

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.

" ____ " _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: "Розробка інформаційної системи веб-платформи оголошень про
нерухомість"

Виконав: студент групи 4142

_____ *Токаренко О. М.*
(підпис)

Керівник роботи:

_____ *Старший викладач*
(посада, науковий ступінь вчене звання)

_____ *Беркунський Є. Ю.*
(підпис)

м. Миколаїв – 2024 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.
" ____ " _____ 2024 р.

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

Студенту _____ Токаренко Олександр Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: "Розробка інформаційної системи веб-платформи оголошень про нерухомість"

Керівник роботи _____ Беркунський Є. Ю.

Затверджені наказом ректора № 243-уч від "18" березня 2024 року.

2. Термін подання роботи: 20 червня 2024 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. Розробка концепції інформаційної системи. 2. Розробка проекту інформаційної системи. 3. Реалізація проекту інформаційної системи. 4. Охорона праці. Висновки.

5. Перелік презентаційних матеріалів Слайди: Актуальність, ціль, завдання; Модель концепції; Концептуальна модель даних; Логічна модель даних; Фізична модель даних; Математичне забезпечення; Програмне забезпечення; Технічне забезпечення; Реалізація проекту; Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Беркунський Є. Ю., старший викладач		
2	Беркунський Є. Ю., старший викладач		
3	Беркунський Є. Ю., старший викладач		
4	Беркунський Є. Ю., старший викладач		

7. Дата видачі завдання 05 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Розробка змісту інформаційної системи		
2	Розробка концепції інформаційної системи		
3	Розробка технічного проекту інформаційної системи		
4	Реалізація проекту інформаційної системи		
5	Охорона праці		
6	Вступ, висновки, список використаної літератури, додатки		
7	Презентація		
8			
9			
10			

Студент _____
(підпис)

Токаренко О. М.
(ПІБ)

Керівник роботи _____
(підпис)

Беркунський Є. Ю.
(ПІБ)

АНОТАЦІЯ

Обсяг пояснювальної записки складає 69 сторінок. У роботі наведено 16 ілюстрацій, 16 таблиць, 2 додатки та 26 джерел у списку літератури.

Об'єктом проекту є розробка інформаційної системи веб-платформи оголошень про нерухомість. Предметна область проекту охоплює процеси пошуку, реалізації та управління нерухомістю. Метою проекту є створення інноваційної веб-платформи, що забезпечить зручний доступ до інформації про нерухомість та спростить процеси її купівлі, продажу та оренди.

У рамках дослідження використовувалися методи аналізу вимог користувачів, проектування системи та інтерфейсу, розробки програмного забезпечення та тестування. Результати дослідження включають функціональну веб-платформу з оголошеннями про нерухомість, що відповідає потребам користувачів.

Основні технологічні характеристики проекту включають використання мови програмування Java, фреймворку Spring, а також бази даних PostgreSQL.

Отримані результати можуть бути застосовані в сфері нерухомості для полегшення процесів пошуку та управління нерухомістю. Висновки та прогнози щодо розвитку проекту підкреслюють важливість та перспективність подальшої розробки та вдосконалення інформаційної системи.

Ключові слова: *нерухомість, веб-платформа, оголошення, користувач, розробка, технології, база даних.*

ABSTRACT

The volume of the explanatory note is 69 pages. The paper contains 16 illustrations, 16 tables, 2 appendices and 26 sources in the references.

The object of the project is to develop an information system for a web platform for real estate ads. The subject area of the project covers the processes of search, implementation and management of real estate. The goal of the project is to create an innovative web platform that will provide convenient access to information about real estate and simplify the processes of its purchase, sale and lease.

The study used methods for analyzing user requirements, system and interface design, software development and testing. The results of the study include a functional web platform with real estate ads that meets the needs of users.

The main technological characteristics of the project include the use of the Java programming language, the Spring framework, as well as the PostgreSQL database.

The results obtained can be applied in real estate to facilitate the processes of search and management of real estate. Conclusions and forecasts for the development of the project emphasize the importance and prospects of further development and improvement of the information system.

Keywords: real estate, web platform, ads, user, development, technology, database.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 РОЗРОБКА КОНЦЕПЦІЇ ІС ВЕБ-ПЛАТФОРМИ ОГОЛОШЕНЬ ПРО НЕРУХОМІСТЬ.....	9
1.1 Опис об'єкта дослідження та аналіз предметної області.....	9
1.2 Формування вимог до проекту.....	12
1.3 Розробка концепції системи.....	15
РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	19
2.1 Рішення щодо інформаційного забезпечення.....	19
2.2 Рішення з математичного забезпечення.....	31
2.3 Рішення щодо програмного забезпечення.....	32
2.4 Рішення щодо технічного забезпечення.....	37
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ ІС.....	39
РОЗДІЛ 4 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ	45
ВИСНОВОК.....	51
СПИСОК ЛІТЕРАТУРИ.....	52
ДОДАТОК А – Інструкція користувачу	54
ДОДАТОК Б – Текст програми.....	63

ВСТУП

У кваліфікаційній роботі розробляється проект інформаційної системи веб-платформи оголошень про нерухомість, яка дозволить користувачам знаходити відповідні об'єкти нерухомості для купівлі або оренди, а також виставляти власні оголошення. Основний акцент робиться на створенні зручного та інтуїтивно зрозумілого інтерфейсу, який забезпечить користувачів всією необхідною інформацією для прийняття обґрунтованих рішень.

