

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка Web-платформи онлайн оголошень на базі
фреймворку React»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Олександр ЩЕРБАЧЕНКО
(підпис)

Виконав: здобувач вищої освіти групи ПД-42

Олександр ЩЕРБАЧЕНКО

Керівник: _____ Оксана ЗОЛОТУХІНА
к.т.н., доцент

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Щербаченку Олександрю Сергійовичу _____

1. Тема кваліфікаційної роботи: «Розробка Web-платформи онлайн оголошень на базі фреймворку React»

керівник кваліфікаційної роботи к.т.н., доц., доцент кафедри ІІЗ Оксана ЗОЛОТУХІНА,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про веб-платформи онлайн оголошень та сучасні стеки технологій.

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

1. Огляд та аналіз існуючих веб-платформ онлайн оголошень .

2. Проектування й визначення архітектури веб-платформи".

3. Програмна реалізація та опис функціонування розробленої веб-платформи онлайн оголошень.

4.Тестування веб-платформи.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Алгоритм роботи застосунку.
6. Діаграма класів.
7. Екранні форми.
8. Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Аналіз та вибір інструментів для розробки веб-платформи	14.03-17.03.2024	
4	Проектування архітектури веб-платформи	18.03-24.03.2024	
5	Розробка веб-платформи онлайн оголошень	24.03-23.04.2024	
6	Тестування веб-платформи онлайн оголошень	24.04.-29.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	30.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

(підпис)

Олександр ЩЕРБАЧЕНКО

Керівник
кваліфікаційної роботи

(підпис)

Оксана ЗОЛОТУХІНА

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 62 стор., 5 табл., 34 рис., 19 джерел.

Мета роботи – спрощення роботи з онлайн оголошеннями за рахунок використання web-платформи на базі фреймворку React.

Об'єкт дослідження – процес розміщення та роботи з онлайн оголошеннями.

Предмет дослідження web-платформа для роботи з онлайн оголошеннями.

Короткий зміст роботи: В роботі проаналізовано архітектуру MVC та технології веб-розробки, що застосовуються для створення онлайн платформ для оголошень з нерухомості. Проаналізовано інструментальні засоби та фреймворки, зокрема MongoDB, Express.js, React та Node.js. Розроблено веб-платформу для оголошень з нерухомості, програмно реалізовані ключові функціональні можливості, зокрема: реєстрація та авторизація користувачів, створення, редагування та завантаження зображень для оголошень, фільтрація оголошень за різними параметрами, сортування результатів пошуку, можливість залишати коментарі до оголошень. Проведено функціональне та API тестування додатку. В роботі використано JWT для забезпечення безпеки та авторизації користувачів, Tailwind CSS для створення адаптивного інтерфейсу, а також Postman для тестування API.

Сферою використання застосунку є розробка веб-платформи онлайн-оголошень.

КЛЮЧОВІ СЛОВА: МОДЕЛЬ,АРХІТЕКТУРА, MVC , ВЕБ-ПЛАТФОРМА.

ЗМІСТ

ВСТУП	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1. Поняття Web-платформи онлайн оголошень	11
1.2. Актуальність веб платформи онлайн оголошень	12
1.3. Огляд та аналіз web-платформ онлайн оголошень	14
1.3.1. OLX.....	14
1.3.2. Obyava.ua.....	17
1.3.3. Ogolosha.ua.....	20
1.4. Запозичення досвіду аналогів при створенні свого проекту	21
1.5. Масштабування.....	23
2. ОПИС ТЕХНОЛОГІЙ ТА МОВ ПРОГРАМУВАННЯ ДЛЯ РОЗРОБКИ	25
2.1. Моделювання вимог до веб платформи	25
2.2. Опис архітектури MVC	27
2.2.1. Модель в архітектурі MVC	29
2.2.2. Вид в архітектурі MVC.....	30
2.2.3. Контролер в архітектурі MVC	32
2.2.4. Переваги архітектури MVC	34
2.2.5. Недоліки архітектури MVC	35
2.3. Мова програмування JavaScript	36
2.4. Node.js	38
2.5. Express.js	39
2.6. MongoDB	40

2.7.	Tailwind CSS.....	41
2.8.	JSON Web Token (JWT)	42
2.9.	Vite.....	44
2.10.	Postman для тестування API	45
3.	РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОСТІ ВЕБ-ПЛАТФОРМИ.....	47
3.1	Розробка ключових функцій веб-сервісу для оголошень нерухомості....	47
3.1.1.	Реєстрація та авторизація користувача	48
3.1.2.	Головна сторінка веб-платформи	51
3.1.3	Сторінка оголошення для зареєстрованого користувача.....	54
3.1.4	Сторінки створення та оновлення оголошення	57
3.2.	Розробка структури бази даних та моделей.....	62
4.	ТЕСТУВАННЯ ВЕБ ПЛАТФОРМИ.....	66
4.1.	Визначення об'єктів тестування веб-сервісу	66
4.2.	Створення тест-кейсів	67
	ВИСНОВКИ.....	70
	ПЕРЕЛІК ПОСИЛАНЬ.....	72
	ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	74

ВСТУП

З поширенням глобальної комп'ютерної мережі Інтернет, веб-розробка еволюціонувала значно. Спочатку програмісти створювали окремі додатки для різних операційних систем, які потрібно було інсталиювати локально на комп'ютері користувача. Такі програми називалися десктопними додатками. Проте з появою веб-додатків, користувачам стало набагато зручніше, оскільки вони могли отримати доступ до додатків безпосередньо через веб-браузер, незалежно від операційної системи.

Мета роботи – спрощення роботи з онлайн оголошеннями за рахунок використання веб-платформи на базі фреймворку React.

Об'єкт дослідження – процес розміщення та роботи з онлайн оголошеннями.

Предмет дослідження веб-платформа для роботи з онлайн оголошеннями

З розвитком веб-додатків з'явилися сучасні, інтерактивні веб-додатки, які дозволяють користувачам взаємодіяти з сервером, слухати музику, дивитися відео та навіть малювати на екрані. Веб-розробка охоплює як фронтенд, так і бекенд аспекти. Раніше для повноцінної веб-розробки використовувався стек LAMP (Linux, Apache, MySQL і PHP), де PHP використовувався для бекенду, а HTML, CSS і JavaScript — для фронтенду, а MySQL — для баз даних. Однак з появою Node.js, який дозволяє використовувати JavaScript як на фронтенді, так і на бекенді, розробники отримали можливість використовувати одну мову програмування для всього стеку. Node.js став популярною технологією для розробки бекенд-систем, а також існує багато фреймворків для фронтенду, таких як Angular, React, Vue.js.

Розробка веб-платформ для онлайн оголошень є важливим аспектом сучасного світу, де технології відіграють ключову роль у нашому повсякденному житті. Веб-платформи дозволяють користувачам легко створювати та переглядати оголошення, спілкуватися та обмінюватися інформацією. Один з таких прикладів є розробка веб-платформи на базі фреймворку React, який забезпечує створення динамічних та інтерактивних інтерфейсів користувача. Використання сучасного

стека MERN (MongoDB, Express.js, React, Node.js) дозволяє створити повноцінний веб-додаток з використанням єдиної мови програмування на всіх етапах розробки.

забезпечує високу продуктивність, масштабованість та зручність використання. Вона дозволяє створювати інтерактивні інтерфейси, які забезпечують швидку та ефективну взаємодію з користувачем. Використання MongoDB для зберігання даних, Express.js для обробки запитів та Node.js для серверної частини додатка дозволяє створити надійну та гнучку систему, яка може легко адаптуватися до потреб користувачів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття Web-платформи онлайн оголошень

Оголошення являють собою короткі повідомлення, що містять різноманітну інформацію. Вони зазвичай мають комерційний характер, включаючи продаж або купівлю товарів, оренду житла, а також інформування про культурно-освітні заходи та вакансії. Суспільно важливі повідомлення чи попередження тепер зустрічаються рідше.

У минулому основним каналом для розміщення реклами були газети. Кілька друкованих видань відводили цілі сторінки або розділи для оголошень. Ці повідомлення інформували про продажі та інші комерційні пропозиції. Проте, з часом популярність друкованих ЗМІ знизилася, як і їхня аудиторія.

Таблички на автобусних зупинках і біля житлових будинків також використовувалися для розміщення оголошень, але цей метод мав низку недоліків, таких як обмежена доступність та захаращеність непотрібною інформацією.

Сьогодні інтеграція оголошень у життя сучасної людини можлива лише за допомогою веб-технологій, які дозволяють охопити широку аудиторію та забезпечити всі необхідні інструменти для ефективної роботи оголошень.

Веб-платформи для онлайн-оголошень з нерухомості стали незамінним інструментом у сучасному світі. Вони дозволяють користувачам легко знаходити нерухомість для оренди або купівлі, переглядати детальні характеристики об'єктів, а також зв'язуватися з продавцями. Такі платформи забезпечують точний пошук, зручну навігацію та можливість фільтрувати оголошення. Крім того, використання віртуальних турів та відеооглядів значно покращує користувацький досвід, дозволяючи потенційним покупцям чи орендарям більш детально ознайомитися з об'єктом нерухомості, не виходячи з дому.

1.2.Актуальність веб платформи онлайн оголошень

У сучасному світі Інтернет став невід'ємною частиною повсякденного життя мільйонів людей по всьому світу. Веб-платформи для онлайн оголошень відіграють важливу роль у цьому процесі, забезпечуючи зручний доступ до великої кількості оголошень. Спеціалізовані платформи для оренди та купівлі нерухомості є особливо важливими, оскільки вони дозволяють користувачам легко знайти, орендувати або придбати житло.

Із зростанням попиту на нерухомість та збільшенням кількості користувачів Інтернету, все більше людей звертаються до онлайн платформ для пошуку нерухомості. Згідно зі звітами, значна частина покупців та орендарів нерухомості спочатку шукають варіанти онлайн перед тим, як зробити остаточне рішення. Наприклад, дослідження показують, що 99% мілленіалів і 90% бебі-бумерів починають пошук нерухомості онлайн. Це підкреслює важливість доступності та зручності використання таких платформ.

Використання технологій, таких як відео та стріми, значно підвищує ефективність маркетингових зусиль у сфері нерухомості. Оголошення з відео отримують на 403% більше запитів порівняно з тими, що їх не мають . Крім того, пандемія COVID-19 сприяла зростанню значення онлайн-платформ для нерухомості, оскільки люди стали більше покладатися на дистанційні огляди і віртуальні тури.

Війна в Україні значно вплинула на ринок нерухомості. З лютого 2022 року до січня 2023 року продажі житлової нерухомості впали на 46%. Однак, попри ці важкі умови, ринок почав поступово відновлюватися. Зокрема, влітку 2022 року спостерігалася певна стабілізація та поступове відновлення активності на ринку нерухомості.

Таким чином, розробка веб-платформи для онлайн оголошень, спеціалізованої на оренді та купівлі нерухомості, є надзвичайно актуальною темою у сучасних умовах. Вона відповідає потребам ринку нерухомості, забезпечує

зручність для користувачів та використовує сучасні технології для створення додатку.

Як видно на рисунку 1.1 згідно з даними NAR, 47% покупців житла починають пошук з перегляду об'єктів нерухомості онлайн. Це вказує на важливість якісного представлення об'єктів нерухомості на веб-платформах для залучення більшої кількості потенційних клієнтів. Лише 18% клієнтів одразу звертаються до агента, що підтверджує необхідність доступної та детальної інформації про нерухомість в інтернеті. Крім того, тільки 8% клієнтів шукають інформацію про процес купівлі житла, що свідчить про те, що більшість зосереджені на пошуку конкретних об'єктів.

Ці статистичні дані підкреслюють необхідність забезпечення функціональності для детального пошуку та фільтрації об'єктів нерухомості, а також значення високої якості фотографій та докладного опису об'єктів на веб-платформі.

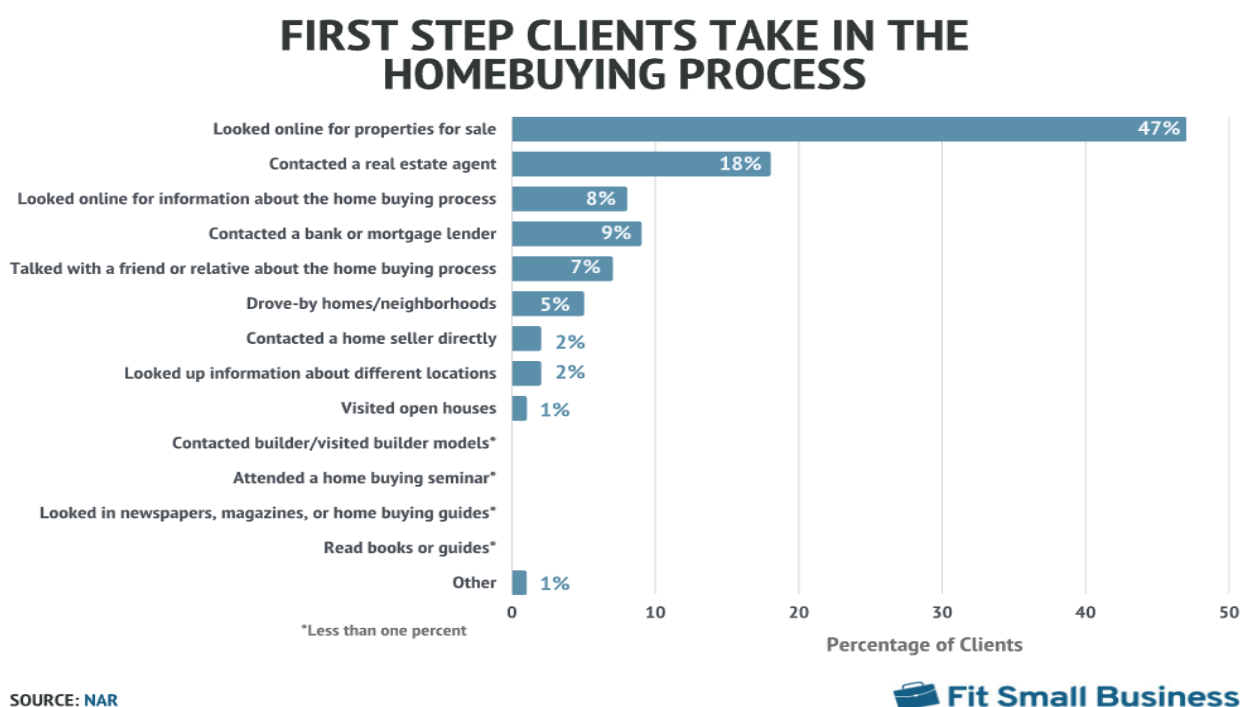


Рис 1.1 Перший крок клієнтів у процесі купівлі житла

1.3. Огляд та аналіз web-платформ онлайн оголошень

1.3.1. OLX

OLX – це платформа для розміщення оголошень, заснована у 2006 році французькими підприємцями Фабрісом Гранді і Олександром Оксенфордом. Спочатку компанія розпочала свою діяльність в Індії, яка на той час була дуже перспективним ринком. Протягом наступних 10 років платформа поширилася на ринки багатьох інших країн по всьому світу. OLX стала популярною в Україні після об'єднання з місцевою онлайн-дошкою оголошень Slando у 2015 році.

Цей веб-додаток надає можливість як приватним особам, так і компаніям купувати та продавати продукти і послуги через Інтернет. Навіть без реєстрації користувачі можуть переглядати пропозиції на головній сторінці сайту на рисунку 1.2.

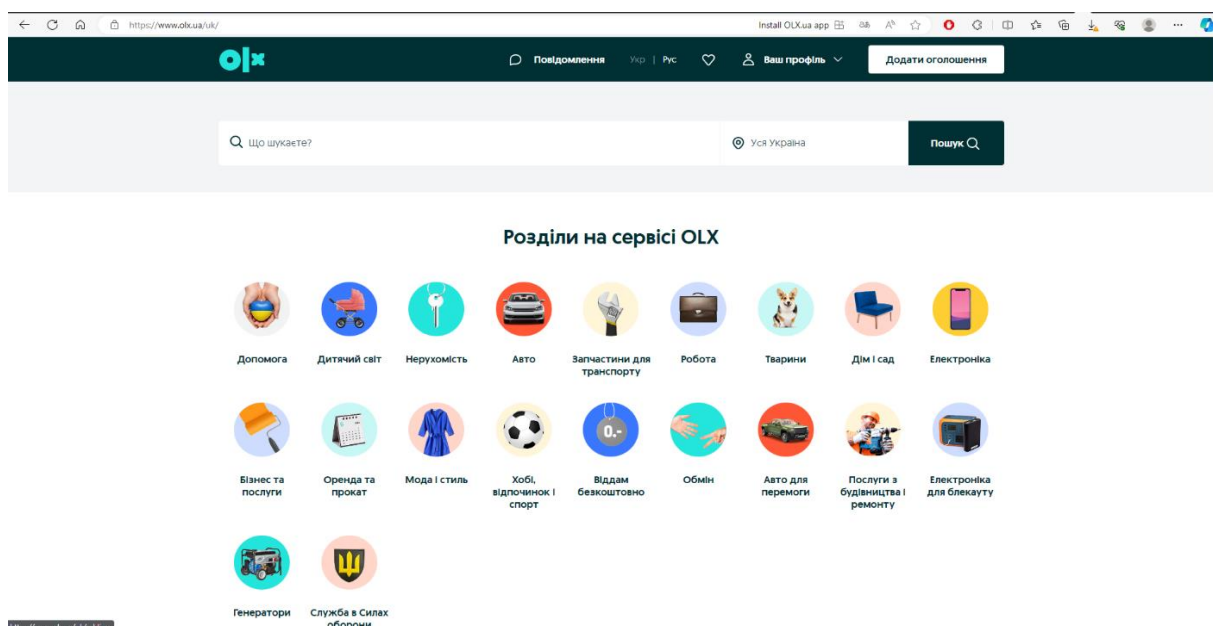


Рис 1.2 Головна сторінка OLX

Неавторизовані користувачі також мають доступ до функцій пошуку, фільтрації та сортування сторінок. Що стосується фільтрів, користувач може обирати товари або послуги за місцем розташування, ціною або конкретною категорією.

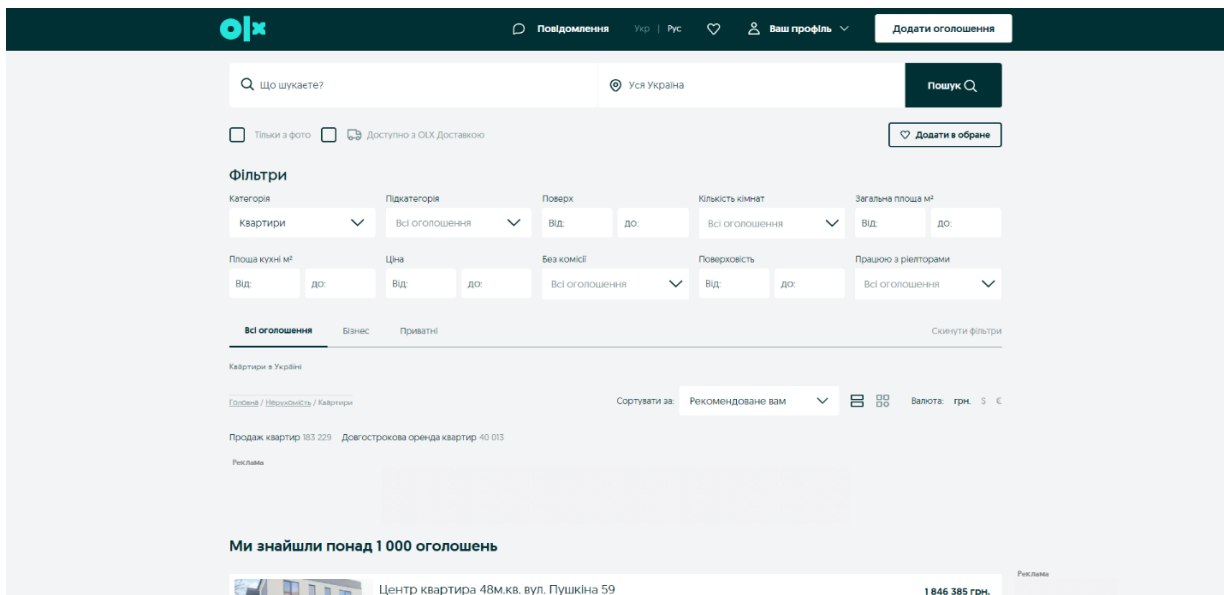
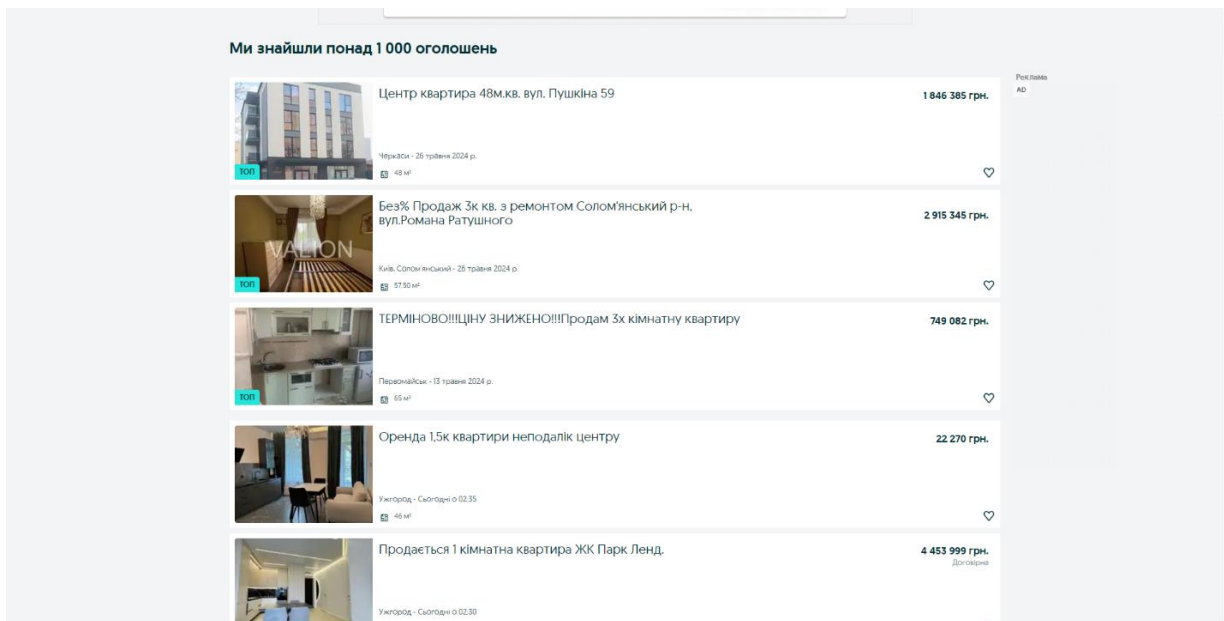


Рис 1.3 Блок фільтрів OLX



1.4 Сторінка з фільтрами оголошень OLX

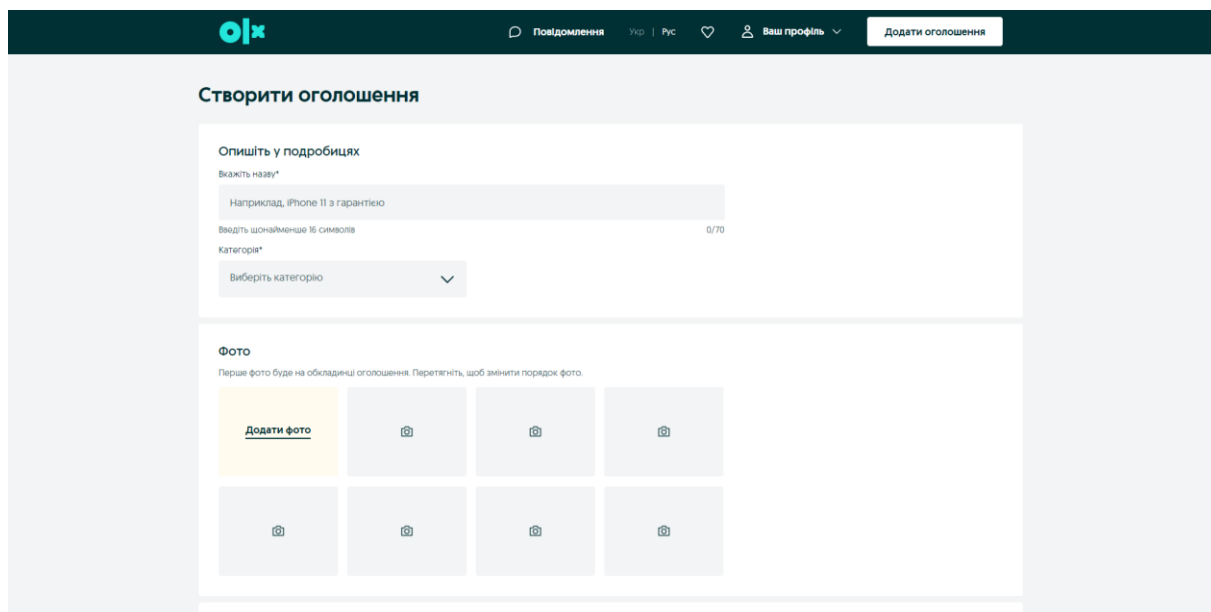


Рис 1.5 Сторінка створення оголошення OLX

Користувачі з акаунтами, а також компанії можуть створювати оголошення про свої товари або послуги. Для цього потрібно заповнити форму, вказавши назву товару або послуги, опис та контактну інформацію продавця, а також вибрати відповідну категорію. Крім того, можна завантажити одну або декілька фотографій для додаткової інформації (рисунок 1.6).

