

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система з підбору об'єктів нерухомості»**

Виконала:

студентка IV курсу, групи ІС-13
Коваленко Катерина Миколаївна

Керівник:

Доцент кафедри ІСТ, к.т.н.,
Сперкач Майя Олегівна

Рецензент:

Професор каф ОТ, д.т.н., доцент,
Клименко Ірина Анатоліївна

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студентці

Коваленко Катерині Миколаївні

1. Тема проєкту «Інформаційна система з підбору об'єктів нерухомості», керівник проєкту Сперкач Майя Олегівна, доцент кафедри ІСТ, к.т.н., затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «9» червня 2025 р.
3. Вихідні дані до проєкту: Технічне завдання, середовище розробки Visual Studio Code, платформа Node.js, фреймворк Express.js, бібліотека React.js, система керування базами даних PostgreSQL, середовище розгортання Render.
4. Зміст пояснювальної записки: проаналізувати предметну область та існуючі онлайн-платформи для пошуку житла, сформулювати вимоги до функціональності системи, побудувати архітектуру застосунку, обґрунтувати вибір технологій, спроектувати базу даних, описати математичне забезпечення, реалізувати вебзастосунок, протестувати готову систему та оцінити результати.
5. Перелік графічного матеріалу: структурна схема системи, даталогічна модель бази даних, діаграма прецедентів, інфологічна модель бази даних.
7. Дата видачі завдання «8» березня 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Визначення теми дипломного проєкту та погодження з дипломним керівником	08.03.2025	
2.	Складання ТЗ	19.04.2025	
3.	Аналіз предметної області	26.04.2025	
4.	Розробка функціональної моделі	03.05.2025	
5.	Огляд існуючих аналогів	10.05.2025	
6.	Формування цілей і задач розробки	17.05.2025	
7.	Розробка дизайну ПЗ	27.05.2025	
8.	Дослідження і розробка математичного забезпечення	04.05.2025	
9.	Розробка програмного забезпечення	05.06.2025	
10.	Тестування ПЗ	08.06.2025	

Студентка

Катерина КОВАЛЕНКО

Керівник

Майя СПЕРКАЧ

АНОТАЦІЯ

Система з підбору об'єктів нерухомості.

Проект містить 63 с. тексту, 23 рисунки, 8 таблиць, посилання на 18 літературних джерел, додатки та 4 конструкторських документів.

ПІДБІР ЖИТЛА, АНАЛІЗ ДАНИХ, ОНЛАЙН-ПЛАТФОРМА, ІНФОРМАЦІЙНА СИСТЕМА.

Об'єктом розробки є інформаційна система з підбору об'єктів нерухомості.

Метою розробки є спрощення процесу підбору житла для оренди шляхом автоматизації збору та оцінки оголошень, виявлення шахрайських записів і формування персоналізованих рекомендацій для користувача.

У дипломному проекті розроблено веборієнтовану інформаційну систему, що об'єднує дані з декількох платформ, формує єдину базу актуальних оголошень, фільтрує їх за критеріями та ранжує за релевантністю. В основі аналітичної частини – прості моделі машинного навчання: для виявлення ймовірності шахрайства та оцінки відповідності об'єкта.

Робота складається з основних розділів. У вступі обґрунтовано актуальність теми, визначено мету, задачі, об'єкт і предмет дослідження. Перший розділ описує предметну область та постановку задачі. Другий – аналіз існуючих рішень, їхні переваги, недоліки та порівняння функціональності. Третій розділ містить вимоги до системи. Четвертий – вибір технологій і інструментів. П'ятий – архітектуру системи, структуру бази даних, клієнтську і серверну частини, сценарії використання та результати тестування. Шостий – математичне забезпечення з описом моделей машинного навчання та алгоритмів. У висновках підсумовано результати, практичну значущість і перспективи розвитку.

Розроблена система може бути використана для підвищення ефективності взаємодії користувачів з сервісами оренди житла, а також як основа для подальшого розвитку рекомендаційних платформ у сфері нерухомості.

SUMMARY

Information system for the selection of real estate objects.

The project contains 63 pages of text, 23 figures, 8 tables, references to 18 literary sources, annexes, and 4 design documents.

Keywords: real estate, housing selection, data analysis, online platform, information system.

The object of development is an information system for real estate selection. The purpose of the development is to simplify the process of searching for rental housing by automating the collection and evaluation of advertisements, detecting fraudulent listings, and generating personalized recommendations for users.

The graduation project presents a web-based information system that integrates data from several platforms into a unified database of up-to-date listings. It includes filtering options based on user-defined criteria and ranking mechanisms based on relevance. The analytical core is based on basic machine learning models used to identify potential fraud and evaluate listing suitability.

The work consists of main chapters. The introduction substantiates the relevance of the topic and defines the goals, objectives, object, and subject of the study. The first chapter describes the subject area and problem statement. The second chapter analyzes existing solutions, their advantages, disadvantages, and compares functionalities. The third chapter details system requirements. The fourth chapter covers the choice of technologies and tools. The fifth chapter presents the system architecture, database structure, client and server parts, use cases, and testing results. The sixth chapter contains mathematical support with descriptions of machine learning models and algorithms. The conclusion summarizes the results, practical significance, and development prospects.

The developed system can be used to improve the efficiency of user interaction with rental services and serves as a foundation for further development of recommendation platforms in the real estate domain.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC13.050БАК.005 ПЗ	Інформаційна система з	63		
6			підбору об'єктів нерухомості.			
7			Пояснювальна записка			
8						
9	A3	IC13.050БАК.005 Д1	Інформаційна система з	1		
10			підбору об'єктів нерухомості.			
11			Діаграма прецедентів			
12						
13	A3	IC13.050БАК.005 Д2	Інформаційна система з	1		
14			підбору об'єктів нерухомості.			
15			Інфологічна модель бази даних			
16						
17	A3	IC13.050БАК.005 Д3	Інформаційна система з	1		
18			підбору об'єктів нерухомості.			
19			Даталогічна модель бази даних			
20						
21	A3	IC13.050БАК.005 Д4	Інформаційна система з	1		
22			підбору об'єктів нерухомості.			
23			Структурна схема системи			
24						
25						
26						
Зм.	Лист	№ докум.	Підпис			
Розробив		Коваленко К.М.				
Перевірив		Сперкач М.О.				
Затв.						
IC13.050БАК.005 ТП						
Інформаційна система з підбору об'єктів нерухомості.				Літ.	Арк.	Аркушів
				Т	1	1
Відомість дипломного проєкту				КПІ ім. Ігоря Сікорського Група IC-13		

**Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система з підбору об'єктів
нерухомості»**

Київ – 2025 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	7
1.1 Опис процесу діяльності	7
1.2 Постановка задачі	8
1.2.1 Призначення розробки	9
1.2.2 Цілі та задачі розробки.....	9
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	12
2.1 Онлайн-платформа OLX Недухомість	12
2.2 Сервіс Dom.gia	14
2.3 Платформа LUN	15
2.4 Мобільний застосунок Bird.....	16
2.5 Порівняльний аналіз функціональності платформ	17
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	20
3.1 Вимоги до системи в цілому	20
3.2 Вимоги до функціональних характеристик	21
3.3 Вимоги до видів забезпечення.....	24
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ.....	28
5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	33
5.1 Структура системи	33
5.2 Функціональна модель системи.....	34
5.3 Інфологічна модель бази даних	35
5.4 Модель бази даних	36
5.5 Передавання та обробка даних	39

					ІС13.050БАК.005 ПЗ			
Зм.	Лист	№ докум.	Підпис		Інформаційна система з підбору об'єктів нерухомості. Пояснювальна записка			
Розробив	Коваленко К.М.							
Перевірив	Сперкач М.О.							
Затв.								
					Літ.	Арк.	Аркушів	
					Т		2	63
					КПІ ім. Ігоря Сікорського Група ІС-13			

5.6 Архітектура програмного забезпечення.....	40
5.6.1 Специфікація функцій.....	40
5.6.2 Структурна схема системи.....	43
5.7 Тестування системи.....	43
6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	50
6.1 Змістовна постановка задачі	50
6.2 Математична постановка задачі	50
6.3 Обґрунтування методу розв'язання.....	52
6.4 Опис методу розв'язання	53
6.5 Приклад застосування математичної моделі	55
ВИСНОВКИ	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	—	База даних.
СУБД	—	Система управління базами даних.
API	—	Application Programming Interface.
CRUD	—	Create Read Update Delete.
CSV	—	Comma-Separated Values.
HTTP	—	HyperText Transfer Protocol.
HTTPS	—	HyperText Transfer Protocol Secure.
JSON	—	JavaScript Object Notation.
ML	—	Machine Learning.
MAE	—	Mean Absolute Error.
Node.js	—	Node JavaScript.
NoSQL	—	Not Only SQL.
PDF	—	Portable Document Format.
REST	—	Representational State Transfer.
SQL	—	Structured Query Language.
TOPSIS	—	Technique for Order Preference by Similarity to Ideal Solution.
UI	—	User Interface.
WSM	—	Weighted Sum Model.

ВСТУП

У сучасному світі, де доступ до інформації стає все легшим, пошук та оренда житла здійснюється найпростішим способом – через онлайн-платформи. Однак з їх розвитком з'явилися й нові виклики для шукачів, серед яких найсерйознішим є шахраї та недобросовісних орендодавців. Користувачі часто можуть стикатися з неякісними оголошеннями, фальшивими пропозиціями та несправжніми відгуками. Такі ситуації можуть спричинити серйозні фінансові збитки, оскільки оренда є найсуттєвішою статтею витрат бюджету більшості українців, та негативний користувацький досвід, що, у свою чергу, підриває довіру до онлайн-платформ загалом.

Актуальність теми підтверджується аналітичними даними, які свідчать, що у 2024 році попит на оренду житла в Україні перевищив рекордні показники 2023 року та збільшився на 8,4%. Таке зростання зумовлене внутрішньою міграцією, змінами місця роботи та зростаючою потребою у власному житлі [1].

Об'єктом дослідження даної роботи є процеси збору, обробки та аналізу даних на платформах для підбору житла для оренди, що включають технічні аспекти функціонування: пошук, перегляд, підбір, вибір та оцінка оголошень, а також інформаційні фактори, що впливають на достовірність представленої інформації. Увага також приділяється ризикам, що пов'язані з дезінформацією, яка може впливати на ефективність прийняття рішень користувачами.

Предметом дослідження є методи та інструменти побудови інформаційної системи для підтримки процесу підбору житла для оренди, з урахуванням індивідуальних критеріїв користувача, якості та достовірності даних. Це включає:

- автоматизований збір та обробку даних з різних джерел;
- формалізацію критеріїв пошуку житла;
- побудову системи оцінювання оголошень за різними параметрами;
- розробку моделей для виявлення підозрілих шаблонів.

Метою дослідження є спрощення процесу підбору житла для оренди, підвищення ефективності взаємодії користувачів з онлайн-платформами, а також

зниження ризиків, пов'язаних із отриманням недостовірної або неповної інформації. Передбачається використання сучасних методів автоматизованого збору даних та оцінки достовірності інформації.

Задля того, щоб досягти цієї мети передбачено виконання таких завдань:

- аналіз потреб користувачів та специфіки роботи ринку оренди житла в інформаційному просторі;
- дослідження вже наявних сучасних інформаційних систем та технологічних рішень;
- організація процесу збору, очищення та структурування даних з релевантних онлайн-ресурсів;
- реалізація інформаційної системи, яка відповідає меті роботи;
- проведення експериментального дослідження ефективності та відповідності результатів користувацьких запитів;
- забезпечення підтримку подальшого розвитку системи, включно з можливістю масштабування та інтеграції з іншими сервісами.

Інформаційна система розроблена задля того, щоб стати зручним та ефективним інструментом для користувачів-орендаторів, що передбачає такі практичні значення:

- надання зручних інструментів персоналізованого вибору варіантів нерухомості;
- забезпечення точною інформацією, що відповідає критеріям пошуку та потребам;
- зменшення ймовірності натрапити на недостовірне оголошення;
- підвищення загальної довіри до онлайн-платформ.

Результатом розробки є спрощення процесу оренди житла та зниження ризиків для користувачів та підвищено ефективність взаємодії з платформами.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел із 17 найменувань, 1 додаток. Графічна частина включає 4 кресленики формату А3. Загальний обсяг 64 сторінки.

1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА

1.1 Опис процесу діяльності

У сучасних умовах оренда житла є однією з найпоширеніших потреб населення, особливо в умовах внутрішньої міграції, переїздів або пошуку житла для навчання чи роботи. Основним способом пошуку варіантів оренди стали онлайн-платформи (OLX, Dom.ria, Rieltor.ua тощо), які надають можливість швидкого доступу до великої кількості оголошень.

Попри доступність інформації та зручність пошуку, ринок оренди житла в Україні стикається з низкою проблем:

- низька якість оголошень через відсутність та неповноту опису, розмиті та неактуальні фотографії, неточні дані;
- шахрайство та фейкові оголошення;
- велика кількість дублікатів;
- значні часові витрати на перегляд та аналіз оголошень з декількох джерел.

Процес охоплює декілька етапів, починаючи від пошуку та перегляду оголошень до заселення, схему можна переглянути на рисунку 1.1.

Перший етап полягає з перегляду усіх популярних сайтів, таких як OLX, Dom.ria, Rieltor.ua. Потенційні орендарі вводять свої критерії пошуку, такі як ціна, розташування, площа, кількість кімнат, умови оренди. Інтерфейс зазвичай передбачає зручність фільтрації варіантів за вибраними критеріями та дає змогу одразу виключити з пошуку оголошення, що не задовольняють вимогам.

Другий етап це перегляд того, що пропонує платформа та передбачає детальне вивчення наданої інформації. Відбувається оцінка за наданими фотографіями, умовами оренди, а також пошук відгуків попередніх орендарів в інтернеті. Цей етап займає найбільше часу, адже досить часто оголошення розміщуються на одразу в декількох місцях і одну й ту саму інформацію необхідно аналізувати пару разів.

У разі відповідності вимогам, орендар переходить до комунікації з власником оголошення з метою уточнення додаткових деталей, таких як умови проживання,

					IC13.050БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

наявність додаткових послуг, можливість перегляду житла, строки заселення або інші аспекти майбутнього договору.