Сучасний ринок нерухомості активно розвивається, збагачуючись новими технологіями та інструментами, що сприяють поліпшенню процесів пошуку, купівлі та оренди нерухомості. У контексті цього динамічного розвитку виникає потреба в ефективних та зручних інформаційних системах, які забезпечать користувачів широким вибором об'єктів нерухомості та комфортною взаємодією з ними.

Метою кваліфікаційної роботи є підвищення ефективності просування нерухомості і задоволення потреб людей. Основною ціллю є створення інформаційної системи, яка надасть користувачам можливість швидкого та зручного пошуку та розміщення оголошень про нерухомість.

У розділі 1 розроблена концепція інформаційної системи. Зроблений опис об'єкту дослідження і аналіз предметної області. Сформовані вимоги до інформаційної системи. На основі сформованих вимог розроблена концепція, яка визначає функціонал інформаційної системи.

У розділі 2 надано детальний огляд розробки інформаційної системи, включаючи рішення щодо інформаційного забезпечення, математичне та програмне забезпечення, а також технічне забезпечення.

У розділі 3 представлена реалізація проекту та інструкція для користувача, що дозволить ефективно використовувати розроблену веб-платформу.

У розділі 4 описана інформація про охорону праці і техніку безпеки, що є важливою складовою будь-якого проекту, зокрема в розробці програмного забезпечення.

Реалізація проекту буде досягнута за рахунок новаторського підходу до використання програмних засобів, які дозволять забезпечити надійність, швидкість та зручність використання веб-платформи для користувачів.

Отже, враховуючи динаміку ринку нерухомості та зростаючу популярність онлайн-платформ, розробка даної інформаційної системи виявляється актуальною та перспективною задачею.

РОЗДІЛ 1 РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ-ПЛАТФОРМИ ОГОЛОШЕНЬ ПРО НЕРУХОМІСТЬ

1.1 Опис об'єкта дослідження та аналіз предметної області

Об'єктом дослідження є фізична особа, яка займається розробкою інформаційної системи веб-платформи оголошень про нерухомість. Основними напрямками діяльності є збір, реєстрація та відображення інформації про нерухомість, обробка запитів користувачів та підтримка взаємодії між покупцями та продавцями.

Проведемо огляд поточного стану ринку нерухомості та його основних тенденцій. Це включає аналіз попиту та пропозиції на ринку житлової та комерційної нерухомості, особливості ринкової конкуренції, а також оцінку перспектив розвитку галузі в середньостроковій і довгостроковій перспективі. Цей аналіз дозволить визначити основні сегменти аудиторії, їх потреби та переваги.

В даний час український ринок нерухомості зазнає певних викликів та змін, зокрема, пов'язаних з політичною, військовою та економічною ситуацією в країні. Однак, незважаючи на це, спостерігається стабільний попит на житло як з боку населення, так і інвесторів.

Попит на ринку житлової нерухомості в Україні ґрунтується переважно на потребах населення у покращенні житлових умов. У той самий час, попит на комерційну нерухомість визначається, переважно, бізнес-потребами та інвестиційними можливостями.

Основні тенденції ринку включають збільшення пропозиції житла в новобудовах, активний розвиток ринку оренди житлової та комерційної нерухомості, а також зростання цін на нерухомість у столичному регіоні і великих містах.

Ринкова конкуренція нерухомості України є досить високою, особливо у великих містах, де діє безліч агентств нерухомості, забудовників та приватних

продавців. Це створює умови для активної боротьби за клієнтів та постійне вдосконалення послуг.

Також зараз у військовий час багато родин покидають своє житло через небезпеку обстрілів та руйнування. Через це зростає попит на пошук житла. Такі системи будуть зручні для користувачів, адже вони можуть перебуваючи в одному місті знаходити житло по всій Україні.

Оцінка перспектив розвитку галузі у середньостроковій та довгостроковій перспективі пов'язана з очікуваним зростанням економіки країни, покращенням інвестиційного клімату, а також із впровадженням інноваційних підходів у сфері нерухомості, таких як цифровізація процесів продажу та оренди. Для рішення таких задач потрібно розробити інформаційну веб-платформу оголошень про нерухомість.

Проведемо аналіз існуючих веб-платформ оголошень про нерухомість. Це дозволить виявити основні переваги та недоліки конкуруючих продуктів, а також зрозуміти, які функціональні можливості є найбільш популярними серед користувачів. На основі цього аналізу можна буде визначити основні вимоги до функціоналу платформи, що розробляється.

