

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка інформаційної системи обліку даних про оренду житла

Виконав: студент VI курсу, групи СНм-61

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Козак В. І.

(прізвище та ініціали)

Керівник

(підпис)

Готович В. А.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Никитюк В. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Карпінський М.П.

(прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет \_\_\_\_\_ комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)  
Кафедра \_\_\_\_\_ комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
\_\_\_\_\_ Боднарчук І.О.  
(підпис) (прізвище та ініціали)  
«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня \_\_\_\_\_ Магістр  
(назва освітнього ступеня)  
за спеціальністю \_\_\_\_\_ 122 Комп'ютерні науки  
(шифр і назва спеціальності)  
Студенту \_\_\_\_\_ Козак Василь Іванович  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи обліку даних про оренду житла  
Керівник роботи \_\_\_\_\_ Готович Володимир Анатолійович, к.т.н., доцент кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » листопада 2023 року № 4/7-1099

2. Термін подання студентом завершеної роботи \_\_\_\_\_ 20 грудня 2023р.

3. Вихідні дані до роботи \_\_\_\_\_ Наукові публікації та літературні джерела щодо розробки інформаційної системи обліку даних про оренду житла

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз відомих інформаційних систем обліку даних про оренду житла. 1.1. Загальна характеристика 1.2. Історія розвитку 1.3 Причини використання інформаційних систем для задач обліку даних про оренду житла 1.4 Сучасний стан використовуваних технологій

1.5 Використання хмарних технологій 1.6 Використання технологій штучного інтелекту

1.7 Висновок до першого розділу 2. Аналіз методів проектування інформаційних систем

обліку даних про оренду житла 2.1. Загальні підходи та методи досліджень. 2.2. Використання наявних методологій розробки 2.3. Аналіз відомих систем. 2.4. Формування вимог до розробки

інформаційної системи обліку даних про оренду житла 2.5. Висновок до другого розділу.

3. Розробка інформаційної системи обліку даних про оренду житла. 3.1 Актанти та варіанти

використання інформаційної системи. 3.2. Розробка архітектури інформаційної системи. 3.3

Обґрунтування вибору засобів розробки 3.4. Створення інформаційної панелі (Dashboard)

3.5. Методи узагальнення даних 3.6. Функціонування та інтерфейс користувача інформаційної

системи 3.7. Висновок до третього розділу. 4. Охорона праці та безпека в надзвичайних

ситуаціях. 4.1 Розробка раціональної діяльності та створення сприятливих і безпечних умов

персоналу. 4.2 Вплив електромагнітного випромінювання ПК на здоров'я користувача.

4.3. Організація цивільного захисту на житлових об'єктах. 4.4. Висновок до четвертого.

розділу. Висновки. Додатки.

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

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 24 листопада 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.11.2023	<i>Виконано</i>
2.	Підбір наукових джерел про розробку інформаційної системи обліку даних про оренду житла	26.11.2023-28.11.2023	<i>Виконано</i>
3.	Опрацювання наукових публікацій та збір даних по темі роботи	29.11.2023-1.12.2023	<i>Виконано</i>
4.	Виконання дослідження згідно мети кваліфікаційної роботи	2.12.2023-4.12.2023	<i>Виконано</i>
5.	Оформлення розділу «Аналіз предметної області розробки інформаційної системи обліку даних про оренду житла»	5.12.2023-7.12.2023	<i>Виконано</i>
6.	Оформлення розділу «Аналіз методів проектування інформаційних систем обліку даних про оренду житла»	8.12.2023-10.12.2023	<i>Виконано</i>
7.	Оформлення розділу «Розробка інформаційної системи обліку даних про оренду житла»	11.12.2023-13.12.2023	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Охорона праці»	14.12.2023-15.12.2023	<i>Виконано</i>
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.12.2023-17.12.2023	<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи	18.12.2023-19.12.2023	<i>Виконано</i>
11.	Нормоконтроль	19.12.2023-20.12.2023	<i>Виконано</i>
12.	Перевірка на плагіат	21.12.2023	<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи	22.12.2023	<i>Виконано</i>
14.	Захист кваліфікаційної роботи	26.12.2023	

Студент

(підпис)

Козак В. І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Готович В. А.

(прізвище та ініціали)

## АНОТАЦІЯ

Розробка інформаційної системи обліку даних про оренду житла // Кваліфікаційна робота освітнього рівня «Магістр» // Козак Василь Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2023 // С. 79, рис. – 9, табл. – 2, кресл. – 10, додат. – 3, бібліогр. – 54.

**Ключові слова:** інформаційна панель, облік даних, оренда житла, управління нерухомістю, оптимізація процесів, dashboard, React.

Кваліфікаційна робота присвячена розробці інформаційної системи обліку даних про оренду житла.

В першому розділі кваліфікаційної роботи описано аналіз відомих інформаційної системи обліку даних про оренди житла. Подано загальну характеристику інформаційних систем обліку даних про оренду житла. Розглянуто історичну хронологію розвитку інформаційних системи обліку даних про оренду житла. Висвітлено основні причини використання інформаційних системи обліку даних про оренду житла. Проаналізовано сучасний стан технологій інформаційних систем обліку даних про оренду житла. Досліджено використання хмарних технологій для системи обліку даних про оренду житла. Обґрунтовано використання технологій штучного інтелекту для системи обліку даних про оренду житла. Сформовано висновок до першого розділу.

В другому розділі кваліфікаційної роботи описано загальні підходи та методи розробки інформаційної системи. Досліджено наявні методології проектування інформаційних систем обліку даних оренди житла. Подано наявні методології розробки програмного забезпечення для обліку даних оренди житла та відомі системи. Сформовано основні вимоги до розробки інформаційної системи обліку даних про оренду житла.

В третьому розділі кваліфікаційної роботи розроблено архітектуру інформаційної системи обліку даних житла. Запропоновано основні інструменти

для розробки системи обліку даних житла. Спроектовано інформаційну панель обліку даних про оренду житла.

Загалом, робота включає в себе комплексний аналіз, проектування та розробку системи обліку даних оренди житла, і вносить вагомий внесок у розвиток цього напрямку. У роботі висвітлено не тільки історію та сучасний стан інформаційних систем обліку даних оренди житла, але й досліджено актуальні методології та підходи до їх розробки. Запропонована архітектура системи та розроблені інструменти для її створення враховують сучасні тенденції.

## ANNOTATION

Development of an information system of accounting on housing rent // The educational level "Master" qualification work // Vasyl Kozak // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNm-61 group // Ternopil, 2023 // P. 79, fig. - 9, tables - 2, posters - 10, annexes - 3, ref. - 54.

**Key words:** information panel, data accounting, housing rental, real estate management, process optimization, dashboard, React.

The qualification work is devoted to the development of an information system for recording data on housing rent.

The first section of the qualification work describes the analysis of the well-known information system for recording data on housing rentals. The general characteristics of information systems for recording data on housing rent are given. The historical chronology of the development of the information system for recording data on housing rent is considered. The main reasons for using the information system for recording data on housing rent are highlighted. The current state of technologies of information systems for recording data on housing rent is analyzed. The use of cloud technologies for the housing rental data accounting system was studied. The use of artificial intelligence technologies for the housing rental data accounting system is substantiated. A conclusion to the first section has been formed.

The second section of the qualification work describes the general approaches and methods of information system development. The existing methodologies for designing information systems for housing rental data accounting were studied. Available methodologies for software development for housing rental data accounting and known systems are presented. The main requirements for the development of an information system for recording data on housing rent have been formed.

In the third section of the qualification work, the architecture of the housing data accounting information system was developed. The main tools for the development of

a housing data accounting system are offered. An information panel for recording housing rental data has been designed.

In general, the work includes a comprehensive analysis, design and development of a housing rental data accounting system, and makes a significant contribution to the development of this direction. The work covers not only the history and current state of information systems for accounting for housing rental data, but also researches current methodologies and approaches to their development. The proposed system architecture and developed tools for its creation take into account modern trends.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

БД (База даних) – система для зберігання та організації даних.

ІСОДОЖ (Інформаційна система обліку даних про оренду житла) – скорочення, яке використовується для позначення розробленої інформаційної панелі обліку даних про оренду житла.

ОДОЖ (Оренда житла) – скорочення, що вказує на процес надання житла в користування на певний термін за визначеною платою.

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер.

API (Інтерфейс програмування застосунків) – технологія, яка дозволяє взаємодіяти між різними програмами та системами.

API ключ – унікальний ідентифікатор, який дозволяє програмі чи сервісу використовувати функції або дані іншого програмного забезпечення чи сервісу.

SQL (Structured Query Language) – мова структурованих запитань, використовувана для управління та маніпуляції даними у базах даних.

UI/UX (Інтерфейс користувача/Взаємодія з користувачем) – терміни, які означають дизайн і функціональність користувацького інтерфейсу та користувацького досвіду взагалі.



## ЗМІСТ

ВСТУП .....	11
1 АНАЛІЗ ВІДОМИХ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА .....	13
1.1 Загальна характеристика .....	13
1.2 Історія розвитку .....	13
1.3 Причини використання інформаційних систем для задач обліку даних про оренду житла .....	16
1.3.1 Загальна характеристика комплексу причин .....	16
1.3.2 Підвищення ефективності .....	17
1.3.3 Використання аналітичних інструментів .....	18
1.3.4 Покращення комунікацій з клієнтами .....	19
1.4 Сучасний стан використовуваних технологій .....	20
1.5 Використання хмарних технологій .....	21
1.6 Використання технологій штучного інтелекту .....	22
1.7 Висновок до першого розділу .....	24
2 АНАЛІЗ МЕТОДІВ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА .....	25
2.1 Загальні підходи та методи досліджень .....	25
2.1.1 Початковий етап аналізу методів проектування інформаційних систем .....	25
2.1.2 Другий етап аналізу методів проектування інформаційних систем	26
2.1.3 Третій етап аналізу методів проектування інформаційних систем.	27
2.1.4 Четвертий етап аналізу методів проектування інформаційних систем .....	28
2.2 Використання сучасних методологій розробки .....	29
2.2.1 Загальна характеристика методологій .....	29
2.2.2 Використання методології Agile для інформаційних систем обліку даних про оренди житла .....	30
2.2.3 Методологія Waterfall .....	31

2.2.4	Методологія Scrum .....	31
2.2.5	Методологія DevOps .....	32
2.3	Аналіз відомих систем обліку даних про оренду житла.....	33
2.3.1	Загальна характеристика відомих систем .....	33
2.3.2	Інтерфейс користувача.....	35
2.3.3	Функціональність .....	35
2.3.4	Гнучкість системи .....	37
2.3.5	Безпека, надійність та швидкість.....	38
2.3.6	Вартість реалізації .....	40
2.4	Формування вимог до розробки інформаційної системи обліку даних про оренду житла .....	41
2.5	Висновок до другого розділу .....	43
3	РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА .....	44
3.1	Актанти та варіанти використання інформаційної системи обліку даних про оренду житла .....	44
3.2	Розробка архітектури інформаційної системи .....	46
3.2.1	Використання фреймворка React Refine .....	46
3.2.2	Шаблону архітектура .....	47
3.2.3	Інтеграція з сервісом авторизації Google .....	47
3.2.4	Використання системи управління базами даних MongoDB .....	48
3.3	Обґрунтування вибору засобів розробки.....	50
3.4	Створення інформаційної панелі (Dashboard).....	51
3.4.1	Моніторинг процесів оренди житла .....	51
3.4.2	Функціональна сторінка агента з нерухомості.....	52
3.4.3	Функціональна сторінка з доступним житлом.....	53
3.4.4	Функція додавання нового житла у систему .....	55
3.5	Методи узагальнення даних.....	56
3.6	Функціонування та інтерфейс користувача інформаційної системи.....	57
3.7	Висновок до третього розділу.....	59
4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	60

4.1 Розробка раціональної діяльності та створення сприятливих і безпечних умов персоналу .....	60
4.2 Вплив електромагнітного випромінювання комп'ютера на здоров'я користувача.....	63
4.3 Організація цивільного захисту на житлових об'єктах під час воєнного стану .....	65
4.3.1 Створення відповідної інфраструктури для безпеки населення під час воєнного стану.....	65
4.3.2 Евакуаційні плани та зони під час воєнного стану .....	67
4.3.3 Підготовка населення до дій під час воєнного стану .....	68
4.3.4 Контроль та захист комунікацій для житлових об'єктів під час воєнного стану .....	69
4.3.5 Роль влади та міжнародних організацій для забезпечення цивільної безпеки житлових об'єктах .....	69
4.4 Висновок до четвертого розділу.....	70
ВИСНОВКИ.....	72
ПЕРЕЛІК ДЖЕРЕЛ .....	74
ДОДАТКИ	

## ВСТУП

**Актуальність теми.** Протягом останнього часу спостерігається стрімке зростання обсягів оренди житла як в мегаполісах, так і в провінційних містах. Цей тренд свідчить про важливий етап у розвитку сучасного суспільства, де гнучкість вибору місця проживання стає ключовою складовою життєвого стилю. Однак, разом з розширенням ринку оренди виникають нові виклики, пов'язані з ефективністю обліку даних про оренду житла та управління цими процесами. У зв'язку з цим, розробка інформаційної панелі для обліку даних про оренду житла стає актуальною задачею, що відповідає вимогам сучасного ринку та відображає потреби суспільства в удосконаленні житлово-експлуатаційних процесів.

**Мета і задачі дослідження.** Метою кваліфікаційної роботи є розробка інформаційної системи, яка оптимізує бізнес-процеси обліку даних про оренду житла. Для досягнення цієї мети необхідно вирішити наступні задачі:

- 1) Аналіз існуючих систем обліку оренди житла.
- 2) Розробка структури та алгоритму функціонування інформаційної системи.
- 3) Реалізація системи обліку даних про оренди житла.
- 4) Тестування та оптимізація розробленої інформаційної системи.

**Об'єкт дослідження:** процеси обліку даних, пов'язаних із сферою оренди житла.

**Предмет дослідження:** розробка інформаційної системи для оптимізації бізнес-процесів обліку даних про оренду житла.

**Наукова новизна одержаних результатів** кваліфікаційної роботи полягає у впровадженні передових технологій, створенні ефективної системи управління даними та підвищенні якості обслуговування клієнтів в сфері оренди нерухомості.

**Практичне значення одержаних результатів.** Розроблена інформаційна панель матиме практичне застосування в сфері управління орендою житла, сприяючи автоматизації та оптимізації рутинних процесів. Це, в свою чергу,

позитивно вплине на якість обслуговування орендарів та надасть нові можливості для розвитку сучасних технологій у житлово-експлуатаційній галузі.

**Апробація результатів магістерської роботи.** Основні результати проведених досліджень обговорювались на XII Міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 6-7 грудня 2023 року) і на XI Науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 13-14 грудня 2023 року).

**Публікації.** Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (додатки А, Б).

**Структура й обсяг кваліфікаційної роботи.** Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 54 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 79 сторінок, з них 63 сторінки основного тексту, який містить 9 рисунків та 2 таблиць.

# 1 АНАЛІЗ ВІДОМИХ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА

## 1.1 Загальна характеристика

На сьогоднішній день інформаційні системи обліку даних є однією з найважливіших складових ефективної роботи ріелторського бізнесу. Ці системи включають в себе програмне забезпечення та технічні рішення, спрямовані на збір, зберігання, обробку та аналіз інформації, пов'язаної з процесами оренди нерухомості. Такі системи допомагають ефективно вести облік і контроль над об'єктами оренди, сприяючи автоматизації рутинних завдань і покращенню управлінських процесів [1].

Такі системи можуть містити функції як Реєстрація інформації про об'єкти оренди, клієнтська база, фінансовий облік, автоматизовані оповіщення, забезпечення безпеки даних, звітність та аналітика, інтеграція з іншими системами, підтримка документообігу.

## 1.2 Історія розвитку

Історія розвитку інформаційних систем обліку даних про оренду житла є захоплюючим шляхом, який віддзеркалює технологічні досягнення та зміни у вимогах до управління нерухомістю. З початкових етапів, коли комп'ютери лише розпочинали свій шлях у сфері автоматизації, до сучасних днів, коли штучний інтелект та хмарні технології творять нові можливості для оптимізації управління орендними процесами [2].

З кожним етапом розвитку відзначалися нові досягнення, які дозволяли підвищувати ефективність та зручність інструментів обліку, спрощувати ведення документації та покращувати взаємодію між учасниками орендних відносин.

У 1960-х роках з'явилися перші комп'ютерні програми, які могли використовуватися для управління орендою житла. Ці програми були простими

і виконували лише базові завдання, такі як відстеження платежів за оренду та створення договорів оренди.

Однак, навіть ці прості програми були значним кроком вперед у порівнянні з ручним управлінням орендою. Вони дозволяли власникам нерухомості автоматизувати деякі завдання, що звільняло їхній час для інших аспектів свого бізнесу.

У 1970-х роках комп'ютерні програми для управління орендою житла стали більш складними. Вони могли виконувати такі завдання, як управління оголошеннями про оренду, перевірка кредитоспроможності потенційних орендарів та ведення журналів ремонту [2].

Ці додаткові функції зробили інформаційні системи обліку даних про оренду житла ще більш цінним інструментом для власників нерухомості. Вони дозволили власникам нерухомості ефективніше управляти своїми активами та орендодавцями.

У 1980-х роках комп'ютерні програми для управління орендою житла стали більш доступними. Це призвело до того, що їх стали використовувати не лише великі власники нерухомості, а й дрібні орендодавці.

Це розширення ринку призвело до появи більшої конкуренції серед постачальників інформаційних систем обліку даних про оренду житла. Це, в свою чергу, призвело до зниження цін на ці системи та підвищення їхньої якості.

У 1990-х роках з'явилися перші веб-орієнтовані програми для управління орендою житла. Це дозволило власникам нерухомості відстежувати свої активи та орендарів з будь-якого місця в Інтернеті.

Ця нова функція була особливо цінною для власників нерухомості, які мали кілька об'єктів оренди в різних місцях. Вона дозволила їм ефективно управляти своїм бізнесом з одного центрального місця [3].

У 2000-х роках інформаційні системи обліку даних про оренду житла стали ще більш потужними та функціональними. Вони стали використовуватися не лише для управління орендою, а й для аналізу даних про ринок оренди.

Ці нові функції дозволили власникам нерухомості приймати більш обґрунтовані рішення про ціноутворення, управління активами та маркетинг.

У 2010-х роках інформаційні системи обліку даних про оренду житла стали ще більш зручними та інтуїтивно зрозумілими. Вони стали використовуватися не лише для власників нерухомості, а й для потенційних орендарів.

Ці зміни зробили інформаційні системи обліку даних про оренду житла більш доступними та привабливими для широкого кола користувачів.

У 2020-х роках інформаційні системи обліку даних про оренду житла продовжують розвиватися. Вони стають все більш інтелектуальними та здатними до самостійного навчання [2].

Ці нові функції дозволяють інформаційним системам обліку даних про оренду житла автоматизувати ще більше завдань та давати власникам нерухомості більш точні та корисні дані.

Основними тенденціями розвитку інформаційних систем обліку даних про оренду житла є:

- Інформаційні системи обліку даних про оренду житла стають все більш доступними, що дозволяє використовувати їх не лише великим власникам нерухомості, а й дрібним орендодавцям.

- Інформаційні системи обліку даних про оренду житла все частіше інтегруються з іншими системами, такими як системи управління нерухомістю та системи бронювання. Це дозволяє власникам нерухомості ефективно керувати своїм бізнесом.

- Штучний інтелект використовується для розвитку нових функцій ІСОДОЖ (інформаційна система обліку даних про оренду житла), таких як автоматизація завдань, прогнозування попиту та управління ризиками.

Ці тенденції свідчать про те, що інформаційні системи обліку даних про оренду житла будуть продовжувати розвиватися і ставати ще більш цінним інструментом для власників нерухомості [6].



## **1.3 Причини використання інформаційних систем для задач обліку даних про оренду житла**

### **1.3.1 Загальна характеристика комплексу причин**

Використання інформаційних систем обліку даних про оренду житла має безліч переваг, які полегшують та оптимізують управління нерухомістю. Ці системи дозволяють систематизувати та легко вести облік усіх аспектів оренди житла, включаючи договори, фінансові транзакції, технічний стан об'єктів та інше. Автоматизація рутинних операцій, таких як ведення обліку оплати, формування звітності чи нагадування про терміни дії договорів, дозволяє значно зменшити час і зусилля, витрачені на управління нерухомістю [6].

Ось детальніший огляд основних причин, чому такі системи є необхідним елементом в сучасному управлінні орендною діяльністю:

1. Ефективний облік даних. Інформаційні системи дозволяють систематизувати та легко вести облік усіх аспектів оренди житла, включаючи договори, фінансові транзакції, технічний стан об'єктів та інше. Зберігання всієї необхідної інформації в централізованій базі даних – робить доступ до неї простим і ефективним. Легке створення, зберігання та відстеження договорів оренди, у тому числі їх термінів, умов та змін.

2. Оптимізація багатьох рутинних процесів, таких як ведення обліку оплати, формування звітності чи нагадування про терміни дії договорів, дозволяє значно зменшити час і зусилля, витрачені на управління нерухомістю.

