

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

Розробка та реалізація сайту оренди квартир у місті Києві

**Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення» 121**

Керівник курсової роботи
к.ф.-м.н., ст.викл. Гречко А. В.

(підпис)

“ ” 2021 р.

Виконав студент 3-го курсу
Євтушенко І.О.

“ ” 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

ЗАТВЕРДЖУЮ
Зав. Кафедри
Малашонок Геннадій Іванович

(підпис)
“ ____ ” _____ 2021 р

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

Студента Євтушенка Ігоря Олеговича факультету інформатики 3 курсу
Тема: Розробка сайту для оренди квартири у місті Києві

Зміст ТЧ до курсової роботи

Календарний план

Зміст

Перелік умовних позначень

Вступ

Розділ 1 Аналіз предметної області. Постановка завдання курсової роботи

Розділ 2 Теоретичні відомості

Розділ 3 Опис реалізації програмного продукту

Висновки

Перелік використаних джерел

Додатки

Дата видачі “ ____ ” _____ 2021 р.

Керівник _____ Завдання отримав _____
(підпис) (підпис)

Календарний план виконання курсової роботи

Тема: Розробка сайту для оренди квартир у місті Києві

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	Вересень-Жовтень 2020	
2.	Ознайомлення з існуючими сайтами-аналогами роботи	Березень 2021	
3.	Вибір технологій для роботи над практичною частиною	Березень 2021	
4.	Огляд літератури та онлайн курсів за темою курсової роботи	Квітень 2021	
5.	Розробка дизайну сайту	Квітень 2021	
6.	Створення Баз даних	Квітень 2021	
7.	Розробка веб-застосунку	Квітень-Травень 2021	
8.	Тестування розробленого сайту для оренди квартир	Травень 2021	
9.	Написання текстової частини	Травень 2021	
10.	Перегляд курсової науковим керівником	Травень 2021	
11.	Створення презентації	Травень 2021	

Студента Євтушенка І.О.

Керівник Гречко А.В.

“ ”

Зміст

Перелік умовних позначень	5
ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАННЯ КУРСОВОЇ РОБОТИ	8
1.1 Аналіз сучасного стану питання та обґрунтування теми	8
1.2 Огляд існуючих аналогів розробки	8
1.3 Постановка задачі	13
Розділ 2 Теоретичні відомості	14
Основні поняття про популярну бібліотеку для розробки веб-сервісів React	14
Головний патерн архітектурної розробки складних програм на React	15
JWT (JSON Web Token)	16
Розділ 3 Опис реалізації програмного продукту	19
3.1 Аналіз технічного завдання	19
3.2 Обґрунтування алгоритму й структури програми	19
3.3 Обґрунтування вибору засобів розробки	21
3.4 Опис розробки програми	23
3.5 Створення об'єктів і розробка головної програми	23
3.6 Опис файлів даних та інтерфейсу програми	24
3.7 Тестування програми і результати її виконання	30
Висновки	41
Посилання	42
Додатки	44
Додаток А	44

Перелік умовних позначень

1. СКБД - система керування базами даних.
2. БД - база даних.
3. online - у мережі інтернет.
4. offline - не в мережі інтернет.
5. валідація - це перевірка коректності інформації, що введена у формі для заповнення.
6. AWS - Amazon Web Services.
7. unit test - процес покриття коду тестами, що забезпечує коректну та передбачувану роботу вашого коду.
8. debugging - це процес пошуку та відлагодження помилок у вашій програмі.
9. MVC - Model View Controller.
10. localStorage - це методи та протоколи веб-застосунку, який дозволяє зберігати дані в браузерах.
11. cookie - дані, які відправляє веб-сервер та зберігається на вашому комп'ютері.
12. КМА - Києво-Могилянська академія

ВСТУП

Сьогодні стрімко розвиваються онлайн платформи і всі offline магазини, послуги, бізнес поступово переходять в online. Так ще два роки тому, важко собі було уявити онлайн консультацію у лікарів чи такий стрімкий попит на online викладачів. Але це сьогоднішні реалії й від них нікуди не втекти. Сьогодні простіше відкрити сайт, сидячи вдома, і замовити якийсь товар ніж відстояти довжелезну чергу в магазині, чи, приїхавши додому після важкого робочого дня, замовити вечерю через Racket чи Glovo, а не витратити свій час на приготування.

З початком карантину, ми зіштовхнулися з новою проблемою — з житлом, а саме — якістю пристосування житла до робочих умов. З початком карантину наша квартира стала не лише місцем, де ми їмо та спимо, вона стала нашим офісом, спортзалом, місцем збирання компаній друзів та місцем, де можна відпочити після робочого дня. На жаль, більшість квартир не були готовими до цього, адже в основному, квартири, що здаються в оренду, мають основну ціль — тимчасового притулку, поки мешканець не збере кошти та не купить квартиру своєї мрії. Більше того, сам ринок оренди квартир зазнав певних змін, адже сьогодні йде швидке омолодження міст, тобто все більше та більше молоді не залишаються у своїх маленьких містечках, а перебираються до міст мільйонників (Київ, Харків, Дніпро, Львів та інші). І старі сайти для оренди квартир не зважають на цей фактор, тому і не розвиваються альтернативні способи залучення клієнтів.

Flatty - це простий, зручний, інтуїтивно-зрозумілий у використанні сайт для оренди квартири, який орієнтується на людей віком від 18 до 39 років. Основною особливістю Flatty є режим пошуку квартир Badoo Search, що копіює модель пошуку пари з сайтів знайомств, таких як Badoo чи Tinder. Тобто користувачу у режимі Badoo Search пропонується рандомна квартира з коротким описом та можливістю переглянути всі фотографії. Користувач, залежно від того чи сподобалася дана квартира чи ні, може натиснути на кнопку “Like” та квартиру буде додано до списку вподобаних квартир користувача або ж “Dislike”, і ця квартира більше користувачу не траплятиметься у цьому режимі.

Мета: зробити конкурентний сайт для оренди квартир, який мав би новий особливий функціонал, що не зустрічався б у сайтах-анагогах, для заохочення молоді аудиторії.

Завдання курсової роботи — створити такий сайт для оренди квартир, щоб він задовольняв основні потреби звичайного користувача (шукача квартири для оренди), був візуально приємним та зручним у використанні для відвідувачів сайту.

Для створення веб-сервісу Flatty використовувалося таке програмне забезпечення:

- Для клієнтської частини використовувалася бібліотека React.

- Для встановлення пакетів та інших потрібних бібліотек використовувався NPM (node package manager).
- За дизайн відповідав фреймворк Bootstrap та стилі SCSS.
- Всі дані зберігалися у реляційній системі управління базами даних MySQL.
- Мова програмування - JavaScript, із використанням Node.js та фреймворку Express для серверної частини.

Курсова робота складається зі вступу, трьох розділів, висновку, списку використаних джерел та додатку.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.

ПОСТАНОВКА ЗАВДАННЯ КУРСОВОЇ РОБОТИ

1.1 Аналіз сучасного стану питання та обґрунтування теми

Вже понад рік через небезпечну коронавірусну інфекцію COVID-19 та її штампи десятки мільйонів осіб були змушені повністю піти на самоізоляцію, залишивши звичний спосіб життя, роботи, відпочинку, спорту задля збереження особистої безпеки та припиненню розповсюдження вірусу. На жаль, сьогодні в Україні ситуація з вірусом не стабілізувалася, через це люди й досі продовжують перебувати вдома та обмежувати себе у розвагах, а саме — рідше виходити з квартири, рідше ходити в басейн, рідше ходити в кінотеатр. Через це основним місцем, де проводиться більшість часу і досі залишається квартира, яка слугує не лише для сну та перекусу, як це було до карантину, а й виконує функції проведення зустрічей з друзями, відпочинку в компаніях, є місцем роботи та занять спортом. Також не варто забувати, що майже більшість перейшли на віддалену роботу, через що тисячі людей втратили сенс орендувати дороге житло в Києві.

Але через вакцинування у перспективі року ми знову можемо повернутися до нормального, звичного нам способу життя. А разом з цим збільшиться кількість людей, що переїде до Києва у пошуках роботи, і їм, звичайно ж, потрібне буде житло.

Тому знову розпочнеться активна боротьба на ринку оренди квартир. Flatty спрямований не лише створити непогану конкуренцію, а й зацікавити молодь режимом Badoo Search.

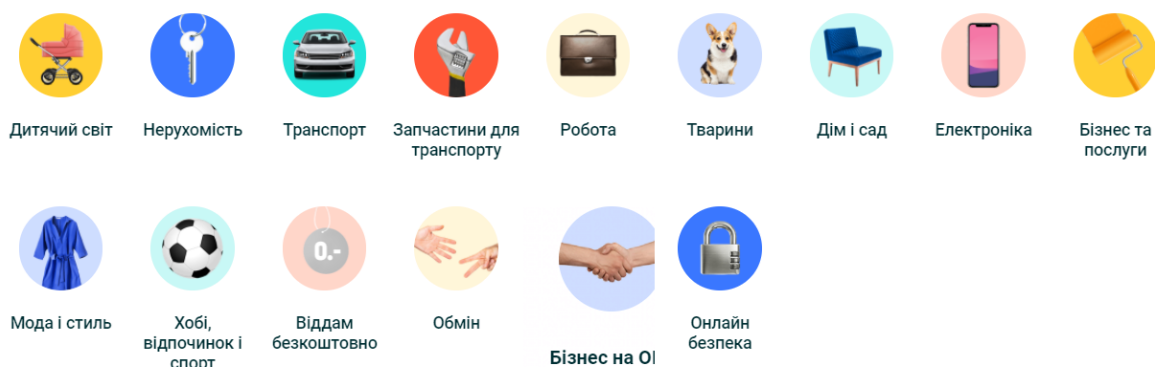
1.2 Огляд існуючих аналогів розробки

Сьогодні лідерами на ринку оренди квартир є Flatfy, OLX, Booking. Усі три сервіси - це титани цієї індустрії і вони охоплюють більшість ринку. Але як і у всіх, у них є недоліки.

OLX

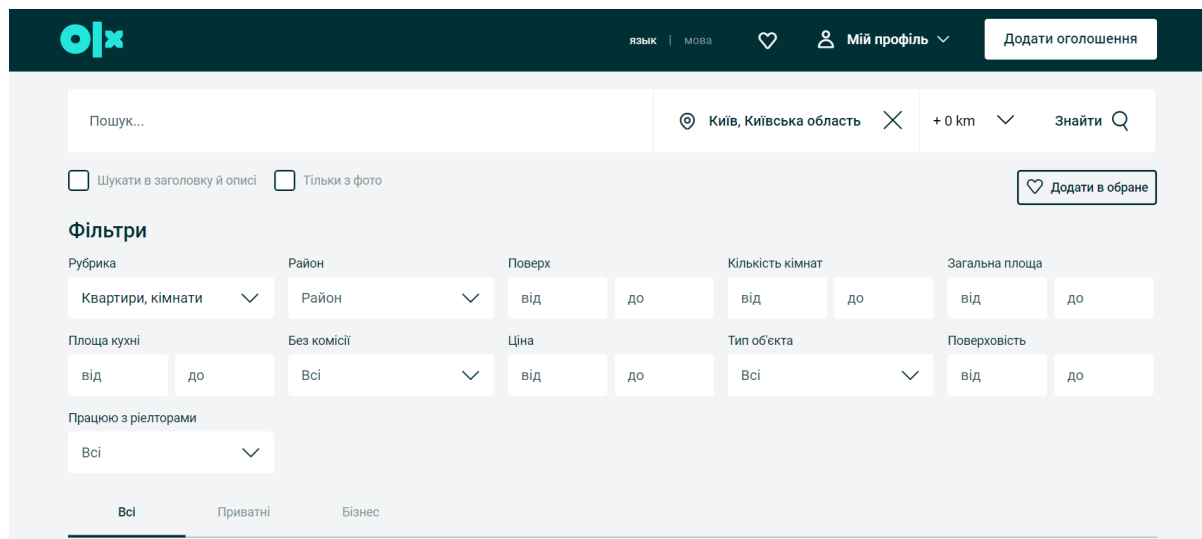
OLX - це веб-сайт чи додаток на IOS та Android. OLX пропонує безліч рубрик (рис.1), тому є універсальним сайтом, де можна знайти майже все, що вам потрібно.