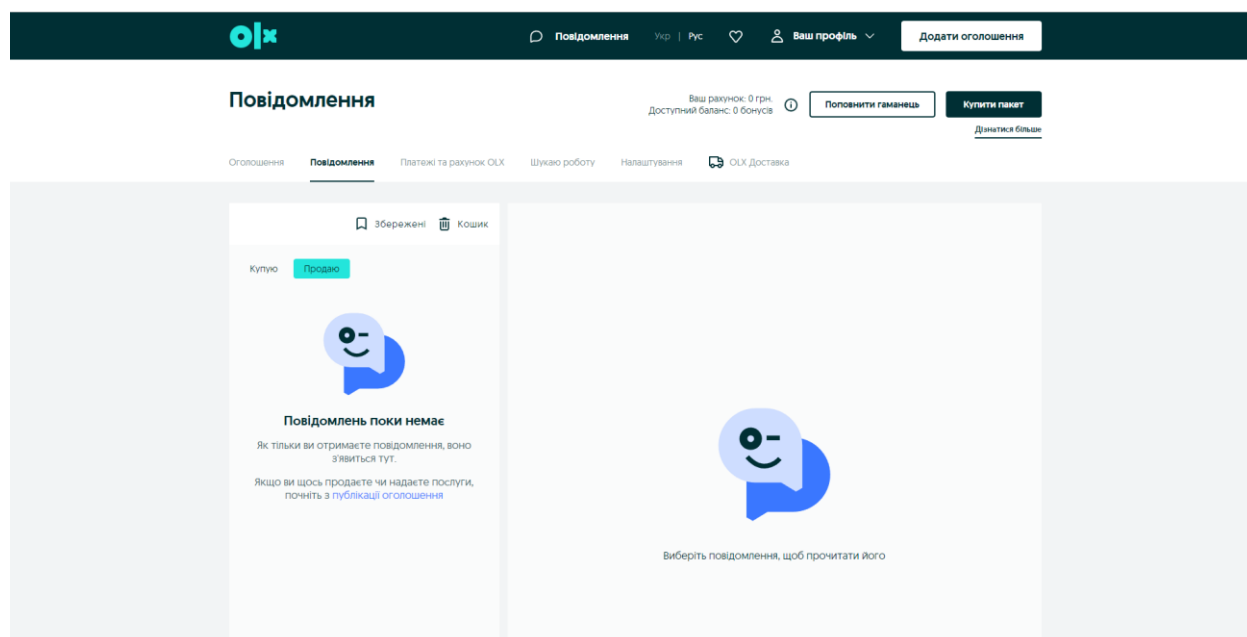


Рис 1.6 Сторінка повідомлень та чатів OLX

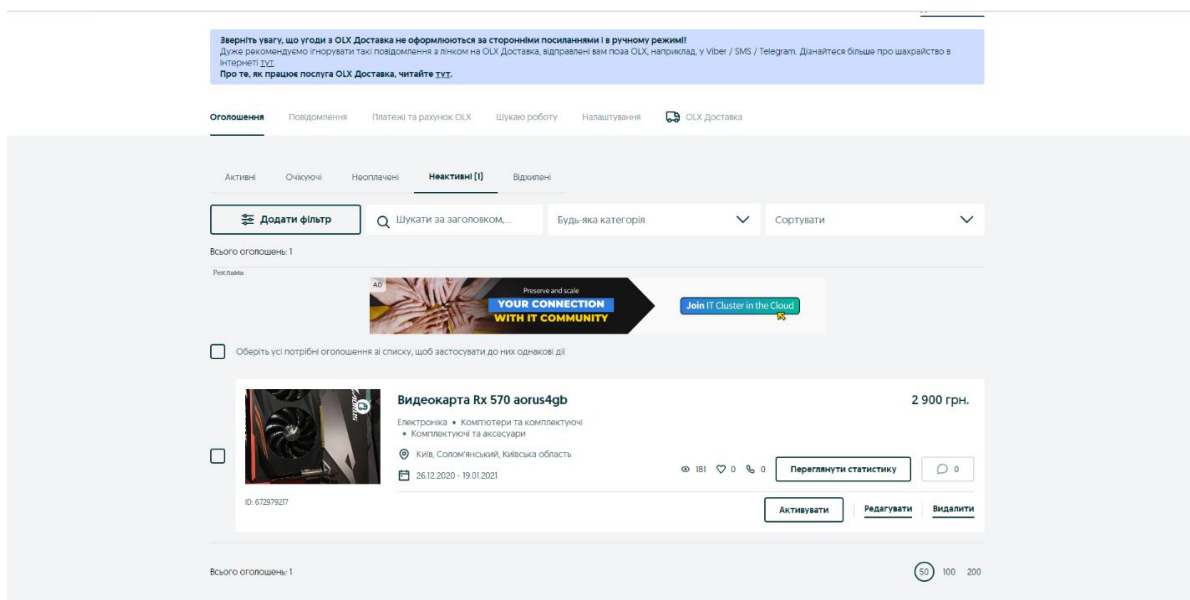


Рис 1.7 Сторінка оголошень користувача OLX

Особа, яка подала оголошення, може переглядати, редагувати або видаляти фотографії, показано на рисунку 1.7. Також можна відповісти на повідомлення щодо оголошення чи послуги, написаних іншими користувачами

1.3.2. Obyava.ua

Obyava.ua – це українська платформа для розміщення оголошень, яка почала свою роботу наприкінці серпня 2015 року. Проект був задуманий спочатку як регіональний ресурс для мешканців Чернігова, які мали потребу у додаткових коштах через продаж своїх речей під час економічно складних часів. Однак, під час розробки було прийнято рішення охопити всі регіони України, що дозволило платформі швидко набрати популярність по всій країні.

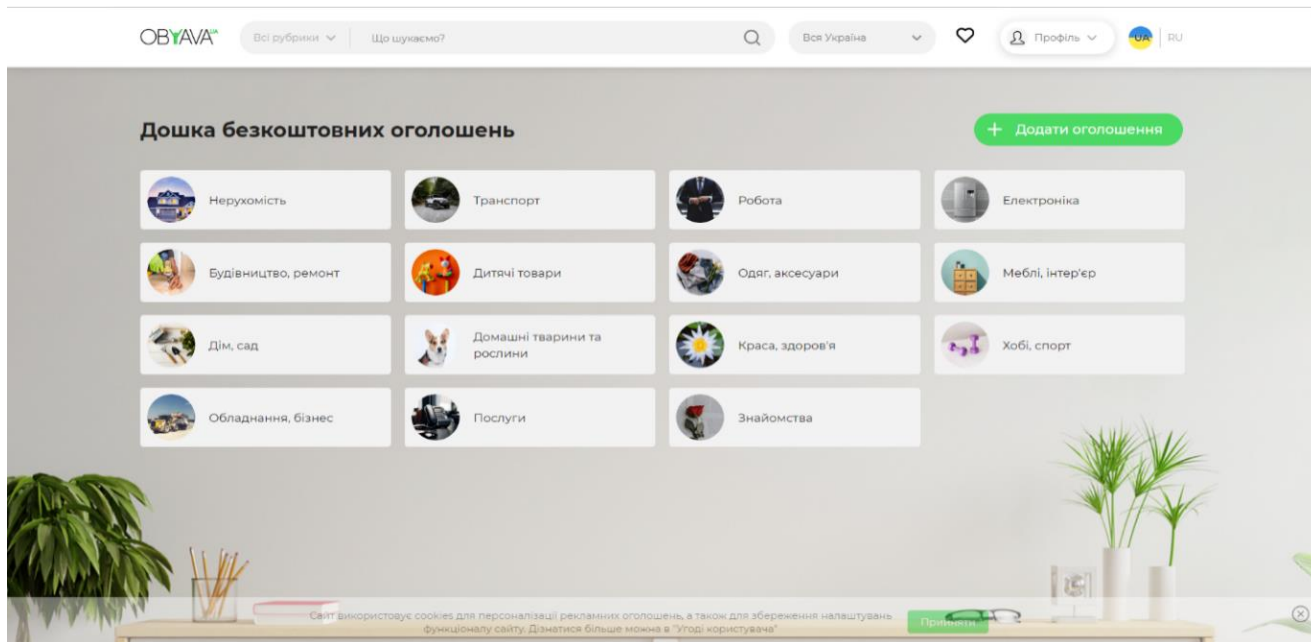


Рис 1.8 Головна сторінка obyava.ua

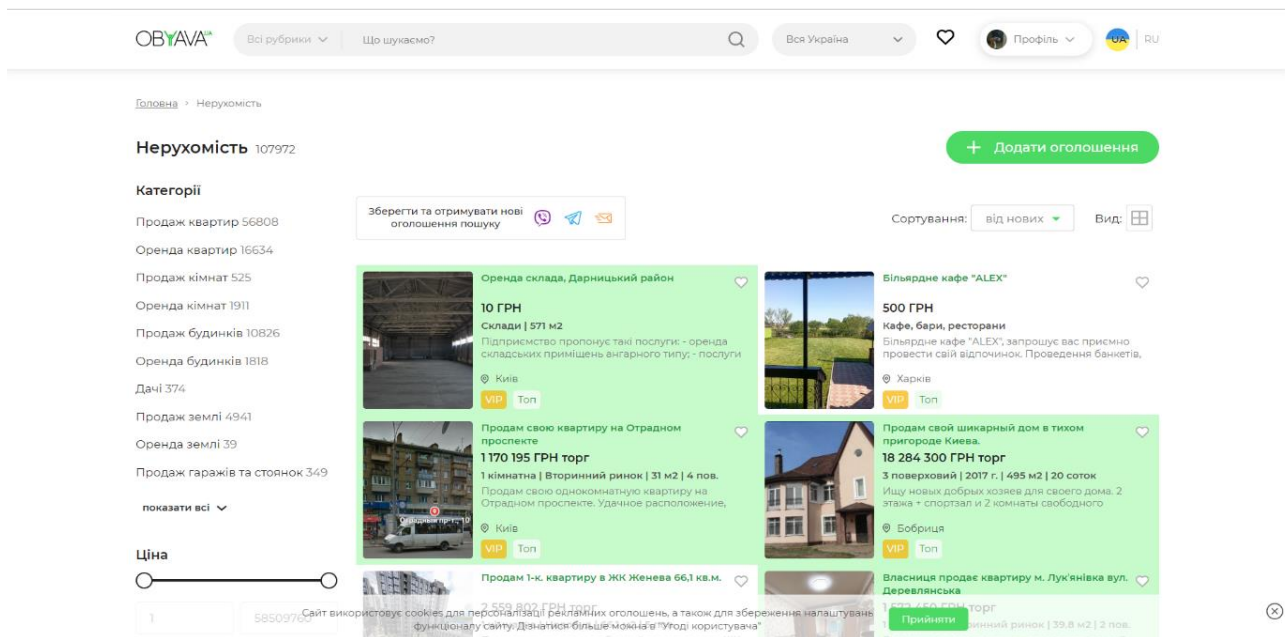


Рис 1.9 Головна сторінка obyava.ua

Рис 1.10 Сторінка створення оголошення obuava.ua

Рис 1.11 Сторінка оголошень користувача obuava.ua

Платформа Obyava.ua надає аналогічні функції, що й OLX, включаючи можливість створення, редагування та видалення оголошень, а також використання подібних фільтрів для пошуку товарів і послуг за місцем розташування, ціною або категорією. Крім того, Obyava.ua аналогічно дозволяє користувачам відповідати на

повідомлення, що стосуються їхніх оголошень або послуг, створених іншими користувачами, забезпечуючи комунікацію між продавцями та покупцями.

1.3.3. Ogolosha.ua

Ogolosha.ua – українська платформа для розміщення оголошень, яка функціонально подібна до інших популярних платформ, таких як OLX та Обуава.ua. Вона дозволяє користувачам публікувати оголошення про нерухомість, транспорт, електроніку, одяг, роботу та інші товари і послуги.

Неавторизовані користувачі можуть переглядати оголошення, користуватися пошуком, фільтрами та сортуванням. Авторизовані користувачі можуть створювати оголошення, заповнюючи необхідні поля, завантажувати фотографії, редагувати та видаляти свої оголошення, а також відповідати на повідомлення від інших користувачів й залишати коментарі на сторінці оголошення (рисунок 1.12.)

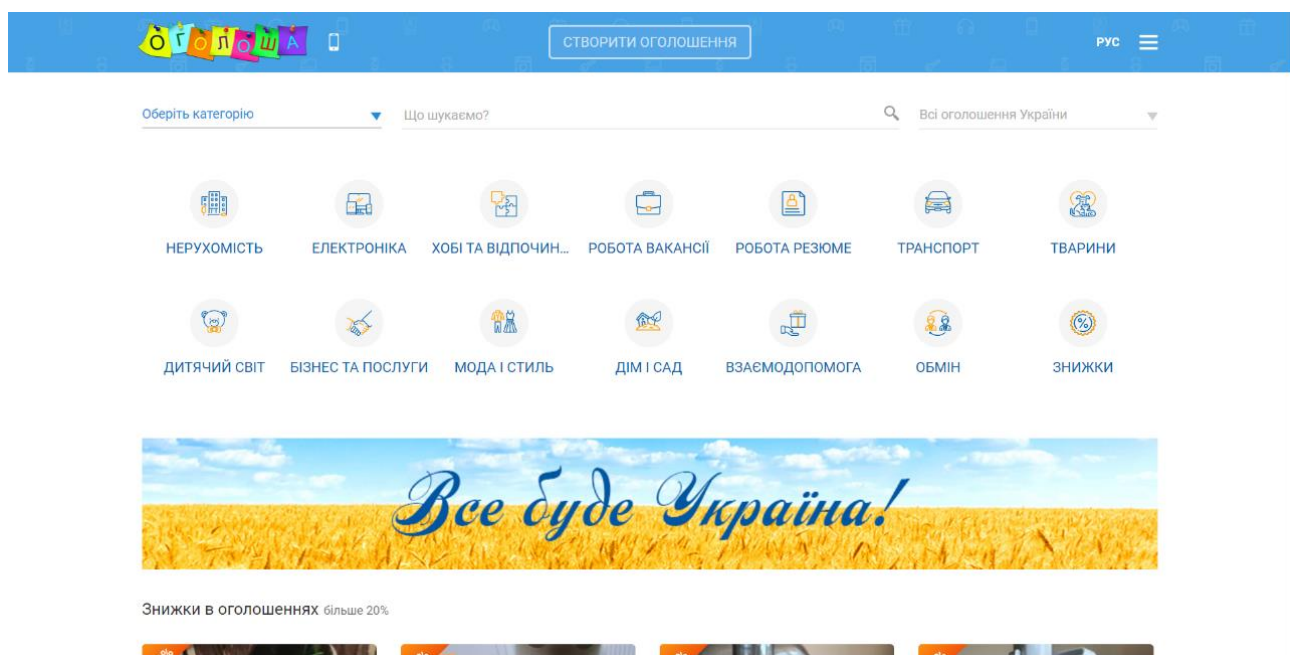


Рис 1.12 Головна сторінка ogolosha.ua

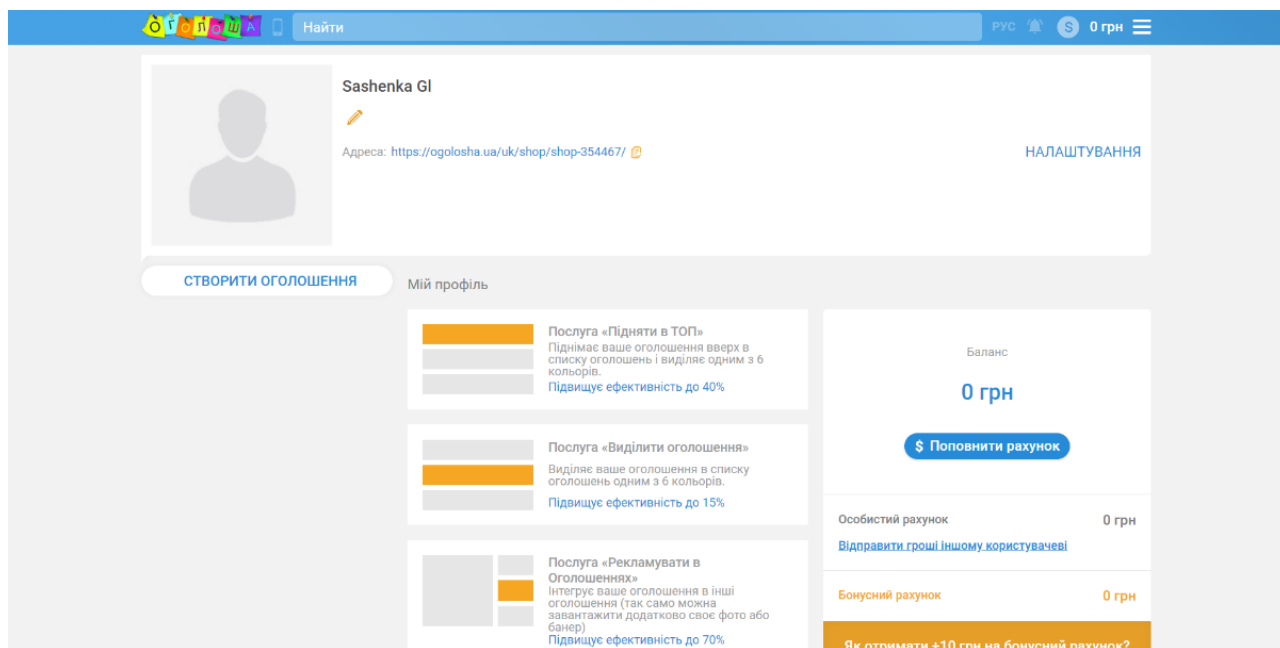


Рис 1.13 Сторінка оголошень користувача ogolosha.ua

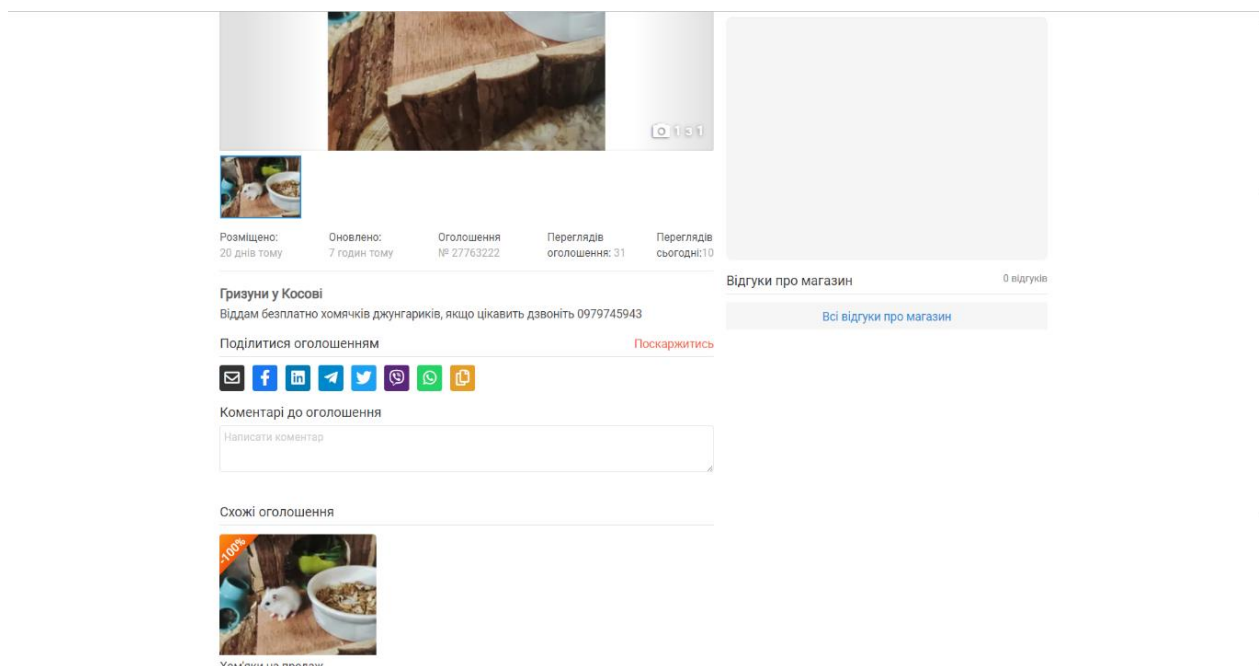


Рис 1.14 Сторінка оголошення з вікном коментар ogolosha.ua

1.4. Запозичення досвіду аналогів при створенні свого проекту

Головними недоліками цих платформ є відсутність адаптивної верстки, що забезпечує коректне відображення інтерфейсу на різних пристроях, та недостатня масштабованість інтерфейсу. Крім того, вони не підтримують можливість

прикріплення відеооглядів нерухомості, що обмежує потенційних покупців у доступі до більш детальної інформації про об'єкт.

