

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: Розроблення автоматизованого робочого місця підприємця з надання
послуг ріелторської діяльності

Виконав: студент групи 4141

 Юшко К. В.
(підпис)

Керівник роботи:

К.Т.Н, ДОЦ.
(посада, науковий ступінь вчене звання)
 Книрик Н.Р.
(підпис)

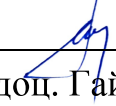
м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми


к.т.н, доц. Гайда А.Ю.

" " 2025 р.

*ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ*
на здобуття ступеня вищої освіти "бакалавр"

Студенту Юшко Костянтину Вячеславовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення автоматизованого робочого місця підприємця з надання послуг ріелторської діяльності

Керівник роботи Книрик Наталя Ромуальдівна

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 червня 2025 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів): Вступ. 1. Розроблення автоматизованого робочого місця підприємця з надання послуг ріелторської діяльності. 2. Розроблення проекту інформаційної системи. 3. Реалізація проекту інформаційної системи. 4. Охорона праці і техніка безпеки. Висновки.

5. Перелік презентаційних матеріалів: Слайд 1. Назва теми кваліфікаційної роботи. Слайд 2. Актуальність обраної теми. Слайд 3. Мета дипломного проєкту. Слайд 4. Завдання, поставлені в межах проєкту. Слайд 5. Концепція інформаційної системи. Слайд 6. ER-діаграма сутностей у базі даних. Слайд 7.

Логічна модель бази даних. Слайд 8. Фізична модель бази даних. Слайд 9. Програмно-технічні засоби реалізації. Слайд 10-18. Візуалізація проекту. Слайд 19. Основні висновки з реалізації проекту. Слайд 20. Завершення презентації.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Книрик Н. Р. к.т.н, доц.	03.02.2025	22.02.2025
2	Книрик Н. Р. к.т.н, доц.	22.02.2025	21.03.2025
3	Книрик Н. Р. к.т.н, доц.	21.03.2025	25.04.2025
4	Книрик Н. Р. к.т.н, доц.	25.04.2025	07.06.2025

7. Дата видачі завдання 03 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	13.02.2025	
2	Постановка задачі	15.02.2025	
3	Розробка концепції	22.02.2025	
4	Розробка проекту ІС	21.03.2025	
5	Реалізація проекту	25.04.2025	
6	Розробка питань з охорони праці	07.06.2025	
7	Розробка графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

Студент



(підпис)

Юшко К. В.

(ПІБ)

Керівник роботи



(підпис)

Книрик Н. Р.

(ПІБ)

АНОТАЦІЯ

Обсяг пояснювальної записки складає 88 сторінок. У роботі наведено 28 ілюстрацій, 10 таблиць, 2 додатки та 21 джерел у списку літератури.

Об'єктом проекту є розробка інформаційної системи маркетплейсу оголошень про нерухомість. Предметна область проекту охоплює процеси пошуку, публікації, реалізації та управління нерухомістю. Метою проекту є створення інноваційного веб-маркетплейсу, що забезпечить зручний інструмент для роботи ріелторів, користувачів та адміністраторів, надаючи доступ до актуальної інформації про об'єкти нерухомості та спрощуючи процеси купівлі, продажу та оренди.

У межах роботи були застосовані методи аналізу вимог користувачів, моделювання предметної області, проектування архітектури програмного забезпечення, реалізації та тестування функціоналу. У результаті реалізовано функціональну веб-систему, що дозволяє взаємодіяти користувачам різних ролей, публікувати та переглядати оголошення, а також керувати ними через зручний інтерфейс.

Основні технології, що використовувалися у проєкті: мова програмування PHP, база даних MySQL, HTML, CSS, JavaScript.

Отримані результати можуть бути використані в ріелторській діяльності для автоматизації роботи з об'єктами нерухомості та підвищення ефективності комунікації між усіма учасниками ринку.

Ключові слова: нерухомість, ріелтор, маркетплейс, оголошення, веб-система, розробка, інформаційна система.

ABSTRACT

The volume of the explanatory note is 88 pages. The work contains 28 illustrations, 10 tables, 2 appendices and 21 sources in the list of references.

The object of the project is the development of an information system for a real estate marketplace. The subject area of the project covers the processes of searching, publishing, selling and managing real estate. The goal of the project is to create an innovative web marketplace that will provide a convenient tool for realtors, users and administrators, providing access to up-to-date information about real estate and simplifying the processes of buying, selling and renting.

The work involved analyzing user requirements, modeling the subject area, designing the software architecture, implementing and testing the functionality. As a result, a functional web-based system has been implemented that allows users of different roles to interact, publish and view ads, and manage them through a user-friendly interface.

The main technologies used in the project are PHP programming language, MySQL database, HTML, CSS, JavaScript, and SOAP protocol for data exchange between client and server.

The results obtained can be used in real estate activities to automate work with real estate objects and increase the efficiency of communication between all market participants.

Keywords: real estate, realtor, marketplace, ads, web system, development, information system.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ I РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ ПІДПРИЄМЦЯ З НАДАННЯ ПОСЛУГ РІЕЛТОРСЬКОЇ ДІЯЛЬНОСТІ.....	10
1.1 Опис об'єкта дослідження та аналіз предметної області.....	10
1.2 Формування вимог до системи.....	15
1.3 Розроблення концепції автоматизованого робочого місця підприємця з надання послуг ріелторської діяльності.....	18
РОЗДІЛ II РОЗРОБЛЕННЯ ПРОЕКТУ ІФОРМАЦІЙНОЇ СИСТЕМИ.....	24
2.1 Загальні системні рішення.....	24
2.2 Проектування баз даних.....	25
2.3 Проектування логіки функціонування системи.....	35
2.4 Рішення з програмного забезпечення.....	37
2.5 Рішення з технічного забезпечення.....	39
РОЗДІЛ III РЕАЛІЗАЦІЯ ПРОЕКТУ ІФОРМАЦІЙНОЇ СИСТЕМИ.....	41
РОЗДІЛ IV ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ.....	70
ВИСНОВОК.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТОК - Текст програми.....	76

ВСТУП

Сьогодні ІТ-технології відіграють ключову роль у розвитку ринку нерухомості, оскільки дозволяють автоматизувати складні бізнес-процеси й покращити обслуговування клієнтів. На ринку нерухомості, де кожна угода пов'язана з великими фінансовими ризиками, швидким обміном даних та участю багатьох посередників, автоматизація дозволяє скоротити час оформлення операцій, мінімізувати людські помилки й забезпечити прозорість взаємодії між усіма сторонами.

Сучасний ринок нерухомості вимагає нових підходів до організації пошуку, купівлі та продажу об'єктів. Традиційні методи часто пов'язані з великою кількістю посередників і тривалим часом на оформлення угод. У цьому контексті зростає актуальність створення веб-маркетплейсів, які об'єднують користувачів і рієлторів, спрощуючи процеси взаємодії та управління нерухомістю. Впровадження єдиного веб-маркетплейсу, що об'єднує продавців, покупців і рієлторів, надає можливість централізовано керувати оголошеннями, договорами й транзакціями, водночас підвищуючи якість обслуговування клієнтів.

Проектування такої системи ґрунтується на поетапному моделюванні: від аналізу предметної області й формування концептуальної моделі до логічного та фізичного проєкту бази даних. Послідовний перехід між рівнями моделі гарантує узгодженість між бізнес-вимогами та програмною реалізацією, а також спрощує подальшу підтримку й масштабування.

Актуальність даної кваліфікаційної роботи зумовлена відсутністю універсальної платформи, що одночасно враховує специфіку українського законодавства, потреби різних ролей користувачів і вимоги до безпечних онлайн-платежів. У ході дослідження проаналізовано існуючі сервіси, виявлено їхні

обмеження, сформульовано комплексні вимоги до системи, спроектовано структуру даних і реалізовано клієнт-серверний веб-застосунок.

Метою роботи є створення інформаційної системи-маркетплейсу, що автоматизує процеси публікації та керування оголошеннями, забезпечує надійні механізми фінансових розрахунків, рейтингування рієлторів і моніторинг угод у реальному часі. Запропоноване рішення має підвищити продуктивність рієлторів, спростити пошук і купівлю об'єктів для клієнтів та закласти основу для подальшого масштабування бізнесу на ринку нерухомості.

РОЗДІЛ 1 РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ ПІДПРИЄМЦЯ З НАДАННЯ ПОСЛУГ РІЕЛТОРСЬКОЇ ДІЯЛЬНОСТІ

1.1 Опис об'єкта дослідження та аналіз предметної області

У сучасному ринку нерухомості діє велика кількість ріелторів - як фізичних осіб-підприємців, так і співробітників спеціалізованих агентств. Ріелтор - це фахівець, який надає послуги з підбору, продажу, оренди та купівлі об'єктів нерухомості, виступаючи посередником між продавцем і покупцем. Його діяльність охоплює підготовку документів, аналіз ринку, організацію показів, консультації клієнтів, перевірку прав власності тощо.

Ріелтори можуть працювати індивідуально, відкривши ФОП, або входити до складу ріелторських агентств. Агентства, як правило, мають внутрішню структуру управління, що включає керівника, команду ріелторів і супровідний персонал. Такі фірми можуть мати офіси в одному чи кількох районах міста, працювати за територіальним розподілом (наприклад, один ріелтор відповідає за певний мікрорайон), або за тематикою (житлова/комерційна нерухомість). У межах агентств часто реалізовані внутрішні системи для обліку клієнтів, розподілу замовлень, ведення статистики угод [5].

У деяких випадках при надходженні нового замовлення (наприклад, від продавця або покупця) формується черга або система пріоритетів, і заявка передається в роботу вільному ріелтору. Це дає змогу ефективніше розподіляти навантаження між співробітниками.

Однак у зв'язку з розвитком цифрових технологій дедалі більше ріелторів обирають шлях самостійної діяльності без участі агентств. Для цього вони створюють власні сайти або акаунти на платформах, просуюючи послуги

напряму. Така модель дозволяє уникати сплати частини прибутку агентству, але водночас ускладнює залучення клієнтів та управління власним обсягом роботи.

У зв'язку з цим актуальним є створення єдиного централізованого маркетплейсу для купівлі та продажу нерухомості, на якому можуть взаємодіяти всі учасники ринку: продавці, рієлтори, покупці [5]. Така система дозволить об'єднати зусилля окремих фахівців і компаній, надати їм зручний інструмент для розміщення об'єктів, управління угодами та взаємодії з клієнтами. Крім того, рієлтори зможуть отримувати нові замовлення, брати участь у розподілі заявок (наприклад, за категоріями або географією), вести особистий кабінет, отримувати рейтинг за підсумками успішних угод.

Таким чином, маркетплейс виступає як цифровий хаб для учасників ринку нерухомості, забезпечуючи ефективну та прозору взаємодію, що є суттєвим кроком вперед порівняно з класичними підходами.

Маркетплейс - це онлайн-платформа, що об'єднує покупців і продавців для здійснення угод у цифровому середовищі. В умовах сучасного ринку нерухомості такі платформи відіграють ключову роль, оскільки забезпечують зручний доступ до великої кількості пропозицій, автоматизують взаємодію між учасниками ринку та спрощують процес укладання угод [5].

Використання маркетплейсу дозволяє покупцям швидко знайти потрібні об'єкти, а продавцям - ефективно презентувати свої пропозиції без необхідності залучення сторонніх посередників.

Сфера нерухомості завжди була однією з найбільш значущих для економіки, адже попит на житло та комерційні приміщення залишається стабільно високим. Люди змінюють місце проживання через особисті або професійні обставини, компанії розширюють бізнес, і в кожному з цих випадків виникає потреба в купівлі або оренді нерухомості. Традиційні методи пошуку - через агентства або приватні оголошення - часто є затратними та неефективними,

тому онлайн-маркетплейси стали основним інструментом для швидкого доступу до актуальних пропозицій.

Також перепродаж нерухомості завжди був невіддільною потребою в житті людей, адже через різні ситуації людям доводиться переміщатися в інші країни або міста, де в них може і не бути своєї нерухомості, або для різних комерційних компаній, де так само є необхідність. Особливо помітний приріст пошуку нерухомості з початком війни, адже тепер дуже багато переселенців потребує даху над головою. І найлегше в новому місті покупцеві шукати якраз таки використовуючи браузер, де можна подивитися детально всі потрібні деталі щодо нерухомості.

Завдяки інтеграції сучасних технологій маркетплейси нерухомості дозволяють автоматизувати більшість процесів - від фільтрації оголошень і спілкування між сторонами до здійснення фінансових операцій. Це значно підвищує прозорість ринку, знижує ризики шахрайства та робить угоди більш безпечними.

Попри всі переваги використання маркетплейсів, перш ніж покладатися на існуючі рішення, необхідно провести системний аналіз предметної області та функціональності таких платформ, щоб визначити доцільність їхнього використання у готовому вигляді або ж потребу в розробці власного рішення [6].

У сфері нерухомості одним із ключових суб'єктів виступає ріелтор - спеціаліст, який виконує функцію посередника між покупцем і продавцем, надаючи послуги з підбору, продажу, оренди або купівлі об'єктів. Ріелторська діяльність охоплює широкий спектр завдань: від аналізу ринку, організації показів, консультацій клієнтів - до перевірки прав власності та супроводу угод. У сучасних умовах дедалі більше фахівців працюють самостійно - поза межами агентств, зокрема у форматі ФОП або як незалежні консультанти.

Разом із тим, такі спеціалісти часто не мають у своєму розпорядженні централізованих цифрових інструментів для управління своєю діяльністю. Ріелтори, які тільки починають кар'єру, можуть зіткнутися з браком досвіду в організації бізнес-процесів, обліку клієнтів і об'єктів, контролі фінансових операцій та веденні бази нерухомості. Без надійної інформаційної підтримки така діяльність ускладнюється, що знижує конкурентоспроможність фахівця. Отже, постає потреба у виборі відповідного засобу керування своєю діяльністю. Одним з можливих напрямів є використання типових проєктних рішень, зокрема - сучасних маркетплейсів, які частково виконують функції систем підтримки ріелторської роботи.

Без надійної інформаційної підтримки така діяльність ускладнюється, що знижує конкурентоспроможність фахівця. Отже, постає потреба у виборі відповідного засобу керування своєю діяльністю. Одним з можливих напрямів є використання типових проєктних рішень, зокрема - сучасних маркетплейсів, які частково виконують функції систем підтримки ріелторської роботи.

Для обґрунтування доцільності розробки власної інформаційної системи доцільно проаналізувати існуючі маркетплейси, що функціонують на українському ринку нерухомості. Це дозволить виявити рівень розвитку цифрових сервісів у галузі, оцінити функціональні можливості вже реалізованих рішень, а також з'ясувати їхні переваги та недоліки з погляду ефективного управління процесами купівлі-продажу об'єктів нерухомості.

Аналіз існуючих маркетплейсів

Dom.ria.com - одна з найпопулярніших платформ у сфері нерухомості, що охоплює всю територію України. Сервіс активно співпрацює як із приватними продавцями, так і з ріелторами, надаючи інструменти для зручного управління оголошеннями [2].

Функціонал включає фільтри за ціною, місцем розташування, типом об'єкта, площею тощо. Користувачі можуть ефективно шукати нерухомість, яка відповідає їхнім потребам [2].

Платформа також передбачає вбудований чат для прямої комунікації між учасниками [2].

Rem.ua - український ресурс, який охоплює різні категорії товарів і послуг, зокрема нерухомість. Сервіс орієнтований на продаж і оренду як житлової, так і комерційної нерухомості [3].

Користувачі можуть шукати об'єкти за базовими параметрами: місце розташування, ціна, тип, площа. Існує система профілів, що дозволяє ознайомитися з активністю продавців [3].

OLX.ua є найбільшою в Україні онлайн-дошкою оголошень, яка охоплює безліч категорій. У розділі «Нерухомість» користувачі мають змогу розміщувати оголошення про продаж, оренду житла, земельних ділянок, комерційних приміщень [4].

Система забезпечує простий інтерфейс із можливістю пошуку за основними критеріями. Завдяки великій базі користувачів, оголошення мають високу видимість [4].

Узагальнені висновки за результатами аналізу.

Проведений аналіз свідчить про наявність популярних платформ, які частково вирішують завдання, пов'язані з купівлею та продажем нерухомості. Водночас жодна з них не забезпечує повноцінної підтримки всього циклу взаємодії між сторонами, зокрема:

- більшість платформ не мають вбудованих механізмів для здійснення безпечних фінансових операцій, а розрахунки відбуваються поза межами сайтів, що створює ризики для користувачів;

- відсутні системи рейтингу ріелторів, що ускладнює об'єктивну оцінку їхньої надійності;
- часто відсутня вузька спеціалізація на ринку нерухомості, через що користувачі отримують обмежений набір функцій, пристосованих до специфіки таких угод;
- іноді спостерігається застарілий або перевантажений інтерфейс, що негативно впливає на користувацький досвід;
- не реалізована технічна інтеграція з банківськими сервісами, яка могла б забезпечити зручність і безпеку в процесі передачі коштів.

Крім того, використання готових платформ обмежує гнучкість у впровадженні нових функцій, оскільки більшість із них є закритими системами без доступу до внутрішніх механізмів, необхідних для адаптації під конкретні бізнес-завдання.

Узагальнюючи результати аналізу об'єкта дослідження та існуючих рішень на ринку, можна дійти висновку про необхідність створення сучасної, комплексної інформаційної системи, яка усуне недоліки наявних платформ і надасть користувачам повний інструментарій для взаємодії в межах ринку нерухомості.

Для реалізації такої системи потрібно чітко сформулювати вимоги до її функціоналу, інтерфейсу, структури користувачів, безпеки та масштабованості.

1.2 Формування вимог до системи

Вимоги до розроблюваної інформаційної системи сформовані з урахуванням недоліків наявних маркетплейсів нерухомості. Основними критеріями є безпечність транзакцій, прозора робота з ріелторами, зручний

інтерфейс, підтримка української мови та функціональна орієнтація виключно на ринок нерухомості.

Загальні вимоги:

- реалізація системи у вигляді веб-додатку з доступом через браузер;
- механізм авторизації та автентифікації;
- модерація оголошень ріелторами;
- система ролей із чітким розмежуванням прав доступу;
- підтримка роботи з кількома типами користувачів (адміністратор, ріелтор, покупець/продавець);
- інтуїтивно зрозумілий та зручний для користувача інтерфейс;
- інтеграція з платіжними сервісами або банківськими API для поповнення рахунку, виведення коштів та проведення внутрішніх транзакцій;
- укладення договорів між ріелтором і користувачем із зазначенням строків, обов'язків та вартості;
- облік договорів зі статусами (активні, завершені, прострочені);
- формування звітів за кожним договором;
- можливість встановлення зв'язку між ріелтором і користувачем як за ініціативою клієнта, так і ріелтора.

Функціональні вимоги:

- створення, перегляд, редагування та модерація оголошень про нерухомість;
- чат між користувачами з історією повідомлень;
- редагування особистих даних користувачем;
- фільтрація оголошень за заданими критеріями;
- доступ ріелтора до списку об'єктів, що підлягають модерації або вже перевірені;

- доступ адміністратора до журналу транзакцій та загального списку оголошень.

Вимоги до програмного забезпечення:

- використання сучасних мов програмування та веб-фреймворків;
- зберігання даних у реляційній системі керування базами даних (СУБД);
- інтеграція з платіжними API для фінансових операцій;
- реалізація заходів з інформаційної безпеки, зокрема:
 - захист від SQL-ін'єкцій та інших типових атак;
 - захищене зберігання паролів (хешування);
 - розмежування доступу до критичних функцій на основі ролей.

Вимоги до технічного забезпечення:

- розміщення веб-додатку на хмарному сервері з постійним доступом до Інтернету;
- зберігання файлів сайту у хмарному середовищі або на веб-хостингу (наприклад, AWS, Google Cloud, Azure);
- розміщення бази даних у хмарному середовищі з підтримкою масштабування та резервного копіювання;
- доступ користувачів до системи через веб-браузер за доменним іменем;
- стабільне інтернет-з'єднання для безперебійної взаємодії між клієнтом і сервером;
- використання SSL-сертифікатів для забезпечення захищеного з'єднання;
- підтримка механізмів резервного копіювання та відновлення даних у разі збоїв;
- вибір технічної платформи із забезпеченням високої доступності, масштабованості та надійності сервісу.

1.3 Розроблення концепції автоматизованого робочого місця підприємця з надання послуг ріелторської діяльності

На основі сформульованих вимог до системи було змодельовано концепцію інформаційної системи, що зображено на рисунку 1.3.1.

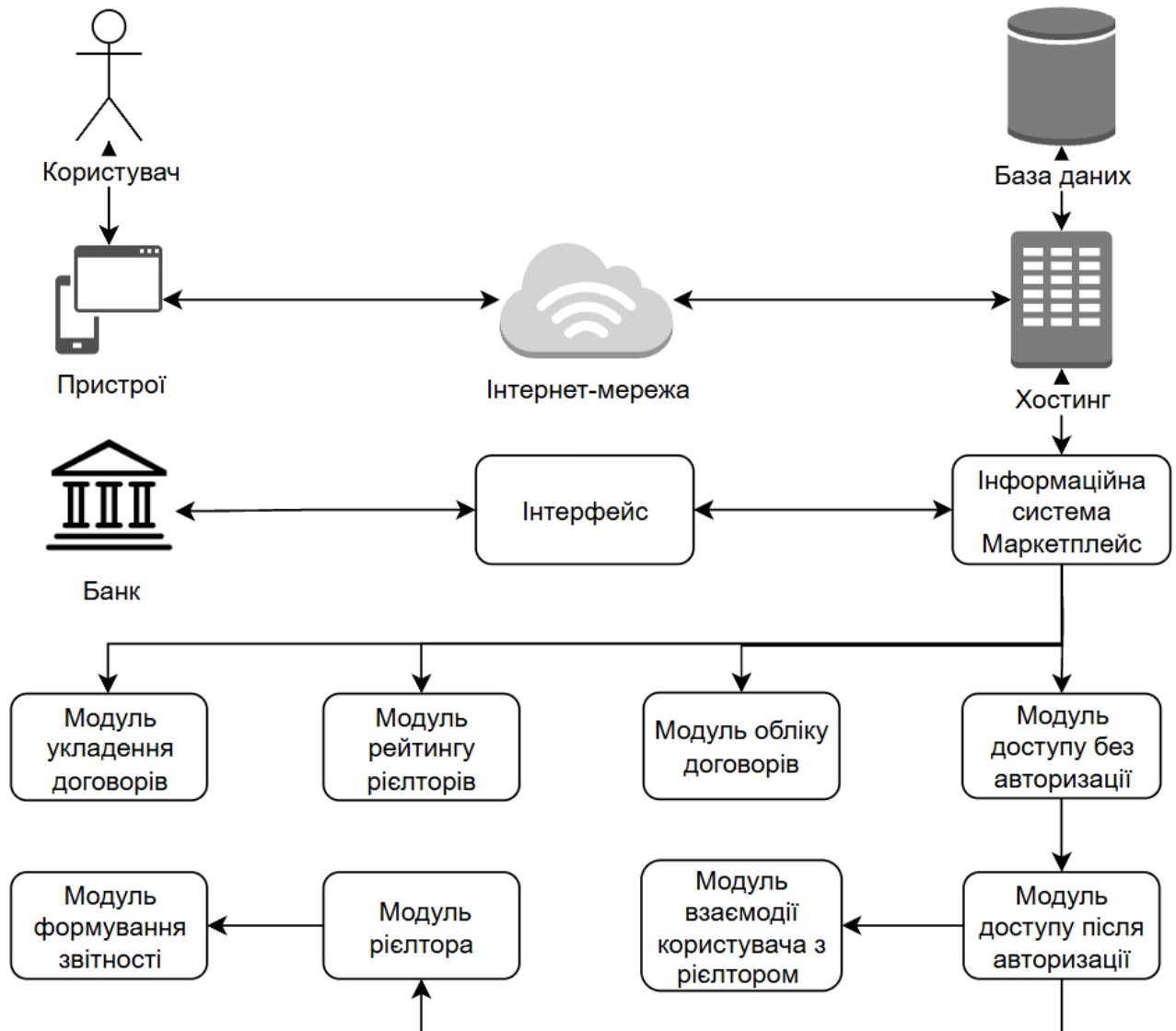


Рисунок 1.3.1 - Модель концепції інформаційної системи.