Однією з найбільших таких систем є OLX. OLX – популярний сервіс оголошень в Україні, де користувачі можуть купити або продати б/у товари, нерухомість, автомобілі, послуги та багато іншого. Перевагами цієї платформи є широка аудиторія користувачів, інтуїтивно зрозумілий інтерфейс, розміщення безкоштовних оголошень. Проте має деякі недоліки, а саме: низька модерація оголошень, часті випадки шахрайства, обмежені можливості фільтрації та пошуку [3]. На рисунку 1.1 зображена головна сторінка системи OLX.

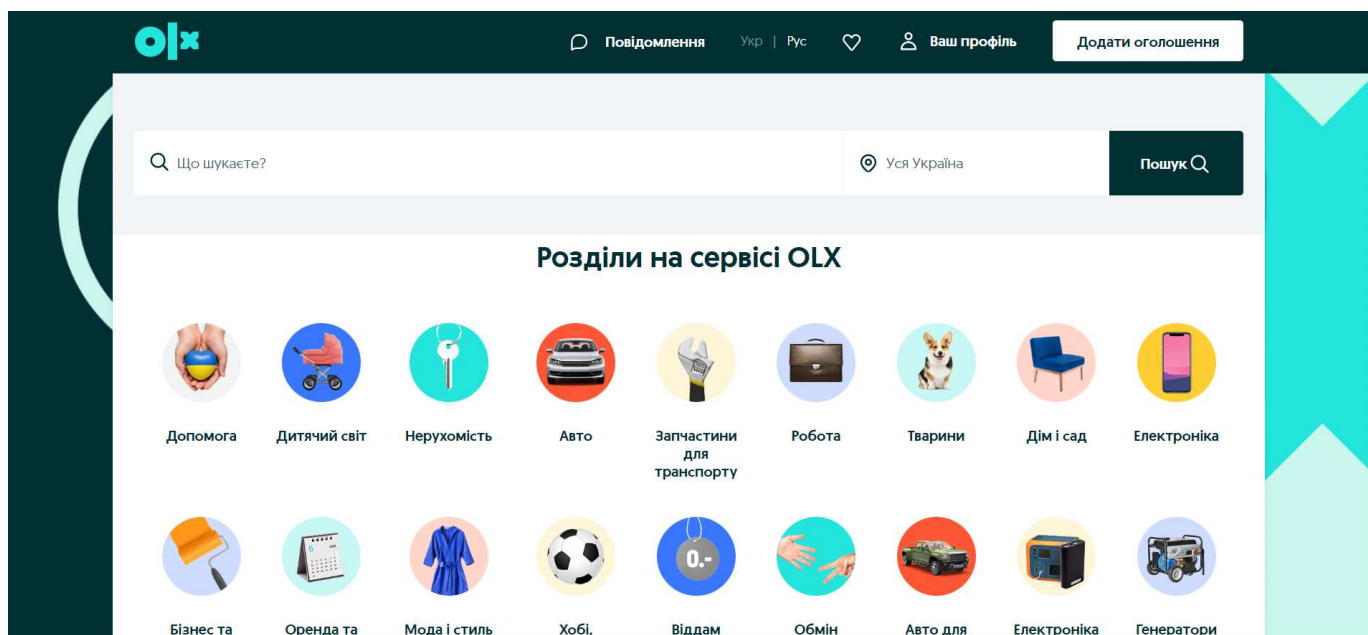


Рисунок 1.1 – Інформаційна система OLX

Також на ринку представлена система Dim.ria. Ця система на рівні з OLX займає провідне місце в Україні серед платформ онлайн-оголошень та має високий рівень довіри з боку користувачів, проте, як показує практика, багато користувачів скаржаться на складний інтерфейс [4]. На рисунку 1.2 зображена головна сторінка системи Dim.ria.