Головні рубрики

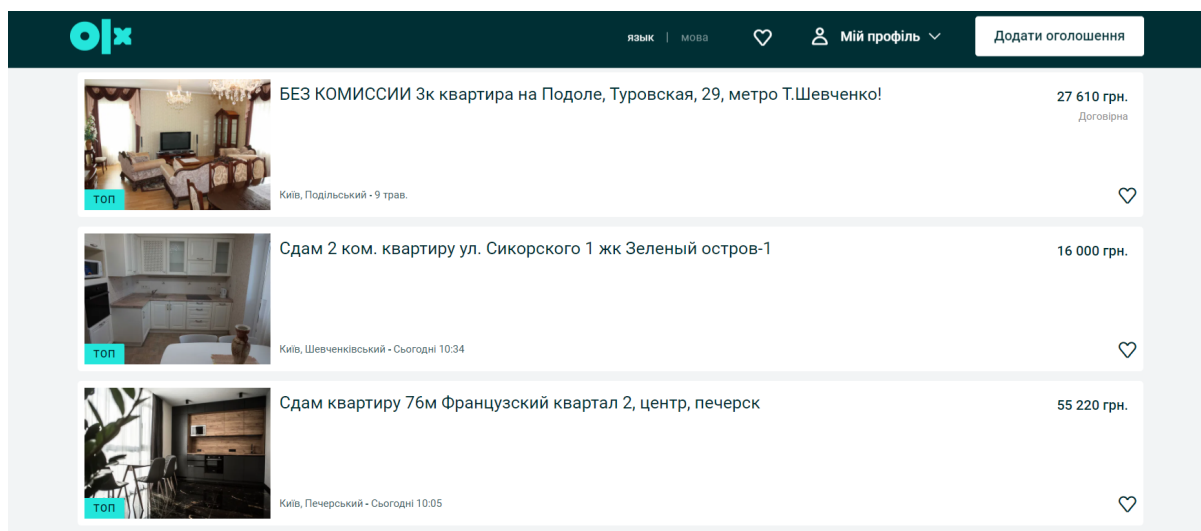


(Рисунок 1. Головні рубрики сайту OLX)

OLX не нова система (у минулому SLANDO), яка сьогодні має не найкращу репутацію. З нею пов'язані сотні афер як від продавців так і від покупців. Тому хоч сайт має непоганий функціонал (рис.2) та широкий вибір квартир (рис.3), але через свою репутацію та шахраїв, які винаходять десятки нових способів шахрайств щороку, послуги сайту вважаються не надійними для нових користувачів. А також на сайті діє обмеження для кількості безкоштовних оголошень.



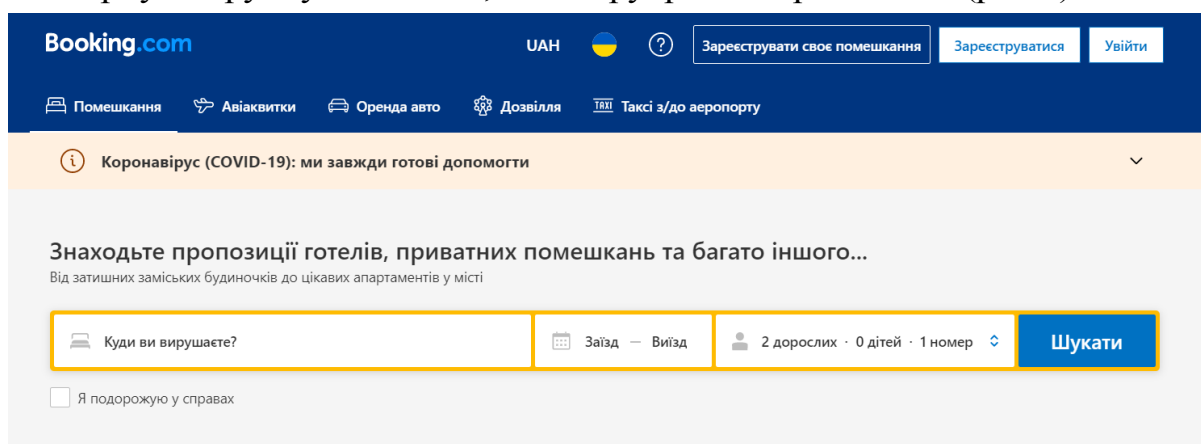
(Рисунок 2. Функціонал сайту OLX)



(Рисунок 3. Результат пошуку квартир на сайті OLX)

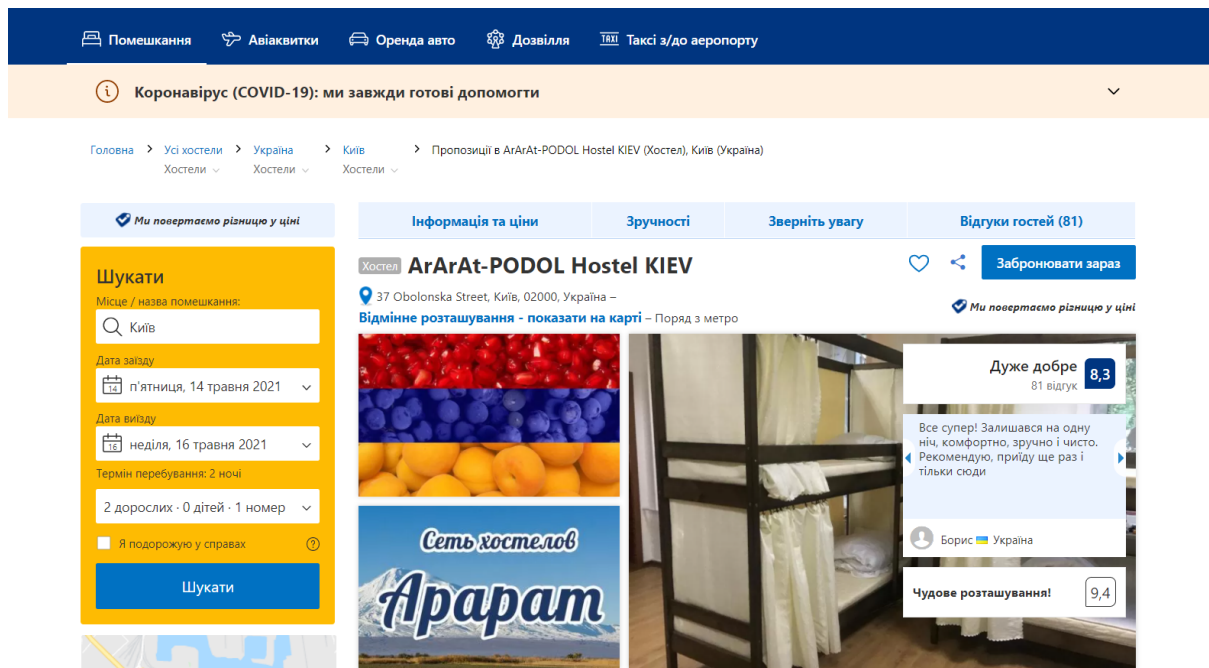
Booking.com

Booking.com - напевно одной з найбільших та найпопулярніших систем бронювання у світі, адже нею користуються майже чи не у всіх країнах. Сайт має гарну та зручну навігацію, безліч рубрик та пропозицій (рис.4).



(Рисунок 4. Головна сторінка сайту Booking.com)

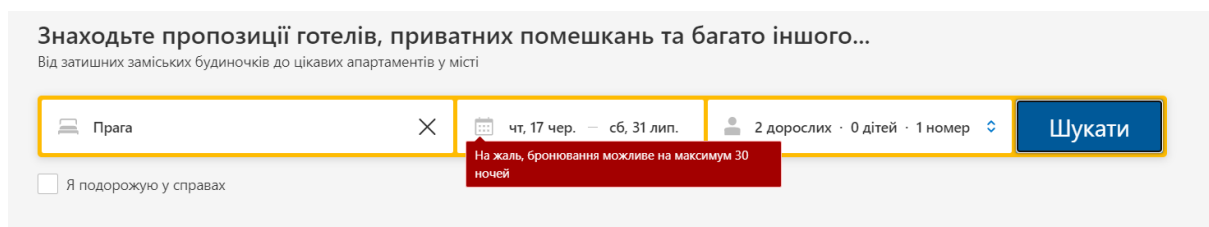
Але на даному сайті є неприємна для користувачів система штрафів за відміну бронювання. Також через величезну популярність, сайт демонструє тобі кількість залишених місць у момент часу твого перегляду (рис. 5) та (рис. 6), що штучно змушує тебе приймати швидкі, необдумані та не найкращі рішення оренди. А також недоліком є те, що сайт базується на короткотривалій оренді (до 30 діб) (рис. 7).



(Рисунок 5. Сторінка бронювання апартаментів)

		базовий тариф		<ul style="list-style-type: none"> Жодних комісій за бронювання або використання кредитної картки!
Представницький двомісний номер з видом на місто.	👤	UAH 11 137	🍳 Сніданок коштує UAH 602 - оцінка Добре	
<p>🔴 Лише 3 номери залишились на нашому сайті</p> <p>1 широке двоспальне ліжко</p> <p>Двомісний номер з кондиціонером, зоною відпочинку та міні-баром.</p> <p>📺 22 кв. м Вид на місто</p> <p>🌿 Кондиціонер 🌿 Власна ванна кімната</p> <p>📺 Телевізор з плоским екраном</p> <p>🔊 Звукоізоляція 🍳 Кавоварка</p> <p>📶 Міні-бар 📶 Безкоштовний Wi-Fi</p> <p>Ще</p>	👤	UAH 11 137	<p>🍳 Сніданок коштує UAH 602 - оцінка Добре</p> <p>✓ БЕЗКОШТОВНЕ скасування до 23:59, 15 червня 2021</p> <p>✓ БЕЗ ПЕРЕДОПЛАТИ – сплачуєте в помешканні</p>	0
	👤	UAH 12 822	<p>🍳 Сніданок включено – оцінка добре</p> <p>✓ БЕЗКОШТОВНЕ скасування до 23:59, 15 червня 2021</p> <p>✓ БЕЗ ПЕРЕДОПЛАТИ – сплачуєте в помешканні</p>	0