Останній етап передбачає безпосереднє заключення умови та заселення.

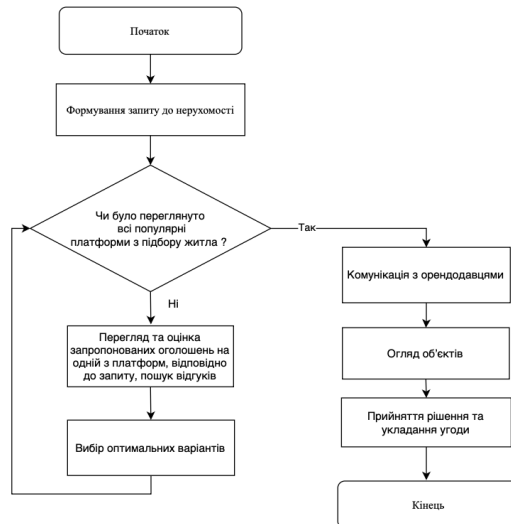


Рисунок 1.1 — Схема процесу прийняття рішення стосовно вибору квартири

Схема діаграми діяльності інформаційної системи, що пропонується в рамках даної роботи, наведена на рисунку 1.2.

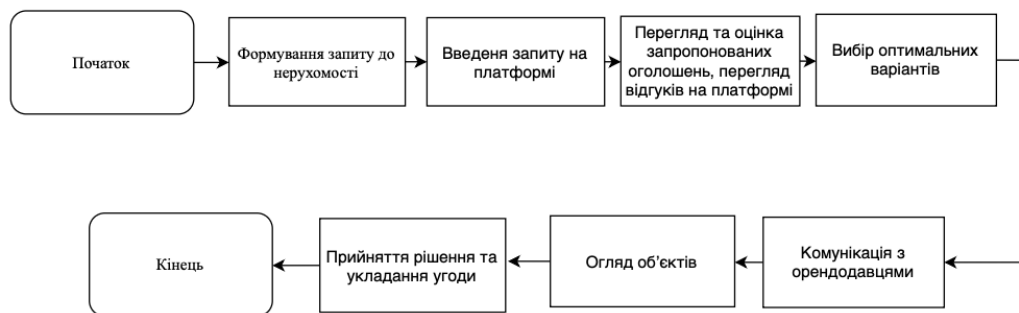


Рисунок 1.2 — Схема прийняття рішення з розроблюваною системою

Ця схема ілюструє, як розроблювана система підтримує процес пошуку нерухомості, допомагаючи усунути основні проблеми традиційних методів, зменшити ризик помилкових рішень та значно підвищити швидкість і ефективність прийняття рішень користувачем.

1.2 Постановка задачі

Користувач має необхідність знайти житло для проживання у великому незнайомому місті. Він змушений аналізувати десятки оголошень на різних вебресурсах. Через наявність дублікатів і повторюваних записів, значна частина інформації постійно повторюється та призводить до втрати часу та зниження ефективності. Для релевантного формування списку пропозицій зазвичай вводять базові критерії пошуку: тип житла – квартира, місцезнаходження, вартість та інше.

Для досягнення мети дослідження та реалізацію було обрано задачу, яка передбачає створення інформаційної системи з автоматизації процесу пошуку, фільтрації та оцінки оголошень про оренду нерухомості з онлайн-ресурсів.

1.2.1 Призначення розробки

Інформаційна система призначена для автоматизації оренди передбачає:

- централізація збору даних із декількох вебресурсів з виявленням дублікатів, що дозволяє мати єдину базу актуальних оголошень;
- формування персоналізованих рекомендацій на основі фільтрації за критеріями та найбільшої відповідності;
- виявлення потенційно шахрайських пропозицій на основі виявлення підозрілих шаблонів, повторюваності або невідповідної ціни;
- забезпечення можливості залишати відгуки та оцінку оголошенням.

1.2.2 Цілі та задачі розробки

Цілями даної інформаційної системи є:

- оптимізація процесу пошуку житла, шляхом швидкого доступу до актуальних оголошень у зручному та нативному форматі;
- підвищення ефективності та якості прийняття рішень, наданням точних, персоналізованих рекомендацій, що базуються на основі заданих критеріїв;

					IC13.050БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

– забезпечення достовірності оголошень та зниження ймовірності взаємодії з шахрайськими пропозиціями;

– гнучкість та масштабованість системи, що передбачає створення адаптивної архітектури, здатної до інтеграції з іншими сервісами задля підвищення функціональності відповідно до змін ринку нерухомості.

Розробка інформаційної системи передбачає виконання низки взаємопов'язаних задач. Першою задачею є організація збору та обробки даних, що включає:

- розробити модулі автоматичного збір пропозицій з різних платформ оренди;
- провести аналіз первинної перевірки зібраних даних на достовірність;
- розробити алгоритм очищення та структурування даних для подальшого аналізу та відображення.

Наступною та основною задачею є реалізація персоналізованого пошуку, що потребує:

- реалізувати інтерфейс введення критеріїв пошуку (тип житла, площа, ціна, кількість кімнат, місто тощо);
- забезпечити обробку введених даних.

Третя задача це забезпечення виведення результатів, що передбачає:

- реалізувати модуль відображення знайдених об'єктів у вигляді карток;
- додати візуалізацію об'єктів на карті.

Четвертою задачею є підвищення зручності користувачів за допомогою багатокритеріальної фільтрації оголошень, що включає:

- розробити механізм багатокритеріальної фільтрація результатів пошуку, відповідно до потреб користувача;
- оптимізувати процес ранжування результатів пошуку.

П'ятою задачею є оцінка якості оголошень, що охоплює:

- розробити функціонал для збору та відображення відгуків користувачів;
- провести аналіз вмісту та структури оголошень для виявлення підозрілих записів.

					ІС13.050БАК.005 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

Висновки до розділу 1

У даному розділі було розглянуто предметну область, пов'язану з процесом пошуку та оренди житла. Описано ключові етапи, які проходить користувач, зокрема: пошук за критеріями, фільтрація, оцінка варіантів і прийняття рішень. Також було виявлено основні проблеми, з якими стикаються користувачі.

Дипломний проєкт орієнтований на розробку інформаційної системи, яка автоматизує процес збору, аналізу та оцінювання оголошень про оренду нерухомості. Така система має забезпечити не лише централізовану обробку даних з різних джерел, а й надання користувачеві актуальності, достовірності та зручності для сприйняття інформації, сприяючи прийняттю ефективних рішень щодо вибору оптимального варіанту.

Запропонований підхід дає змогу підвищити якість прийняття рішень, забезпечити прозорість у виборі житла та зменшити ризик взаємодії з недобросовісними орендодавцями. Водночас система має залишатися простою у використанні, швидкою та адаптованою до реальних сценаріїв пошуку житла у великому місті.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

У сучасному цифровому світі існує чимало популярних як українських онлайн-платформ для підбору нерухомості відповідно до заданих параметрів, якими користуються тисячі людей. Далі будуть наведені приклади вже готових наявних рішень, які мають подібні функції або схожі компоненти.

2.1 Онлайн-платформа OLX Нерухомість

OLX Нерухомість є однією з наймасовіших платформ для розміщення інформації для оренду житла, тому про неї буде зазначено більше. Сервіс дозволяє фільтрувати пропозиції за базовими критеріями, переглянути пошукову систему сайту можна на рисунку 2.1, такими як ціна, розташування, кількість кімнат, площа, тип житла тощо.

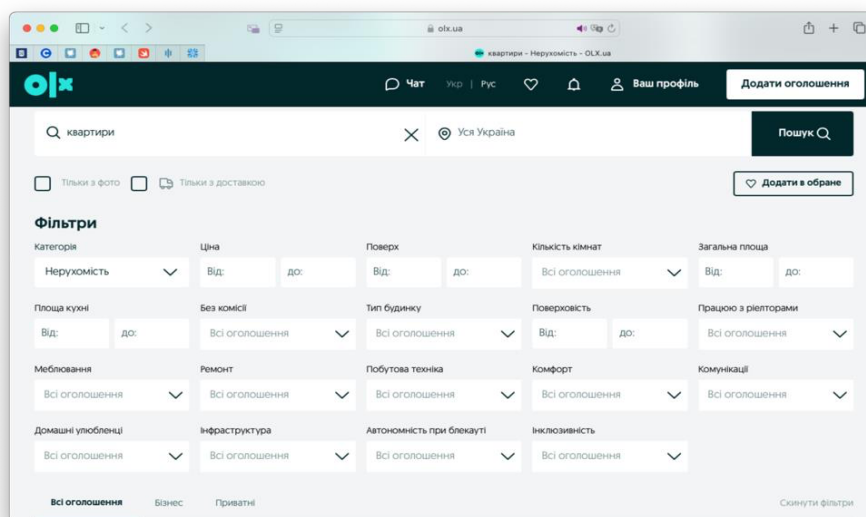


Рисунок 2.1 — Вкладка пошуку сайту OLX

Проте серед недоліків платформи можна вважати велику кількість фейкових та неактуальних пропозицій, неправдиві відомості про квартири та ціни, а також чимало однакових пропозицій [2]. Єдиний спосіб перевірки надійності

орендодавців це рейтинг, який можна переглянути на рисунку 2.2, але з’являється він за умови успішно виконаної угоди на цьому сайті.

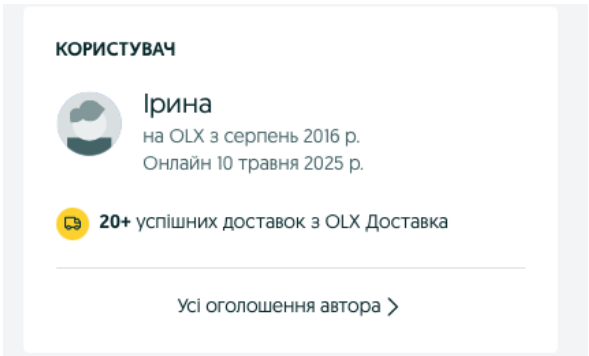


Рисунок 1.2 — Рейтинг користувача

Як вже і зазначалося досить часто можна натрапити на пропозиції, які не відповідають дійсності. Наприклад, на рисунку 2.3 представлено оголошення квартири з гарним описом та фотографіями.

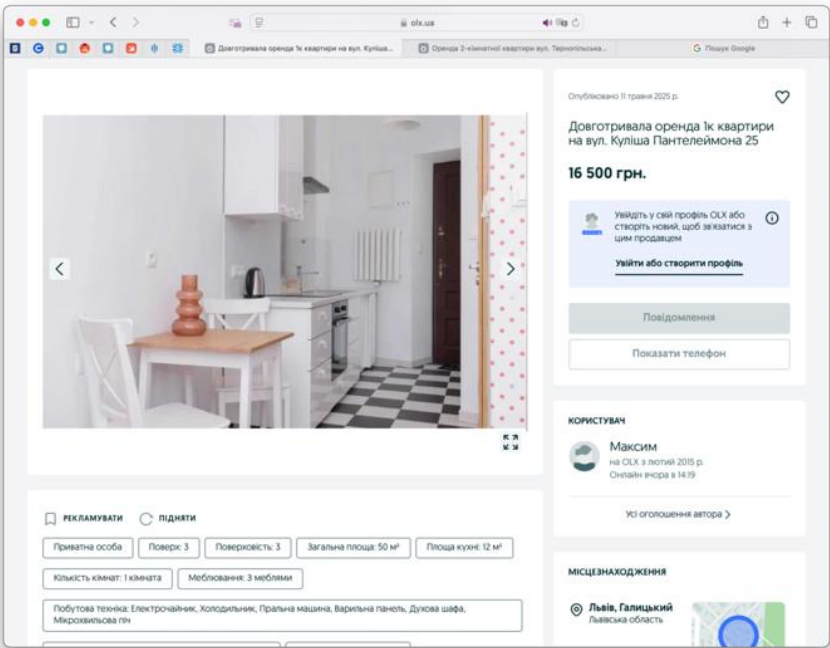


Рисунок 2.2 — Приклад оголошення

Але якщо здійснити пошук фотографією, можна знайти пропозиції з ідентичними фото, але об’єктів, що розташовані за іншими адресами, рисунок 2.4,

або знаходяться навіть у інших країнах. Що може ввести в оману людей, які потребують житло терміново та не мають достатньо часу або досвіду для перевірки інформації.

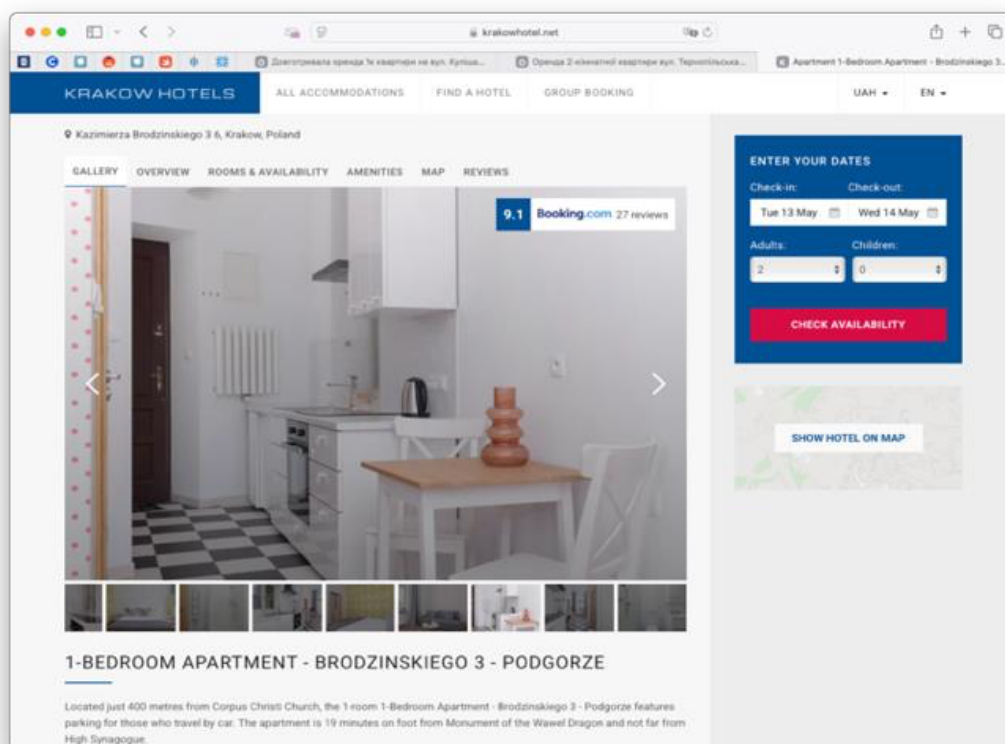


Рисунок 2.4 — Оголошення

У результаті користувачі можуть втратити кошти, натрапити на неякісні умови проживання, а також багато інших негативних наслідків.