3. Забезпечення точності та змістовності даних. Інформаційні системи мінімізують ризик помилок та дублювання даних, забезпечуючи точність та змістовність інформації про орендні об'єкти та учасників процесу.

4. Підвищення доступності інформації. Зручний та швидкий доступ до даних забезпечується завдяки централізованій базі даних, що полегшує взаємодію між різними учасниками, такими як орендарі, орендодавці, бухгалтери та менеджери нерухомості [4].

5. Інформаційні системи надають можливість контролю доступу до конфіденційної інформації, забезпечуючи високий рівень безпеки даних та виключаючи несанкціонований доступ.

6. Інтегровані аналітичні інструменти дозволяють проводити аналіз ефективності орендних операцій, прогнозувати попит та забезпечувати підтримку при прийнятті управлінських рішень.

7. Покращення відносин з клієнтами. Інформаційні системи надають засоби для ефективної комунікації з клієнтами, відслідковування їхніх потреб та надання якісного обслуговування.

Всі ці переваги роблять інформаційні системи обліку даних про оренду житла невід'ємною частиною сучасного управління нерухомістю, сприяючи підвищенню ефективності та оптимізації бізнес-процесів в цій галузі .

### **1.3.2 Підвищення ефективності**

Ефективний облік даних у сфері оренди житла забезпечується за допомогою інформаційних систем, які виконують комплексне управління всіма аспектами цього бізнесу. Ці системи дозволяють систематизувати та легко вести облік різних елементів, починаючи від договорів і закінчуючи технічним станом об'єктів [7].

Одним із ключових аспектів є облік договорів оренди. Інформаційні системи забезпечують можливість зберігання та ефективного керування усіма деталями договорів, включаючи дані про орендарів, строки дії угоди та фінансові умови. Це робить процес управління договорами більш прозорим та ефективним.

Фінансові транзакції, пов'язані з орендою, також піддаються ефективному обліку. Інформаційні системи дозволяють фіксувати та моніторити всі грошові перекази, пов'язані з орендною діяльністю. Це включає в себе оплату оренди, зміни в фінансових угодах та інші фінансові взаємодії.

Технічний стан об'єктів нерухомості також враховується в інформаційних системах. Регулярні технічні інспекції, інформація про проведені ремонтні

роботи та плани збереження технічної надійності – усе це заноситься в систему для забезпечення акуратного та систематизованого контролю.

Крім того, інформаційні системи полегшують комунікацію з орендарями. Засоби для обміну повідомленнями, автоматичні нагадування про платежі та інформаційні портали для орендарів полегшують взаємодію та сприяють більш ефективній комунікації [7].

Найважливішою перевагою інформаційних систем є їхня аналітична та звітна функціональність. Завдяки цьому, власники та менеджери можуть отримувати детальну статистику щодо ефективності орендного бізнесу, визначати ключові показники продуктивності та приймати обґрунтовані управлінські рішення.

Усе це разом утворює надійну та ефективну основу для управління орендною діяльністю, що впливає на підвищення ефективності бізнесу та задоволеність всіх сторін угоди.

### **1.3.3 Використання аналітичних інструментів**

Підтримка аналітики та прийняття рішень в контексті інформаційних систем обліку даних про оренди житла є важливою складовою для досягнення ефективного управління нерухомістю. Інтегровані аналітичні інструменти, доступні в таких системах, відіграють ключову роль у забезпеченні власників та менеджерів надійною та змістовною інформацією для прийняття обґрунтованих рішень.

Аналіз ефективності орендних операцій є важливою частиною аналітичних можливостей. Системи обліку дозволяють враховувати різні показники ефективності, включаючи вартість оренди, рівень наповненості об'єктів, темпи зростання прибутків, а також витрати на утримання нерухомості. Це надає можливість визначати стратегії розвитку, коригувати орендні ставки та вчасно реагувати на зміни на ринку [6].

Прогнозування попиту є іншим аспектом аналітики, який дозволяє власникам та менеджерам приймати передчасні рішення для оптимізації

наповненості об'єктів. За допомогою аналітичних моделей та обробки великого обсягу даних, системи можуть надавати прогнози та рекомендації щодо взаємодії з ринком.

Підтримка при прийнятті управлінських рішень полягає в забезпеченні високоякісною інформацією для прийняття обґрунтованих управлінських рішень. Власники та менеджери можуть використовувати аналітичні дані для розуміння ефективності різних стратегій, визначення найбільш вдалих інвестицій та планування майбутніх дій.

Аналіз витрат та прибутковості окремих орендних угод дозволяє глибоко розуміти фінансовий стан кожного об'єкта та здійснювати докладний облік рентабельності. Це дозволяє власникам розуміти, які об'єкти приносять найбільше прибутку та як оптимізувати свій портфель нерухомості [4].

В цілому, підтримка аналітики та прийняття рішень через інформаційні системи обліку даних про оренди житла стає ключовим фактором у формуванні успішного та стратегічного підходу до управління нерухомістю. Вона допомагає уникнути помилок, оптимізувати процеси та досягти максимальної ефективності бізнесу в галузі нерухомості.

#### **1.3.4 Покращення комунікацій з клієнтами**

Покращення відносин з клієнтами у сфері оренди житла завдяки інформаційним системам є важливим аспектом для забезпечення високого рівня обслуговування та задоволеності клієнтів. Інформаційні системи, розроблені для управління даними про оренду житла, пропонують низку можливостей, спрямованих на покращення взаємодії та підтримки клієнтів [3].

Ефективна комунікація є ключовою частиною цього процесу, і інформаційні системи надають різноманітні засоби для взаємодії з клієнтами. Це може включати в себе інтерактивні портали, чати, електронні листування та інші канали зв'язку, що дозволяють клієнтам отримувати швидкі та зручні відповіді на свої запитання та потреби.

Зібрання та аналіз даних про клієнтів дозволяє відслідковувати їхні потреби та переваги. Інформаційні системи можуть автоматично аналізувати історію попиту клієнта, їхні уподобання та фінансові можливості, щоб надавати персоналізовані пропозиції та обслуговування.

Персоналізоване обслуговування, побудоване на основі зібраних даних, дозволяє пропонувати індивідуальні рішення для кожного клієнта. Наприклад, система може рекомендувати конкретні об'єкти нерухомості відповідно до індивідуальних уподобань та історії попиту [6].

Ефективна обробка запитів та скарг також стає можливою завдяки інформаційним системам. Автоматизація процесів допомагає зменшити час очікування відповіді та покращує загальний рівень обслуговування.

Системи зворотного зв'язку, вбудовані в інформаційні системи, дозволяють збирати відгуки та оцінки клієнтів. Це не лише допомагає вимірювати рівень задоволення, але й надає можливість отримати конструктивні пропозиції для подальшого вдосконалення сервісу.

Загалом, інформаційні системи обліку даних про оренди житла сприяють вдосконаленню відносин з клієнтами, забезпечуючи ефективну комунікацію, персоналізоване обслуговування та зручний механізм збору та обробки відгуків. Це допомагає компаніям зберігати та залучати клієнтів, сприяючи створенню позитивного досвіду та розвитку довгострокових відносин.

#### **1.4 Сучасний стан використовуваних технологій**

У сучасному світі, де нерухомість виступає як важливий складник економіки, ефективне управління процесами оренди житла набуває все більшого значення. Щоб відповісти на різноманітні виклики та потреби цієї галузі, інформаційні системи обліку даних про оренду житла стали ключовим інструментом для бізнесу та управлінців нерухомості [10].

Сучасний стан цих інформаційних систем свідчить про їхню високу функціональність та адаптованість до вимог сучасного ринку нерухомості. Розглянемо докладніше ключові аспекти та тенденції, що визначають їхню

сучасну ефективність та вплив на управління процесами оренди житла. Ось деякі аспекти сучасного стану таких систем:

- Застосування хмарних технологій дозволяє зберігати та обробляти великі обсяги даних онлайн, полегшуючи доступ до інформації з будь-якого місця та пристрою.

- Розвиток мобільних технологій призвів до створення мобільних додатків для управління орендною нерухомістю. Орендарі та орендодавці можуть легко взаємодіяти та отримувати доступ до важливої інформації за допомогою смартфонів та планшетів.

- Модерні системи обліку оренди житла можуть легко інтегруватися з іншими інформаційними системами, такими як системи фінансового обліку чи онлайн-платежів.

- Вбудовані інструменти бізнес-аналітики дозволяють здійснювати глибокий аналіз даних, виявляти тенденції, оцінювати ефективність стратегій оренди та приймати обґрунтовані рішення.

- Звернення до передових методів шифрування та заходів безпеки допомагає запобігати несанкціонованому доступу та забезпечує конфіденційність важливої інформації.

- Сучасні системи обліку дозволяють генерувати різноманітні звіти, які спрощують моніторинг фінансового стану, стану об'єктів оренди та інших ключових показників.

Перейдемо до питання використання хмарних технологій для системи обліку оренди житла.

## **1.5 Використання хмарних технологій**

Застосування хмарних технологій в інформаційних системах обліку даних про оренду житла представляє собою перехід від традиційних методів зберігання та обробки даних до більш гнучких та ефективних рішень. Цей підхід принесе численні переваги для управління нерухомістю та орендними операціями.

Однією з ключових переваг є доступність даних з будь-якого місця та пристрою. Хмарні технології дозволяють власникам нерухомості, менеджерам та орендарям отримувати доступ до інформації через веб-інтерфейси чи мобільні додатки, полегшуючи спільну роботу та моніторинг стану об'єктів оренди [11].

Забезпечення безпеки даних – ще одна важлива перевага хмарних технологій. Великі постачальники хмарних послуг інвестують в передові заходи безпеки, такі як шифрування даних, двофакторна аутентифікація та системи моніторингу безпеки, що робить їх привабливими для зберігання конфіденційної інформації про нерухомість та орендні угоди.

Скасування необхідності великих капіталовкладень в інфраструктуру – ще одна перевага хмарних технологій. Використання віддалених серверів для зберігання та обробки даних дозволяє уникнути витрат на встановлення та підтримку великих серверних інфраструктур.

Масштабованість та гнучкість є ще однією важливою характеристикою хмарних технологій. Зміна обсягу даних чи розширення функціоналу системи можливе без значних зусиль, завдяки готовності хмарних сервісів до масштабування. Крім того, хмарні технології дозволяють легко інтегрувати різноманітні інформаційні системи, такі як системи фінансового обліку чи системи аналітики, що полегшує обмін даними та узгодженість роботи різних частин бізнесу [10].

Впровадження хмарних технологій в інформаційні системи обліку даних про оренду житла сприяє підвищенню доступності та ефективності управління нерухомістю, що відповідає вимогам сучасного динамічного ринку оренди.

## **1.6 Використання технологій штучного інтелекту**

Використання штучного інтелекту (ШІ) в інформаційних системах обліку даних про оренду житла призводить до переваг, що трансформують спосіб управління та оптимізації орендного бізнесу. Однією з ключових областей застосування ШІ є аналіз великих обсягів даних для прогнозування ринкових тенденцій та визначення оптимальних стратегій оренди.

Алгоритми ШІ можуть враховувати різноманітні фактори, такі як динаміка цін на ринку, зміни у попиті та пропозиції, економічні та соціальні тенденції. Це дозволяє системам прогнозувати ринкові умови та адаптувати стратегії оренди в реальному часі, щоб максимізувати прибуток та підтримувати конкурентоспроможність [10].

Ще однією ключовою перевагою є використання ШІ для оптимізації ціноутворення. Системи можуть автоматично аналізувати конкурентоспроможність ринкових цін, враховуючи різні фактори, такі як розташування нерухомості, її характеристики та умови довгострокової оренди. Це сприяє оптимізації ціноутворення для максимізації прибутковості та привабливості об'єктів для орендарів.

ШІ використовується і для автоматизації обслуговування клієнтів в інформаційних системах оренди житла. Чат-боти та віртуальні асистенти можуть автоматично відповідати на запитання клієнтів, надавати інформацію про умови оренди та вести комунікацію через різні канали, що спрощує процеси комунікації та покращує обслуговування клієнтів.

Оцінка кредитоспроможності також стає більш точною та ефективною завдяки ШІ. Системи можуть аналізувати широкий спектр фінансових та особистих даних для швидкого та об'єктивного прийняття рішень щодо можливостей оренди [6].

Щодо управління витратами, ШІ може допомагати визначати ефективні стратегії управління енергоспоживанням та розподілу ремонтних робіт. Аналіз даних дозволяє зменшити витрати та забезпечити оптимальне використання ресурсів для забезпечення комфорту та збереження стану нерухомості.

Використання ШІ в інформаційних системах обліку даних про оренду житла позначає суттєвий крок вперед у напрямку інтелектуалізації та оптимізації управління орендною нерухомістю, підвищуючи ефективність та конкурентоспроможність бізнесу.



## 1.7 Висновок до першого розділу

В першому розділі кваліфікаційної роботи розглянуто різні аспекти інформаційних систем обліку даних про оренду житла та їхніх технологічних вдосконалень. Починаючи з історичного розвитку і переходячи до сучасного стану, ми розглянули важливі аспекти використання інформаційних систем у цій сфері.

Застосування штучного інтелекту та хмарних технологій значно покращило ефективність управління нерухомістю та орендним бізнесом. ШІ дозволяє здійснювати аналіз даних, прогнозування ринкових тенденцій, оптимізацію ціноутворення та автоматизацію процесів, тим самим забезпечуючи більш точне та ефективне управління.

Хмарні технології за своєю структурою забезпечують доступність та безпеку зберігання даних, роблячи їх доступними з будь-якого пристрою та місця. Вони дозволяють використовувати ресурси в інтересах управління нерухомістю, зменшуючи витрати на інфраструктуру та підтримку систем.

Загалом, використання передових технологій в інформаційних системах обліку даних про оренду житла є ключовим чинником у вдосконаленні управлінських процесів та підвищенні конкурентоспроможності в сучасному ринковому середовищі.

## **2 АНАЛІЗ МЕТОДІВ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА**

### **2.1 Загальні підходи та методи досліджень**

При аналізі методів проектування інформаційних систем обліку даних про оренду житла слід звертати увагу на різні аспекти, охоплюючи функціональність системи, аспекти безпеки, зручність використання та інші фактори. Загалом, поділяють на чотири основні етапи аналізу методів проектування інформаційних систем [9].

#### **2.1.1 Початковий етап аналізу методів проектування інформаційних систем**

Початковий етап аналізу методів проектування інформаційних систем для обліку даних про оренду житла передбачає визначення конкретних вимог, які повинна виконувати система. Це охоплює як функціональні, так і нефункціональні аспекти.

У процесі визначення функціональних вимог формуються основні завдання системи, такі як реєстрація користувачів, введення та збереження інформації про нерухомість, облік та відстеження орендної оплати. Також важливо враховувати потреби різних сторін, таких як орендарі, власники, та можливо орендні агенти чи менеджери [13].

Нефункціональні вимоги визначають аспекти продуктивності, безпеки та зручності використання системи. Наприклад, можуть бути встановлені вимоги до часу відповіді системи, рівня захищеності конфіденційної інформації, або зручного інтерфейсу користувача.

Ключовим етапом є вивчення бізнес-процесів в сфері оренди житла. Це дозволяє зрозуміти, як система може взаємодіяти з існуючими процесами та які саме вдосконалення можна ввести для оптимізації ефективності та забезпечення високоякісного обліку оренди.

Важливим аспектом є також вибір технологій для реалізації інформаційної системи. Це включає в себе вивчення сучасних баз даних, мов програмування та фреймворків, щоб забезпечити оптимальну функціональність та довгострокову стабільність системи.

Останнім етапом на цьому етапі є розгляд питань безпеки. Необхідно розробити стратегії для захисту від несанкціонованого доступу, забезпечення конфіденційності та цілісності даних, а також забезпечити відповідність стандартам безпеки [13].

Цей комплексний підхід на початковому етапі аналізу гарантує належне визначення вимог та забезпечення ефективного розроблення інформаційної системи для обліку даних про оренду житла.

### **2.1.2 Другий етап аналізу методів проектування інформаційних систем**

Другий етап аналізу методів проектування інформаційних систем для обліку даних про оренду житла полягає в детальному вивченні існуючих бізнес-процесів, що стосуються оренди нерухомості. Цей етап важливий для зрозуміння та визначення, як саме інформаційна система може взаємодіяти з усіма аспектами орендного бізнесу, щоб забезпечити оптимальну ефективність та вдосконалити робочі процеси [17].

На даному етапі проводиться аналіз поточних процесів, пов'язаних із задачею в оренду та управлінням нерухомістю. Це включає вивчення всіх етапів від пошуку нерухомості до укладання орендних угод та подальшого управління.

Також важливо враховувати взаємодію з різними учасниками процесу, такими як орендарі, власники, агенти з нерухомості. Розуміння їхніх потреб і очікувань дозволяє враховувати різні підходи та можливості при розробці системи.

Важливим аспектом аналізу є ідентифікація можливих вузьких місць чи проблем в поточних бізнес-процесах. Це може включати затримки у виконанні певних завдань, недоліки в організації інформаційного потоку, або неефективне використання ресурсів [16].

Здійснення глибокого аналізу бізнес-процесів дозволяє визначити потреби та очікування користувачів системи, а також визначити ті області, в яких інформаційна система може допомогти в оптимізації та автоматизації завдань, що стосуються оренди житла.

### **2.1.3 Третій етап аналізу методів проектування інформаційних систем**

Третій етап аналізу методів проектування інформаційних систем для обліку даних про оренду житла включає вибір технологій, які будуть використані для реалізації інформаційної системи. Цей етап є критичним, оскільки правильний вибір технологій може суттєво вплинути на функціональність, продуктивність та довгострокову життєздатність системи.

В процесі вибору технологій слід враховувати сучасні тенденції в галузі розробки програмного забезпечення, доступність і популярність різних мов програмування, баз даних і фреймворків. Також важливо враховувати, наскільки легко можна інтегрувати вибрані технології з існуючими системами чи сервісами [13].

Вибір мов програмування може визначати зручність розробки та підтримки системи, а вибір бази даних може впливати на ефективність та швидкодію системи. Фреймворки дозволяють швидше та ефективніше розробляти функціональність системи, але їхні особливості також слід враховувати.

Окрім того, слід враховувати можливість масштабування системи у майбутньому та її готовність до використання новітніх технологій, таких як штучний інтелект, блокчейн тощо.

Вибір технологій також може бути визначений бюджетом та ресурсами, які доступні для розробки та підтримки системи. Важливо забезпечити збалансований підхід, обираючи технології, які відповідають потребам та можливостям вашого проекту.

#### **2.1.4 Четвертий етап аналізу методів проектування інформаційних систем**

Четвертий етап аналізу методів проектування інформаційних систем для обліку даних про оренду житла стосується аспектів безпеки інформації. Це важливий етап, оскільки забезпечення конфіденційності, цілісності та доступності даних визначається якістю та надійністю інформаційної системи [16].

Аналіз безпеки включає в себе ретельне вивчення потенційних загроз та вразливостей системи. Це може бути пов'язано з ризиками несанкціонованого доступу до даних, атаками на інфраструктуру, втратою даних через несприятливі події, такі як аварії чи катастрофи [13].

Стратегії безпеки повинні визначити заходи, призначені для запобігання атакам, виявлення та реагування на інциденти, а також відновлення роботи системи після випадку втрати даних чи інших подій.

Крім того, важливо забезпечити дотримання відповідних стандартів і законодавства у сфері безпеки даних, зокрема щодо персональної інформації орендарів та власників нерухомості. Застосування різноманітних методів шифрування, контролю доступу та аудиту дозволяє створити ефективну систему захисту даних. Також важливо регулярно оновлювати програмне забезпечення, включаючи всі компоненти системи, для усунення виявлених вразливостей та забезпечення високого рівня безпеки [14].

Необхідно відзначити, що аналіз безпеки є постійним і невпинним процесом, оскільки сучасні загрози та атаки постійно змінюють свою природу та вдосконалюють свої методи. В цьому контексті важливо усвідомлювати, що технологічні ризики ніколи не залишаються сталими, і організації повинні бути завжди насторожі щодо потенційних загроз, які можуть виникнути в будь-який момент. Такий постійний характер еволюції загроз вимагає від інформаційних систем гнучкості та готовності до швидкої реакції на нові виклики. Організації повинні вдосконалювати свої заходи захисту, враховуючи останні тенденції.

## 2.2 Використання сучасних методологій розробки

### 2.2.1 Загальна характеристика методологій

Дослідження та порівняння різних методологій розробки програмного забезпечення для обліку даних про оренду житла є важливим етапом при визначенні оптимального підходу до розробки системи. Різні методології мають свої переваги та недоліки, які можуть бути важливими в контексті конкретного проекту [18].

Один із найпопулярніших підходів – це методологія Agile. Дана методологія відома своєю гнучкістю та можливістю адаптації до змін. Вона розглядає розробку як ітеративний процес, де функціональність може бути швидко впроваджена та оновлюватися на основі змінних вимог. Для проекту з обліку даних про оренду житла це може бути корисним, оскільки бізнес-вимоги можуть змінюватися в залежності від потреб ринку чи стратегії компанії.