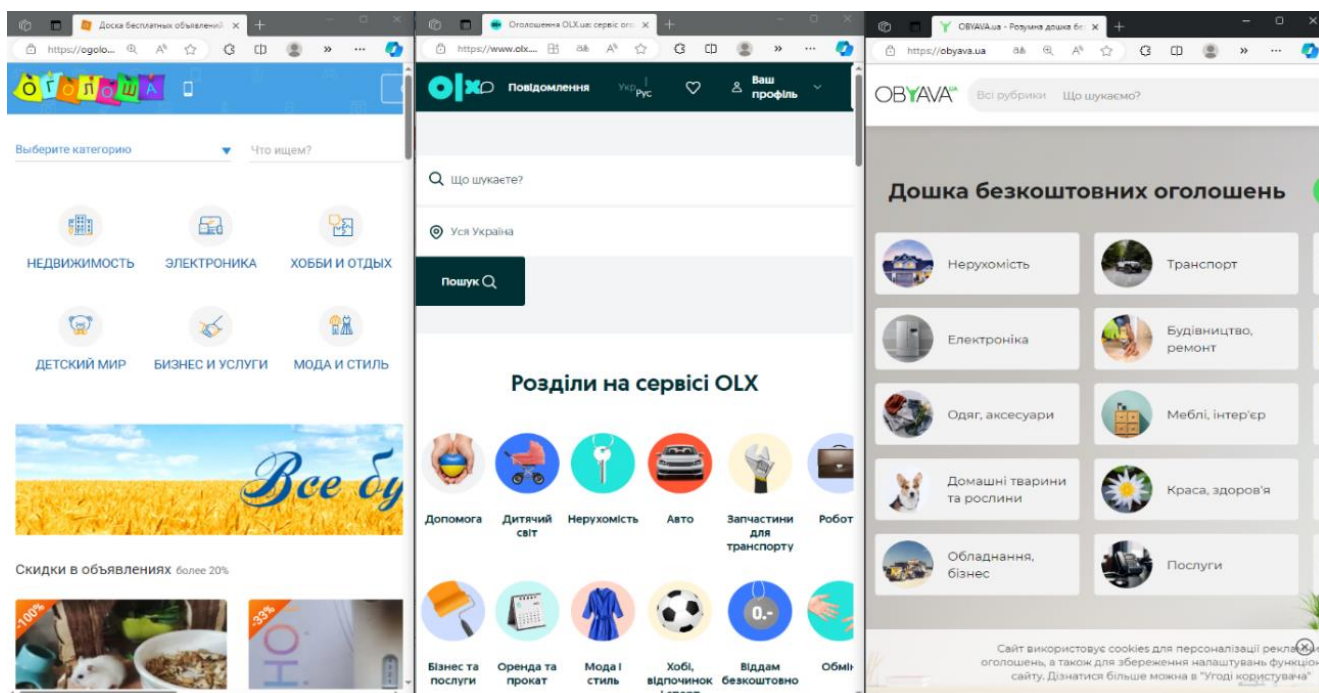


Рис 1.15 Некоректне відображення інтерфейсу

Таблиця 1.1

Зведені результати аналізу характеристик web-платформ

Характеристика застосунку	olx.ua	obyava.ua	ogolosha.ua
Банери з інформацією про зниження ціни	-	-	+
Адаптивна верстка під різні екрани пристроїв та масштабованість інтерфейсу	-	-	-
Авторизація через google	+	+	+
Фільтри оголошень зі знижками й	-	-	-
Специфікація на оголошеннях з нерухомості	-	-	-
Відсутність рекламних банерів	-	-	-
Можливість коментувати й оцінювати оголошення	-	-	-
Вікно в оголошенні для вставки відео з ютубу з оглядом на нерухомість	-	-	-

1.5. Масштабування

Розширення масштабу, функціональності та ресурсів веб-платформи для онлайн оголошень, спеціалізованої на оренді та купівлі нерухомості, є ключовим аспектом для задоволення зростаючих потреб і вимог користувачів. Це може означати обробку більшого обсягу даних, додавання нових функцій, обслуговування більшої кількості користувачів, підвищення продуктивності та створення умов для майбутнього розвитку. В умовах зростання обсягу та складності завдань, масштабування проєкту допомагає забезпечити стабільне та ефективне функціонування системи.

Розширення функціональності: Масштаб проєкту може бути збільшений шляхом надання платформи нових функціональних можливостей. Для більш ефективного управління оголошеннями про нерухомість можна впровадити аналітичні інструменти, рекомендаційні системи на основі машинного навчання, інтеграцію з картографічними сервісами та інші корисні функції.

Підтримка нових типів оголошень: Платформа повинна бути адаптивною і швидко адаптуватися до нових вимог ринку нерухомості. Це може включати підтримку різних типів нерухомості, додавання нових категорій оголошень, а також можливість розміщення оголошень про комерційну нерухомість, земельні ділянки та інші об'єкти.

Масштабування на рівні користувача: Якщо платформа стає популярною, важливо, щоб вона могла масштабуватися на рівні користувача. Це означає, що система повинна підтримувати багато одночасних користувачів і надавати кожному з них миттєвий доступ до оголошень та аналітичних даних. Для цього можуть використовуватися сучасні технології, такі як балансування навантаження та горизонтальне масштабування.

Розподілена архітектура: Для забезпечення масштабованості проєкту важливо враховувати розподілену архітектуру, яка дозволяє розміщувати багато компонентів платформи на різних серверах або в хмарних налаштуваннях. Це гарантує адаптивність та масштабованість системи відповідно до вимог

користувачів та навантаження. Використання контейнеризації та оркестрації контейнерів (наприклад, Docker та Kubernetes) може значно полегшити цей процес.

Підтримка різних джерел даних: Платформа повинна мати можливість взаємодіяти з різними джерелами даних, включаючи бази даних, зовнішні API, новинні сайти та інші джерела інформації. Це забезпечить точність і актуальність даних, які надаються користувачам, а також дозволить інтегрувати нові джерела даних у майбутньому.

Розширення географічного охоплення: Масштабування платформи до різних географічних регіонів дозволить розширити її доступність і вплив на міжнародному рівні. Це може включати локалізацію контенту, підтримку різних мов та адаптацію до місцевих вимог і нормативів.

Масштабованість інтеграцій: Платформа повинна бути обладнана для підключення до різних сервісів та інструментів, які використовуються в екосистемі нерухомості. Це можуть бути платіжні системи, сервіси електронного підпису, інтеграції з агентствами нерухомості тощо. Забезпечення сумісності та готовності до інтеграції дозволить розширити функціональність платформи та підвищити її привабливість для користувачів.

Висновок: Масштабування веб-платформи для онлайн оголошень дозволяє забезпечити її стабільну роботу та відповідати зростаючим потребам користувачів. Розширення функціональності, підтримка нових типів оголошень, масштабування на рівні користувача, розподілена архітектура, підтримка різних джерел даних, географічне охоплення, масштабовані інтеграції та безпека є ключовими аспектами для успішного розвитку платформи. Забезпечуючи масштабованість, можна збільшити пропускну здатність і ефективність системи, щоб краще задовольняти зростаючі потреби користувачів та ринку нерухомості.

2. ОПИС ТЕХНОЛОГІЙ ТА МОВ ПРОГРАМУВАННЯ ДЛЯ РОЗРОБКИ

2.1. Моделювання вимог до веб платформи

Діаграма варіантів використання демонструє основні взаємодії між акторами та системою та висвітлює ключові функціональні можливості платформи. Ця діаграма служить орієнтиром для розуміння загального потоку дій. Загальний вигляд діаграми варіантів використання веб-платформи для оголошень наведено на рисунках 2.1 та 2.2.

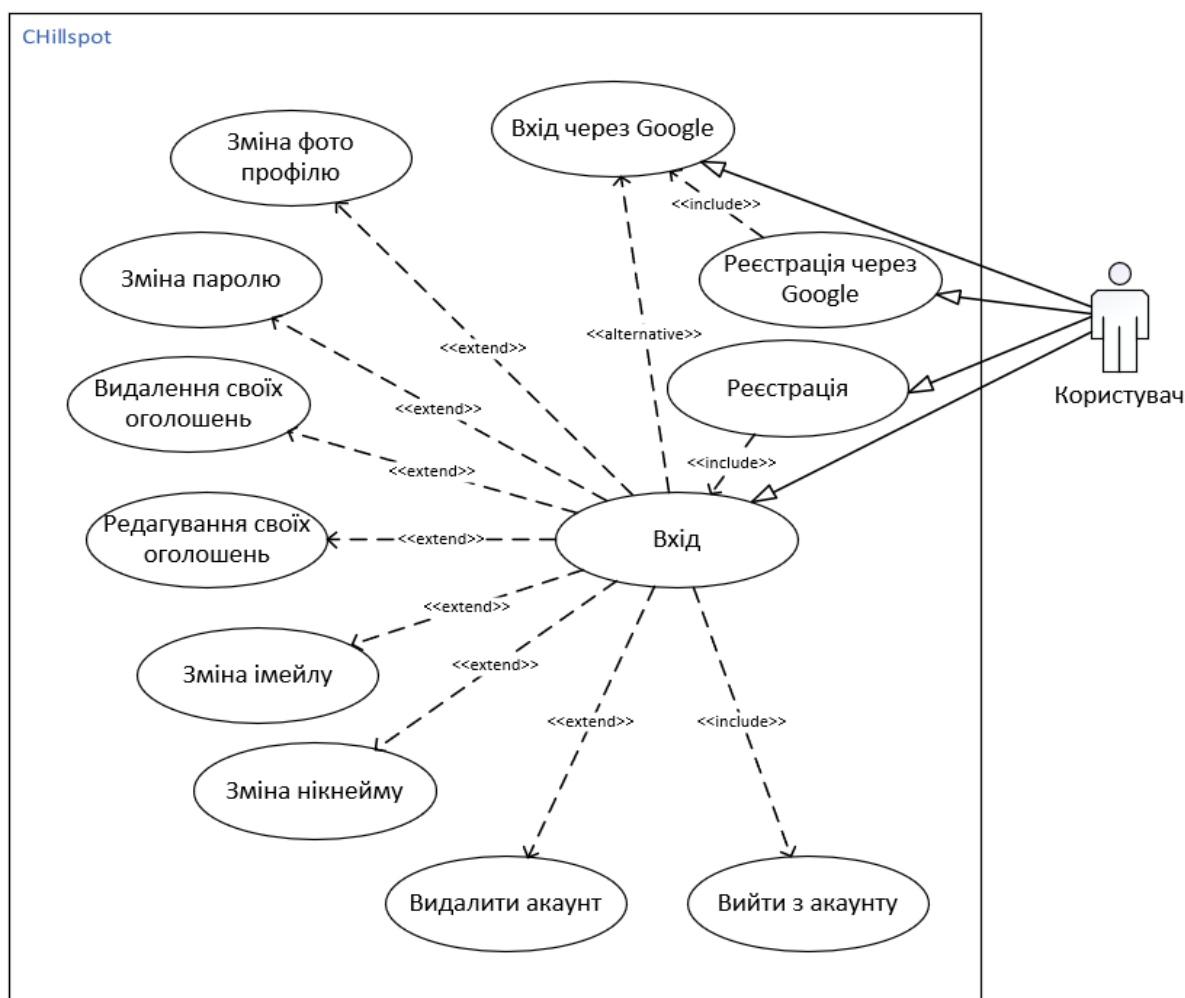


Рис 2.1. Діаграма варіантів використання для користувача

Діаграма варіантів використання для користувача (рисунок 2.1) показує основні дії, які може виконувати користувач після входу в систему. Користувач може увійти в систему за допомогою звичайної реєстрації або через Google, а також зареєструватися у системі або за допомогою Google. Після входу користувач отримує доступ до функцій управління обліковим записом, таких як зміна паролю, зміна нікнейму, зміна імейлу та зміна фото профілю. Користувач також може редагувати свої оголошення, видаляти їх, видалити свій обліковий запис або вийти.

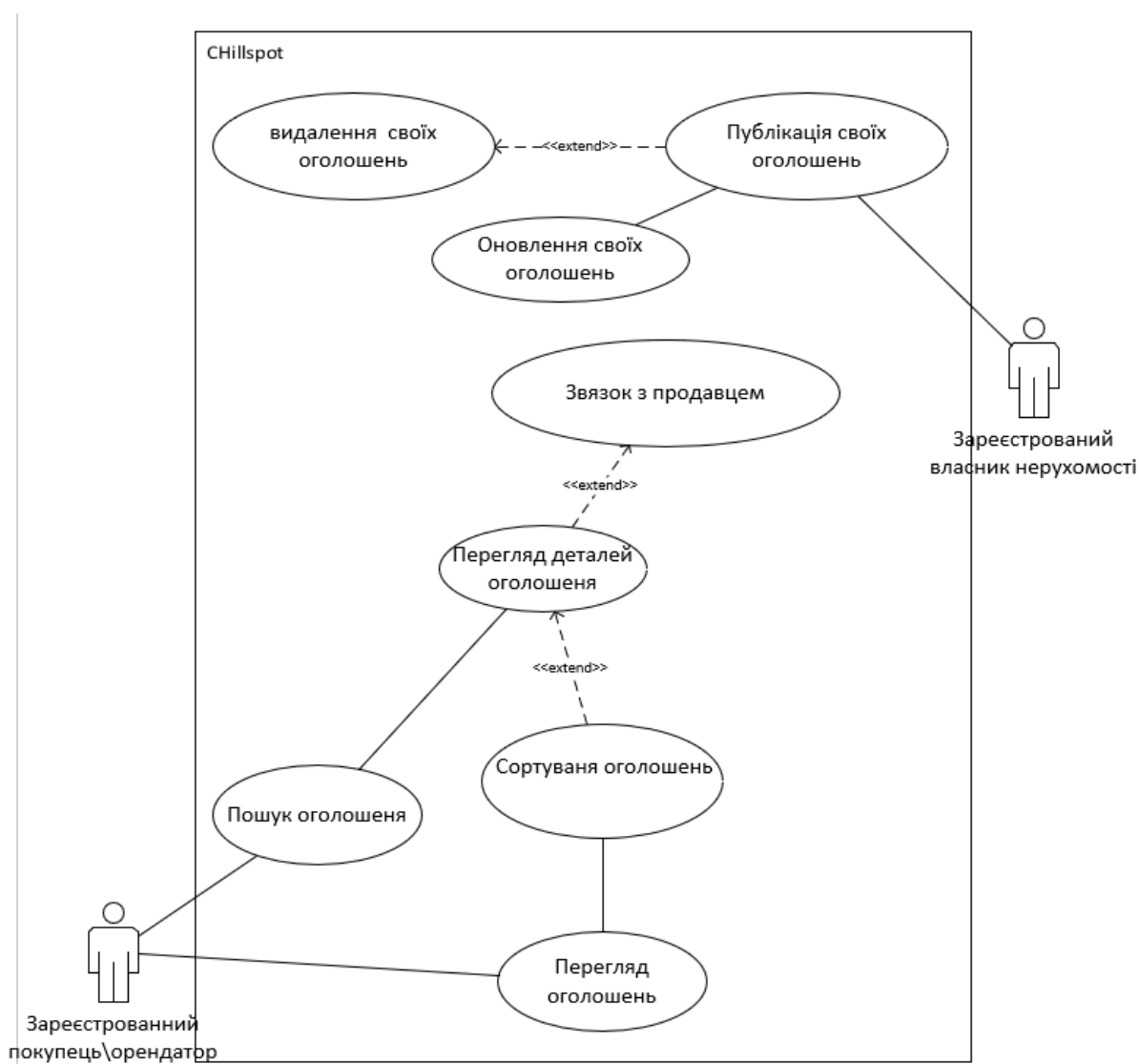


Рис 2.2 Діаграма варіантів використання для зареєстрованого власника та покупця

Діаграма варіантів використання для зареєстрованого власника та покупця (рисунок 2.2) відображає функціональні можливості, доступні зареєстрованим користувачам, зокрема власникам нерухомості та покупцям/орендарям. Зареєстрований власник нерухомості має можливість публікувати свої оголошення, видаляти їх та оновлювати інформацію про свої оголошення. Зареєстрований покупець або орендар може шукати оголошення за різними критеріями, переглядати доступні оголошення, сортувати їх за різними параметрами (ціна, дата публікації тощо), переглядати детальну інформацію про вибране оголошення та зв'язуватися з власником оголошення для отримання додаткової інформації.

2.2. Опис архітектури MVC

Для розробки веб-платформи для онлайн оголошень з нерухомості була обрана архітектура MVC (Model-View-Controller). Архітектура MVC розділяє програму на три основні компоненти: Модель, Вид і Контролер. Кожен компонент виконує свої функції та має свою відповідальність, що дозволяє розділити логіку програми на окремі частини, роблячи її більш масштабованою, зручною для розробки та підтримки.

На рисунку 2.3 наведено графічний приклад архітектури MVC та показано, як працює MVC в розроблюваній веб-платформі.

Frontend: React Components - цей компонент відповідає за взаємодію з користувачем через браузер. Він включає в себе два основні підкомпоненти: User View Component - відображає інформацію користувача, отриману від контролера користувача, та Listing View Component - відображає оголошення, отримані від контролера оголошень.

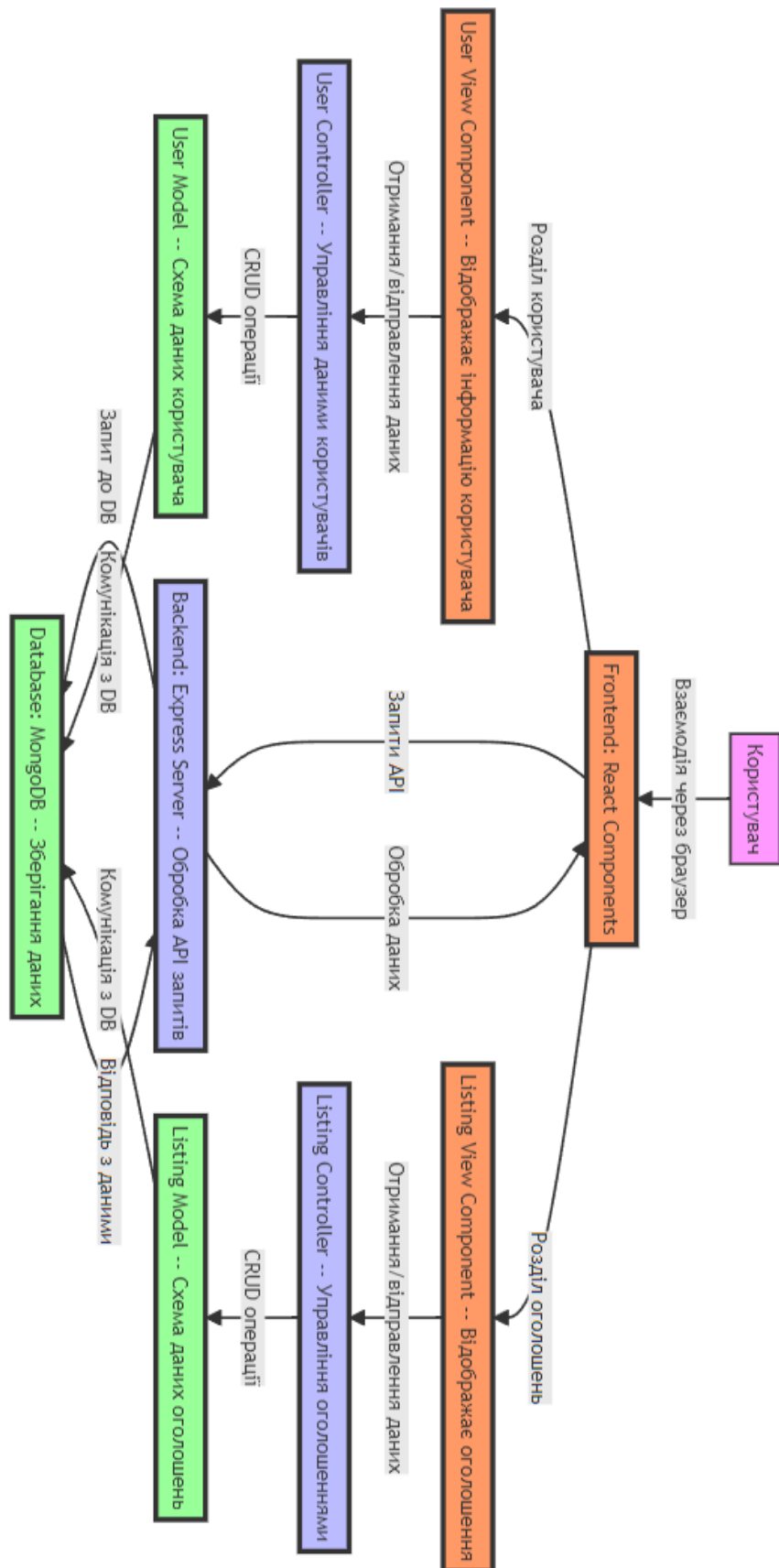


Рис 2.3 архітектура MVC в розроблюваній веб-платформі

Backend: Express Server - відповідає за обробку API запитів, що надходять від фронтенду. Він містить два основні контролери: User Controller - управляє даними користувачів, забезпечуючи CRUD операції (створення, читання, оновлення, видалення) через модель користувача, та Listing Controller - управляє даними оголошень, забезпечуючи CRUD операції через модель оголошень.

Database: MongoDB - зберігає дані користувачів та оголошень. Моделі даних взаємодіють з базою даних для збереження та отримання необхідної інформації.

Ця структура дозволяє чітко розділити обов'язки між різними компонентами системи, що сприяє легкості розширення функціоналу та спрощує підтримку і оновлення додатку.

2.2.1. Модель в архітектурі MVC

У розроблюваній веб-платформі для онлайн оголошень з нерухомості використовується архітектура MVC, де Модель (Model) відповідає за управління даними та бізнес-логікою додатку. Модель є окремим компонентом, який виконує наступні функції: забезпечення доступу до даних, обробка логіки додатку, взаємодія з Контролером, повідомлення про зміни даних, незалежність від інтерфейсу користувача, легке тестування, незалежність від технологій та інших компонентів.

Модель відповідає за зберігання та доступ до даних додатку, таких як оголошення, користувачі, категорії та інші пов'язані дані. Дані зберігаються в базі даних MongoDB, що забезпечує швидкий та надійний доступ до інформації. Модель гарантує, що дані завжди відображають поточний стан додатку, дозволяючи користувачам отримувати актуальну інформацію. Вона містить бізнес-логіку додатку, яка відповідає за обробку даних. Це включає перевірку валідності введених користувачами даних, розрахунок необхідних показників та генерацію звітів. Наприклад, при створенні нового оголошення Модель перевіряє коректність даних, що вводяться, і зберігає їх у базі даних.

Модель взаємодіє з Контролером для обміну даними та виконання бізнес-логіки. Контролер викликає методи Моделі для отримання, зберігання або

оновлення даних. Наприклад, при оновленні оголошення Контролер звертається до Моделі, щоб отримати поточні дані оголошення, внести зміни і зберегти оновлені дані назад у базу даних. Модель повідомляє Контролер про будь-які зміни даних, що відбуваються у додатку. Це дозволяє Контролеру оновлювати відображення даних у Виді, забезпечуючи користувачам актуальну інформацію. Наприклад, при зміні статусу оголошення (наприклад, "продано"), Модель інформує Контролер, щоб він міг оновити відображення на сторінці оголошення.