2.2 Сервіс Dom.ria

Dom.ria пропонує користувачам вже більш розширені можливості для пошуку житла, включаючи панорамні фото у 360 градусів, інтерактивні карти та систему верифікації оголошень, рисунок 2.5. Завдяки цим функціям користувач отримує більше інформації про об'єкт ще до перегляду наживо, що значно підвищує зручність та ефективність вибору. Платформа також впровадила підтвердження особи через Дія.Підпис, що сприяє підвищенню рівня довіри до

оголошень та мінімізує ризики шахрайства. Це робить Dom.ria однією з найнадійніших платформ на українському ринку оренди житла.

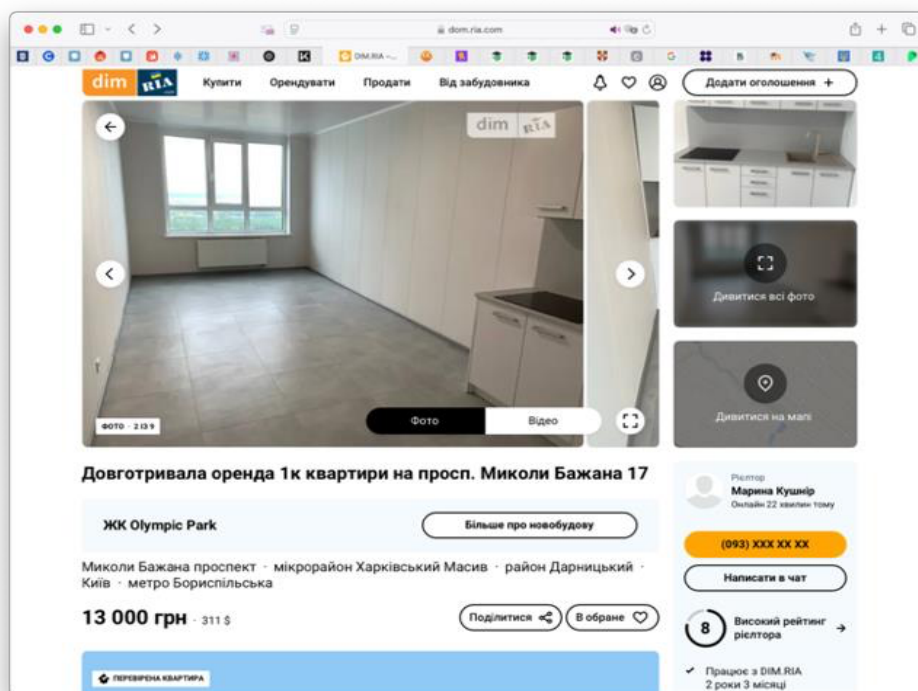


Рисунок 2.5 — Dom.ria

Однак серед недоліків також варто зазначити обмежену кількість нових оголошень у порівнянні з OLX, що зменшує вибір користувачів, а також наявність неактуальної інформації, яка може вводити в оману, а також відсутності можливості залишати попереднім орендарам відгуки та фото.

2.3 Платформа LUN

ЛУН є агрегатором оголошень про нерухомість, що об'єднує пропозиції з різних джерел, що дозволяє користувачам отримувати максимально повну інформацію в одному місці. Платформа пропонує зручні та гнучкі фільтри для пошуку, які охоплюють всі основні параметри, що важливі для вибору житла, такі як ціна, площа, кількість кімнат, район, тип нерухомості та інші. На інтерактивній карті, представленій у рисунку 2.6, об'єкти відображаються з точним географічним

розташуванням, що допомагає оцінити інфраструктуру навколо та зручність добирання. Крім того, ЛУН надає користувачам розгорнуту інформацію про забудовників та новобудови.

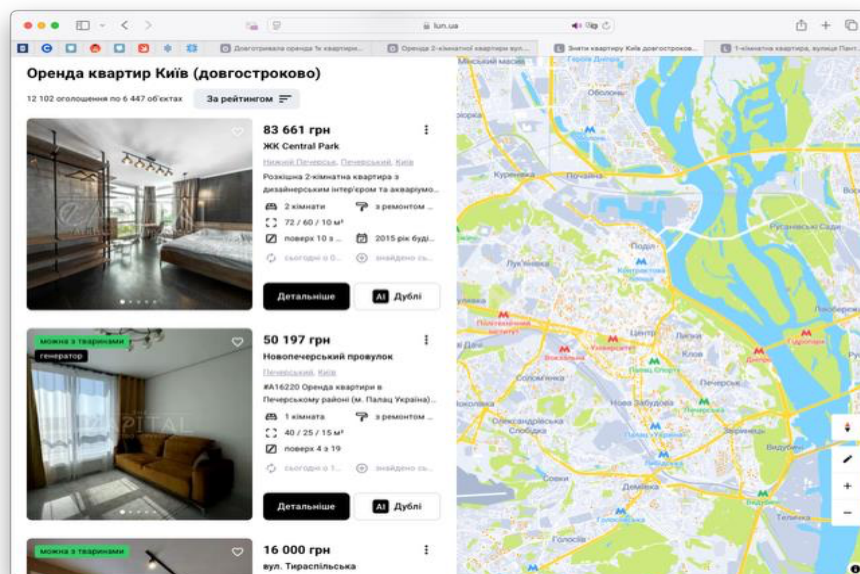


Рисунок 2.6 — ЛУН

До недоліків можна віднести неактуальні дані та фейкові оголошення з інших платформ, хоча команда платформи регулярно проводить оновлення та перевірку інформації, повністю виключити такі випадки не вдається. А також варто відмітити скарги користувачів про «запозичення» оголошень — коли одна й та сама пропозиція нерухомості дублюється кількома орендаторами або публікується без їх згоди.

2.4 Мобільний застосунок Bird

Це мобільний застосунок, який збирає оголошення про оренду житла з понад 24 різних вебсайтів, що дозволяє користувачам бачити широкий вибір пропозицій. Застосунок використовує алгоритми штучного інтелекту для автоматичної перевірки оголошень на наявність можливих шахрайств і низькоякісного контенту. Це допомагає покращити якість інформації, що відображається, та зменшити

кількість недостовірних пропозицій. На рисунку 2.7 представлений інтерфейс застосунку, який дозволяє здійснювати пошук за різними параметрами, переглядати фото та порівнювати варіанти. Серед всіх перерахованих альтернатив ця є найбільш наближеною до мети дипломного проєкту.

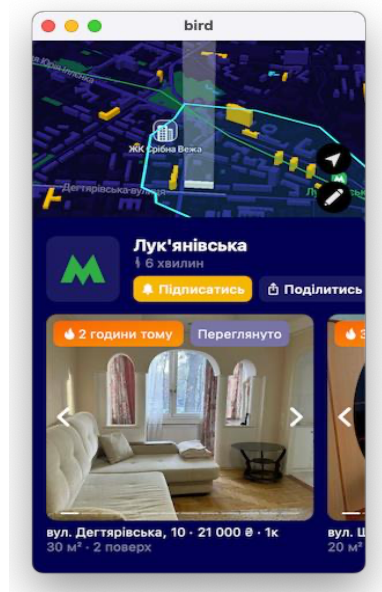


Рисунок 2.7 — Bird

Але платформа поки передбачає невелику кількість міст та немає можливості залишати відгуки. Також у мережі можна знайти негативні відгуки про Bird щодо оголошень, що не відповідають дійсності: вони або неактуальні, або можна натрапити на шахраїв. Це знижує довіру до платформи і створює додаткові ризики для орендарів, особливо тих, хто покладається виключно на онлайн-ресурси при пошуку житла.

2.5 Порівняльний аналіз функціональності платформ

Для комплексного виявлення як сильних сторін, так і недоліків існуючих онлайн-платформ, що спеціалізуються на пошуку та аналізі оголошень про оренду житла, було проведено детальний порівняльний аналіз за ключовими функціональними аспектами. Основна мета такого аналізу — оцінити, наскільки

сучасні сервіси відповідають вимогам користувачів, зокрема у питаннях зручності, безпеки та точності інформації, його результати можна переглянути в таблиці 2.1.

Основними критеріями порівняння стали:

- агрегація даних – збір оголошень з кількох джерел;
- фільтрація за критеріями – наявність зручного механізму пошуку за параметрами;
- перевірка оголошень – аналіз шахрайських оголошень;
- штучний інтелект – використання інтелектуальних методів аналізу або рекомендацій;
- відгуки орендаторів – можливість залишати або переглядати реальні відгуки про орендодавців або житло.

Таблиця 2.1 — Порівняльний аналіз аналогічних платформ

Платформа	Агрегація даних	Фільтрація за критеріями	Перевірка оголошень	ШІ	Відгуки орендаторів
OLX.ua	-	+	-	-	-
Dom.ria	-	+	+	-	-
LUN.ua	+	+	+	-	-
Bird	+	+	+	+	-

Як видно з таблиці, жодна з проаналізованих платформ не пропонує повного набору функцій, здатних повністю задовольнити потреби користувачів у безпечному, ефективному та прозорому пошуку житла. Особливо помітним є відсутність систем виявлення підозрілих оголошень із застосуванням штучного інтелекту, а також механізмів відкритих відгуків користувачів про досвід оренди, що знижує довіру та ускладнює прийняття обґрунтованих рішень.

Це створює передумови для розробки нової інформаційної системи, яка б поєднувала найкращі існуючі функціональні можливості та доповнювала їх інтелектуальними сервісами перевірки, аналізу та персоналізованих рекомендацій, забезпечуючи більш високий рівень безпеки, зручності та ефективності для користувачів.

Висновки до розділу 2

Аналіз існуючих онлайн-платформ для оренди житла в Україні показує, що незважаючи на їх популярність та доступність, жоден із сервісів не забезпечує повного та комплексного вирішення ключових завдань користувача – швидкого, безпечного та персоналізованого підбору житла. Це підтверджує актуальність вибраної теми та створює потребу в розроблюваній інформаційній системі. Для себе відмічено як і технології, які можна застосувати в застосунку, так і проблеми, які потребують вирішення.

Окрему увагу було приділено питанням організації зручного користувацького інтерфейсу, оскільки зручність та інтуїтивність використання системи є ключовими факторами залучення і утримання користувачів. Інтерфейс повинен бути адаптивним, швидким та містити інструменти для персоналізації пошуку, що сприяє більш точному задоволенню індивідуальних потреб. Крім того, важливим аспектом є забезпечення прозорості і зрозумілості процесів, що формують рекомендації, що підвищує довіру до системи.

Отримані у цьому розділі висновки формують міцну основу для подальших етапів розробки — від проектування архітектури системи до її реалізації та тестування. Вони допомагають чітко визначити пріоритети у функціоналі, технологічному стеку та дизайні, що сприятиме створенню ефективного, надійного та зручного інструменту для пошуку житла на сучасному ринку.

3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

3.1 Вимоги до системи в цілому

Інформаційна система складається з трьох основних підсистем: пошуку та фільтрації, аналітики і третьої – агрегації та обробки даних.

Підсистема пошуку та фільтрації має забезпечити інтуїтивно зручний інтерфейс для користувача з гнучким налаштуванням параметрів пошуку оголошень. Це передбачає здійснення фільтрації за ключовими критеріями: ціною, кількістю кімнат, типом житла, площею, наявністю додаткових зручностей. Підсистема також реалізує базовий текстовий пошук, розширений пошук з фільтрами та підтримує сортування результатів за датою та іншими параметрами.

Підсистема аналітики полягає в оцінці житла, виявленні шахрайських оголошень та рекомендації оптимальних варіантів відповідно до запиту користувача. Забезпечує візуалізацію цих даних та звіти отриманих результатів.

Підсистема агрегації та обробки даних відповідає за автоматизований збір даних з зовнішніх джерел, зокрема онлайн-платформ оренди нерухомості. Вона забезпечує парсинг сторінок, виконання API запитів та періодичну синхронізацію з базами даних. Отримані дані попередньо проходять фільтрацію, нормалізацію структури записів та збереження в форматі, який буде зручним для подальшої обробки та відображення.

Основними аварійними ситуаціями, для яких регламентуються вимоги надійності:

- втрата зв'язку з сервером;
- втрата зв'язку з базою даних;
- непередбачені збої під час роботи програмного забезпечення.

Вимоги до надійності технічних засобів передбачають забезпечення стабільного функціонування системи в умовах щоденного інтенсивного використання, мінімізацію ризику втрати або пошкодження даних, а також здатність оперативно відновлювати ключові сервіси після виникнення технічних збоїв, перебоїв у живленні чи збою в роботі серверного обладнання. Крім того,

система має бути здатною до безперервної роботи у фоновому режимі з автоматичним контролем стану та діагностикою несправностей.

Програмне забезпечення обов'язково має включати інструменти для оперативного відновлення інформації при її втраті та гарантувати збереження критичних даних за:

- аварійного завершення роботи серверу;
- збоїв під час парсингу або запису у базу даних;
- оновленнях чи перезапуску системних сервісів.

Також для забезпечення надійного збереження даних передбачається щоденне автоматичне резервне копіювання всієї бази даних 1 раз на добу у вечірній час, орієнтовно о 23.00, у період мінімального навантаження на систему. Створені копії зберігаються принаймні ще декілька днів в хмарному середовищі задля швидкого відновлення у разі потреби.

3.2 Вимоги до функціональних характеристик

На момент написання дипломного проєкту передбачається лише один актор, а саме користувач, який займається пошуком нерухомості. Усі функціональні можливості системи орієнтовані на задоволення його потреб.

Основні функціональні сценарії взаємодії користувача із системою проаналізовано у вигляді діаграми прецедентів, що представлена в кресленнику IC13.050БАК.005 Д1. Вона відображає ключові можливості системи.