Іншим підходом є Waterfall, який передбачає лінійний порядок розробки, де кожен етап послідовно виконується після завершення попереднього. Цей метод може бути корисним, якщо вимоги до проекту достатньо стабільні та визначені заздалегідь. Однак, у сфері оренди житла, де ринкові умови можуть швидко змінюватися, Waterfall може виявитися менш ефективним [17].

Методологія Scrum, яка є однією з Agile-підходів, використовує короткі ітерації, відомі як "спринти", для швидкої поставки функціоналу. Це може бути корисно в проектах, де важливо швидко впроваджувати нові функції чи змінювати існуючі відповідно до потреб користувачів чи бізнесу.

DevOps – це підхід, який поєднує розробку та операції в єдиний цикл. Це може полегшити впровадження та підтримку системи, забезпечуючи швидше виправлення помилок та постійне вдосконалення.

При виборі методології важливо враховувати конкретні потреби та характеристики проекту. Наприклад, для проектів з оренди житла може бути корисним використання Agile-підходів для швидкого реагування на зміни в ринкових умовах.

## 2.2.2 Використання методології Agile для інформаційних систем обліку даних про оренди житла

Методологія Agile є гнучким підходом до розробки програмного забезпечення, який активно використовується в сучасному інформаційному середовищі, включаючи системи обліку даних про оренди житла. Основна ідея Agile полягає в тому, щоб робити розробку більш адаптивною, реагуючи на зміни вимог і взаємодіючи з клієнтом на кожному етапі розробки.

Однією з ключових характеристик Agile є розбиття проекту на короткі ітерації, відомі як "спринти". Кожен спринт зазвичай триває від двох до чотирьох тижнів і включає в себе повний цикл розробки, від визначення вимог до виведення готового продукту.

Спринти дозволяють ефективно впроваджувати нові функції та вносити корективи, сприяючи більшому заохоченню до змін в процесі розробки. У випадку систем обліку даних про оренди житла це може бути надзвичайно корисним, оскільки бізнес-вимоги та правила можуть змінюватися в залежності від ринкових умов та потреб користувачів [17].

Ще однією ключовою характеристикою є активне включення клієнта чи замовника в процес розробки. Замість традиційного підходу "визначення вимог і виконання", Agile включає замовника на кожному етапі, отримуючи від нього зворотний зв'язок та корекції в реальному часі. У сфері оренди житла це дозволяє більш ефективно враховувати зміни в умовах ринку, вимогах орендарів та інших факторах.

Також Agile сприяє розвитку багатofункціональних команд, які працюють разом над усіма аспектами проекту. Це забезпечує гнучкість у розподілі завдань і сприяє більшому взаєморозумінню між різними командами, наприклад, між розробниками, тестувальниками та аналітиками.

Використання Agile в системах обліку даних про оренду житла дозволяє створювати продукт, який краще відповідає динаміці бізнес-середовища, швидше реагує на потреби користувачів і є більш адаптивним до змін у вимогах.

### **2.2.3 Методологія Waterfall**

Методологія Waterfall є лінійним та послідовним підходом до розробки інформаційних систем. Вона передбачає чітко визначений порядок виконання етапів, починаючи з аналізу вимог та закінчуючи етапом експлуатації.

На етапі аналізу проводиться глибоке вивчення вимог бізнесу та користувачів. Важливо чітко визначити обсяг та характеристики системи обліку даних про оренди житла. Наступний етап проектування – включає в себе визначення архітектури системи та технічних специфікацій [18].

Реалізація або програмування включає написання коду відповідно до розроблених на попередніх етапах планів. Після програмування проводиться тестування, щоб перевірити відповідність системи визначеним вимогам та виявити можливі помилки.

Впровадження передбачає деплоймент системи в реальному середовищі, після чого вона переходить у режим експлуатації. На цьому етапі важливо забезпечити технічну підтримку та вирішення будь-яких проблем, що можуть виникнути в процесі роботи системи.

Методологія Waterfall дозволяє строго визначити послідовність та розподіл завдань, що може бути ефективним у випадках, коли вимоги відомі наперед. Однак вона може бути менш гнучкою, особливо у випадках, коли вимоги можуть змінюватися в процесі розробки.

### **2.2.4 Методологія Scrum**

Методологія Scrum є гнучким та ітеративним підходом до розробки програмного забезпечення, що добре підходить для інформаційних систем обліку даних про оренди житла.

Ключовою особливістю Scrum є фокус на колективній роботі та ітераційному покроковому підході. Команди формуються з крос-функціональних спеціалістів, які працюють разом над проектом. Спринти, або



ітерації, роблять розробку більш гнучкою, дозволяючи впроваджувати зміни та вдосконалення на кожному етапі [17].

Планування спринта визначає, які завдання команда може виконати протягом певного періоду. Це дозволяє ефективно розподіляти завдання та визначати пріоритети в процесі розробки.

Прозорість та звітність – це ключові аспекти Scrum. Зустрічі щоденно допомагають команді триматися на одній хвилі, ділитися інформацією та швидко реагувати на зміни чи проблеми.

Зміна пріоритетів між спринтами дозволяє команді адаптуватися до змін в вимогах або стратегії бізнесу. Це важливо для забезпечення того, що розробка відповідає поточним потребам та можливостям ринку.

Узагальнюючи, Scrum сприяє високому рівню гнучкості, співпраці та прозорості в розробці інформаційних систем обліку даних при оренди житла, що дозволяє створювати продукт відповідно до змінюваних вимог і досягати кращих результатів.

### **2.2.5 Методологія DevOps**

Методологія DevOps є сучасним підходом до розробки та управління інформаційними системами, який покликаний злити функції розробки та операцій в єдиний, гнучкий процес. У контексті інформаційних систем обліку даних при оренди житла, DevOps вносить ряд принципів та практик, спрямованих на поліпшення швидкості розробки, надійності та ефективності [18].

Одним із ключових аспектів DevOps є автоматизація процесів. Це охоплює автоматизацію тестування, збірки, впровадження та моніторингу. Автоматизація дозволяє зменшити людський фактор, забезпечуючи швидше виявлення та виправлення помилок.

Культура співпраці в DevOps відіграє ключову роль. Здатність розробників, тестувальників та операторів ефективно співпрацювати та обмінюватися інформацією робить процес більш гнучким та адаптивним.

Керування версіями та конфігурацією дозволяє ефективно відслідковувати та керувати змінами в кодї та конфігураціях системи. Це робить процес розробки стабільним та передбачуваним.

Моніторинг та аналіз грати важливу роль в DevOps. Систематичне збирання та аналіз даних дозволяє оперативно виявляти та усувати проблеми, а також оптимізувати продуктивність системи [16].

Принцип "інфраструктура як код" дозволяє управляти інфраструктурою так само, як програмним кодом, що полегшує розгортання та конфігурацію. DevOps активно підтримує концепцію постійної доставки, забезпечуючи можливість швидко та надійно постачати нові функції чи зміни у виробництво.

Узагальнюючи, DevOps не лише вносить зміни в технічні аспекти розробки, а й створює культуру, де швидкість, якість та співпраця визначають успіх.

## **2.3 Аналіз відомих систем обліку даних про оренду житла**

### **2.3.1 Загальна характеристика відомих систем**

Існує безліч автоматизованих систем обліку даних про оренду житла. Деякі з них є загальними, призначеними для будь-якого типу оренди, а інші спеціалізовані для певних типів нерухомості, таких як квартири, будинки, або комерційні приміщення [11].

Ось деякі з найпопулярніших автоматизованих систем обліку даних про оренди житла:

- Clarify – це система обліку оренди житла, яка призначена для управління всіма аспектами оренди, від пошуку орендарів до управління договорами та оплатою [11].
- Propertyware – це інша популярна система обліку оренди житла, яка пропонує широкий спектр функцій, включаючи управління нерухомістю, управління орендарями та управління фінансовими операціями [51].

- AppFolio – це система обліку оренди житла, яка спеціалізується на управлінні багатоквартирними будинками та комерційною нерухомістю [52].

- Rentometer – це система обліку оренди житла, яка спеціалізується на аналізі даних оренди та прогнозуванні цін [53].

- Zillow Rental Manager – це система обліку оренди житла, яка пропонує простий і доступний інтерфейс [54].

Ці системи пропонують широкий спектр функцій, які можуть допомогти власникам нерухомості та менеджерам оренди покращити ефективність своєї роботи та підвищити прибуток [10]. Деякі з найпоширеніших функцій включають:

- Управління нерухомістю – ці системи дозволяють власникам нерухомості та менеджерам оренди вести облік своєї нерухомості, включаючи адресу, розмір, тип і ціну.

- Управління орендарями – ці системи дозволяють власникам нерухомості та менеджерам оренди вести облік своїх орендарів, включаючи особисту інформацію, контактні дані та історію платежів.

- Управління договорами – ці системи дозволяють власникам нерухомості та менеджерам оренди створювати, зберігати та управляти договорами оренди.

- Управління оплатою – ці системи дозволяють власникам нерухомості та менеджерам оренди приймати платежі від орендарів, включаючи онлайн-платежі, банківські перекази та чеки.

- Аналіз даних – ці системи дозволяють власникам нерухомості та менеджерам оренди аналізувати дані оренди, щоб отримати уявлення про ринок оренди та свою власну діяльність.

При виборі автоматизованої системи обліку даних про оренди житла важливо враховувати свої конкретні потреби та вимоги. Деякі фактори, які слід враховувати, включають:

- 1) тип нерухомості яка орендується;
- 2) кількість нерухомості;
- 3) кількість орендарів;
- 4) фінансові можливості.

Проаналізувавши подані пункти, сформується основні критерії для розробки інформаційної системи для обліку даних про оренди житла.

### **2.3.2 Інтерфейс користувача**

UI (Інтерфейс користувача) – це те, як система взаємодіє з користувачем. Він включає в себе дизайн, організацію та взаємодію елементів системи. Інтерфейс користувача є важливим фактором, який слід враховувати при виборі системи обліку даних про оренду житла. Зручний і інтуїтивно зрозумілий інтерфейс допоможе швидко та легко освоїти систему та розпочати роботу [16].

Усі п'ять систем, які були розглянуто вище, мають сучасний і інтуїтивно зрозумілий інтерфейс користувача. Вони доступні як на веб-платформі, так і на мобільних пристроях.

Clarify, Propertyware та AppFolio мають особливо зручний і інтуїтивно зрозумілий інтерфейс користувача. Він добре продуманий і організований, що робить його легким у використанні.

Rentometer має простий і зрозумілий інтерфейс користувача. Він підходить для користувачів, які не мають досвіду роботи з системами управління нерухомістю.

Zillow Rental Manager також має простий і зрозумілий інтерфейс користувача. Він підходить для власників нерухомості, які хочуть самостійно управляти своєю нерухомістю [17].

Інтерфейс користувача є важливим фактором, який слід враховувати при виборі системи обліку даних про оренду житла. Зручний і інтуїтивно зрозумілий інтерфейс допоможе швидко та легко освоїти систему та розпочати роботу.

### **2.3.3 Функціональність**

Функціональність – це ще один важливий фактор, який слід враховувати при виборі системи обліку даних про оренду житла. Система повинна

відповідати вашим конкретним потребам та вимогам. Усі п'ять систем, які були розглянуті вище, пропонують широкий спектр функцій, включаючи:

Управління нерухомістю. Ведення обліку нерухомості, включаючи адресу, розмір, тип і ціну. Створення та управління профілями нерухомості, включаючи фотографії, описи і відомості про наявність [11].

Управління орендарями. Ведення обліку орендарів, включаючи особисту інформацію, контактні дані та історію платежів. Створення та управління заявками на оренду, включаючи анкети, рекомендації і звіти про перевірки.

Управління договорами. Створення, зберігання та управління договорами оренди. Створення та управління шаблонами договорів, а також відстежування терміну дії договорів.

Управління оплатою. Прийняття платежів від орендарів, включаючи онлайн-платежі, банківські перекази та чеки. Створення та управління платежами, включаючи історію платежів і звіти про несплати.

Аналіз даних. Аналіз даних оренди, щоб отримати уявлення про ринок оренди та свою власну діяльність. Створення і управління звітами про аналіз даних, включаючи звіти про прибуток, вартість і продуктивність [19].

Clarify, Propertyware та AppFolio пропонують найбільш широкий спектр функцій. Вони можуть задовольнити потреби як малих, так і великих компаній з управління нерухомістю.

Rentometer спеціалізується на аналізі даних оренди та прогнозуванні цін. Вона особливо підходить для компаній з управління нерухомістю, які хочуть приймати обґрунтовані рішення про ціноутворення [17].

Zillow Rental Manager пропонує базовий набір функцій, який підходить для малих і середніх компаній.

Ось деякі конкретні функції, які слід враховувати при виборі системи обліку даних про оренду житла:

- Чи може система вести облік нерухомості за такими параметрами, як тип, розмір, адреса, ціна?
- Чи може система створювати і управляти профілями нерухомості, включаючи фотографії, описи і відомості про наявність?

- Чи може система вести облік орендарів за такими параметрами, як ім'я, адреса, контактні дані, історія платежів?
- Чи може система створювати і управляти заявками на оренду, включаючи анкети, рекомендації і звіти про перевірки?
- Чи може система створювати, зберігати і управляти договорами оренди?
- Чи може система створювати і управляти шаблонами договорів, а також відстежувати термін дії договорів?
- Чи може система приймати платежі від орендарів такими способами, як онлайн-платежі, банківські перекази та чеки?
- Чи може система створювати і управляти платежами, включаючи історію платежів і звіти про несплати?
- Чи може система аналізувати дані оренди, щоб отримати уявлення про ринок оренди та свою власну діяльність?
- Чи може система створювати і управляти звітами про аналіз даних, включаючи звіти про прибуток, вартість і продуктивність?

Функціональність є важливим фактором, який слід враховувати при виборі системи обліку даних про оренди житла. Система повинна відповідати конкретним потребам та вимогам.

#### **2.3.4 Гнучкість системи**

Гнучкість системи – це важливий фактор, який слід враховувати при виборі системи обліку даних про оренду житла. Система повинна бути здатною адаптуватися до ваших потреб у міру їх зростання.

Clarify, Propertyware та AppFolio є масштабованими системами, які можуть адаптуватися до потреб різних компаній з управління нерухомістю. Вони пропонують широкий спектр функцій і налаштувань, які дозволяють вам адаптувати систему до своїх конкретних потреб [17].

Rentometer і Zillow Rental Manager є менш масштабованими системами, які підходять для малих і середніх компаній. Вони пропонують менш широкий

спектр функцій і налаштувань, ніж Clarify, Propertyware та AppFolio. У таблиці 2.1 подано порівняння систем за гнучкістю.

Таблиця 2.1 – Порівняння гнучкості запропонованих систем

<b>Система</b>	<b>Масштабованість</b>	<b>Функції та налаштування</b>
Clarify	Так	Широкий спектр функцій і налаштувань
Propertyware	Так	Широкий спектр функцій і налаштувань
AppFolio	Так	Широкий спектр функцій і налаштувань
Rentometer	Ні	Обмежений спектр функцій
Zillow Rental Manager	Ні	Обмежений спектр функцій і налаштувань

Якщо потрібна система, яка може адаптуватися до потреб у міру їх зростання, то Clarify, Propertyware або AppFolio будуть хорошим вибором.

Якщо потрібна система з базовим набором функцій, яка підходить для малих і середніх компаній, то Rentometer або Zillow Rental Manager будуть хорошим вибором.

### **2.3.5 Безпека, надійність та швидкість**

Безпека є важливим фактором, який слід враховувати при виборі будь-якої системи, яка обробляє конфіденційну інформацію. Усі п'ять систем, які були розглянуті, пропонують високий рівень безпеки. У системах використовують сучасні технології безпеки, такі як шифрування даних, багатфакторна аутентифікація та захист від зловмисного програмного забезпечення [13].

Деякі конкретні особливості безпеки, які пропонують ці системи, включають:

- Шифрування даних. Дані шифруються під час зберігання та передачі, щоб захистити їх від несанкціонованого доступу.

- Багатофакторна аутентифікація. Вимагає від користувачів ввести кілька факторів аутентифікації, таких як пароль, код з мобільного телефону та ключ безпеки, щоб отримати доступ до системи.

- Захист від зловмисного програмного забезпечення. Використовує антивірусне програмне забезпечення та інші технології для захисту від зловмисного програмного забезпечення.

Надійність є важливим фактором, який слід враховувати при виборі будь-якої системи, яка буде використовуватись щодня. Усі п'ять систем є надійними. Вони мають високу доступність і низький рівень відмов [18].

Деякі конкретні особливості надійності, які пропонують ці системи, включають:

- Висока доступність, системи доступні 24/7/365;
- Низький рівень відмов;
- Системи регулярно оновлюються для усунення помилок і вразливостей.

Швидкість є важливим фактором, який слід враховувати при виборі будь-якої системи, яку ви будете використовувати для обробки великої кількості даних. Усі п'ять систем є швидкими. Вони можуть обробляти великі обсяги даних швидко і ефективно. Деякі конкретні особливості швидкості, які пропонують ці системи, включають:

- Системи використовують інноваційні технології, такі як хмарне обчислення, для підвищення продуктивності;
- Системи оптимізовані для підвищення продуктивності;
- Системи регулярно оновлюються для підвищення продуктивності.

Усі п'ять систем, які були розглянуті вище, пропонують широкий спектр функцій, гнучкість і безпеку. Вони є надійними і швидкими. Якщо потрібна система з широким спектром функцій і можливістю масштабування, то Clarify, Propertyware або AppFolio будуть хорошим вибором [17].

Якщо потрібна система з базовим набором функцій, яка підходить для малих і середніх компаній, то Rentometer або Zillow Rental Manager будуть більш



доступними варіантами. Остаточне рішення залежить від конкретних потреб і бюджету.

### 2.3.6 Вартість реалізації

Вартість – це важливий фактор, який слід враховувати при виборі системи обліку даних про оренду житла. Вартість тарифних планів може варіюватися залежно від кількості об'єктів нерухомості, кількості орендарів і функцій, які потрібні. Системи Clarify, Propertyware та AppFolio мають найвищі тарифні плани. Детальне порівняння вартості продемонстровано у таблиці 2.2 [17].

Таблиця 2.2 – Порівняння вартості тарифів

Система	Тарифний план	Ціна
Clarify	Starter	\$99/місяць
Clarify	Professional	\$199/місяць
Clarify	Enterprise	\$299/місяць
Propertyware	Essentials	\$199/місяць
Propertyware	Pro	\$299/місяць
Propertyware	Enterprise	\$399/місяць
AppFolio	Essential	\$199/місяць
AppFolio	Professional	\$299/місяць
AppFolio	Enterprise	\$399/місяць
Rentometer	Basic	\$29/місяць
Rentometer	Pro	\$59/місяць
Zillow Rental Manager	Starter	\$39/місяць
Zillow Rental Manager	Pro	\$59/місяць

Крім кількості об'єктів нерухомості, кількості орендарів і функцій, вартість системи обліку даних про оренду житла може також залежати від таких факторів:

- Розмір компанії. Великі компанії, як правило, платять більше, ніж малі і середні компанії.

- Місцезнаходження компанії. Компанії, які розташовані в дорогих регіонах, як правило, платять більше, ніж компанії, які розташовані в дешевих регіонах.

- Складність бізнесу. Компанії, які мають складний бізнес, як правило, платять більше, ніж компанії, які мають простий бізнес.

Загалом, вартість системи обліку даних про оренду житла може варіюватися залежно від конкретних потреб і вимог. Важливо порівняти тарифні плани різних систем, щоб знайти систему, яка відповідає бюджету і пропонує потрібні функції.

## **2.4 Формування вимог до розробки інформаційної системи обліку даних про оренду житла**

Формування вимог до розробки інформаційної системи обліку даних про оренду житла – це важливий етап, який визначає успіх проекту. Важливо враховувати всі потреби і вимоги користувачів системи, щоб вона була максимально ефективною. Система повинна забезпечувати виконання всіх необхідних функцій для управління процесом оренди житла. До таких функцій можна віднести [9]:

- 1) зберігання інформації про нерухомість, включаючи адресу, розмір, тип і ціну;
- 2) зберігання інформації про орендарів, включаючи особисту інформацію, контактні дані та історію платежів;
- 3) створення звітів про аналіз даних оренди, це можна реалізувати використовуючи інформаційну панель «дашборд»;
- 4) система повинна бути гнучкою і адаптивною, щоб відповідати змінним потребам користувачів;
- 5) система повинна забезпечувати високий рівень безпеки даних;
- 6) система повинна бути надійною і доступною, щоб не заважати роботі користувачів;

7) система повинна бути швидкою і ефективною, щоб обробляти великі обсяги даних.

Необхідно провести аналіз вимог бізнесу, визначити умови оренди, параметри фінансового обліку, а також необхідність взаємодії з іншими системами, якщо така є потреба [11].

Нефункціональні вимоги, такі як безпека, швидкодія, масштабованість та інші, визначаються для забезпечення ефективності та стабільності роботи системи.

Окремо виділяється важливість інтерфейсу користувача. Вимоги до інтерфейсу повинні враховувати потреби різних користувачів та забезпечувати інтуїтивність, легкість навігації та можливість взаємодії з системою.

Важливим етапом є також визначення технічних аспектів, таких як використання платформи, вибір мов програмування, технології розробки та інше.