Для забезпечення зручного та ефективного процесу купівлі і продажу об'єктів нерухомості необхідно створити інформаційну систему у вигляді веб-маркетплейсу. Такий маркетплейс має бути реалізований у формі веб-сайту, який дозволяє користувачам взаємодіяти з системою через браузер на будь-якому

пристрої. Важливими вимогами до функціоналу системи є її інтуїтивна зрозумілість, багатофункціональність, відповідність вимогам безпеки та адаптивність до різних сценаріїв використання.

Особливу увагу приділено безпечності фінансових операцій. У системі реалізовано механізм захищених транзакцій із використанням депозитних рахунків, що гарантує збереження коштів до моменту підтвердження завершення угоди. Крім того, маркетплейс дотримується сучасних стандартів фінансової безпеки, таких як відповідність PCI DSS, підтримка моніторингу транзакцій у режимі реального часу, що мінімізує ризики шахрайства без зниження швидкості обробки платежів.

Основні функціональні модулі системи

Система складається з таких функціональних модулів:

Модулі доступу без авторизації - надають можливість перегляду публічно затверджених об'єктів нерухомості, ознайомлення з функціоналом платформи та загальними умовами співпраці.

Модулі доступу після авторизації - відкривають повний функціонал користувачу, зокрема:

- створення та редагування оголошень;
- спілкування через чати (листування);
- подання запитів на співпрацю з ріелтором;
- укладення договорів;
- перегляд звітності;
- взаємодія з банківською системою для здійснення платежів;
- підтвердження завершення угоди.

Модуль ріелтора - дозволяє ріелторам:

- переглядати запити клієнтів;

- приймати об'єкти у роботу;
- укладати договори з користувачами;
- супроводжувати угоди купівлі-продажу;
- формувати звіти про власну діяльність.

Модуль обліку договорів - відповідає за збереження, облік та контроль чинних договорів. Також модуль реалізує обмеження на кількість одночасних договорів у рієлтора (наприклад, не більше двох), що сприяє рівномірному розподілу об'єктів між фахівцями.

Модуль укладення договорів - реалізує створення формалізованих угод між користувачем і рієлтором. Договір передбачає зобов'язання рієлтора з пошуку покупця, розмір його винагороди та орієнтовні терміни виконання.

Модуль рейтингу рієлторів - забезпечує автоматичне формування рейтингу рієлторів на основі таких показників:

- кількість завершених угод;
- середній час продажу об'єкта;
- задоволеність користувачів;
- активність рієлтора;
- співвідношення взятих у роботу та завершених об'єктів.

Модуль взаємодії користувача з рієлтором - реалізує двосторонню комунікацію, де користувач може подати запит на співпрацю, а рієлтор може відгукнутись на оголошення. При цьому система враховує рейтинг фахівця, що дозволяє більш ефективно розподіляти об'єкти.

Модуль формування звітності - надає рієлторам детальну інформацію про:

- кількість об'єктів у роботі;
- кількість завершених угод;
- середню тривалість реалізації;
- суму реалізованих об'єктів.

Модуль взаємодії з банківськими системами - реалізований через інтеграцію з інтерфейсами фінансових установ, забезпечує:

- резервування коштів на депозитному рахунку;
- автоматичний розподіл коштів між сторонами угоди;
- безпечну обробку транзакцій з урахуванням вимог фінансових стандартів.

Взаємодія користувачів з системою.

Інформаційна система функціонує у вигляді сучасного веб-сайту, доступного з будь-якого пристрою через браузер. Основним завданням сайту є забезпечення повного циклу купівлі-продажу нерухомості: від створення оголошення до завершення фінансової угоди.

Користувач отримує доступ до функціоналу після реєстрації: може створювати оголошення про продаж або пошук житла, надсилати запити на співпрацю з ріелторами, переглядати стан угод, підписувати договори, проводити оплату через захищену систему депозитів, підтверджувати завершення угоди. Ріелтор працює через окреме інтерфейсне робоче місце на сайті, та має такі функціональні можливості:

- переглядає надіслані запити користувачів;
- обирає об'єкти для супроводу;
- укладає договори з клієнтами через функціонал системи;
- веде облік прийнятих об'єктів і стану виконання;
- отримує винагороду після завершення угоди;
- формує звіти щодо своєї діяльності;
- отримує автоматичну оцінку свого рейтингу.

Робоче середовище ріелтора дозволяє повністю контролювати кожен етап угоди. Він не лише супроводжує купівлю-продаж, а й спілкується з клієнтом через внутрішній чат, може надсилати пропозиції, ініціювати зміну умов

договору тощо. Усі договори укладаються через сайт, в електронному вигляді, та фіксуються в модулі обліку.

Всі процеси - від пошуку ріелтора до підтвердження платежу - автоматизовані та зафіксовані в системі, що виключає можливість підробки або шахрайських дій з боку будь-якої зі сторін[6].

Програмно-технічні засоби для реалізації проекту

Для реалізації веб-маркетплейсу будуть використані такі програмно-технічні засоби:

- Мова програмування: PHP - для реалізації серверної логіки веб-застосунку. Додатково використовується шаблонізатор Smarty, що забезпечує розділення логіки і представлення;
- Фронтенд: HTML, CSS, JavaScript - для створення інтерфейсу користувача;
- Серверна платформа: хмарне серверне середовище (наприклад, Render, Heroku або інфраструктура VPS на базі Ubuntu Server), що забезпечує розміщення веб-застосунку з використанням Apache HTTP Server;
- Система керування базами даних: MySQL - для зберігання інформації про користувачів, об'єкти нерухомості, договори, транзакції тощо;
- Операційна система: Windows - середовище розробки та тестування (можливе подальше розгортання на Linux-сервері, у ролі серверного середовища на хостинг-платформі);
- Інструменти розробки: PHPStorm - інтегроване середовище розробки, що використовується для написання та налагодження коду;
- Протоколи зв'язку: HTTP/HTTPS - для взаємодії між клієнтом і сервером;
- Фінансові сервіси: Інтеграція з банківською інфраструктурою здійснюється за допомогою API платіжного сервісу (наприклад, Monobank

API або аналогічного), що дозволяє реалізувати функції резервування коштів, підтвердження платежів і їх автоматичний розподіл;

- Сервісні компоненти:
 - Інтернет-з'єднання для забезпечення доступу до веб-ресурсу;
 - Засоби резервного копіювання (наприклад, автоматичні дампи БД) для забезпечення надійності зберігання даних;
 - Засоби моніторингу серверного навантаження для підтримання стабільної роботи системи.

Таке поєднання програмно-технічних засобів забезпечує стабільну, безпечну та масштабовану основу для реалізації всіх функціональних можливостей системи.

На основі сформованої концепції веб-маркетплейсу буде виконано реалізацію проєкту, яка дозволить досягти поставленої мети - підвищення ефективності роботи рієлтора у сфері купівлі та продажу нерухомості шляхом автоматизації основних процесів, забезпечення прозорості взаємодії з клієнтами та надання інструментів для зручного управління угодами.

У розділі 1 здійснено огляд предметної області маркетплейсів нерухомості, проведено аналіз існуючих рішень з визначенням їх переваг та недоліків. Виявлено потребу у розробці власної інформаційної системи, яка враховує особливості українського ринку та забезпечує підвищення ефективності купівлі-продажу нерухомості. Запропоновано концептуальні підходи до побудови системи, що створюють базу для подальшої розробки і впровадження функціонального маркетплейсу.

РОЗДІЛ 2 РОЗРОБЛЕННЯ ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Загальні системні рішення

Процес купівлі-продажу нерухомості в Україні у 2025 році відбувається згідно з чітко визначеним законодавством та включає кілька етапів, кожен з яких супроводжується оформленням офіційних документів та сплатою обов'язкових податків. З метою реалізації інформаційної системи, що автоматизує цей процес, необхідно враховувати як юридичні, так і технічні вимоги, які забезпечують коректну підтримку угод між покупцем, продавцем та рієлтором.

Укладення договору купівлі-продажу здійснюється у письмовій формі та підлягає обов'язковому нотаріальному посвідченню. Перед підписанням договору нотаріус перевіряє усі правостановлюючі документи на об'єкт нерухомості, зокрема: договір купівлі-продажу, свідоцтво про право власності, документи про спадщину або рішення суду. Вся інформація повинна бути внесена до Державного реєстру речових прав на нерухоме майно. Саме наявність цієї інформації в державному реєстрі дозволяє забезпечити юридичну чистоту угоди [15].

Система також має враховувати процедуру державної реєстрації права власності на нового власника. Цей етап є ключовим у завершенні угоди, оскільки право переходить лише після внесення відповідних даних до реєстру.

Окремо слід виділити питання податкового навантаження. Продавець, як правило, сплачує податок на доходи фізичних осіб у розмірі 5% або 18% (в залежності від тривалості володіння майном і кількості продажів протягом року), а також 1,5% військового збору. Покупець, у свою чергу, сплачує 1% від вартості об'єкта до Пенсійного фонду. Усі ці платежі повинні бути здійснені до

нотаріального посвідчення договору, а відповідні квитанції - передані нотаріусу [16].

Ріелтор виконує роль посередника: організовує покази об'єктів, консультує клієнтів, допомагає в підготовці документів, а також може супроводжувати сторони під час нотаріального оформлення. Саме через ріелтора в системі передбачається передача даних про угоду та контроль виконання обов'язкових дій усіх сторін.

Інформаційна система повинна враховувати всі ці процеси, забезпечуючи надійне збереження інформації, підтримку доступу до неї різними ролями (користувач, ріелтор, адміністратор), а також фіксацію статусу кожної угоди. Запровадження такої системи дозволить мінімізувати людський фактор, скоротити час на обробку даних і зменшити ймовірність помилок, що є критично важливим у сфері нерухомості.

1.2 Проектування баз даних

Опис предметної галузі.

Для забезпечення стабільної, надійної та масштабованої роботи інформаційної системи необхідно використовувати потужну систему керування базами даних (СКБД), яка дозволяє ефективно організовувати, зберігати та обробляти великі обсяги інформації, що генерується під час взаємодії користувачів з платформою. У межах даного проєкту було обрано MySQL - одну з найпоширеніших реляційних СКБД, яка входить до складу програмного середовища ХАМРР. Такий вибір зумовлений як функціональними можливостями системи, так і її широкою підтримкою в екосистемі РНР - основної мови, що використовується у серверній частині проєкту.

MySQL забезпечує високу продуктивність при обробці запитів до бази даних, що особливо важливо при роботі з оголошеннями, історією переглядів, транзакціями, користувацькими чатами та іншими критичними для системи модулями. Завдяки підтримці складних SQL-запитів і транзакцій, СКБД дозволяє реалізувати гнучкі механізми фільтрації, сортування та агрегації даних, що є необхідним для зручного пошуку об'єктів нерухомості [7].

Ключовими перевагами MySQL, що стали визначальними у контексті даної системи, є:

- **Продуктивність:** MySQL оптимізовано для високошвидкісної обробки запитів, що дозволяє ефективно працювати з великими обсягами інформації про об'єкти нерухомості, користувачів, фінансові транзакції та перегляди.

- **Гнучкість:** підтримка реляційної моделі та складних зв'язків між сутностями дозволяє проєктувати структуровану і цілісну базу даних, що спрощує обробку інформації та підтримку логіки предметної області.

- **Інтеграція з PHP:** MySQL безперешкодно взаємодіє з серверною логікою, реалізованою на PHP, що дає змогу швидко реалізовувати функції створення, оновлення, фільтрації та відображення даних у межах платформи.

- **Безпека:** підтримка механізмів авторизації, ролей і прав доступу дозволяє надійно контролювати доступ до даних, забезпечуючи захист від несанкціонованих дій.

Зважаючи на специфіку предметної області - а саме купівлю та продаж нерухомості - важливою вимогою до бази даних є її здатність зберігати та обробляти різні типи інформації, включно з такими атрибутами, як ціна, місцезнаходження, тип об'єкта, площа, кількість кімнат, стан, активність користувачів, історія дій, повідомлення в чатах тощо. Система також повинна надавати можливості для формування аналітики та статистики - наприклад,

середньої ціни по регіону або часу продажу об'єкта - що є важливою перевагою для ріелторів і адміністраторів платформи.

Для забезпечення цілісного представлення структури інформації, необхідної для функціонування системи, нижче представлено концептуальну, логічну та фізичну моделі бази даних. Вони були побудовані в середовищі draw.io на основі результатів аналізу предметної області, що дозволило врахувати всі сутності, взаємозв'язки між ними та відповідні атрибути.

Концептуальна модель даних - це загальний вигляд усіх даних, які побудовані на інформації, яка була отримана після аналізу предметної області. Також ця модель дозволяє побачити весь інформаційний зміст предметної області. На рисунку 2.2.1 представлена ER-діаграма, що зображує концептуальний рівень проектування, інформації системи бази даних для ріелторської контори. Діаграма має скорочений вид, без представлених атрибутів.

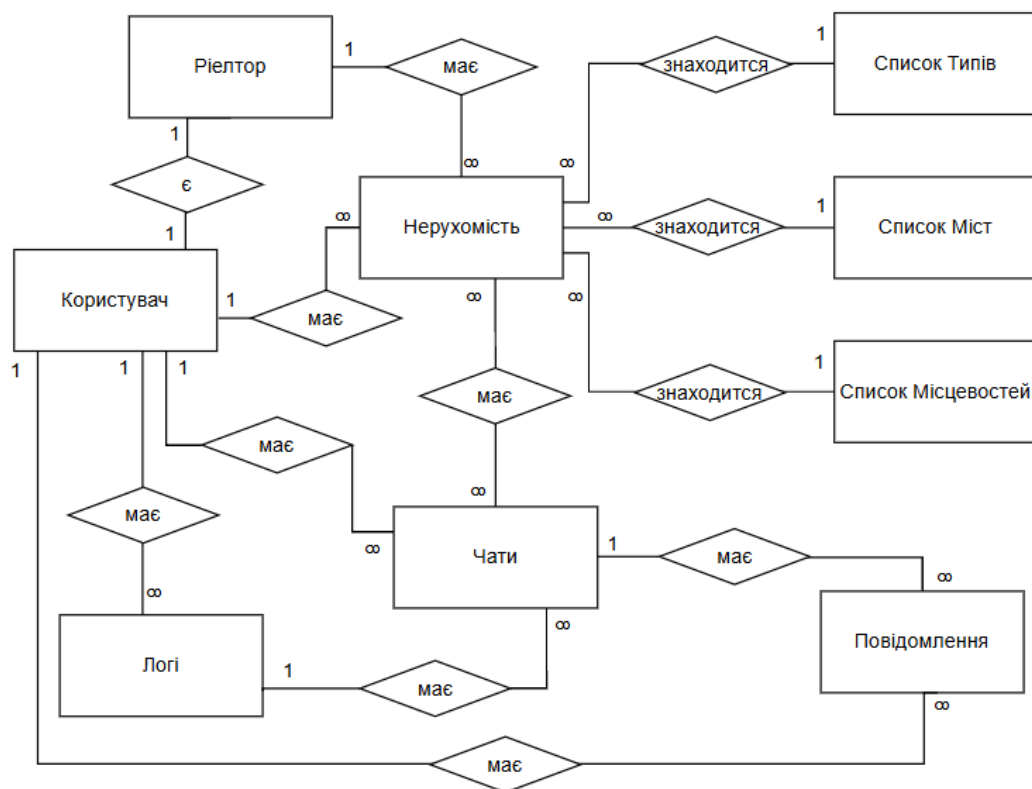


Рисунок 2.2.1 - ER-Діаграма сутностей у базі даних.

Під час аналізу предметної області, було виділено саме такі сутності: Користувач, Ріелтор, Нерухомість, Чати, Повідомлення, Історія перегляду, Логі, Список Типів, Список Міст, Список Місцевостей. У таблиці 2.2.1, зображено усі атрибути до цих сутностей.

Таблиця 2.2.1 - Опис всіх сутностей зі своїми атрибутами.

Сутність	Атрибути	Тип ключа
Користувач	id	Primary Key
	name	
	mail	
	password	
	birthday	
	pid	
	balance	
	time	
Ріелтор	id	Primary Key
	percent	
	experience	
Нерухомість	id	Primary Key
	title	
	cost	
	time	
	accepted	
	description	
	area	
	bedrooms	

	floors	
Чати	id	Primary Key
	time	
	sold	
Повідомлення	id	Primary Key
	message	
	time	
Логі	id	Primary Key
	type	
	link	
	desc	
	amount	
	time	
Список Типів	id	Primary Key
	name	
Список Міст	id	Primary Key
	name	
Список Місцевостей	id	Primary Key
	name	

Розробка логічної моделі даних для предметної області ріелторської контори включає ідентифікацію сутностей, визначення їхніх атрибутів і зв'язків між ними, а також побудована на базі спроектованої моделі даних. Також логічна модель даних визначає, як ці сутності взаємодіють і зберігаються у базі даних. При розгляданні концептуальної моделі бази даних ріелторської контори, ми отримуємо наступні зв'язки, що використовуємо у логічній моделі предметної області, що зображена на рисунку 2.2.2.

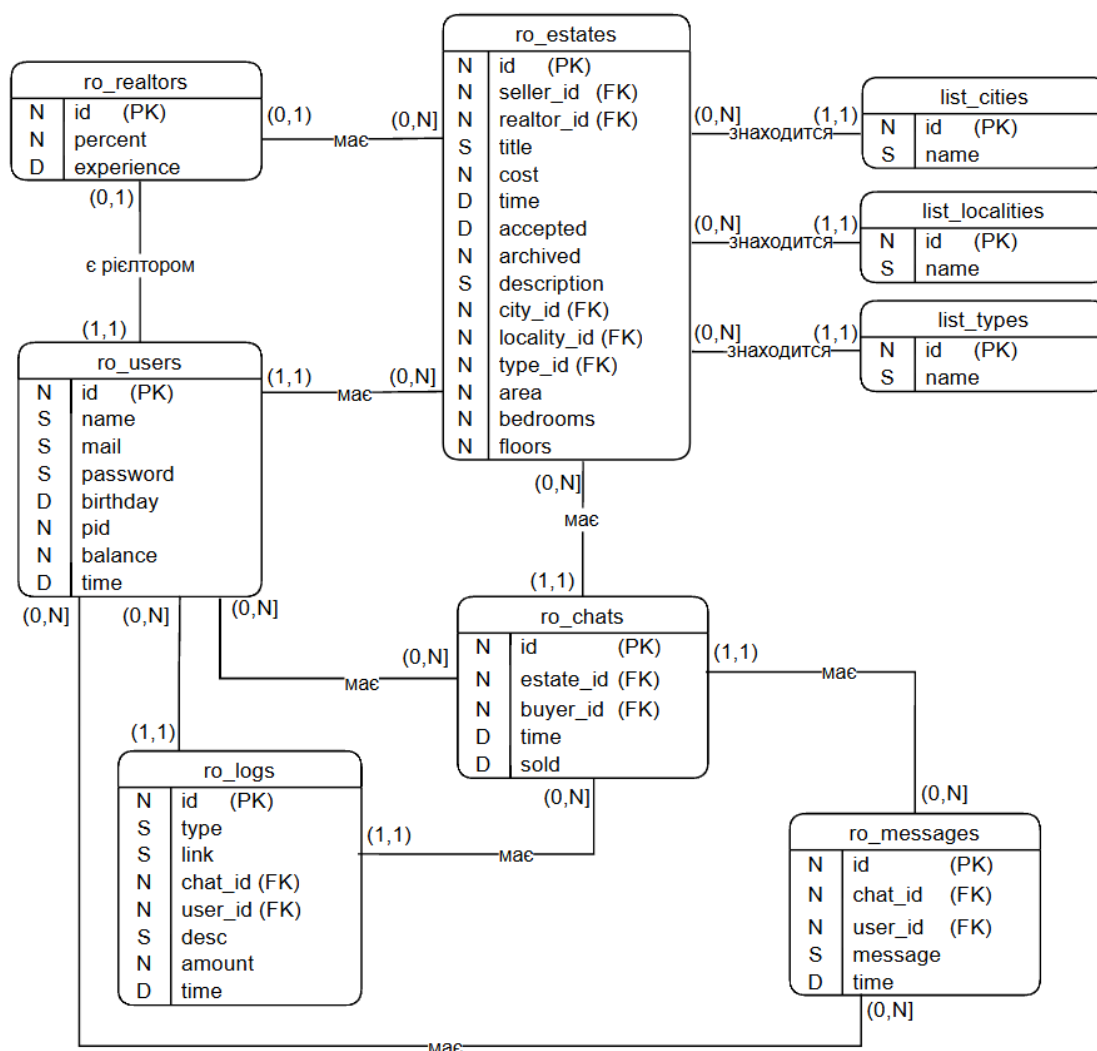


Рисунок 2.2.2 - ER-Діаграма логічної моделі предметної області.

Після опрацювання попередніх етапів розробки бази даних, було створено взаємозв'язані моделі, які використовувались під час формування загального уявлення про структуру ріелторської контори, та візуалізування його опису. Наступний етап проектування - розробка фізичної моделі даних, де вона буде описувати базу даних у вигляді конкретних технічних характеристик, а саме нормалізувати, тобто зробити декомпозицію попередніх відношень бази даних на більш прості відношення.

Детальні схемі всіх відношень бази даних з переліком атрибутів, їх типів і розмірів наведені у таблицях 2.2.2-2.2.10, а також на рисунку 2.2.3.

Таблиця 2.2.2 - Користувач.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id користувача	INT	11	Primary Key
name	ПІБ	VARCHAR	255	
mail	Електрона пошта	VARCHAR	255	
password	Пароль	VARCHAR	255	
birthday	Дата народження	DATE		
pid	Номер паспорта	INT	9	
balance	Баланс	DOUBLE		
time	Дата створення	DATETIME		

Таблиця 2.2.3 - Ріелтори.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id ріелтора	INT	11	Primary Key
percent	Процента ставка	DOUBLE		
experience	Дата початку	DATE		

Таблиця 2.2.4 - Нерухомість.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id нерухомості	INT	11	Primary Key
seller_id	Id продавця	INT	11	Foreign Key
realtor_id	Id ріелтора	INT	11	Foreign Key

title	Назва	TEXT		
cost	Ціна	DOUBLE		
time	Дата створення	DATETIME		
accepted	Дата узгодження	DATETIME		
archived	Статус активності	INT	1	
type_id	Тип нерухомості	INT	1	Foreign Key
locality_id	Місцевість	INT	1	Foreign Key
city_id	Місто	INT	1	Foreign Key
area	Площа	DOUBLE		
bedrooms	Кіл-ть кімнат	INT	5	
floors	Кіл-ть поверхів	INT	5	

Таблиця 2.2.5 - Чати.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id чати	INT	11	Primary Key
estate_id	Id нерухомості	INT	11	Foreign Key
buyer_id	Id покупця	INT	11	Foreign Key
time	Дата створення	DATETIME		
sold	Дата продажу	DATETIME		

Таблиця 2.2.6 - Повідомлення.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id повідомлення	INT	1	Primary Key
chat-id	Id чату	INT	11	Foreign Key
user_id	Id користувача	INT	11	Foreign Key
message	повідомлення	TEXT		

time	Дата створення	DATETIME		
------	----------------	----------	--	--

Таблиця 2.2.7 - Логі.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id логів	INT	11	Primary Key
type	Тип запису	TEXT		
link	Адреса виконання	TEXT		
user_id	Id користувачів	INT	11	Foreign Key
estate_id	Id нерухомості	INT	11	Foreign Key
desc	Опис запису	TEXT		
amount	Сумма переказу	INT	11	
time	Дата створення	DATETIME		

Таблиця 2.2.8 - Список типів.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id списка	INT	11	Primary Key
text	Назва	TEXT		

Таблиця 2.2.9 - Список міст.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
id	Id списка	INT	11	Primary Key
text	Назва	TEXT		

Таблиця 2.2.10 - Список місцевостей.