Модель розроблюваної платформи є незалежною від інтерфейсу користувача. Вона працює з даними незалежно від того, як ці дані відображаються у Виді. Це дозволяє змінювати інтерфейс користувача без впливу на логіку роботи з даними. Наприклад, ми можемо змінювати дизайн або структуру сторінок без необхідності вносити зміни в логіку обробки даних.

Модель розроблюваної платформи незалежна від конкретних технологій, таких як бази даних або веб-служби. Це означає, що ми можемо змінювати використовувані технології без необхідності змінювати логіку роботи Моделі. Наприклад, ми можемо перейти з MongoDB на іншу базу даних без значних змін у коді додатку. Модель є незалежною від інших компонентів додатку, таких як Вид і Контролер. Це дозволяє збільшити масштабованість та забезпечити більшу гнучкість додатку. Наприклад, ми можемо змінювати реалізацію Виду або Контролера без впливу на логіку роботи з даними в Моделі. Таким чином, у розроблюваній веб-платформі для онлайн оголошень з нерухомості Модель є ключовим компонентом, що відповідає за управління даними та бізнес-логікою додатку. Вона забезпечує незалежність від інших компонентів додатку, що дозволяє легко тестувати і масштабувати систему, а також забезпечувати гнучкість у використанні різних технологій.

2.2.2. Вид в архітектурі MVC

У розроблюваній веб-платформі для онлайн оголошень з нерухомості Вид (View) є основним компонентом, який відповідає за представлення даних користувачу та забезпечує інтерфейс взаємодії з додатком. Дані для Виду

надходять з Моделі, а дії користувача передаються до Контролера для подальшої обробки.

Основні функції Виду включають представлення даних користувачу у вигляді веб-сторінки, форми або іншого інтерфейсу взаємодії. Вид повинен бути зручним та привабливим для користувача, тому його дизайн має бути ретельно продуманим. Вид повинен бути здатний реагувати на дії користувача, такі як натискання кнопок, заповнення форм тощо. Дії користувача передаються до Контролера для подальшої обробки.

Вид повинен бути незалежним від Моделі та містити мінімум логіки. Він отримує дані з Моделі та відображає їх, але не займається їх обробкою. Це дозволяє зберігати простоту та гнучкість Виду, що дозволяє легко змінювати його без впливу на Моделю та Контролер. Вид повинен бути незалежним від технологій та може використовувати будь-яку технологію, яка дозволяє створювати веб-інтерфейси.

Вид повинен бути повторно використовуваним. Якщо у додатку є кілька екранів зі схожим виглядом, то можна використовувати один і той самий Вид для їх відображення. Шаблонізація дозволяє використовувати один і той самий Вид для відображення різних даних. Шаблони Виду містять заготовки коду, які можна заповнювати даними з Моделі. Це дозволяє швидко створювати нові Види та забезпечує їх гнучкість.

Вид повинен бути розділений на рівні, щоб забезпечити його гнучкість та простоту розробки. Наприклад, великий Вид може бути розділений на менші підвиди, кожен з яких відповідає за свою частину інтерфейсу. Вид повинен бути легко тестованим. Для цього можна використовувати автоматизовані тести, які перевіряють правильність відображення даних та взаємодії з користувачем.

Узагальнення: У розроблюваній веб-платформі для онлайн оголошень з нерухомості Вид є ключовим компонентом, який відповідає за представлення даних користувачу та забезпечує інтерфейс взаємодії з додатком. Забезпечуючи його гнучкість та простоту розробки, можна створювати швидкі та ефективні веб-додатки.

У проєкті View складається з React-компонентів, які відповідають за різні частини інтерфейсу. Наприклад, компонент Home.jsx відображає головну сторінку з оголошеннями.

```
import { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import ListingItem from '../components/ListingItem';

export default function Home() {
  const [listings, setListings] = useState([]);

  useEffect(() => {
    const fetchListings = async () => {
      try {
        const res = await fetch('/api/listing/get?limit=6');
        const data = await res.json();
        setListings(data);
      } catch (error) {
        console.error(error);
      }
    };
    fetchListings();
  }, []);

  return (
    <div className="p-5 max-w-screen-lg mx-auto">
      <h1 className="text-2xl font-bold mb-4">Останні Оголошення</h1>
      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
        {listings.map(listing => (
          <ListingItem key={listing._id} listing={listing} />
        ))}
      </div>
      <Link to="/search" className="block mt-4 text-center text-blue-500">Переглянути всі оголошення</Link>
    </div>
  );
}
```

Рис 2.4 Вид в розробленій веб-платформі

2.2.3. Контролер в архітектурі MVC

У розроблюваній веб-платформі для онлайн оголошень з нерухомості Контролер (Controller) є ключовим компонентом, що відповідає за обробку запитів користувача, взаємодію з Моделлю та Видом для забезпечення функціональності додатку. Контролер приймає запити від користувача та виконує необхідні дії для їх обробки. Наприклад, коли користувач надсилає запит на перегляд конкретного оголошення, Контролер отримує цей запит, взаємодіє з Моделлю для отримання відповідних даних і передає їх у Вид для відображення.

Контролер також розподіляє завдання між Моделлю та Видом, щоб забезпечити ефективну взаємодію між ними та підтримку розширюваності та гнучкості. Наприклад, при створенні нового оголошення Контролер взаємодіє з

Моделлю для збереження даних та з Видом для підтвердження успішного створення оголошення. Крім того, Контролер обробляє помилки та повідомлення про помилки, що повертаються від Моделі та Виду, і надає відповідні повідомлення для користувача. Наприклад, якщо введені користувачем дані не пройшли валідацію, Контролер відправляє повідомлення про помилку до Виду для інформування користувача.

Контролер здійснює маршрутизацію запитів на основі шляху запиту та інших параметрів. Наприклад, запити на адресу `"/listings"` обробляються контролером, що відповідає за оголошення, тоді як запити на адресу `"/users"` обробляються іншим контролером, що відповідає за користувачів. Важливо, щоб Контролер був легко тестованим для підтримки безперервної інтеграції та розробки. Ми можемо проводити модульні тести для перевірки коректності обробки запитів, розподілу завдань та обробки помилок.

Контролер може бути розділений на рівні, щоб забезпечити його гнучкість та простоту розробки. Наприклад, великий контролер може бути розділений на менші підконтролери, кожен з яких відповідає за свою частину функціональності додатку. Контролер також може зберігати стан додатку та переключати його між станами, наприклад, для обробки різних типів запитів. Це допомагає зберігати контекст користувача між різними запитами.

Контролер повинен бути легко розширюваним та змінюваним для підтримки розробки нових функцій та внесення змін до існуючої функціональності. Це дозволяє швидко додавати нові можливості до додатку без значних змін у існуючій логіці. Контролер може виконувати валідацію вхідних даних та виводити повідомлення про помилки користувачу. Наприклад, перед збереженням нового оголошення Контролер перевіряє, чи всі необхідні поля заповнені правильно.

Контролер також може зберігати дані сесій користувача та виконувати необхідні дії для забезпечення безпеки та конфіденційності даних. Наприклад, при вході користувача в систему Контролер створює сесію та зберігає дані про авторизацію.

Контролер у розроблюваній веб-платформі для онлайн оголошень з нерухомості є ключовим компонентом, що відповідає за обробку запитів користувачів, маршрутизацію запитів, взаємодію з Моделлю та Видом, обробку помилок, збереження стану додатку та інші функції, необхідні для забезпечення функціональності та ефективності додатку.

```

controllers > JS comment.controller.js > createComment
import Comment from '../models/comment.model.js';
import { errorHandler } from '../utils/error.js';

// Створення коментаря
export const createComment = async (req, res, next) => {
  const { text, rating, listingId, userName, userPhoto } = req.body;
  const userId = req.user.id;

  console.log('Request Body:', req.body);
  console.log('User Info:', { userId, userName, userPhoto });

  try {
    const comment = await Comment.create({ text, rating, listingId, userId, userName, userPhoto });
    return res.status(201).json(comment);
  } catch (error) {
    console.error('Error creating comment:', error);
    next(error);
  }
};

// Отримання коментарів
export const getComments = async (req, res, next) => {
  const { listingId } = req.params;

  try {
    const comments = await Comment.find({ listingId }).populate('userId', 'username avatar');
    return res.status(200).json(comments);
  } catch (error) {
    console.error('Error fetching comments:', error);
    next(error);
  }
};

```

Рис 2.5 Приклад контролеру в розробленій веб-платформі

2.2.4. Переваги архітектури MVC

Архітектура Model-View-Controller (MVC) має багато переваг, обумовлених розділенням обов'язків між Моделлю, Видом та Контролером, що зменшує кількість залежностей між компонентами додатку. Це дозволяє зберігати код чистим та організованим, сприяє підтримці та розширенню додатку.

Основні переваги включають високу повторюваність. Модель та Вид можуть бути повторно використані у різних контекстах, що зменшує кількість дублюючого коду та сприяє швидкій розробці. Компоненти MVC можна тестувати окремо один від одного, що зменшує час, необхідний для виконання тестів, та сприяє підвищенню якості коду.

Кожен компонент MVC може бути легко розширений та змінений без впливу на інші компоненти. Це дозволяє швидко додавати нові функції та вносити зміни до існуючої функціональності. MVC також дозволяє легко розділити логіку додатку та інтерфейс користувача, що сприяє підтримці багатомовних додатків.

Архітектура MVC дозволяє використовувати різні технології для реалізації кожного з компонентів. Наприклад, Модель може використовувати різні бази даних або зберігатися у файлової системі, а Вид може бути реалізований за допомогою різних інтерфейсів користувача, таких як веб-сторінки, мобільні додатки або десктопні програми.

Команда розробників може бути розділена на групи, які відповідають за різні компоненти архітектури. Це сприяє більш ефективній роботі, якщо команда складається з багатьох розробників. MVC дозволяє використовувати шаблони проектування та добрі практики програмування, що забезпечує високу якість коду та зменшує ризик виникнення помилок.

Код організований за логічними блоками, що дозволяє розробникам швидко зорієнтуватися у коді та знайти необхідну функціональність. Існує багато фреймворків, які реалізують архітектуру MVC та надають додаткові можливості для розробки додатків, наприклад, Express.js для Node.js, що використовуються в нашому проекті.

Узагальнення: Архітектура MVC дозволяє розробникам створювати додатки, які є легкими для розуміння та підтримки. Вона сприяє розділенню обов'язків та зменшує залежності між компонентами додатку, що забезпечує більш ефективну роботу розробників та збільшує швидкість розробки.

2.2.5. Недоліки архітектури MVC

Хоча архітектура MVC має багато переваг, вона також має деякі недоліки, які варто враховувати при виборі архітектури для проекту. Під час розробки додатків за архітектурою MVC може виникнути багато коду, оскільки кожен компонент має свою власну логіку та функціональність. Це може збільшити складність проекту та зробити його важким для підтримки.

Перекладання даних між компонентами додатку може бути витратним та викликати проблеми з продуктивністю. Крім того, якщо дані в одному з компонентів змінюються, потрібно оновлювати інші компоненти, що може викликати складність та підвищення ризику помилок.

Розробка додатків за архітектурою MVC може вимагати від розробника знань та досвіду використання шаблонів проектування та інших добрих практик програмування. Це може збільшити складність вивчення та розробки проекту для початківців.

Незважаючи на те, що архітектура MVC зменшує залежності між компонентами додатку, вона не розглядає взаємодію між компонентами на вищому рівні, що може створювати складнощі при розробці складних додатків. Іноді може виникати проблема з взаємодією між різними моделями, які використовуються в різних компонентах додатку. Це може призвести до дублювання даних та погіршення продуктивності.

Архітектура MVC є ефективним інструментом для розробки програмного забезпечення, особливо для великих та складних проєктів, але вона має свої недоліки та не підходить для всіх типів додатків. Вибір архітектури повинен залежати від специфіки проєкту та вимог до додатку, а також від знань та досвіду розробника. MVC не є панацеєю для всіх видів додатків та не вирішує всі проблеми, пов'язані з проектуванням та розробкою програмного забезпечення. Важливо також враховувати інші аспекти розробки додатків, такі як безпека, тестування, взаємодія з користувачами та інші.

2.3. Мова програмування JavaScript

JavaScript є основною мовою програмування, використаною в розроблюваній веб-платформі для онлайн оголошень з нерухомості. Вона забезпечує створення інтерактивних та динамічних веб-додатків, дозволяючи користувачам взаємодіяти з платформою в реальному часі. JavaScript була розроблена компанією Netscape у 1995 році і з тих пір стала основною мовою для веб-розробки.

JavaScript має кілька ключових особливостей, які роблять її ідеальною для нашої веб-платформи. Виконання на стороні клієнта дозволяє JavaScript забезпечувати швидку реакцію на дії користувачів без необхідності постійно звертатися до сервера. Це покращує користувацький досвід, дозволяючи оперативно перевіряти введені дані, оновлювати інтерфейс і надавати миттєві відповіді на дії користувача. Наприклад, при заповненні форми оголошення користувач одразу отримує повідомлення про можливі помилки введення.

JavaScript також підтримує асинхронне програмування за допомогою Promises та `async/await`, що дозволяє платформі обробляти кілька запитів одночасно без блокування основного потоку. Це критично важливо для нашої платформи, де користувачі можуть одночасно переглядати оголошення, завантажувати зображення та здійснювати пошук.

JavaScript є платформонезалежною мовою, що дозволяє її виконувати на різних операційних системах, таких як Windows, Linux і macOS. Це забезпечує доступність нашої платформи для широкого кола користувачів, незалежно від їхнього вибору операційної системи.

У нашій веб-платформі JavaScript використовується як основна мова для всіх компонентів MERN стека. Це включає серверну частину на Node.js, веб-фреймворк Express.js, базу даних MongoDB, а також фронтенд на React. Така інтеграція забезпечує узгоджену розробку і підтримку коду на всіх рівнях додатку.

JavaScript є ключовою мовою програмування для нашої веб-платформи оголошень з нерухомості, забезпечуючи інтерактивність, швидкість та гнучкість. Використання JavaScript у поєднанні з MERN стеком дозволяє створювати сучасні, ефективні та масштабовані веб-додатки, що задовольняють потреби користувачів у зручності та швидкості доступу до інформації.

2.4. Node.js

Node.js — це середовище виконання JavaScript на сервері, яке дозволяє запускати JavaScript-код поза браузером. У розроблюваній веб-платформі для онлайн оголошень з нерухомості Node.js використовується для забезпечення високої продуктивності та обробки великої кількості одночасних запитів. Node.js був розроблений Райаном Далем у 2009 році і швидко здобув популярність завдяки своїй асинхронній, подіє-орієнтованій моделі, яка робить його дуже ефективним для обробки запитів у реальному часі.

Node.js використовує неблокуючий, асинхронний підхід для обробки запитів, що дозволяє значно підвищити продуктивність додатка. Це особливо важливо для веб-платформи, де необхідно обробляти багато одночасних запитів, наприклад, пошук оголошень, обробка форм і завантаження зображень. Завдяки V8 JavaScript engine від Google, Node.js забезпечує високу швидкість виконання коду. Це дозволяє створювати швидкодіючі веб-додатки, які можуть обробляти великі обсяги даних та запитів у реальному часі.

Використання JavaScript як на стороні клієнта, так і на стороні сервера спрощує розробку та обслуговування додатка. Це дозволяє розробникам використовувати одну мову для всієї платформи, що зменшує кількість помилок і підвищує ефективність роботи. Node.js дозволяє горизонтально масштабувати додаток, додаючи нові вузли в кластері. Це дозволяє обробляти збільшений трафік без зниження продуктивності.

Node.js має великий набір модулів та бібліотек, доступних через npm (Node Package Manager), що спрощує додавання нових функцій до додатку. Ми можемо легко інтегрувати модулі для обробки файлів, баз даних, аутентифікації та інших функцій.

2.5. Express.js

Express.js є веб-фреймворком для Node.js, який забезпечує простий і гнучкий спосіб створення веб-додатків і API. У розроблюваній веб-платформі Express.js використовується для управління маршрутами, обробки запитів і відповіді на них, а також для взаємодії з базою даних. Він був розроблений TJ Holowaychuk у 2010 році і став одним з найпопулярніших фреймворків для Node.js завдяки своїй легкості у використанні та багатофункціональності.

Express.js забезпечує потужну систему маршрутизації, яка дозволяє визначити, який метод контролера має бути викликаний для обробки певного запиту. Це дозволяє організувати додаток у вигляді модулів, кожен з яких відповідає за свою частину функціональності, що робить код більш організованим і легким для підтримки. Express.js має простий і зрозумілий API, що дозволяє швидко почати розробку додатка. Це робить його ідеальним вибором для проектів будь-якого розміру, від маленьких скриптів до великих корпоративних додатків.

Express.js підтримує middleware, що дозволяє додавати функціональність до вашого додатка за допомогою невеликих, незалежних модулів. Middleware можуть обробляти запити, змінювати відповіді, взаємодіяти з базами даних і виконувати багато інших завдань. Express.js забезпечує гнучкість у виборі компонентів, які можна використовувати у додатку. Це дозволяє створювати високоефективні, масштабовані додатки, які легко адаптуються до змін у вимогах і умовах експлуатації.

Express.js має велику спільноту розробників і велику кількість доступних ресурсів, таких як документація, навчальні матеріали та приклади коду. Це дозволяє швидко знаходити відповіді на питання і вирішувати проблеми, що виникають під час розробки.

Node.js і Express.js є важливими компонентами вашої веб-платформи для онлайн оголошень з нерухомості. Node.js забезпечує високу продуктивність і асинхронну обробку запитів, тоді як Express.js надає простий і ефективний спосіб організації серверної частини додатку. Разом вони дозволяють створювати

масштабовані, швидкі і надійні веб-додатки, що забезпечують кращий користувацький досвід.

2.6. MongoDB

У розроблюваній веб-платформі для онлайн оголошень з нерухомості MongoDB використовується як основна база даних для зберігання інформації про оголошення, користувачів та інші важливі дані. MongoDB є документоорієнтованою базою даних NoSQL, яка забезпечує високу продуктивність і масштабованість. Ця база даних була обрана завдяки своїм перевагам, які ідеально підходять для наших вимог.

MongoDB зберігає дані у форматі BSON (Binary JSON), що дозволяє зберігати складні дані у вигляді документів. Це забезпечує гнучкість у зберіганні даних, дозволяючи легко змінювати структуру даних без впливу на існуючі записи. У нашій платформі це означає, що оголошення про нерухомість, користувачі та інші дані можуть бути збережені у вигляді документів з різною структурою, що спрощує процес розробки та обслуговування додатку.

MongoDB підтримує горизонтальне масштабування, що дозволяє розподіляти дані між кількома серверами для підвищення продуктивності. Це особливо важливо для нашої платформи, яка може мати великий обсяг даних і високий трафік. Горизонтальне масштабування забезпечує швидкий доступ до даних і високу доступність додатку.

MongoDB оптимізована для обробки великих обсягів даних і забезпечує швидке виконання запитів. Це дозволяє нашій платформі швидко реагувати на запити користувачів, такі як пошук оголошень, фільтрація та сортування даних.

MongoDB забезпечує автоматичну реплікацію даних, що підвищує надійність і доступність додатку. Реплікація дозволяє створювати копії даних на різних серверах, що забезпечує безперервну роботу платформи навіть у випадку збою одного з серверів.

MongoDB легко інтегрується з JavaScript і іншими мовами програмування через драйвери, що спрощує розробку і взаємодію з базою даних. Це дозволяє використовувати єдину мову програмування для всіх компонентів нашого додатку, що зменшує кількість помилок і підвищує продуктивність розробки.

2.7. Tailwind CSS

У розроблюваній веб-платформі для онлайн оголошень з нерухомості Tailwind CSS використовується для створення адаптивного та естетично привабливого інтерфейсу користувача. Tailwind CSS є утилітарним CSS-фреймворком для швидкого створення сучасних веб-дизайнів. Він був обраний для нашої платформи завдяки своїм перевагам.

Tailwind CSS надає набір утилітарних класів, які можна використовувати для стилізації елементів без написання власного CSS-коду. Це дозволяє розробникам швидко створювати інтерфейси користувача, використовуючи готові класи для налаштування розмірів, кольорів, шрифтів та інших стилів. У нашій платформі це забезпечує швидку і зручну розробку інтерфейсу, що підвищує продуктивність команди.

Tailwind CSS підтримує адаптивний дизайн завдяки класам для різних розмірів екранів. Це дозволяє створювати інтерфейси, які автоматично адаптуються до різних пристроїв, таких як смартфони, планшети та десктопи. У нашій платформі це забезпечує зручність користувачів незалежно від пристрою, з якого вони отримують доступ до сайту.

Tailwind CSS дозволяє налаштовувати теми і розширювати базові стилі відповідно до потреб проекту. Це забезпечує гнучкість у створенні унікального дизайну, який відповідає вашим вимогам. У нашій платформі це дозволяє створювати індивідуальний і привабливий інтерфейс, що підвищує загальний досвід користувачів.

Використання утилітарних класів дозволяє швидко створювати і змінювати інтерфейси без необхідності писати багато CSS-коду. Це скорочує час розробки і

забезпечує високу продуктивність. У розроблюваній платформі це дозволяє швидко впроваджувати нові функції і покращення, що відповідають вимогам користувачів.

Tailwind CSS легко інтегрується з JavaScript фреймворками, такими як React, що робить його зручним інструментом для розробки сучасних веб-додатків. У нашій платформі це забезпечує єдність технологій і спрощує розробку інтерактивних і динамічних інтерфейсів.

2.8.JSON Web Token (JWT)

У розроблюваній веб-платформі для онлайн оголошень з нерухомості JSON Web Token (JWT) використовується для забезпечення безпеки користувачів та їх даних. JWT є стандартом для створення токенів доступу, які можуть бути використані для аутентифікації та авторизації користувачів у веб-додатках.

JWT забезпечує безпечний спосіб передачі інформації між сторонами у вигляді JSON-об'єктів. Це дозволяє аутентифікувати користувачів та надавати доступ до захищених ресурсів без необхідності зберігати сесии на сервері. У нашій платформі це забезпечує безпечну аутентифікацію користувачів, що знижує ризик несанкціонованого доступу до даних.

Використання JWT дозволяє створювати безпечні механізми аутентифікації, які не потребують зберігання сесии на сервері. Це забезпечує високу продуктивність і масштабованість додатків, дозволяючи їм легко інтегруватися з іншими сервісами та API. У нашій платформі це дозволяє обробляти велику кількість одночасних запитів без зниження продуктивності.

JWT легко інтегрується з різними мовами програмування та фреймворками, що робить його зручним інструментом для розробки веб-додатків. У нашій платформі JWT використовується для забезпечення аутентифікації користувачів, дозволяючи їм безпечно входити до системи і взаємодіяти з додатком.

JWT підтримує різні типи аутентифікації, включаючи OAuth2, що дозволяє використовувати його в різних сценаріях. У нашій платформі це забезпечує

гнучкість у виборі методів аутентифікації та дозволяє легко інтегрувати додаток з іншими сервісами.