У процесі формування вимог важливо враховувати побажання та відгуки від користувачів та зацікавлених сторін. Комунікація і взаємодія з клієнтом грають ключову роль у точному визначенні та зрозумінні його потреб [10].

Для успішного формування вимог до інформаційної системи обліку даних про оренду житла важливо враховувати такі рекомендації:

1. Варто залучити до процесу формування вимог користувачів системи. Користувачі краще за всіх знають свої потреби і вимоги, тому їх участь в процесі є обов'язковою.

2. Використовувати стандартизовані методи і інструменти. Стандартизовані методи і інструменти допоможуть сформулювати вимоги більш точно і ефективно.

3. Перевірити вимоги на реалістичність і здійсненність. Вимоги повинні бути такими, що їх можна реалізувати в рамках наявних ресурсів.

Завершальним етапом є розробка тестової документації, яка включає план тестування, тестові сценарії та критерії прийняття. Тестування грає важливу роль у перевірці відповідності системи вимогам та її коректному функціонуванню [12].

Цей процес є ітеративним і може змінюватися в залежності від нових відомостей, з'явлення нових вимог чи виправлень під час розробки системи.

## **2.5 Висновок до другого розділу**

У другому розділі проведено аналіз основних аспектів проектування інформаційних систем для обліку даних про оренду житла. Починаючи з загального розуміння процесу проектування та дослідження наявних методів розробки програмного забезпечення, розглянуто методології розробки, такі як Agile, Waterfall, Scrum, DevOps, зосереджуючись на їхніх особливостях та застосуванні до інформаційних систем обліку оренди житла.

Проведено дослідження наявних програм для обліку оренди житла, включаючи такі системи, як Clarify, Propertyware, AppFolio, Rentometer та Zillow Rental Manager. Наведено порівняння їх функціональності, інтерфейсу користувача та інших ключових характеристик.

Окремо акцентовано увагу на формування вимог до розробки інформаційної системи, враховуючи бізнес-потреби, функціональні вимоги, нефункціональні вимоги, вимоги до інтерфейсу користувача, інтеграцію та технічні аспекти.

Проаналізована інформація вказує на важливість збалансованого підходу до всіх аспектів розробки інформаційних систем для оренди житла, враховуючи потреби бізнесу, користувачів та технічні можливості. Комунікація з клієнтом та взаємодія з зацікавленими сторонами є критичними для успішного впровадження таких систем. Зрозуміння та врахування контексту проекту визначають ефективність інформаційних систем та їхню спроможність задовольняти потреби користувачів.

## **3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ДАНИХ ПРО ОРЕНДУ ЖИТЛА**

### **3.1 Актанти та варіанти використання інформаційної системи обліку даних про оренду житла**

В контексті інформаційної системи обліку даних про оренду житла, актанти та варіанти використання є ключовими поняттями, які визначають ролі та можливості учасників системи. У даному підрозділі буде розглянуто варіанти використання та учасники системи [13].

Актанти – це суб'єкти або об'єкти, які беруть участь в конкретному сценарії або взаємодіють з інформаційною системою. У системі обліку даних про оренди житла можуть виступати різні актанти.

Основним користувачем системи є ріелтор. Ріелтори – професіонали, які допомагають укладати орендні угоди між орендарами та власниками. Вони можуть використовувати систему для пошуку об'єктів нерухомості та введення в облік нових об'єктів нерухомості [29].

Роль ріелтора створити базу з нерухомості, укладати орендні угоди, ввести статистичний облік. Користувач "ріелтор" у системі обліку даних про оренду житла має ряд важливих функцій, пов'язаних з управлінням об'єктами нерухомості та процесом оренди. Опишемо детальніше можливі функції ріелтора:

1. Пошук Об'єктів Нерухомості: Ріелтор може використовувати систему для пошуку доступного житла відповідно до критеріїв клієнта. Є варіанти використання фільтрів (ціна, розташування, тип нерухомості), перегляд інформації про доступні об'єкти.

2. Управління Об'єктами Нерухомості: Ріелтор може додавати нові об'єкти нерухомості до системи та оновлювати інформацію про існуючі. Здійснює заповнення та збереження даних про нерухомість, встановлення умов оренди, завантаження фотографій та інші характеристики.

3. Ведення Переговорів та Угод: Ріелтор може вести переговори між орендаторами та власниками нерухомості та укладати орендні угоди. В інформаційній сторінці обраного об'єкту вказано методи комунікації з орендаторами та власниками, встановлення умов угод, вирішення питань, пов'язаних із орендою.

4. Моніторинг статистичних даних: Ріелтор може відстежувати статус заявок на оренду та інформацію про активні угоди, динаміку цін на оренду та коефіцієнт вигоди з контрактів.

5. Надання Консультацій: Ріелтор може за допомогою системи проконсультувати стосовно ринкових умов, процедур оренди та характеристик об'єктів нерухомості. Використовуючи наявну базу житла, ріелтор може дати відповіді на питання клієнтів, надання рекомендацій, пояснення юридичних аспектів.

Основні варіанти використання актора “Ріелтор” продемонстровано на рисунку 3.1 [30].

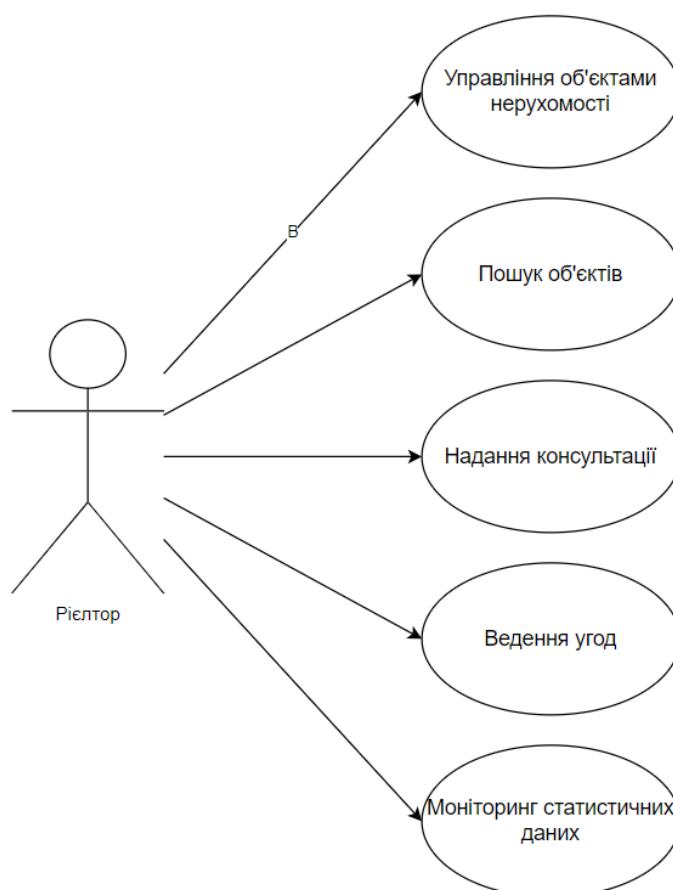


Рисунок 3.1 – Варіанти використання системи актором “Ріелтор”

Ці функції дозволяють ріелтору ефективно взаємодіяти з системою, управляти нерухомістю, надавати послуги клієнтам та утримувати повний контроль над орендними процесами. Ріелтор стає ключовим учасником системи, сприяючи оптимізації та покращенню процесів оренди житла.

## **3.2 Розробка архітектури інформаційної системи**

У даному підрозділі «Розробка архітектури» описано процес проектування інформаційної системи обліку даних про оренду житла. В підрозділі вказано ключові аспекти архітектури, компоненти та виконання [32].

### **3.2.1 Використання фреймворка React Refine**

React Refine – це фреймворк для створення веб-додатків, заснований на React. Він надає широкий спектр функцій, які допомагають розробникам створювати якісні та масштабовані додатки [31].

Серед основних функцій React Refine варто відмітити:

- Шаблонна архітектура – це один із основних принципів React Refine. Вона дозволяє централізовано зберігати код UI, що полегшує його підтримку та розширення.
- Генератор коду – React Refine включає в себе генератор коду, який допомагає розробникам швидко створювати шаблони та компоненти.
- Система стилів – включає в себе систему стилів, яка дозволяє розробникам легко створювати стилі для своїх додатків.
- Менеджер стану – React Refine включає в себе менеджер стану, який дозволяє розробникам зберігати стан своїх додатків.
- Система тестування – включає в себе систему тестування, яка допомагає розробникам тестувати та виявляти помилки у своїх додатках.

React Refine пропонує розробникам повний інструментарій для створення веб-додатків на базі React, зосереджуючись на шаблонній архітектурі, швидкості розробки, управлінні стилів та системою тестування.

### 3.2.2 Шаблони архітектури

Шаблонна архітектура є важливим аспектом при розробці веб-застосунків, оскільки це дозволяє структурувати код, полегшує утримання та розширення застосунку. Основна ідея полягає в тому, щоб розділити програму на певні компоненти або шари, кожен з яких відповідає за конкретну функціональність. Такий підхід полегшує розуміння, тестування та зміну коду.

Шаблонна архітектура React Refine заснована на ідеї використання шаблонів для повторюваних елементів UI. Шаблони дозволяють централізовано зберігати код UI, що полегшує його підтримку та розширення [36].

При використанні шаблонної архітектури проект React Refine зазвичай складається з наступних шарів:

- Шаблони – це компоненти React, які описують вигляд UI. Шаблони не містять жодного бізнес-логіки, а лише визначення HTML-структури та стилів.
- Контролери – відповідають за взаємодію з користувачем. Контролери отримують дані від користувача та передають їх шаблонам для відображення.
- Моделі – це об'єкти, які зберігають дані про стан UI. Моделі можуть бути як простими, так і складними.

Перейдемо до опису інтеграції з сервісом авторизації Google.

### 3.2.3 Інтеграція з сервісом авторизації Google

Для зручного використання інформаційної системи обліку даних про оренду житла з сторони користувача, було вирішено використовувати Google авторизацію. Також це дозволить детально контролювати режим доступу до системи обліку даних. При генерації проекту за допомогою Refine, передбачено попереднє встановлення плагіну авторизації Google. Для отримання ключів API (Application Programming Interface), необхідно створити проект у Google Cloud Console [47]. Панель керування доступу до згенерованого API показано на рисунку 3.2.



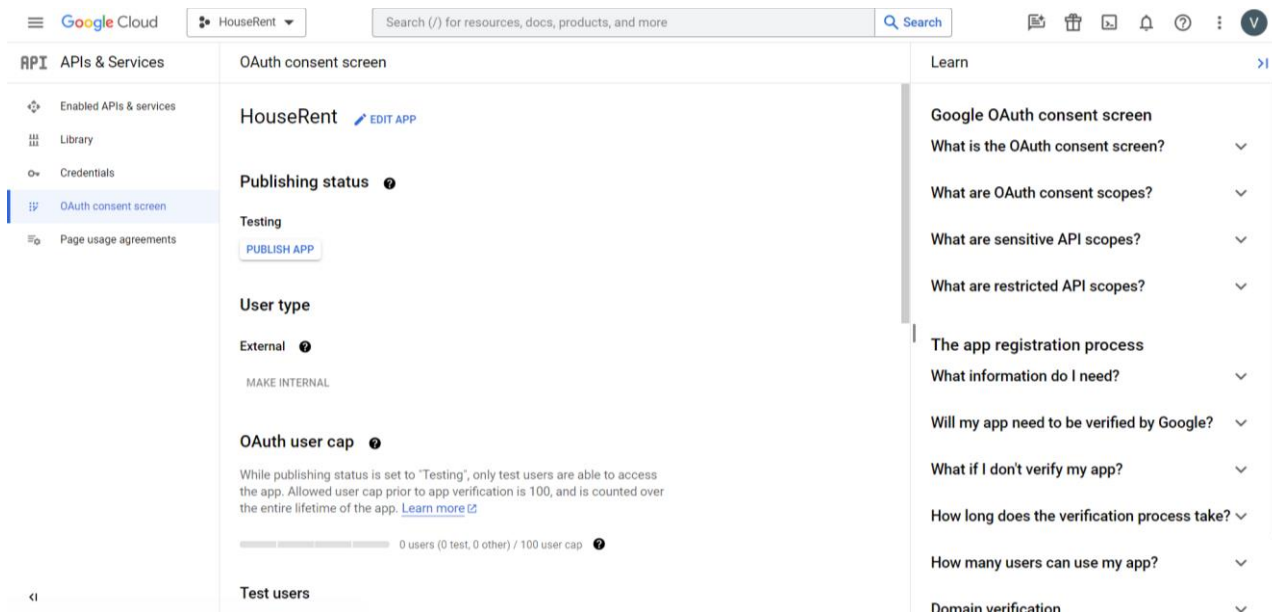


Рисунок 3.2 – Процес генерації ключа API

Ключ API потрібно інтегрувати у корінь структури коду для безперешкодного виконання функцій.

### 3.2.4 Використання системи управління базами даних MongoDB

MongoDB дозволяє успішно виконувати запити json у проєкті для зберігання даних у вигляді JSON-подібних документів. Ця база даних має численні переваги, які роблять її популярним вибором для багатьох проєктів [37].

MongoDB використовує гнучку схему, що означає, що кожен документ у колекції може мати свою власну структуру. Це особливо корисно для даного проєкту, де структура даних може змінюватися з часом без необхідності міграцій бази даних [14].

Швидкий доступ до читання та запису даних. Система використовує вбудований механізм індексації та може легко масштабуватися горизонтально для обробки великої кількості запитів. MongoDB може зберігати різні типи даних, такі як тексти, числа, дати, масиви та документи, що дозволяє зберігати дані у вигляді, які найкраще відповідає потребам обліку даних про оренду житла.

Легке масштабування – це також ключовий момент, через який було обрано дану систему бази даних. Це дозволяє підтримувати проекти з великою кількістю даних та великою кількістю одночасних запитів.

MongoDB підтримує автоматичну реплікацію, що забезпечує збереження кількох копій даних для забезпечення високої доступності. Вона також має механізми відмовостійкості, що гарантують збереження даних навіть при виникненні непередбачуваних ситуацій [38].

Для успішної інтеграції MongoDB у проект, варто використовувати документацію. MongoDB має велику та активну спільноту розробників, а також добре документована офіційна документація, що спрощує вивчення та вирішення проблем. Процес створення БД (база даних) для проекту показано на рисунку 3.3.

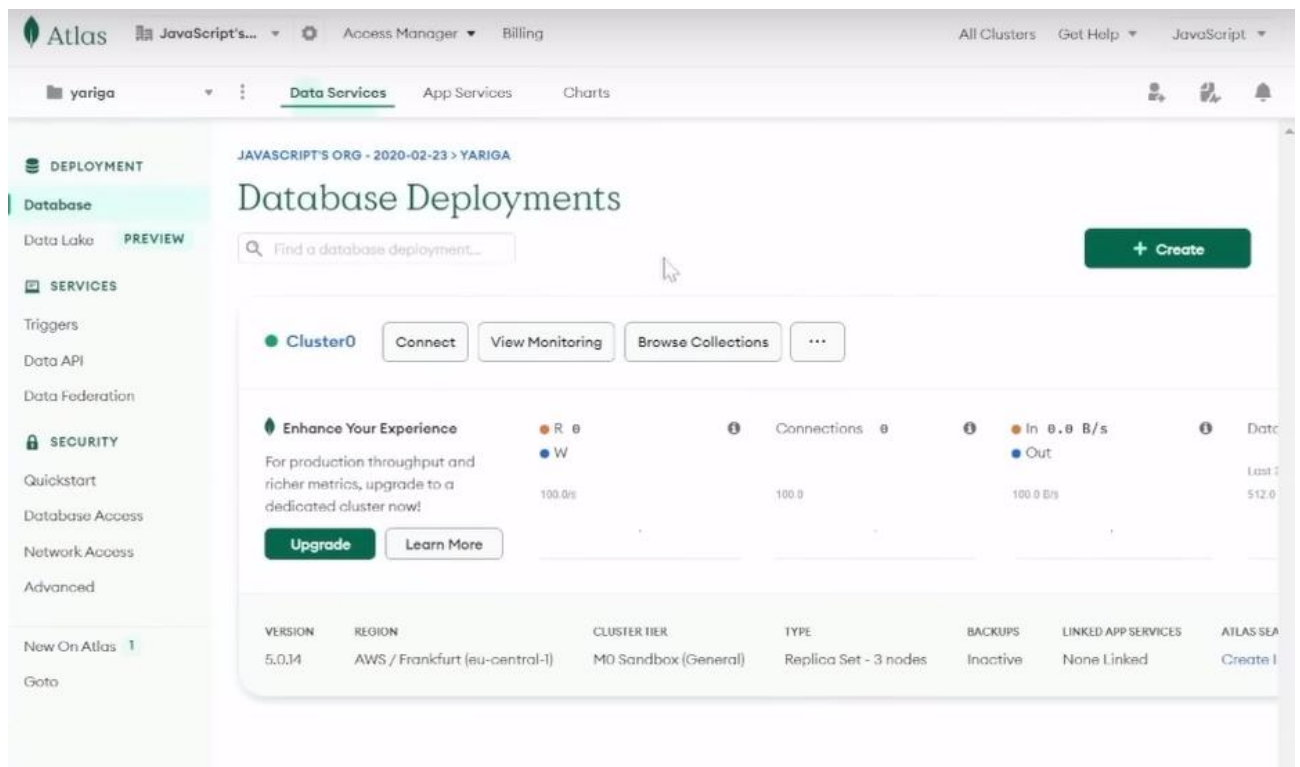


Рисунок 3.3 – Створена БД для обліку даних про оренду житла

MongoDB є потужним рішенням для розробки інформаційної системи обліку даних про оренду житла. Вона надає інструменти для зручної роботи з даними та вирішення багатьох завдань у сфері розробки програмного забезпечення.

### 3.3 Обґрунтування вибору засобів розробки

У даному підрозділі буде детально розглянуто засоби які було обрано для реалізації інформаційної системи обліку даних та обґрунтування вибору. Вибір конкретних засобів для створення інформаційної системи обліку даних житла базується на конкретних потребах проекту [32, 40]. У даному випадку, такий стек технологій може бути доцільним:

- Node.js: Ефективне серверне рішення для створення масштабованих веб-додатків. Легка інтеграція з MongoDB для обробки запитів та зберігання даних. Використання JavaScript як мови програмування на обох сторонах (клієнт та сервер) для спрощення розробки та обслуговування коду.
- React.js: Потужний фреймворк для створення інтерфейсу користувача (UI).
- Віртуальний DOM та компонентна архітектура для високої продуктивності та швидкої реакції на зміни в додатку. Ідеально поєднується з Node.js для створення односторінкових додатків (SPA) та забезпечення зручного користувацького інтерфейсу.
- MongoDB: Гнучка та масштабована база даних, що забезпечує зручне зберігання та управління даними у форматі JSON-подібних документів. Просте масштабування та висока продуктивність для збереження великих обсягів даних.
- TypeScript: Надмножина JavaScript, що дозволяє визначати типи даних для полегшення розробки та підвищення надійності коду. Корисно для розробки інформаційних систем зі складною структурою даних.
- Material UI: Бібліотека готових компонентів у стилі Material Design для зручного та привабливого користувацького інтерфейсу. Легка у використанні та адаптивна для ефективної розробки UI-елементів [34].
- Google Auth: Інтеграція забезпечить швидку та зручну автентифікацію через облікові записи Google. Забезпечить безпеку та впевненість користувачів у системі [47].

Застосування даних технологій може забезпечити високоефективний та зручний для користувачів інструмент для обліку даних житла. Даний вибір також

впливає на сам процес реалізації веб-застосунку, враховуючи специфіку проекту та його вимоги.

### **3.4 Створення інформаційної панелі (Dashboard)**

Інформаційна панель (Dashboard) є ключовим інструментом у реалізації системи обліку даних про оренду житла. Дана система надає користувачу графічний огляд ключових показників та даних, необхідних для ефективного відстеження, аналізу та прийняття рішень у реальному часі. Використання системи Dashboard в обліку даних житла дозволить реалізувати додаткові інструменти для формування бази оренди житла [43].

#### **3.4.1 Моніторинг процесів оренди житла**

Моніторинг оренди житла через систему Dashboard є ключовим інструментом для користувача у сучасному бізнес-середовищі. Використання інформаційної панелі дозволяє візуалізувати та аналізувати ключові показники ефективності та інші дані, пов'язані з бізнес-процесами оренди житла [42]. Це включає в себе відображення даних у вигляді графіків, діаграм, числових значень та інших візуальних елементів. Використання системи в режимі реального часу дозволяє миттєво отримувати оновлену інформацію про поточний стан нерухомості, що сприяє швидкому реагуванню на зміни та виявленню проблем. Графічні представлення на Dashboard дозволяють виводити тренди та динаміку змін в часі, а аналітичні засоби допомагають розуміти причини змін. Цей інструмент також дозволяє визначити ефективність різних аспектів діяльності та виявляти можливості для оптимізації процесів. Інформаційну сторінку Dashboard продемонстровано на рисунку 3.4.