Назва в БД	Атрибути	Тип	Розмір	Тип ключа
------------	----------	-----	--------	-----------

id	Id списка	INT	11	Primary Key
text	Назва	TEXT		

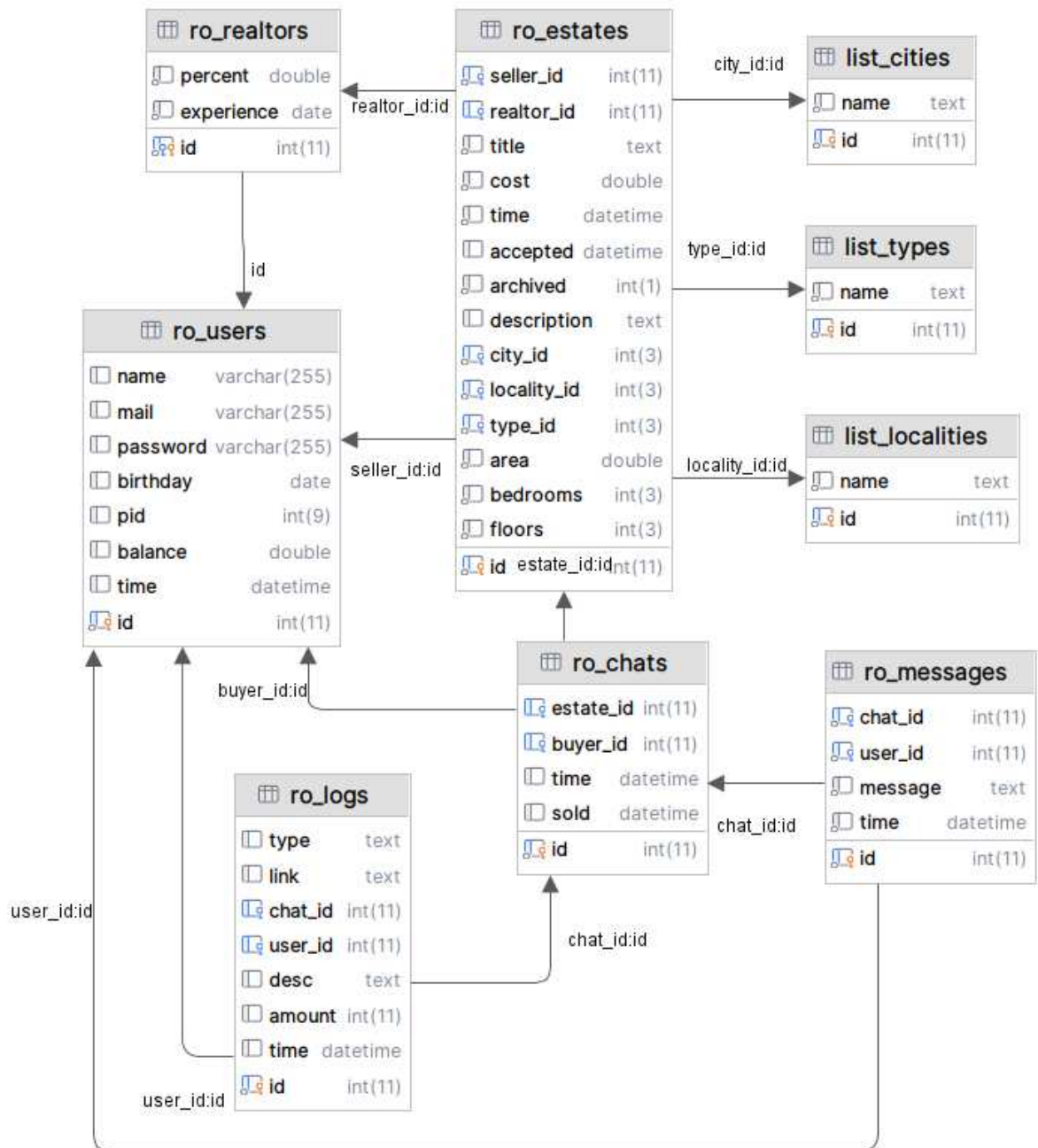


Рисунок 2.2.3 - ER-Діаграма фізичної моделі предметної області.

1.3 Проектування логіки функціонування системи

У подальшому розвитку інформаційної системи передбачається реалізація модуля рейтингу ріелторів, що має забезпечити об'єктивну оцінку ефективності роботи кожного фахівця. Такий підхід сприятиме прозорості сервісу, формуванню довіри серед користувачів та підвищенню якості наданих послуг.

Формування рейтингу ґрунтується на кількох ключових показниках, що відображають як професійну активність ріелтора, так і рівень задоволеності клієнтів. До таких показників належать:

- кількість завершених угод;

- середній час продажу об'єкта (чим швидше - тим краще);

- оцінки клієнтів після завершення угоди;

- загальна активність ріелтора у системі (наприклад, оновлення оголошень, відповіді на повідомлення);

- співвідношення між кількістю об'єктів, які ріелтор узяв у роботу, і кількістю успішно завершених угод.

Система обраховує для кожного ріелтора загальний рейтинг на основі цих параметрів. Для цього кожному показнику призначається певна вага, яка визначає його вплив на загальну оцінку. Наприклад, відгуки користувачів можуть мати найбільшу вагу, а активність - трохи меншу.

Фінальна формула обчислення рейтингу може мати вигляд:

$$R = (A * 0.3) + (B * 0.3) + (C * 0.2) + (D * 0.1) + (F * 0.1)$$

де:

- A - кількість успішно завершених угод (у нормалізованому вигляді);

- B - середня оцінка клієнтів після завершення угод (за шкалою від 1 до 5);

- C - середній час продажу об'єкта (перетворений у коефіцієнт: чим менший час - тим вищий бал);

D - активність ріелтора у системі (кількість оновлень оголошень, відповідей на повідомлення тощо);

F - співвідношення об'єктів, узятих у роботу, до успішно завершених (у вигляді частки або коефіцієнта).

Кожен показник у формулі береться у зручному для порівняння вигляді - або у балах (наприклад, оцінка клієнтів від 1 до 5), або у вигляді коефіцієнтів. Значення автоматично оновлюються після завершення нових угод чи отримання відгуків.

Алгоритм розрахунку рейтингу ріелтора подано на рисунку 2.3.1, що ілюструє основні етапи обробки даних та формування фінального показника.

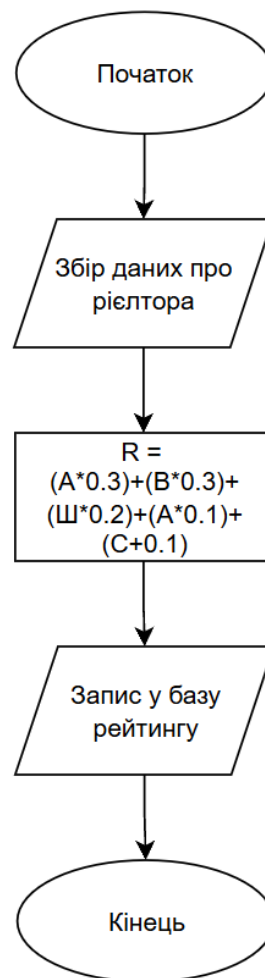


Рисунок 2.3.1 - Алгоритм розрахунку рейтингу ріелтора.

Таким чином, система зможе автоматично формувати рейтинг усіх рієлторів, сортувати їх за ефективністю та рекомендувати користувачам найкращих фахівців. У перспективі також передбачається можливість використання цих даних для побудови інтелектуальних рекомендацій або навіть прогнозування успішності майбутніх угод.

1.4Рішення з програмного забезпечення

Для створення веб-маркетплейсу було обрано набір програмних інструментів, що поєднує високу продуктивність, доступність у хмарних середовищах та активну підтримку спільнот. Нижче подано детальний опис кожного компонента із зазначенням його ролі та переваг у межах проєкту.

Серверна частина реалізована мовою PHP - популярним інструментом для розробки динамічних веб-застосунків, який відзначається великою кількістю готових бібліотек та швидким циклом оновлень [8]. Для відокремлення бізнес-логіки від представлення використовується шаблонізатор Smarty; він спрощує підтримку коду й дає змогу фронтенд-розробникам працювати з HTML-шаблонами, не торкаючись PHP-скриптів [9][10]. Веб-сервер Apache HTTP Server обробляє всі запити HTTPS і легко масштабується у хмарному VPS-оточенні на базі Ubuntu Server (Render, Heroku, DigitalOcean) [13][21].

База даних збудована на MySQL 8 - реляційній СУБД, оптимізованій для швидкої обробки складних запитів і великого обсягу транзакційних даних. MySQL підтримує ACID-транзакції, індекси та реплікацію, що гарантує цілісність інформації про користувачів, об'єкти нерухомості та фінансові операції [7]. Резервні копії формуються автоматично: повний дамп раз на добу та інкрементні - щогодини з відправленням у віддалене сховище.

Фронтенд побудовано на HTML5, CSS3 та JavaScript ES6 [10]. Для створення адаптивного інтерфейсу використано сучасні можливості CSS, включно з Flexbox і Grid-сіткою, що забезпечують коректне відображення сторінок на різних пристроях. На клієнтському боці реалізовано валідацію форм, сортування оголошень та асинхронне оновлення контенту.

Платіжна інтеграція здійснена через офіційні REST-API українських банків (Monobank тощо). Запити підписуються ключами, що зберігаються у файлах-оточення; усі параметри шифруються на сервері, а взаємодія відбувається лише через HTTPS. Це дає змогу резервувати кошти покупця, блокувати їх до підтвердження угоди й автоматично розподіляти комісію між продавцем, рієлтором та платформою.

Зберігання мультимедійного контенту (фотографії об'єктів нерухомості) організовано у хмарному сховищі Amazon - сервісі, що гарантує масштабованість і високу доступність файлів [22].

Інструменти розробки. Код пишеться й відлагоджується у середовищі PHPStorm, яке підтримує інтеграцію з MySQL, автодоповнення та статичний аналіз [12]. Система керування версіями Git використовується для командної роботи, гілкування й безпечного зберігання історії змін [11]. Управління залежностями виконує Composer (PHP) та npm (фронтенд). Налаштовано CI-процес: після кожного push запускаються тести, збірка та автоматичний деплой у тестове середовище.

Безпека забезпечується параметризованими SQL-запитами (захист від ін'єкцій), хешуванням паролів алгоритмом bcrypt, CSRF-токенами у формах і перевіркою ролей на кожному маршруті. Усі логіни, транзакції та помилки заносяться до захищених журналів для аудиту.

Обраний стек дає змогу швидко розгортати сервіс у хмарі, підтримувати високе навантаження та безпечно обробляти фінансові операції, що критично для ринку нерухомості.

1.5 Рішення з технічного забезпечення

Технічне забезпечення охоплює сукупність апаратних і програмних ресурсів, необхідних для розгортання та стабільної роботи веб-маркетплейсу з купівлі та продажу нерухомості. У межах проєкту передбачено використання таких компонентів технічної інфраструктури:

- Серверна частина: веб-застосунок може бути розгорнутий як на виділеному VPS-сервері з операційною системою Linux (наприклад, Ubuntu Server), так і на хмарних платформах, таких як Heroku, Render, або DigitalOcean. Для обробки HTTP-запитів використовується веб-сервер Apache у зв'язці з PHP.
- Клієнтські пристрої: доступ до маркетплейсу здійснюється з персональних комп'ютерів, ноутбуків, планшетів або смартфонів через веб-браузер. Інтерфейс адаптований для різних типів пристроїв, що забезпечує зручність користування.
- База даних: для зберігання структурованої інформації про користувачів, нерухомість, транзакції, договори тощо застосовується СУБД MySQL. Розгортання бази можливе як у межах серверного середовища, так і в хмарних базах даних (наприклад, ClearDB, PlanetScale).
- Засоби інтеграції з платіжними сервісами: для реалізації фінансової взаємодії між користувачами передбачена інтеграція з банківським API (зокрема, Monobank API або аналогічним), що дозволяє здійснювати бронювання коштів, підтвердження транзакцій і розподіл платежів між сторонами (продавець, ріелтор, платформа).
- Резервне копіювання: з метою збереження цілісності даних передбачено

регулярне створення резервних копій бази даних (автоматичні дампи з архівацією), які зберігаються у захищеному сховищі.

- Мережева інфраструктура: необхідне стабільне інтернет-з'єднання для підтримки постійного доступу до веб-ресурсу як зі сторони користувачів, так і для внутрішніх серверних процесів.
- Моніторинг і обслуговування: застосовуються інструменти контролю за навантаженням на сервер, збирання логів помилок і статистики доступів, що дозволяє вчасно виявляти аномалії та забезпечувати високу доступність платформи.

У розділі 2 проведено аналіз і розробку інформаційної системи для автоматизації купівлі-продажу нерухомості в Україні з урахуванням юридичних, податкових та організаційних вимог. Обрано СКБД MySQL завдяки її продуктивності, безпеці та сумісності з РНР. Спроектовано концептуальну, логічну та фізичну моделі бази даних, що відображають основні сутності та взаємозв'язки предметної області. Розроблена система сприяє прозорості угод, оптимізації роботи ріелторів та покращенню комунікації між учасниками процесу. Запропоновані рішення створюють надійну основу для подальшої розробки та впровадження платформи, яка відповідатиме вимогам сучасного ринку нерухомості, а також забезпечують масштабованість, безперервність та надійність функціонування інформаційної системи, створюючи основу для її подальшого розвитку та комерційного використання.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У межах реалізації інформаційної системи було розроблено комплексну функціональну частину, що охоплює основні потреби користувачів, рієлторів та адміністратора. Для зручності взаємодії з системою створено інтуїтивно зрозумілий інтерфейс, що супроводжується підказками та повідомленнями про помилки під час заповнення форм.

Кожна форма в системі має набір полів із валідацією, яка запобігає некоректному введенню даних. Користувач отримує повідомлення у випадку помилки або неповного заповнення, що дозволяє зменшити кількість неправильних операцій під час роботи з оголошеннями, повідомленнями чи транзакціями.

Перед впровадженням система була протестована на предмет коректної роботи всіх функціональних модулів. Особливу увагу приділено перевірці обміну даними між клієнтською та серверною частинами, а також коректності роботи сторонніх бібліотек і сервісів, використаних під час розробки.

Інформаційна система складається з низки функціональних модулів, що забезпечують комплексне виконання завдань користувачів різних ролей:

- Модулі доступу без авторизації - забезпечують перегляд публічних оголошень про нерухомість, ознайомлення з функціоналом платформи та умовами співпраці;
- Модулі доступу після авторизації - відкривають користувачам можливість створювати і редагувати оголошення, вести листування через внутрішні чати, подавати запити на співпрацю з рієлторами, укладати договори, переглядати звіти та здійснювати платежі;
- Модуль рієлтора - надає інтерфейс для прийому та обробки запитів

клієнтів, ведення договорів, супроводу угод, формування звітності та управління рейтингом;

- Модуль обліку договорів - відповідає за збереження, контроль та обмеження кількості активних договорів;
- Модуль рейтингу ріелторів - автоматично формує рейтинг на основі кількості угод, часу реалізації об'єктів, активності та задоволеності користувачів;
- Модуль взаємодії користувачів з ріелторами - реалізує двосторонню комунікацію та розподіл об'єктів з урахуванням рейтингу;
- Модуль формування звітності - надає детальні звіти про діяльність ріелторів;
- Модуль інтеграції з банківськими системами - забезпечує безпечне резервування коштів, розподіл платежів та обробку транзакцій.

Крім того, значна увага була зосереджена на аспектах безпеки: усі сторінки системи мають перевірку рівня доступу користувача, що гарантує, що доступ до критичних функцій і даних можливий лише для авторизованих ролей (користувач, ріелтор, адміністратор). Запити до бази даних і відображення інформації здійснюються з урахуванням прав доступу, що мінімізує ризик несанкціонованого доступу та забезпечує безпечну взаємодію з системою. Базу даних було попередньо заповнено тестовими даними, що дозволило перевірити працездатність системи в умовах, наближених до реальних.

Опис основних SQL-запитів

Нижче наведено найважливіші запити, що забезпечують роботу ключових модулів системи.

- Отримання списку ріелторів

Метод: `getArrayRealtorsId()` наведено на лістингу 3.1.

Лістинг 3.1

```
SELECT id
FROM ro_realtors;
```

- Дані для головної сторінки (активні оголошення)

Метод: `getHomePageEstates()` наведено на лістингу 3.2.

Лістинг 3.2

```
SELECT    e.id, e.title, e.cost, e.time, e.description, cities.name AS
city_name, types.name AS type_name
FROM ro_estates AS e
JOIN list_cities AS cities ON e.city_id = cities.id
JOIN list_types  AS types  ON e.type_id = types.id
JOIN ro_chats AS c ON e.id = c.estate_id
WHERE e.accepted IS NOT NULL
AND e.archived = 0
AND c.sold IS NULL
ORDER BY e.time DESC;
```

- Авторизація користувача

Метод: `login($mail, $password)` наведено на лістингу 3.3.

Лістинг 3.3

```
SELECT id
FROM ro_users
WHERE password = '$password'
      AND mail      = '$mail'
LIMIT 1;
```

- Реєстрація користувача

Метод: `register($name, $mail, $password, $birthday, $PID)` наведено на лістингу 3.4.

Лістинг 3.4

```
-- Перевірка унікальності e-mail та PID
SELECT id
```

```

FROM ro_users
WHERE mail = '$mail' OR PID = '$PID';

-- Вставка нового користувача
INSERT INTO ro_users (name, mail, password, birthday, PID, balance, time)
VALUES ('$name', '$mail', '$password', '$birthday', '$PID', '0', '$time');

```

- Реєстрація ріелтора

Метод: registerRealtor(\$name, \$mail, ...) наведено на лістингу 3.5.

Лістинг 3.5

```

INSERT INTO ro_realtors (id, percent, experience)
VALUES ('$realtor_id', '$percent', '$experience');

```

- Повна інформація про конкретний об'єкт

Метод: getEstateById(\$estate_id) наведено на лістингу 3.6.

Лістинг 3.6

```

SELECT e.id, e.seller_id, e.realtor_id, e.title, e.cost, e.time,
e.description, e.area, e.bedrooms, e.floors, cities.name AS city_name,
localities.name AS locality_name, types.name AS type_name, e.city_id,
e.locality_id, e.type_id, e.archived
FROM ro_estates AS e
JOIN list_cities AS cities ON e.city_id = cities.id
JOIN list_localities AS localities ON e.locality_id = localities.id
JOIN list_types AS types ON e.type_id = types.id
WHERE e.id = '$estate_id'
LIMIT 1;

```

- Створення чату «покупець - об'єкт»

Метод: createChat(\$id, \$buyer_id) наведено на лістингу 3.7.

Лістинг 3.7

```

INSERT INTO ro_chats (estate_id, buyer_id, time)
VALUES ('$id', '$buyer_id', '$time');

```

- Отримання чатів користувача (варіант для покупця)

Метод: `getChatsByUID($uid, $accessLevel, 'buy')` наведено на лістингу 3.8.

Лістинг 3.8

```
SELECT e.id AS estate_id, e.title, e.seller_id, e.realtor_id, e.cost,
d.buyer_id, d.time, d.id AS chat_id, d.sold
FROM ro_chats AS d
JOIN ro_estates AS e
ON d.estate_id = e.id
WHERE d.buyer_id = '$uid'
ORDER BY d.sold, d.time DESC;
```

- Підтвердження продажу та фінансовий розподіл

Метод: `buyEstate($estate_id)` (Фрагмент набору запитів усередині транзакції) наведено на лістингу 3.9.

Лістинг 3.9

```
-- Фіксація факту продажу
UPDATE ro_chats
SET sold = '$time'
WHERE estate_id = '$estate_id' AND buyer_id = '$uid';

-- Архівация оголошення
UPDATE ro_estates
SET archived = 1
WHERE id = '$estate_id';

-- Зміна балансу покупця
UPDATE ro_users
SET balance = balance - $estate_cost
WHERE id = '$uid';

-- Зміна балансу ріелтора
UPDATE ro_users
SET balance = balance + $realtor_tax
WHERE id = '$realtor_id';
```



```
-- Зміна балансу продавця
UPDATE ro_users
SET balance = balance + $seller_part
WHERE id = '$seller_id';
```

- Рейтинг ріелторів за рік

Метод: `selectTopRealtorsByYear($year)` наведено на лістингу 3.10.

Лістинг 3.10

```
SELECT r.id, u.`name`, COUNT(*) AS count, $year AS year
FROM ro_realtors AS r
JOIN ro_users AS u ON r.id = u.id
JOIN ro_estates AS e ON e.realtor_id = r.id
JOIN ro_chats AS c ON c.estate_id = e.id AND YEAR(c.sold) = '$year'
GROUP BY r.id
HAVING count = (
SELECT MAX(TotalSales)
FROM (SELECT COUNT(*) AS TotalSales
FROM ro_realtors r1
JOIN ro_users u1 ON r1.id = u1.id
JOIN ro_estates e1 ON e1.realtor_id = r1.id
JOIN ro_chats c1 ON c1.estate_id = e1.id
WHERE YEAR(c1.sold) = '$year'
GROUP BY r1.id) AS MaxResult);
```

- Перелік усіх оголошень для адміністратора

Метод: `selectEstates()` наведено на лістингу 3.11.

Лістинг 3.11

```
SELECT e.id, u_s.id AS seller_id, u_s.`name` AS seller_name, u_s.mail AS
seller_mail, u_r.id AS realtor_id, u_r.`name` AS realtor_name, u_r.mail AS
realtor_mail, e.title, e.cost, e.time, e.accepted, e.archived, e.`description`,
cities.name AS city_name, types.name AS type_name, localities.name AS
locality_name, e.area, e.bedrooms, e.floors
FROM ro_estates AS e
JOIN list_cities AS cities ON e.city_id = cities.id
JOIN list_types AS types ON e.type_id = types.id
```

```

JOIN list_localities AS localities ON e.locality_id = localities.id
JOIN ro_users AS u_s ON e.seller_id = u_s.id
JOIN ro_users AS u_r ON e.realtor_id = u_r.id
ORDER BY e.time DESC;

```

- Додавання нового об'єкта

Метод addEstate(...) наведено на лістингу 3.12.

Лістинг 3.12

```

INSERT INTO ro_estates (seller_id, realtor_id, title, cost, `time`,
`description`,city_id, locality_id, type_id, area, bedrooms, floors)
VALUES ($seller_id, NULL, '$title', $cost, '$created', '$description',
$city, $locality, $type, $area, $bedrooms, $floors);

```

Наведені запити покривають ключові CRUD-операції, авторизацію, фінансові транзакції та звітність, забезпечуючи повний цикл роботи веб-маркетплейсу: від реєстрації користувачів і розміщення оголошень до завершення угод і формування статистики.

На головній сторінці (рисунок 3.1) відображається загальний перегляд усіх активних нерухомостей, які доступні зареєстрованим і не зареєстрованим користувачам. А також доступно у шапці сторінці лише посилання на домашню сторінку, вхід до застосунку, та реєстрація.