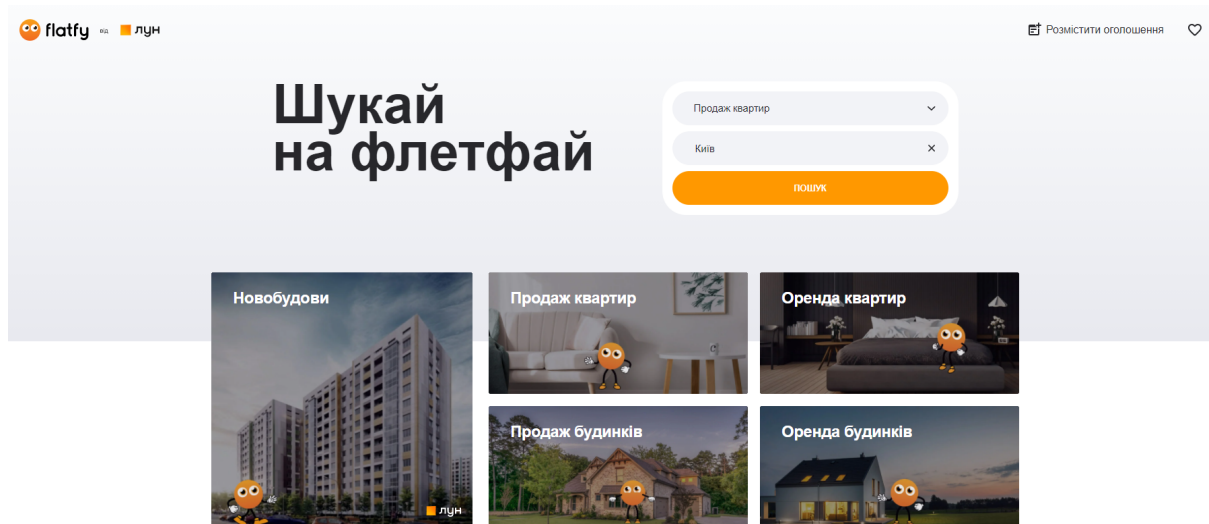
(Рисунок 6. Обмежена кількість місць на сайті Booking.com)



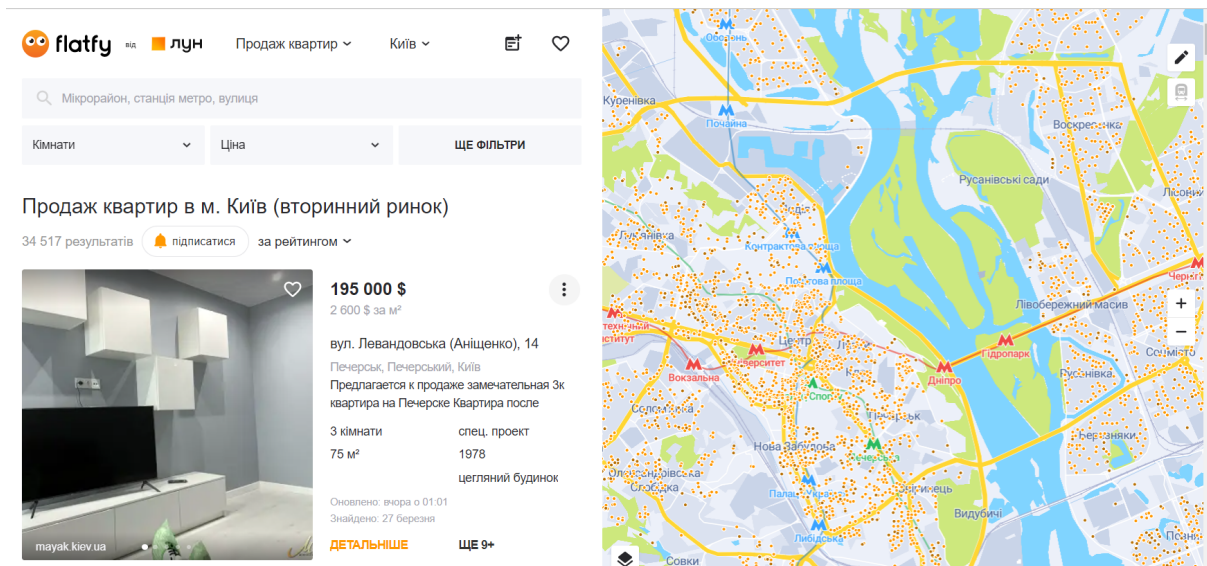
(Рисунок 7. Обмеження бронювання на сайті Booking.com)

Flatfy

Flatfy - це сервіс, який спрямований на роботу лише із нерухомістю, тобто оренда/продаж. Послуги надаються клієнтам із 29 країн. Сайт має гарну репутацію та гарантує якість оголошень. Особливістю є надання геолокації квартир (рис. 9).



(Рисунок 8. Головна сторінка сайту Flatty)



(Рисунок 9. Геолокація розташування квартир із результату пошуку)

Особливих мінусів немає, адже оголошення квартир, за словами творців сайту, перевіряються. Але є один неприємний нюанс - при виборі квартири тебе переправляються на сайт-партнера, де розміщується це оголошення, і трохи неприємно, що при пошуці квартир вас перекидають на десятки різних сайтів.

1.3 Постановка задачі

Основною ідеєю курсової роботи - створити зручну платформу для оренди квартир, яка б була популярною серед молоді та виділялася б на фоні інших сайтів оренди квартир завдяки своєму режиму Badoo Search.

Основним завданням курсової роботи є:

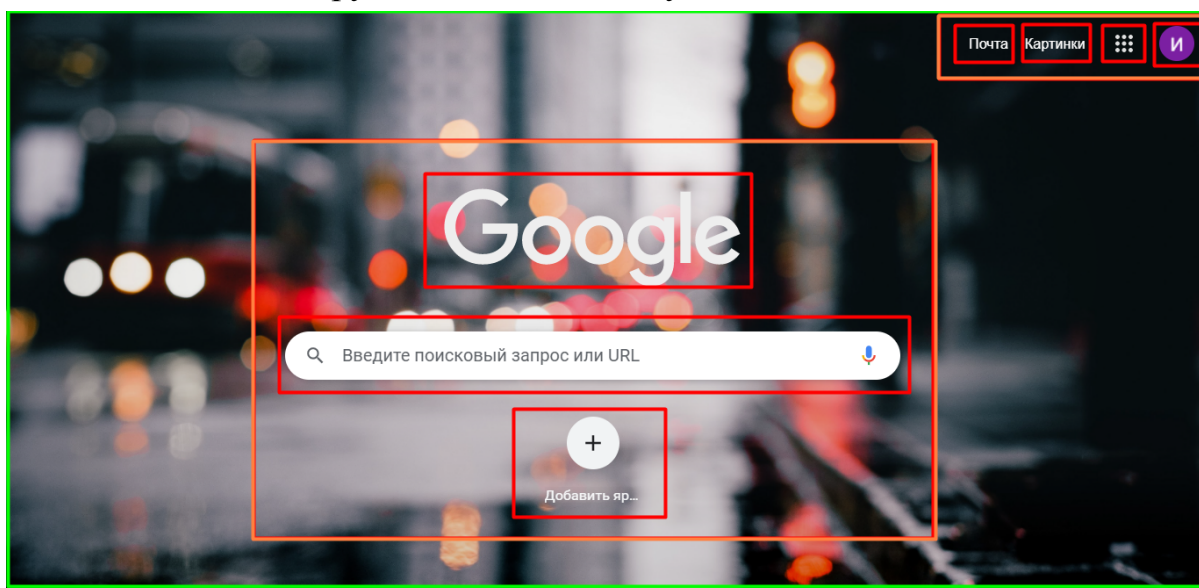
1. Дослідження найбільших сайтів на ринку оренди квартири.
2. Вибір підходящих технологій для створення сайту аналога.
3. Розробка веб-сервісу для пошуку квартир у місті Києві, який буде виконувати наступне:
 - a. авторизація на сайті;
 - b. реєстрація на сайті;
 - c. перегляд всіх квартир;
 - d. фільтрування квартир залежно від обраних функцій;
 - e. перехід на квартиру;
 - f. видалення квартири;
 - g. створення квартири;
 - h. використання пошуку у режимі Badoo Search;
 - i. лайкнути чи дізлайкнути квартиру;
 - j. вийти з акаунту.

Розділ 2 Теоретичні відомості

Основні поняття про популярну бібліотеку для розробки веб-сервісів React

React - це JavaScript бібліотека, яка розробляється компанією Facebook з 2013 року. На сьогоднішній день найактуальнішою версією бібліотеки є версія 17.0.2. Ця бібліотека створена для вирішення проблем із частковим завантаженням певної частини односторінкового застосунку а також для створення великих веб-проектів. Основна ідея - це розбиття всієї сторінки на невеликі блоки, які зветься компонентами.

На рисунку зображена головна сторінка Google.com (рис. 10), яка розбита на маленькі компоненти, що разом формують більший компонент, а більші компоненти формують цілу сторінку. Тому головною задачею є розбиття сторінки на великі компоненти, а ці компоненти на менші компоненти і так далі, поки не дійдемо до зручного для нас стану.



(Рисунок 10. Головна сторінка Google Chrome разбита на компоненти)

Як правило, компоненти пишуться на JSX (комбінація JavaScript та XML), що за своєю структурою схоже на HTML, але має ряд відмінностей. Хоча JSX і був створеним спеціально для роботи з React, часто його опускають і пишуть на стандартному JavaScript.

Зручний підхід у створенні компонентів полягає у варіації їх створення, бо якщо ви прийшли із об'єктно орієнтованих мов програмування, то вам буде зручно створювати компоненти через класи. Якщо ж ви працювали з функціональним програмуванням - то з легкістю можете створювати функціональні компоненти.

Головною особливістю React є використання віртуального DOM, що забезпечує більшу швидкість обробки елементів, ніж пряма робота з DOM.

Сьогодні React складає конкуренцію таким фреймворкам як Angular та Vue.

Головний патерн архітектурної розробки складних програм на React

Хоч React - це мега зручна бібліотека для розробки веб-сервісів, але зі зростанням проекту виникають складнощі із контролем поточного стану, ускладнюється слідкування за компонентами, які рендеряться та які не рендеряться, ускладнюється відслідковування даних, які змінюють поточний стан, написання unit tests та debugging.

До 2015 року Facebook використовував архітектуру MVC.

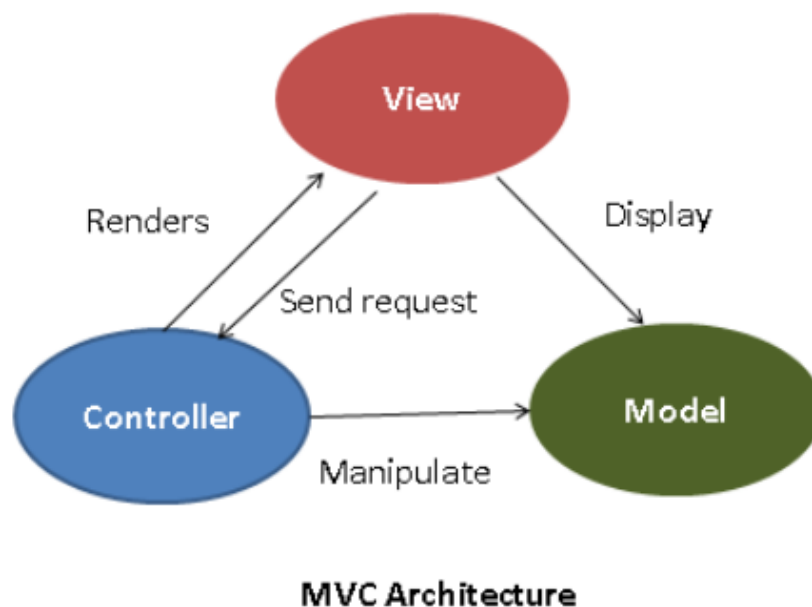
MVC - це одна з найвідоміших та перевірених архітектур проектування програм, адже її концепція з'явилася ще в 1978 році. MVC має в собі 3 компоненти (рис.12):

- Model;
- View;
- Controller.

Model - підтримує дані та поведінку програми.

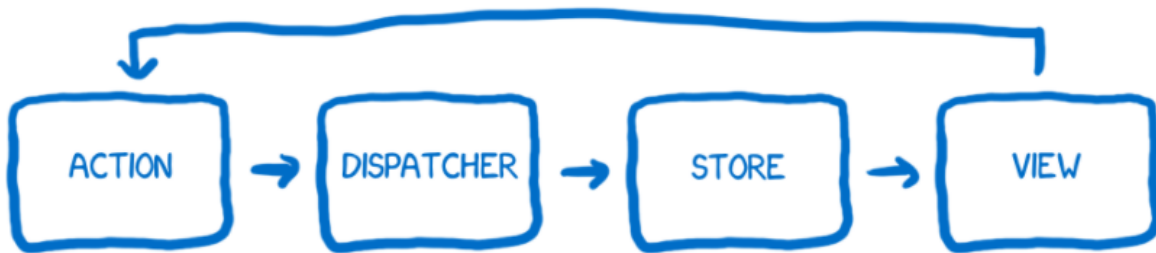
View - відображає компонент в інтерфейсі.