У майбутньому планується можливість авторизації користувачів та розподіл на ролі: орендодавець або ріелтор та майбутній орендатор.

Система повинна забезпечувати автоматизацію основних процесів, пов'язаних з пошуком, аналізом та оцінюванням оголошень про оренду житла. До основних комплексів задач, що підлягають автоматизації, належать:

- автоматизований пошук оголошень;
- фільтрація оголошень за критеріями;
- сортування результатів пошуку;

- оцінка якості оголошення;
- виявлення потенційно шахрайських або підозрілих оголошень;
- формування рейтингу оголошення;
- можливість додавання відгуків і оцінок;
- формування персоналізованих рекомендацій;
- відображення статистики та аналітики;
- автоматичний збір даних із зовнішніх джерел;
- очищення, структурування зібраної інформації та збереження в базі;
- оновлення та видалення застарілих оголошень;
- експорт результатів пошуку або аналітики у форматі Excel, або PDF;
- логування подій та повідомлення про помилки.

Більш детальну інформацію про функціональні характеристики системи, їхні вимоги та взаємозв'язки наведено у таблиці 3.1, що дозволяє чітко уявити очікуваний функціонал і забезпечити послідовну реалізацію.

Таблиця 3.1 — Деталізація функціональних вимог з пріоритетністю

Варіант використання	№	Функціональні вимоги	Пріоритет
Агрегація даних	1	Система збирає дані із зовнішніх джерел	Високий
	1.1	Система перевіряє нові записи, порівнюючи з існуючою базою даних	Високий
	1.2	Система забезпечує парсинг полів: ціну, локацію, опис, фото	Високий
	1.3	Система зберігає попередньо нормалізовані дані з оголошень в єдину базу даних	Високий
	1.4	Система видаляє оголошення, коли воно втрачає свою актуальність	Високий
Аналітика	2	Система розраховує рейтинг житла на основі ключових параметрів	Високий

Варіант використання	№	Функціональні вимоги	Пріоритет
	2.1	Система оновлює рейтинг при зміні даних	Високий
Пошук	4	Система забезпечує фільтрацію оголошень за основним параметром	Високий
	4.1	Система підтримує також розширений пошук	Середній
	4.2	Система надає можливість користувачу здійснити пошук та отримати результат пошуку	Високий
Безпека	5	Система виявляє підозрілі оголошення	Високий
	5.1	Система позначає оголошення, які є потенційно шахрайськими та потребують додаткової перевірки	Високий
Відгуки та оцінки	6	Система дозволяє користувачам залишати відгуки та оцінки оголошень	Високий
	6.1	Система відображає середній оголошення на основі відгуків	Високий
	6.2	Система передбачає можливість прикріплювати до відгуків фотографії та відео	Високий
	6.3	Система надає можливість оцінити відгук як корисний іншим користувачам	Високий
	6.4	Система фільтрує відгуки за датою та рейтингом	Середній
Експорт даних	7	Система дозволяє експортувати результати пошуку у формати Excel або PDF	Середній

Варіант використання	№	Функціональні вимоги	Пріоритет
Робота з картами	8	Система позначає маркером оголошення на карті	Високий
	8.1	Система інтегрує інтерактивні карти з позначеними результатами пошуку	Середній
	8.2	Система підтримує пошук оголошень у межах певного радіусу на карті	Середній
	8.3	Система відмічає переглянуті оголошення на карті	Низький
Логування	9	Система веде лог подій: помилок, успішних операцій для для детального аналізу роботи.	Високий

3.3 Вимоги до видів забезпечення

Етап проєктування інформаційної системи потребує попереднього формалізованого визначення вимог, які слугують основою для уявлення про очікуваний результат, а також забезпечення відповідності поставленим цілям розробки, досягнення необхідного рівня функціональності, ефективності, зручності та надійності використання системи кінцевими користувачами.

Математичне забезпечення інформаційної системи це сукупність математичних моделей, алгоритмів та методів, які забезпечують реалізацію основних функціональних можливостей системи, зокрема для обробки, аналізу отриманих даних. До основних вимог належать:

- використання типових алгоритмів для обробки і нормалізації великих масивів зібраних даних про нерухомість, фільтрації та сортування оголошень, виявлення дублікатів;
- застосування методів машинного навчання для виявлення аномалій та потенційно шахрайських пропозицій;

- алгоритми повинні бути адаптивними до змін у вхідних даних та масштабованими;
- алгоритми мають забезпечити високу продуктивність і стабільну роботу системи.

Інформаційне забезпечення розроблюваної системи – це комплексне рішення, щодо складу, структури, формату і організації роботи з даними, а також механізмів взаємодії між різними компонентами системи. Воно також має стати основою для ефективного функціонування програмної частини.

До основних вимог до інформаційного забезпечення належать:

- система має забезпечити структуровану організацію даних та їх цілісність за будь-яких умов;
- зберігання даних має бути в централізованому сховищі з чітко визначеною структурою та взаємозв'язками між об'єктами даних;
- система має підтримувати різні формати даних, а також їх резервне копіювання та швидке відновлення;
- система має бути адаптивною до змін функціоналу, потреб масштабування, а також забезпечити достовірність та точність інформації.

Програмне забезпечення системи це насамперед набір інструментів, бібліотек та середовищ необхідних для розробки, розгортання та експлуатації інформаційної системи. У дипломному проєкті будуть використовуватися виключно ліцензійні програмні продукти, або open-source.

Основними вимогами до програмного забезпечення є:

- надійні та перевірені технології розробки, такі як JavaScript (React), Node.js;
- системи керування базами даних, які підтримують масштабованість та зберігання великих об'ємів інформації, наприклад, PostgreSQL;
- підтримка кросплатформеності та сумісності з поширеними веббраузерами;
- підтримка стабільної роботи програмних модулів.

					IC13.050БАК.005 ПЗ	Арк.
						25
Зм.	Лист	№ докум.	Підпис	Дата		

Технічне забезпечення користувача для розроблюваної системи є досить мінімальним, оскільки вона реалізована як веборієнтований застосунок і не потребує встановлення спеціалізованого програмного забезпечення на стороні клієнта. Доступ буде здійснюватися через стандартні браузері, що забезпечує широку сумісність та зручність у використанні.

Система підтримується на будь-якій операційній системі як Windows, Linux, MacOS, а також Android та iOS для користування на мобільних пристроях, або планшетах.

Основними вимогами є:

- стабільне підключення до Інтернету;
- Google Chrome 80+;
- Mozilla Firefox 78+;
- Microsoft Edge 80+;
- Safari 13+ (для iOS та MacOS).

Система доступна для пристроїв навіть з обмеженими технічними характеристиками, для широкого кола користувачів. Для отримання стабільної та надійної роботи рекомендовано використовувати саме останні стабільні версії браузерів.

Функціональні характеристики.

Технічні засоби системи повинні забезпечувати повноцінну підтримку всіх основних функцій для ефективної роботи, включаючи підтримку сучасних протоколів, стандартів веброзробки, безпекових механізмів а також масштабованість, розширення кількості користувачів, обсягу даних та нових функцій без суттєвих змін у базовій архітектурі..

Експлуатаційні характеристики.

Технічна частина системи має забезпечувати стабільну безперебійну роботу з мінімальною потребою технічного втручання, швидке реагування на запити користувачів навіть при високому навантаженні, а також цілодобову доступність системи. Крім того, важливою вимогою є забезпечення надійного збереження даних, захисту персональної інформації користувачів та стійкості до збоїв.

					IC13.050БАК.005 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

Висновки до розділу 3

У третьому розділі було детально сформовано систему вимог до функціонування інформаційної системи пошуку та аналізу оголошень про оренду житла. Було визначено основні підсистеми, що охоплюють як обробку вхідних даних, так і управління, планування, аналіз результатів. Всі ці компоненти узгоджено між собою і спрямовані на досягнення спільної мети – створення надійного, адаптивного та зручного інструменту для підтримки користувача у прийнятті рішень.

Також у розділі представлено перелік функціональних вимог до системи, які охоплюють ключові можливості: збір і збереження даних з різних джерел, нормалізація і фільтрація, формування прогнозів, сортування за релевантністю, виявлення потенційних шахрайських оголошень, робота з аналітикою та вивід результатів у зручній формі. Функціональність системи охоплює як клієнтську, так і серверну частини, з чітко визначеними зонами відповідальності. Було враховано питання надійності, обробки аварійних ситуацій, взаємодії з базою даних, а також передбачено заходи щодо захисту від непередбачуваних відмов.

Окрему увагу було приділено питанням масштабованості, стабільності та логічної побудови архітектури, що дозволяє реалізувати систему як сукупність взаємопов'язаних, але незалежних модулів, які легко тестуються та підтримуються. Використані сучасні підходи до формування вимог забезпечують узгодження технічних рішень з бізнес-цілями, знижують ризики на етапі реалізації та спрощують перехід до моделювання та розробки.

Результати розділу закладають чітке підґрунтя для наступного етапу – побудови архітектури системи та реалізації її логіки. Чітко визначені вимоги дозволяють не лише сформувати ефективну програмну модель, а й адаптувати її до реальних умов використання, що особливо важливо в динамічному середовищі орендного ринку великих міст. Це сприяє підвищенню якості обслуговування користувачів платформи.

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

На етапі розробки будь-якої інформаційної системи ключовим моментом є вибір програмних засобів та інструментів на основі функціональних вимог. Це необхідно, щоб готовий продукт міг ефективно реалізувати поставлені цілі, був зручним та надійним у використанні. Вибрані технології мають забезпечити легкість розробки та обслуговування, а також гнучкість для подальшого розвитку.

Цей розділ містить обґрунтування вибору мов, середовищ, бібліотек, а також допоміжних інструментів, що використовувалися для створення веборієнтованого застосунку. Розглянуто переваги кожного з вибраних технологічних компонентів, їхню сумісність, продуктивність, відповідність вимогам проєкту, а також зручності для розробки.

Вибір основних технологій для розробки системи, а також компоненти для яких вони було обрані можна переглянути у таблиці 4.1.

Таблиця 4.1 — Вибір технологій для розробки

Компонент	Обрана технологія
Клієнтська частина (frontend)	React.js
Серверна частина (backend)	Node.js з Express
Інтерфейс користувача	Tailwind CSS
Робота з картою	Mapbox GL JS
База даних	PostgreSQL
Виявлення шахрайства та ранжування даних	TensorFlow.js
Мова програмування	JavaScript
Системи контролю версій	Git + GitHub
Середовище розробки	Visual Studio Code

Для розробки інформаційної системи було обрано мову JavaScript [3], оскільки це універсальний та адаптований інструмент, який є основою сучасної веброботи та інших програм. JavaScript підтримується усім сучасними браузерами, активно розвивається та має велику кількість бібліотек та фреймворків для створення як клієнтської, так і серверної частини застосунків різних рівнів складності. Завдяки V8 JavaScript забезпечує високу продуктивність на сервері.

Як основне середовище розробки було вибрано Visual Studio Code – це простий та потужний редактор, який підтримує значну кількість мов програмування, розширень та налаштувань. Має вбудований термінал, автодоповнення коду та інтеграцію з Git та JavaScript.

React.js є сучасною JavaScript-бібліотекою, орієнтованою на розробку динамічних інтерфейсів для вебзастосунків. Спочатку створена компанією Meta (раніше Facebook), ця технологія швидко здобула популярність завдяки своїй ефективності, масштабованості та широкій підтримці серед спільноти розробників. Основною ідеєю React є розбиття інтерфейсу на автономні компоненти, які відповідають за виведення конкретних елементів інтерфейсу [4].

Перевагами є:

- потужні бібліотеки з великою кількістю корисних компонентів;
- компонентна архітектура;
- Virtual DOM.

У межах даного проєкту бібліотека використовується для реалізації інтерактивного відображення списків нерухомості, оперативного оновлення контенту за фільтрами користувача та створення зручного адаптивного UI [5].

Node.js – це середовище виконання JavaScript, яке працює поза браузером та дозволяє створювати серверну логіку застосунку [6]. Його побудовано на основі високошвидкісного рушія V8, що забезпечує швидку обробку запитів [7]. Express, як надбудова над Node.js, є мінімалістичним фреймворком, що надає засоби для маршрутизації, обробки HTTP-запитів і створення API [8]. Це поєднання стало оптимальним для розробки серверної частини проєкту, оскільки дозволяє швидко

організувати взаємодію з клієнтом, обробку запитів до БД та запуск аналітичних модулів [9].

Перевагами є:

- асинхронна та подієво-орієнтована архітектура;
- мінімалізм;
- велика екосистема;
- підтримка маршрутизації та REST API.

Для проєсту було вибрано через високу продуктивність при багатьох одночасних запитах, простоту інтеграції з фронтендом та швидкої розробки API.

PostgreSQL – це одна з найпотужніших систем управління базами даних з відкритим вихідним кодом. Вона забезпечує надійне зберігання, гнучку обробку та ефективне управління великими наборами структурованих і напівструктурованих даних. PostgreSQL підтримує стандарт SQL і водночас надає можливості NoSQL, зокрема роботу з JSON, що робить її універсальним інструментом для розробки сучасних систем [10].

Перевагами є:

- реляційна модель;
- підтримка JSON;
- багатий функціонал та гнучкість;
- сумісність з багатьма інструментами;
- розширені типи даних.

Tailwind CSS – це сучасний CSS-фреймворк, побудований на концепції класів, які описують окремі стилі. Такий підхід дозволяє будувати інтерфейси без написання кастомних CSS-файлів. Розробник безпосередньо задає стилі через класи в HTML, що суттєво пришвидшує верстку, спрощує підтримку та забезпечує адаптивність.

Цей інструмент використовується в проєкті для формування зручного і привабливого візуального оформлення інтерфейсу користувача [11].

TensorFlow.js є сучасною бібліотекою для виконання моделей машинного навчання на мові JavaScript, яка була створена в межах проєкту TensorFlow

					IC13.050БАК.005 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

компанією Google. Її основна перевага полягає в тому, що вона дає змогу запускати як вже навчені моделі, так і створювати нові безпосередньо у браузері користувача або у серверному середовищі Node.js. Це забезпечує значну гнучкість і знижує вимоги до додаткових ресурсів, оскільки все обчислення може виконуватись на стороні клієнта або у хмарі.