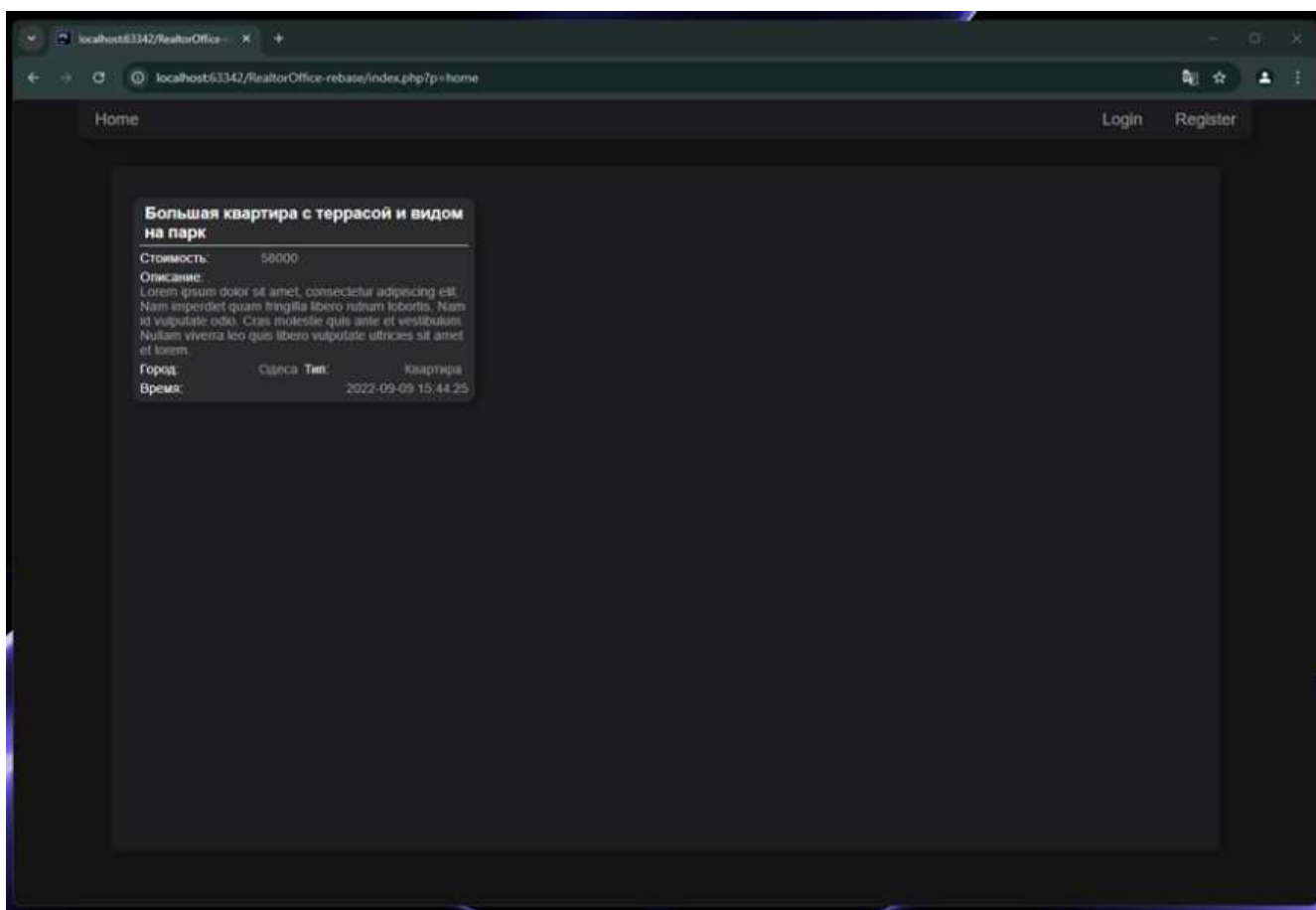


Рисунок 3.1 - Домашня сторінка.

Можна перейти на сторінку конкретної нерухомості натиснувши на неї, щоб побачити більш детальні характеристики нерухомості, що зображено на рисунку 3.2.

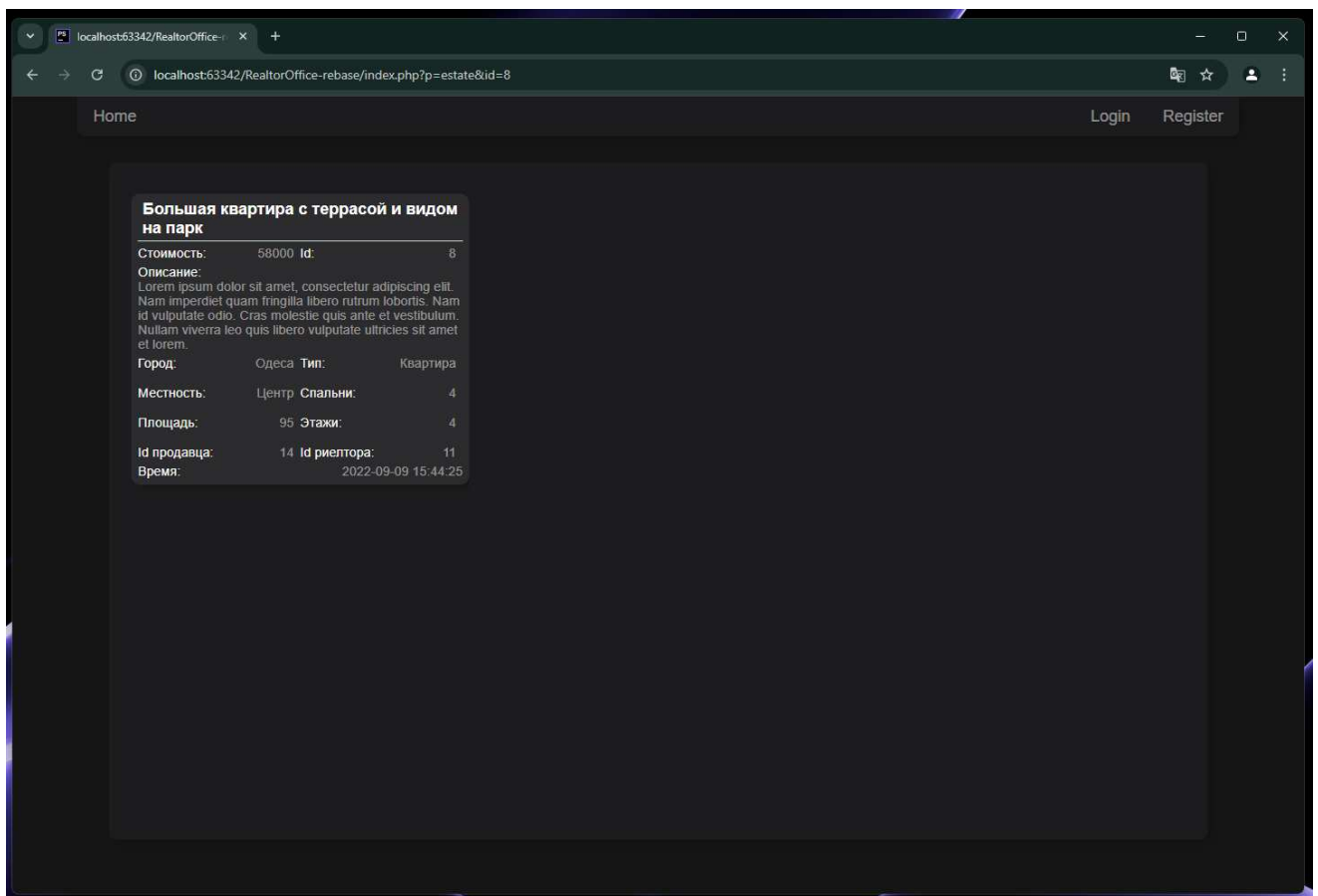


Рисунок 3.2 - Сторінка нерухомості.

На сторінці входу до аккаунту (рисунок 3.3) відображається форма заповнення username та password, які потім перевіряються з даними із бази даних ріелторської контори після натискання кнопки Login.

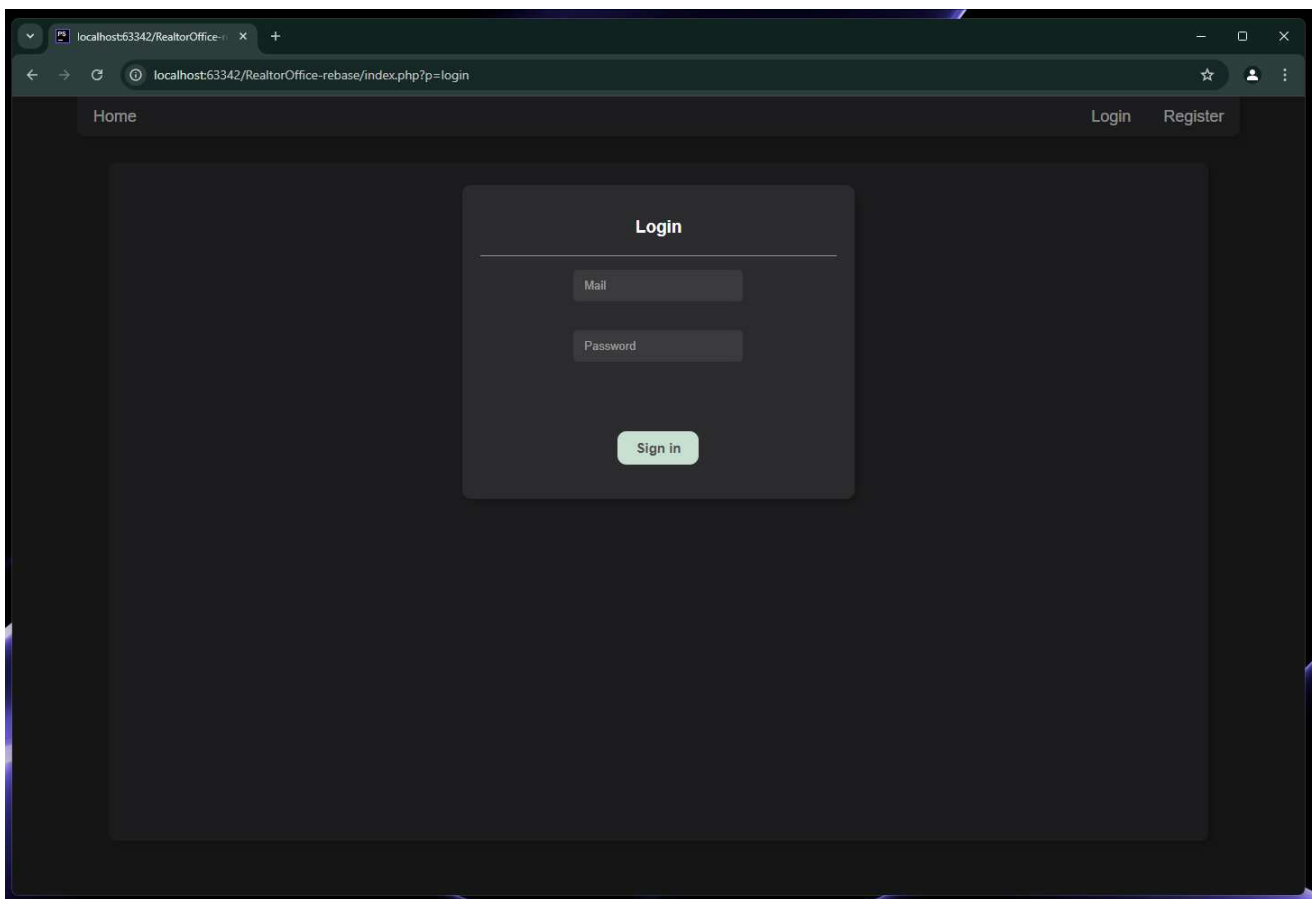


Рисунок 3.3 - Сторінка авторизації.

На сторінці реєстрації акаунту (Рисунок 3.4) користувача пропонується заповнити форму з особистими даними, такими як ім'я, прізвище, електронна пошта, телефон та пароль. Після натискання кнопки “Sign Up” введені дані будуть перевірені на коректність та, в разі успішної перевірки, збережені в базі даних для подальшого використання користувачем при вході до системи. Також він може зареєструватися як ріелтор, натиснувши кнопку “Sign Up as Realtor”.

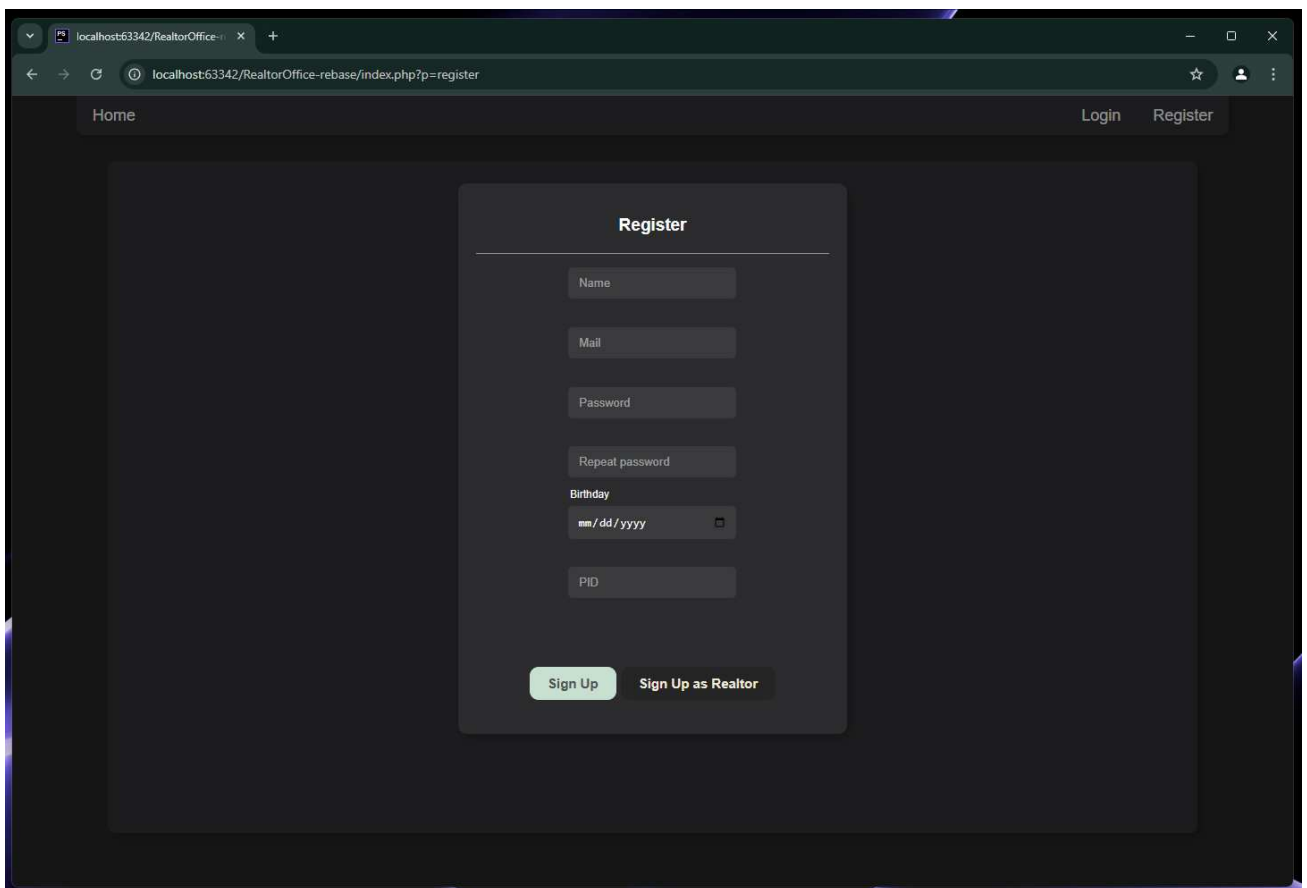


Рисунок 3.4 - Сторінка реєстрації.

На сторінці реєстрації як ріелтор (Рисунок 3.4) користувачу пропонується заповнити форму з особистими даними, такими як ім'я, прізвище, електронна пошта, телефон та пароль. Та додатково ще потрібно заповнити форму з датою розпочату кар'єри ріелтора.

Після натискання кнопки “Sign Up” введені дані будуть перевірені на коректність та, в разі успішної перевірки, збережені в базі даних для подальшого використання користувачем при вході до системи.

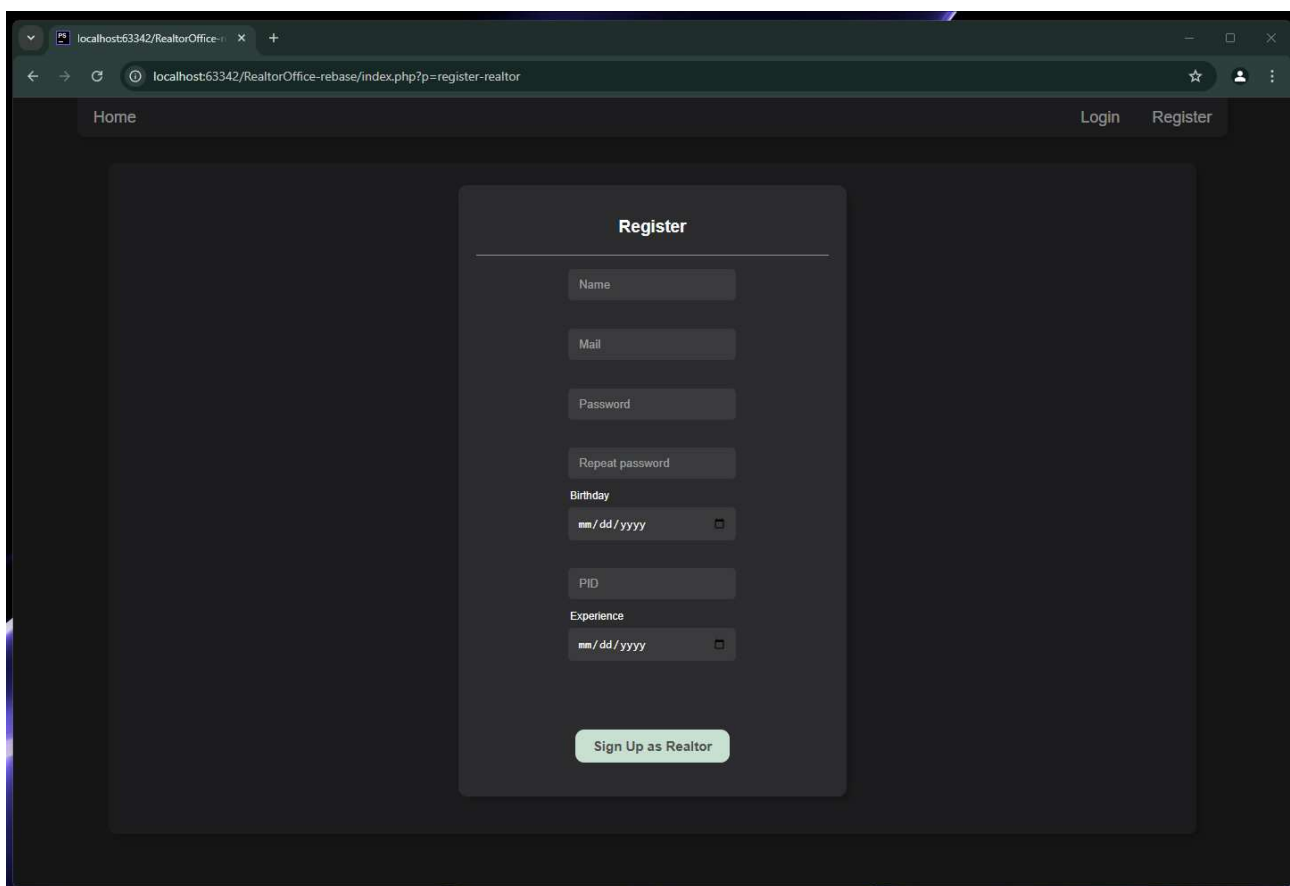


Рисунок 3.5 - Сторінка реєстрації як ріелтора.

На сторінці редагування нерухомості (рисунок 3.6) користувач побачить всі дані, які вже наявні в базі даних щодо вибраної нерухомості. Ці дані автоматично завантажуються у відповідні поля форми, дозволяючи користувачу переглянути та відредагувати їх. Поряд зі звичайними полями для введення інформації про нерухомість, користувач також бачитиме переключатель для встановлення статусу активності нерухомості. Після внесення змін користувач може натиснути кнопку "Save", щоб дані були оновлені в базі даних.

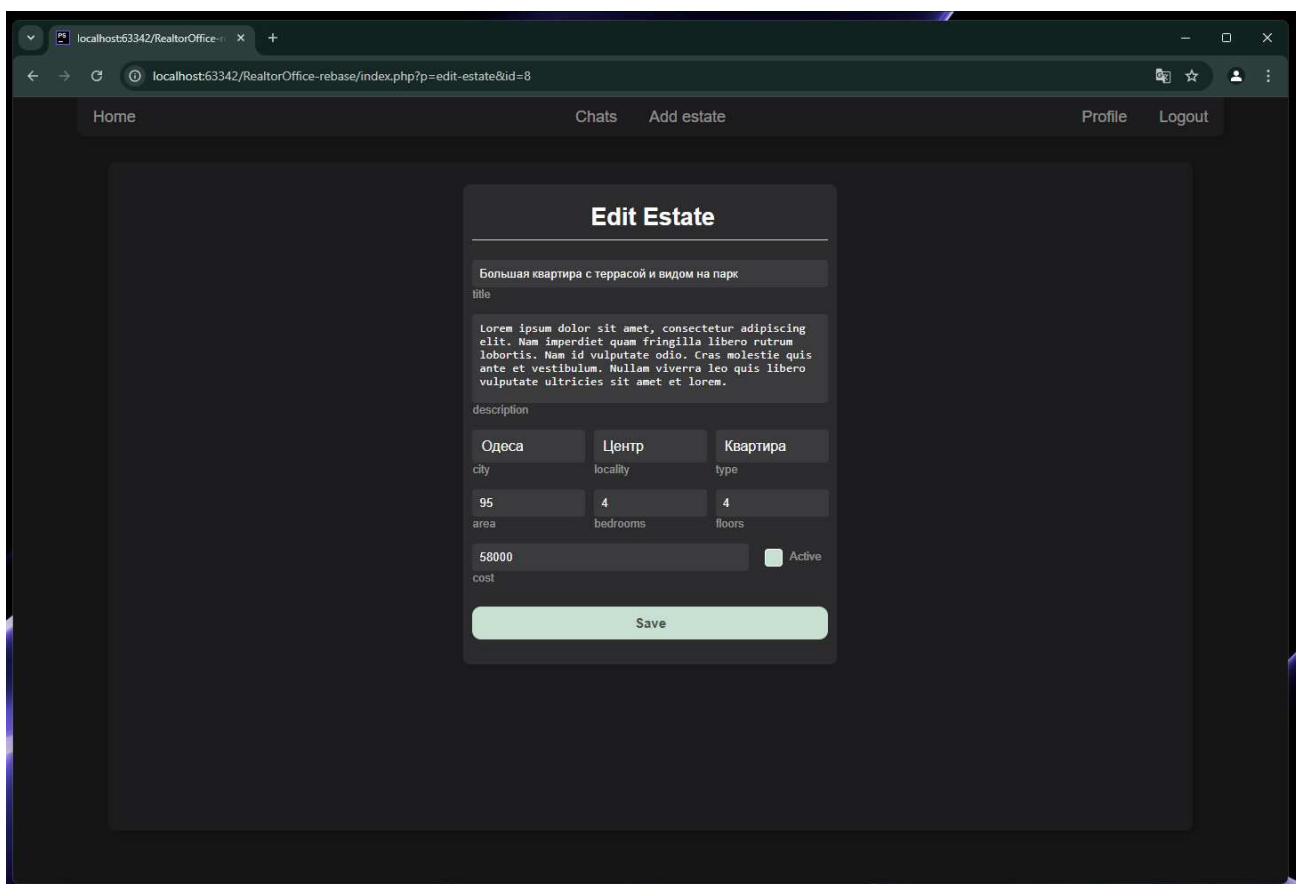


Рисунок 3.6 - Сторінка редагування нерухомості.

На сторінці додавання нерухомості (рисунок 3.7) користувач може ввести всі необхідні дані про нову нерухомість, такі як назва, ціна, площа, кількість кімнат, інформація про поверх, тип нерухомості, місто та місцевість. Після введення даних користувач натискає кнопку "Add Estate", і дані відправляються до бази даних.