Controller - це інтерфейс між Model та View.



(Рисунок 12. Архітектура MVC)

Хоч MVC і прекрасно себе показувала на серверному рівні, але на клієнтській частині були проблеми, що ускладнювали масштабування системи. Тому у 2015 році компанія Facebook розробила та перейшла на нову архітектуру проектування програм Flux. Основною відмінністю Flux від попередньої системи проектування MVC стала одностороння направленість передачі даних між задіяними компонентами (рис. 11).



(Рисунок 11. Архітектура Flux, де дані передаються в одному напрямку)

У Flux ми бачимо 4 блоки передачі даних:

- action;
- dispatch;
- store
- view

Action - це блок, що генерує дії і через нього проходять всі зміни та взаємодії з системою. Якщо у вас змінився стан застосунку чи вид - потрібно створити action. Action передає всі дані до Dispatch.

Dispatch - це великий реєстр функцій, який отримує всі дії та передає їх до потрібного store. Всі ці операції відбуваються синхронно.

Store - це сховище, містить стан застосунку і вся логіка зміни стану всередині лежить всередині.

View - це процес рендеринга, відмальовки стану для користувача, а також відповідає за прийом інформації, яку вводить користувач.

На даний момент на архітектурі Flux було розроблено бібліотеку Redux, яку розробники широко використовуються у своїх проектах.

JWT (JSON Web Token)

Варто розділяти авторизацію та аутентифікацію.

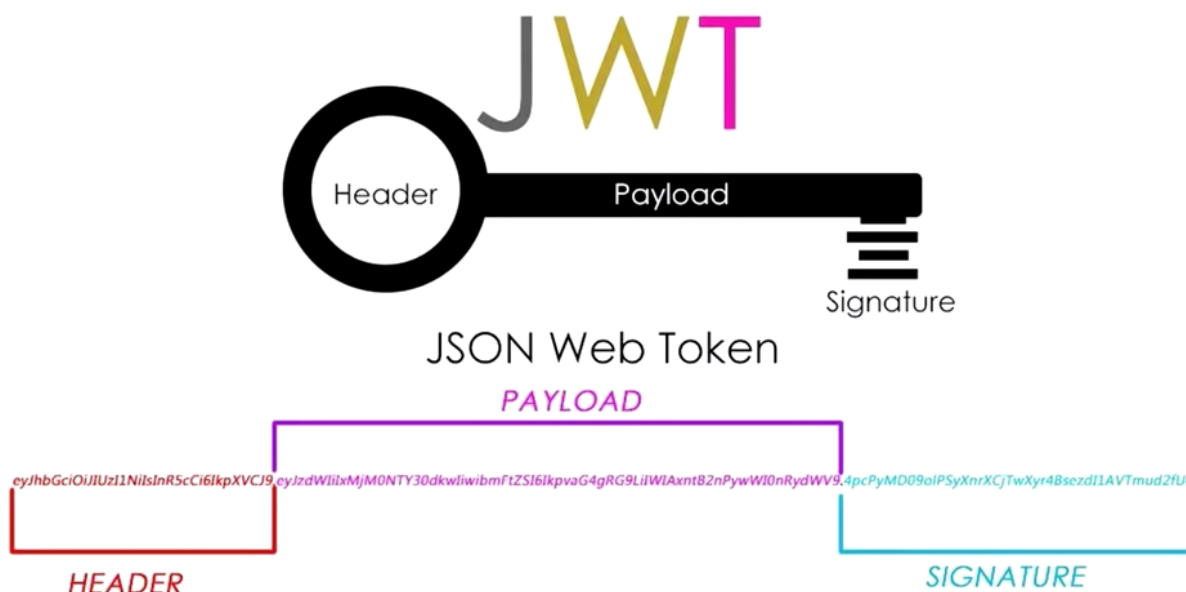
Аутентифікація - це процес перевірки введених даних користувача (найчастіше логін та пароль), для підтвердження справжності особи. Дані

порівнюються порівнянням введених даних особою та даних, що зберігаються у БД.

Авторизація - це перевірка прав чи ролі користувача, задля подальшого доступу до певних ресурсів. Гарним прикладом є реєстрація на AWS, де ви є супер користувачем, і можете додати інших користувачів з певними ролями та особливостями. Наприклад користувачу1 ви надали можливість видаляти дані, а користувачу2 - лише їх переглядати.

JSON Web Token (JWT) - це JSON об'єкт, який визначений за стандартом RFC 7519. Він вважається безпечним способом у передачі персональних даних між клієнтською та серверною частиною застосунку. JWT виглядає як звичайна стрічка, що розділена 3 крапками, і ці крапки розбивають токен на 3 частини (рис. 13):

- header
- payload
- signature



(Рисунок 13. Вигляд JWT)

header.payload.signature - ось так виглядає тіло нашого токена,

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlbnR5cCI6IkpXVCJ9.1A9e7PzoZTtUon-L_rfZsJGalyPWy6POyfyhpkWJoo - це приклад JWT при авторизації користувача на сайті. А ось як розшифровується токен, за допомогою декодера токенів (рис. 14).

Encoded

PASTE A TOKEN HERE

```
:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e  
yJpZCI6MTAsInJvbGVJZCI6MiwiYW0IjoxNjIx  
MDE5MTczLCJleHAiOjE2MjExMDU1NzN9.1A9e7P  
zoZTtUon-L_rfZsJGalypWy6P0yfyhpkWJod
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD: DATA