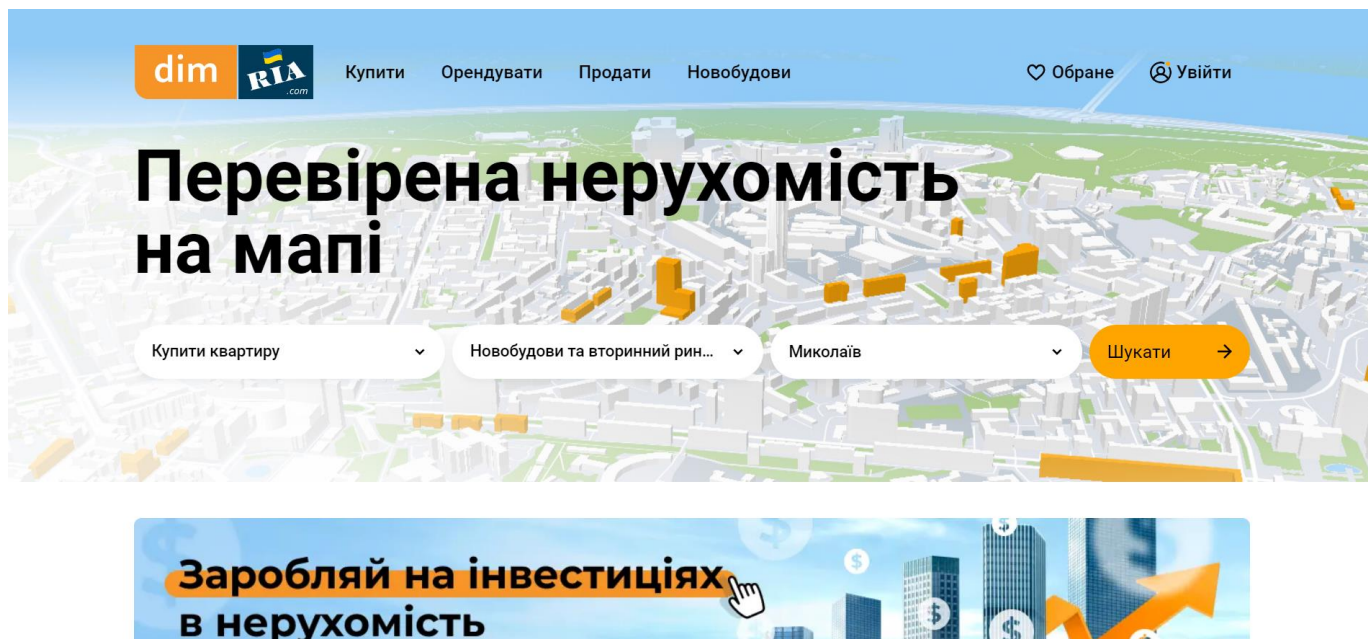


Рисунок 1.2 – Інформаційна система Dim.ria

Аналіз проблем, із якими стикаються користувачі під час використання наявних ІС оголошень в Україні, є ключовим для виявлення напрямків поліпшення розроблюваної платформи. Зокрема, багато користувачів відмічають недоліки у розміщенні інформації про об'єкти нерухомості, такі як відсутність ключових параметрів, що ускладнює процес вибору. Незручності з пошуком також викликають занепокоєння, особливо через складний інтерфейс.

Для вирішення виявлених проблем та покращення функціоналу розроблюваної платформи необхідно враховувати найкращі практики наявних систем та впроваджувати нові технології, спрямовані на забезпечення зручності та ефективності використання. Однією з перших важливих кроків є підвищення якості розміщення інформації про нерухомість, забезпечивши повноту та достовірність даних. Це можна досягти, створивши чіткі критерії для оголошень, які міститимуть всі необхідні характеристики об'єктів нерухомості.

Далі, важливо зробити процес пошуку нерухомості максимально зручним для користувачів. Це можна зробити шляхом розробки потужного та ефективного інструменту фільтрації, який дозволить швидко і точно знаходити потрібні об'єкти за заданими критеріями. Додатково, важливо забезпечити інтуїтивно зрозумілий та легкий для користувачів інтерфейс, що сприятиме зручності взаємодії з платформою.

З огляду на зазначені проблеми та потреби користувачів, метою розробки є створення інноваційного та ефективного сервісу, який забезпечить користувачам зручність, безпеку та надійність в пошуку та розміщенні нерухомості.

1.2 Формування вимог до проекту

Користувачі можуть переглядати всі оголошення, які були виставлені, а також свої власні та відфільтровані оголошення. Крім того, вони мають можливість відкрити

або приховати свої оголошення, а також редагувати та видаляти їх. Фільтр може бути застосований до всіх характеристик нерухомості.

Має бути можливість авторизації та реєстрації.

Неавторизованим користувачам надається можливість переглядати всі оголошення, зберігати оголошення, дивитись детальну інформацію про них та використовувати фільтр для фільтрації оголошень за характеристиками. Авторизовані користувачі можуть викладати оголошення, дивитись свої оголошення, видаляти їх та редагувати.

Користувачам доступна можливість відновлення свого паролю.

Інформація про ціну нерухомості має бути відображена в двох валютах: в доларі та гривні. Наприклад, якщо користувач при викладенні оголошення вкаже ціну в гривні, то має бути також відображена відповідна ціна в доларах і навпаки. Обчислення повинне виконуватись за актуальним курсом Національного банку України.

Викладення оголошень є безкоштовним. Користувачі нічого не витрачають за перегляд та розміщення оголошень.

При викладенні оголошень має бути перевірка зображень та опису нерухомості на цензуру. Це може бути можливе через використання сторонніх сервісів.

Якщо оголошення виставлено більше 60 днів тому, воно автоматично приховується. Щоб знову відкрити оголошення, користувач, який виставив це оголошення, повинен зробити це самостійно.

На сторінках повинна розташовуватись реклама від Google.

Якщо користувач довго не взаємодіє з платформою та має виставлені оголошення, він отримує повідомлення на електронну скриньку з попередженням про видалення його оголошень через 30 днів після останнього візиту на сайт. Користувачу дається 5 днів на відвідування платформи.

Система зберігає повну та точну інформацію про кожне оголошення, включаючи всі характеристики нерухомості, контактні дані власника та додаткові фотографії нерухомості.

Веб-платформа повинна бути розроблена з використанням сучасних технологій та фреймворків, що забезпечують безпеку та масштабованість програми

Передбачити регулярне оновлення системи.