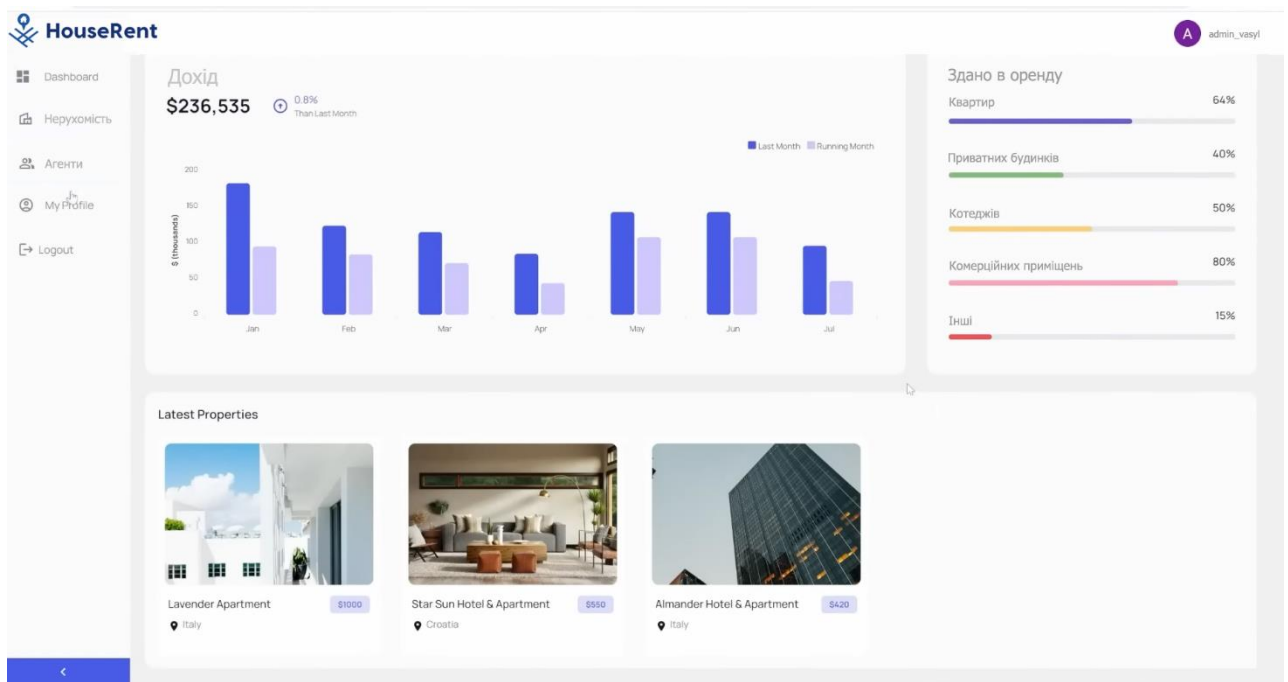


Рисунок 3.4 – Сторінка з Dashboard

Враховуючи показники результативності та моніторинг у реальному часі, Dashboard стає необхідним інструментом для прийняття обґрунтованих рішень щодо маніпуляцій з нерухомістю.

### 3.4.2 Функціональна сторінка агента з нерухомості

Функціональна сторінка ріелтора в інформаційній панелі (Dashboard) є ключовим інструментом для ефективного управління та моніторингу нерухомості. Ріелтор має зручний доступ до різних функцій, що допомагають оптимізувати його робочий процес та забезпечують повний контроль над даними та діяльністю [41].

На цій сторінці (рисунок 3.5) ріелтор може переглядати та редагувати інформацію про об'єкти нерухомості, легко додавати нові оголошення та актуалізувати існуючі. Зручна візуалізація статусу кожного оголошення (вільно/зайнято, активно/неактивно) дозволяє швидко спрямовувати зусилля на потрібні завдання.

Крім того, ріелтор може взаємодіяти з іншими членами команди, обмінюючись інформацією та спільно працюючи над проектами. Повідомлення

та нагадування допомагають оперативно реагувати на важливі події та завдання [44].

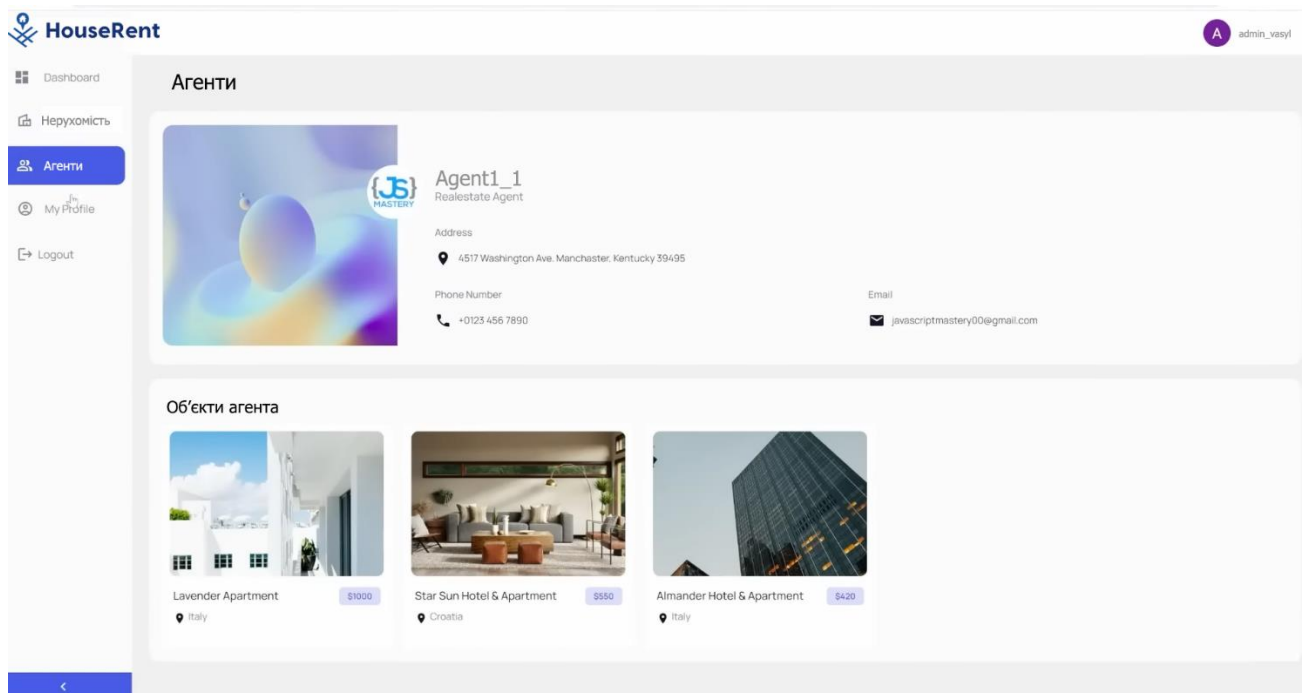


Рисунок 3.5 – Інформаційна сторінка ріелтора

Загальний обсяг функціональності сторінки ріелтора в інформаційній панелі сприяє підвищенню продуктивності, зручності роботи та успішному веденню бізнесу в сфері нерухомості.

### 3.4.3 Функціональна сторінка з доступним житлом

Функціональна сторінка з житлом в інформаційній панелі є важливим ресурсом для ріелторів та агентів нерухомості, надаючи їм повний огляд та контроль над усіма аспектами їхнього портфеля. Завдяки цій сторінці, фахівці можуть ефективно керувати та аналізувати інформацію про кожен об'єкт нерухомості [45].

Здійснюючи перегляд об'єктів нерухомості, ріелтор може отримати доступ до повної інформації, такої як розташування, тип власності, фінансові показники та статус доступності. Можливість застосовувати фільтри та сортувати об'єкти полегшує навігацію та швидке знаходження необхідних даних.

Статус кожного об'єкта, його фінансові аспекти та останні зміни відображаються в зручному форматі. Аналітичні інструменти надають можливість вивчати тенденції ринку та аналізувати динаміку змін у ціновому та попитовому аспектах.

Інтерактивна карта дозволяє візуалізувати розташування об'єктів на мапі, що допомагає врахувати географічний контекст. Взаємодія з кожним об'єктом надає ріелтору повний контроль над його характеристиками та станом [45].

Стратегічне ведення звітності та взаємодія з клієнтами через цю сторінку сприяє покращенню обслуговування та адаптації до змін на ринку. Вигляд сторінки з житлом продемонстровано на рисунку 3.6.

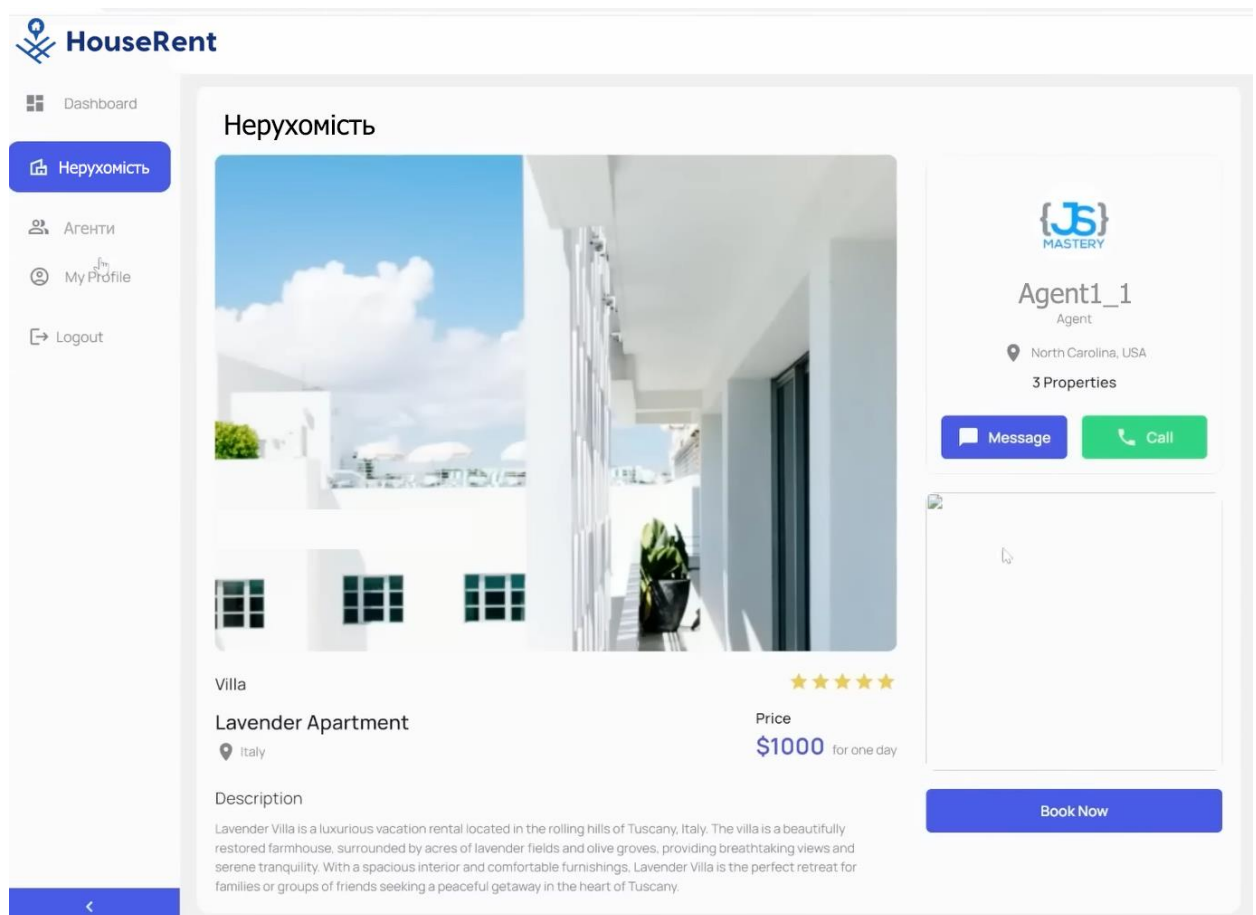


Рисунок 3.6 – Інформаційна сторінка з інформацією про житло

Усе це робить функціональну сторінку "Нерухомість" не тільки потужним інструментом для управління даними, але й стратегічним партнером для успішної роботи в сфері нерухомості.

### 3.4.4 Функція додавання нового житла у систему

Функціональна сторінка "Додавання житла" у складі інформаційної системи обліку оренди нерухомості є необхідним інструментом для ефективного управління нерухомістю та роботи ріелторів. Ця функція забезпечує можливість оперативно та систематично вносити актуальні дані про нові об'єкти житла у централізовану базу даних, що є ключовим компонентом процесу управління житловою нерухомістю [46].

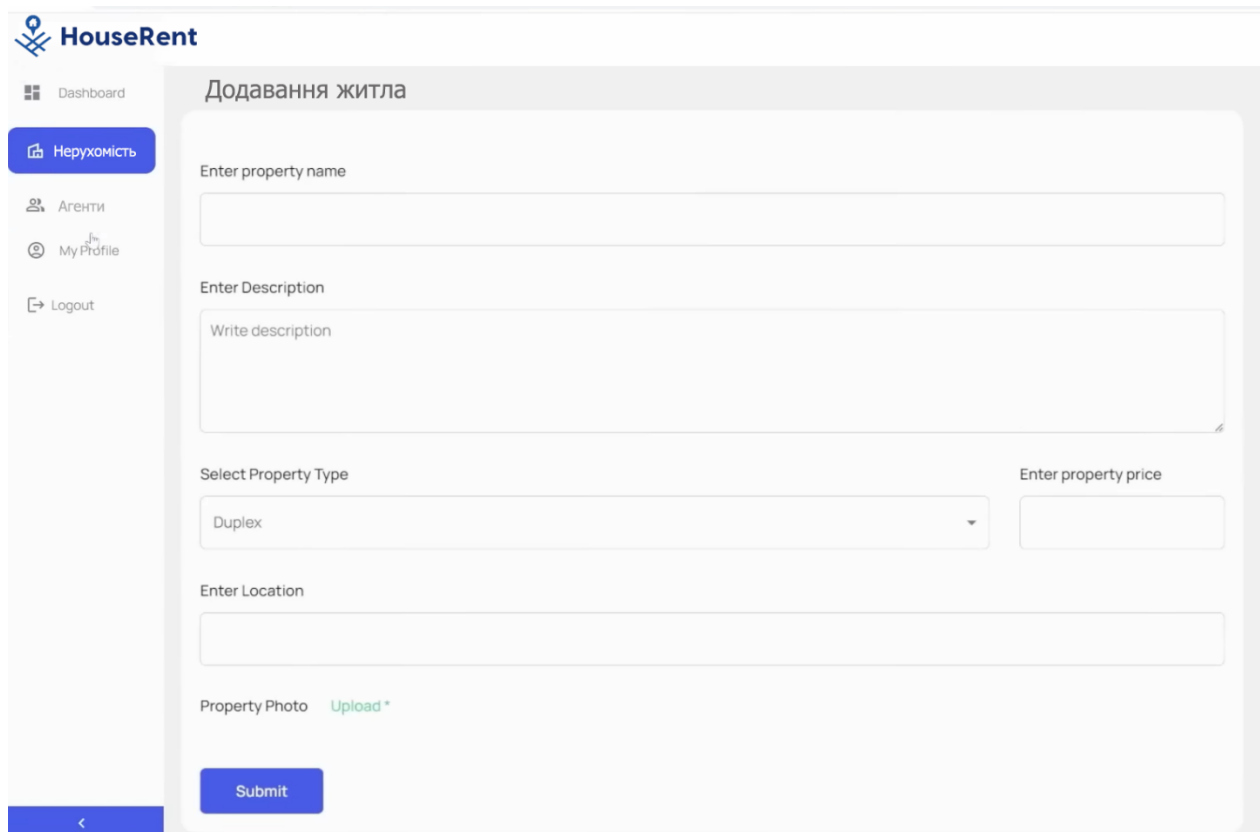
Сторінка "Додавання житла" детально розроблена, включаючи широкий спектр полів для заповнення, які охоплюють різноманітні аспекти властивостей, такі як тип орендованого житла, кількість кімнат, фінансові параметри, розташування, опис, фото та інші характеристики. Ріелтор, використовуючи цю функцію, здатен надати повну та докладну інформацію про об'єкт нерухомості, що сприяє ефективній комунікації з потенційними клієнтами та забезпечує якісний облік наявних житлових можливостей.

Наголошено на важливості додавання фотографій, що не лише візуалізує нерухомість, але й підвищує її привабливість для клієнтів. Це стає ключовим елементом вирішення об'єкта серед конкурентів і сприяє швидшому та ефективнішому укладенню угод.

Однак, після внесення даних, процес збереження у базі даних MongoDB виступає як критичний крок. Це не лише забезпечує миттєвий доступ до актуальної інформації, але також відкриває можливість для подальшого аналізу ринку, ураховуючи останні зміни та тенденції в сфері нерухомості. Підкреслюється важливість безпеки під час взаємодії з базою даних, щоб уникнути дублювання та зберегти унікальність інформації [48].

Створення ефективної та зручної системи для додавання нового житла в інформаційній панелі визначає успіх реальних агентів нерухомості. Це допомагає підтримувати актуальність бази даних, сприяє ефективній взаємодії з клієнтами та забезпечує успішний облік даних у динамічному середовищі ринку нерухомості.





The screenshot displays the 'HouseRent' web application interface. On the left, a sidebar menu contains 'Dashboard', 'Нерухомість' (highlighted in blue), 'Агенти', 'My Profile', and 'Logout'. The main content area is titled 'Додавання житла' (Add Property). It features several input fields: 'Enter property name', 'Enter Description' (with a text area), 'Select Property Type' (a dropdown menu showing 'Duplex'), 'Enter property price', and 'Enter Location'. Below these is a 'Property Photo Upload' section with an 'Upload\*' button. A blue 'Submit' button is located at the bottom of the form.

Рисунок 3.7 – Сторінка додавання нового житла

Перейдемо до опису методів узагальнення даних.

### 3.5 Методи узагальнення даних

Реалізація методу узагальнення даних про оренду житла з бази даних MongoDB та їх виведення в інформаційній панелі, використовуючи технології React, Node.js, відкриває широкі можливості для створення інтерактивного веб-застосунку по обліку даних про оренду житла.

На початковому етапі, серверна частина, побудована на Node.js, взаємодіє з базою даних MongoDB. Це забезпечує ефективний обмін даними та можливість витягування інформації за допомогою створених API. Одночасно, Node.js дозволяє реалізувати асинхронні операції та обробку багатьох запитів одночасно, підвищуючи продуктивність серверної частини [49].

Підключення до MongoDB виконується за допомогою шаблонної інтеграції (через фреймворк Refine), що гарантує стабільність та надійність взаємодії з базою даних. Дані, які надходять з MongoDB, обробляються на

сервері, що дозволяє виконати необхідні маніпуляції, такі як фільтрація, сортування чи агрегація.

На клієнтській стороні, React використовується для розробки інтерфейсу та відображення отриманих даних на інформаційній панелі. Компоненти React, які можуть бути динамічними при зміні стану, гарантують плавну та ефективну візуалізацію даних для користувачів. Використання Axios та інших засобів для взаємодії з API на сервері дозволяє отримувати актуальні дані та підтримувати їх оновлення в режимі реального часу. Це важливо для забезпечення користувачам актуальної та точної інформації [50].

Загальний підхід, який комбінує React, Node.js та MongoDB, створює потужні та гнучкі інструменти для розробки інформаційних панелей, що задовільнить вимоги до реалізації системи обліку даних про оренду житла та забезпечать ефективну взаємодію з даними.

### 3.6 Функціонування та інтерфейс користувача інформаційної системи

У даному підрозділі описано варіації використання інформаційної системи. Перше з чим зустрінеться користувач – сторінка авторизації. Це продемонстровано на рисунку 3.8.

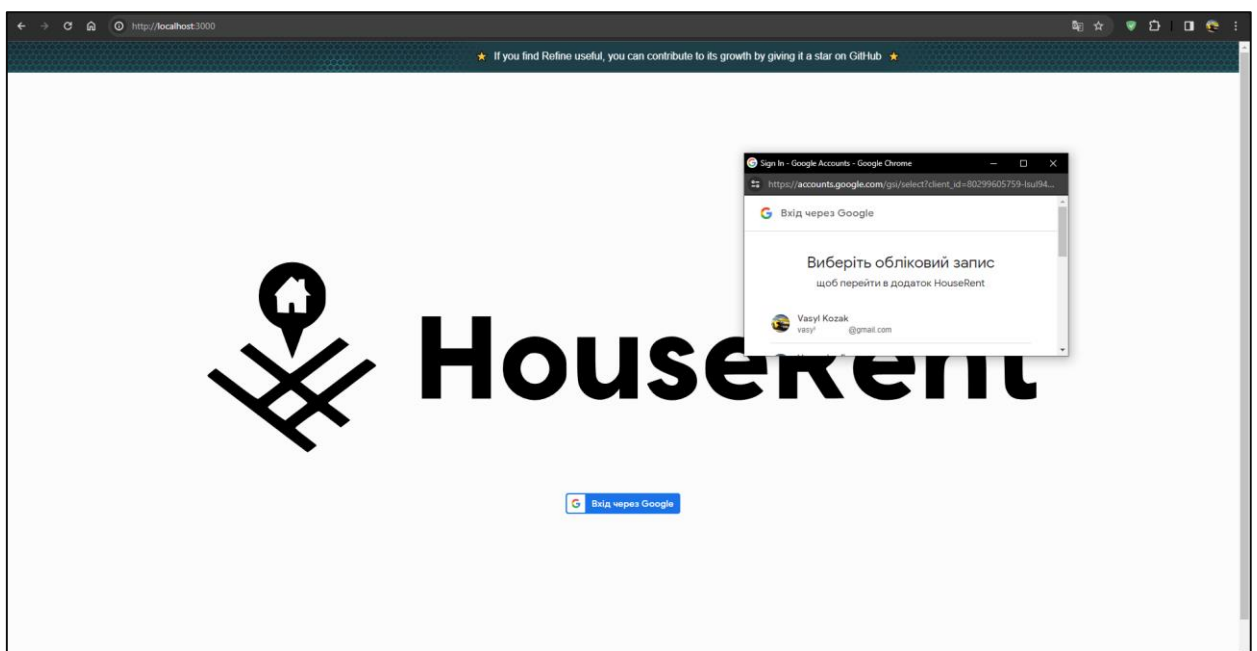


Рисунок 3.8 – Сторінка авторизації в інформаційну панель

На сторінці авторизації використовується інтеграція з Google Auth, яка надає додатковий рівень зручності та безпеки, забезпечуючи надійний механізм автентифікації користувачів.