TensorFlow.js активно підтримує імпорт моделей з Python-середовища, що полегшує інтеграцію вже існуючих рішень у вебінтерфейс. Окрім того, бібліотека дозволяє здійснювати повний цикл розробки – від підготовки та обробки даних до побудови, тренування та валідації моделей [12].

Перевагами цієї бібліотеки є:

- універсальність, адже можна моделі запускати як на сервері, так у браузері;
- легка інтеграція;
- велика кількість вже готових моделей;
- гнучкість у побудові архітектури;
- швидкі обчислення.

У дипломному проєкті TensorFlow.js використовується для реалізації моделей машинного навчання, зокрема для виявлення шахрайських оголошень та прогнозування рейтингу об'єктів оренди. Забезпечує аналітику в реальному часі без потреби у складних серверних обчисленнях.

Mapbox GL JS – це інструмент для інтерактивного відображення карт у браузері за допомогою WebGL. На відміну від статичних карт, він дозволяє працювати з векторними даними у режимі реального часу, змінювати стилі та додавати інтерактивні елементи [13].

У даному застосунку ця бібліотека слугує основою для відображення локацій об'єктів нерухомості, а також забезпечує зручну навігацію та візуалізацію просторових характеристик.

Перевагами є:

- рендеринг 2D, 3D графіки в браузері без потреби у плагінах;
- гнучка стилізація;
- інтерактивність;

– інтеграція з іншими сервісами роботи з картами (маршрутизація, матриця відстаней).

І останнє – Git. Для керування вихідним кодом використовується система контролю версій Git, разом з GitHub та GitHub Desktop. Завдяки Git є можливість фіксувати кожну зміну у коді, зберігати історію змін, а також повертатися до будь-яких попередніх версій проєкту. Це особливо важливо для відстеження помилок, аналізу прогресу розробки та забезпечення цілісності коду. Платформа GitHub забезпечує централізоване сховище коду з онлайн-доступом, що сприяє коментарями, а також інтеграції з іншими інструментами та автоматизації. Використання GitHub Desktop робить роботу з Git інтуїтивно зрозумілою.

Висновки до розділу 4

У цьому розділі було детально проаналізовано доступні технології, інструменти та фреймворки з метою формування оптимального стеку для реалізації інформаційної системи пошуку та аналізу оголошень про оренду житла.

Було обґрунтовано вибір клієнтської, серверної та базової інфраструктури з урахуванням вимог до безпеки, масштабованості, зручності для кінцевого користувача та легкості у розгортанні. Усі компоненти системи узгоджено між собою та мають високий рівень сумісності, що забезпечує цілісність і стабільність у процесі розробки та подальшого супроводу. Особливу увагу приділено можливості гнучкої інтеграції із зовнішніми сервісами та API, підтримці з боку спільноти, а також наявності документації, що дає змогу розширювати функціонал відповідно до змін ринку та потреб користувачів. Обраний підхід дозволяє знизити вартість підтримки, підвищити швидкість оновлень і забезпечити адаптивність.

Отже, сформовано надійну технологічну платформу, яка є основою для подальших етапів реалізації проєкту. Рішення, прийняті на цьому етапі, визначають успішність реалізації функціональних вимог, стабільність системи та її готовність до зростання.

5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

5.1 Структура системи

Система передбачає собою клієнт-серверну архітектуру, адже зараз майже всі вебсайти та інтернет-сервіси побудовані саме на її основі. Схему [14] зображено на рисунку 5.1.

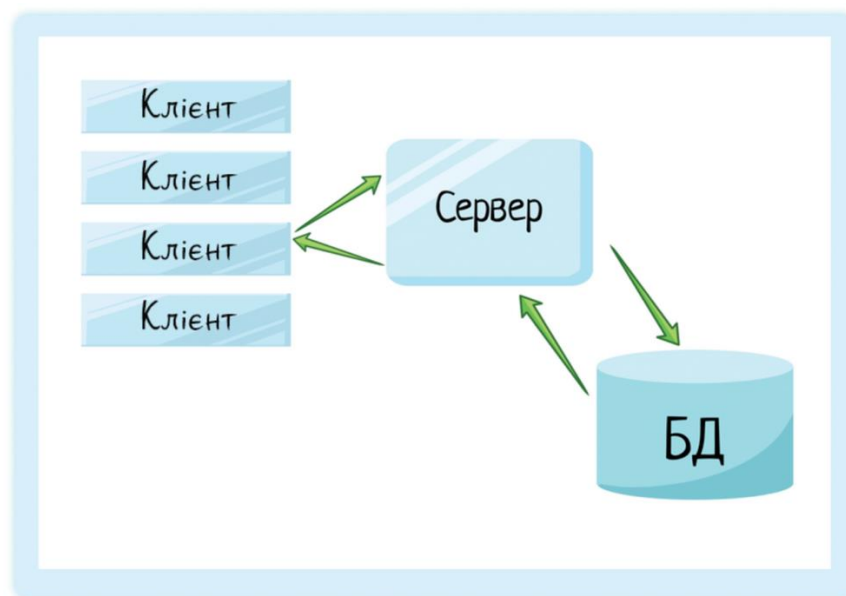


Рисунок 5.1 — Схема вибраної архітектури

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів.

Перевагами є:

- використання одного потужного сервера замість великої кількості клієнтських машин дозволяє забезпечити хорошу продуктивність програми, кілька серверів можуть знадобитися тільки при великих навантаженнях, або забезпечення безперебійної роботи системи, навіть за аварійних ситуацій, але їх кількість буде значно меншою, ніж клієнтів;

- основний код програми зберігається на сервері, тоді як клієнтська частина відповідає за візуалізацію та прості операції;

– прості користувачі не мають доступу до зайвої інформації, що забезпечує безпеку та конфіденційність даних.

Клієнтська частина відповідає за інтерфейс взаємодії користувача з системою. Вона реалізує функціонал, що працює безпосередньо у браузері користувача: динамічне відображення інформації, реакція на дії користувача, відправлення запитів до сервера, валідація форм та інше [15].

Для розробки клієнтської частини використано наступні технології.

React.js – популярна бібліотека JavaScript для створення інтерактивних користувацьких інтерфейсів. Вона забезпечує компонентний підхід, що дозволяє будувати гнучкі, масштабовані та підтримувані вебдодатки.

Tailwind CSS – утилітарний CSS-фреймворк, який дозволяє швидко та гнучко оформлювати елементи інтерфейсу за допомогою готових класів, забезпечуючи сучасний та адаптивний дизайн.

Серверна частина реалізує бізнес-логіку системи, обробку запитів клієнта, перевірку прав доступу, взаємодію з базою даних, а також забезпечує безпечне зберігання та обробку інформації.

Основні технології для серверної частини перераховано далі.

Node.js + Express – серверне середовище виконання JavaScript, що дозволяє будувати масштабовані та високопродуктивні сервери. Express.js є легковагим фреймворком, який забезпечує зручне створення RESTful API, маршрутизацію та обробку запитів.

PostgreSQL – сучасна об’єктно-реляційна система керування базами даних з відкритим вихідним кодом. Забезпечує надійне, зберігання структурованих даних та підтримує розширені функції, як-от JSON, геодані, пошук тощо.

5.2 Функціональна модель системи

У вимогах було вже зазначено, що в межах дипломного проєкту передбачається лише один актор – користувач. Це означає, що розподілу на ролі та

права доступу не відбувається і кожен, хто користується системою має рівний доступ до функціональних можливостей, що можна переглянути на рисунку 5.2.

А саме:

- здійснювати пошук оголошень;
- уточнювати критерії пошуку;
- переглядати запропоновані результатів у вигляді карток із короткою інформацією;
- переходити до детальної інформації про вибраний об’єкт нерухомості;
- відкривати первинне оголошення на сторонній платформі;
- залишати та переглядати відгуки про оголошення;
- переглядати рейтинг оголошення та рекомендації системи.

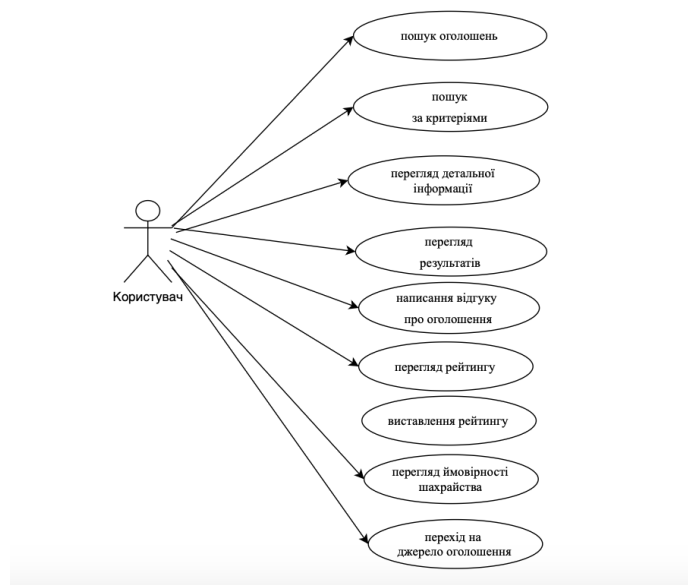


Рисунок 5.2 — Діаграма варіантів використання

5.3 Інфологічна модель бази даних

Інфологічна модель даних – узагальнений неформальний опис створюваної бази даних, виконаний з використанням природної мови, математичних формул, таблиць, графіків і інших засобів, зрозумілих всім людям, які працюють над проектуванням бази даних [16].

Це перший етап проєктування бд. Спочатку формулюється логічна модель, яка відображає основні сутності системи, їх властивості та взаємозв'язки без деталізації тимчасових аспектів фізичного зберігання даних чи конкретних SQL-схем. А вже далі – фізична модель, що враховує обрану СУБД та оптимізації зберігання й обробки даних.

У межах дипломного проєкту була створена інфологічна модель бази даних, яка включена до графічних матеріалів проєкту: кресленик IC13.050БАК.005 Д2.

5.4 Модель бази даних

База даних є основою функціонування інформаційної системи з підбору об'єктів нерухомості. Вона забезпечує централізоване зберігання, швидкий пошук, аналіз та оновлення інформації про об'єкти нерухомості, користувачів, запити, відгуки та інші сутності. Архітектура бази даних побудована за принципами реляційної моделі, що забезпечує логічну структурування інформації, нормалізацію таблиць та підтримку цілісності даних.

У базі даних реалізовано 6 таблиць-сутностей, що покривають усі основні процеси – зберігання оголошень, критерії відбору, оцінки, аналітичні показники, відгуки та повідомлення. Нижче у таблиці 5.1 наведено опис основних таблиць.

Таблиця 5.1 — Опис таблиць бази даних

№	Назва таблиці	Опис
1	Property	Таблиця для зберігання основної інформації про оголошення: унікальний ідентифікатор, назву, опис, адресу, місто і тд. для зручного та доступного відображення даних у застосунку
2	PropertyImage	Таблиця для зберігання URL-зображення для кожного оголошення
3	Review	Таблиця для зберігання відгуків та дані користувачів про конкретні оголошення
4	ReviewImage	Таблиця для зберігання додаткових фотографій, прикріплених до відгуків

№	Назва таблиці	Опис
5	Evaluation	Таблиця для зберігання інформації про рейтинг оголошення
6	FraudReport	Таблиця для зберігання інформації про підозру на шахрайство

Всі сутності розробленої бази даних між собою пов'язані через зовнішні ключі, що відображає їх логічні зв'язки. Основним вузлом є таблиця Property, яка зберігає всі текстові дані картки-оголошення про нерухомість, яка надається користувачеві. А з інших таблиць збирається уся інша інформація, яку більш детально можна переглянути в таблиці 5.2 з описом полів. Зображення зберігаються у таблиці PropertyImage, відгуки та оцінки користувачів з прикріпленими фотографіями з таблиць Review та ReviewImage, які впливають на рейтинг, який відображається для користувачів (таблиця Evaluation). З таблиці FraudReport відображається інформація про шахрайство. Усі ці таблиці взаємодіють між собою, забезпечуючи цілісність даних та дозволяючи формувати повну, структуровану картину кожного оголошення для подальшого аналізу, фільтрації та виведення на інтерфейс користувача. Така організація бази даних не лише забезпечує зручність доступу до пов'язаної інформації, а й спрощує розширення функціоналу системи в майбутньому без порушення цілісності даних.

Таблиця 5.2 — Опис полів баз даних

Таблиця	Поле	Тип	Опис
Property	id	INTEGER	Ідентифікатор
	title	VARCHAR(255)	Назва або короткий заголовок оголошення
	description	TEXT	Детальний опис нерухомості
	address	VARCHAR(255)	Адреса (вулиця, номер будинку)
	city	VARCHAR(100)	Місто, в якому розташована нерухомість
	price	NUMERIC(12,2)	Ціна оренди за місяць
	rooms	INT	Кількість кімнат

Таблиця	Поле	Тип	Опис
	type	VARCHAR(50)	Тип об'єкту
	source	VARCHAR(50)	Джерело оголошення (OLX, Dom.ria)
	link	VARCHAR(500)	URL-адреса, за якою можна переглянути оригінал оголошення
PropertyImage	id	INTEGER	Ідентифікатор
	property_id	INT (FK)	Зовнішній ключ для таблиці оголошень
	image_url	VARCHAR(500)	URL-адреса фотографій оголошенням
Review	id	INTEGER	Ідентифікатор
	property_id	INT (FK)	Зовнішній ключ для таблиці оголошень
	rating	DECIMAL(2,1)	Оцінка оголошення коментатора
	user	VARCHAR(255)	Ім'я користувача, який залишив коментар
	useremail	CITEXT	Пошта користувача, який залишив коментар
ReviewImage	id	INTEGER	Ідентифікатор
	review_id	INT (FK)	Зовнішній ключ для таблиці з відгуками
	image_url	VARCHAR(500)	URL-адреса фотографій оголошенням
Evaluation	id	INTEGER	Ідентифікатор
	property_id	INT (FK)	Зовнішній ключ для таблиці оголошень
	relevance_score	DECIMAL(2,1)	Рейтинг оголошення
FraudReport	id	INTEGER	Ідентифікатор
	property_id	INT (FK)	Зовнішній ключ для таблиці оголошень
	fraud_probability	DECIMAL(2,1)	Ймовірність шахрайства

Між таблицями встановлено такі зв'язки:

- Property та PropertyImage мають зв'язок 1 до N (один до багатьох), оскільки одне оголошення Property може мати багато зображень PropertyImage. У таблиці PropertyImage є поле property_id як зовнішній ключ на Property;
- Property та Review мають зв'язок 1 до N (один до багатьох), оскільки одне оголошення Property може мати багато відгуків Review. У таблиці Review є поле property_id як зовнішній ключ на Property;
- Review та ReviewImage мають зв'язок 1 до N (один до багатьох), оскільки одне оголошення Review може мати багато зображень ReviewImage. У таблиці ReviewImage є поле review_id як зовнішній ключ на Review;
- Property та Evaluation мають зв'язок 1 до 1 (один до одного) для кожного оголошення існує поточний розрахований рейтинг. Така сама залежність у основної таблиці з FraudReport. Зовнішній ключ так само поле property_id.

