.03.2026).
19. Tailwind CSS – Rapidly build modern websites. URL: <https://tailwindcss.com/> (дата звернення: 26.02.2026).
20. Visual Studio Code – Code Editing. Redefined. URL: <https://code.visualstudio.com/> (дата звернення: 26.03.2026).