Платформа має працювати на різних пристроях та браузерах, включаючи комп'ютери, планшети та смартфони.

Інтерфейс має бути інтуїтивно зрозумілим та зручним для користувачів усіх рівнів досвіду.

Платформа повинна забезпечувати швидкий та ефективний пошук нерухомості, а також швидке завантаження сторінок та обробку запитів.

Система повинна забезпечувати безпеку та захист даних користувачів.

На основі цих вимог розробляється концепція інформаційної системи.

1.3 Розробка концепції інформаційної системи

На основі сформованих вимог розроблено концепцію інформаційної системи, яка представлена на рисунку 1.3.

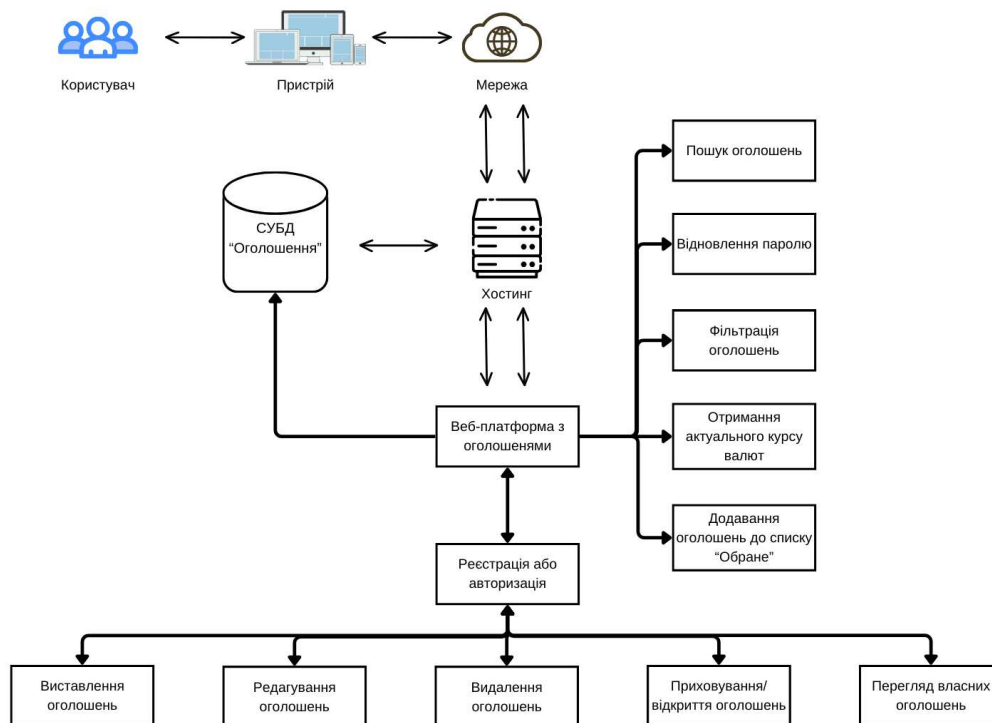


Рисунок 1.3 – Модель концепції інформаційної системи

Інформаційна система повинна виконувати такі функціональні задачі:

- реєстрація та авторизація користувачів. Користувачі можуть зареєструватись або авторизуватись для доступу до додаткових функцій;
- перегляд оголошень. Всі користувачі можуть переглядати оголошення;
- викладення оголошень. Тільки авторизовані користувачі можуть викладати власні оголошення;
- відновлення паролю. Користувачі можуть відновити пароль, який забули.
- управління оголошеннями. Авторизовані користувачі можуть редагувати, видаляти та приховувати свої оголошення;

- фільтрація оголошень. Користувачі відфільтрувати оголошення за різними характеристиками;
- додавання оголошень до списку «Обране». Користувачі можуть зберегти оголошення, які їх зацікавили.
- отримання актуального курсу валют. Кожен день система самостійно отримує актуальний курс валют наданих Національним банком України.

Опишемо систему за функціями управління. Одна з головних функцій системи це управління оголошеннями. Сюди входить викладення оголошень, редагування оголошень, видалення оголошень та приховування/відкриття оголошень. Також система забезпечує управління користувачами, а саме реєстрація, авторизація відновлення паролю. І насамперед управління фільтрацією та відображенням оголошень.

При переході на головну сторінку, користувач отримує всі актуальні оголошення. На цій сторінці він може знайти основну інформацію про нерухомість, таку як ціна, кількість кімнат, житлова площа та заголовок оголошення. Також біля кожного з оголошень є кнопка для відкриття вікна з повною інформацією про нерухомість.

Для того, щоб отримати відфільтровані оголошення, він повинен у верхній частині екрану натиснути Фільтр, потім вибрати, яку нерухомість хоче отримати, а потім вже вибрати або вписати ті параметри нерухомості, які йому потрібні.

Для реєстрації або авторизації користувач повинен натиснути відповідні кнопки у верхньому правому кутку і ввести особисті дані.