Всі зв'язки можна переглянути в даталогічній моделі бази даних, яка включена до графічних матеріалів проєкту: кресленик IC13.050БАК.005 ДЗ.

5.5 Передавання та обробка даних

У системі з підбору об'єктів нерухомості обробка та передавання даних реалізовані за допомогою архітектури клієнт-сервер. Обмін інформацією між клієнтською частиною (React.js) та серверною частиною (Node.js з Express) відбувається через REST API з використанням формату JSON, що дозволяє ефективно передавати структуровані дані, а також дозволяє легко інтегрувати сервіси машинного навчання або штучного інтелекту.

Типовий цикл передачі даних виглядає наступним чином:

- користувач на фронтенді взаємодіє з інтерфейсом, наприклад, задає фільтри для пошуку нерухомості;
- ці параметри у форматі JSON відправляються запитом на відповідний маршрут Express-сервера;

					IC13.050БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		39

- серверна частина приймає запит, обробляє його, виконує відповідні SQL-запити до бази вже проаналізованих даних моделю і формує відповідь згідно результатам;
- відповідь із результатами (наприклад, список об'єктів нерухомості) повертається у форматі JSON назад на клієнт, де вона візуалізується користувачу.

5.6 Архітектура програмного забезпечення

При створенні інформаційної системи для підбору нерухомості було застосовано сучасний SPA-підхід (Single Page Application), який передбачає завантаження всього необхідного інтерфейсу у вигляді єдиного HTML-документа при першому запиті. Така архітектура дозволяє системі динамічно оновлювати контент у реальному часі без необхідності повного перезавантаження сторінки під час роботи користувача.

Використання SPA-підходу значно покращує взаємодію з системою, забезпечуючи миттєву реакцію на дії користувача та плавні переходи між різними екранами додатка. Крім того, ця технологія зменшує кількість запитів до сервера, оскільки основна частина логіки обробки даних та рендерингу інтерфейсу виконується на стороні клієнта.

Така архітектура не лише підвищує продуктивність системи, але й спрощує процес її подальшого обслуговування та розширення функціоналу, оскільки дозволяє розробникам працювати з окремими модулями незалежно один від одного.

5.6.1 Специфікація функцій

У межах розробленої інформаційної системи передбачено повний набір функцій, що охоплюють збір та агрегацію даних із зовнішніх джерел, їх фільтрацію за заданими критеріями, оцінювання релевантності та виявлення потенційного шахрайства, а також інтерактивну візуалізацію на карті. Усі обчислювальні

					IC13.050БАК.005 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

операції виконуються в реальному часі за допомогою алгоритмів на сервері та в браузері, з можливістю подальшого навчання моделей машинного навчання. Для прискорення повторних запитів система використовує кешування результатів на боці сервера і клієнта, а також централізоване логування всіх подій. Код реалізації системи можна переглянути у репозиторії за посиланням у додатку А, рисунок А.1.

Таблиця 5.3 узагальнює призначення кожної функції, перелік її вхідних параметрів та формат вихідних даних.

Таблиця 5.3 — Специфікація функцій

Функція	Опис	Вхід	Вихід
getAll()	Завантажує повний список оголошень із бази, при наявності фільтрів ще й застосовує їх	filters (city?, price?, rooms?, area?, type?)	Список об'єктів Property
getById()	Повертає одне оголошення за id	propertyId – числовий ідентифікатор оголошення	JSON-об'єкт Property із полями images, fraudProbability, relevanceScore.
getOlxProperties()	Агрегує OLX, повертає сирі дані	-	<Array<OlxProperty>>, де кожен об'єкт містить { title, price, link, location, description, area, rooms, city, images }
getDomRiaProperties()	Агрегує DOM.RIA	-	<Array<DomRiaProperty>>, з аналогічними параметрами як в OLX
getLunProperties()	Агрегує LUN.ua	-	<Array<LunProperty>>>>, з аналогічними параметрами як в OLX
useRealEstate()	Клієнтський хук для фільтрів, завантаження й кешування	-	Об'єкт { properties, loading, filters, updateFilters }, де properties – масив поточних оголошень, loading – булевий прапорець під час очікування

Функція	Опис	Вхід	Вихід
			відповіді, filters – поточні значення фільтрів, updateFilters – функція для їх зміни і повторного запиту.
getComments()	завантажує всі відгуки (Review) для вказаного оголошення	propertyId – числовий ідентифікатор оголошення	Масив відгуків, кожен з яких містить { id, user_name, user_email, rating, comment }
getReviewImages()	Повертає усі зображення, прикріплені до відгуку	reviewId – числовий ідентифікатор відгуку	Масив фотографій відгуку із полями { id, url }
normalize()	Лінійна нормалізація	v, min, max	Число в межах [0,1]
haversine()	Відстань між координатами	[lon,lat], [lon2,lat2] – два масиви координат	Відстань між координатами у км
predictFraud()	Детектор шахрайства на основі класичної логістичної регресії	Property – (об'єкт з його атрибутами)	Число в межах [0,1] (ймовірність шахрайства)
detectFraudTF()	Інференс TensorFlow.js-моделі	Property – (об'єкт з його атрибутами)	Число в межах [0,1]
scoreProperty()	«скорер» на базі лінійної зваженої регресії для ранжування оголошень користувача.	Property – (об'єкт з його атрибутами)	Рейтинг оголошення

Функція	Опис	Вхід	Вихід
trainFraudModel()	Тренує модель логістичної регресії	масив об'єктів виду $\{x, y: 0 1\}$, тобто даних для навчання об'єкту	Об'єкт $\{ \text{weights: number[]} \}$ із навченими k-ми моделі
trainScoringModel()	Тренує лінійний скорер	масив об'єктів виду $\{x, y: 0 1\}$, тобто даних для навчання об'єкту	Об'єкт $\{ \text{weights: number[]} \}$ із навченими k-ми моделі

5.6.2 Структурна схема системи

Структурна схема передбачає наочне представлення основних компонентів системи і їх взаємодію між собою. Іншими словами це план архітектури з всіма чітко позначеними серверами, базами даних, клієнтською частиною, зовнішніми сервісами та зв'язками між ними. Стрілки показують кожен етап обробки даних від запиту користувача до повернення назад їх клієнтові у вигляді готового списку квартир із маркерами на карті або деталями обраного об'єкта, дозволяючи зрозуміти рух потоку даних всередині системи.

Структурну схему даного проєкту можна знайти серед графічних матеріалів проєкту: кресленик IC13.050БАК.005 Д4.

5.7 Тестування системи

Заключним етапом життєвого циклу розробки системи є тестування. Перевірка чи реально було виконано всі поставлені задачі та функціональні вимоги, що система працює стабільно, передбачені алгоритми дають коректний результат, а сам інтерфейс є нативним та не створює бар'єрів для користувача.

					IC13.050БАК.005 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

Розглянемо роботу готового застосунку.

На рисунку 5.3 представлено головну сторінку застосунку, натискаючи на кнопку «Переглянути всі оголошення», відбувається перехід на відповідну сторінку каталогу оголошень. Користувачеві одразу стає доступною панель фільтрів та пошуку, можна переглянути на рисунку 5.4. Поля можна залишити за замовчуванням пустими, або вибрати необхідні параметри.

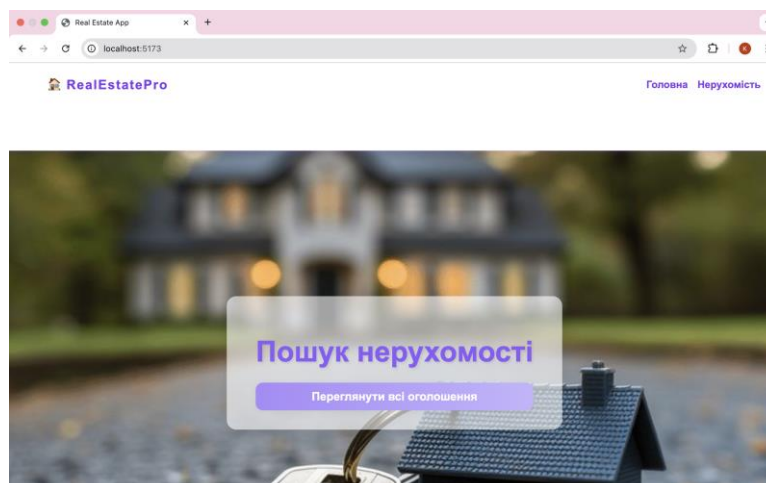


Рисунок 5.3 — Головна сторінка

Натиснувши кнопку «Пошук», система миттєво формує запит до REST-маршруту `/api/properties`, підставивши всі обрані користувачем параметри, фронтенд отримує дані й сортує їх за полем `relevanceScore` у спадаючому порядку, тому першим бачимо «найкращий збіг», останнім – «найслабший», рисунок 5.5.

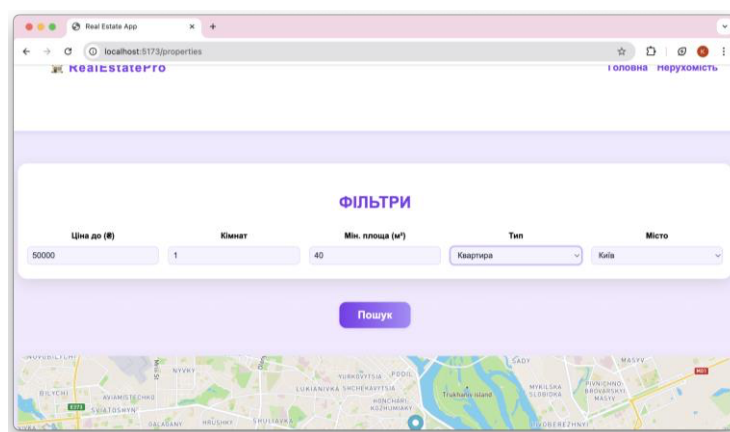


Рисунок 5.4 — Задані фільтри для пошуку результатів

Користувач бачить лише ті оголошення, що повністю відповідають його фільтрам, і вже впорядковані системою від найбільш релевантних до найменш корисних. Це підтверджує, що навчені моделі працюють коректно й дозволяють прийняти рішення значно швидше, ніж у звичайних списках без інтелектуальної сортування.

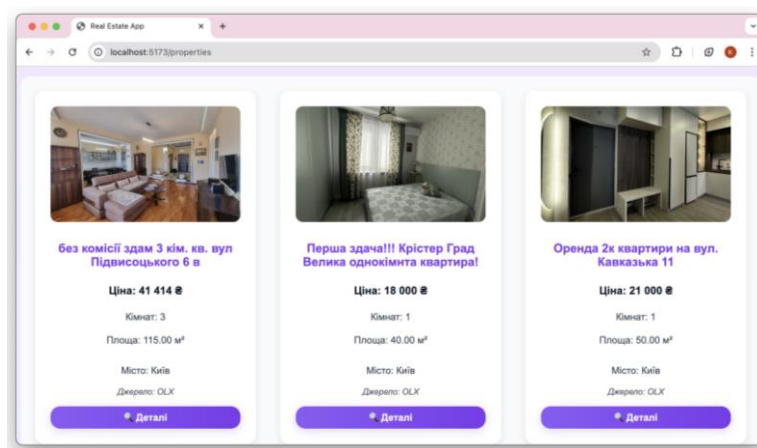


Рисунок 5.5 — Вивід результатів відповідно до запиту пошуку

Пропозиції об'єктів, у яких відсутня важлива інформація, автоматично визначаються системою як підозрілі, рисунок 5.6.

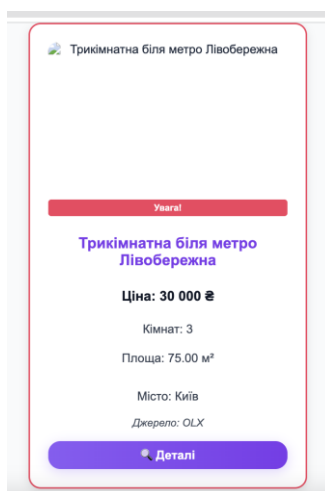


Рисунок 5.6 — Підозріле оголошення, яке підсвічується червоним

Це може бути, наприклад, відсутність фотографій або відсутність ключових параметрів. Такі оголошення також часто мають низький рейтинг відгуків, що

додатково свідчить про їхню ненадійність. Для зручності користувачів і привернення уваги до можливих проблемних оголошень, система підсвічує їх червоним кольором, що дозволяє швидко та ефективно ідентифікувати потенційно ризикові варіанти житла.

Якщо натиснути кнопку «Деталі» на картці оголошення, то буде представлено більше інформації про оголошення, рисунок 5.7. А саме галерея фотографій, позначка оголошення на карті, опис нерухомості.