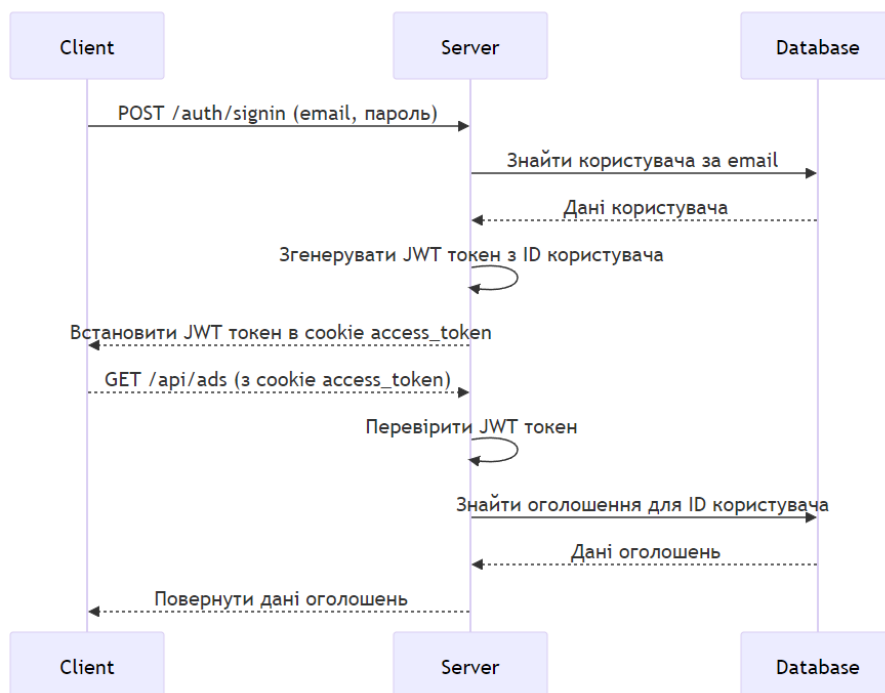


Рис 2.6 Послідовності використання JWT токену

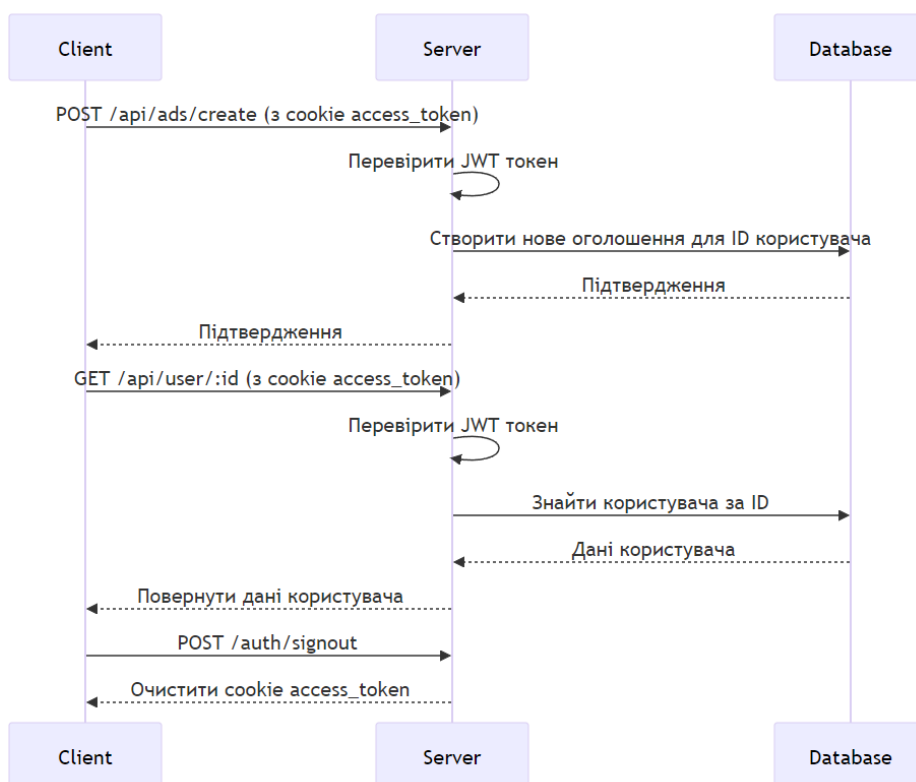


Рис 2.7 Послідовності використання JWT токену

На діаграмах наведена послідовність використання JWT токену:

1. Рис 2.6 Послідовності використання JWT токену:

- Користувач надсилає POST-запит на `/auth/signin` з email і паролем.
- Сервер знаходить користувача за email, генерує JWT токен з ID користувача і встановлює його в cookie `access_token`.
- Користувач надсилає GET-запит на `/api/ads`, сервер перевіряє JWT токен, знаходить оголошення для ID користувача і повертає дані оголошень.

2. Рис 2.7 Послідовності використання JWT токену:

- Користувач надсилає POST-запит на `/api/ads/create` з JWT токеном в cookie.
- Сервер перевіряє JWT токен, створює нове оголошення для ID користувача і повертає підтвердження.
- Користувач надсилає GET-запит на `/api/user/`
- , сервер перевіряє JWT токен, знаходить користувача за ID і повертає дані користувача.
- Користувач надсилає POST-запит на `/auth/signout`, сервер очищає cookie з `access_token`.

Ці діаграми ілюструють, як JWT використовується для аутентифікації та авторизації в нашій платформі, забезпечуючи безпечний і ефективний доступ до даних.

2.9. Vite

Vite є інструментом для розробки веб-додатків, який забезпечує швидку збірку та гаряче перезавантаження. У нашій веб-платформі Vite використовується для забезпечення швидкого і ефективного процесу розробки. Він був розроблений Еваном Ю у 2020 році як альтернатива Webpack для забезпечення більшої продуктивності та зручності розробки.

Vite використовує нативні модулі ES для забезпечення швидкої збірки та оптимізації додатків. Це дозволяє розробникам швидко перевіряти зміни в коді без тривалого часу на збірку. У нашій платформі це скорочує час розробки і підвищує продуктивність команди.

Vite забезпечує гаряче перезавантаження модулів, що дозволяє миттєво оновлювати додаток при внесенні змін до коду. Це забезпечує безперервний процес розробки, дозволяючи розробникам бачити результати своїх змін у реальному часі.

Vite підтримує різні фреймворки, такі як React, Vue і Svelte, що дозволяє розробникам використовувати його у будь-яких проектах. У нашій платформі Vite інтегрується з React, забезпечуючи швидку і ефективну розробку інтерфейсу користувача.

Vite має мінімалістичний підхід до конфігурації, що дозволяє розробникам швидко налаштувати середовище розробки з мінімальною кількістю конфігурації. У нашій платформі це забезпечує швидкий старт розробки та зменшує кількість потенційних помилок.

Vite забезпечує оптимізацію додатків для продакшн середовища, включаючи мінімізацію коду і видалення невикористовуваних модулів. Це забезпечує високу продуктивність і зменшує розмір додатків, що підвищує швидкість їх завантаження та роботи.

2.10. Postman для тестування API

Postman є одним з найпопулярніших інструментів для тестування API, який використовується для розробки, тестування та документування API. У розроблюваній веб-платформі для онлайн оголошень з нерухомості Postman використовується для тестування серверної частини додатку, що допомагає забезпечити коректну роботу API.

Postman має простий і зрозумілий інтерфейс, який дозволяє швидко створювати і надсилати запити до API. Це дозволяє розробникам легко тестувати різні кінцеві точки API і перевіряти їх роботу.

Postman дозволяє створювати автоматизовані тести для API, що забезпечує постійне тестування та виявлення помилок на ранніх стадіях розробки. Це підвищує якість додатку і зменшує ризик виникнення проблем у продакшн середовищі.

Postman надає можливість створювати документацію для API, що спрощує процес ознайомлення нових розробників з проектом. Це забезпечує легкий доступ до інформації про кінцеві точки API, параметри запитів та відповіді.

Postman дозволяє зберігати запити в колекціях, що дозволяє організувати тестування та спрощує спільну роботу в команді. Кожен член команди може легко знайти і використовувати необхідні запити для тестування.

Postman може інтегруватися з системами безперервної інтеграції та доставки (CI/CD), що забезпечує автоматичне тестування API під час розгортання нових версій додатку. Це підвищує надійність та стабільність роботи додатку.

3. РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОСТІ ВЕБ-ПЛАТФОРМИ

3.1 Розробка ключових функцій веб-сервісу для оголошень нерухомості

Аналіз процесу розміщення оголошень про нерухомість та існуючих програмних рішень дозволяє сформулювати вимоги до майбутнього програмного продукту. Веб-платформа для оголошень нерухомості повинна забезпечувати наступні функції:

Реєстрація та авторизація користувача, у тому числі за допомогою Google. Система повинна надавати можливість реєстрації нових користувачів з введенням особистих даних. Для існуючих користувачів забезпечується авторизація через електронну пошту та пароль. Крім того, система має інтеграцію з Google, що дозволяє користувачам використовувати свої облікові записи Google для реєстрації та входу в систему. Користувачі можуть увійти в систему, щоб переглядати свої оголошення, редагувати свій профіль та користуватися іншими функціями платформи.

Відображення каталогу оголошень. Користувачі можуть переглядати список доступних оголошень про нерухомість. Інтерфейс повинен бути зручним та інтуїтивно зрозумілим, з можливістю перегляду фото, відео з YouTube та основної інформації про об'єкти нерухомості.

Пошук та фільтрація оголошень. Користувачі можуть шукати конкретні типи нерухомості за допомогою форми пошуку на головній сторінці. Пошук підтримує фільтри за типом оголошення (оренда, продаж), наявністю меблів, парковки, кількістю кімнат та іншими параметрами. Це дозволяє швидко знайти необхідну інформацію та зменшує час на пошук.

Сортування результатів пошуку. Користувачі можуть сортувати оголошення за ціною, кількістю кімнат, знижкою або датою створення, що допомагає знайти найбільш підходящі пропозиції.

Додавання оголошень. Авторизовані користувачі можуть створювати нові оголошення про нерухомість, заповнюючи детальну інформацію про об'єкт. Вони можуть завантажувати зображення та відео з YouTube, щоб надати більш повну інформацію про нерухомість. Крім того, користувачі можуть редагувати свої оголошення для внесення змін або оновлення інформації.

Коментарі та оцінки. Користувачі можуть залишати коментарі та оцінки до оголошень. Це створює систему зворотного зв'язку, яка допомагає іншим користувачам приймати рішення на основі досвіду попередніх покупців або орендарів. Відгуки та рейтинги сприяють підвищенню довіри до платформи.

Зміна паролю, нікнейму та електронної пошти. Користувачі мають можливість змінювати свій пароль, нікнейм та електронну пошту через налаштування профілю. Це забезпечує гнучкість та безпеку для користувачів.

Можливість видалення облікового запису. Користувачі можуть видалити свій акаунт, якщо вони більше не бажають використовувати платформу. Це забезпечує користувачам контроль над їхніми даними та конфіденційністю.

Відображення банерів з інформацією про зниження ціни. Система повинна відображати банери з інформацією про знижки на об'єкти нерухомості, щоб привернути увагу користувачів до вигідних пропозицій.

3.1.1. Реєстрація та авторизація користувача

У веб-сервісі для оголошень про нерухомість реалізована функціональність реєстрації та авторизації користувачів, що дозволяє новим користувачам створювати облікові записи, а вже зареєстрованим користувачам – входити в систему.

Реєстрація користувача

Для реєстрації користувача необхідно натиснути кнопку "Зареєструватися", яка зображена на рисунку 3.1.

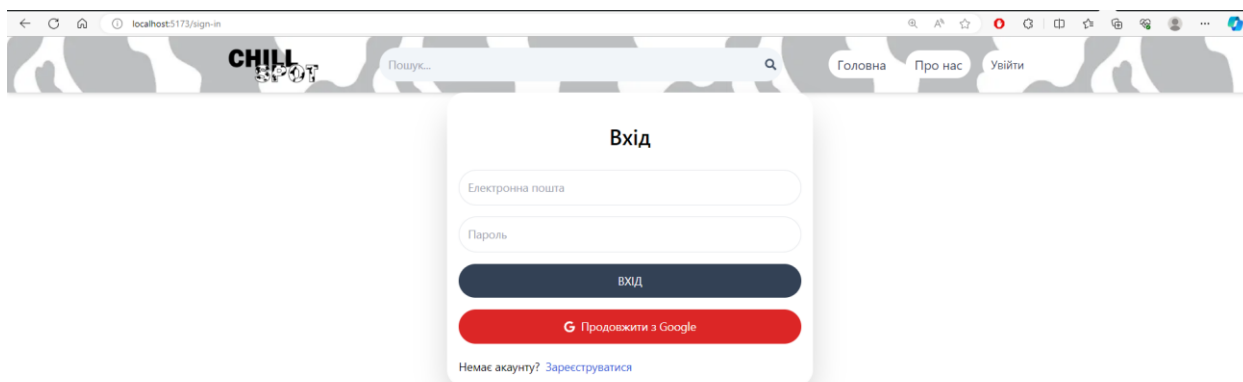


Рис 3.1 Реєстрація та авторизація користувача

Після цього відбувається перехід на сторінку реєстрації, що зображена на рисунку 3.2, де користувач повинен заповнити форму з наступними полями:

- Ім'я користувача
- Електронна пошта
- Пароль

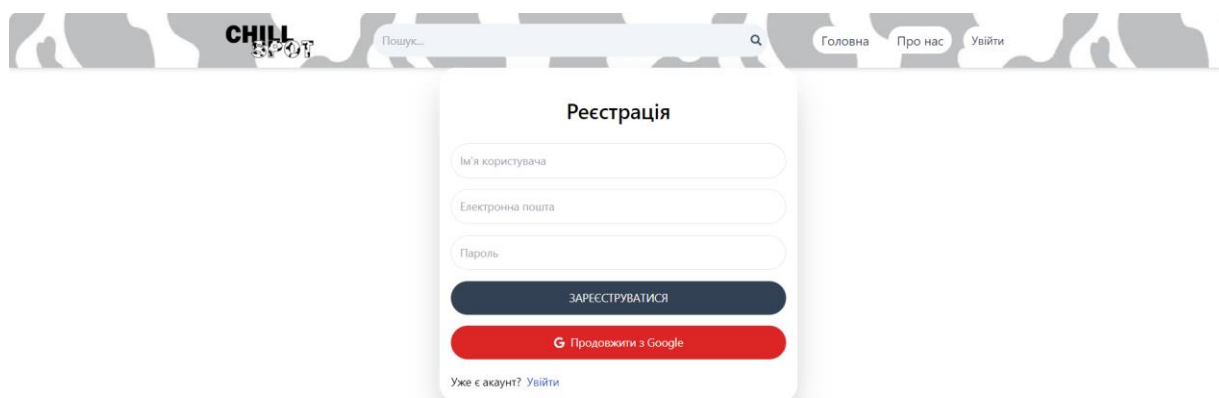


Рис 3.2 Реєстрація нового користувача

Поля "Ім'я користувача", "Електронна пошта", і "Пароль" є обов'язковими для заповнення. У разі, якщо введені паролі не співпадають, з'являється повідомлення про помилку та користувач повинен ввести паролі ще раз. Після заповнення всіх обов'язкових полів та натиснення кнопки "Зареєструватися", інформація про нового користувача зберігається в базі даних. Крім того, користувач може побачити інформацію про свій обліковий запис, а також редагувати його, натиснувши на відповідні кнопки у верхньому правому куті сторінки.

Авторизація користувача

Для того, щоб авторизуватися зареєстрованому користувачу потрібно натиснути на кнопку "Увійти" у правому кутку сторінки, після чого для входу на сторінці заповнити поля "Електронна пошта" та "Пароль", які зображені на рисунку 3.2.

Для авторизації використовується механізм аутентифікації на основі токенів, який забезпечує безпеку передачі даних між клієнтом та сервером. Користувач може увійти до системи, використовуючи електронну пошту та пароль. Авторизація користувача реалізована за допомогою JWT токенів, які генеруються під час входу та зберігаються у cookies.

Користувач також може увійти за допомогою свого облікового запису Google. Для цього необхідно натиснути кнопку "Продовжити з Google", після чого користувач буде перенаправлений на сторінку вибору облікового запису Google (Рисунок 3.3).

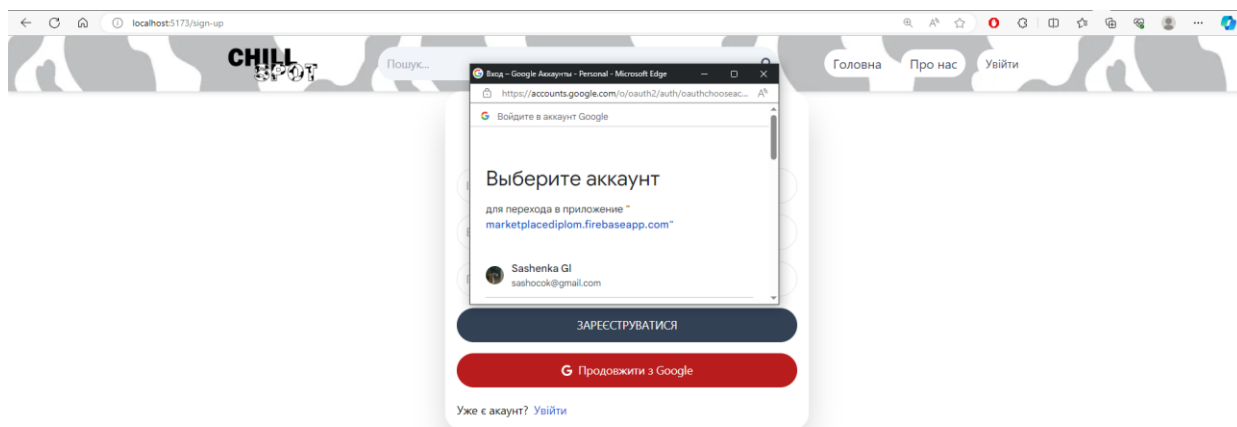


Рис 3.3 Авторизація через обліковий запис Google

Основні функції

- **Реєстрація та авторизація користувачів через електронну пошту та Google:** користувачі можуть зареєструватися та увійти в систему за допомогою електронної пошти та пароля, а також використовувати обліковий запис Google для швидкої авторизації.
- **Управління профілем:** користувачі можуть змінювати свої дані (пароль, нікнейм, електронну пошту) через налаштування профілю.
- **Захист даних:** забезпечується безпечна авторизація та автентифікація користувачів з використанням JWT токенів для захисту сесій користувачів.

Ця система забезпечує зручність та безпеку для користувачів, дозволяючи їм легко створювати облікові записи, входити до системи та керувати своїми даними.

3.1.2. Головна сторінка веб-платформи

На головній сторінці веб-платформи Chill Spot користувачі можуть знайти оголошення про оренду та продаж нерухомості. Головна сторінка структурована

таким чином, щоб надати користувачам зручний доступ до ключових функцій платформи та найпопулярніших оголошень.

Основні елементи головної сторінки

1. Верхня навігаційна панель:

— Верхня навігаційна панель містить кнопки для переходу на інші сторінки платформи, такі як "Головна", "Про нас" та "Увійти". Також є поле пошуку, де користувачі можуть ввести ключові слова для швидкого знаходження потрібних оголошень.

2. Заголовок та опис платформи:

— Під навігаційною панеллю розташований заголовок "Знайдіть пейзаж своєї мрії" з коротким описом платформи. Це дозволяє новим користувачам швидко зрозуміти призначення платформи та її основні функції.

3. Гарячі Пропозиції:

— Розділ з гарячими пропозиціями демонструє найвигідніші та найбільш популярні оголошення. Тут користувачі можуть переглянути обрані пропозиції з великими знижками.

4. Чудові Місця для Оренди:

— Цей розділ містить оголошення про оренду нерухомості з привабливими описами та фотографіями. Кожне оголошення містить інформацію про ціну, кількість кімнат та місце розташування.

5. Чудові Місця для Продажу:

— Розділ з оголошеннями про продаж нерухомості дозволяє користувачам переглядати доступні об'єкти для купівлі. Як і в розділі оренди, кожне оголошення містить деталі про ціну, кількість кімнат та іншу важливу інформацію.

6. Інтерактивні елементи:

- Користувачі можуть використовувати різні інтерактивні елементи, такі як фільтри, для сортування оголошень за ціною, кількістю кімнат, знижками або датою створення. Також є можливість переглядати банери з інформацією про знижки та спеціальні пропозиції.

Візуалізація

На головній сторінці використано яскраві та привабливі зображення нерухомості, які допомагають користувачам візуалізувати потенційні об'єкти для оренди або купівлі. Структура сторінки та її вміст адаптовані для зручного перегляду на різних пристроях, включаючи десктопи, планшети та смартфони.

Ця організація та функціональність головної сторінки веб-платформи Chill Spot забезпечують зручність використання та сприяють швидкому доступу користувачів до потрібної інформації про нерухомість.

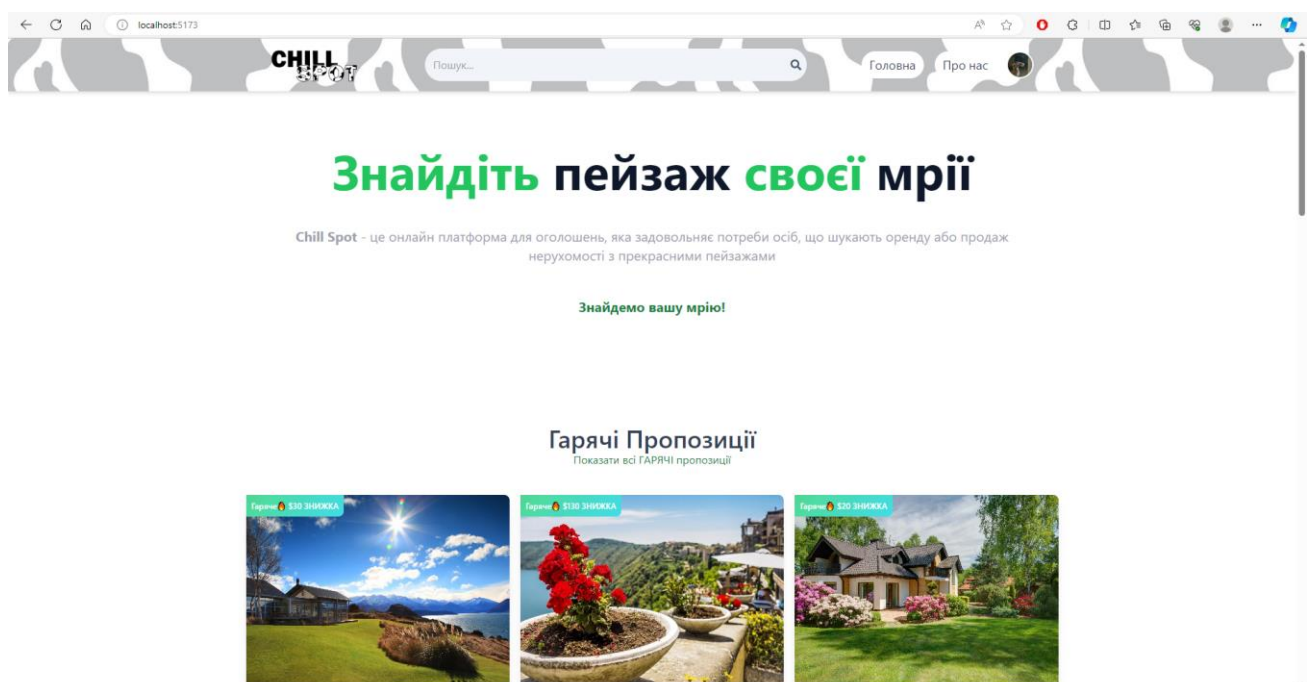


Рис 3.4 Головна сторінка

3.1.3 Сторінка оголошення для зареєстрованого користувача

На веб-платформі для нерухомості, сторінка оголошення для зареєстрованого користувача надає розширену функціональність та можливості взаємодії з оголошенням. Вона включає детальну інформацію про об'єкт нерухомості, коментарі інших користувачів та можливість додавання власних коментарів.