Після успішної авторизації, користувач опиниться на головній сторінці Dashboard і отримає доступ до функціоналу системи. На даній сторінці система проводить звітність за критеріями: дохід від зданих об'єктів, відсоток кожного типу нерухомості, який вже зданий в оренду. Дану інформацію можна побачити на рисунку 3.9. На даній сторінці, користувач може брати яку наступну дію виконає: переглянути список нерухомості який є у системі, переглянути список ріелторів і об'єктів які за ними закріплені. Система пропонує ріелторам інтуїтивно зрозумілий інтерфейс для швидкого та ефективного пошуку житла на сторінці «Нерухомість». Використовуючи різні параметри, такі як ціна, розташування, тип нерухомості та інші, користувачі можуть знайти потрібну інформацію для нерухомості.

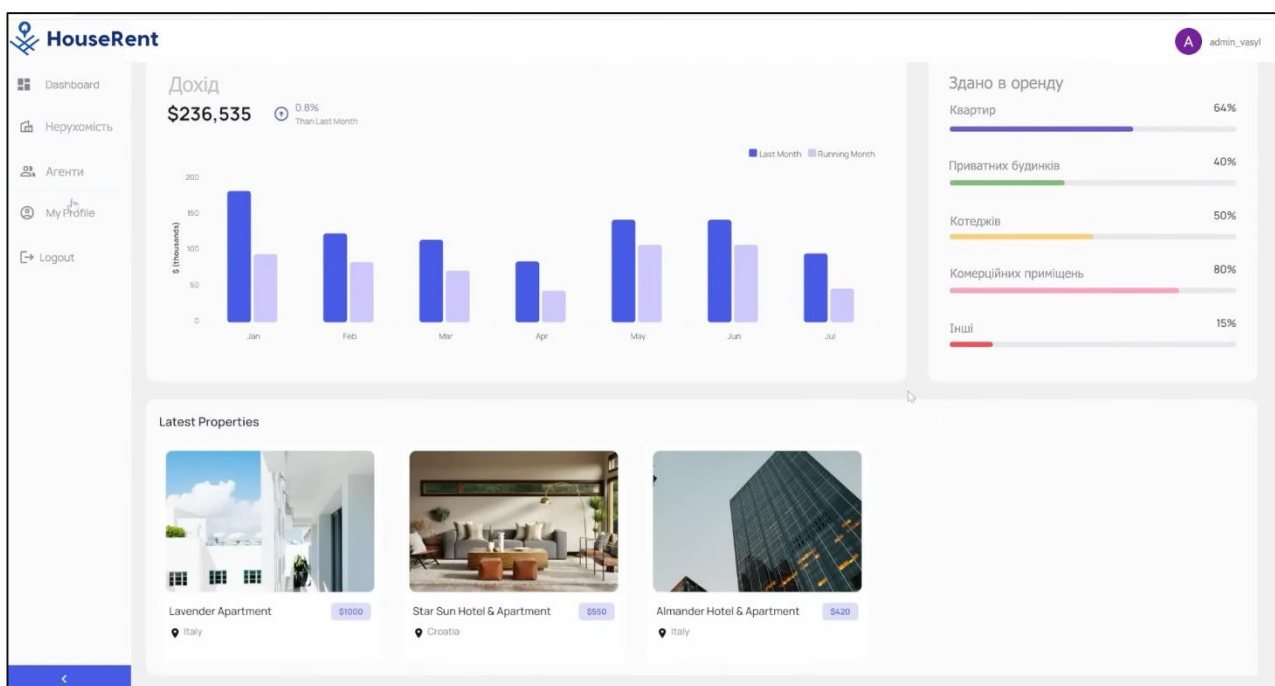


Рисунок 3.9 – Інтерфейс користувача інформаційної панелі

Функціонал системи дозволяє ріелторам керувати об'єктами нерухомості, додавати нові об'єкти та оновлювати інформацію про них. Ріелтори можуть вести моніторинг за успішністю своєї роботи за кількість зданих об'єктів.

Інформаційна система обліку даних про оренду житла надає повний спектр функцій для керування нерухомістю та орендними процесами, спрощуючи взаємодію між різними сторонами та забезпечуючи оптимальний досвід використання.

### **3.7 Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи описано технології та їх взаємодії в контексті інформаційної системи обліку даних про оренду житла. Використання Node.js як серверної платформи дозволяє створювати високопродуктивні та швидкодіючі застосунки, забезпечуючи гнучкість в роботі з асинхронними операціями та обробкою багатьох одночасних запитів.

Використання фреймворку Refine дозволяє пришвидшити процес розгортання проекту та зберегти цілісність структури системи. MongoDB виправдовує свою популярність та універсальність, пропонуючи адаптовану для потреб обліку житла модель зберігання даних. Її гнучкість та можливість масштабування стають важливими в архітектурі, де потрібно зберігати та швидко отримувати доступ до великого обсягу різнорідних даних.

React, у свою чергу, дозволяє створювати інтерактивні та динамічні інтерфейси для користувачів. Його компонентний підхід полегшує розробку та підтримку коду, що важливо для швидкого реагування на зміни вимог та удосконалення користувацького інтерфейсу.

Інформаційна панель (Dashboard) стає центральною точкою взаємодії користувачів з системою, де вони можуть здійснювати моніторинг, аналізувати дані та приймати управлінські рішення. Функціонал моніторингу даних житла для оренди додає значущої вартості, допомагаючи збирати та аналізувати дані для оптимізації стратегій та підвищення ефективності діяльності.

Інформаційна система дозволяє не лише збирати та обробляти дані, але і створює зручні та ефективні інструменти для їхнього відображення, аналізу та взаємодії.

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Розробка раціональної діяльності та створення сприятливих і безпечних умов персоналу

Тема кваліфікаційної роботи освітнього рівня «Магістр» присвячена розробці інформаційної системи обліку даних житла. Тому доцільно розглянути питання розробки середовища для ефективної та сприятливої роботи працівників з обліком даних оренди житла.

Чим більше людей залучено до спільної діяльності в організації, тим складнішим і відповідальнішим є процес організації та регулювання цієї діяльності. Наприклад, у офісі де створюються веб-застосунки, є край важливо створити сприятливі умови для трудового колективу. Навіть у малочисельній організації спільна діяльність персоналу є різноманітною і складною: відбувається у просторі й часі, супроводжується використанням різноманітних ресурсів, потребує великих витрат управлінської праці на нормування, планування, мотивацію, координацію, облік і контроль тощо [20].

Організація праці – приведення трудової діяльності людей в певну систему, що характеризується внутрішньою впорядкованістю, узгодженістю і спрямованістю взаємодії для реалізації спільної програми та цілі. Організація праці має як статичний аспект, тобто структуру праці, так і динамічний, тобто функціонування, поведінку, взаємодію людей в процесі трудової діяльності [23].

Організація праці на даному рівні передбачає:

- 1) поділ і кооперацію праці;
- 2) розробку методів праці, за допомогою яких може бути виконаний той чи інший вид роботи;
- 3) організацію робочих місць;
- 4) створення умов праці;
- 5) підбір і розстановку кадрів;
- 6) визначення переліку функцій та обов'язків кожного працівника;
- 7) встановлення міри праці за допомогою нормування;

- 8) організацію оплати праці;
- 9) встановлення правил дисципліни праці, які забезпечують порядок і узгодженість в роботі, режим праці і відпочинку працівників.

Регулювання трудової діяльності персоналу – це системно організований процес забезпечення і підтримання високоефективної праці кожного окремого працівника і трудового колективу в цілому; це найважливіше і постійне завдання менеджменту персоналу в організації. Регулювання трудової діяльності персоналу за метою, змістом і значенням має багато спільного з організацією праці, тому їх нелегко розвести, відділити одне від одного, та в цьому немає й потреби. Головне – чітко визначити, які лінійні керівники, функціональні підрозділи, допоміжні виробництва за що відповідають [22].

Основними завданнями регулювання трудової діяльності персоналу є:

- 1) забезпечення продуктивної праці всіх категорій персоналу;
- 2) підтримання якості продукції та послуг на конкурентоспроможному рівні;
- 3) створення безпечних умов праці;
- 4) підтримання у трудовому колективі сприятливого соціально-психологічного клімату.

Ефективність трудової діяльності персоналу залежить і формується під впливом багатьох різнопланових чинників. Усі їх з певною мірою умовності можна поєднати у дві групи. В першій групі варто виділити заходи, у процесі виконання яких створюються необхідні умови для діяльності персоналу, без чого взагалі неможлива нормальна робота структурного підрозділу, а про якийсь рівень ефективності й мови немає [24]. До першої групи чинників доцільно зарахувати наступні заходи:

- 1) визначення мети діяльності;
- 2) планування діяльності;
- 3) кадрове забезпечення;
- 4) нормативно-правове забезпечення;
- 5) організаційно-економічне забезпечення;
- 6) інженерно-технічне забезпечення.

Наступні чинники спрямовані на підтримання установленого ритму виробництва, виконання календарних графіків поставок замовникам, забезпечення руху транспортних засобів за розкладом. До другої групи належать такі заходи:

- 1) ресурсне забезпечення;
- 2) поточне обслуговування виробництва, трудових, транспортних та інших процесів;
- 3) оперативне регулювання виробничих і трудових процесів;
- 4) облік, контроль та оцінювання результатів діяльності.

Фахівці з управління персоналом виявили: одним з основних джерел незадоволеності роботою в організації – неефективна організація праці. Наприклад, не розмежовано зони відповідальності, співробітники виконують непотрібні дії, порушують технологічну послідовність операцій тощо. У даному випадку причина незадоволеності працівників – управлінські помилки, а саме: неправильно спроектовані робочі процеси, які можуть негативно впливати на продуктивність праці, створювати в підрозділі конфліктні ситуації, приводити до звільнення ключових співробітників, погіршувати задоволеність клієнтів [25].

Для регулювання трудової діяльності в організації розробляють та використовують такі документи:

- колективний договір організації;
- правила внутрішнього трудового розпорядку;
- положення про структурні підрозділи;
- посадові інструкції;
- правила експлуатації технологічного та іншого устаткування;
- норми і правила з охорони праці та техніки безпеки;
- штатний розпис організації в цілому та її структурних підрозділів (формування малих і великих трудових колективів: ланок, бригад, виробничих дільниць, цехів, відділів, секторів, бюро, лабораторій);
- графіки змінності;
- норми і ліміти витрачання всіх видів ресурсів (праці, зарплати, основних і допоміжних матеріалів, інструментів, енергетичних ресурсів тощо);

- календарні графіки виробництва продукції, відвантаження її замовникам;
- графіки і регламенти обслуговування робочих місць, потокових і конвеєрних ліній, складних технічних систем;
- графіки планово-попереджувального ремонту устаткування;
- положення щодо систем матеріального і морального заохочення працівників.

Звичайно, на ефективність діяльності організації справляють свій вплив різні чинники – внутрішні і зовнішні, об'єктивні й суб'єктивні. Однак усе залежить від людського чинника, від рівня конкурентоспроможності персоналу в цілому, від компетентності вищого менеджменту, його вміння долати вплив негативних зовнішніх чинників та розумно використовувати наявні ресурси, у першу чергу людські. Адже, як відомо, саме кадри все вирішують. Не дарма японські менеджери найбільше уваги приділяють питанням розвитку людських ресурсів.

#### **4.2 Вплив електромагнітного випромінювання комп'ютера на здоров'я користувача**

В сучасному світі, де використання персональних комп'ютерів стало необхідністю, дедалі більше уваги приділяється питанням безпеки праці, зокрема впливу електромагнітних випромінювань на функціональний стан організму користувачів. Технологічний прогрес неперервно поширює межі користування електронікою, із чим пов'язані і нові виклики для охорони праці [23].

Електромагнітні випромінювання (ЕМВ) виникають під час роботи персональних комп'ютерів через електричний струм та взаємодію електронів у пристроях. Ці випромінювання можуть бути електричними, магнітними та електромагнітними за своєю природою. Основними джерелами ЕМВ є монітори, системні блоки, клавіатури та миші [24].

Дослідження свідчать про те, що тривалий час роботи за комп'ютером може викликати ряд проблем, пов'язаних із здоров'ям. До них відносяться



головні болі, проблеми із зором, погіршення концентрації уваги та інші симптоми, які можуть впливати на ефективність праці та загальний комфорт користувачів.

При облаштуванні робочого місця користувачів персональних комп'ютерів важливо враховувати ряд аспектів для забезпечення безпеки та здоров'я працівників. Одним із ключових аспектів є ергономіка. Забезпечення ергономічних меблів та обладнання, яке враховує індивідуальні особливості користувача, може запобігти численним проблемам з м'язами, хребтом і зором.

Освітлення робочого місця впливає на зоровий комфорт та загальний стан працівника. Достатнє та природне освітлення є важливим елементом, а штучне освітлення повинно бути розташоване так, щоб уникати блисків на екрані монітора.

Використання антиблікових екранів та фільтрів може знизити вплив відбиття світла на моніторах, що допомагає уникнути втоми та дискомфорту для очей. Крім того, регулярні перерви під час роботи дозволяють відпочивати м'язам та очам, попереджуючи втому та подразнення [24].

Спеціальні окуляри, розроблені для роботи за комп'ютером, можуть надавати додатковий захист від впливу екранів та поліпшувати комфорт для очей користувачів. Організація робочого часу та правильний розподіл завдань також грають важливу роль у збереженні фізичного та психічного здоров'я.

Не менш важливим є питання чистоти та порядку на робочому місці. Правильне розташування обладнання та усунення зайвих предметів може поліпшити оглядовість та зменшити стрес для користувача.

Загальна мета цих заходів – забезпечити комфортні умови праці та запобігти можливим негативним впливам, зокрема від електромагнітних випромінювань. Навчання та інформування працівників щодо правил безпеки та заходів з охорони праці є важливою частиною процесу, оскільки вони повинні розуміти, як самостійно дбати про своє здоров'я при роботі з комп'ютерами [25].

Для мінімізації впливу ЕМВ на здоров'я працівників важливо вживати ефективні заходи з охорони праці. До них входить використання екранів, що

зменшують випромінювання, регулярні перерви під час роботи, встановлення оптимального освітлення та відпочинок очей.

З удосконаленням технологій виробництва та конструкції пристроїв для зменшення випромінювань, можна очікувати поліпшення у сфері охорони праці. З'явлення нових матеріалів та дизайнів може сприяти зниженню ризиків для користувачів персональних комп'ютерів.

Електромагнітні випромінювання персональних комп'ютерів – це важливий аспект, який потребує уваги охоронців праці. Збалансований підхід до організації робочого місця та дотримання заходів з охорони праці можуть допомогти забезпечити безпеку та комфорт користувачів, підвищуючи ефективність праці та забезпечуючи їхнє здоров'я в довгостроковій перспективі.

### **4.3 Організація цивільного захисту на житлових об'єктах під час воєнного стану**

Тема кваліфікаційної роботи освітнього рівня «Магістр» присвячена розробці інформаційної системи обліку даних житла. Тому доцільно розглянути питання організації цивільного захисту на житлових об'єктах під час воєнного стану [26].

Надзвичайні ситуації, такі як воєнний стан, ставлять перед суспільством виклики, які потребують обдуманих та ефективних заходів забезпечення безпеки громадян. Однією з найважливіших складових цього процесу є організація цивільного захисту на житлових об'єктах, яка визначається не тільки створенням інфраструктури для запобігання небезпекам, але й швидкою та виваженою реакцією на можливі загрози.

#### **4.3.1 Створення відповідної інфраструктури для безпеки населення під час воєнного стану**

Створення інфраструктури безпеки є ключовим елементом забезпечення захисту населення під час воєнного стану або інших надзвичайних ситуацій. Цей

процес включає в себе широкий спектр заходів, спрямованих на підготовку об'єктів і регіонів до можливих небезпек та швидкого реагування на виниклі ситуації [27].

Кодекс цивільного захисту України визначає, що інфраструктура безпеки для населення включає в себе:

- укриття для населення;
- системи оповіщення населення про надзвичайні ситуації;
- системи зв'язку та управління силами цивільного захисту;
- системи забезпечення населення продовольством, водою, медикаментами та іншими необхідними матеріальними ресурсами;
- системи надання першої медичної допомоги постраждалим;
- системи ліквідації наслідків надзвичайних ситуацій.

Закон України "Про критичну інфраструктуру" визначає, що критична інфраструктура – це сукупність об'єктів, систем та процесів, які забезпечують життєдіяльність людини та суспільства та порушення функціонування яких може призвести до значних суспільних наслідків [29].

Також ефективна система оповіщення є важливою частиною інфраструктури безпеки. Ці системи повинні включати сучасні технології, такі як масове повідомлення через мобільні додатки, соціальні мережі, радіо та телевізійні ефіри. Оповіщення повинно бути не тільки ефективним, але і швидким, щоб громадяни могли вчасно отримати інформацію та прийняти відповідні заходи.

Створення чітких та деталізованих планів евакуації – це ще один важливий аспект інфраструктури безпеки. Плани повинні визначати шляхи руху, пункти збору, а також мати конкретні інструкції для населення. Крім того, важливо визначити безпечні евакуаційні зони, де громадяни можуть знаходитися під час надзвичайних ситуацій [28].

Організація безпечних приміщень та укриття є невід'ємною частиною інфраструктури безпеки на житлових об'єктах. Ці приміщення повинні бути спроектовані та обладнані таким чином, щоб забезпечити захист від різних видів небезпек, включаючи атаки, хімічні загрози або радіаційні випромінювання.

Забезпечення стійкості систем газопостачання та електропостачання є важливим елементом безпеки. Важливо мати плани та технічні рішення для уникнення перебоїв в енергопостачанні, що можуть виникнути внаслідок техногенних чи природних катастроф.

Впровадження систем охорони та відеоспостереження дозволяє підвищити рівень безпеки. Вони допомагають вчасно виявляти потенційні загрози, сприяють взаємодії з правоохоронними органами та забезпечують відслідковування ситуацій [26].

Проведення регулярних тренувань і симуляцій дозволяє перевірити ефективність інфраструктури безпеки та підготовленість населення до дій у надзвичайних ситуаціях. Такі заходи також допомагають виявляти слабкі місця та вносити виправлення в плани та процедури.

Створення інфраструктури безпеки на житлових об'єктах під час воєнного стану вимагає комплексного підходу та врахування різноманітних факторів, що можуть впливати на безпеку громадян. Лише завдяки ретельному плануванню, ефективній координації та використанню сучасних технологій можна забезпечити високий рівень захисту та безпеки у надзвичайних ситуаціях.

#### **4.3.2 Евакуаційні плани та зони під час воєнного стану**

Евакуаційні плани та зони є ключовими компонентами інфраструктури безпеки в умовах воєнного стану чи інших надзвичайних ситуацій. Розробка та впровадження цих елементів є критичними для ефективного управління та захисту населення.

Евакуаційні плани включають визначення чітких шляхів руху, організацію пунктів збору та надання населенню інструкцій для ефективної евакуації. Ці плани повинні бути розроблені з урахуванням географічних особливостей регіону та можливих видів загроз [25].

Щодо евакуаційних зон, важливо визначити різні типи загроз та розподілити територію на зони відповідно до цих ризиків. Врахування

географічних особливостей, планування відстаней та належна інфраструктура є ключовими факторами визначення місць евакуації.

Потрібно також передбачити системи відстеження та контролю, що дозволять ефективно керувати евакуацією та реагувати на зміни обстановки. Взаємодія з правоохоронними органами, рятувальними службами та іншими структурами є важливою для координації дій та оптимізації заходів з цивільного захисту.

Ефективність планів евакуації та належно організованих евакуаційних зон визначається якістю підготовки та розумінням населення щодо дій у надзвичайних ситуаціях. Лише завдяки цілеспрямованому плануванню та взаємодії всіх відповідальних структур можна забезпечити високий рівень безпеки та ефективності у випадку надзвичайних подій.