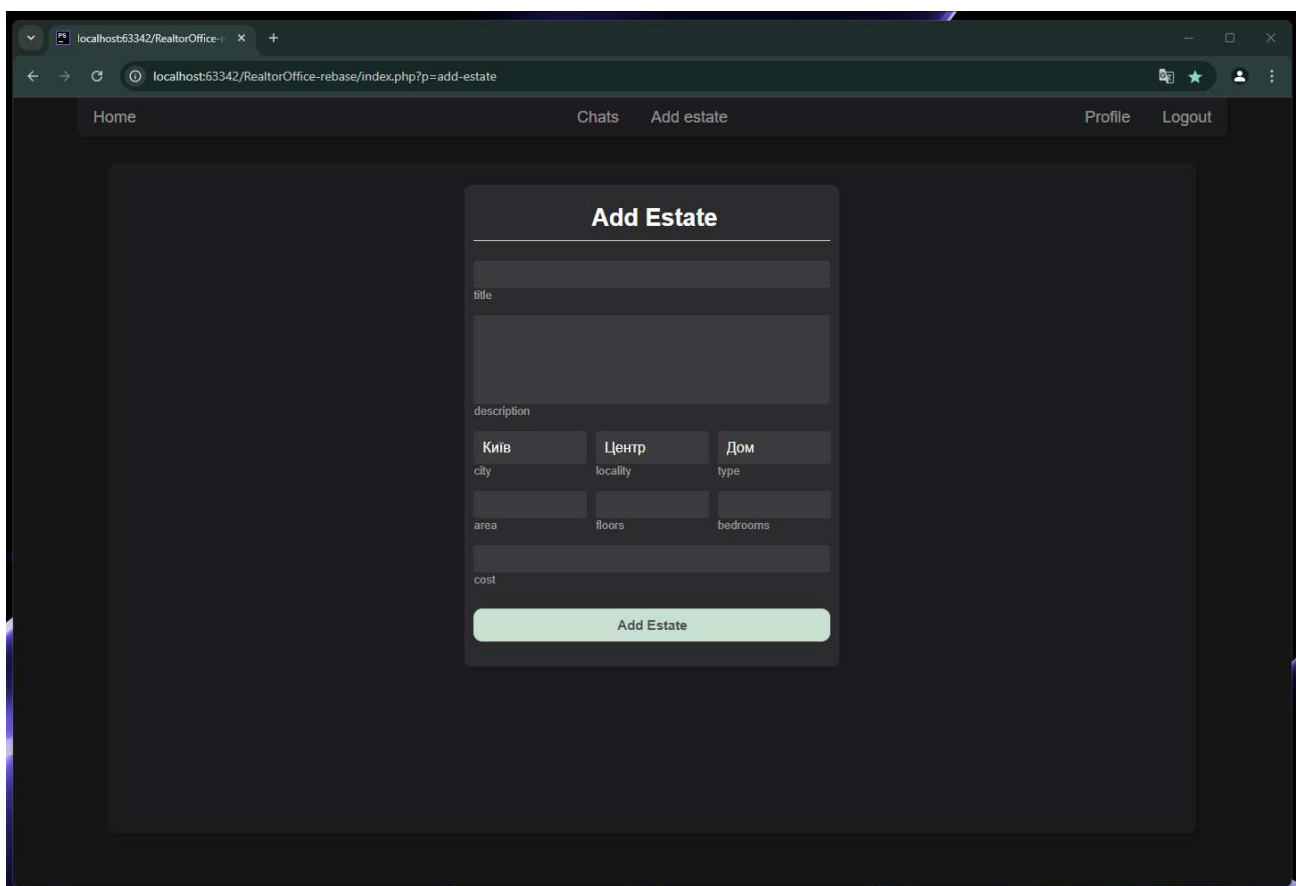


Рисунок 3.7 - Сторінка додавання нерухомості.

На сторінці переліку чатів (Рисунок 3.8 та 3.9) користувачу пропонується обрати спочатку чат, у якому він зможе побачити повідомлення від нього, чи інших користувачів, що є у цьому чаті, усі повідомлення беруться із бази даних. Також користувач має змогу відправляти повідомлення у чат, що виконується при натисненні на кнопку Send, яка обробляє це, і відправляє запит до бази даних, з відправкою даних.

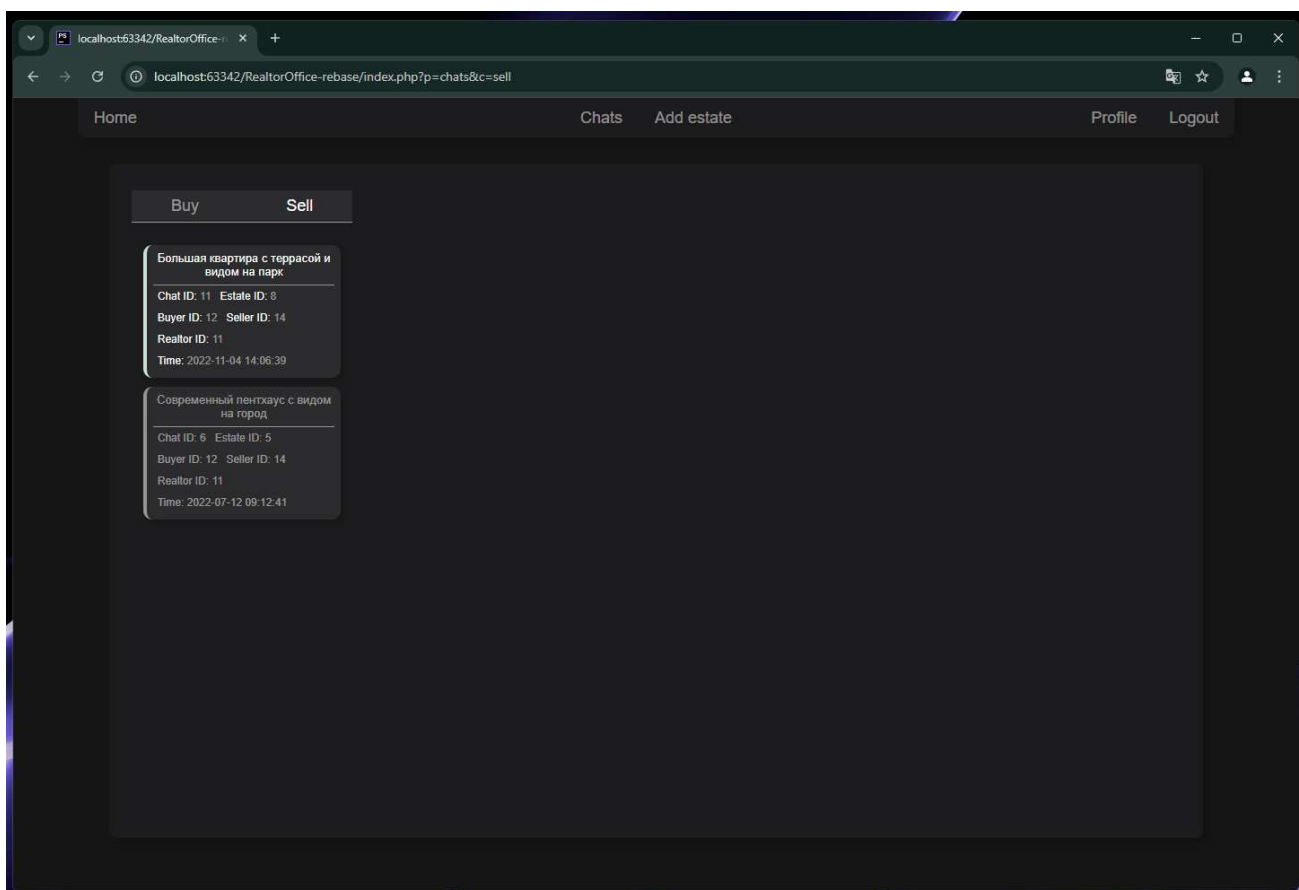


Рисунок 3.8 - Початкова сторінка чатів.

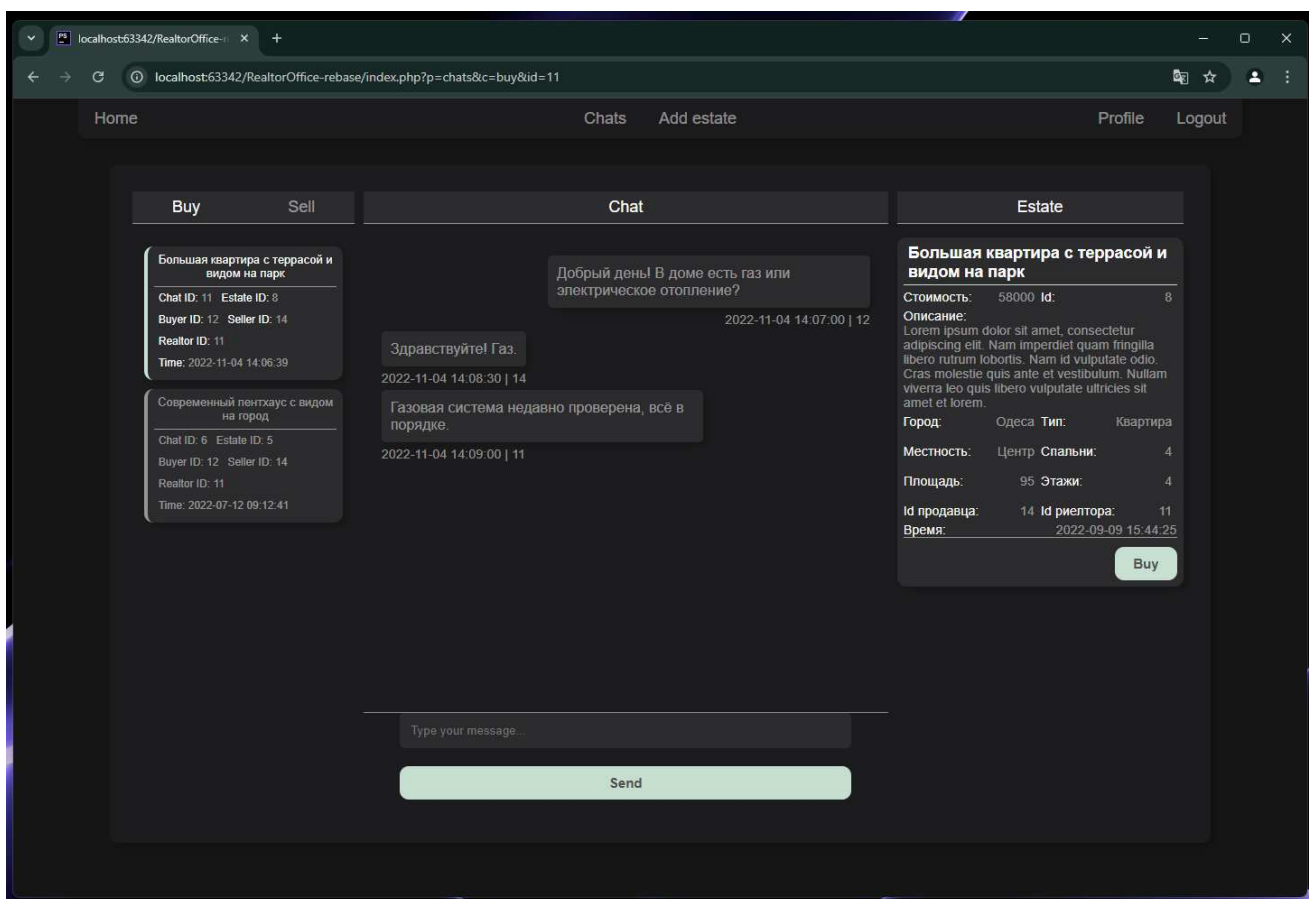


Рисунок 3.9 - Сторінка з повідомленнями обранного чату.

На сторінці покупки нерухомості (рисунок 3.10) покупцем спочатку відображається інформація про саму нерухомість, яку вони збираються купити. Це може бути назва, тип, ціна та інші характеристики. Під цими даними відображається поле з розрахунковою повною вартістю нерухомості з урахуванням відсотку ріелтора. У випадку, якщо на балансі покупця недостатньо коштів для покупки, відображається інформація про поточний баланс.

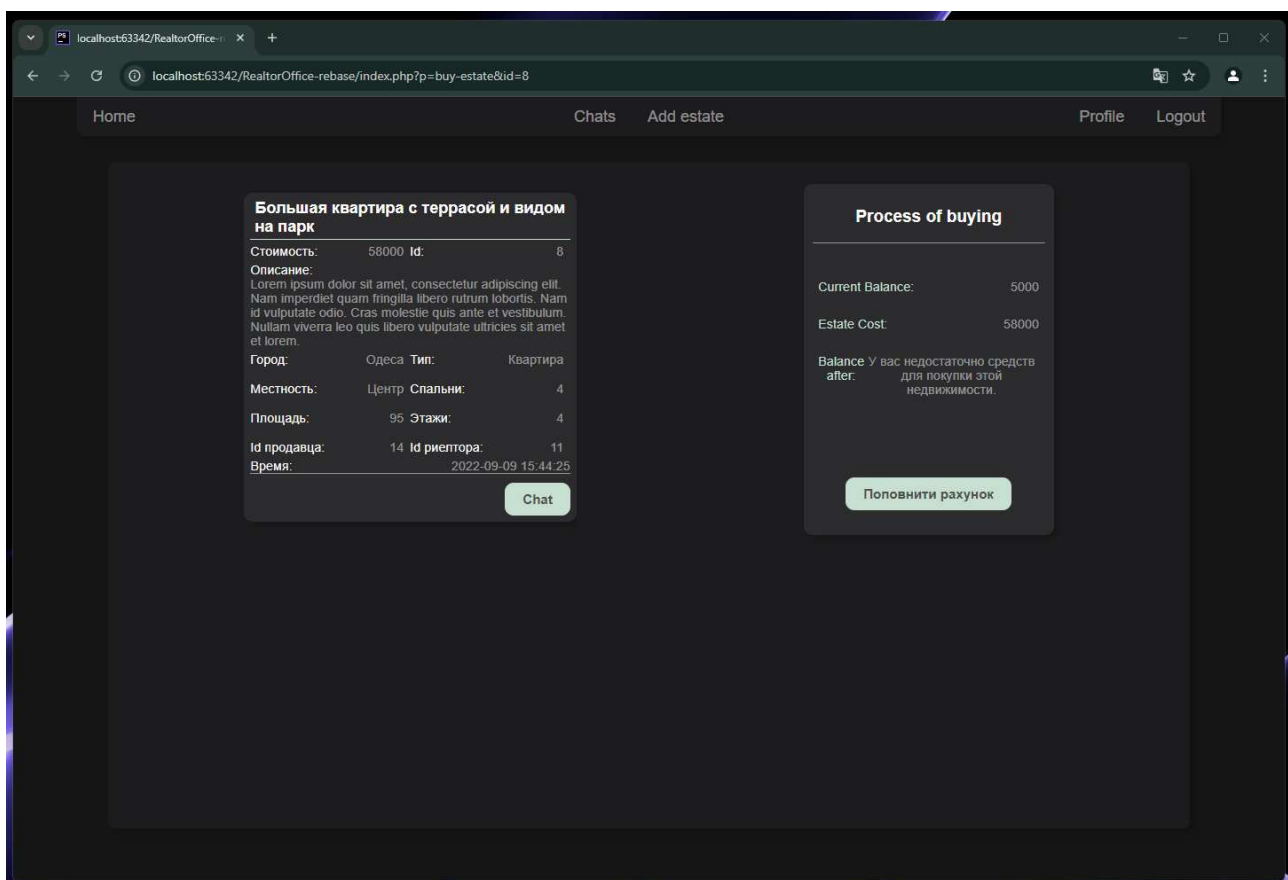


Рисунок 3.10 - Сторінка процессу покупку нерухомості (коли недостатньо коштів).

(При наявності рахунку) на сторінці покупки нерухомості (рисунок 3.11) відображається та ж сама інформація про нерухомість, а також розрахункова вартість з урахуванням комісії ріелтора. Під цими даними розташована кнопка "Підтвердити", яка ініціює процес транзакції. При натисканні на цю кнопку кошти знімаються з балансу покупця, а ріелтору передаються кошти з урахуванням відсотку. Нерухомість також видаляється з ринку та додається до історії транзакцій у вигляді запису в логах.

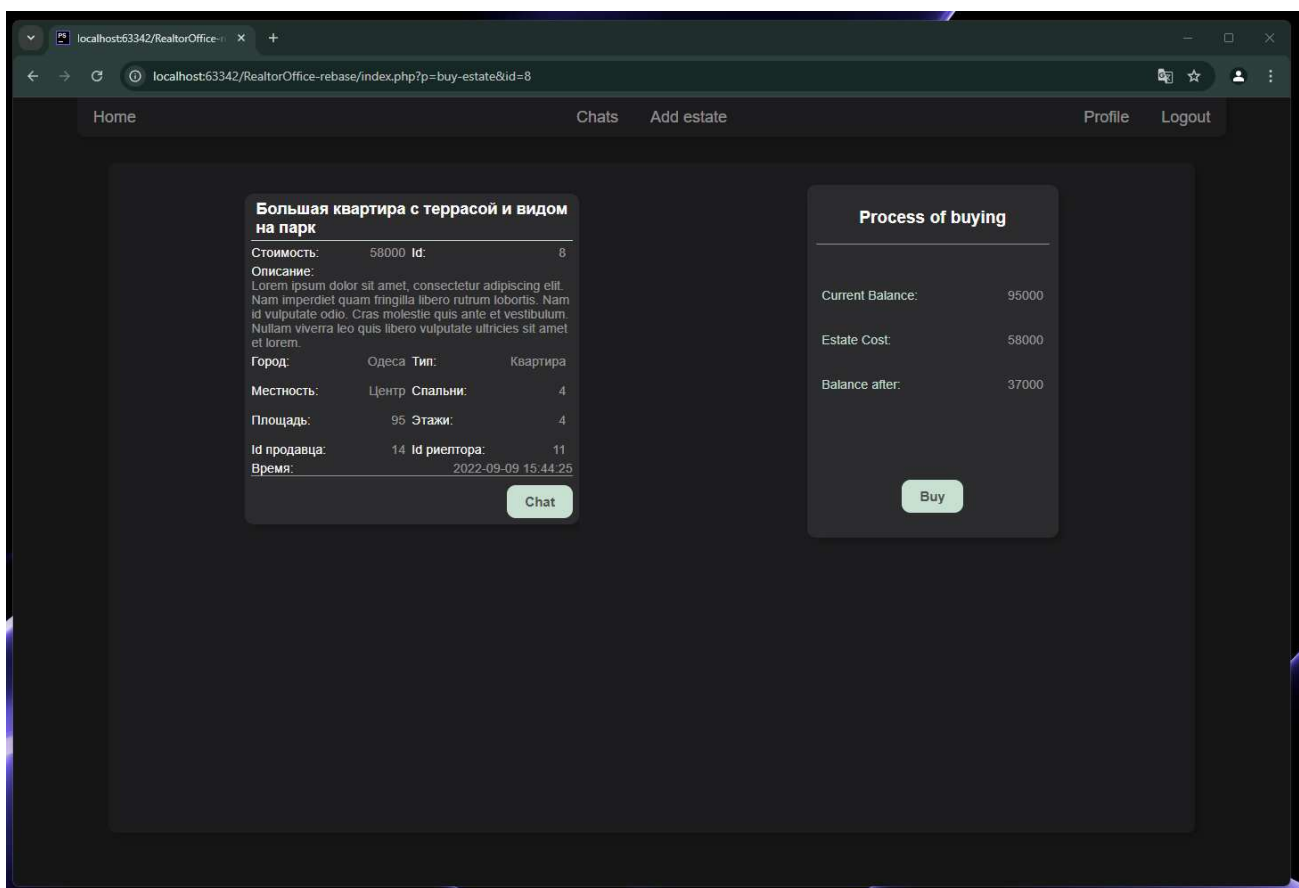


Рисунок 3.11 - Сторінка процесу покупки нерухомості (коли достатньо рахунку).

На цій сторінці (рисунок 3.12) відображається список нерухомості, яка була відправлена на затвердження ріелтору. Кожен запис містить детальну інформацію про об'єкт: адресу, опис, вартість, та інші характеристики.

Під кожним записом розташована кнопка "Асерт", натискання на яку ініціює процес затвердження нерухомості. Тільки ріелтор, авторизований у системі, може приймати об'єкти, відправлені на затвердження.

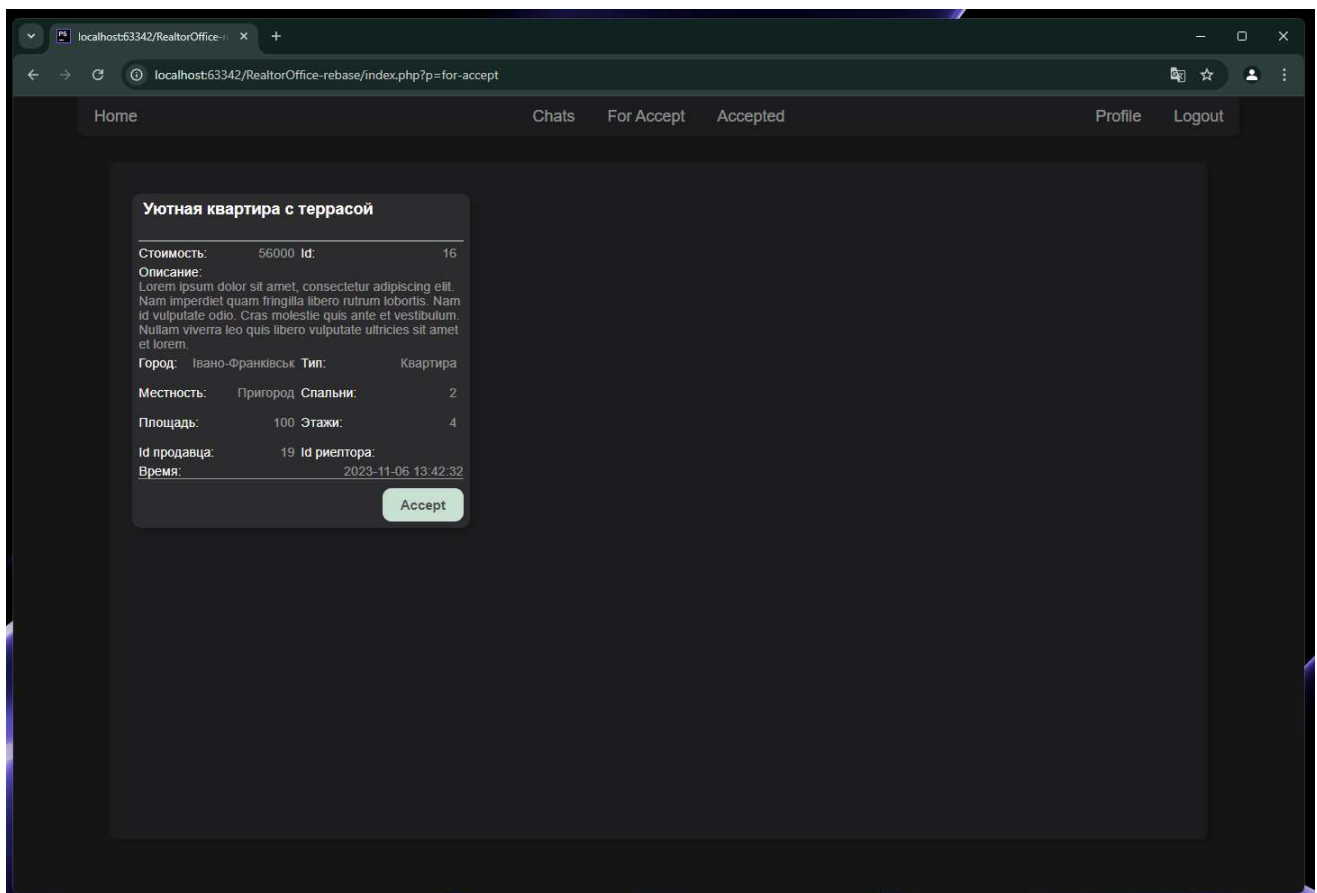


Рисунок 3.12 - Сторінка зі списком нерухомостей на розгляд.