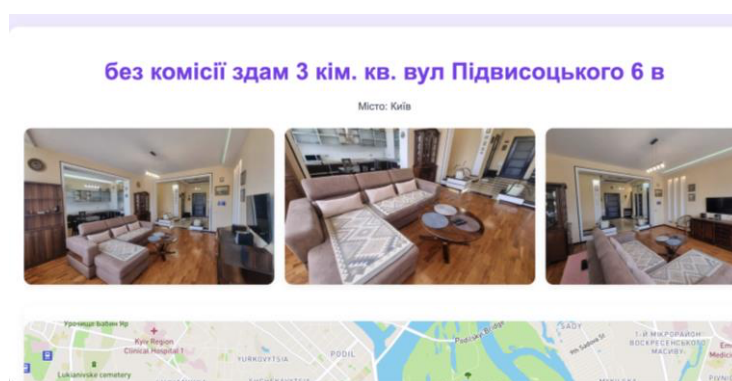


Рисунок 5.7 — Сторінка деталей оголошення

А також, найголовніше, тут можна переглянути рейтинг оголошення та ймовірність шахрайства. Це дає користувачеві змогу не лише побачити загальний рейтинг, сформований на основі різних критеріїв (як-от якість опису, наявність фото, відгуки інших користувачів), а й оцінити ступінь ризику, пов'язаний із конкретним пропозицією. На рисунку 5.8 продемонстровано інтерфейс, де користувач може детально ознайомитися з цими показниками, що значно полегшує прийняття рішення при виборі житла.

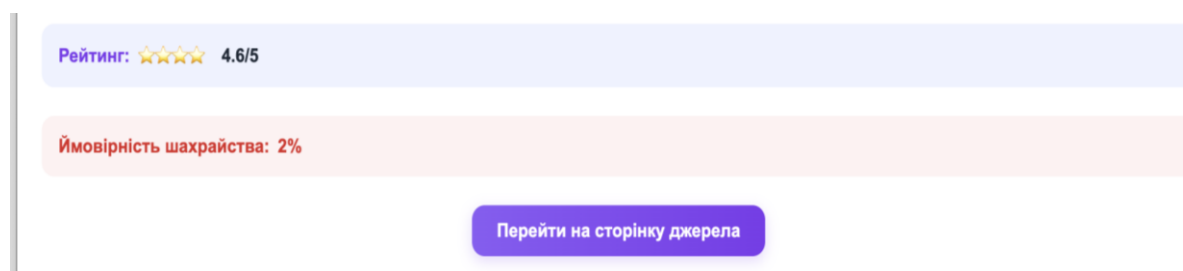


Рисунок 5.8 — Результат роботи алгоритмів

Однією з важливих функцій системи є можливість залишати коментарі до оголошень, а також додавати фотографії для більш повного висвітлення досвіду користувача. На рисунку 5.9 представлено інтерфейс, який забезпечує ці можливості, дозволяючи користувачам ділитися власними враженнями та допомагати іншим приймати рішення при виборі житла. Цей функціонал підвищує загальну цінність системи та сприяє активній взаємодії користувачів.

Рисунок 5.9 — Форма для коментарів

Після відправки їх може переглянути кожен користувач, який зацікавився даним оголошенням, рисунок 5.10.

Рисунок 5.10 — Коментарі відвідувачів

Рейтинг кожного коментаря враховується системою та може бути корисним для інших.

Якщо оголошення зацікавило, то можна перейти за посиланням, натиснувши на кнопку та зв'язатися з орендодавцем для подальшої можливості оренди, першоджерело оголошення, рисунок 5.11.

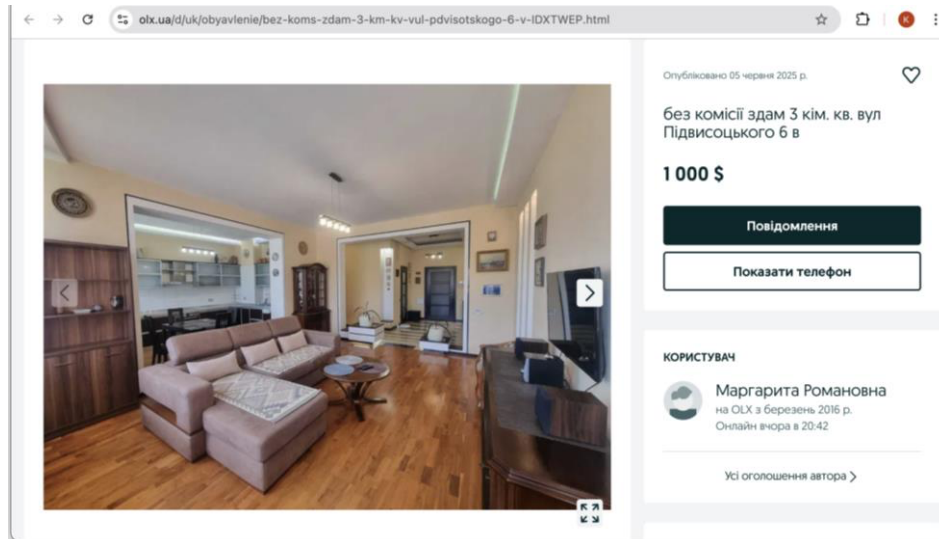


Рисунок 5.11 — Першоджерело оголошення

З результатів тестування можна впевнено зазначити, що всі поставлені вимоги до системи були успішно виконані. Реалізовані функції працюють стабільно, забезпечуючи необхідний рівень зручності, точності та безпеки для користувачів, що підтверджує ефективність розробленої інформаційної системи.

Висновки до розділу 5

У розділі було всебічно досліджено архітектурні засади функціонування розроблюваної інформаційної системи для аналізу та оцінювання оголошень про оренду житла. На основі сформульованих вимог було обґрунтовано вибір архітектури, яка дозволяє чітко розділити логіку подання даних, обробки запитів та їхнього збереження. Такий підхід забезпечує гнучкість, масштабованість і легкість супроводу системи на всіх етапах її життєвого циклу.

Було детально розглянуто складові клієнтської, серверної та базової частин, їх взаємодію, а також описано канали та формати обміну даними. Особливу увагу

приділено моделюванню потоків даних, зокрема механізмам запитів користувача, обробки фільтрів, передачі об'єктів нерухомості з відповідними атрибутами, а також їх збереження та оновлення в базі даних. Наведена структурна модель бази даних забезпечує нормалізоване й узгоджене зберігання інформації з урахуванням можливості розширення за рахунок нових джерел, додаткових параметрів та функціоналу.

У межах розділу було також представлено діаграми, що наочно ілюструють логіку взаємодії між підсистемами та дозволяють чітко простежити сценарії обробки даних – від формування запиту користувачем до відображення результату з відповідними оцінками.

Розроблена архітектура, будучи технічно обґрунтованою, слугує основою для ефективної реалізації функціональних вимог, визначених у попередніх розділах. Її модульність, сумісність із сучасними технологіями веб-розробки, підтримка асинхронної взаємодії та можливість подальшої інтеграції з модулями штучного інтелекту створюють ґрунт для побудови масштабованої, стабільної та орієнтованої на користувача системи підтримки прийняття рішень у сфері оренди житла.

6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

6.1 Змістовна постановка задачі

Користувач шукає житло для оренди у великому незнайомому місті X . Для вибору підходящого варіанту користувач задає низку критеріїв: бюджет: оренда квартири не повинна перевищувати m гривень на місяць; кількість кімнат: квартира повинна мати мінімум a кімнат; площа: площа квартири має бути не менша за k м².

Надати відсортовані результати, які задовільняють усі вимоги користувача та першими виводити вже перевірені варіанти з відгуками.

6.2 Математична постановка задачі

Вхідні дані:

Множина X (формула 6.1) – множина характеристик, які передаються для тренування та тестування моделі:

$$x_i = (x_1, x_2, x_3, x_4, \dots, x_m), \quad (6.1)$$

де x_1 – ціна,

x_2 – площа,

x_3 – кількість кімнат,

x_m – інші характеристики.

Оцінку ймовірності шахрайства будемо вираховувати логістичною регресією (формули 6.2 та 6.3), адже це різновид багатомірної регресії, загальне призначення якої полягає в тому, щоб проаналізувати зв'язок між декількома незалежними змінними та залежною змінною, що особливо важливо при виявленні шахрайських оголошень на основі сукупності ознак [17].

Цю модель можна використовувати для оцінки ймовірності результату події за допомогою стандартних коефіцієнтів регресії, які дозволяють кількісно

					IC13.050БАК.005 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

інтерпретувати ступінь впливу кожної змінної на ймовірність настання досліджуваного результату в межах обраної задачі класифікації [18].

$$\hat{p} = \sigma(z) = \frac{1}{1+e^{-z}}, \quad (6.2)$$

$$z = w_0 + \sum_{i=1}^m w_i x_i, \quad (6.3)$$

де w_0 – зсув (intercept),

w_i – вагові коефіцієнти, що навчаються під час тренування моделі,

$\sigma(z)$ – логістична (сигмоїдна) функція.

Для навчання використовується логістична функція втрат, формула 6.4:

$$L = -[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})] \quad (6.4)$$

Оцінка оголошення обчислюється за формулою 6.5 як зважена лінійна комбінація нормалізованих ознак об'єкта (ціни, площі, кількості кімнат, середнього рейтингу тощо), де ваги w_i визначаються під час навчання моделі й відображають відносну важливість кожного критерію.

$$\hat{r} = w_0 + \sum_{i=1}^m w_i x_i, \quad (6.5)$$

де $\hat{r} \in [0, 5]$ – розрахований рейтинг оголошення,

x_i – нормалізовані ознаки,

w_0 – зсув (intercept), що компенсує середнє значення,

w_i – вагові коефіцієнти, що визначають важливість кожного критерію.

Список результатів сортується у спадаючому порядку за $\hat{r}$, що дозволяє виводити найрелевантніші або найбільш ймовірні об'єкти на перших позиціях.

Оптимізація моделі здійснюється шляхом мінімізації середньої абсолютної помилки, яка вимірює середню величину абсолютних різниць між передбаченими значеннями моделі та фактичними спостереженнями, формула 6.6. Зменшення

MAE означає покращення узгодженості моделі з реальними даними та підвищення точності її прогнозів.

$$L = \frac{1}{n} \sum_{i=1}^n |\hat{r}_i - \hat{r}_i^{true}| \quad (6.6)$$

Кожна модель має власну функцію втрат і оптимізаційні критерії, хоч і мають один набір даних для навчання. Задачі відрізняються як за математичними властивостями, так і за метою використання.

6.3 Обґрунтування методу розв'язання

Цю задачу можна розв'язати також і традиційними методами такі як метод TOPSIS, метод взважених сум WSM. Вони можуть бути ефективними та показувати чітко оптимальний розв'язок для розв'язання задач з меншими обсягами даних або при чітко визначених критеріях важливості. Однак, вони мають обмеження в контексті обробки складних, нелінійних зв'язків між характеристиками даних (наприклад, взаємозв'язок між ціною, місцезнаходженням, відгуками користувачів і так далі).

TOPSIS дозволяє ранжувати альтернативи за наближеністю до ідеального рішення, яке визначається на основі вхідних даних користувача. Однак цей метод потребує чітко визначених вагових коефіцієнтів і не враховує адаптивність до нових патернів у даних.

Метод зважених сум WSM є простим і широко застосовуваним у системах підтримки прийняття рішень. У роботі він схожий принцип для розрахунку підсумкового рейтингу оголошення, оскільки дозволяє оцінити якість об'єкта як зважену суму нормалізованих показників. Проте цей метод не враховує можливі нелінійні взаємозв'язки між критеріями та не адаптується до зміни поведінки користувачів.

У роботі модель навчається на основі нормалізованих даних і автоматично підбирає вагові коефіцієнти, що відповідають внеску кожної ознаки.

У межах даної роботи для аналізу оголошень використовуються базові моделі машинного навчання, які забезпечують достатню точність при невисокій складності реалізації. Зокрема, для виявлення потенційно шахрайських оголошень застосовується логістична регресія, що дозволяє розрахувати ймовірність шахрайства на основі параметрів, таких як ціна, кількість фото, опис та джерело.

Такий підхід дозволяє автоматично визначати важливість ознак під час навчання моделі та забезпечує баланс між точністю, адаптивністю та обчислювальною простотою – що є доцільним для практичного застосування в інформаційній системі пошуку житла.

6.4 Опис методу розв'язання

Розв'язання задачі здійснюється через послідовну обробку вебданих і побудову двох незалежних моделей машинного навчання – ймовірність шахрайства $\hat{p}_{\text{fraud}}$ і рейтинг релевантності $\hat{r}$. Загальний процес включає наступні етапи:

КРОК 1. Дані про оголошення збираються та парсяться з платформ OLX, DOM.RIA, LUN. Далі інформація трансформується в уніфіковану структуру Property із полями: ціна, площа, кімнати, опис, фото, джерело тощо.

КРОК 2. Попередня обробка. Числові ознаки підлягають нормалізації до інтервалу $[0;1]$ за допомогою методу мінімакс-масштабування. Категоріальні змінні (наприклад, джерело оголошення) перетворюються на числові значення згідно з коефіцієнтом довіри.

КРОК 3. Після обробки всі записи розподіляються у співвідношенні 80:20 на тренувальну та тестову вибірки. Для задачі класифікації додатково використовується стратифікація за міткою шахрайства, щоб зберегти пропорції класів у кожній підвибірці.

КРОК 4. Навчання моделей. Модель логістичної регресії реалізована у формі нейронної мережі з сигмоїдною активацією на виході та функцією втрат типу `binary_crossentropy`. Вона призначена для класифікації оголошень і обчислює

ймовірність шахрайства $\hat{p}_{\text{fraud}} \in [0, 1]$. Модель лінійної регресії реалізується як нейронна мережа з лінійною активацією на вихідному шарі, що прогнозує інтегральний рейтинг об'єкта $\hat{r} \in [0, 5]$. Навчання здійснюється шляхом мінімізації середньої абсолютної похибки (MAE).

КРОК 5. У разі відсутності рейтингу система підставляє середнє значення по всіх оголошеннях. Такий підхід дозволяє уникнути упередженого заниження рейтингу нових оголошень і вирішити проблему холодного старту.