```
{  
  "id": 10,  
  "roleId": 2,  
  "iat": 1621019173,  
  "exp": 1621105573  
}
```

(Рисунок 14. Розшифрування токена)

При успішній автентифікації на клієнтську частину повертається токен. Оскільки токен містить дані про користувача та його ролі ми маємо його десь зберегти, найчастіше його зберігаються в `localStorage` чи `Cookies`. Через те, що токен містить персональну інформацію, ми не повинні зберігати токен довго, тому у токена є час життя, найчастіше - це від декількох годин до доби. Це потрібно для того, щоб забезпечити безпеку нашому користувачу. Після цього токен прикріплюється до запитів, які потребують авторизацію.

Розділ 3 Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

Flatty - це зручний молодіжний сервіс для бронювання квартир у місті Києві. Аби виділятися на фоні інших сайтів, Flatty має простий, яскравий та зручний дизайн, який не викликає спантеличення чи нерозуміння. Кожен крок автоматизовано і не потрібно робити зайвих кліків чи переходів на інші сторінки. Застосунок має валідацію, тому навіть якщо користувач введе некоректні дані, Flatty повідомляє вас про це.

У сервісі наявні такі групи користувачів: адміністратор сайту, власник квартири та орендар(зареєстрований користувач). Адміністратор має доступ до всіх квартир та за потреби може їх видалити. Власник квартири має доступ до всіх своїх квартир, може видаляти свої квартири та може створити нову квартиру. Орендар може скористатися режимом Badoo Search і лайкнути ті квартири, що йому сподобалися та переглянути лайкнуті квартири.

Технічні вимоги користувачів до функціоналу веб-сервісу

Не зареєстрований користувач сайту :

- перегляд всіх квартир;
- фільтрація квартир за параметрами;
- перегляд певної квартири;
- автентифікація чи реєстрація як власника квартири чи користувача;
- перегляд сторінки “ПРО НАС”;

РОЛЬ : Власник квартири :

- створення нової квартири;
- перегляд всіх своїх квартир;
- видалення квартири;

Зареєстрований користувач :

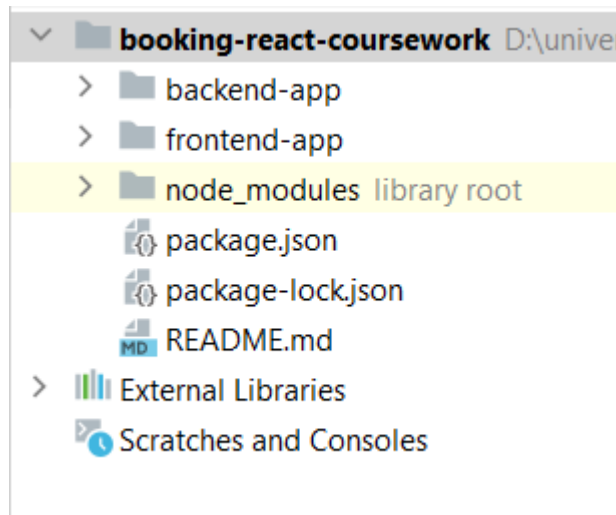
- користування режимом Badoo Search;
- перегляд всіх квартир, які користувач лайкнув;

Адміністратор сайту :

- видалення будь-якої квартири;

3.2 Обґрунтування алгоритму й структури програми

Даний проект має структуру клієнт-серверного застосунку, тому проект розташовується в одній папці, але містить розбиття на окремі частини: клієнтську (frontend-app) та серверну (backend-app) (рис.15).

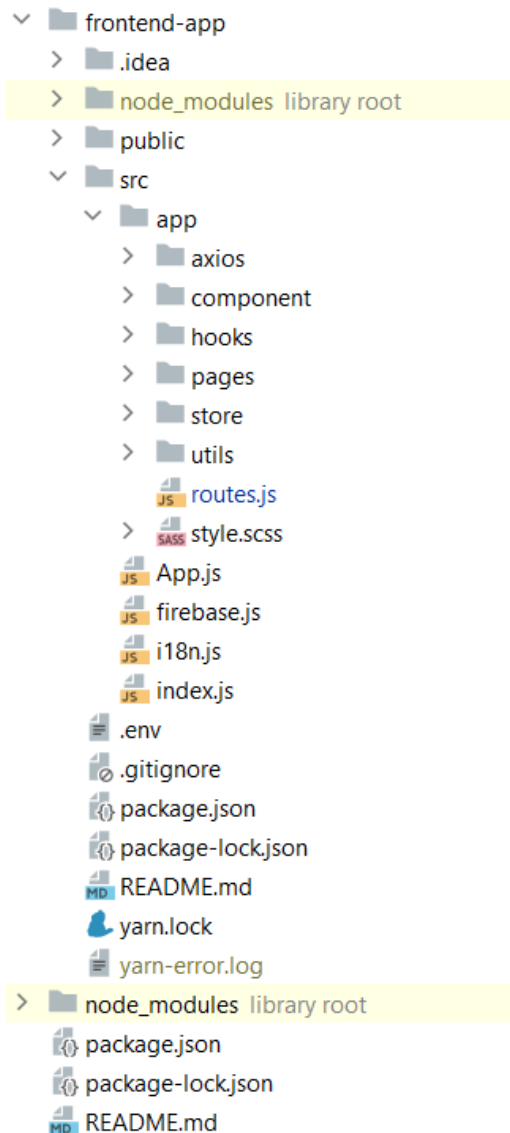


(Рисунок 15. Структура курсової)

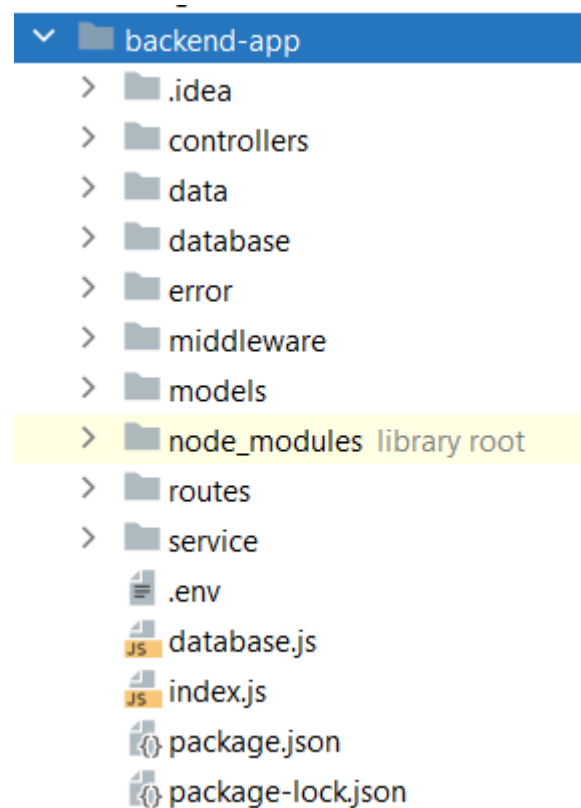
Клієнтська частина реалізований на бібліотеці React, Серверна частина реалізована на Node.js. Веб-сервіс запускається локально за адресою localhost:3000 за допомогою команди npm start, що здійснюється із проекту backend-app та запускає одночасно серверну частину, а потім і клієнтську.

Структура клієнтської частини (рис. 16)

Структура серверної частини(рис.17).



(Рисунок 16. Структура клієнтської частини)



(Рисунок 17. Структура серверної частини)

3.3 Обґрунтування вибору засобів розробки

Одним із найцікавіших і одночасно найвідповідальніших моментів розробки проекту є правильний вибір технологій, адже якщо обрати найсучасніші технології можна втрапити в халепу через невирішені баги, а якщо ж вибрати перевірені, але застарілі технології - можна витратити купу часу на доробку потрібного функціоналу.

Веб-сервіс Flatty складається із 3 частин:

- клієнт
- сервер
- БД

При виборі клієнтської частини, я розглядав Angular, Vue та React. Кожен з них має свої переваги та недоліки. Наприклад Angular був моїм фаворитом, тому що рік назад я на ньому вже писав тестовий проект і відносно розумів як та що відбувається, але через брак досвіду та складність самого фреймворку я вирішив не обирати його. Vue як фреймворк мені сподобався - зручний і доволі простий в освоєнні, він здався мені гарним рішенням для моєї курсової, але проаналізувавши ринок вакансій та проектів, які пишуть на Vue, я мав сумніви, тому що він займає менше 1% у той час як Angular 21% та React 78%. Через це основною причиною відмови від цієї технології була її мала поширеність. Враховуючи всі вищевказані фактори, я зупинився на новій для себе технології React.

З вибором серверної частини все було не так очевидно. Тому що досвіду із розробки серверної частини у мене ще ніколи не було. А у команді, де ми виконували проект, я завжди працював із частиною клієнта чи БД. Після довгого гугління я зупинився на 3 варіантах:

- Java Spring Boot
- Node.js
- Django

Java Spring Boot - це бекенд-фреймворк, що створений на Java. І цей фактор мені найбільше сподобався, тому що Java строго типізована мова і в КМА студенти спеціальності інженерія програмного забезпечення працювали з нею перших 2 роки.

Django - це бекенд-фреймворк на Python, що зараз користується величезним попитом на ринку. Я не мав досвіду роботи на Python, тому вирішив не обирати цей фреймворк для курсової роботи через брак досвіду.

Node.js - це бекенд-фреймворк на JavaScript. У мене не було досвіду із Node.js, але переваги цього фреймворку мене переконали:

- легкість обміну даними з клієнтською частиною;
- клієнтська та серверна частина писатимуться на одній мові - JavaScript, і це пришвидшить роботу, тому що не буде потреб переключатися між 2 різними мовами програмування;
- підтримка тисячі різноманітних бібліотек, що спрощують написання коду;
- велике ком'юніті.

База даних

Стосовно баз даних мій вибір зупинився на 2 реляційних БД та 1 не реляційній БД;

- PostgreSQL;
- MySQL;
- MongoDB.

Всі три БД гарно підходили для цієї курсової, але я все ж вирішив обрати MySQL, тому що вона одна з найвідоміших та найпровіреніших БД, до того ж у мене вже був досвід роботи з нею.

Збереження фотографій квартир я вирішив зберігати на Firebase, тому що він безкоштовний, зручний у використанні, безпечний та швидкий при запитах.

3.4 Опис розробки програми

- 1) Створення БД MySQL;
- 2) Створення дизайну;
- 3) Створення загального проекту для клієнтської та серверної частини;
- 4) Створення таблиць в БД, використовуючи sequelize;
- 5) Заповнення БД тестовими даними;
- 6) Створення роутів, контролерів та сервісів у серверній частині;
- 7) Написання get, post запитів;
- 8) Реалізація Navbar та інших компонентів;
- 9) Додавання сторінок “Головна”, “Про Нас”, “Квартири”, “Квартира”, “Логін”, “Реєстрація”;
- 10) Підключення валідації до форм реєстрації та логіну;
- 11) Розробка двомовності сайту;
- 12) Додавання Redux;
- 13) Виправлення помилок у проектування БД;
- 14) Додавання фільтру та створення квартири;
- 15) Додавання Firebase для збереження фотографій квартир;
- 16) Реалізація Badoo Search;
- 17) Налаштування та виправлення багів.

3.5 Створення об’єктів і розробка головної програми

Створення об’єктів відбувалося за допомогою ORM-бібліотеки sequelize, яку ми встановили командою `npm install sequelize`. Всі об’єкти створюються та додаються до БД при запуску серверної частини. У

програмі наявно 34 об'єкти, тобто кожну табличку із БД (рис. А.1.) ми перетворили в об'єкт і можемо його використовувати. Найпопулярнішими об'єктами є:

- Account;
- Flat;
- House.

3.6 Опис файлів даних та інтерфейсу програми

Flatty - працює з БД MySQL. Схема БД системи наведено в Додатку А на рис. А.1.) На рисунку вище зображена діаграма БД ті її залежності.