На цій сторінці (рисунок 3.13) відображається список нерухомості, яка вже була затверджена ріелтором та виставлена на продаж. Кожен запис містить детальну інформацію про об'єкт: адресу, опис, вартість, статус продажу та інші характеристики.

Список нерухомості формується залежно від ріелтора, який авторизований у системі. Ріелтор бачить лише ті об'єкти, які знаходяться у його відповідальності.

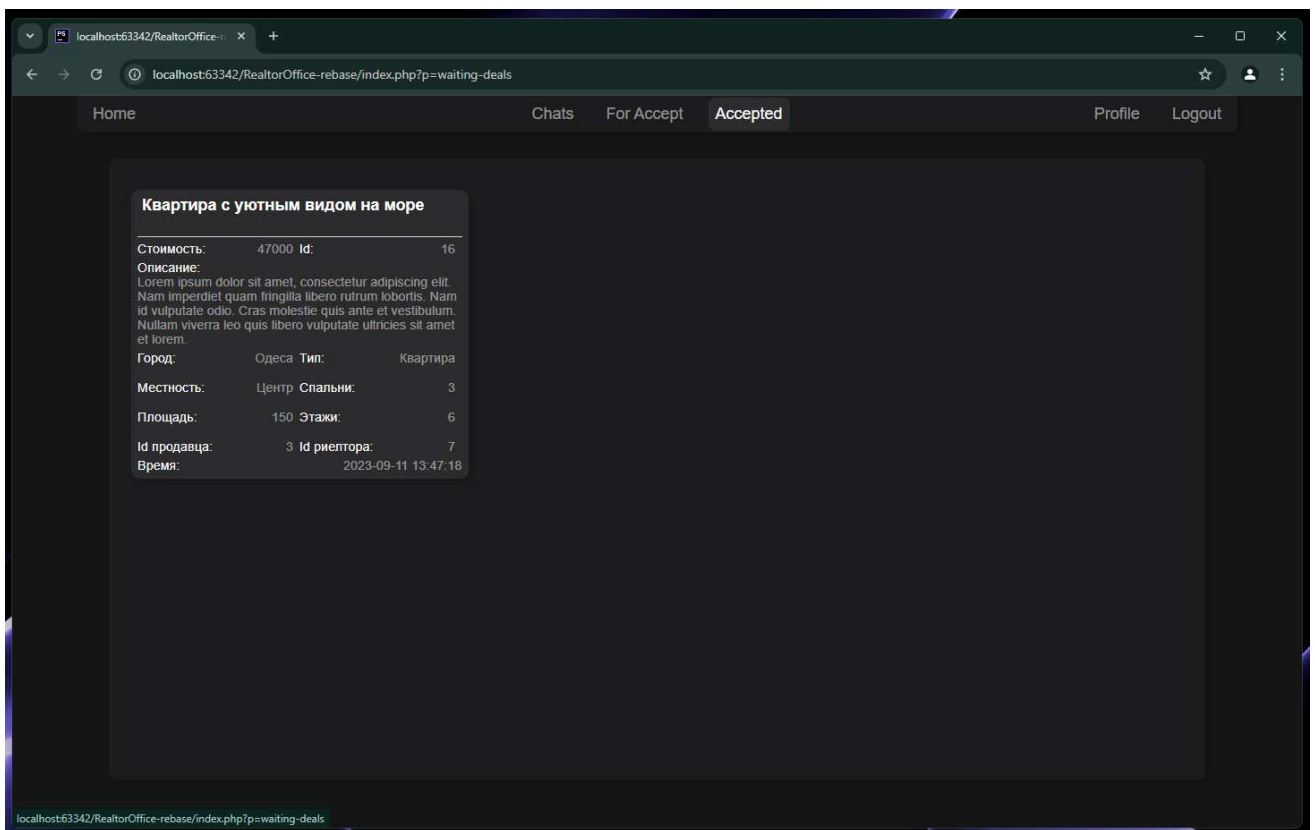


Рисунок 3.13 - Сторінка зі списком нерухомості що були затверджені.

На сторінці з логами (рисунок 3.14) відображується таблиця, що містить наступні стовбці: ID, тип запису, посилання, ID чату, ID користувача, опис, Сумма транзакції та дата. Дані для цієї таблиці беруться з бази даних.

ID	Type	Link	Chat ID	User ID	Description	Amount	Time
1	transaction	?p=buy-estate&id=3	2	2	Пользователь 2 начал покупку недвижимости 3 стоимостью 83000.	83000	2022-03-10 19:15:21
2	transaction	?p=buy-estate&id=3	2	2	Баланс покупателя 2 изменен с 100000 на 17000	83000	2022-03-10 19:15:21
3	transaction	?p=buy-estate&id=3	2	7	Баланс риэлтора 7 изменен с 0 на 2075	2075	2022-03-10 19:15:21
4	transaction	?p=buy-estate&id=3	2	6	Баланс продавца 6 изменен с 0 на 80925	80925	2022-03-10 19:15:21
5	sale	?p=buy-estate&id=3	2	2	Недвижимость 3 была продана пользователем 6 пользователю 2.	83000	2022-03-10 19:15:21
6	transaction	?p=buy-estate&id=1	3	4	Пользователь 4 начал покупку недвижимости 1 стоимостью 45000.	45000	2022-03-14 16:08:27
7	transaction	?p=buy-estate&id=1	3	4	Баланс покупателя 4 изменен с 50000 на 5000	45000	2022-03-14 16:08:27
8	transaction	?p=buy-estate&id=1	3	7	Баланс риэлтора 7 изменен с 2075 на 3200	1125	2022-03-14 16:08:27
9	transaction	?p=buy-estate&id=1	3	3	Баланс продавца 3 изменен с 0 на 43875	43875	2022-03-14 16:08:27
10	sale	?p=buy-estate&id=1	3	4	Недвижимость 1 была продана пользователем 3 пользователю 4.	45000	2022-03-14 16:08:27
11	transaction	?p=buy-estate&id=4	5	5	Пользователь 5 начал покупку недвижимости 4 стоимостью 35000.	35000	2022-04-07 12:08:30
12	transaction	?p=buy-estate&id=4	5	5	Баланс покупателя 5 изменен с 40000 на 5000	35000	2022-04-07 12:08:30
13	transaction	?p=buy-estate&id=4	5	7	Баланс риэлтора 7 изменен с 3200 на 4075	875	2022-04-07 12:08:30
14	transaction	?p=buy-estate&id=4	5	9	Баланс продавца 9 изменен с 0 на 34125	34125	2022-04-07 12:08:30
15	sale	?p=buy-estate&id=4	5	5	Недвижимость 4 была продана пользователем 9 пользователю 5.	35000	2022-04-07 12:08:30
16	transaction	?p=buy-estate&id=5	6	12	Пользователь 12 начал покупку недвижимости 5 стоимостью 60000.	60000	2022-07-19 14:15:28
17	transaction	?p=buy-estate&id=5	6	12	Баланс покупателя 12 изменен с 65000 на 5000	60000	2022-07-19 14:15:28
18	transaction	?p=buy-estate&id=5	6	11	Баланс риэлтора 11 изменен с 0 на 1500	1500	2022-07-19 14:15:28
19	transaction	?p=buy-estate&id=5	6	14	Баланс продавца 14 изменен с 0 на 58500	58500	2022-07-19 14:15:28
20	sale	?p=buy-estate&id=5	6	12	Недвижимость 5 была продана пользователем 14 пользователю 12.	60000	2022-07-19 14:15:28
21	transaction	?p=buy-estate&id=2	7	13	Пользователь 13 начал покупку недвижимости 2 стоимостью 32000.	32000	2022-08-07 15:47:14
22	transaction	?p=buy-estate&id=2	7	13	Баланс покупателя 13 изменен с 35000 на 3000	32000	2022-08-07 15:47:14

Рисунок 3.14 - Сторінка з логами.

На сторінці зі списком ріелторів (рисунок 3.15) відображається таблиця з даними про всіх ріелторів, які ми отримуємо з бази даних у вигляді (ID користувача, ім'я ріелтора, відсоток, досвід, баланс, дата, дія). У стовпчику "дія" є кнопка "редагувати", яка перенаправляє на сторінку редагування даних ріелтора (рисунок 3.16).

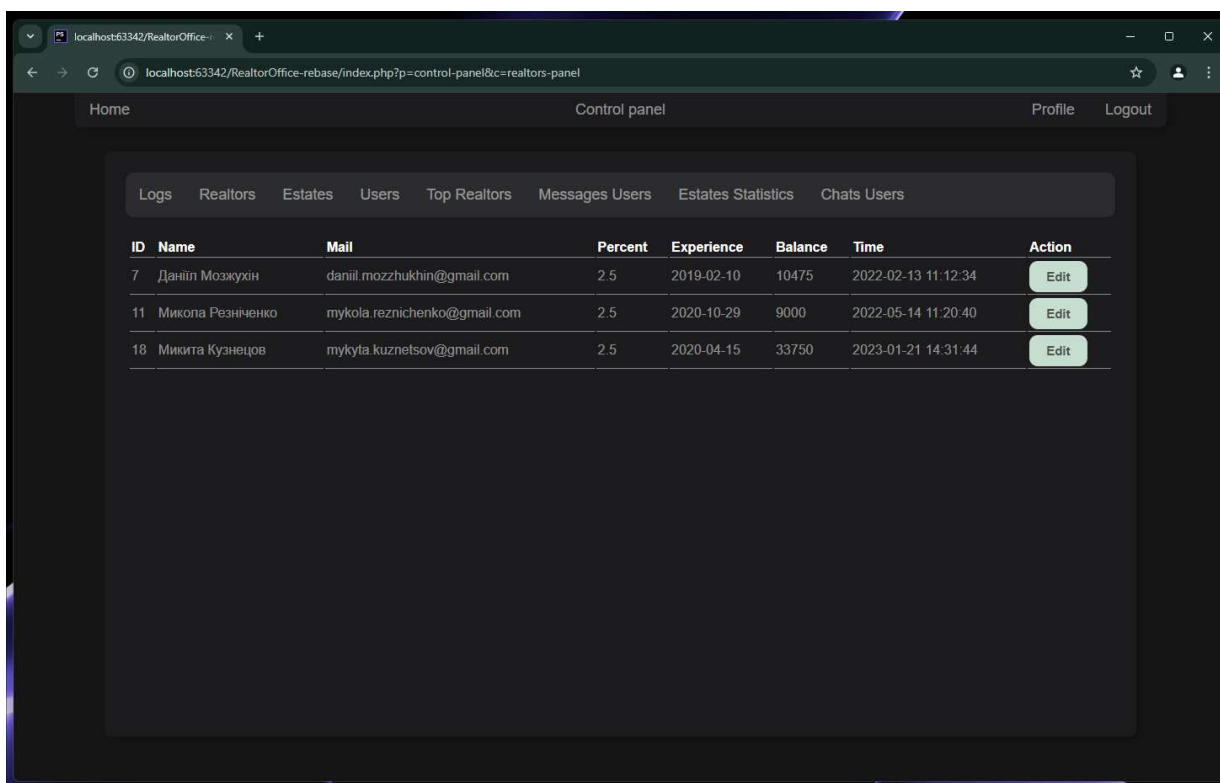


Рисунок 3.15 - Сторінка редагування ріелторів.

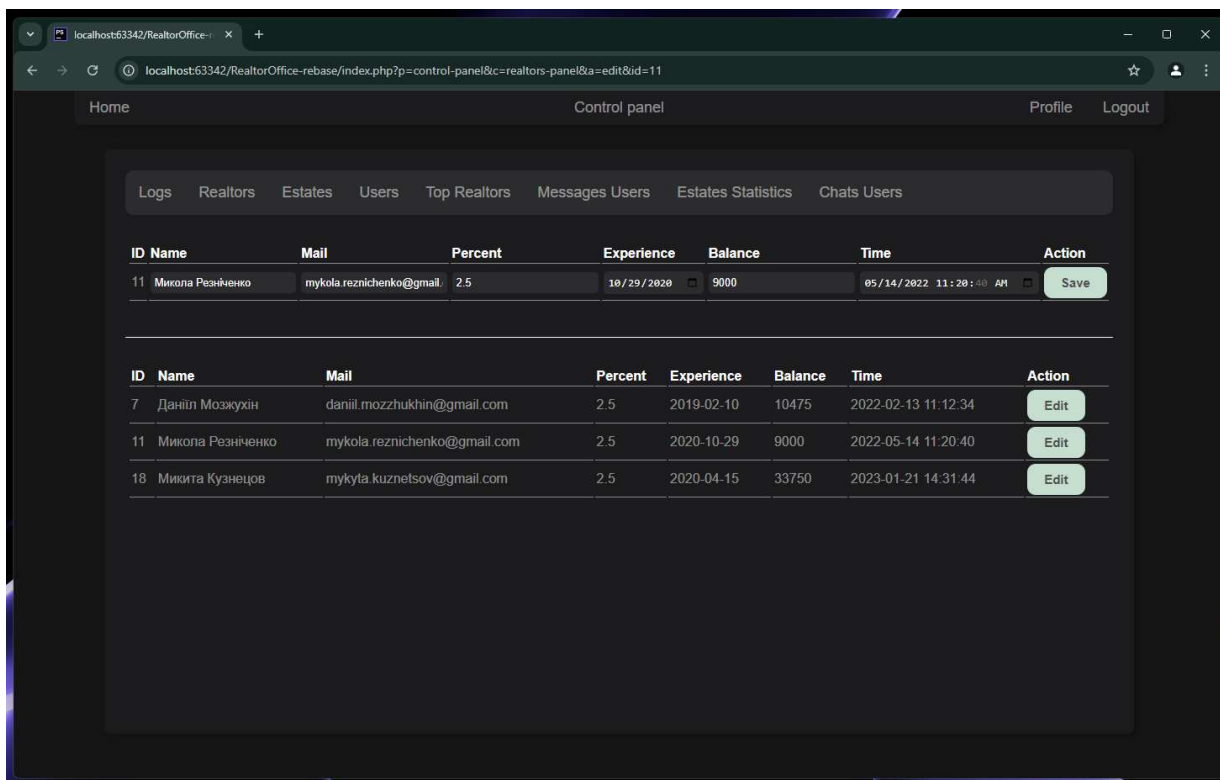


Рисунок 3.16 - Сторінка редагування ріелтора.

На цій сторінці (рисунок 3.17) відображається Control Panel, доступ до якої має лише адміністратор системи. У панелі управління знаходиться список усіх нерухомостей, незалежно від їх статусу: затверджені, на продажу, продані або відхилені. Кожен запис містить повну інформацію про об'єкт: адресу, опис, вартість, статус, відповідального ріелтора та інші характеристики.

Адміністратор може переглядати всі об'єкти, здійснювати пошук та сортування нерухомості, а також виконувати дії з керування, такі як редагування чи видалення записів.

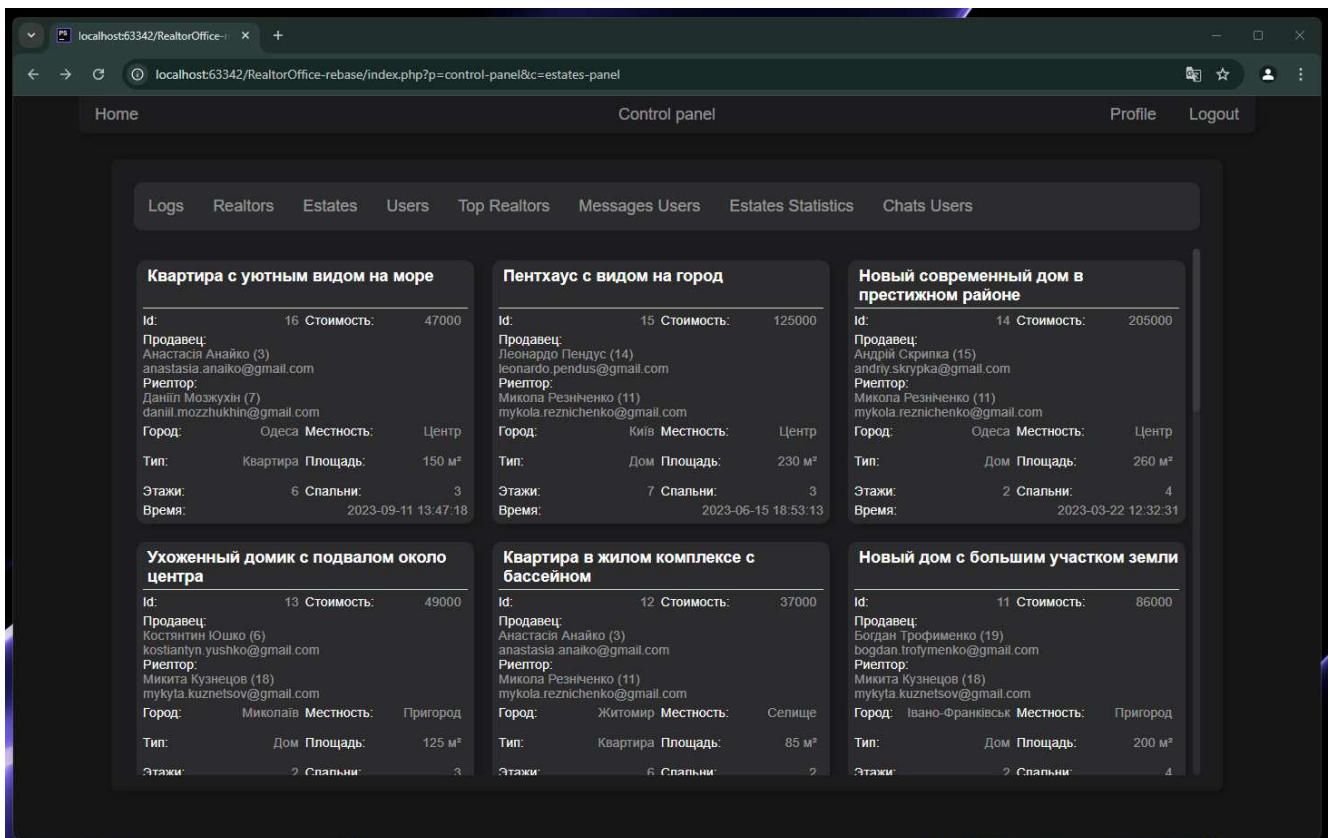
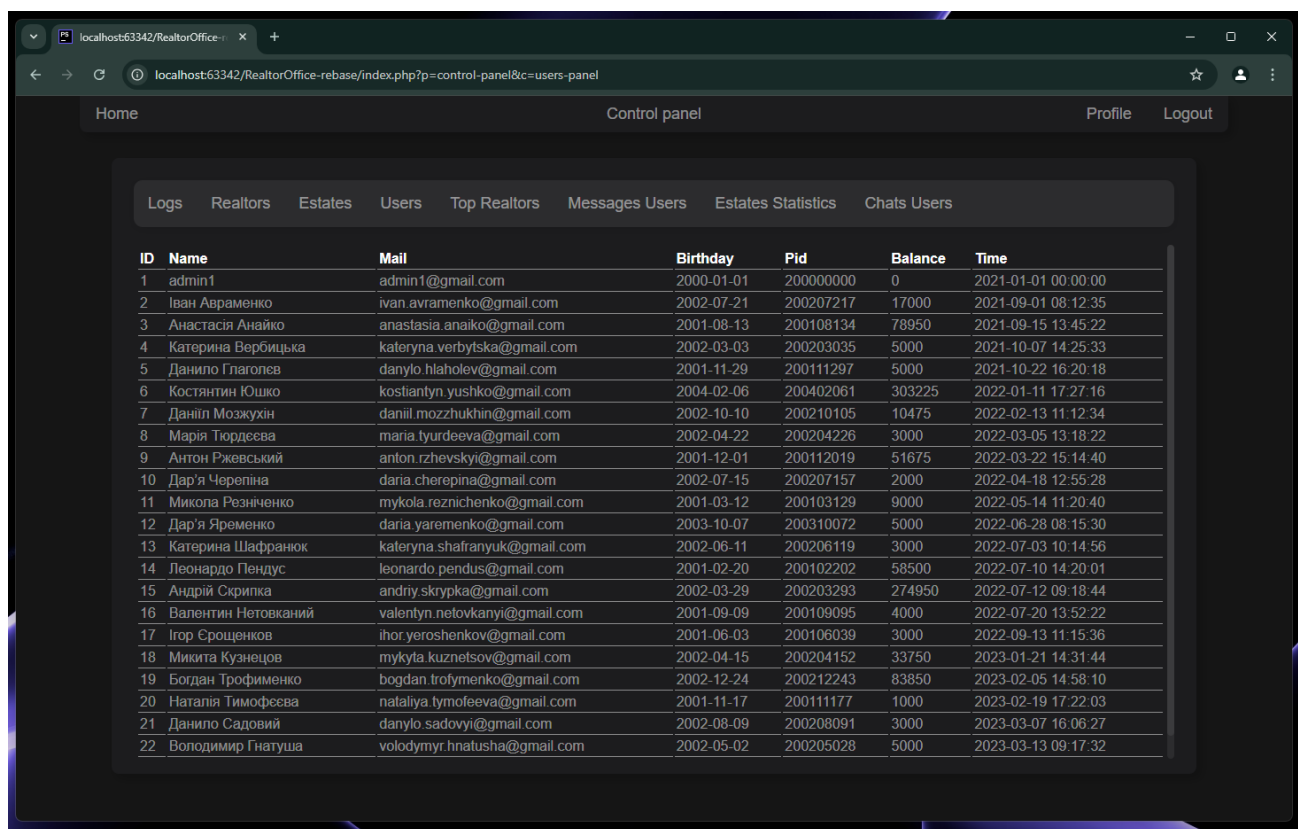


Рисунок 3.17 - Сторінка з усіма нерухомостями.

На цій сторінці (рисунок 3.18) відображається Control Panel, доступ до якої має лише адміністратор системи.

У панелі управління знаходиться список усіх користувачів системи, включаючи звичайних користувачів, ріелторів та адміністраторів. Кожен запис містить інформацію про користувача: ім'я, електронну пошту, роль у системі, баланс рахунку та інші деталі.

Адміністратор може переглядати всю базу користувачів, здійснювати пошук та сортування, а також виконувати дії з керування, такі як редагування даних, зміна ролі або видалення користувача з системи.



ID	Name	Mail	Birthday	Pid	Balance	Time
1	admin1	admin1@gmail.com	2000-01-01	200000000	0	2021-01-01 00:00:00
2	Іван Авраменко	ivan.avramenko@gmail.com	2002-07-21	200207217	17000	2021-09-01 08:12:35
3	Анастасія Анайко	anastasia.anaiko@gmail.com	2001-08-13	200108134	78950	2021-09-15 13:45:22
4	Катерина Вербицька	kateryna.verbyska@gmail.com	2002-03-03	200203035	5000	2021-10-07 14:25:33
5	Данило Глаголев	danylo.hlaholev@gmail.com	2001-11-29	200111297	5000	2021-10-22 16:20:18
6	Костянтин Юшко	kostiantyn.yushko@gmail.com	2004-02-06	200402061	303225	2022-01-11 17:27:16
7	Данііл Мозжухін	daniil.mozzhukhin@gmail.com	2002-10-10	200210105	10475	2022-02-13 11:12:34
8	Марія Тюрдеева	maria.tyurdeeva@gmail.com	2002-04-22	200204226	3000	2022-03-05 13:18:22
9	Антон Ржевський	anton.rzhevskiy@gmail.com	2001-12-01	200112019	51675	2022-03-22 15:14:40
10	Дар'я Черепіна	daria.cheropina@gmail.com	2002-07-15	200207157	2000	2022-04-18 12:55:28
11	Микола Резніченко	mykola.reznichenko@gmail.com	2001-03-12	200103129	9000	2022-05-14 11:20:40
12	Дар'я Яременко	daria.yaremenko@gmail.com	2003-10-07	200310072	5000	2022-06-28 08:15:30
13	Катерина Шафранок	kateryna.shafanyuk@gmail.com	2002-06-11	200206119	3000	2022-07-03 10:14:56
14	Леонардо Пендус	leonardo.pendus@gmail.com	2001-02-20	200102202	58500	2022-07-10 14:20:01
15	Андрій Скрипка	andriy.skrypka@gmail.com	2002-03-29	200203293	274950	2022-07-12 09:18:44
16	Валентин Нетовканий	valentyn.netovkanyi@gmail.com	2001-09-09	200109095	4000	2022-07-20 13:52:22
17	Ігор Єрошенко	igor.yeroshenko@gmail.com	2001-06-03	200106039	3000	2022-09-13 11:15:36
18	Микита Кузнецов	mykyta.kuznetsov@gmail.com	2002-04-15	200204152	33750	2023-01-21 14:31:44
19	Богдан Трофименко	bogdan.trofymenko@gmail.com	2002-12-24	200212243	83850	2023-02-05 14:58:10
20	Наталія Тимофеева	nataliya.tymofeeva@gmail.com	2001-11-17	200111177	1000	2023-02-19 17:22:03
21	Данило Садовий	danylo.sadovyi@gmail.com	2002-08-09	200208091	3000	2023-03-07 16:06:27
22	Володимир Гнатуша	volodymyr.hnatusha@gmail.com	2002-05-02	200205028	5000	2023-03-13 09:17:32

Рисунок 3.18 - Сторінка з усіма користувачами.