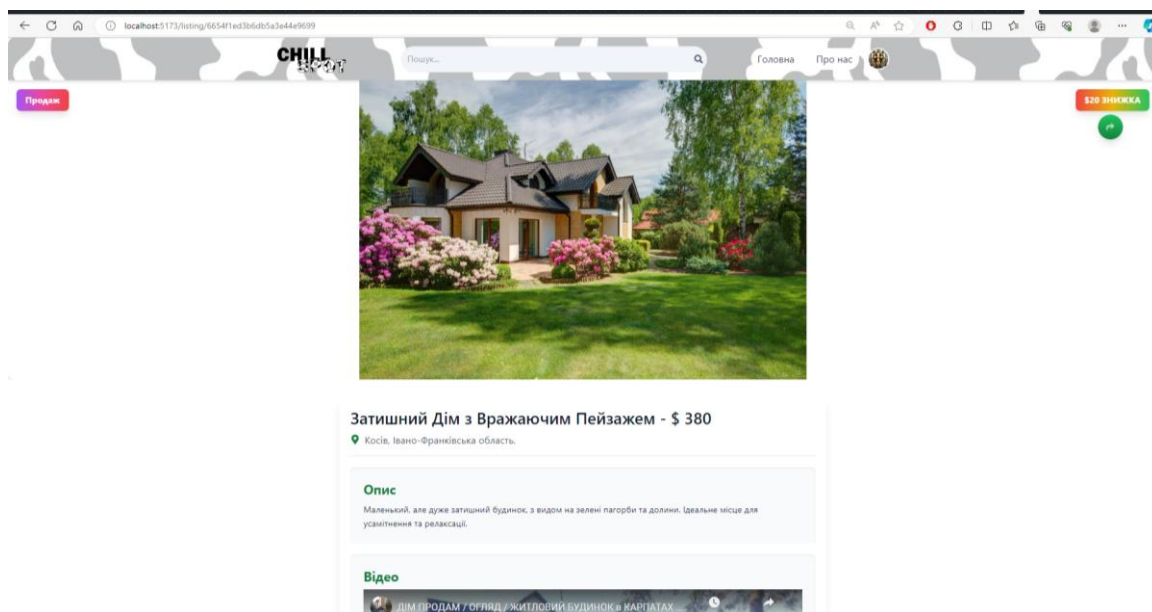


Рис 3.4 Сторінка оголошення

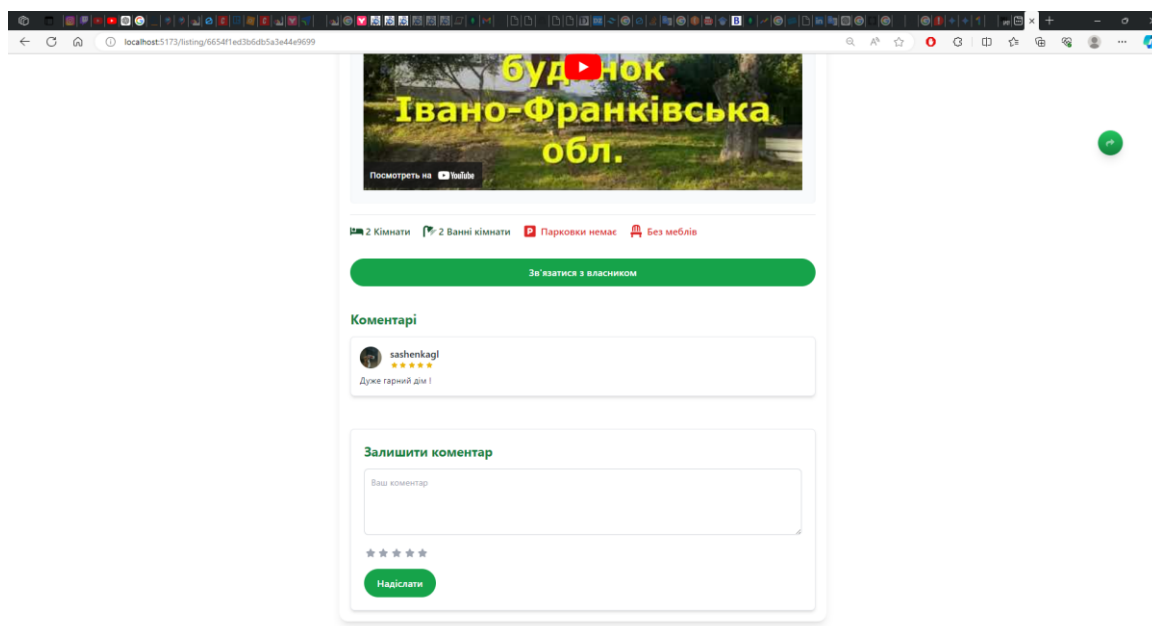


Рис 3.5 Сторінка оголошення

Основні елементи сторінки оголошення:

1. Зображення та Заголовок:

- На верхній частині сторінки розміщене велике зображення нерухомості, що привертає увагу.
- Під зображенням розташований заголовок з назвою нерухомості та її вартістю, наприклад, "Затишний Дім з Вражаючим Пейзажем - \$380".
- Також відображається статус оголошення (продаж або оренда) у вигляді позначки.

2. Опис:

- Блок з текстовим описом нерухомості, де зазначені основні характеристики та переваги об'єкта. Наприклад, "Маленький, але дуже затишний будинок, з видом на зелені пагорби та долини. Ідеальне місце для усамітнення та релаксації."

3. Відеоогляд:

- Відео, прикріплене до оголошення, дозволяє переглянути детальний огляд нерухомості.
- Відео вбудоване з платформи YouTube і може бути переглянуте безпосередньо на сторінці оголошення.

4. Додаткові Характеристики:

- Під відеооглядом відображаються іконки з додатковою інформацією про нерухомість: кількість кімнат, ванних кімнат, наявність парковки та меблів.

5. Коментарі та Оцінки:

- Зареєстровані користувачі можуть переглядати та залишати коментарі та оцінки до оголошення.
- Коментарі відображаються з інформацією про автора коментаря, рейтинг (у вигляді зірочок) та текст коментаря.

6. Форма для Додавання Коментаря:

- Під блоком коментарів знаходиться форма для додавання нового коментаря.
- Користувач може ввести текст коментаря, обрати рейтинг (від однієї до п'яти зірок) та натиснути кнопку "Надіслати".

7. Кнопка для Зв'язку з Власником:

- Зареєстровані користувачі мають можливість зв'язатися з власником оголошення через кнопку "Зв'язатися з власником", розташовану під блоком додаткових характеристик.

— Опис сторінки власного профіля користувача

Сторінка профіля користувача у веб-платформі нерухомості є важливим елементом, що дозволяє користувачам управляти своїми даними та оголошеннями. На ній реалізовані наступні функції:

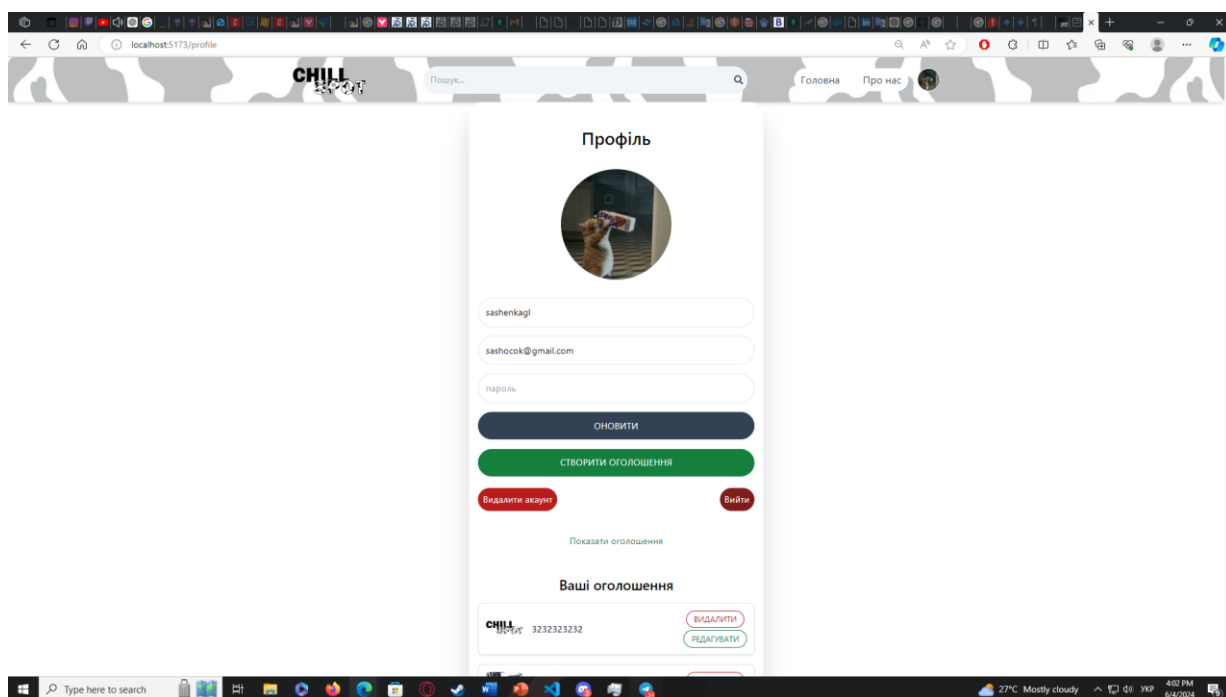


Рис 3.6 Сторінка особистого профілю

1. Особиста інформація:

- **Аватар користувача:** Відображається у верхній частині сторінки.
- **Ім'я користувача:** Поле для введення імені, що відображається на сторінці.

- **Електронна пошта:** Поле для введення або редагування електронної пошти.
- **Пароль:** Поле для зміни пароля.

2. Оновлення профіля:

- Кнопка "ОНОВИТИ" дозволяє користувачам зберегти зміни в особистих даних.

3. Управління оголошеннями:

- **Створення оголошення:** Кнопка "СТВОРИТИ ОГОЛОШЕННЯ" для додавання нових оголошень про нерухомість.
- **Видалення облікового запису:** Кнопка "Видалити акаунт" дозволяє користувачу повністю видалити свій обліковий запис з системи.
- **Вихід:** Кнопка "Вийти" для завершення сесії користувача.

4. Перегляд оголошень:

- Список оголошень користувача відображається нижче основних даних профіля.
- Кожне оголошення має кнопки "ВИДАЛИТИ" і "РЕДАГУВАТИ", що дозволяє легко управляти існуючими оголошеннями.

Ця сторінка забезпечує зручний доступ до ключових функцій для управління обліковим записом та оголошеннями, підтримуючи гнучкість і зручність користування платформою.

3.1.4 Сторінки створення та оновлення оголошення

На веб-платформі для оголошень про нерухомість користувачі мають можливість створювати та редагувати оголошення за допомогою відповідних сторінок. Обидві сторінки мають схожий інтерфейс і функціонал, що полегшує їх використання. Ось детальний опис функцій кожної сторінки.

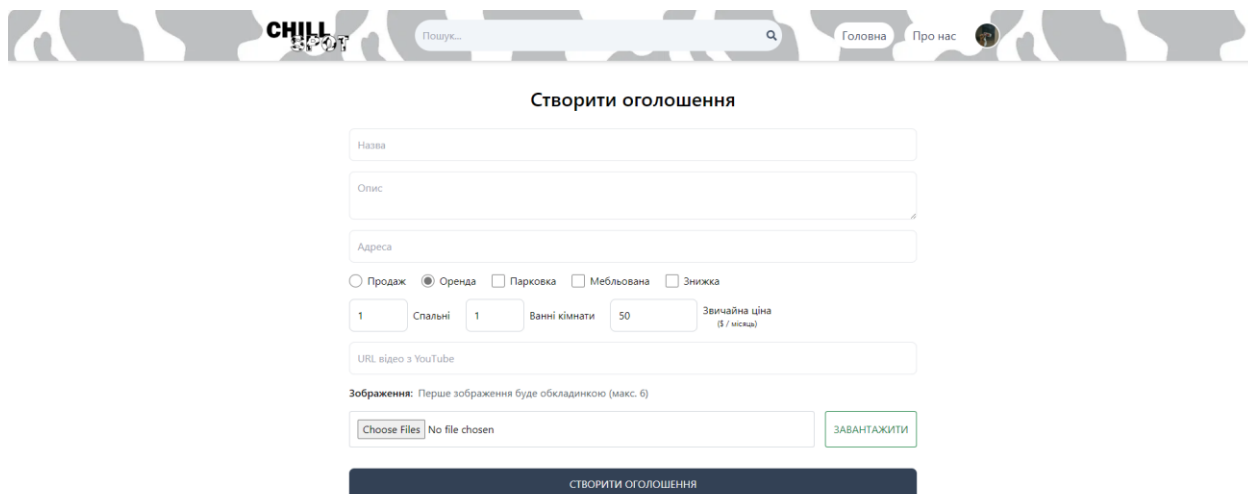


Рис 3.7 Сторінка створення оголошення

Ця сторінка дозволяє користувачам створювати нові оголошення про нерухомість. Основні елементи та функції:

- **Назва:** Поле для введення назви оголошення.
- **Опис:** Поле для детального опису об'єкта нерухомості.
- **Адреса:** Поле для введення адреси нерухомості.
- **Тип оголошення:** Користувач може вибрати тип оголошення - продаж або оренда.
- **Додаткові параметри:** Чекбокси для зазначення наявності парковки, меблів та знижки.
- **Кількість спальних та ванних кімнат:** Поля для введення кількості спальних та ванних кімнат.
- **Ціна:** Поле для введення звичайної ціни та ціни зі знижкою (якщо є).
- **URL відео з YouTube:** Поле для введення посилання на відеоогляд нерухомості з YouTube.
- **Зображення:** Можливість завантаження до шести зображень об'єкта нерухомості. Перше зображення буде обкладинкою оголошення.

- Кнопка "Завантажити": Дозволяє користувачам вибрати та завантажити файли зображень.
- Кнопка "Створити оголошення": Після заповнення всіх полів користувач може натиснути цю кнопку для створення оголошення.

Рис 3.8 Сторінка оновлення оголошення

Сторінка оновлення оголошення дозволяє користувачам редагувати існуючі оголошення. Основні елементи та функції:

- **Назва:** Поле для редагування назви оголошення.
- **Опис:** Поле для редагування детального опису об'єкта нерухомості.
- **Адреса:** Поле для редагування адреси нерухомості.
- **Тип оголошення:** Можливість змінити тип оголошення - продаж або оренда.
- **Додаткові параметри:** Чекбокси для редагування наявності парковки, меблів та знижки.
- **Кількість спальних та ванних кімнат:** Поля для редагування кількості спальних та ванних кімнат.
- **Ціна:** Поле для редагування звичайної ціни та ціни зі знижкою (якщо є).

- **URL відео з YouTube:** Поле для редагування посилання на відеоогляд нерухомості з YouTube.
- **Зображення:** Можливість завантаження нових зображень об'єкта нерухомості та видалення існуючих.
- **Кнопка "Завантажити":** Дозволяє користувачам вибрати та завантажити нові файли зображень.
- **Кнопка "Оновити оголошення":** Після внесення всіх змін користувач може натиснути цю кнопку для збереження оновлень.

Ці сторінки забезпечують користувачам необхідні інструменти для ефективного управління їх оголошеннями, надаючи інтуїтивний інтерфейс і функціональність для створення та редагування оголошень про нерухомість.

Сторінка пошуку на веб-платформі для оголошень нерухомості призначена для зручного та швидкого знаходження необхідних об'єктів користувачами. На сторінці пошуку користувачі можуть скористатися різноманітними фільтрами та опціями для деталізації свого запиту. На рис. 3.1 наведено приклад інтерфейсу сторінки пошуку.

Основні компоненти сторінки пошуку:

1. Поле для текстового пошуку:

- Розташоване у верхній частині сторінки, дозволяє вводити ключові слова для швидкого пошуку оголошень за назвою або описом.

2. Фільтри пошуку:

- **Тип оголошення:** Можливість вибору між орендою, продажем або обома варіантами.
- **Знижка:** Відображення оголошень зі знижкою.
- **Зручності:** Вибір оголошень з певними зручностями, такими як наявність парковки або меблів.
- **Кількість спалень та ванних кімнат:** Дозволяє вибрати об'єкти з конкретною кількістю кімнат.

- **Сортування:** Сортування результатів за ціною (від високої до низької або від низької до високої), кількістю кімнат, датою створення тощо.

3. Результати пошуку:

- Після застосування фільтрів і введення ключових слів користувач отримує список оголошень, які відповідають критеріям пошуку.
- Кожне оголошення містить коротку інформацію про об'єкт: фотографію, назву, місцезнаходження, ціну, кількість кімнат та інші характеристики.

Функціональність сторінки пошуку:

- **Інтерактивні фільтри:** Користувачі можуть динамічно змінювати фільтри та бачити оновлені результати без необхідності перезавантаження сторінки.
- **Візуалізація результатів:** Оголошення відображаються у вигляді карток з можливістю перегляду детальної інформації при натисканні.
- **Зручність користування:** Інтуїтивно зрозумілий інтерфейс дозволяє швидко знаходити необхідні оголошення та адаптувати пошук під свої потреби.

На рисунку 3.9 показана сторінка пошуку з активованими фільтрами та результатами, що відповідають запиту користувача. Сторінка пошуку є важливим елементом веб-платформи, який забезпечує ефективний і зручний спосіб знаходження нерухомості для оренди або купівлі.

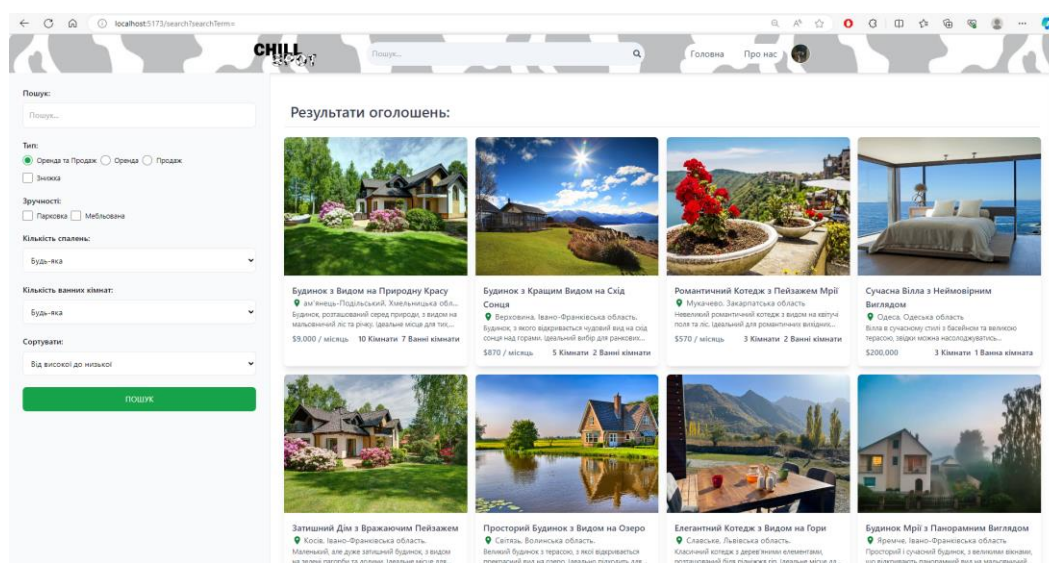


Рис 3.9 Сторінка пошуку

3.2. Розробка структури бази даних та моделей

Структура бази даних для веб-платформи нерухомості розроблена з урахуванням специфічної функціональності сервісу. Основними компонентами бази даних є таблиці для збереження даних про користувачів, оголошення, та коментарі. Нижче наведені основні таблиці та їх взаємозв'язки, відповідно до діаграм:

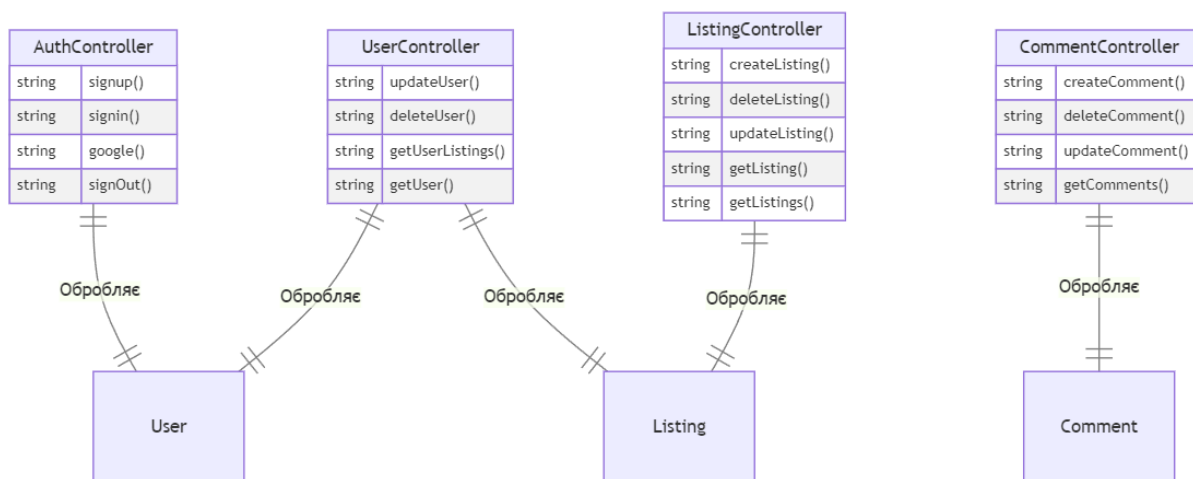


Рис 3.10 Діаграма сутність-зв'язок з контролерами

Основні таблиці та їх поля:

1. Таблиця користувачів (User):

- **id**: унікальний ідентифікатор користувача (PK).
- **username**: ім'я користувача.
- **email**: електронна пошта користувача.
- **password**: хеш пароля користувача.
- **avatar**: зображення аватара користувача.
- **createdAt**: дата створення облікового запису.
- **updatedAt**: дата останнього оновлення облікового запису.

2. Таблиця оголошень (Listing):

- **id**: унікальний ідентифікатор оголошення (PK).
- **name**: назва оголошення.
- **description**: опис оголошення.
- **address**: адреса об'єкта нерухомості.
- **regularPrice**: звичайна ціна об'єкта.
- **discountPrice**: ціна зі знижкою (якщо є).
- **bathrooms**: кількість ванних кімнат.
- **bedrooms**: кількість спальних кімнат.
- **furnished**: наявність меблів.
- **parking**: наявність парковки.
- **type**: тип оголошення (оренда або продаж).
- **offer**: наявність спеціальної пропозиції.
- **imageUrls**: масив URL зображень.
- **userRef**: зовнішній ключ користувача (FK).
- **createdAt**: дата створення оголошення.
- **updatedAt**: дата останнього оновлення оголошення.
- **videoUrl**: URL відеоогляду з YouTube.