На сторінці «Мої оголошення» знаходяться всі оголошення, які виклав користувач. Свої оголошення користувач може видалити, редагувати, або приховати/відкрити натиснувши на відповідні кнопки.

У вікні викладення оголошень користувач вводить повну інформацію про нерухомість.

Для отримання актуального курсу валют можна розробити асинхронну частину, яка буде отримувати автоматично данні в певний час.

Для розробки надійної, масштабованої та продуктивної інформаційної системи варто уважно підібрати інструменти та технології. На початку, розглянемо використання мови програмування Java. Java відома своєю надійністю та потужністю, широко використовується для розробки веб-додатків завдяки своїй портативності та масштабованості [3]. Однією з ключових складових також є фреймворк Spring, який дозволяє створювати ефективні та модульні програми, забезпечуючи управління залежностями та інверсію управління [8]. Для зберігання даних ми можемо скористатися Amazon S3, яке забезпечує надійне та масштабоване хмарне сховище [9]. Для бази даних рекомендується використання реляційної бази даних PostgreSQL, яка добре поєднується з Java та Spring, забезпечуючи стабільність та ефективність [4]. На рівні фронтенду рекомендується використання HTML, CSS, а також Bootstrap для створення адаптивного дизайну. Для забезпечення безпеки додатка та контролю доступу до даних використовується Spring Security, що забезпечує аутентифікацію та авторизацію користувачів. Для перевірки інформації про нерухомість рекомендується використати сервіс Sighengine, який працює на базі штучного інтелекту та надає розробникам та бізнесу потужні технології аналізу зображень, відео та тексту.

В рамках сервісних програмних засобів використовується Log4j для ефективного логування подій у системі. Log4j дозволяє відстежувати та записувати різноманітні дії користувачів та системні події, що відбуваються в системі. Використання цього інструменту допомагає виявляти помилки, відлагоджувати програмний код та аналізувати роботу системи з метою підвищення її ефективності та стабільності [11].

Система повинна бути розгорнута на надійних серверах з достатньою потужністю та резервними копіями даних для забезпечення безперервної доступності та безпеки даних.

На основі розробленої концепції інформаційної системи розроблюється технічний проект.

РОЗДІЛ 2 РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Рішення щодо інформаційного забезпечення

Для забезпечення роботи інформаційної системи потрібна база даних. Для управління базою даних використовується СУБД PostgreSQL.

PostgreSQL — це потужна об'єктно-реляційна система керування базами даних (СКБД), яка відкрита, безкоштовна та розповсюджується під ліцензією PostgreSQL. Це одна з найпопулярніших баз даних серед розробників через її надійність, гнучкість та великий набір функцій. PostgreSQL підтримує широкий спектр функціоналу, включаючи складні SQL запити, відносини між таблицями, транзакції зі збереженням цілісності даних, реплікацію та багато іншого. Вона використовується в різноманітних сферах, від веб-розробки до наукових досліджень та корпоративних систем.

Визначимо структуру та організацію бази даних, необхідної для ефективного функціонування платформи. Нижче представлені концептуальна, логічна та фізична модель бази даних. Моделі створені за допомогою програмного забезпечення draw.io.

Концептуальна модель даних являє собою високорівневий опис основних сутностей і їх взаємозв'язків в системі. На рисунку 2.1 зображено концептуальну модель даних, виділено основні сутності. Кожна з цих сутностей має свої атрибути, що визначають їхні властивості та характеристики. Концептуальна модель допомагає зрозуміти загальну структуру бази даних та визначити основні компоненти, необхідні для подальшого проектування та розробки платформи.