На сторінці (рисунок 3.19) відображається функціонал для фільтрації найбільш успішних ріелторів за продажами за конкретний рік. У верхній частині сторінки розташовано поле для введення року, після чого відображається список ріелторів, які досягли найвищих результатів у вибраний період. Для кожного ріелтора показуються ім'я, кількість проданих об'єктів та інші релевантні деталі.

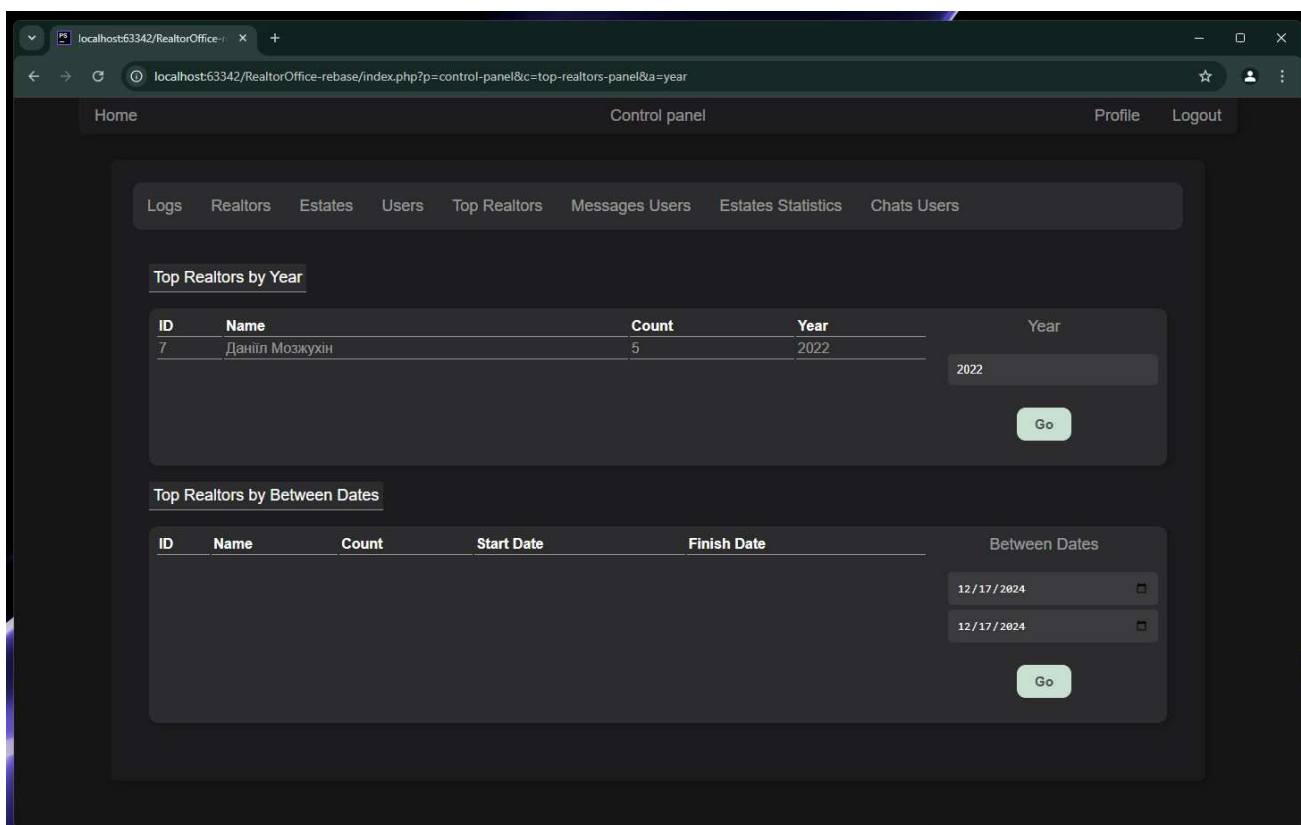


Рисунок 3.19 - Сторінка з фільтрацією продажів за вказаний рік.

На сторінці (рисунок 3.20) представлений той самий функціонал, але з можливістю фільтрації за періодом між двома зазначеними датами. У верхній частині розташовані поля для введення початкової та кінцевої дати. Після застосування фільтра відображається список ріелторів, які були найбільш активними у вказаний період.

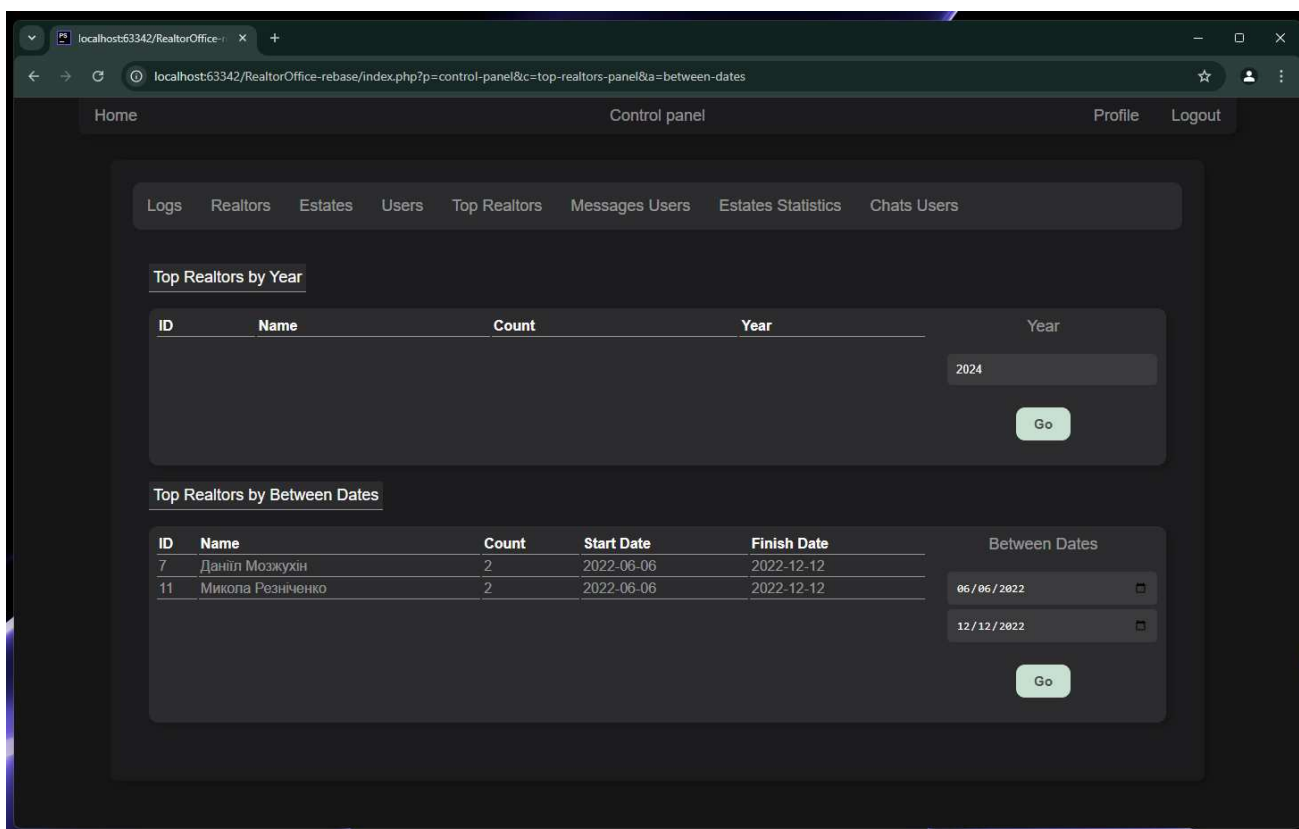


Рисунок 3.20 - Сторінка з фільтрацією продажів за період між датами.

На сторінці (рисунок 3.21) відображаються всі повідомлення користувача за обраний місяць. У верхній частині сторінки розташоване поле для вибору місяця, після чого відображається список повідомлень, які користувач надіслав або отримав у зазначений період.

Кожне повідомлення містить деталі, такі як дата, час, вміст повідомлення та учасники чату. Інтерфейс дозволяє швидко переглядати та аналізувати комунікацію за вказаний проміжок часу.

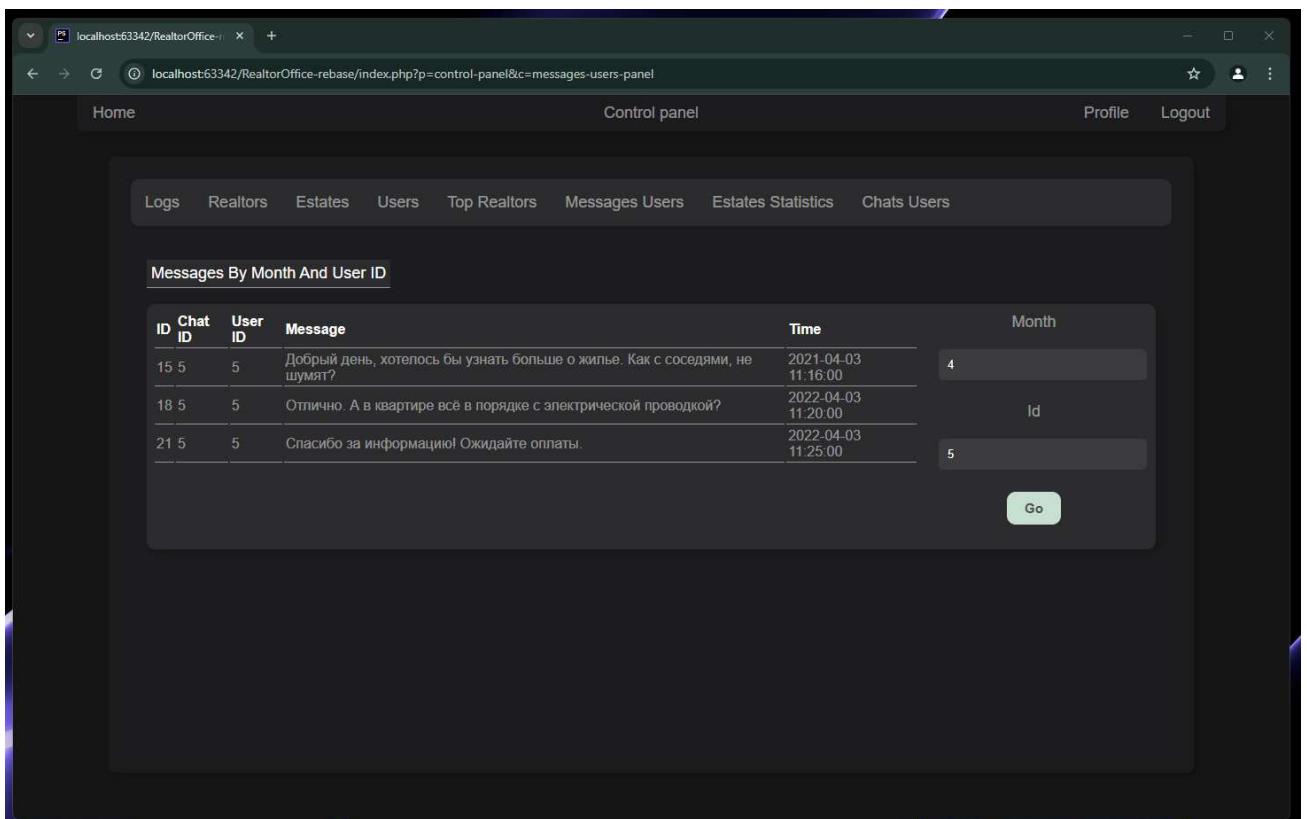


Рисунок 3.21 - Сторінка з усіма повідомленнями користувача за вказаний місяць.

На сторінці (рисунок 3.22) відображається загальна сума всіх площ для вибраного типу нерухомості. У верхній частині сторінки розташоване поле для вибору типу нерухомості.

Після вибору типу застосунок показує дві секції:

1. Сума площ: загальна площа всіх об'єктів нерухомості вказаного типу.
2. Список найбільших об'єктів: перелік об'єктів нерухомості обраного типу, відсортованих за максимальною площею. Кожен об'єкт містить деталі, такі як адреса, площа, вартість та інші характеристики.

Ця сторінка допомагає аналізувати та порівнювати об'єкти певного типу за їх розмірами.

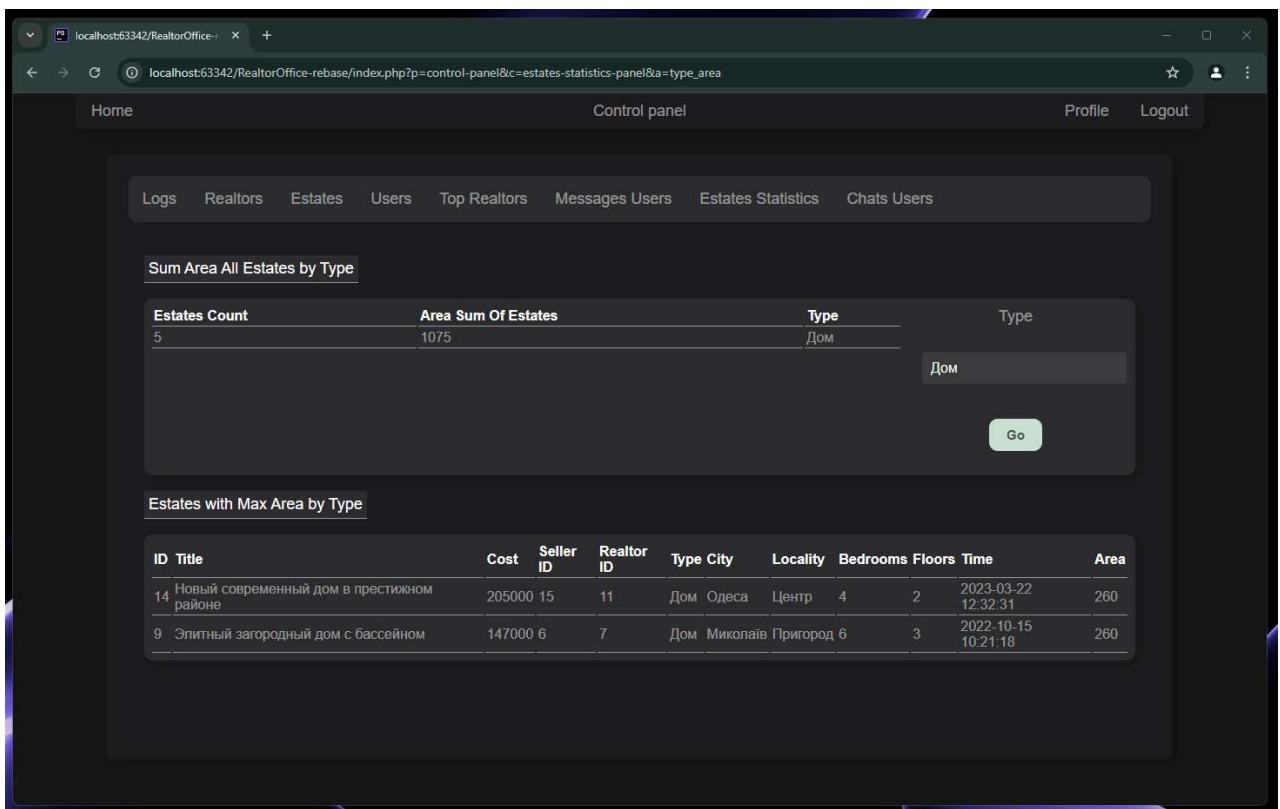


Рисунок 3.22 - Сторінка з пошуком суми та максимальної площі за типом нерухомості.

На сторінці (рисунок 3.23) відображається список чатів, який формується на основі вказаного ID користувача. Для кожного чату перевіряється тип користувача (ріелтор, покупець, продавець) і виводиться наступна інформація:

1. ID чату: унікальний ідентифікатор чату.
2. ID нерухомості: ідентифікатор об'єкта нерухомості.
3. ID продавця: ідентифікатор користувача, який є власником нерухомості.
4. ID ріелтора: ідентифікатор ріелтора, відповідального за об'єкт.
5. ID покупця ідентифікатор користувача, який зацікавлений у покупці.
6. Час створення чату: дата і час, коли чат був ініційований.
7. Час покупки: дата і час, коли покупка була завершена (якщо застосовно).

Сторінка надає зручний інтерфейс для перегляду всієї інформації, пов'язаної з чатами, з урахуванням типу користувача.

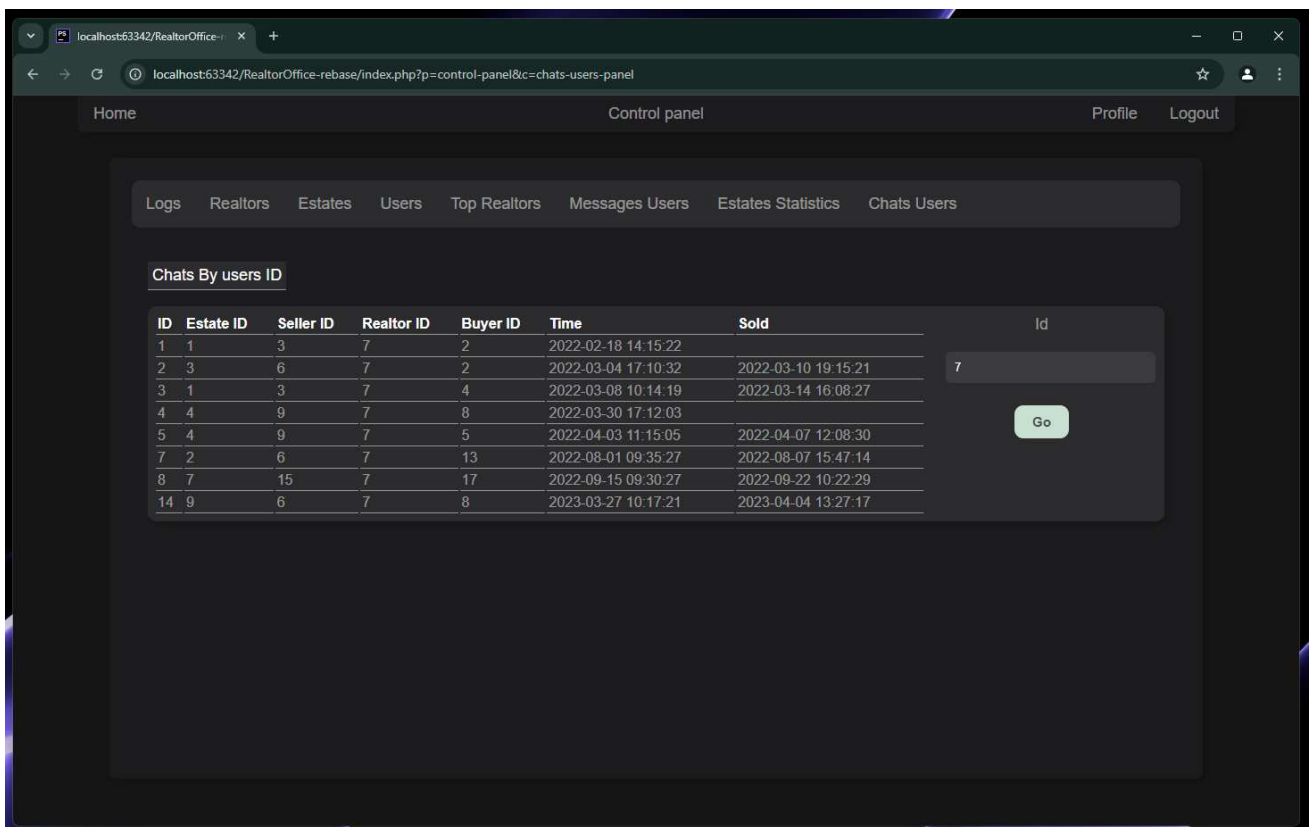


Рисунок 3.23 - Сторінка з пошуком чату для конкретного користувача.

У розділі 3 в результаті реалізації інформаційної системи веб-маркетплейсу нерухомості створено функціональне, безпечне та зручне середовище для взаємодії користувачів, ріелторів і адміністраторів. Система включає взаємопов'язані модулі, що охоплюють увесь цикл операцій з нерухомістю — від розміщення оголошень до укладення угод і фінансових розрахунків. Інтерфейс забезпечує коректну взаємодію з користувачем завдяки валідації форм. Забезпечено належний рівень безпеки шляхом контролю доступу та захисту транзакцій. SQL-запити відображають ключові механізми взаємодії з базою даних: реєстрацію, авторизацію, управління оголошеннями, діалоги, транзакції та адміністрування. Розроблена система відповідає вимогам сучасного веб-сервісу та має потенціал до масштабування.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ

Розробка та впровадження інформаційної системи веб-маркетплейсу для купівлі та продажу нерухомості передбачає участь персоналу, що працює в офісному середовищі за комп'ютерами - адміністраторів, ріелторів, модераторів, операторів підтримки та інших фахівців. При організації їхніх робочих місць та під час виконання службових обов'язків необхідно дотримуватись вимог чинного законодавства з охорони праці.

Основні положення щодо безпечних умов праці встановлюються відповідно до Закону України «Про охорону праці», Державних санітарних правил і норм ДСанПіН 3.3.2.007-98, ДСТУ EN ISO 9241-5:2002, Правил пожежної безпеки в Україні (НАПБ А.01.001-2004), а також наказу Міністерства соціальної політики України № 207 від 14.02.2018 «Про затвердження Вимог безпеки та захисту здоров'я працівників при роботі з екранними пристроями».

Робоче місце працівника повинно відповідати ергономічним стандартам: висота столу 720-750 мм, крісло повинно мати регулювання висоти та кута нахилу спинки, відстань від очей до монітора - 60-70 см, з кутом огляду 15-20° нижче горизонтальної лінії зору. Освітленість має становити не менше 300 люкс при природному освітленні та не менше 500 люкс при штучному. Слід уникати відблисків на екрані, бажано використовувати фільтри синього світла та автоматичну яскравість моніторів.

З метою зниження навантаження на зір та опорно-руховий апарат рекомендується організовувати перерви тривалістю 10-15 хвилин після кожної години роботи з комп'ютером. У цей час доцільно виконувати вправи для очей, шиї та спини.

Особливу увагу слід приділяти електробезпеці. Усе обладнання повинно бути справним, сертифікованим і мати захисне заземлення. Не допускається

експлуатація техніки без нагляду або при наявності пошкоджень у мережах і кабелях. Необхідно уникати перевантаження електричних мереж, а всі пристрої мають вмикатися через справні подовжувачі з захистом.