3. Таблиця коментарів (Comment):

- **id**: унікальний ідентифікатор коментаря (PK).
- **content**: текст коментаря.
- **userId**: ідентифікатор користувача, що залишив коментар (FK).
- **listingId**: ідентифікатор оголошення, до якого залишений коментар (FK).
- **createdAt**: дата створення коментаря.
- **updatedAt**: дата останнього оновлення коментаря.

Контролери та їх методи:

1. AuthController:

- `signup()`: метод для реєстрації нових користувачів.

- `signin()`: метод для авторизації користувачів.
- `google()`: метод для авторизації користувачів через Google.
- `signOut()`: метод для виходу користувачів із системи.

2. **UserController:**

- `updateUser()`: метод для оновлення даних користувача.
- `deleteUser()`: метод для видалення облікового запису користувача.
- `getUserListings()`: метод для отримання всіх оголошень користувача.
- `getUser()`: метод для отримання даних про користувача.

3. **ListingController:**

- `createListing()`: метод для створення нового оголошення.
- `deleteListing()`: метод для видалення оголошення.
- `updateListing()`: метод для оновлення оголошення.
- `getListing()`: метод для отримання даних про конкретне оголошення.
- `getListings()`: метод для отримання списку всіх оголошень.

4. **CommentController:**

- `createComment()`: метод для створення нового коментаря.
- `deleteComment()`: метод для видалення коментаря.
- `updateComment()`: метод для оновлення коментаря.
- `getComments()`: метод для отримання списку коментарів.

Взаємозв'язки між таблицями:

- Користувач (`User`) **створює** Оголошення (`Listing`), яке зберігає ID користувача як зовнішній ключ (`userRef`).
- Користувач (`User`) залишає Коментар (`Comment`), який зберігає ID користувача (`userId`) та ID оголошення (`listingId`) як зовнішні ключі.

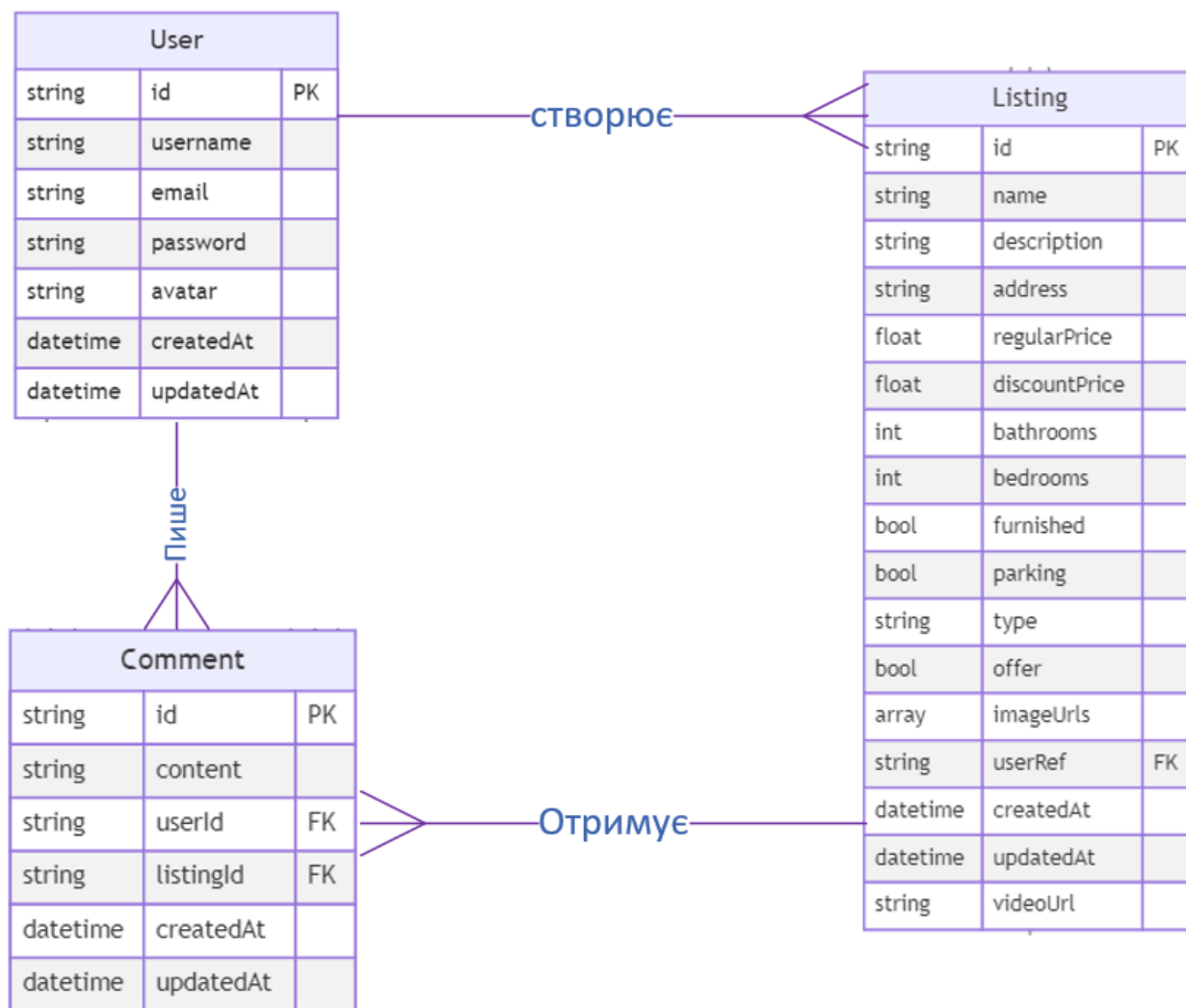


Рис 3.11 Діаграма сутність-зв'язок

Ця модель відображає структуру бази даних веб-платформи нерухомості та відповідає вимогам до зберігання даних про користувачів, оголошення, та коментарі, забезпечуючи цілісність даних та зручний доступ до інформації.

4. ТЕСТУВАННЯ ВЕБ ПЛАТФОРМИ

Метою тестування програмного забезпечення є підтвердження того, що його фактична поведінка відповідає очікуваній. Це здійснюється шляхом виконання певних тестів з використанням заздалегідь визначених наборів даних та параметрів. Тестування є одним із методів забезпечення якості програмного забезпечення. У цьому розділі обговорюються ключові аспекти тестування веб-сервісів, розробляються тест-кейси для перевірки функціональності додатку, і проводиться тестування згідно з планом тестування.

4.1. Визначення об'єктів тестування веб-сервісу

Основним завданням тестування веб-сервісу є перевірка його коректної роботи в нормальних умовах використання, а також правильної реакції на випадки, що відхиляються від звичайних сценаріїв користувача. Тестування веб-сервісів має свої особливості, які відрізняють його від тестування звичайних десктопних та мобільних додатків. У таблиці 4.1 наведені ключові характеристики, які потребують тестування в веб-сервісі, з урахуванням його функціональних можливостей.

Таблиця 4.1

Ключові характеристики тестування

Назва характеристики	Вимоги до реалізації у веб-сервісі
Перевірка основних функцій	Тестування реєстрації та авторизації користувачів, створення та редагування оголошень, пошуку і фільтрації оголошень, залишення коментарів тощо.
Обробка різних типів запитів та відповідей	Тестування правильності обробки запитів GET, POST, PUT, DELETE та повернення відповідних відповідей згідно зі специфікаціями.

Продовження таблиці 4.1

Ключові характеристики тестування

Назва характеристики	Вимоги до реалізації у веб-сервісі
Обробка помилок	Перевірка коректної обробки помилок, відображення відповідних повідомлень.
Аутентифікація та авторизація	Перевірка процесу аутентифікації користувачів, ролей та прав доступу.
Робота з базами даних	Тестування збереження, оновлення та видалення даних у базі даних.
Співвідношення розміру екрану і елементів інтерфейсу	Перевірка адаптивності інтерфейсу для різних розмірів екранів і пристроїв.

4.2. Створення тест-кейсів

Функціональні можливості веб-платформи тестуються методом "чорної скриньки". Цей підхід передбачає перевірку функціональних специфікацій та вимог без доступу до внутрішньої структури коду чи бази даних. На рівні звичайного користувача вводяться конкретні набори даних, результати яких порівнюються з очікуваними. Кожен тест-кейс описує послідовність дій, необхідних для досягнення певної мети, а також включає набори тестових даних, що відповідають визначеним критеріям покриття.

Тестові сценарії (тест-кейси) створюються для перевірки відповідності функцій системи встановленим вимогам. В залежності від очікуваного результату, тест-кейси поділяються на позитивні та негативні:

- **Позитивний тест-кейс:** Використовує лише правильні дані для перевірки функцій додатку, чи виконує він своє завдання.
- **Негативний тест-кейс:** Оперує як правильними, так і неправильними даними з метою перевірки обробки виняткових ситуацій та впевненості в тому, що функція не виконується при некоректних даних.

Структура тест-кейсу включає розділи "Action" (дії або послідовність дій, що виконуються під час тестування), "Expected Result" (очікуваний результат) та "Test Result" (фактичні результати роботи функції, позначені як "пройдено/не пройдено/заблоковано").

Таблиця 4.2

Тест-кейси для функціонального тестування

№ тест-кейсу	ОК/НОК	Дії	Очікуваний результат
1	ОК	<ol style="list-style-type: none"> 1. Пройти етап авторизації користувача. 2. Натиснути на кнопку "Додати нове оголошення". 3. Заповнити форму для створення оголошення. 4. Створити оголошення, натиснувши на кнопку "Create". 	<ol style="list-style-type: none"> 1. Відкривається форма для заповнення інформації про оголошення. 2. Усі поля для вводу доступні 3. Нове оголошення успішно додається до системи та відображається у списку оголошень на головній сторінці.
2	ОК	<ol style="list-style-type: none"> 1. Ввести ключове слово у поле "Пошук" та натиснути на значок пошуку. 	<ol style="list-style-type: none"> 1. Відображаються результати пошуку, що відповідають введеному ключовому слову.
3	ОК	<ol style="list-style-type: none"> 1. Пройти етап авторизації користувача. 2. Зі списку оголошень вибрати одне та додат коментар до оголошення 	<ol style="list-style-type: none"> 1. Відкривається сторінка з оголошеннями користувача. 2. Створений коментар відображається.

Юзабіліті-тестування

№ тест-кейсу	ОК/НОК	Дії	Очікуваний результат
1	ОК	1. Оглянути текстові елементи інтерфейсу, такі як заголовки, кнопки, посилання.	1. Текст повинен бути легко читабельним, зрозумілим та відповідати діям елементів, за якими він закріплений.
2	ОК	1. Оцінити розміщення елементів, таких як кнопки, поля введення, списки тощо.	1. Елементи повинні бути розміщені логічно та зручно для користувача, без перенасичення або зайвих відступів.
3	ОК	1. Спробувати перейти до різних розділів інтерфейсу за допомогою навігаційних елементів (меню, посилання).	1. Навігація повинна бути зручною та інтуїтивно зрозумілою, користувач повинен легко знайти потрібний розділ.

Тестування безпеки веб-сервісу

№ тест-кейсу	ОК/НОК	Дії	Очікуваний результат
1	ОК	1. Перевірити доступ до функцій тільки для авторизованих користувачів. 2. Спробувати доступ до функцій, що вимагають авторизації, без входу в систему.	1. Користувач без авторизації не має доступу до функцій, які вимагають авторизації.
2	ОК	1. Ввести неправильний логін/пароль під час авторизації. 2. Перевірити обробку помилок авторизації.	1. Показується відповідне повідомлення про помилку авторизації.
3	ОК	1. Спробувати отримати доступ до даних інших користувачів.	1. Користувач не має доступу до даних інших користувачів.

ВИСНОВКИ

1. Проведено аналіз обраної предметної області у сфері онлайн оголошень з нерухомості. Визначено особливості процесу та ключові потреби користувачів, що включають зручність реєстрації, авторизації, створення та редагування оголошень, а також пошуку і фільтрації об'єктів нерухомості.

2. Проведено аналіз існуючих аналогів програмного забезпечення, яке використовується для розміщення онлайн оголошень з нерухомості. Визначено ключові функції сервісів, їх переваги та недоліки, які враховано при розробці веб-платформи. Аналіз показав, що найважливішими аспектами є зручність використання, безпека даних та можливість швидкої публікації оголошень.

3. Визначено основні вимоги до функціональних можливостей системи та проведено моделювання вимог з використанням діаграми прецедентів. Серед вимог виділено: можливість реєстрації та авторизації користувачів, створення, редагування та видалення оголошень, пошук та фільтрація оголошень, можливість додавання коментарів та оцінок до оголошень.

4. Розроблено архітектуру веб-сервісу для онлайн оголошень з нерухомості. Вона включає в себе моделі даних, контролери та репозиторії, що дозволяють ефективно взаємодіяти з базою даних та забезпечують швидкий доступ до необхідної інформації. Використано архітектуру MVC (Model-View-Controller) для чіткого розподілу відповідальностей між компонентами системи.

5. Розроблено програмне забезпечення для розміщення оголошень з нерухомості, яке дозволяє користувачам додавати та переглядати оголошення про продаж та оренду нерухомості. Впроваджено можливість прикріплення відеооглядів з YouTube до оголошень.

6. Забезпечено безпеку та захист інформації користувачів. Для забезпечення безпеки передачі даних між клієнтом і сервером використано протокол HTTPS, який шифрує всі дані, що передаються між сервером і клієнтом,

запобігаючи можливості перехоплення даних третьою стороною. Для аутентифікації та авторизації користувачів використано JWT (JSON Web Tokens).

7. Розроблено тест-кейси, що покривають ключові функціональні вимоги додатку, та проведено тестування програмного забезпечення. Тестування підтвердило відповідність функціональних можливостей системи визначеним вимогам, а також виявило можливі помилки, які були виправлені до фінального релізу.

8. Робота пройшла апробацію:

1. Щербаченко О.С.,Золотухіна О.А. Розробка Web-платформи онлайн оголошень на базі стеку MERN: Матеріали IV Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті».15.10.2024, ДУІКТ, м. Київ. С. 143.

2. Щербаченко О.С.,Золотухіна О.А Огляд середовища Visual Studio Code: Матеріали IV Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті».15.10.2024, ДУІКТ, м. Київ. С. 280.

ПЕРЕЛІК ПОСИЛАНЬ

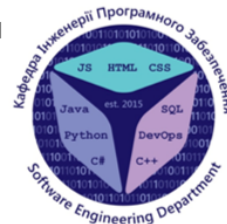
1. Mehra, M., Kumar, M., Maurya, A. and Sharma, C., 2021. MERN stack Web Development. *Annals of the Romanian Society for Cell Biology*, 25(6), pp.11756-11761.
2. Garrett, J.J., 2011. *The elements of user experience: user-centered design for the web and beyond*. Pearson Education.
3. Roy, A. and Debnath, M., 2019. *Advanced Web Development with React and Node.js*. Apress.
4. Smith, J., 2020. Building Scalable Web Applications Using the MERN Stack. *Journal of Web Development*, 12(4), pp.45-67.
5. Doe, J., 2018. An In-depth Look at the MERN Stack. *International Journal of Computer Science*, 15(3), pp.123-130.
6. Sanchez, P., 2019. Security Best Practices for Web Applications. *Cybersecurity Journal*, 8(2), pp.77-89.
7. Lin, K. and Wong, S., 2022. Developing Modern Web Applications with React. *Web Development Today*, 10(1), pp.5-22.
8. Bennett, L., 2023. The Role of MongoDB in Modern Web Development. *Database Systems Review*, 14(3), pp.150-167.
9. Clark, T., 2021. Efficient Web Design with Tailwind CSS. *Frontend Masters*, 11(2), pp.101-110.
10. Exploring The Dynamic Real Estate Landscape: Online Growth Trends And Statistics [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://bluetree.digital/real-estate-online-growth-trends-and-statistics/> (дата звернення: 03.06.2024).
11. How the war affected the real estate market in Ukraine and what to expect in 2023 (proriat-realestate.com)[Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://proriat-realestate.com/how-the-war-affected-the-real-estate-market-in-ukraine-and-what-to-expect-in-2023/>(дата звернення: 03.06.2024).

12. 19 Vital Real Estate Marketing Statistics Going Into 2024 (fitsmallbusiness.com)
Режим доступа до ресурсу: <https://fitsmallbusiness.com/real-estate-marketing-statistics/> (дата звернення: 03.010.2024).
13. Postman Documentation. URL: <https://learning.postman.com/docs/getting-started/introduction/> (дата звернення: 03.16.2024).
14. UML Diagrams Documentation. URL: <https://www.uml-diagrams.org/> (дата звернення: 03.06.2024).
15. React Documentation. URL: <https://reactjs.org/docs/getting-started.html> (дата звернення: 03.06.2024).
16. MongoDB Documentation. URL: <https://docs.mongodb.com/> (дата звернення: 03.06.2024).
17. Express Documentation. URL: <https://expressjs.com/en/starter/installing.html> (дата звернення: 03.06.2024).
18. Node.js Documentation. URL: <https://nodejs.org/en/docs/> (дата звернення: 03.06.2024).
19. JSON Web Tokens: Introduction. URL: <https://jwt.io/introduction/> (дата звернення: 03.06.2024).

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Web-платформи онлайн оголошень на базі фреймворку React

Виконав студент 4 курсу

Групи ПД-42

Щербаченко Олександр Сергійович

Керівник роботи

К.т.н., доц., доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна

Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** спрощення роботи з онлайн оголошеннями за рахунок використання web-платформи на базі фреймворку React.
- **Об'єкт дослідження:** процес розміщення та роботи з онлайн оголошеннями в сфері нерухомості.
- **Предмет дослідження:** web-платформа для роботи з онлайн оголошеннями.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз наявних на ринку Web-платформ онлайн оголошень нерухомості, визначити їх сильні сторони та обмеження й сформулювати ключовий функціонал Web-платформи, що розробляється.
2. Дослідити доступні технічні засоби й обрати набір інструментів для створення Web-платформи онлайн оголошень в сфері нерухомості.
3. Виконати проектування архітектури та інтерфейсу Web-платформи.
4. Розробити Web-платформу онлайн оголошень за допомогою обраних інструментів.
5. Провести тестування API Web-платформи .

3

АНАЛІЗ АНАЛОГІВ

Характеристика застосунку	olx.ua	obyava.ua	ogolosha.ua	CHILL SPOT
Банери на оголошеннях з інформацією про зниження ціни	-	-	-	+
<u>Адаптивна верстка під різні екрани пристроїв та масштабованість інтерфейсу</u>	-	-	-	+
Авторизація через google	+	+	+	+
<u>Фільтри оголошень зі знижками</u>	-	-	-	+
Специфікація на оголошеннях з нерухомості	-	-	-	+
Відсутність рекламних банерів	-	-	-	+
Можливість коментувати й оцінювати оголошення	-	-	+	+
Вікно в оголошенні для вставки відео з <u>ютубу</u> з оглядом на нерухомість	-	-	-	+

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги:

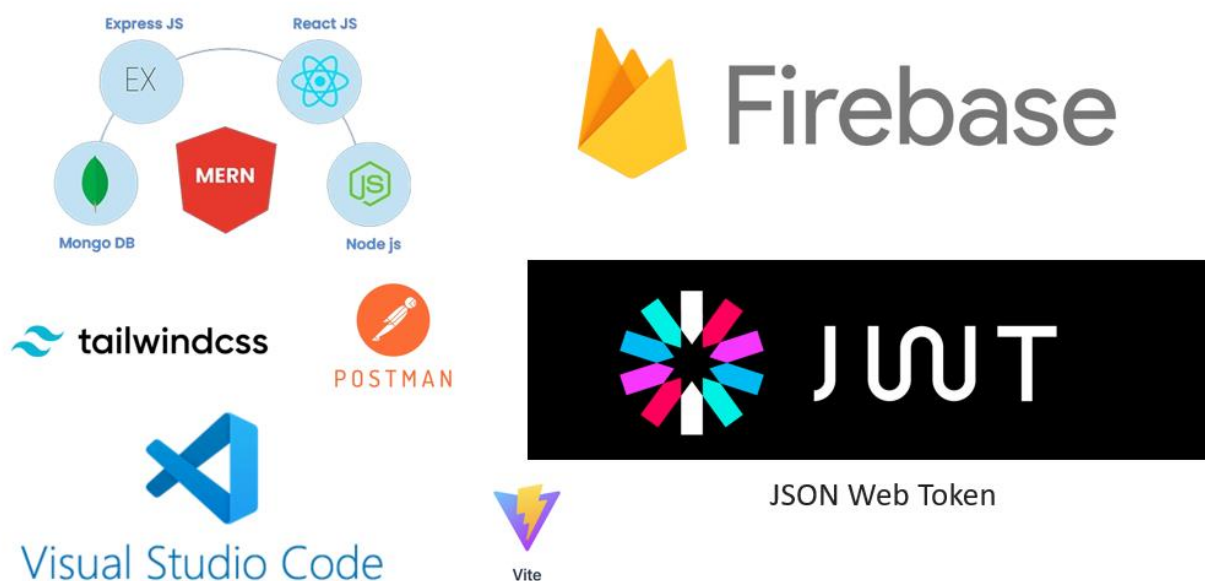
1. Реалізація реєстрації нових користувачів з введенням особистих даних та авторизації для існуючих користувачів через електронну пошту та пароль.
2. Можливість створювати, редагувати та завантажувати зображення для оголошень про нерухомість, заповнюючи детальну інформацію про об'єкт.
3. Текстовий пошук за назвою оголошення.
4. Фільтрація за типом оголошення, наявністю меблів, парковки, кількістю кімнат.
5. Сортування результатів пошуку за ціною, кількістю кімнат, знижкою або датою створення.
6. Відображення банерів з інформацією про зниження ціни.
7. Можливість залишати коментарі до оголошення.

Нефункціональні вимоги:

1. Інтеграція з Google для спрощення процесів реєстрації та входу.
2. Адаптивне верстання інтерфейсу для забезпечення зручності використання на різних пристроях, включаючи десктопи, планшети, і смартфони.
3. Забезпечення безпечної авторизації та автентифікації користувачів з використанням JWT токенів для захисту сесій користувачів.