#### **4.3.3 Підготовка населення до дій під час воєнного стану**

Підготовка населення до дій у надзвичайних ситуаціях є ключовим аспектом цивільного захисту. Цей процес включає проведення інформаційних кампаній через різні канали, такі як телебачення, радіо та соціальні мережі, з метою забезпечення населення доступною та зрозумілою інформацією щодо потенційних загроз та правил дій у надзвичайних ситуаціях [28].

Основна увага приділяється тренуванням та семінарам, де населення вчаться не лише теоретичним аспектам, але й отримують практичні навички, такі як евакуація, надання першої допомоги та використання екстрених засобів. Розробка особистих та сімейних планів дій, які включають контактну інформацію, місця збору та інші деталі, також є важливою частиною готовності.

На психологічному рівні важливо забезпечити психологічну підтримку та тренування для керування стресом та панікою. Також враховується використання технологій, таких як мобільні додатки та онлайн-ресурси для забезпечення доступу до інформації та навчання в будь-який час.

Тестування готовності через періодичні вправи та тести допомагає перевірити ефективність підготовки населення та внести корективи у плани дій.

Всі ці заходи спрямовані на створення високого рівня готовності населення до дій у надзвичайних ситуаціях та забезпечення його безпеки та виживання.

#### **4.3.4 Контроль та захист комунікацій для житлових об'єктів під час воєнного стану**

Контроль та захист комунікацій для житлових об'єктів важливі для забезпечення безпеки населення в надзвичайних ситуаціях чи воєнного стану. Це включає розробку стратегій для ефективного контролю та забезпечення конфіденційності зв'язку в житлових районах [25].

З метою контролю комунікацій, важливо визначити ключові мережеві вузли та застосовувати резервні канали для надійності зв'язку. Системи аварійного електронного попередження та розсилки інформації населенню також є важливим елементом.

Захист комунікацій передбачає використання шифрування для забезпечення конфіденційності та фізичний захист інфраструктури. Безпека бездротових технологій та планування дій у випадку відмови основних систем також є актуальними аспектами.

Призначення планів дій та підготовка населення до використання різних засобів зв'язку, включаючи альтернативні, також спрямовані на забезпечення оптимального комунікаційного середовища в умовах надзвичайних ситуацій. Загальна мета – створити безпечні та надійні канали зв'язку для ефективного управління та інформування населення в критичних ситуаціях [26].

#### **4.3.5 Роль влади та міжнародних організацій для забезпечення цивільної безпеки житлових об'єктах**

Роль влади та міжнародних організацій у забезпеченні цивільної безпеки на житлових об'єктах полягає в визначенні стратегій та реалізації заходів для ефективного управління та реагування на надзвичайні ситуації.

Влада відіграє ключову роль у розробці планів та координації заходів цивільного захисту. Це включає планування евакуацій, фізичний захист інфраструктури та публічне повідомлення для громадськості. Також важливо забезпечити підготовку громадян до дій у надзвичайних ситуаціях.

Міжнародні організації надають технічну та фінансову підтримку для удосконалення систем цивільного захисту та забезпечення стійкості житлових об'єктів. Вони сприяють обміну досвідом, розробці міжнародних стандартів безпеки та організації навчань для підвищення кваліфікації фахівців [28].

Загалом, співпраця між владою та міжнародними партнерами визначає ефективність та стійкість систем цивільного захисту, спроможних ефективно реагувати на різноманітні загрози та надзвичайні ситуації.

#### **4.4 Висновок до четвертого розділу**

В четвертому розділі кваліфікаційної роботи розглянуто варіанти утворення сприятливих умов у колективі, електромагнітне випромінювання персональних комп'ютерів на людину. Розглянуто організацію цивільного захисту на житлових об'єктах під час воєнного стану.

В контексті сучасного використання персональних комп'ютерів, забезпечення безпеки та здоров'я працівників вимагає комплексного підходу до організації робочого середовища. Ефективні заходи з охорони праці включають в себе використання ергономічного обладнання, оптимізацію освітлення, застосування антиблікових екранів та фільтрів, регулярні перерви та використання спеціальних окулярів.

Важливо також акцентувати увагу на навчанні та інформуванні працівників щодо правил безпеки та користування комп'ютерами. Це дозволяє їм усвідомлювати важливість власного здоров'я та вживати необхідні заходи для попередження можливих ризиків.

Загальна концепція полягає в створенні умов, які дозволяють працівникам працювати ефективно, не зашкоджуючи своєму фізичному та психічному здоров'ю. Із розвитком технологій та впровадженням нових рішень для

зменшення впливу електромагнітних випромінювань, можна сподіватися на подальше поліпшення у сфері охорони праці та створення ще безпечніших робочих умов.

У контексті забезпечення безпеки на житлових об'єктах в надзвичайних ситуаціях та в умовах воєнного стану, важливо враховувати комплексний підхід, який об'єднує зусилля влади та міжнародних організацій. Влада відіграє визначальну роль у плануванні, координації та реалізації заходів цивільного захисту на місцевому та національному рівнях. Це включає розробку евакуаційних планів, фізичний захист інфраструктури та підготовку громадян до дій у надзвичайних ситуаціях.

Міжнародні організації виступають у ролі партнерів, надаючи технічну, фінансову підтримку та сприяючи обміну досвідом. Створення міжнародних стандартів безпеки та здійснення спільних тренувань роблять можливим впровадження найкращих практик для ефективного управління надзвичайними ситуаціями. Взаємодія з громадськістю та залучення її до процесів підготовки та реагування є ще однією важливою складовою успішного управління безпекою.

У цілому, висновки полягають в тому, що співпраця влади та міжнародних партнерів є невід'ємним елементом створення стійкої системи цивільного захисту, спроможної ефективно реагувати на різноманітні виклики та забезпечувати безпеку та захист населення.



## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи досягнуто мету, яка полягає в дослідженні предметної області, аналізі літературних джерел та розробці інформаційної системи обробки даних про оренду житла. Дана інформаційна система дозволяє не лише збирати та обробляти дані, але і створює зручні та ефективні інструменти для їхнього відображення, аналізу та взаємодії. Загалом, вибір таких технологій відображає сучасні стандарти розробки програмного забезпечення та сприяє створенню інноваційних та конкурентоздатних рішень в галузі обліку даних про оренду житла.

В першому розділі кваліфікаційної роботи:

1. Подано загальну характеристика інформаційних систем обліку даних про оренду житла.
2. Розглянуто історичну хронологію розвитку інформаційних систем обліку даних про оренду житла.
3. Висвітлено основні причини використання інформаційних систем обліку житла про оренду житла.
4. Проаналізовано технологій інформаційних систем обліку даних про оренду житла.
5. Досліджено використання хмарних технологій для системи обліку даних про оренду житла.
6. Обґрунтовано використання технологій штучного інтелекту для системи обліку даних про оренду житла.
7. Сформовано висновок до першого розділу.

В другому розділі кваліфікаційної роботи:

1. Описано загальні підходи та методи досліджень.
2. Досліджено наявні методології розробки програмного забезпечення для обробки даних оренди житла.
3. Подано Аналіз відомих систем обліку даних про оренді житла.
4. Сформовано основні вимоги до розробки інформаційної системи обліку даних про оренду житла.

В третьому розділі кваліфікаційної роботи:

1. Розроблено архітектуру інформаційної системи обліку даних про оренду житла.
2. Запропоновано основні інструменти для розробки системи обліку даних житла.
3. Спроектовано інформаційну панель обліку даних про оренду житла.

В розділі «Охорона праці та безпека в надзвичайних ситуаціях» проаналізовано варіанти створення сприятливих і безпечних умов для персоналу, вплив електромагнітного випромінювання ПК, організація цивільного захисту житлових об'єктів під час воєнного стану.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Любар О. О., Іщенко Я. П. Організація та методика обліку орендних операцій у підприємствах. Економіка, фінанси, менеджмент: актуальні питання науки і практики, Вінницький національний аграрний університет, 2018. С. 102-103.
2. Реут А. Г., Когатько Ю. Л. Житлова проблема в Україні на тлі війни. Якість життя та соціальна політика: зб. матеріалів доп. учасн. демографія та соціальна економіка. Інститут демографії та соціальних досліджень ім. М. В. Птухи НАН України. 2022. С. 123.
3. Державна житлова політика в Україні: сучасний стан та перспективи реформування. *Cedos* : веб-сайт. URL: <https://cedos.org.ua> (дата звернення: 10.12.2023).
4. Федорів П. Державна житлова політика в Україні: сучасний стан та перспективи. Якість життя та соціальна політика: зб. матеріалів доп. учасн. демографія та соціальна економіка. Інститут демографії та соціальних досліджень ім. М. В. Птухи НАН України. 2022. С. 113.
5. Для чого потрібні "header" і "footer"? *CSS In* : веб-сайт. URL: <https://css.in.ua/html/tag/header> (дата звернення: 10.12.2023).
6. 37 тез про сучасний стан, виклики і принципи нової житлової політики в Україні. *Cedos* : веб-сайт. URL: <https://cedos.org.ua/researches/37-tez-pro-suchasnuj-stan-vyklyky-i-pryncyipy-novoyi-zhytlovoi-polityky-v-ukrayini/> (дата звернення: 10.12.2023).
7. Житло і війна в Україні (24 березня — 3 червня 2022). *Cedos* : веб-сайт. URL: <https://cedos.org.ua/researches/zhytlo-i-vijna-v-ukrayini-24-bereznya-3-cherhvnya-2022/> (дата звернення: 10.12.2023).
8. Черенько Л.М. Оренда житла в Україні: стан та тенденції". Дослідження Інституту демографії та соціальних досліджень ім. М.В. Птухи НАН України. 2022. Вип. 148. С.40-41.
9. Аналіз предметної області. *Grafiati* : веб-сайт. URL: <https://helpiks.org/> (дата звернення: 10.12.2023).

10. Пілевич Д. С. Ідентифікація сутності інформаційної системи обліку та її основні характерні риси. Економічний вісник. Серія: фінанси, облік, оподаткування. 2018. Вип. 2. С.153-155.
11. Напрямки розвитку інформаційних систем і технологій обліку підприємств. *Ефективна економіка* : веб-сайт. URL: <http://www.economy.nauka.com.ua/> (дата звернення: 10.12.2023).
12. Що таке архітектура веб-сайту. *Theastrologypage* : веб-сайт. URL: <https://uk.theastrologypage.com/website-architecture> (дата звернення: 10.12.2023).
13. Життєвий цикл програмного забезпечення. *StudFiles* : веб-сайт. URL: <https://studfile.net/preview/9295990/page:3> (дата звернення: 10.12.2023).
14. Готович В.А., Ралік І.Р. Програмне забезпечення на основі клієнт-серверної архітектури для обліку реалізації товарів в торгівлі. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 7-8 грудня 2022 року. С.126.
15. HTML5+CSS3. *Astwellsoft* : веб-сайт. URL: <https://astwellsoft.com/uk> (дата звернення: 10.12.2023).
16. Дизайн інтерфейсу користувача. *Wikipedia* : веб-сайт. URL: [https://en.wikipedia.org/wiki/User\\_interface\\_design](https://en.wikipedia.org/wiki/User_interface_design) (дата звернення: 10.12.2023).
17. Огляд методів управління проектами. *ITpedia* : веб-сайт. URL: <https://uk.itpedia.nl> (дата звернення: 10.12.2023).
18. Готович В.А., Мачужак А.В. Застосування методології CI/CD для автоматизації процесів тестування та розгортання програмного забезпечення. Матеріали XI Міжнародної науково-практичної конференції молодих учених та студентів «АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ» – Тернопіль, 7-8 грудня 2022 року. С.131-132.
19. Gotovych V., Nazarevych O., Shcherbak L. Mathematical modeling of the regular-mode electric power supply and electric power consumption processes of the organization. *Scientific Journal of TNTU (Tern., 2018)*, vol 91, No 3, pp. 134–142. URL: [https://doi.org/10.33108/visnyk\\_tntu2018.03.134](https://doi.org/10.33108/visnyk_tntu2018.03.134)

20. Шлях підвищення життєдіяльності. КонфКонтат : веб-сайт. URL: [http://www.confcontact.com/2013-specproekt/tn15\\_grinyuk.htm](http://www.confcontact.com/2013-specproekt/tn15_grinyuk.htm). (дата звернення: 10.12.2023).

21. Факти зниження і шляхи підвищення життєдіяльності людини. UkrReferat : веб-сайт. URL: <https://ukrreferat.com/chapters/bjd/fakti-znizhennya-i-shlyahi-pidvishhennya-zhittediyalnosti-lyudini-referat-1.html> (дата звернення: 10.12.2023).

22. Управління персоналом. *Освіта.ua* : веб-сайт. URL: <https://osvita.ua/vnz/reports/management/13649/> (дата звернення: 10.12.2023).

23. Створення безпечних і нешкідливих умов праці. Державний нагляд за охороною праці. *WikiLegalAid* : веб-сайт. URL: <https://wiki.legalaid.gov.ua/> (дата звернення: 10.12.2023).

24. Створення сприятливих умов праці на виробництві. *Охорана праці і пожежна безпека* : веб-сайт. URL: <https://oppb.com.ua/news/stvorennya-spryuatlyvuyh-umov-prasi-na-vyrobnyctvi> (дата звернення: 10.12.2023).

25. Організація цивільного захисту на житлових об'єктах під час воєнного стану: теоретичні та практичні аспекти. *LIGA 360* : веб-сайт. URL: <https://ips.ligazakon.net/document/view/ji09215g?an=2> (дата звернення: 10.12.2023).

26. Забезпечення безпеки населення на житлових об'єктах під час воєнного стану. *WikiLegalAid* : веб-сайт. URL: <https://wiki.legalaid.gov.ua/> (дата звернення: 10.12.2023).

27. Особливості організації цивільного захисту на житлових об'єктах у сучасних умовах. *На Урок* : веб-сайт. URL: <https://naurok.com.ua/prezentaciya-organizaciya-civilnogo-zahistu-v-suchasnih-umovah-49856.html> (дата звернення: 10.12.2023).

28. Забезпечення безпеки населення на житлових об'єктах під час воєнного стану: моніторинговий звіт. *WikiLegalAid* : веб-сайт. URL: <https://wiki.legalaid.gov.ua/> (дата звернення: 10.12.2023).

29. UML діаграма. *Evergreen* : веб-сайт. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 10.12.2023).

30. Етапи створення сайту. *ВебСайт* : веб-сайт. URL: <https://websait.uz.ua/website-development/etapy-rozrobky-sajtu/> (дата звернення: 10.12.2023).

31. React Native. Learn once, write anywhere. *React Native* : веб-сайт. URL: <https://reactnative.dev/> (дата звернення: 10.12.2023).

32. Проектування веб-сайту. *Сайт ЗНУ* : веб-сайт. URL: <https://sites.znu.edu.ua/webprog/lect/1194.ukr.html> (дата звернення: 10.12.2023).

33. Що таке Open Server? *Open Server* : веб-сайт. URL: <https://ospanel.io/> (дата звернення: 10.12.2023).

34. Client-server model. *Monovm* : веб-сайт. URL: <https://monovm.com/blog/what-is-client-server-model/> (дата звернення: 10.12.2023).

35. What is Web Hosting? *Namecheap* : веб-сайт. URL: <https://www.namecheap.com/hosting/what-is-web-hosting-definition/> (дата звернення: 10.12.2023).

36. Node - Up and Running. *Academia* : веб-сайт. URL: [https://www.academia.edu/35253057/Node\\_Up\\_and\\_Running](https://www.academia.edu/35253057/Node_Up_and_Running) (дата звернення: 10.12.2023).

37. Mastering Node.js: Your Comprehensive Guide to Going from Zero to Hero and Nailing that Interview. *Medium* : веб-сайт. URL: <https://medium.com/@debasisacharya1977/mastering-node-js-your-comprehensive-guide-to-going-from-zero-to-hero-and-nailing-that-interview-3c65410d078f> (дата звернення: 10.12.2023).

38. JavaScript. Основи веб-програмування. *W3schools* : веб-сайт. URL: <https://w3schoolsua.github.io/js/index.html#gsc.tab=0> (дата звернення: 10.12.2023).

39. Design Patterns in Node.js. *Medium* : веб-сайт. URL: <https://medium.com/@techsuneel99/design-patterns-in-node-js-31211904903e> (дата звернення: 10.12.2023).

40. React Dashboard Libraries: Which One To Use in 2023? *Luzmo* : веб-сайт. URL: <https://www.luzmo.com/blog/react-dashboard> (дата звернення: 10.12.2023).
41. Іконки для веб-застосунків. *Flaticon* : веб-сайт. URL: <https://www.flaticon.com/> (дата звернення: 10.12.2023).
42. Lupenko, S., Lytvynenko, I., Hotovych, V. Simulation of cyclic signals (generalized approach). 4th International Conference on Informatics and Data-Driven Medicine, IDDM 2021. Valencia. 19 November 2021. CEUR Workshop Proceedings. Vol. 3038, P. 86-92. ISSN 1613-0073 URL: <https://ceur-ws.org/Vol-3038/short2.pdf>
43. Information Dashboard Design: The Effective Visual Communication of Data. ACM : веб-сайт. URL: <https://dl.acm.org/doi/10.5555/1206491> (дата звернення: 10.12.2023).
44. Information Design: A User-Centered Approach. *Interaction Design* : веб-сайт. URL: <https://www.interaction-design.org/literature/topics/user-centered-design> (дата звернення: 10.12.2023).
45. Підбір кольорів. *Imagecolorpicker* : веб-сайт. URL: <https://imagecolorpicker.com/uk/color-code/2596be> (дата звернення: 10.12.2023).
46. React Dashboards and Data Visualization. *Medium* : веб-сайт. URL: <https://blog.stackademic.com/enhancing-a-real-time-data-visualization-dashboard-with-react-js-and-python-0e553df57855> (дата звернення: 10.12.2023).
47. Google Cloud Platform documentation. OAuth 2.0. *Rapid7* : веб-сайт. URL: <https://docs.rapid7.com/insightidr/google-cloud-platform/> (дата звернення: 10.12.2023).
48. Practical Guidelines for Effective Dashboard Design. *RubyGarage* : веб-сайт. URL: <https://rubygarage.org/blog/information-design-on-dashboards> (дата звернення: 10.12.2023).
49. Information Design: A User-Centered Approach. *Usability.gov* : веб-сайт. URL: <https://www.usability.gov/what-and-why/user-centered-design.html> (дата звернення: 10.12.2023).
50. React for Data Visualization. *React for Dataviz* : веб-сайт. URL: <https://reactfordataviz.com/> (дата звернення: 10.12.2023).

51. Property management software. *Propertyware* : веб-сайт. URL: <https://www.propertyware.com/> (дата звернення: 10.12.2023).

52. AppFolio Property Manager. *AppFolio* : веб-сайт. URL: <https://www.appfolio.com/> (дата звернення: 10.12.2023).

53. Rentometer About Us. *Rentometer* : веб-сайт. URL: <https://www.rentometer.com/about> (дата звернення: 10.12.2023).

54. Buy your next home with a brand. *Zillow Home Loans* : веб-сайт. URL: <https://www.zillow.com/homeloans> (дата звернення: 10.12.2023).



# ДОДАТКИ

Тези конференції

---

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Тернопільський національний технічний університет імені Івана Пулюя (Україна)  
Університет імені П'єра і Марії Кюрі (Франція)  
Маріборський університет (Словенія)  
Технічний університет у Кошице (Словаччина)  
Вільнюський технічний університет ім. Гедимінаса (Литва)  
Міжнародний університет цивільної авіації (Марокко)  
Наукове товариство ім. Т.Шевченка

**АКТУАЛЬНІ ЗАДАЧІ  
СУЧАСНИХ ТЕХНОЛОГІЙ**

**Збірник**  
тез доповідей

**XII Міжнародної науково-практичної  
конференції молодих учених та студентів**  
6-7 грудня 2023 року



**УКРАЇНА**  
**ТЕРНОПІЛЬ – 2023**

**УДК 004.51**

**В. І. Козак; В. А. Готович, к.т.н.**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

**ДОСЛІДЖЕННЯ ВАРІАНТІВ ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ  
КОРИСТУВАЧА В ІНФОРМАЦІЙНИХ ІНТЕРАКТИВНИХ АНАЛІТИЧНИХ  
ПАНЕЛЯХ**

**V. Kozak, V. Hotovych, Ph.D.**

**RESEARCH OF USER INTERFACE DESIGN OPTIONS IN INFORMATION  
INTERACTIVE ANALYTICAL PANELS**

Інформаційні інтерактивні аналітичні панелі (англ. Dashboard, дашборд) є потужним інструментом для візуалізації даних та підтримки прийняття рішень. Вони дозволяють користувачам швидко і легко отримувати інформацію про стан бізнесу або інших процесів. Однак, для того щоб дашборди були ефективними, вони повинні мати добре продуманий інтерфейс користувача (UI).

Першим фактором, який слід враховувати при проектуванні UI дашборда, є тип даних, які відображаються в ньому. Дані можуть бути статичними або динамічними. Статичні дані не змінюються з часом, тоді як динамічні дані постійно оновлюються. Статичні дані можна представити за допомогою таких елементів UI, як таблиці, діаграми та карти. Динамічні дані можна представити за допомогою таких елементів UI, як індикатори, сигнали та інфографіка [1, 4].