У приміщенні повинні бути вогнегасник, затверджена схема евакуації, інструкції з пожежної безпеки та номери екстрених служб. Роботодавець зобов'язаний проводити первинні та періодичні інструктажі з техніки безпеки, а також організовувати обов'язкові медичні огляди відповідно до наказу МОЗ № 246 від 21.05.2007.

Важливо також враховувати психоемоційне навантаження, пов'язане з роботою з великою кількістю інформації та спілкуванням із клієнтами. Слід уникати понаднормової роботи без письмової згоди працівника, забезпечити сприятливий психологічний клімат, організовувати тренінги з управління стресом.

Робоча зміна не повинна перевищувати 8 годин, при цьому має бути обідня перерва не менше 30 хвилин та можливість коротких перерв упродовж дня. Робоче місце слід організовувати так, щоб працівник міг змінювати положення тіла й не зазнавав надмірного фізичного навантаження. Рівень шуму повинен відповідати нормам, встановленим ДСН 3.3.6.037-99.

У разі необхідності роботодавець повинен ініціювати лабораторні дослідження умов праці, проводити оцінку ризиків і реалізовувати заходи для їх усунення. Дотримання вимог охорони праці сприяє збереженню здоров'я працівників, підвищенню продуктивності праці та стабільному функціонуванню інформаційної системи.

У розділі 4 детально описуються нормативно-правові та організаційні вимоги щодо забезпечення безпечних та комфортних умов праці персоналу веб-маркетплейсу для купівлі та продажу нерухомості. Встановлюються вимоги, які

регламентують відповідність робочих місць ергономічним стандартам, зокрема параметри меблів, розміщення моніторів, освітлення, а також організація режиму роботи з урахуванням перерв для зниження фізичного та психоемоційного навантаження. Розглядаються питання електробезпеки, пожежної безпеки, проведення інструктажів і медичних оглядів працівників. Окремо виділяються рекомендації щодо створення сприятливого психологічного клімату на робочому місці. Також визначається необхідність проведення лабораторних досліджень умов праці, оцінки ризиків та реалізації заходів щодо їх усунення. Виконання цих вимог сприяє збереженню здоров'я працівників, підвищенню продуктивності праці та забезпеченню стабільного функціонування інформаційної системи.

ВИСНОВОК

У кваліфікаційній роботі було розроблено інформаційну систему у вигляді веб-маркетплейсу для автоматизації процесів купівлі, продажу та оренди нерухомості. Проведено аналіз предметної області, визначено ключові функціональні вимоги до системи та спроектовано архітектуру, яка враховує потреби користувачів, ріелторів і адміністраторів.

Розроблена система забезпечує зручну взаємодію між учасниками ринку, дозволяє ефективно управляти оголошеннями про нерухомість, здійснювати фінансові розрахунки та вести комунікацію між сторонами. Упроваджено механізми контролю доступу, захисту даних та валідації, що гарантує безпеку і надійність функціонування.

У результаті реалізовано функціональний веб-сервіс, який відповідає сучасним вимогам до цифрових платформ. Система може бути використана в реальних умовах ринку для оптимізації діяльності ріелторів та підвищення якості обслуговування клієнтів.

Отримані результати свідчать про ефективність обраного підходу до розробки. Система має потенціал до масштабування, подальшого вдосконалення та інтеграції з зовнішніми сервісами, що відкриває перспективи її комерційного застосування та розширення функціональності відповідно до потреб ринку нерухомості.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ажищев В. Правила оформлення кваліфікаційної (магістерської) роботи - (дата звернення 13.03.2025).
2. DIM.RIA - <https://dom.ria.com/uk/about-us> - (дата звернення: 13.03.2025).
3. REM.UA - <https://rem.ua/ua> - (дата звернення 12.04.2025).
4. OLX - <https://www.olx.ua/uk> - (дата звернення 12.04.2025).
5. Сучасна ріелторська культура в Україні: як змінюється ринок - <https://realtyschool.com.ua/suchasna-riyeltorska-kultura> - (дата звернення 22.04.2025)
6. Марченко А.В. Проектування інформаційних систем, Київ, 2016 - https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content-20160217112601.pdf - (дата звернення 06.05.2025)
7. MySQL - <https://uk.wikipedia.org/wiki/MySQL> - (дата звернення: 15.05.2025)
8. PHP - <https://uk.wikipedia.org/wiki/PHP> - (дата звернення: 15.05.2025)
9. Smarty - <https://uk.wikipedia.org/wiki/Smarty> - (дата звернення: 15.05.2025)
10. HTML - <https://uk.wikipedia.org/wiki/HTML> - (дата звернення: 15.05.2025)
11. Git - <https://uk.wikipedia.org/wiki/Git> - (дата звернення: 15.05.2025)
12. PHPStorm - <https://uk.wikipedia.org/wiki/PhpStorm> - (дата звернення 15.05.2025)
13. Apache HTTP Server - https://uk.wikipedia.org/wiki/Apache_HTTP_Server - (дата звернення 15.05.2025).
14. Amazon S3 - <https://aws.amazon.com/s3/> - (дата звернення 15.05.2025).
15. Які документи потрібні для продажу квартири в Україні (2025 рік) - <https://mayak.kiev.ua/news/dokumenty-dlya-prodazhu-kvartyry> - (дата звернення 21.05.2025)

- 16.Купівля та продаж нерухомості у 2025, нові податки -
<https://pravdop.com/ua/publications/novosti-kompaniy/izmeneniya-v-nalogooblozhenii-nedvizhimosti-v-2025-godu-02-2025-158> - (дата звернення 21.05.2025)
- 17.Вендров А. М. Проектування програмного забезпечення економічних інформа-ційних систем. М.: Фінанси та статистика, 2006. - 544 с. - (дата звернення: 30.03.2025).
- 18.Грекул В. І. Проектування інформаційних систем. Практикум. М: Національний відкритий університет «ІНТУІТ», 2012. -159 с. - (дата звернення: 30.03.2025).
- 19.Ременяк Л.В. Проектування інформаційних систем: конспект лекцій. Одеса, Одеський державний екологічний університет, 2016. 152. - (дата звернення: 30.03.2025).
- 20.UML [Електронний ресурс]: Вікіпедія. Вільна енциклопедія. -
<https://uk.wikipedia.org/wiki/UML> - (дата звернення: 07.03.2025).
- 21.М. Фаулер UML. Основи, 3-видання; [В оригіналі] - (дата звернення: 07.03.2025).

ДОДАТОК - ТЕКСТ ПРОГРАМИ

Усі розробки та файли проекту можна знайти за посиланням:

<https://github.com/shizvslvt/RealtorOffice-rebase>

Auth.php

```
<?php

class Auth
{
    public function login($mail, $password): bool
    {
        global $db_controller;
        $user = $db_controller->login($mail, $password);
        if ($db_controller->login($mail, $password)) {
            setcookie("uid", $user, time() + 3600);
            return true;
        }
        return false;
    }
    public function logout()
    {
        setcookie("uid", null, time() - 3600);
        header('Location: ?p=home');
        exit();
    }
    public function register($name, $mail, $password, $birthday, $PID): bool
    {
        global $db_controller;
        return $db_controller->register($name, $mail, $password, $birthday,
$PID);
    }
    public function registerRealtor($name, $mail, $password, $birthday, $PID,
$experience): bool
    {
        global $db_controller;
        return $db_controller->registerRealtor($name, $mail, $password,
$birthday, $PID, $experience);
    }
}
```

Клас Auth:

- login(\$mail, \$password): bool

Виконує пошук користувача в базі даних за переданими електронною поштою та паролем. Якщо користувача знайдено, встановлює cookie для ідентифікації користувача і повертає true. Якщо аутентифікація не вдалася, повертає false.

- logout()

Виходить з облікового запису, видаляючи cookie і перенаправляє користувача на головну сторінку.

- register(\$name, \$mail, \$password, \$birthday, \$PID): bool

Реєструє нового користувача, перевіряючи наявність дублюючих електронних адрес або ідентифікаційних номерів у базі даних. Якщо реєстрація успішна, повертає true. Якщо виявлено дублікати, повертає false.

- registerRealtor(\$name, \$mail, \$password, \$birthday, \$PID, \$experience): bool

Реєструє нового ріелтора, додаючи його до бази даних користувачів та додаючи дані ріелтора до таблиці ro_realtors. Визначає процент комісії ріелтора на основі його досвіду та додає цей запис до бази. Якщо реєстрація успішна, повертає true, інакше false.

Chats.php

```
<?php
```

```
class Chats
{
    public function displayChats($chat_id = null)
    {
        $c = $_GET['c'] ?? 'buy';
        global $db_controller, $access_level;
        $uid = $_COOKIE['uid'];
        if ($access_level == 2) $c = 'sell';
        $chats = $db_controller->getChatsByUID($uid, $access_level, $c);

        global $theme;

        $theme->assign('c', $c);
        $theme->assign('accessLevel', $access_level);

        $theme->assign('chats', $chats);

        $theme->display('chats.tpl');

        if (isset($chat_id)) {
            global $db_controller;
            return $db_controller->getEstateIdByChatId($chat_id);
        }
    }
}
```



```

public function displayMessages($id)
{
    global $theme, $message;
    $theme->assign('chat_id', $id);
    $messages = $message->getMessages($id);
    $theme->assign('uid', $_COOKIE['uid']);
    $theme->assign('c', $_GET['c']);
    $theme->assign('chat_id', $id);
    $theme->assign('messages', $messages);
    $theme->display('messages.tpl');
}

public function sendMessage($uid, $chat_id, $message): void
{
    global $db_controller;
    $db_controller->sendMessage($uid, $chat_id, $message);
}

public function createChat($id, $uid)
{
    global $db_controller;
    return $db_controller->createChat($id, $uid);
}

public function getChatsByUserId($user_id) {
    global $db_controller;
    return $db_controller->getChatsByUserId($user_id);
}
}

```

Клас Chats:

- displayChats(\$chat_id = null)

Відображає список чатів, залежно від ролі користувача (покупець або продавець). Якщо заданий параметр chat_id, повертає ідентифікатор нерухомості, що пов'язана з чатом. Чати отримуються через виклик методу getChatsByUID з класу db_controller.

- displayMessages(\$id)

Відображає повідомлення чату з ідентифікатором \$id. Отримує повідомлення через метод getMessages класу message. Відображає повідомлення на сторінці за допомогою шаблону messages.tpl.

- sendMessage(\$uid, \$chat_id, \$message)

Відправляє повідомлення в чат з заданим ідентифікатором \$chat_id. Використовує метод sendMessage класу db_controller для запису повідомлення в базу даних.

- createChat(\$id, \$uid)

Створює новий чат між покупцем та продавцем, де \$id - це ідентифікатор нерухомості, а \$uid - ідентифікатор покупця. Викликає метод createChat класу db_controller для додавання чату в базу даних.

- getChatsByUserId(\$user_id)

Отримує список чатів для користувача за його ідентифікатором \$user_id. Чати фільтруються за участю користувача (покупець, продавець або ріелтор) в чаті чи нерухомості. Використовує метод getChatsByUserId класу db_controller для отримання даних з бази.

Estate.php

```
<?php
```

```
class Estate
{
    public function displayHomePageEstates(): void
    {
        global $theme, $db_controller;
        $estates = $db_controller->getHomePageEstates();
        echo '<div class="estates-grid">';
        foreach ($estates as $estate) {
            $theme->assign("status", "home");
            $theme->assign('estate', $estate);
            $theme->display('estate.tpl');
        }
        echo '</div>';
    }

    public function getEstateById($estate_id)
    {
        global $db_controller;
        return $db_controller->getEstateById($estate_id);
    }

    public function displayEstate($estate_id): void
    {
        global $theme, $db_controller, $access_level;
```

```

        $estate = $this->getEstateById($estate_id);
        if (!$estate) die("Estate not found");
        $status = 'none';
        if ($access_level == 1) {
            $chat_id = $db_controller->getChatIdByEstateIdAndUID($estate_id,
$_COOKIE['uid']);
            $theme->assign('chat_id', $chat_id);
            $status = 'by-id';
            if($db_controller->CompareSellerIdByEstateIdAndUID($estate_id,
$_COOKIE['uid'])) $status = 'edit';
        } elseif ($access_level == 2) {
            $status = 'realtor';
        }
        if ($db_controller->checkSoldEstate($estate_id)) $status = 'sold';

        $theme->assign('estate', $estate);
        $theme->assign('status', $status);
        echo '<div class="estates-grid-single">';
        $theme->display('estate.tpl');
        echo '</div>';
    }

    public function displayEstatesForAccept(): void
    {
        global $theme, $db_controller;
        $estates = $db_controller->getEstatesForAccept();
        echo '<div class="estates-grid">';
        foreach ($estates as $estate) {
            $theme->assign("status", "for-accept");
            $theme->assign('estate', $estate);
            $theme->display('estate.tpl');
        }
        echo '</div>';
    }

    public function acceptEstate($id, $realtor_id): void
    {
        global $db_controller;
        $db_controller->acceptEstate($id, $realtor_id);
    }

    public function displayRealtorWaitingDealsEstates($realtor_id): void
    {
        global $theme, $db_controller;
        $estates = $db_controller->
>displayRealtorWaitingDealsEstates($realtor_id);
        echo '<div class="estates-grid">';
        foreach ($estates as $estate) {
            $theme->assign("status", "waiting-deals");
            $theme->assign('estate', $estate);
            $theme->display('estate.tpl');
        }
        echo '</div>';
    }

    public function displayEstateForChat($estate_id): void
    {

```

```

        echo '<div class="estate-container">';
        echo '<div class="estate-container-title">Estate</div>';
        global $theme, $db_controller;
        $estate = $db_controller->getEstateForChat($estate_id);
        $theme->assign("status", "buy");
        $theme->assign('buyer', $db_controller-
>getBuyerIdByEstateIdAndUID($estate_id, $_COOKIE['uid']));
        $theme->assign('estate', $estate);
        $check_sold = $db_controller->checkSoldEstate($estate_id);
        $theme->assign('checksold', $check_sold);
        $theme->display('estate.tpl');
        echo '</div>';
        echo '</div>';
    }

    public function buyEstate($estate_id)
    {
        global $db_controller;
        $db_controller->buyEstate($estate_id);
    }

    public function getEstateCost($id)
    {
        global $db_controller;
        return $db_controller->getEstateCost($id);
    }

    public function checkSoldEstate($id)
    {
        global $db_controller;
        return $db_controller->checkSoldEstate($id);
    }

    public function checkEstateAccessLevelForBuyer($id)
    {
        global $db_controller;
        return $db_controller->checkEstateAccessLevelForBuyer($id);
    }

    public function selectEstates()
    {
        global $db_controller;
        return $db_controller->selectEstates();
    }

    public function getData($string)
    {
        global $db_controller;
        return $db_controller->getData($string);
    }

    public function addEstate($seller_id, $title, $description, $cost, $type,
$city, $locality, $area, $bedrooms, $floors, $created)
    {
        global $db_controller;
        return $db_controller->addEstate($seller_id, $title, $description,
$cost, $type, $city, $locality, $area, $bedrooms, $floors, $created);
    }

```

```

public function selectEstatesByType(mixed $type)
{
    global $db_controller;
    return $db_controller->selectEstatesByType($type);
}

public function selectEstatesByTypeWithMaxArea(mixed $type)
{
    global $db_controller;
    return $db_controller->selectEstatesByTypeWithMaxArea($type);
}

public function editEstate($id, $title, $cost, $archived, $description,
$city, $locality, $type, $area, $bedrooms, $floors)
{
    global $db_controller;
    $db_controller->editEstate($id, $title, $cost, $archived, $description,
$city, $locality, $type, $area, $bedrooms, $floors);
}

public function checkEstateAccessLevelForSeller($estate_id, $uid)
{
    global $db_controller;
    return $db_controller->checkEstateAccessLevelForSeller($estate_id,
$uid);
}
}

```

Клас Estate:

- displayHomePageEstates()

Відображає список нерухомості на головній сторінці. Отримує дані через метод getHomePageEstates класу db_controller і відображає кожен елемент через шаблон estate.tpl.

- getEstateById(\$estate_id)

Повертає інформацію про нерухомість за її ідентифікатором \$estate_id через метод getEstateById класу db_controller.

- displayEstate(\$estate_id)

Відображає детальну інформацію про нерухомість. В залежності від ролі користувача (покупець чи ріелтор) встановлює статус та перевіряє чи є чати або можливість редагувати нерухомість.

- displayEstatesForAccept()

Відображає список нерухомості, що чекає на підтвердження через метод getEstatesForAccept класу db_controller.

- acceptEstate(\$id, \$realtor_id)

Приймає нерухомість для ріелтора через метод acceptEstate класу db_controller.

- displayRealtorWaitingDealsEstates(\$realtor_id)

Відображає список нерухомості, яка чекає на укладання угоди для ріелтора. Отримує дані через метод displayRealtorWaitingDealsEstates класу db_controller.

- displayEstateForChat(\$estate_id)

Відображає деталі нерухомості, яку можна обговорювати в чаті. Перевіряє, чи є угода (нерухомість продана) і передає необхідні дані в шаблон estate.tpl.

- buyEstate(\$estate_id)

Купує нерухомість через метод buyEstate класу db_controller.

- getEstateCost(\$id)

Повертає вартість нерухомості за її ідентифікатором через метод getEstateCost класу db_controller.

- checkSoldEstate(\$id)

Перевіряє, чи була продана нерухомість через метод checkSoldEstate класу db_controller.

- checkEstateAccessLevelForBuyer(\$id)

Перевіряє рівень доступу для покупця до нерухомості через метод `checkEstateAccessLevelForBuyer` класу `db_controller`.

- `selectEstates()`

Повертає список усіх нерухомостей через метод `selectEstates` класу `db_controller`.

- `getData($string)`

Отримує дані на основі рядка через метод `getData` класу `db_controller`.

- `addEstate($seller_id, $title, $description, $cost, $type, $city, $locality, $area, $bedrooms, $floors, $created)`

Додає нову нерухомість до бази даних через метод `addEstate` класу `db_controller`.

- `selectEstatesByType($type)`

Отримує список нерухомості за її типом через метод `selectEstatesByType` класу `db_controller`.

- `selectEstatesByTypeWithMaxArea($type)`

Отримує список нерухомості за її типом з найбільшою площею через метод `selectEstatesByTypeWithMaxArea` класу `db_controller`.

- `editEstate($id, $title, $cost, $archived, $description, $city, $locality, $type, $area, $bedrooms, $floors)`

Редагує дані про нерухомість через метод `editEstate` класу `db_controller`.

- `checkEstateAccessLevelForSeller($estate_id, $uid)`

Перевіряє рівень доступу для продавця до конкретної нерухомості через метод `checkEstateAccessLevelForSeller` класу `db_controller`.

Log.php

```
<?php
class Log
{
    public function addLog($type, $link = null, $chat_id = null, $user_id =
null, $desc = null, $amount = null)
    {
        global $db_controller;
        $db_controller->addLog($type, $link, $chat_id, $user_id, $desc,
$amount);
    }

    public function selectLogs()
    {
        global $db_controller;
        return $db_controller->selectLogs();
    }
}
```

Клас Log:

- `addLog($type, $link, $chat_id, $user_id, $desc, $amount)`

Виконує додавання запису до таблиці логів, включаючи тип події, посилання, чат, користувача, опис та суму. Запис додається з поточним часом.

- `selectLogs()`

Отримує всі записи з таблиці логів і повертає їх у вигляді масиву для подальшого аналізу або відображення.

Realtor.php

```
<?php
class Realtor
{
    public function getArrayRealtorsId()
    {
        global $db_controller;
        return $db_controller->getArrayRealtorsId();
    }

    public function selectTopRealtorsByYear($year = 2023)
```



```

    {
        global $db_controller;
        return $db_controller->selectTopRealtorsByYear($year);
    }

    public function selectTopRealtorsByBetweenDates($start_date, $finish_date)
    {
        global $db_controller;
        return $db_controller->selectTopRealtorsByBetweenDates($start_date,
$finish_date);
    }

    public function selectRealtor()
    {
        global $db_controller;
        return $db_controller->selectRealtor();
    }

    public function selectRealtorById($id)
    {
        global $db_controller;
        return $db_controller->selectRealtorById($id);
    }

    public function updateRealtor($id, $name, $mail, $percent, $experience,
$balance, $time)
    {
        global $db_controller;
        return $db_controller->updateRealtor($id, $name, $mail, $percent,
$experience, $balance, $time);
    }
}

```

Клас Realtor:

- getArrayRealtorsId()

Отримує список усіх ідентифікаторів ріелторів з таблиці ro_realtors. Метод виконує SQL-запит, щоб отримати всі ID ріелторів і повертає масив з ідентифікаторами.

- selectTopRealtorsByYear(\$year = 2023)

Отримує ріелторів, які зробили найбільше продажів протягом певного року. Метод використовує SQL-запит, який підраховує кількість продажів кожного

ріелтора та повертає ріелторів, у яких кількість продажів є максимальною за зазначений рік.

- `selectTopRealtorsByBetweenDates($start_date, $finish_date)`

Отримує ріелторів, які зробили найбільше продажів між двома зазначеними датами. SQL-запит підраховує кількість продажів для кожного ріелтора в заданому періоді і повертає тих, у кого найвищі показники.

- `selectRealtor()`

Отримує всіх ріелторів з бази даних разом з їхніми даними, такими як ім'я, електронна пошта, процент комісії, досвід, баланс та час. Повертає масив ріелторів.

- `selectRealtorById($id)`

Отримує дані ріелтора за його ідентифікатором. Метод виконує SQL-запит за заданим ID і повертає інформацію про ріелтора.

- `updateRealtor($id, $name, $mail, $percent, $experience, $balance, $time)`

Оновлює дані ріелтора в базі даних. Метод виконує два SQL-запити: один для оновлення даних у таблиці `ro_realtors`, інший для оновлення даних у таблиці `ro_users`. Якщо обидва запити виконуються успішно, здійснюється коміт транзакції, інакше виконується відкат.

User.php

```
<?php
class User
{
    private $id;

    public function __construct() {
```

```

        if (isset($_COOKIE['uid'])) {
            $this->id = $_COOKIE['uid'];
        }
    }

    public function checkUserId(): bool
    {
        global $db_controller;
        return $db_controller->checkUserId($this->id);
    }

    public function getAccessLevel(): int
    {
        if(!isset($this->id)) {
            return 0;
        }

        global $admins_id, $realtor;
        $realtors_id = $realtor->getArrayRealtorsId();
        if($this->checkUserId()) {
            if(in_array($this->id, $admins_id)) return 3;
            if (in_array($this->id, $realtors_id)) return 2;
            return 1;
        }
        return 0;
    }

    public function getBalance( $id)
    {
        global $db_controller;
        return $db_controller->getBalance($id);
    }

    public function selectUsers()
    {
        global $db_controller;
        return $db_controller->selectUsers();
    }
}

```

Клас User:

- `__construct()`

Конструктор класу, який ініціалізує об'єкт, перевіряючи, чи існує куки uid. Якщо куки є, то зберігається значення в поле \$id.

- `checkUserId(): bool`

Перевіряє, чи існує користувач з ідентифікатором \$id у базі даних. Метод виконує SQL-запит, щоб знайти користувача в таблиці ro_users за заданим ID і повертає true, якщо користувач знайдений, або false, якщо ні.

- getAccessLevel(): int

Повертає рівень доступу користувача. Якщо користувач не існує (немає куки uid), то повертається 0 (відсутність доступу). Якщо користувач є в базі даних, перевіряється, чи є він адміністратором (рівень 3), ріелтором (рівень 2) або звичайним користувачем (рівень 1).

- getBalance(\$id)

Отримує баланс користувача за його ідентифікатором. Метод виконує SQL-запит для отримання значення поля balance для користувача з вказаним \$id і повертає його.

- selectUsers()

Отримує всіх користувачів з таблиці ro_users. Метод виконує SQL-запит для вибірки всіх даних користувачів і повертає масив з інформацією про кожного користувача.