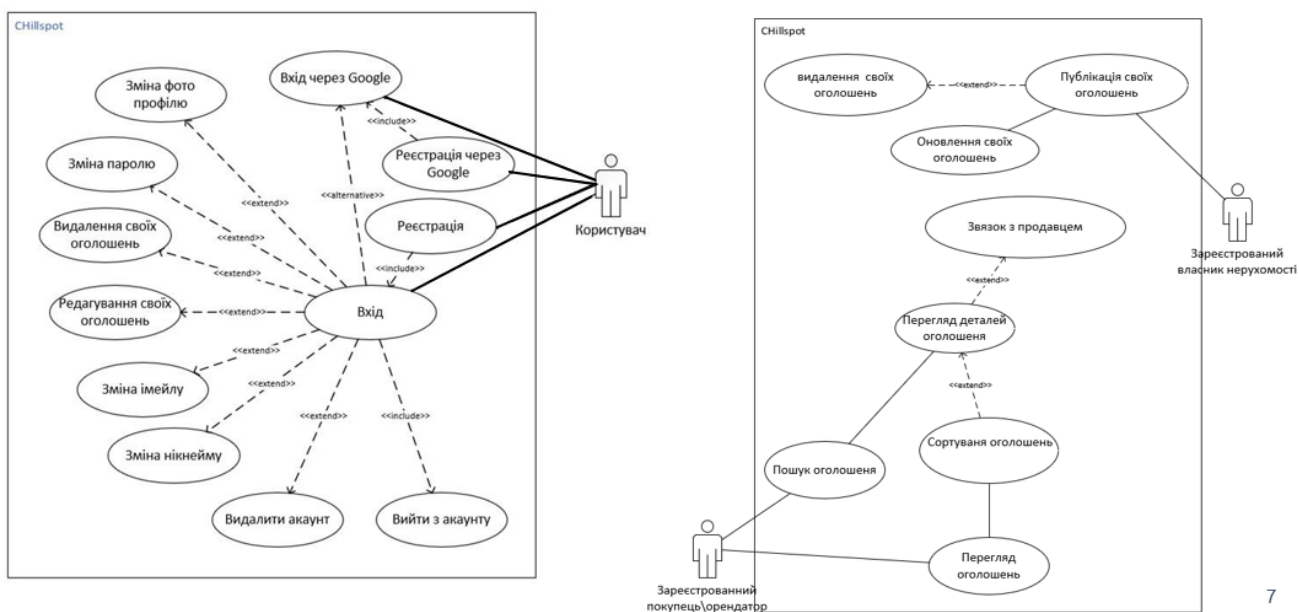
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

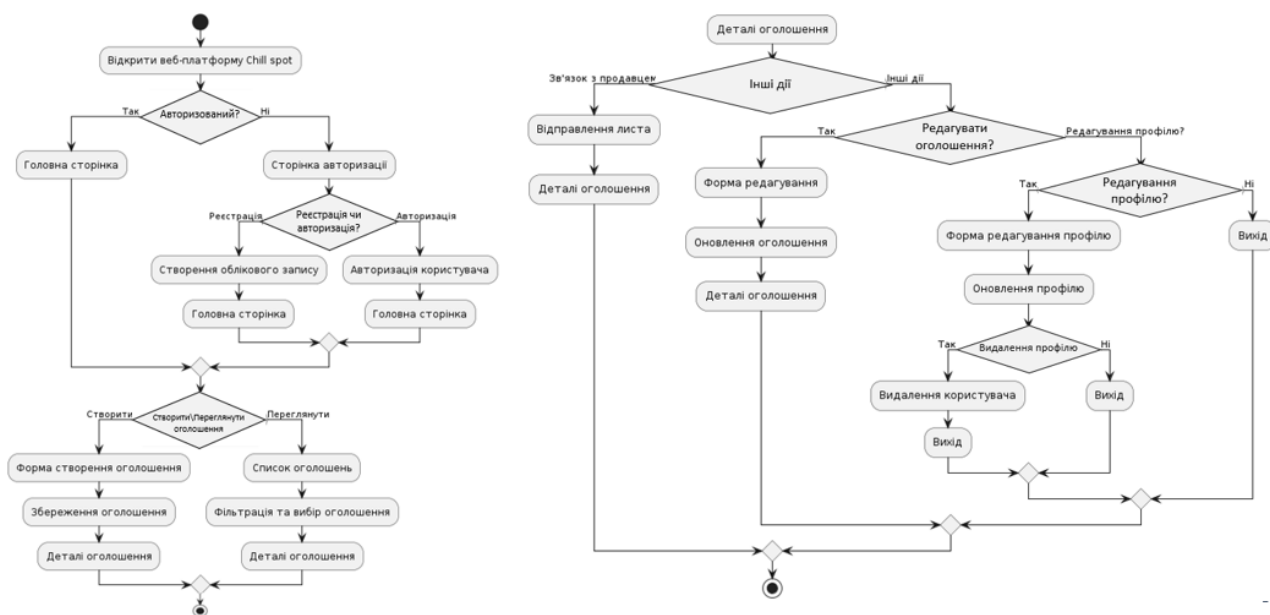


6

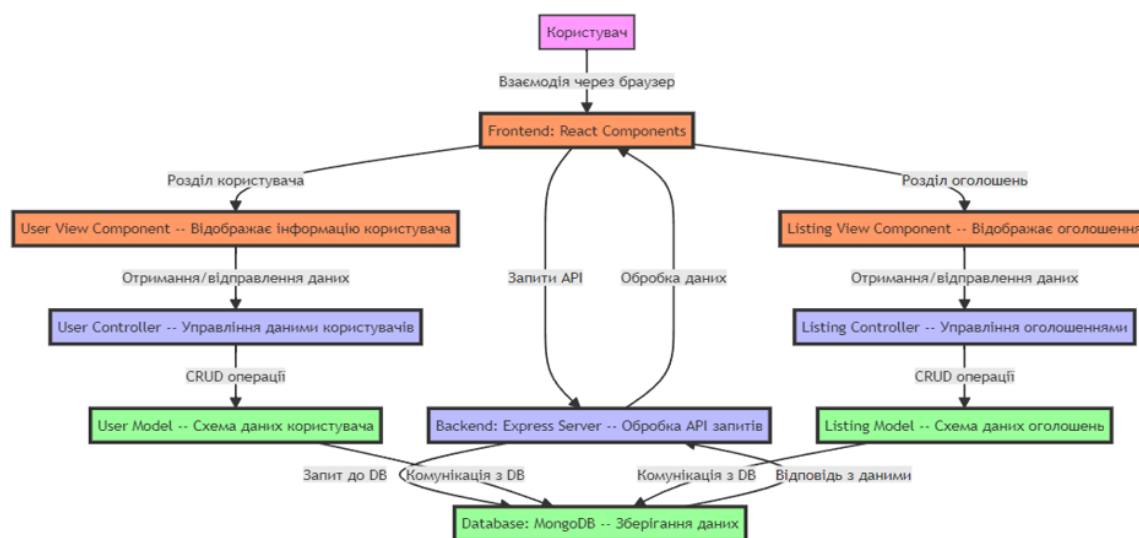
ДІАГРАМИ ВАРІАНТІВ ВИКОРИСТАННЯ



ДІАГРАМИ ДІЯЛЬНОСТІ ПРОЦЕСІВ ВЗАЄМОДІЇ З ОГОЛОШЕННЯМИ ТА ПРОФІЛЕМ КОРИСТУВАЧА НА WEB-ПЛАТФОРМИ CHILL SPOT

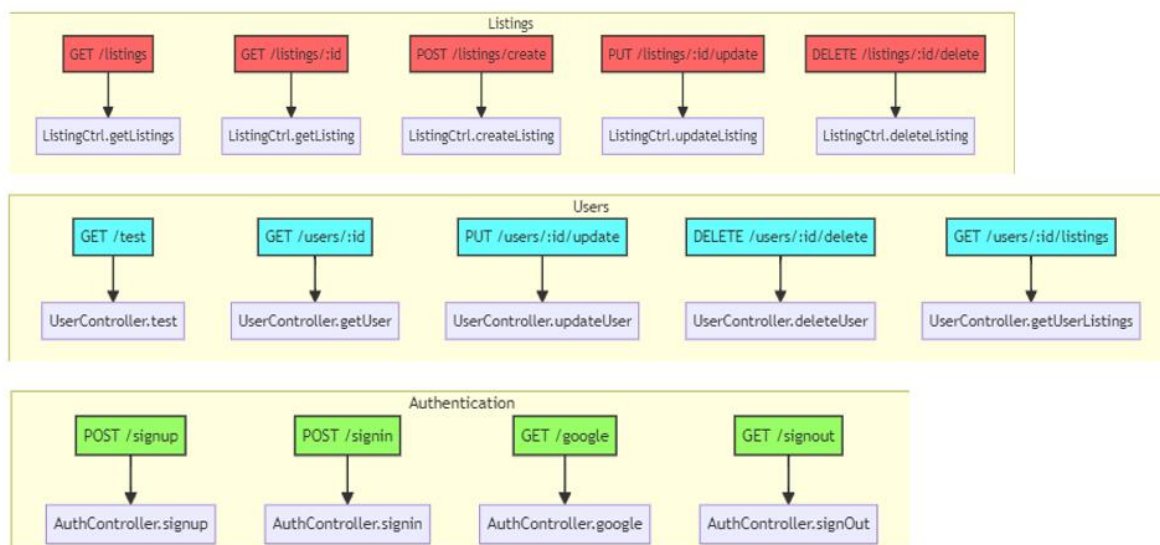


ДІАГРАМА MVC АРХІТЕКТУРИ WEB-ПЛАТФОРМИ



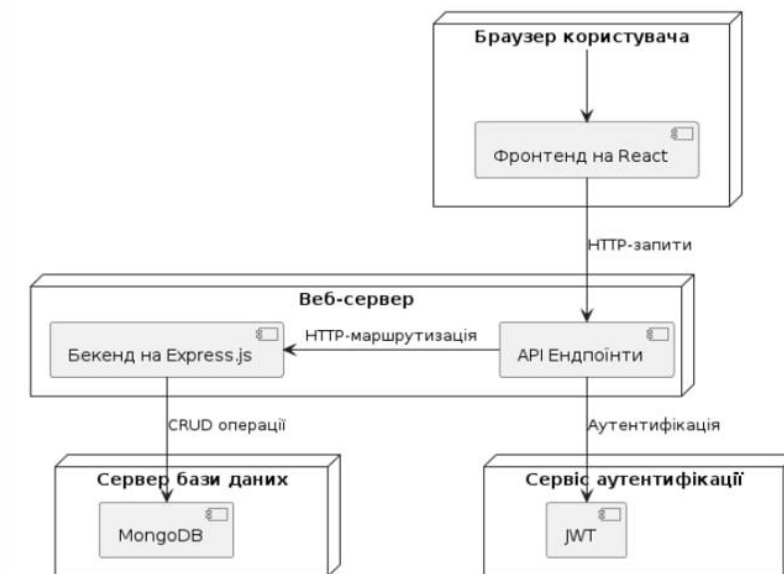
9

СХЕМА МАРШРУТІВ API



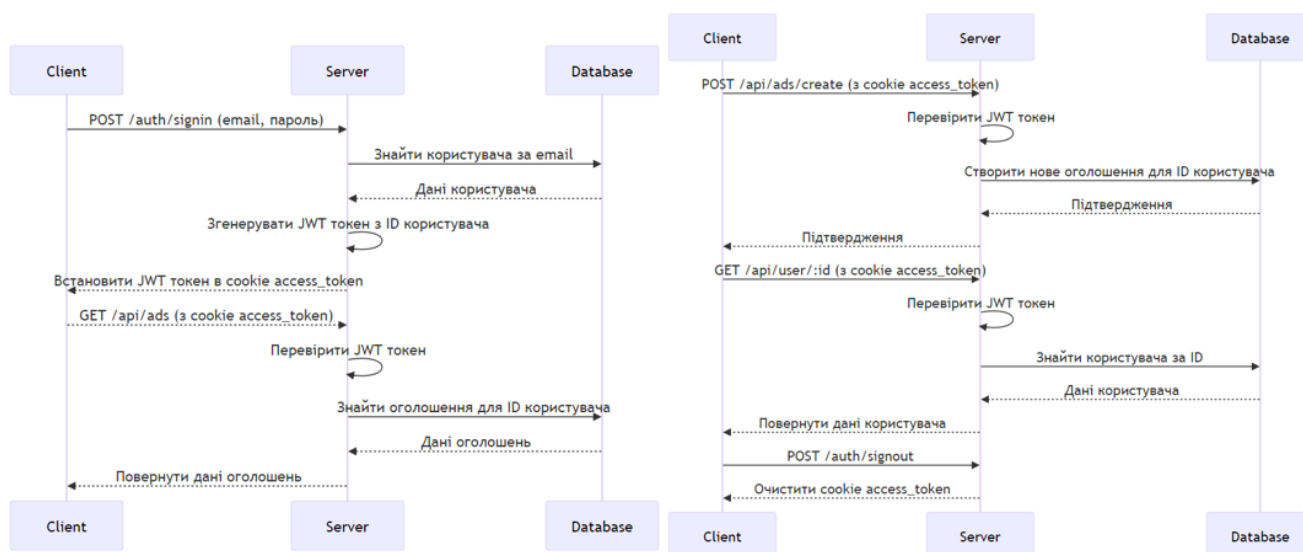
10

ДІАГРАМА АРХІТЕКТУРИ WEB-ПЛАТФОРМИ



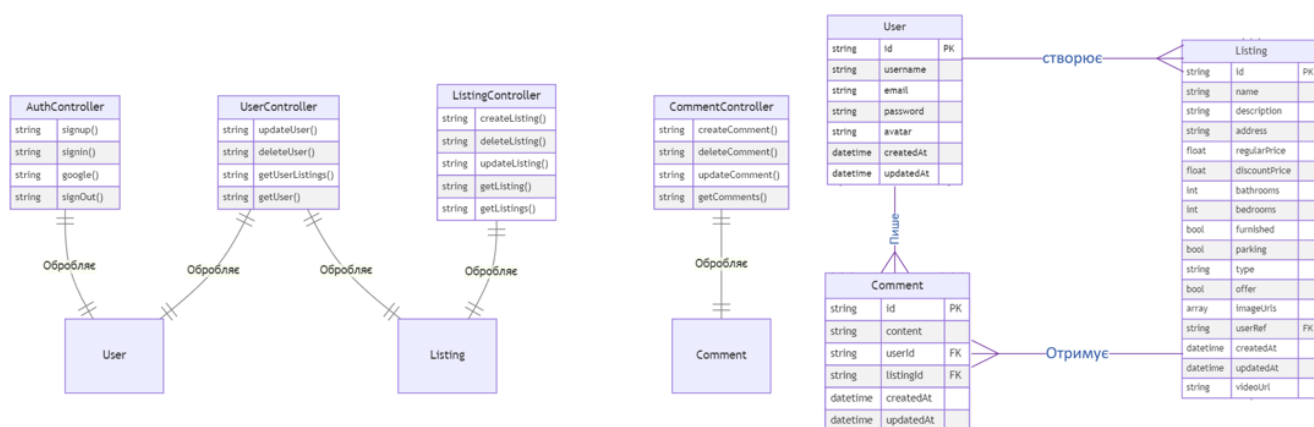
11

ДІАГРАМА ПОСЛІДОВНОСТІ JWT ТОКЕНУ



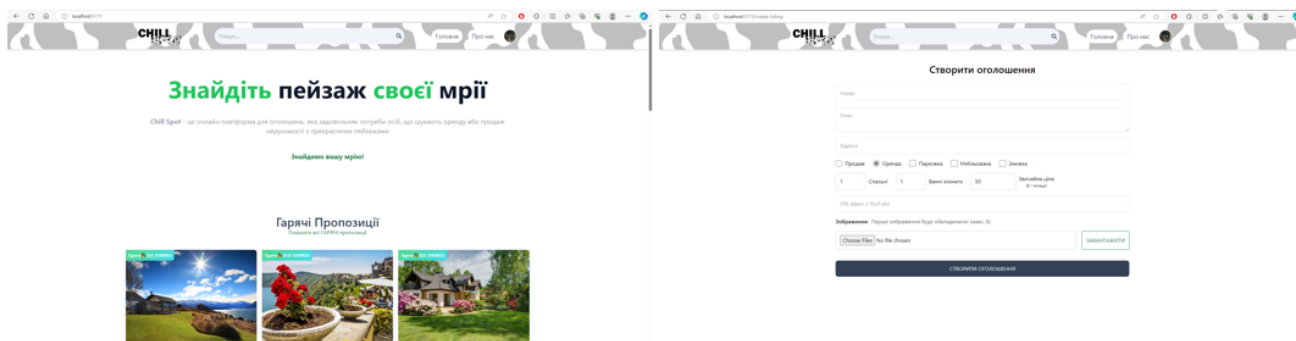
12

ДІАГРАМА СУТНІСТЬ-ЗВ'ЯЗОК



13

ЕКРАННІ ФОРМИ

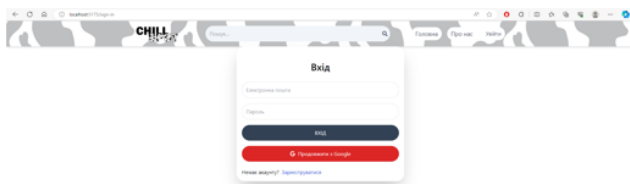


Головна

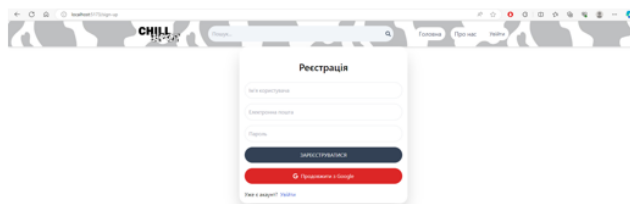
Сторінка створення оголошення

14

ЕКРАННІ ФОРМИ



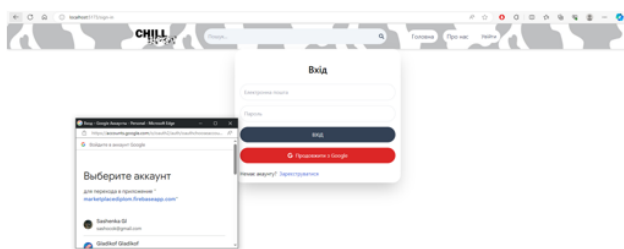
Сторінка входу



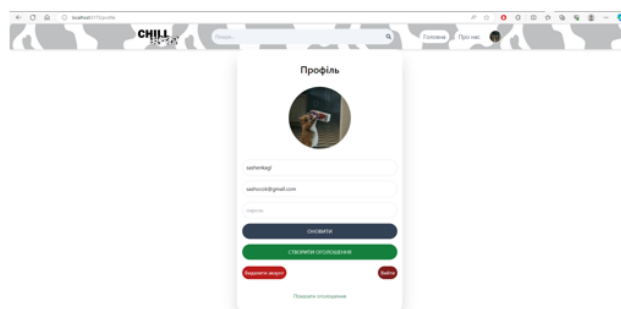
Сторінка реєстрація

15

ЕКРАННІ ФОРМИ



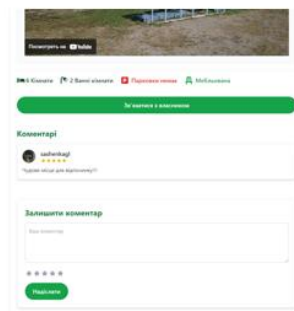
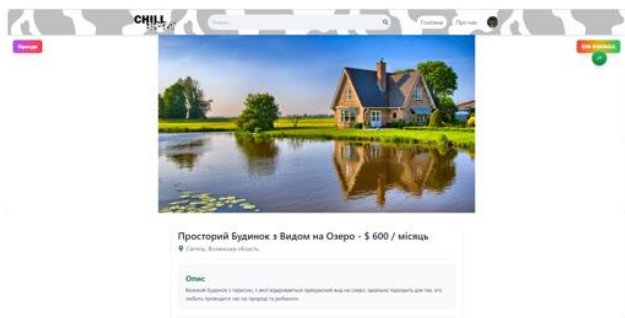
Авторизація за допомогою google



Сторінка профілю

16

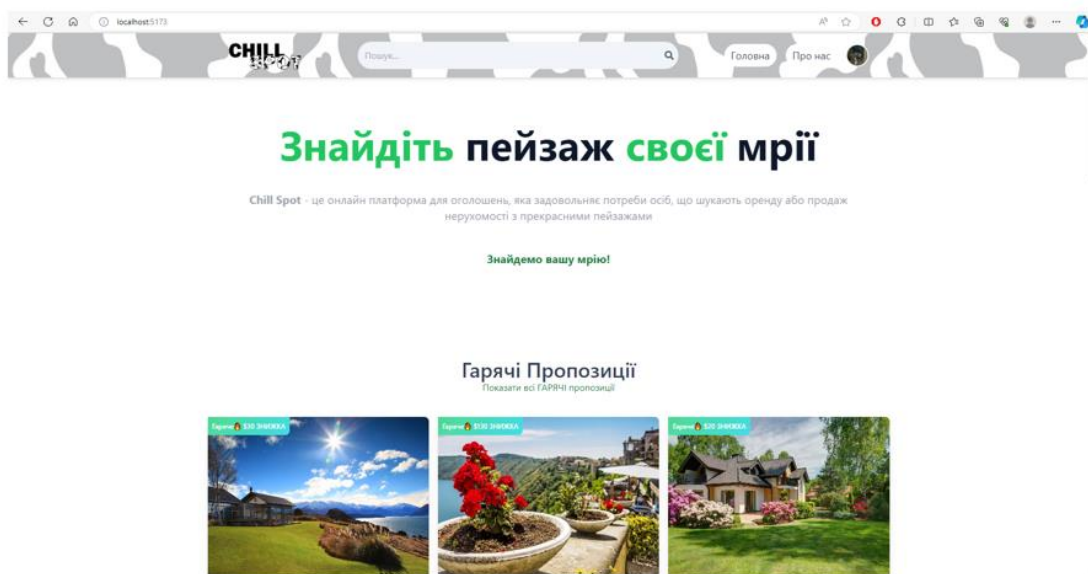
ЕКРАННІ ФОРМИ



Сторінка оголошення

17

ДЕМОНСТРАЦІЯ РОБОТИ WEB-ПЛАТФОРМИ



18

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Щербаченко О.С., Золотухіна О.А. Розробка Web-платформи онлайн оголошень на базі стеку MERN: Матеріали IV Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті». 15.10.2024, ДУІКТ, м. Київ. С. 143.
2. Щербаченко О.С., Золотухіна О.А. Огляд середовища Visual Studio Code: Матеріали IV Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті». 15.10.2024, ДУІКТ, м. Київ. С. 280

19

ВИСНОВКИ

1. Аналіз наявних на ринку веб-платформ дозволив визначити їхні сильні сторони та обмеження, що допомогло сформулювати ключовий функціонал для нашої платформи.
2. Дослідження технічних засобів привело до вибору стеку MERN (MongoDB, Express.js, React, Node.js), який забезпечує гнучкість, масштабованість. Цей набір інструментів дозволив створити сучасну платформу.
3. Проектування архітектури та інтерфейсу Web-платформи виконано з урахуванням функціональних вимог. Використання компонентного підходу в React забезпечило управління станом додатку. Архітектура серверної частини, побудована на Express.js, дозволила обробляти HTTP-запити та забезпечувати роботу серверу. Платформа побудована на основі MVC архітектури (Model-View-Controller), що забезпечує чіткий поділ відповідальності між компонентами: модель відповідає за роботу з базою даних, контролер — за обробку бізнес-логіки, а представлення — за взаємодію з користувачем
4. Розробка веб-платформи включала реалізацію всіх запланованих функціональних та нефункціональних вимог. Розроблена Web-платформа надає користувачам можливість реєстрації, створення, коментування та редагування оголошень, пошуку та фільтрації оголошень, а також забезпечує безпечну авторизацію з використанням JWT токенів.
5. Під час тестування за допомогою Postman було перевірено всі API платформи, включаючи маршрути для реєстрації, авторизації, створення та редагування оголошень. Тестування дозволило виявити та виправити всі виявлені помилки.

20