Ще одним фактором, який слід враховувати при проектуванні UI дашборда, є цільова аудиторія. Дашборди можуть бути призначені для широкого кола користувачів або для вузької групи експертів. Дашборди для широкого кола користувачів повинні бути простими та інтуїтивно зрозумілими. Вони повинні використовувати зрозумілі терміни та символи. Дашборди для експертів можуть використовувати більш складні елементи UI, такі як фільтри та діапазони [3].

На сьогоднішній день існує велика кількість досліджень, присвячених проектуванню UI для інформаційних дашбордів. Однак відомі дослідження часто фокусуються на окремих аспектах інтерфейсу, не розглядаючи комплексне проектування дашбордів. Крім того, такі рішення часто не враховують різноманітність користувачів та їхні потреби.

Деякі дослідження зосереджуються на вирішенні конкретних аспектів дизайну інтерфейсу. Наприклад, деякі праці вивчають оптимальне розміщення ключових показників на дашборді для максимальної зручності сприйняття інформації. Інші досліджують використання кольорів і графіків для підвищення чіткості та зрозумілості представленої інформації [2].

Проте, для вирішення проблеми комплексного підходу до дизайну інтерфейсу, пропонується інтеграція різноманітних підходів. Це може включати в себе врахування психологічних особливостей користувачів, персоналізацію інтерфейсу залежно від виду даних та вдосконалення навігації. Тому, для даного етапу варто інтегрувати технології штучного інтелекту та адаптивних систем. Зрозуміло, що розуміння потреб користувачів є ключовим елементом. Однак, використання алгоритмів машинного навчання для аналізу поведінки користувачів та налаштування інтерфейсу може значно полегшити взаємодію.

Пропонується також розвивати концепції гейміфікації у дизайні інтерфейсу для стимулювання активності та зацікавленості користувачів (рис. 1) у використанні дашбордів. Використання елементів гри може зробити процес взаємодії з даними більш привабливим та підвищити мотивацію користувачів до систематичного використання інформаційних джерел [5, 6].



Рисунок 1. Приклад гейміфікаційного дашборду

Важливою задачею є розробка універсальних стандартів для інтерфейсів дашбордів, що сприятиме вирішенню питань сумісності та переносу інтерфейсних рішень між різними системами. Такий комплексний підхід може значно поліпшити зручність використання інформаційних дашбордів та сприяти їхній широкій інтеграції в різноманітні сфери діяльності, починаючи від бізнесу і закінчуючи наукою та освітою.

**Висновок.** Дослідження у галузі проектування інтерфейсу користувача для інформаційних дашбордів вже досягло значних успіхів, проте потребує подальших розвинених рішень для оптимізації користувацького досвіду та ефективності використання.

Перевагою існуючих досліджень є акцент на окремих аспектах, таких як вибір графічних елементів та оптимальне розміщення інформації на дашборді. Застосування алгоритмів машинного навчання та адаптивних систем для індивідуалізації інтерфейсу є важливим напрямком, спрямованим на поліпшення зручності взаємодії. Згадане в дослідженні використання гейміфікації є цікавим підходом для стимулювання активності користувачів, але його ефективність та прийняття користувачами вимагає додаткового вивчення та експериментів. Щодо стандартизації, ідея розробки універсальних стандартів є важливою для забезпечення сумісності та переносу інтерфейсних рішень. Однак, важливо враховувати динаміку розвитку технологій та бізнес-потреб, щоб стандарти не стали застарілими.

### Література

1. Andy Cotgreave 2022. Data Visualization for Business Intelligence
2. Helen Papagiannis 2018. Augmented Human: How Technology Is Shaping the New Reality.
3. Stephen Few 2018. Information Dashboard Design: Displaying Data for At-a-Glance Monitoring.
4. Robin Williams 2021. Information Design: A User-Centered Approach
5. Deter W. Stone 2018. The Gamification of Learning and Instruction: Game-Based Methods
6. Cole Nussbaumer Knaflic 2019. The Data Visualist's Handbook

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ  
«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»



13-14 грудня 2023 року

ТЕРНОПІЛЬ  
2023



УДК 004.51

**В. І. Козак**

Тернопільський національний технічний університет імені Івана Пулюя, Україна

## **ВИКОРИСТАННЯ БІБЛІОТЕКИ REACT ДЛЯ ВИВЕДЕННЯ СТАТИСТИЧНИХ ДАНИХ У ВИГЛЯДІ ІНФОРМАЦІЙНОЇ ПАНЕЛІ**

**V. Kozak**

### **USING THE REACT LIBRARY TO DISPLAY STATISTICAL DATA IN A DASHBOARD VIEW**

Стрімка динаміка технологічного прогресу та зростання обсягів опрцьовуваних даних вимагає нових підходів до візуалізації та аналізу інформації. На сьогоднішній день React став однією з найбільш популярних бібліотек для розробки графічних інтерфейсів користувача для веб-сторінок. Проте, необхідність в зручних інструментах для відображення статистичних даних залишається актуальною. До поточного моменту в багатьох проєктах використовувалися різні бібліотеки та підходи для створення інформаційних панелей (дашбордів) з використанням бібліотеки React, що викликало неоднозначні результати та труднощі у подальшій підтримці.

У деяких дослідженнях розглянуто питання створення інтерактивних інтерфейсів на React, однак, особливої уваги не було приділено аспектам виведення статистичних даних. Проблема полягає у відсутності єдиної та ефективної методології для створення дашбордів, які б задовольняли потреби розробників та користувачів таких програмних продуктів [1].

Дане дослідження вирішує цю проблему шляхом використання бібліотеки React та сучасних підходів до розробки дашбордів. При розробці необхідно врахувати тип даних. Бібліотеки для створення дашбордів можуть підтримувати різні типи даних, наприклад, числові, текстові, графічні. Слід також звертати увагу на інтерактивність, оскільки дашборди можуть бути різними за ступенем інтерактивності. Бібліотеки можуть надавати різні можливості для взаємодії користувачів з дашбордом. Також бібліотеки для створення дашбордів можуть підтримувати різні можливості для розширення, наприклад, використання кастомних компонентів або створення власних візуалізацій [2, 3].

Вибір конкретної бібліотеки залежить від напрямку розробки. В React є кілька бібліотек, які дозволяють створювати дашборди. Найпопулярнішим є Redash – потужна бібліотека, яка дозволяє створювати складні дашборди з підтримкою різних типів даних та інтерактивності [4, 6].

Створений дашборд може легко адаптуватися до різних потреб та включає в себе компоненти для графіків, діаграм, фільтрів, та інших елементів візуалізації даних (рис. 1). Це спрощує розробку та розширення функціональності додатку.

Також слід відзначити кілька додаткових рекомендацій, які можуть бути корисними для розробників дашбордів з використанням React [5]:

- 1) варто використовувати візуалізації, які відповідають меті дашборду. Наприклад, якщо мета дашборду – відстежити тенденції, то слід використовувати візуалізації, які дозволяють побачити зміни в даних з часом;
- 2) візуалізацію потрібно обирати таку, яку легко зрозуміти користувачеві;
- 3) варто пам'ятати про інтерактивні елементи, щоб користувачі могли взаємодіяти з дашбордом. Інтерактивні елементи дозволяють користувачам отримувати більше інформації з дашборду. Важливим аспектом є можливості взаємодії та інтерактивності, такі як фільтрація, сортування та анімації.



Рисунок 1. Приклад використання дашбордів у React

Найкращі практики для створення дашбордів з використанням React включають розділення на компоненти для зручного управління та розширення коду, оптимізацію відображення за допомогою механізмів, таких як «React.memo», та забезпечення адаптивності до різних пристроїв. Важливо також забезпечити безпеку дашборду та врахувати вартість та тип ліцензії конкретної обраної бібліотеки.

**Висновок.** Компонентний підхід бібліотеки React сприяє простоті розширення та підтримки додатку. Використання спеціалізованої бібліотеки разом із React дозволяє економити час розробки та вдосконалити користувацький досвід.

Аналіз показує, що такий підхід вирішує проблему неоднорідності підходів до створення дашбордів та полегшує їхнє використання в проєктах на React. Однак, для підтримки та розвитку дослідження, можливі напрямки для подальших досліджень включають аналіз ефективності додатку на великих обсягах даних та розширення функціоналу для врахування специфічних потреб проєктів.

Запропонована в даній праці методологія дозволяє розробникам швидко та ефективно виводити статистику на веб-сторінках, що має велике значення для індустрії великих обсягів даних та аналізу інформації.

#### Література

1. Stephen Few 2018. Information Dashboard Design: Displaying Data for At-a-Glance Monitoring.
2. Robin Williams 2021. Information Design: A User-Centered Approach
3. Alex Banks 2022. React for Data Visualization
4. David Khourshid 2021. React Dashboards: Build Data-Driven Web Applications with React
5. Maximilian Schwarzmüller 2020. React Charts and Dashboards
6. Redash. Query and Visualize Graphite in Redash. <https://redash.io/data-sources/graphite>

## Структура базових компонентів для реалізації інформаційної системи обліку даних про оренду житла

### Лістинг В.1 – Сторінка авторизації

```
import { useLogin } from "@refinedev/core";
import { useEffect, useRef } from "react";

import Box from "@mui/material/Box";
import Container from "@mui/material/Container";
import Typography from "@mui/material/Typography";
import { ThemmedTitleV2 } from "@refinedev/mui";

import { houserent } from '../assets';

import { CredentialResponse } from "../interfaces/google";

export const Login: React.FC = () => {
  const { mutate: login } = useLogin<CredentialResponse>();

  const GoogleButton = (): JSX.Element => {
    const divRef = useRef<HTMLDivElement>(null);

    useEffect(() => {
      if (typeof window === "undefined" || !window.google ||
!divRef.current) {
        return;
      }

      try {
        window.google.accounts.id.initialize({
          ux_mode: "redirect",
          client_id: process.env.REACT_APP_GOOGLE_CLIENT_ID,
          callback: async (res: CredentialResponse) => {
            if (res.credential) {
              login(res);
            }
          },
        });
      } catch (error) {
        console.log(error);
      }
    }, []);

    return <div ref={divRef} />;
  };
};
```



```

    };

    return (
    <Box component="div" sx={{ backgroundColor: "#FCFCFC" }}>
        <Container
            component="main"
            maxWidth="xs"
            sx={{
                display: "flex",
                flexDirection: "column",
                justifyContent: "center",
                height: "100vh",
            }}
        >
            <Box
                sx={{
                    display: "flex",
                    justifyContent: "center",
                    flexDirection: "column",
                    alignItems: "center",
                }}
            >
                <div>
                    <img src={houserent} alt="HouseRent
Logo" />
                </div>
                <Box mt={4}>
                    <GoogleButton />
                </Box>
            </Box>
        </Container>
    </Box>
    );
};

```

## Лістинг В.2 – Файл основної сторінки Home

```

import { useList } from "@refinedev/core";
import Typography from "@mui/material/Typography";
import Box from "@mui/material/Box";
import Stack from "@mui/material/Stack";

import {
    PieChart,
    PropertyReferrals,
    TotalRevenue,
    PropertyCard,
} from "components";

const Home = () => {
    const { data, isLoading, isError } = useList({
        resource: "properties",
        config: {
            pagination: {

```

```

        pageSize: 4,
      },
    },
  });

  const latestProperties = data?.data ?? [];

  if (isLoading) return <Typography>Loading...</Typography>;
  if (isError) return <Typography>Something went
wrong!</Typography>;

  return (
    <Box>
      <Typography      fontSize={25}      fontWeight={700}
color="#11142D">
        Dashboard
      </Typography>

      <Box  mt="20px"  display="flex"  flexWrap="wrap"
gap={4}>
        <PieChart
          title="Properties for Sale"
          value={684}
          series=[[75, 25]]
          colors=[["#275be8", "#c4e8ef"]]
        />
        <PieChart
          title="Properties for Rent"
          value={550}
          series=[[60, 40]]
          colors=[["#275be8", "#c4e8ef"]]
        />
        <PieChart
          title="Total customers"
          value={5684}
          series=[[75, 25]]
          colors=[["#275be8", "#c4e8ef"]]
        />
        <PieChart
          title="Properties for Cities"
          value={555}
          series=[[75, 25]]
          colors=[["#275be8", "#c4e8ef"]]
        />
      </Box>

      <Stack
        mt="25px"
        width="100%"
        direction={{ xs: "column", lg: "row" }}
        gap={4}
      >
        <TotalRevenue />
        <PropertyReferrals />
      </Stack>
    </Box>
  );

```

```

        <Box
            flex={1}
            borderRadius="15px"
            padding="20px"
            bgcolor="#fcfcfc"
            display="flex"
            flexDirection="column"
            minWidth="100%"
            mt="25px"
        >
            <Typography fontSize="18px" fontWeight={600}
color="#11142d">
                Latest Properties
            </Typography>

            <Box
                mt={2.5}
                sx={{ display: "flex", flexWrap: "wrap",
gap: 4 }}
            >
                {latestProperties.map((property) => (
                    <PropertyCard
                        key={property._id}
                        id={property._id}
                        title={property.title}
                        location={property.location}
                        price={property.price}
                        photo={property.photo}
                    />
                ))}
            </Box>
        </Box>
    );
};

export default Home;

```

### Лістинг В.3 – Сторінка агента з нерухомості

```

import { useList } from "@refinedev/core";
import Box from "@mui/material/Box";
import Typography from "@mui/material/Typography";

import { AgentCard } from "components";

const Agents = () => {
    const { data, isLoading, isError } = useList({ resource:
"users" });

    const allAgents = data?.data ?? [];

    if (isLoading) return <div>loading...</div>;
    if (isError) return <div>error...</div>;

```

```

        return (
          <Box>
            <Typography  fontSize={25}  fontWeight={700}
color="#11142d">
              Agents List
            </Typography>

            <Box
              mt="20px"
              sx={{
                display: "flex",
                flexWrap: "wrap",
                gap: "20px",
                backgroundColor: "#fcfcfc",
              }}
            >
              {allAgents.map((agent) => (
                <AgentCard
                  key={agent._id}
                  id={agent._id}
                  name={agent.name}
                  email={agent.email}
                  avatar={agent.avatar}
                  noOfProperties={agent.allProperties.l
length}
                </AgentCard
              </>
                ))}
            </Box>
          </Box>
        );
      };

export default Agents;

```

#### Лістинг В.4 – Сторінка з списком нерухомості

```

import Add from "@mui/icons-material/Add";
import { useTable } from "@refinedev/core";
import Box from "@mui/material/Box";
import Stack from "@mui/material/Stack";
import Typography from "@mui/material/Typography";
import TextField from "@mui/material/TextField";
import Select from "@mui/material/Select";
import MenuItem from "@mui/material/MenuItem";
import { useNavigate } from "react-router-dom";
import { useMemo } from "react";

import { PropertyCard, CustomButton } from "components";

const AllProperties = () => {
  const navigate = useNavigate();

  const {

```

```

        tableQueryResult: { data, isLoading, isError },
        current,
        setCurrent,
        setPageSize,
        pageCount,
        sorter,
        setSorter,
        filters,
        setFilters,
    } = useTable();

    const allProperties = data?.data ?? [];

    const currentPrice = sorter.find((item) => item.field ===
"price")?.order;

    const toggleSort = (field: string) => {
        setSorter([
            { field, order: currentPrice === "asc" ?
"desc" : "asc" }
        ]);
    };

    const currentFilterValues = useMemo(() => {
        const logicalFilters = filters.flatMap((item) =>
            "field" in item ? item : [],
        );
        return {
            title:
                logicalFilters.find((item) => item.field ===
"price")?.value ||
                "",
            propertyType:
                logicalFilters.find((item) => item.field ===
"propertyType")
                    ?.value || "",
        };
    }, [filters]);

    if (isLoading) return <Typography>Loading...</Typography>;
    if (isError) return <Typography>Error...</Typography>;

    return (
        <Box>
            <Box mt="20px" sx={{ display: "flex", flexWrap:
"wrap", gap: 3 }}>
                <Stack direction="column" width="100%">
                    <Typography fontSize={25} fontWeight={700}
color="#11142d">
                        {
                            !allProperties.length
                                ? "There are no properties"
                                : "All Properties"
                        }
                    </Typography>
                    <Box
                        mb={2}
                        mt={3}

```

```

display="flex"
width="84%"
justifyContent="space-between"
flexWrap="wrap"
>
<Box
  display="flex"
  gap={2}
  flexWrap="wrap"
  mb={{ xs: "20px", sm: 0 }}
  >
    <CustomButton
      title={`Sort price ${
        currentPrice === "asc" ?
"↑" : "↓"}
      }`
      handleClick={() =>
toggleSort("price")}
      backgroundColor="#475be8"
      color="#fcfcfc"
    />
    <TextField
      variant="outlined"
      color="info"
      placeholder="Search by title"
      value={currentFilterValues.title}
      onChange={(e) => {
        setFilters([
          {
            field: "title",
            operator:
"contains",
            value:
e.currentTarget.value
          },
          {
            field: "title",
            operator:
"contains",
            value:
e.currentTarget.value
          }
        ])
      }}
    />
    <Select
      variant="outlined"
      color="info"
      displayEmpty
      required
      inputProps={{ "aria-label":
"Without label" }}
      defaultValue=""
      value={currentFilterValues.propertyType}
      onChange={(e) => {
        setFilters(

```

```

        {
            field:
"propertyType",
            operator:
"eq",
            value:
e.target.value,
        },
    ],
    "replace",
);
}}
>
<MenuItem
value="">All</MenuItem>
    {[
        "Apartment",
        "Villa",
        "Farmhouse",
        "Condos",
        "Townhouse",
        "Duplex",
        "Studio",
        "Chalet",
    ]}.map((type) => (
        <MenuItem
            key={type}
value={type.toLowerCase()}
        >
            {type}
        </MenuItem>
    )))
</Select>
</Box>
</Box>
</Stack>
</Box>
<Stack
    direction="row"
    justifyContent="space-between"
    alignItems="center"
>
    <CustomButton
        title="Add Property"
        handleClick={() =>
navigate("/properties/create")}
        backgroundColor="#475be8"
        color="#fcfcfc"
        icon={<Add />}
    />
</Stack>

```





```

        )
      }
    >
      {[10, 20, 30, 40, 50].map((size) => (
        <MenuItem key={size} value={size}>
          Show {size}
        </MenuItem>
      ))}
    </Select>
  </Box>
)}
</Box>
);
};

export default AllProperties;

```

### Лістинг В.5 – Форма додавання нового об'єкта у БД

```

import { useState } from "react";
import { useGetIdentity } from "@refinedev/core";
import { useForm } from "@refinedev/react-hook-form";

import { FieldValues } from "react-hook-form";

import Form from "components/common/Form";

const CreateProperty = () => {
  const { data: user } = useGetIdentity({
    v3LegacyAuthProviderCompatible: true,
  });
  const [propertyImage, setPropertyImage] = useState({ name:
"", url: "" });
  const {
    refineCore: { onFinish, formLoading },
    register,
    handleSubmit,
  } = useForm();

  const handleImageChange = (file: File) => {
    const reader = (readFile: File) =>
      new Promise<string>((resolve, reject) => {
        const fileReader = new FileReader();
        fileReader.onload = () =>
          resolve(fileReader.result as string);
        fileReader.readAsDataURL(readFile);
      });

    reader(file).then((result: string) =>
      setPropertyImage({ name: file?.name, url: result
    })),
  );
};

```

```

    const onFinishHandler = async (data: FieldValues) => {
      if (!propertyImage.name) return alert("Please select an
image");

      await onFinish({
        ...data,
        photo: propertyImage.url,
        email: user.email,
      });
    };

    return (
      <Form
        type="Create"
        register={register}
        onFinish={onFinish}
        formLoading={formLoading}
        handleSubmit={handleSubmit}
        handleImageChange={handleImageChange}
        onFinishHandler={onFinishHandler}
        propertyImage={propertyImage}
      />
    );
  };
export default CreateProperty;

```

### Лістинг В.6 – Користувацькі запити до MongoDB

```

import mongoose from "mongoose";

const UserSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true },
  avatar: { type: String, required: true },
  allProperties: [{ type: mongoose.Schema.Types.ObjectId,
ref: "Property" }],
});

const userModel = mongoose.model("User", UserSchema);

export default userModel;

```

### Лістинг В.7 – Процес отримання даних з MongoDB

```

import User from "../mongodb/models/user.js";

const getAllUsers = async (req, res) => {
  try {
    const users = await
User.find({}).limit(req.query._end);

    res.status(200).json(users);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};

```

```

    }
  };

const createUser = async (req, res) => {
  try {
    const { name, email, avatar } = req.body;

    const userExists = await User.findOne({ email });

    if (userExists) return
res.status(200).json(userExists);

    const newUser = await User.create({
      name,
      email,
      avatar,
    });

    res.status(200).json(newUser);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};

const getUserInfoByID = async (req, res) => {
  try {
    const { id } = req.params;

    const user = await User.findOne({ _id: id
}).populate("allProperties");

    if (user) {
      res.status(200).json(user);
    } else {
      res.status(404).json({ message: "User not found"
});
    }
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};

export { getAllUsers, createUser, getUserInfoByID };

```