Опис таблиць та їх компонентів:

1. **“accounts”** - таблиця містить загальні дані про користувачів сайту. Первинний ключ id. Альтернативні ключі email. Зовнішні ключ roleId .

Атрибути :

id	- унікальний номер аккаунта у БД;
first_name	- ім'я користувача;
second_name	- по-батькові користувача;
birth	- дата народження ;
gender	- стать ;
email	- пошта;
password	- пароль (у БД записується зашифрований пароль через хеш-функцію);
phone_num1	- номер телефону користувача;
phone_num2	- додатковий номер телефону, необов'язковий;
phone_num3	- додатковий номер телефону, необов'язковий;;
last_name	- прізвище користувача;
roleId	- зовнішній ключ, що вказує на номер ролі користувача;

2. **“areas”** - таблиця, що містить всі райони міста. Первинний ключ id. Зовнішні ключ regionId.

Атрибути :

id	- унікальний номер району у БД;
crime	- рівень кримінальності району;
pollution	- рівень забруднення району;
name	- назва району;

regionId - зовнішній ключ, що посиляється на номер області(регіону);

3. **“bad_habits”** - таблиця, що містить всі погані звички.

Атрибути :

id - унікальний номер поганої звички у БД;

name - назва поганої звички;

4. **“bathrooms_types”** - таблиця, що містить всі типи ванних кімнат, які є у квартир.

Атрибути :

id - унікальний номер типу ванної кімнати у БД;

name - назва типу ванної кімнати;

5. **“cities”** - таблиця, що містить всі міста.

Атрибути :

id - унікальний номер міста у БД;

name - назва міста;

6. **“client_has_bad_habits”** - таблиця, що містить всі погані звички користувачів із роллю орендар.Складений ключ, що складається із двох зовнішніх ключів accountId, badHabitId.

Атрибути :

accountId - зовнішній ключ посилання на профіль користувача(орендара);

badHabitId - зовнішній ключ посилання на погану звичку;

7. **“client_has_seen_flats”** - таблиця, що містить всі квартири, які вже переглядав користувач із роллю орендар.Складений ключ, що складається із двох зовнішніх ключів accountId, flatId.

Атрибути :

accountId - зовнішній ключ посилання на профіль користувача;

flatId - зовнішній ключ посилання на квартиру;

8. **“client_liked_flats”** - таблиця, що містить всі пролайкані номери квартир, які пролайкав користувач із роллю орендар.Складений ключ, що складається із двох зовнішніх ключів accountId, flatId.

Атрибути :

accountId - зовнішній ключ посилання на профіль користувача;

flatId - зовнішній ключ посилання на квартиру;

9. **“comforts”** - таблиця, що містить всі види комфортів, які наявні у квартирі.

Атрибути :

id - унікальний номер комфорту у БД;

name - назва типу комфорту;

10. “**flat_has_comforts**” - таблиця, що містить всі доступні comfortи у певній квартирі. Складений ключ, що складається із двох зовнішніх ключів flatId, comfortId.

Атрибути :

flatId - зовнішній ключ на квартиру;

comfortId - зовнішній ключ на тип комфорту;

11. “**flat_has_household_appliances**” - таблиця містить всю побутову техніку у квартири. Складений ключ, що складається із двох зовнішніх ключів flatId, householdApplianceId.

Атрибути :

flatId - зовнішній ключ на квартиру;

householdApplianceId - зовнішній ключ на тип побутової техніки;

12. “**flat_has_images**” - таблиця містить всі посилання на квартиру та її фотографії. Складений ключ, що складається із двох зовнішніх ключів flatId, imageId.

Атрибути :

flatId - зовнішній ключ на квартиру;

imageId - зовнішній ключ на картинку, яка описує цю квартиру;

13. “**flat_has_multimedias**” - таблиця містить айді квартири та мультимедіа. Складений ключ, що складається із двох зовнішніх ключів flatId, multimediaId.

Атрибути :

flatId - зовнішній ключ на квартиру;

multimediaId - зовнішній ключ на мультимедію;

14. “**flat_has_people_types**” - таблиця містить айді квартири та тип людей, кому може здаватися ця квартира. Складений ключ, що складається із двох зовнішніх ключів flatId, peopleTypeId.

Атрибути :

flatId - зовнішній ключ на квартиру;

peopleTypeId - зовнішній ключ на тип людей;

15. “**flat_has_rent_types**” - таблиця містить всі типи оренди(довготривала оренда чи поденна). Складений ключ, що складається із двох зовнішніх ключів flatId, rentTypeId.

Атрибути :

- flatId** - зовнішній ключ на квартиру;
- rentTypeId** - зовнішній ключ на тип оренди квартир;

16. **“flat_has_rules”** - таблиця містить всі правила певної квартири. Складений ключ, що складається із двох зовнішніх ключів flatId, ruleId.

Атрибути :

- flatId** - зовнішній ключ на квартиру;
- ruleId** - зовнішній ключ на правило;

17. **“flats”** - таблиця має в собі всю інформацію про квартиру. Первинний ключ id. Зовнішні ключ houseId, bathroomTypeId.

Атрибути :

- id** - унікальний номер квартири у БД;
- flat_floor** - поверх квартири;
- square_all** - загальна площа квартири;
- square_living** - площа для проживання;
- price_month** - ціна за місяць оренди;
- rooms_num** - кількість кімнат;
- balconies_num** - кількість балконів;
- short_description** - короткий опис квартири;
- main description** - головний опис квартири;
- pledge** - застава за квартиру;
- houseId** - зовнішній ключ на айді будинку, у якому знаходиться квартира;

bathroomTypeId - зовнішній ключ на тип ванної кімнати;

18. **“heatings”** - таблиця вміщає в себе види опалення.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва опалення;

19. **“house_has_infrastructures”** - таблиця містить дані про будинок та інфраструктуру навколо нього. Складений ключ, що складається із двох зовнішніх ключів houseId, infrastructureId.

Атрибути :

- houseId** - зовнішній ключ на айді будинку;
- infrastructureId** - зовнішній ключ на айді інфраструктури;

20. **“house_near_metro_stations”** - таблиця вміщає в себе айді будинку та найближчу станцію метро. Складений ключ, що складається із двох зовнішніх ключів houseId, metroStationId.

Атрибути :

- distance** - дистанція до метро;
- houseId** - зовнішній ключ на айді будинку;
- metroStationId** - зовнішній ключ на станцію метро;

21. **“household_appliances”** - таблиця містить типи побутової техніки.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва побутової техніки;

22. **“houses”** - таблиця має в собі всю інформацію про будинок. Первинний ключ id. Зовнішній ключ streetId, wallTypeId, heatingId.

Атрибути :

- id** - унікальний ключ у БД;
- house_num** - номер будинку;
- house_year** - рік побудови будинку;
- floors_num** - поверхність будинку;
- streetId** - зовнішній ключ на айді вулиці;
- wallTypeId** - зовнішній ключ на айді типу стін у будинку;
- heatingId** - зовнішній ключ на айді типу опалення;

23. **“images”** - таблиця вміщує всі назви картинки.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва картинки, для подальшого пошуку її у firebase;

24. **“infrastructures”** - таблиця вміщує всі види інфраструктури.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва інфраструктури;

25. **“metro_stations”** - таблиця вміщує всі станції метро. Зовнішній ключ cityId.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва станції метро;
- line_color** - колір гілки станції метро;
- cityId** - зовнішній ключ на айді міста;

26. **“multimedias”** - таблиця містить всі види мультимедіа.

Атрибути :

- id** - унікальний ключ у БД;