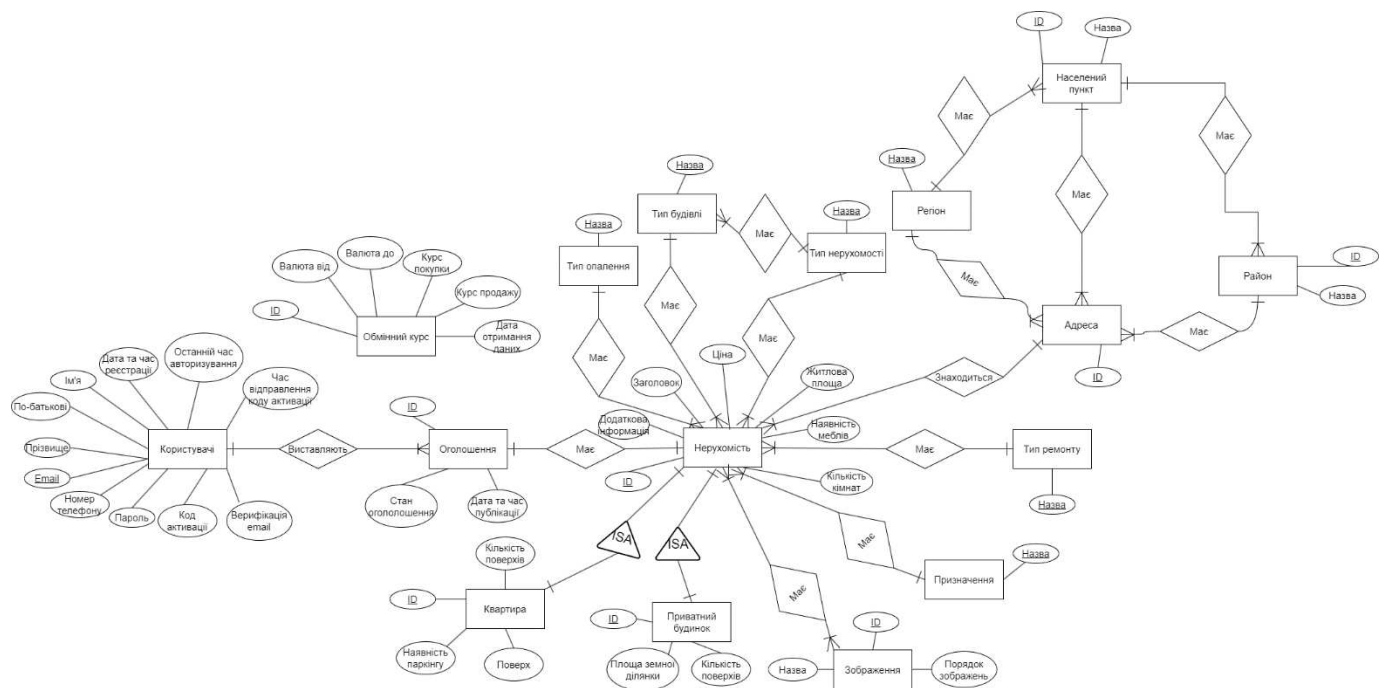


Рисунок 2.1 – ER-модель

Логічна модель дозволяє більш детально розглянути структуру даних та визначити спосіб їхнього зберігання та організації в базі даних, що є основою для подальшої реалізації та оптимізації функціональності платформи оголошень. На рисунку 2.2 показана логічна модель даних, де кожна сутність перетворена в таблицю, а зв'язки між ними відображені з використанням зовнішніх ключів. Наприклад, таблиця «Оголошення» містить атрибути, такі як "ID", "Стан оголошення», "Дата та час публікації", "Email Користувача". Зовнішній ключ "Email Користувача" пов'язує оголошення з його власником.