КРОК 6. Для оцінки якості використовуються метрики. Для класифікації шахрайства використовується метрика ROC-AUC, яка має перевищувати 0.80. Результати навчання можна переглянути на рисунку 6.1.

```
Best threshold based on F1: 0.60
Fraud detection report:
```

	precision	recall	f1-score	support
0	0.6852	0.5873	0.6325	63
1	0.8219	0.8759	0.8481	137
accuracy			0.785	200
macro avg	0.7536	0.7316	0.7403	200
weighted avg	0.7788	0.7850	0.7800	200

```
ROC AUC score: 0.8189

Epoch 1/30
21/21 ————— 1s 6ms/step — loss: 2.9774 — mae: 2.87
```

Рисунок 6.1 — Результати навчання моделі виявлення шахрайства

Для регресії рейтингу – $\text{MAE} \leq 0.35$ на тестовій вибірці. Результат отриманої системи можна переглянути на рисунку 6.2.

```
Epoch 20/20
17/17 ————— 0s 3ms/step — loss: 0.3462 — mae: 0.3462 — val_loss: 0.2101 — val_mae: 0.2101
3/3 ————— 0s 15ms/step
Rating prediction MAE: 0.201
```

Рисунок 6.1 — Результати навчання моделі ранжування

КРОК 7. Навчені моделі зберігаються у форматі .h5 та підключаються до серверної частини вебзастосунку. При кожному запиті до маршруту /api/properties

виконується обчислення двох показників для кожного оголошення: ймовірність шахрайства – у відсотковій формі ($\hat{p}_{fraud} \in [0\%; 100\%]$); інтегральний рейтинг – у шкалі від 0 до 5 ($\hat{r} \in [0; 5]$).

Отримані значення використовуються для динамічного сортування оголошень: спочатку виводяться релевантні, надійні варіанти з високим рейтингом. Записи, для яких ймовірність шахрайства перевищує порогове значення, $\hat{p}_{fraud} > 70\%$, автоматично позначаються спеціальним індикатором у клієнтському інтерфейсі.

6.5 Приклад застосування математичної моделі

Нехай користувач задає такі критерії пошуку житла:

- місто: Київ;
- бюджет: не більше 15 000 грн;
- мінімальна площа: 40 м²;
- мінімум 2 кімнати.

До прикладу в базі даних є оголошення, з такими даними, як в таблиці 6.1. А також найбільша ціна 20000 грн, найменша 8000 грн; мінімальна площа 25 м², максимальна – 90 м², рейтинг знаходиться в межах від 0 до 5.

Таблиця 6.1 — Приклад даних оголошення

ID	Ціна	Площа	Кімнати	Відстань до центру	Рейтинг	Фото	Джерело
101	13500	45	2	3.2	4.5	5	OLX

Першим кроком виконуємо міні-максну нормалізацію ознак до $[0,1]$ за формулою 6.6. Це необхідно для того, щоб різні параметри, які можуть мати суттєво різні масштаби (наприклад, ціна, площа, кількість кімнат), були приведені до порівнянних значень, що покращує стабільність та швидкість навчання моделі. Це дозволяє моделі рівномірно враховувати вплив кожної ознаки під час навчання, не зміщуючи пріоритет у бік чисельно більших параметрів. Значення кожної ознаки обчислюється як відношення різниці між фактичним значенням та мінімальним

значенням ознаки до різниці між максимальним та мінімальним значенням цієї ознаки, тобто:

$$x_i^{(norm)} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \quad (6.6)$$

де x_i – значення ознаки, x_{min} , x_{max} – максимальне та мінімальне значення у тренувальній вибірці.

Нормалізуємо всі дані:

$$x_1 = \frac{13500 - 8000}{20000 - 8000} = 0,458, \text{ ціну};$$

$$x_2 = \frac{45 - 25}{90 - 45} = 0,308, \text{ площу};$$

$$x_3 = \frac{2 - 1}{5 - 1} = 0,25, \text{ кількість кімнат};$$

$$x_4 = \frac{3,2 - 0}{10 - 0} = 0,32, \text{ відстань до центру};$$

$$x_5 = \frac{4,5}{5} = 0,9, \text{ рейтинг};$$

$$x_6 = \frac{5 - 1}{10 - 1} = 0,444, \text{ кількість фото}.$$

Сьома ознака це джерело оголошення, тому для платформ було створено шкалу довіри, на основі проаналізованих даних з 2 розділу, а також даних з платформ про довіру.

Таблицю можна переглянути нижче, таблиця 6.2. Відповідно $x_7 = 0,96$.

Таблиця 6.2 — Шкала довіри

Джерело	Довіра (score)
Dom.Ria	0,98
LUN	0,97
OLX	0,96

Отримуємо проміжний результат: $x_i = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$.

Припустимо, що ваги моделі було навчено й вони мають такі значення:

$$w = (0.4, 0.2, 0.1, -0.1, 0.35, 0.05, 0.2), \quad w_0 = 0.15$$

Цей вектор передається у лінійну модель, яка обчислює підсумковий рейтинг оголошення за формулою 6.4:

$$\begin{aligned} \hat{r} &= w_0 + \sum_{i=1}^m w_i x_i = 0.15 + 0.4 \times 0.458 + 0.2 \times 0.308 + 0.1 \times 0.25 \\ &\quad - 0.1 \times 0.32 + 0.35 \times 0.9 + 0.05 \times 0.444 + 0.2 \times 0.96 = 0.916 \end{aligned}$$

Якщо розглядати за шкалою $[0, 5]$, то результат буде 4.58.

Для моделі логістичної регресії вагові коефіцієнти є іншими:

$$w = (0.5, -0.1, 0.1, 0.3, -0.4, -0.2, -0.5), w_0 = 0.1$$

Підставимо нормалізовані значення ознак:

$$\begin{aligned} z &= 0.1 + 0.5 \times 0.458 + (-0.1) \times 0.308 + 0.1 \times 0.25 \\ &\quad + 0.3 \times 0.32 + (-0.4) \times 0.9 + (-0.2) \times 0.444 + (-0.5) \times 0.96 = -0.5096 \end{aligned}$$

$$\hat{p}_{\text{fraud}} = \frac{1}{1 + e^{-0.5096}} = 0.375$$

Отже, модель вважає, що ймовірність шахрайства даного оголошення становить приблизно 37.5%, що не є критичним, однак може бути достатнім для попередження користувача про можливу підозру. Це дозволяє користувачу бути більш обережним та здійснювати додаткову перевірку перед прийняттям рішення, що значно підвищує безпеку та довіру до платформи. Враховуючи, що ймовірність не надто висока, система не повинна формувати зайвого занепокоєння, а лише допомагати у прийнятті усвідомленого вибору.

Висновки до розділу 6

У даному розділі було детально обґрунтовано вибір математичних моделей, які використовуються для аналізу та класифікації інформації про оголошення з оренди житла. Основну увагу зосереджено на побудові моделей машинного навчання.

Процес побудови моделей включав нормалізацію даних, виділення релевантних ознак, балансування вибірки для уникнення переважання одного з класів, а також підбір оптимальних архітектур. Для підвищення точності класифікації шахрайських оголошень було застосовано вагове коригування класів та механізми ранньої зупинки, що дозволяють уникнути перенавчання моделі. Метричні показники, отримані в результаті експериментального навчання, свідчать про задовільну здатність моделі узагальнювати нові дані та виявляти потенційно шахрайські записи з прийнятною точністю, повнотою та значенням F1-міри.

Особливу увагу також приділено задачі регресійного прогнозування рейтингу оголошення. Побудована модель демонструє адекватну точність у прогнозуванні користувацької оцінки на основі низки кількісних та якісних характеристик. Отримане середнє абсолютне відхилення підтверджує практичну застосовність моделі у реальних умовах, зокрема – для автоматизованого ранжування варіантів нерухомості відповідно до очікувань користувачів.

Отже, розроблене математичне забезпечення є важливою складовою загальної архітектури інформаційної системи. Його застосування дозволяє підвищити об'єктивність, прозорість та ефективність процесу підбору житла, що особливо актуально в умовах великої кількості оголошень, можливих зловживань.

ВИСНОВКИ

У результаті виконання дипломного проєкту було розроблено інформаційну систему для пошуку, аналізу та оцінювання оголошень про оренду житла з різних вебресурсів. Основна мета проєкту – спростити для користувача процес вибору відповідного житла шляхом централізованого збору даних, проведення їх багатокритеріального аналізу, виявлення потенційно шахрайських оголошень та надання персоналізованих рекомендацій.

У першому розділі було детально проаналізовано предметну область. Визначено основні проблеми, з якими стикаються користувачі при пошуку житла: недостовірність оголошень, складність порівняння об'єктів. Сформульовано завдання, які має вирішити система, та описано основні функції, серед яких – агрегація даних, аналіз і фільтрація за критеріями, рейтингова оцінка та система виявлення підозрілих об'єктів.

Поставлені у розділі 1.2 завдання були виконані повністю:

- проаналізовано предметну область та існуючі аналогічні платформи;
- сформульовано вимоги до функціональності системи;
- розроблено структуру системи на основі клієнт-серверної архітектури;
- реалізовано базу даних і механізми обробки та фільтрації інформації;
- розроблено вебдодаток з використанням сучасних технологій;
- досліджено підходи до виявлення ознак недобросовісності в оголошеннях.

Розроблений застосунок дозволяє здійснювати автоматизований пошук оголошень на різних онлайн-платформах, приводити їх до єдиного формату, зберігати у базі даних, здійснювати оцінку об'єктів за низкою параметрів – ціна, місцезнаходження, площа, кількість кімнат, наявність фото, відгуки користувачів тощо – та формувати рейтинг об'єктів відповідно до уподобань користувача. Для аналітики використано алгоритми простого машинного навчання для виявлення підозрілих оголошень.

З практичної точки зору, розроблена система демонструє високу ефективність у вирішенні задачі автоматизації аналізу ринку оренди житла. Її застосування можливе у таких сферах:

- сервіси пошуку житла для студентів, сімей, іноземців;
- аналітика ринку нерухомості;
- доповнення до платформ оголошень;
- використання в агентствах нерухомості.

Система дозволяє адаптувати функціональність під потреби різних категорій користувачів. Важливою перевагою є можливість додавання нових джерел даних без значної модифікації архітектури системи.

З науково-технічної точки зору, розробка демонструє інтеграцію методів інтелектуального аналізу даних та вебагрегації в одному інструменті, що підвищує якість прийняття рішень у сфері оренди житла.

Подальші дослідження та розвиток проєкту доцільно здійснювати у таких напрямках:

- створення мобільного додатку;
- можливість додавати самостійно власні оголошення;
- адаптація системи до інших регіонів або країн;
- розширення критеріїв аналізу за рахунок обліку динаміки цін, сезонності, транспортної доступності тощо.

Усі поставлені в дипломному проєкті завдання були виконані в повному обсязі, а досягнуті результати відповідають сучасному рівню розвитку інформаційних систем у галузі пошуку та аналізу ринку оренди житла. Система рекомендована до використання як помічник при прийнятті рішень щодо вибору житла, з акцентом на безпечність, релевантність та зручність.

					IC13.050БАК.005 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довгострокова оренда квартир: як змінився попит та критерії вибору у 2024 році: вебсайт. URL: <https://news.finance.ua/ua/dovhostroкова-orenda-kvartyr-yak-zminyvsvya-popyt-ta-kryterii-vyboru-u-2024-roci> (дата звернення 20.04.2025).
2. Оренда квартири: огляд чатів сервісів: вебсайт. URL: <https://kosht.media/orenda-kvartyry-ohliad-chativ-ta-servisiv/> (дата звернення 25.04.2025).
3. JavaScript – офіційна документація: вебсайт. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення 1.05.2025).
4. React Documentation: вебсайт. URL: <https://react.dev/learn> (дата звернення 25.04.2025).
5. Що таке React JS: вебсайт. URL: <https://cases.media/article/sho-take-react-js-yak-pochati-vivchati-reakt-navichki-dlya-react-developer> (дата звернення 1.05.2025).
6. Node.js Documentation: веб-сайт. URL: <https://nodejs.org/docs/latest/api/> (дата звернення 1.05.2025).
7. Все, що потрібно знати про Node.js: вебсайт. URL: <https://wezom.com.ua/ua/blog/vse-chto-nuzhno-znat-o-nodejs> (дата звернення 1.05.2025).
8. Express.js – офіційна документація: веб-сайт. URL: <https://expressjs.com> (дата звернення 1.05.2025).
9. Основи Express.js та його можливості: вебсайт. URL: <https://javascript.org.ua/osnovi-express-js-ta-jogo-mozhливosti/> (дата звернення 1.05.2025).
10. Як встановити та налаштувати PostgreSQL: вебсайт. URL: <https://thehost.ua/ua/wiki/administration/database/postgresql-install> (дата звернення 2.05.2025).
11. Tailwind CSS – офіційна документація: вебсайт. URL: <https://tailwindcss.com/> (дата звернення 2.05.2025).
12. Що таке TensorFlow.js і для чого він потрібен: вебсайт. URL: <https://uk.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/advancing-in-machine-learning/introduction-to-tensorflow-js/examination-review-introduction-to-tensorflow-js/> (дата звернення 3.05.2025).

					IC13.050БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

13. Створення інтерактивних мап за допомогою Mapbox GL JS та React.js: вебсайт. URL: <http://ds.knu.edu.ua/jspui/bitstream/123456789/4411/1/Створення%20інтерактивних%20мап%20за%20допомогою%20Mapbox%20GL%20JS%20та%20React.JS.pdf> (дата звернення 3.05.2025).

14. Розуміння Клієнт-Серверної Архітектури на прикладах: вебсайт. URL: <https://dou.ua/forums/topic/44636/> (дата звернення 5.05.2025).

15. Клієнт-серверна архітектура та ролі серверів: вебсайт. URL: <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229> (дата звернення 5.05.2025).

16. Інфологічне моделювання та основи баз даних: вебсайт. URL: https://it.nmu.org.ua/ua/scientific_method_materials/lecture_notes/Конспект_лекцій_Б_Д_частина1_2021.pdf (дата звернення 10.05.2025).

17. Logistic Regression. KPI Repository: вебсайт. URL: <https://ela.kpi.ua/bitstreams/0b004f74-2cd2-4c61-8139-a604e9fa7029> (дата звернення 17.05.2025).

18. Hosmer D. W., Lemeshow S., Sturdivant R. X. Applied Logistic Regression. 3rd ed. Hoboken: Wiley, 2013: електронний ресурс. URL: <https://dl.icdst.org/pdfs/files4/7751d268eb7358d3ca5bd88968d9227a.pdf> (дата звернення 17.05.2025).