name - назва мультимедії;

27. **“owner_has_flats”** - таблиця вміщає айді власника квартири та айді квартири. Складений ключ, що складається із двох зовнішніх ключів accountId, flatId.

Атрибути :

accountId - зовнішній ключ на користувача з роллю власника квартир;

flatId - зовнішній ключ на айді квартири;

28. **“people_types”** - таблиця має в собі всі види людей, яким в подальшому буде здаватися квартира.

Атрибути :

id - унікальний ключ у БД;

name - назва типу людей;

29. **“regions”** - таблиця має в собі всі регіони. Зовнішній ключ cityId.

Атрибути :

id - унікальний ключ у БД;

name - назва регіону;

cityId - зовнішній ключ на айді міста;

30. **“rent_types”** - таблиця вміщує всі види оренди квартири.

Атрибути :

id - унікальний ключ у БД;

name - назва типу оренди квартири;

31. **“roles”** - - таблиця вміщує всі види ролей на сайті.

Атрибути :

id - унікальний ключ у БД;

name - назва ролі;

32. **“rules”** - таблиця містить всі види правил для квартири.

Атрибути :

id - унікальний ключ у БД;

name - назва правила;

33. **“streets”** - таблиця містить всі назви вулиць. Зовнішній ключ areaId.

Атрибути :

id - унікальний ключ у БД;

name - назва вулиці;

areaId - зовнішній ключ на район міста;

34. “wall_types” - таблиця має всі види стін будинку.

Атрибути :

- id** - унікальний ключ у БД;
- name** - назва типу стін будинку;

3.7 Тестування програми і результати її виконання

Тестування відбувалося у браузері Google Chrome.

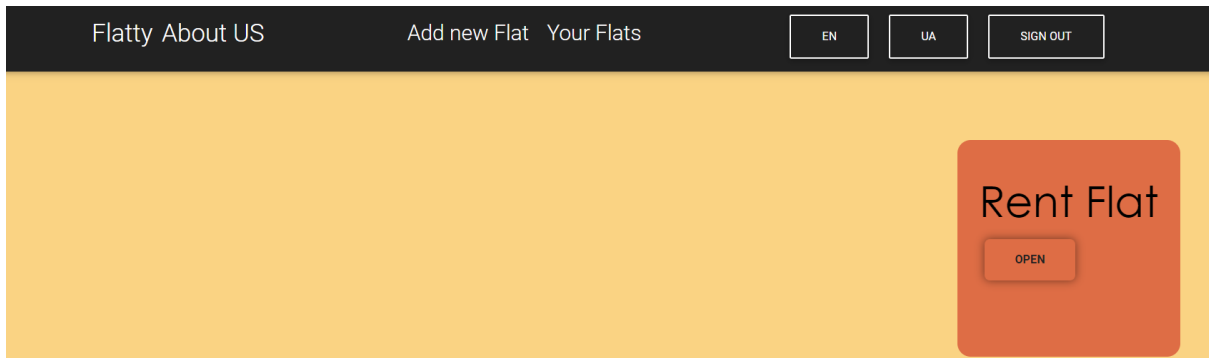
“Головна” - головна сторінка сайту, яка відкривається при вході на сайт чи при спробі ввести некоректну адресу. Сторінка складається з меню, де можна перейти на сторінку “ПРО НАС”(рис. 18), змінити мову (на українську чи англійську)(рис.19), зареєструватися як звичайний користувач чи як власник квартири, а також ряд інструментів залежно від ролі користувача; “Створити квартиру”, “Перегляд ваших квартир” для користувача з роллю власника квартири (рис. 20); “Вподобані квартири”, “Badoo Search” для орендаря (рис. 21). У центральній частині сторінки розташована кнопка ”Перейти”, яка перенаправляє на сторінку всіх квартир.



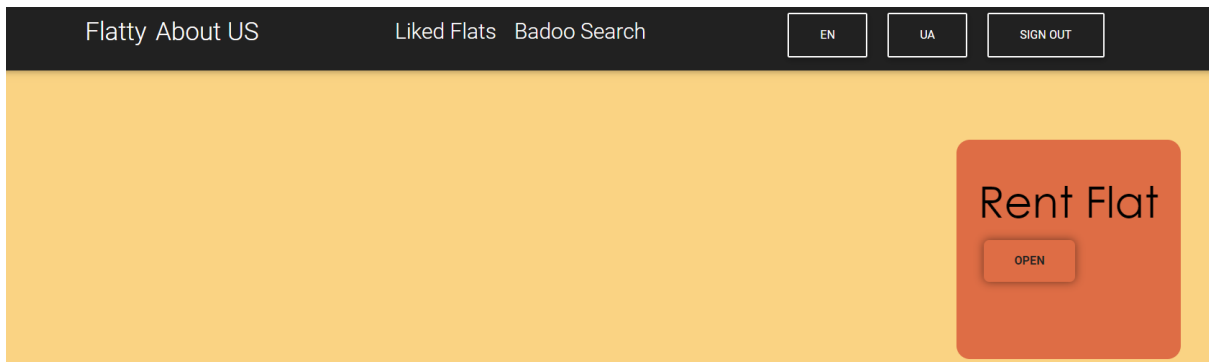
(Рисунок 18. Головна сторінка (англійська мова сайту))



(Рисунок 19. Головна сторінка (українська мова сайту))

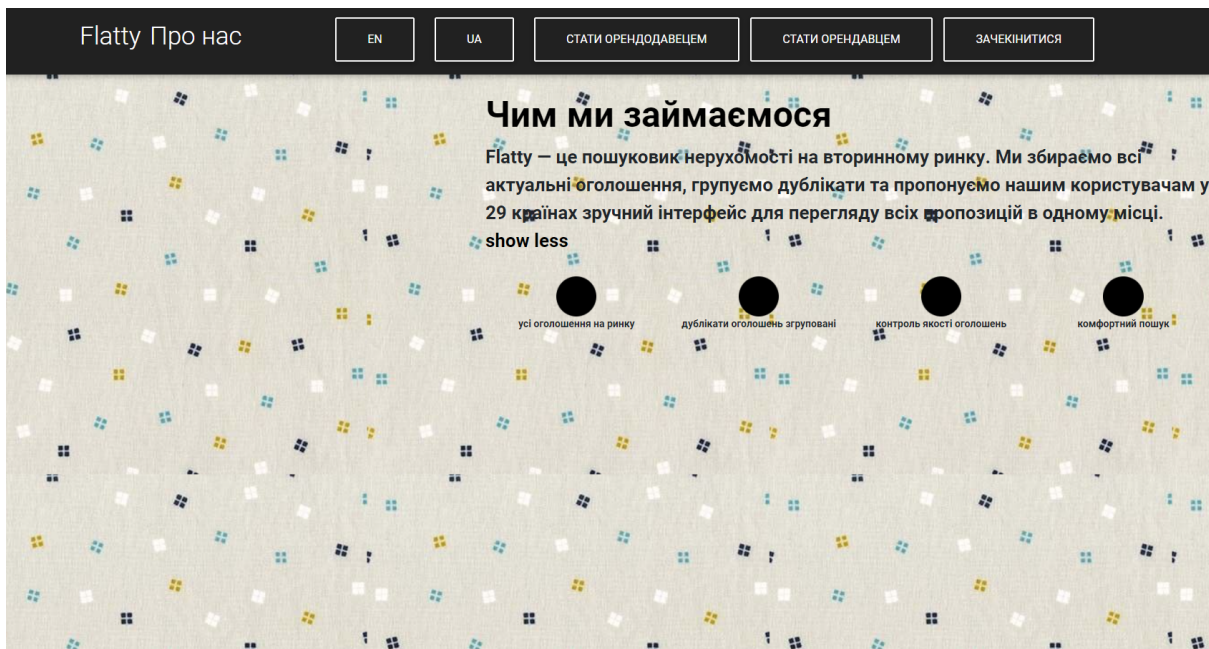


(Рисунок 20. Головна сторінка у користувача з роллю Власника квартири(owner))



(Рисунок 21. Головна сторінка у користувача з роллю орендаря(client))

“Про Нас” - статична сторінка, яка демонструє інформацію Flatty(рис.22). Сторінка “Про Нас” має статичну інформацію, яку можна згорнути чи розгорнути залежно від вашого бажання.



(Рисунок 22. Сторінка ПРО НАС)

“Реєстрації ” - сторінка для реєстрації нового користувача із різними ролями. Якщо ж у вас є акаунт, ви можете перейти на сторінку “Авторизації”. Реєстрація є двох видів - реєстрація нового власника квартири та нового орендаря. Для того щоб зареєструватися як власник квартири, вам потрібно натиснути кнопку “Become Owner” на сторінці “Головна сторінка” після цього вас перенаправить на сторінку реєстрації, де має заповнити всі поля:

- ім'я;
- прізвище;
- по-батькові;
- дата народження;
- стать;
- пошта;
- номер телефону;
- пароль.

Якщо всі поля введено вірно, то стає доступною кнопка “Зареєструватися” і після її натиску відправляється “/post” запит на створення нового власника квартири.

Щоб зареєструватися як користувач, вам потрібно натиснути кнопку “Зареєструватися” і заповнити ідентичну форму, що і при реєстрації власника квартири(рис. 23).

Flatty About US

EN UA BECAME OWNER BECAME RENTER SIGN IN

Registration

Enter your name Enter your surname Enter your father name

05/15/2021 ☐ Woman ☐ Man

Enter your email

Enter your phone

Enter your password

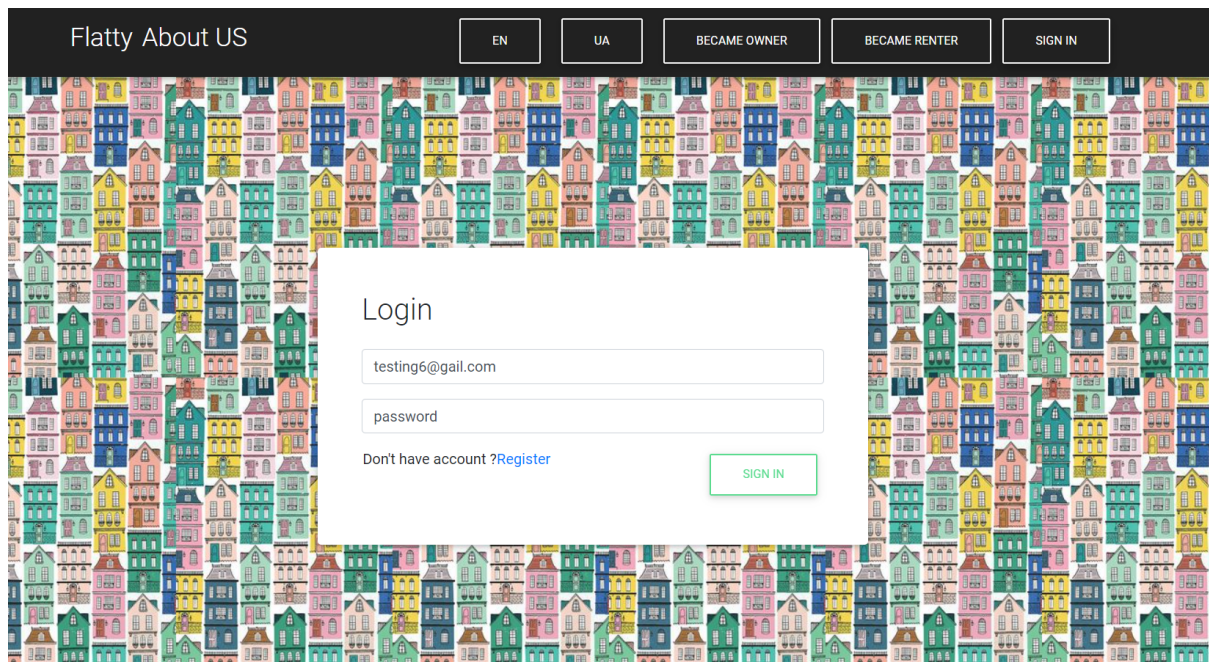
Confirmed your password

Have account ? [Login](#) REGISTER

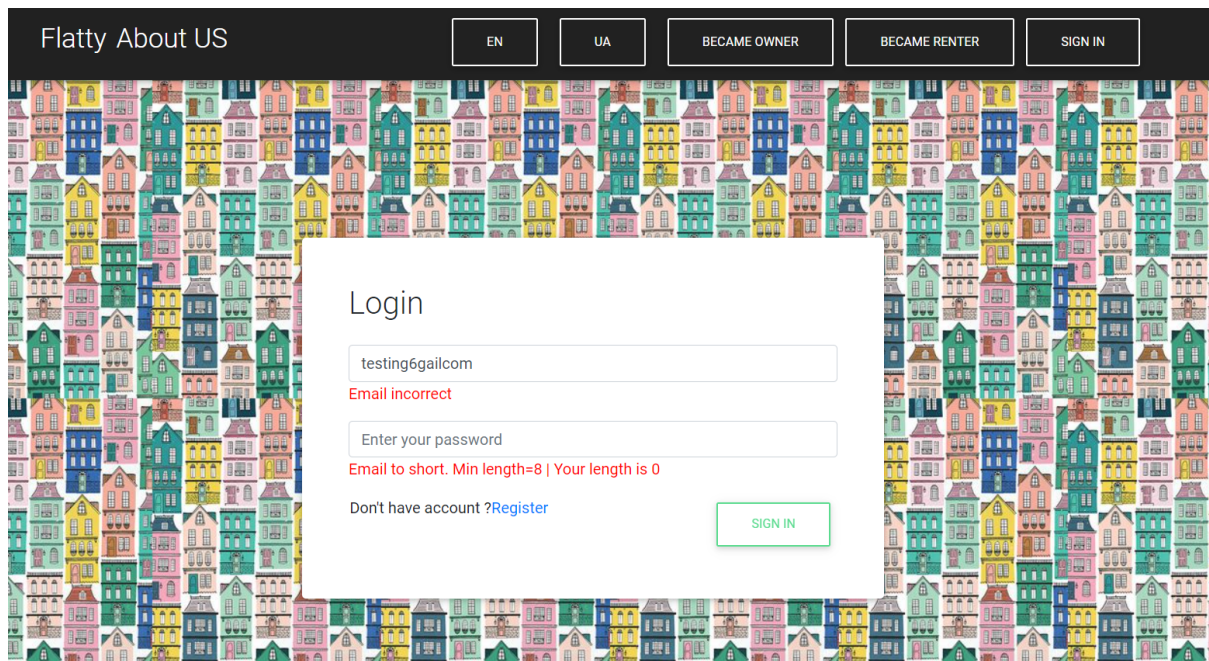
(Рисунок 23. Сторінка Реєстрації)

“Авторизації” - це сторінка для авторизації користувача за поштою та паролем та з можливістю перейти на сторінку реєстрації, якщо ви не зареєстрований користувач (рис. 24).

При натисканні кнопки “SIGN IN” на “Головній сторінці” ви потрапляєте на сторінку “Авторизації”, де маєте заповнити форму авторизації, а саме ввести пошту та пароль, якщо один з введених елементів не проходить валідацію, то кнопка “ВХІД” буде недоступною до натиску та біля некоректних форм з’явиться повідомлення з помилкою, яку користувач допустив у формі вводу (рис.25). Якщо всі поля коректні, ви можете натиснути кнопку “ВХІД”, після цього здійсниться “/post” запит, який спочатку знайде користувача з ідентичною поштою, якщо такого користувача немає, то ви не зайдете, якщо такий користувач є, то порівнюється ваш введений пароль із зашифрованим паролем із БД, якщо паролі ідентичні, то буде створено jwt токен, який міститиме {accountId, roleId}. Після успішної авторизації вас перенаправить на головну сторінку і Navbar додасть ті функції, які доступні користувачам за ролями.



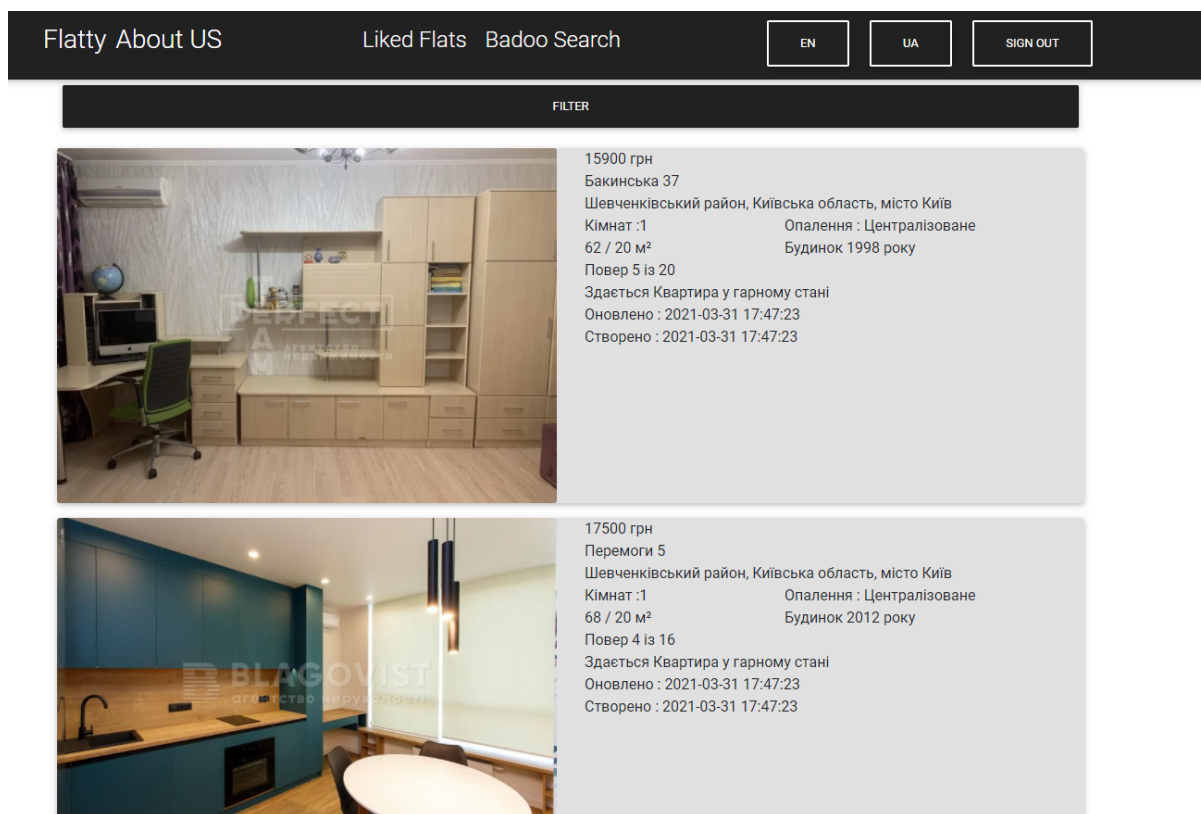
(Рисунок 24. Сторінка Авторизації)



(Рисунок 25. Сторінка Авторизації (приклад валідації))

“Перегляд квартир” - сторінка для перегляду всіх доступних квартир (рис. 26). На цій сторінці працює фільтр, який може з’являтися чи ховатися (рис.27). При натиску на обрану квартиру ви можете перейти на сторінку тієї квартири, яку обрали.

Сторінка всіх квартир, де ви можете здійснити пошук за випаданим фільтром. Пошук квартир здійснюється у стандартному режимі, коли показуються всі квартири і, гортаючи список квартир, можете обрати ту квартиру, яка вам підходить. Або скористатися фільтром. Щоб відкрити фільтр вам потрібно натиснути кнопку “Filter” і перед вами з’явиться фільтр. Ввівши дані, що вас цікавить, вам потрібно натиснути кнопку “Search” і здійсниться пошук квартир за даними характеристиками.



(Рисунок 26. Сторінка Квартири)

Flatty About US EN UA BECAME OWNER BECAME RENTER SIGN IN

FILTER

Price Enter price from Enter price To

Square All Enter square all from Enter square all To

Square Living Enter square living From Enter square living To

Set Area Set Metro Station

Set Flat Rooms Set Flat Floor Set House Floor

Set House Year Set wall Type Set Heating

Виберіть Інфраструктуру in 1 mile around home

☐ Дитячий садок ☐ Школа ☐ Бювет

☐ Зупинка транспорту ☐ Метро ☐ Ринок

☐ Супермаркет, ТРЦ ☐ Парк ☐ Озеро

☐ Дитячий майданчик ☐ Аптека ☐ Лакарня, поліклініка

☐ Центр міста ☐ Ресторан, кафе ☐ Кінотеатр

☐ Театр ☐ Відділення пошти ☐ Відділення банку

☐ Автовокзал ☐ Залізнична станція ☐ Спортзал

☐ Спортивний майданчик ☐ Університет

ПОШУК

(Рисунок 27. Фільтр для пошуку квартир)


“Квартира” - це персональна сторінка квартири, яка містить всю потрібну інформацію про квартиру (рис.28). Ця сторінка надає можливість переглянути всі фотографії квартири та отримати контакти власника квартири, щоб з ним потім зв'язатися. Для користувачів з роллю “АДМІНІСТРАТОР” та “ВЛАСНИК КВАРТИРИ” наявна кнопка видалити, яка видаляє інформацію про квартиру з БД.

Flatty About US
Liked Flats
Badoo Search

EN

UA

SIGN OUT



Здається Квартира у гарному стані
Гоголівська 7
Київська Шевченківський Київ
Кімнати 3, Поверх 3 із 16
Ціна 50000 грн/міс
Площа 110/52 м²
Кімнати 3
Кількість балконів 2
Тип Ванної кімнати Роздільний
Поверхність будинку 16
Тип опалення Централізоване
Вік будинку 2018

М Золоті Ворота

Comfort:

Даних немає

HouseHoldAppliance :

Даних немає

Multimedia :

Даних немає

PeopleType :

Даних немає

Rule :

Даних немає

(Рисунок 28. Сторінка Квартири)

“Створення квартири” - сторінка створення нової квартири власником квартири чи адміністратором (рис.29_1, рис.29_2, рис.29_3). На цій сторінці ви маєте заповнити всі поля для надання повної інформації про квартиру.

Flatty About US
Add new Flat Your Flats

EN
UA
SIGN OUT

Оберіть вулицю ▼

Оберіть Район ▼

houseNum
houseNum

houseYear
houseYear

numOfFloors
numOfFloors

Тип опалення ▼

Оберіть найближчу станцію метро ▼

flatFloor
flatFloor

flat Rooms
roomsNum

balconiesNum
balconiesNum

allSquare
allSquare

livingSquare
livingSquare

Оберіть тип ванної кімнати ▼

priceMonth
priceMonth

pledge
pledge

Оберіть Тип стін у будинку ▼

Вибрати файли

Файл не вибран

short_description

(Рисунок 29_1. Створення квартири)

main_description

Виберіть комфорт

☐ Кондиціонер
☐ Душова кабінка
☐ Лоджія
☐ Панорамні вікна
☐ Пожежна сигналізація
☐ Охорона території
☐ Підземний паркунг
☐ Грузовий ліфт

☐ Підігрів полу
☐ Меблі
☐ Тераса
☐ Грати на вікнах
☐ Відоспостереження
☐ Паркувальне місце
☐ Гараж
☐ Технологія 'розумний будинок'

☐ Ванна
☐ Гардероб
☐ Тераса
☐ Сигналізація
☐ Конс'єрж
☐ Гостьовий паркінг
☐ Ліфт
☐ Автономний електрогенератор

Виберіть Інфраструктуру

☐ Дитячий садок
☐ Зупинка транспорту
☐ Супермаркет, ТРЦ
☐ Дитячий майданчик
☐ Центр міста
☐ Театр
☐ Автовокзал
☐ Спортивний майданчик

☐ Школа
☐ Метро
☐ Парк
☐ Аптека
☐ Ресторан, кафе
☐ Відділення пошти
☐ Залізнична станція
☐ Університет

☐ Бювет
☐ Ринок
☐ Озеро
☐ Лакарня, поліклініка
☐ Кінотеатр
☐ Відділення банку
☐ Спортзал

(Рисунок 29_2. Створення квартири)

Виберіть тип мешканців

☐ Тільки дівчина
☐ Можна студентам

☐ Тільки хлопець
☐ Можна іноземцям

☐ Тільки сім'я
☐ Можна з дітьми

Виберіть Мультимедію в квартирі

☐ Wi-Fi
☐ PlayStation
☐ Проектор

☐ Швидкісний інтернет
☐ Кабельне, цифрове ТБ

☐ Телевізор
☐ Супутникове ТБ

Виберіть Правила проживання

☐ Курити не можна

☐ Тварин не можна

☐ Не можна проводити заходи(вечірки)

Виберіть Побутову техніку

☐ Електрочайник
☐ Плита
☐ Мікрохвильова піч
☐ Морозильна камера
☐ Сушильна машина

☐ кавомашинка
☐ Варильна панель
☐ Мультиварка
☐ Посудомийна машина
☐ Праска

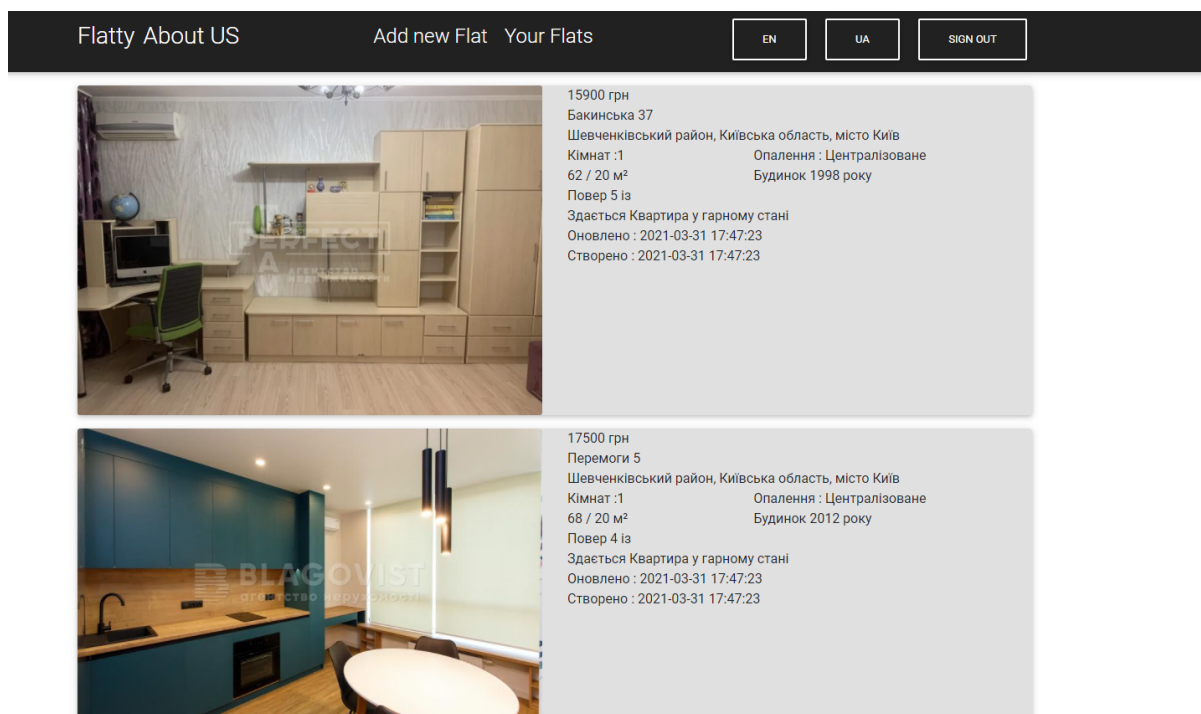
☐ Фен
☐ Духова шафа
☐ Холодильник
☐ Пральна машина
☐ Пилосос

СТВОРИТИ КВАРТИРУ

(Рисунок 29_3. Створення квартири)

“Перегляду всіх квартир власника” - сторінка, що показує всі квартири певного власника. При натиску на квартиру вас перекине на сторінку обраної квартири (рис. 30).

“Вподобані квартир” - сторінка, що показує квартири, які користувач лайкнув на сторінці “Badoo Search” (рис. 31).



(Рисунок 30. Сторінка всіх квартир власника)



(Рисунок 31. Пролайкані квартири)

“Badoo Search” - сторінка, що схожа за функціоналом на сайти знайомств Badoo чи Tinder, яка демонструє основну інформацію про квартиру та її фотографії, а орендар може поставити лайк квартирі чи дізлайк. При натиску на кнопку “Like” користувач додає цю квартиру до свого списку вподобаних квартир. При натиску на кнопку “Dislike” користувач вказує, що дана квартира йому не подобається і більше її у цьому режимі користувач не зустрине.



43000 грн

Мельникова 44

Шевченківський район, Київська область, місто Київ

Кімнат :3

Опалення : Централізоване

110 / 74 м²

Будинок 2018 року

Повер 6 із

Здається Квартира у гарному стані

Оновлено : 2021-03-31 17:47:23

Створено : 2021-03-31 17:47:23

DISLIKE

LIKED

(Рисунок 32. Badoo Search)

Висновки

У результаті виконання курсової роботи було створено веб-сервіс, який допомагає знайти квартиру для оренди у місті Києві. Сайт оренди квартир наділений стандартним функціоналом, що і більшість подібних сайтів. Особливістю сайта є режим Badoo Search, який копіює ідею пошуку квартир у сайтів знайомств таких як: Badoo та Tinder.

Для створення сайту була здійснена робота над аналізом найпопулярніших сайтів оренди квартир та проведено роботу над аналізом плюсів та мінусів цих сайтів.

Проект має багато шансів для подальшого розвитку та вдосконалення, адже БД проектувалася на розширення даної системи, а саме - додавання нових міст, типів оренди, поганих звичок орендарів тощо. Цей сайт є внеском у розробку альтернативним методів пошуку квартир для оренди.

Посилання

1. Booking.com [Електронний ресурс]. Режим доступу: https://www.booking.com/hotel/index.html?aid=1764808&label=lh_goo5wa
2. Flatfy [Електронний ресурс]. Режим доступу: <https://flatfy.ua/uk>
3. OLX [Електронний ресурс]. Режим доступу: <https://www.olx.ua/uk/nedvizhimost/kvartiry-komnaty/arenda-kvartir-komnat/ki-ev/>
4. React hooks [Електронний ресурс]. Режим доступу: <https://ru.reactjs.org/docs/hooks-intro.html>
5. React documentation [Електронний ресурс] <https://ru.reactjs.org/docs/getting-started.html>
6. Flux vs MVC [Електронний ресурс]. Режим доступу: <https://madasamy.medium.com/flux-vs-mvc-design-pattern-de134dfaa12b>
7. Flux [Електронний ресурс]. Режим доступу: <https://facebook.github.io/flux/docs/in-depth-overview>
8. Flux vs MVC vs Redux [Електронний ресурс] <https://www.clariontech.com/blog/mvc-vs-flux-vs-redux-the-real-differences>
9. Дані про JWT [Електронний ресурс]. Режим доступу: <https://gist.github.com/zmts/802dc9c3510d79fd40f9dc38a12bccfc>
10. JWT decoder [Електронний ресурс] <https://jwt.io/>
11. Переваги та недоліки фреймворків [Електронний ресурс]. Режим доступу: <https://medium.com/nuances-of-programming/reactjs-angular5-%D0%B8-vue-js-%D0%BA%D0%B0%D0%BA%D0%BE%D0%B9-%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA-%D0%B2%D1%8B%D0%B1%D1%80%D0%B0%D1%82%D1%8C-%D0%B2-2018-%D0%B3%D0%BE%D0%B4%D1%83-101702e51043>
12. Сильні сторони різних бекенд-фреймворків [Електронний ресурс]. Режим доступу: <https://habr.com/ru/company/ruvds/blog/519478/>
13. Spring documentation [Електронний ресурс] <https://docs.spring.io/spring-framework/docs/current/reference/html/>
14. Django documentation [Електронний ресурс] <https://docs.djangoproject.com/en/3.2/>
15. Node.js documentation [Електронний ресурс] <https://nodejs.org/uk/docs/>
16. React components [Електронний ресурс]. Режим доступу: [https://www.w3schools.com/react/react_components.asp#:~:text=Components%](https://www.w3schools.com/react/react_components.asp#:~:text=Components%20)

[20are%20independent%20and%20reusable,will%20concentrate%20on%20Classes%20components.](#)

17. Node.js MySQL JOIN [Электронный ресурс]. Режим доступа:

https://www.w3schools.com/nodejs/nodejs_mysql_join.asp

18. React bootstrap dropdown with checkbox [Электронный ресурс]. Режим доступа: <https://react-bootstrap.github.io/components/dropdowns/>

19. Map in React [Электронный ресурс]. Режим доступа:

<https://ru.reactjs.org/docs/lists-and-keys.html>

20. Lucid for database [Электронный ресурс]. Режим доступа:

<https://lucid.app/documents#/dashboard>

21. Sequelize in Node.js [Электронный ресурс]. Режим доступа:

<https://bezkoder.com/sequelize-associate-many-to-many/>

Додатки

Додаток А