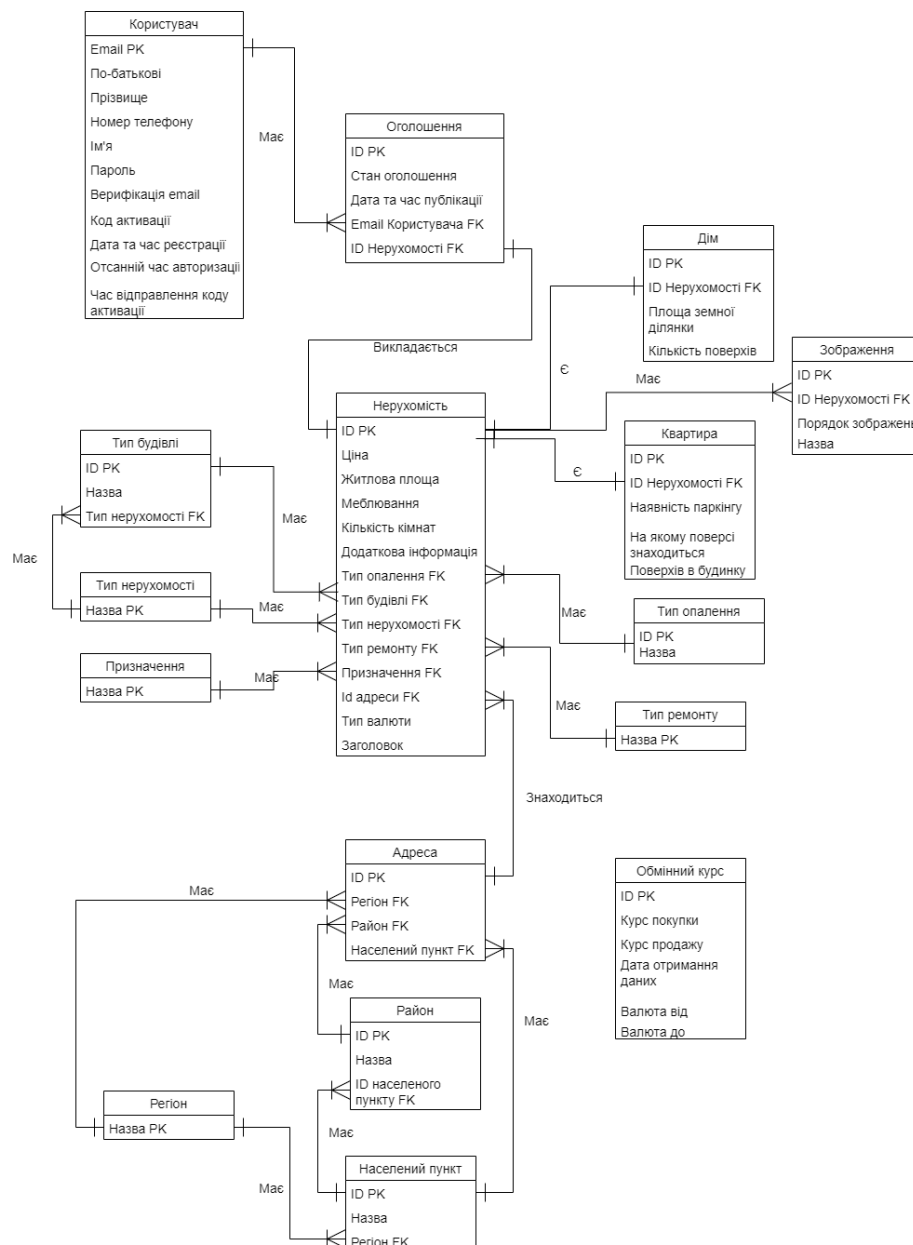


Рисунок 2.2 – Логічна модель даних

Кожне реляційне відношення відповідає одній сутності і до нього вносяться всі атрибути цієї сутності. Для кожного відношення визначаються первинний ключ та зовнішні ключі (відповідно до схеми БД).

У тому випадку, якщо базове відношення не має потенційних ключів, вводиться первинний сурогатний ключ, який не несе смислового навантаження і служить тільки для ідентифікації записів. На таблицях 1-16 зображені всі відношення БД.

Таблиця 2.1

Схема відношення Користувачі (Users)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Email	Email	S(30)	Обов'язкове поле	Первинний
Ім'я	Firstname	S(25)	Обов'язкове поле	Альтернативний
По-батькові	Patronymic	S(25)	Необов'язкове поле	
Прізвище	Lastname	S(25)	Необов'язкове поле	
Номер телефону	Phone_number	S(10)	Обов'язкове поле	Альтернативний
Пароль	Password	S(15)	Обов'язкове поле. Має містити цифри, букви. Довжина від 8 до 15 символів.	
Верифікація email	Verified_email	B	Обов'язкове поле. True, якщо email верифікований, якщо email не верифікований – False.	
Код активації	Activation_code	S(6)	Необов'язкове поле	
Дата та час реєстрації	Date_register	D	Обов'язкове поле	
Останній час авторизації	Last_login	D	Обов'язкове поле	
Час відправлення коду активації	Effective_time	D	Необов'язкове поле	

Таблиця 2.2

Схема відношення Оголошення (Advertisement)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле.	Первинний
Стан оголошення	Open	B	Обов'язкове поле. True, якщо оголошення відкрите, якщо оголошення приховуване – False.	
Дата та час публікації	Date_and_time_publication	D	Обов'язкове поле.	
Email Користувача	User_email	S(30)	Обов'язкове поле.	Зовнішній ключ (до Users)
ID Нерухомості	Realty_ID	N	Обов'язкове поле.	Зовнішній ключ (до Realty)

Таблиця 2.3

Схема відношення Нерухомість (Realty)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
Ціна	Price	N	Обов'язкове поле, > 0	
Житлова площа	Housing_area	N	Обов'язкове поле, > 0	
Кількість кімнат	Number_of_rooms	N	Обов'язкове поле, > 0	

Меблювання	Furnishing	B	Обов'язкове поле	
Додаткова інформація	Additional_information	S(350)	Обов'язкове поле	
Тип опалення	Heating	S(30)	Обов'язкове поле	Зовнішній ключ (до Heating)
ID Типу будівлі	Construction_id	N	Обов'язкове поле	Зовнішній ключ (до Construction)
Тип нерухомості	Property	S(30)	Обов'язкове поле	Зовнішній ключ (до Property)
Тип ремонту	Repair	S(35)	Обов'язкове поле	Зовнішній ключ (до Repair)
Призначення	Purpose	S(20)	Обов'язкове поле	Зовнішній ключ (до Purpose)
Id Адреси	Address_Id	N	Обов'язкове поле	Зовнішній ключ (до Address)
Валюта	Currency	S(3)	Обов'язкове поле	

Таблиця 2.4

Схема відношення Дім (House)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
Площа земної ділянки	land_area	N	Обов'язкове поле, > 0	
Кількість поверхів	Floors	N	Обов'язкове поле, > 0	
ID Нерухомості	Realty_id	N	Обов'язкове поле	Зовнішній ключ (до Realty)

Таблиця 2.5

Схема відношення Квартира (Flat)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
Наявність паркінгу	Parking	B	Обов'язкове поле	
Поверх	Floor	N	Обов'язкове поле, > 0	
ID Нерухомості	Realty_id	N	Обов'язкове поле	Зовнішній ключ (до Realty)

Таблиця 2.6

Схема відношення Тип опалення (Heating)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(30)	Обов'язкове поле	Первинний

Таблиця 2.7

Схема відношення Призначення (Purpose)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(20)	Обов'язкове поле	Первинний

Таблиця 2.8

Схема відношення Тип ремонту (Repair)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(35)	Обов'язкове поле	Первинний

Таблиця 2.9

Схема відношення Тип будівлі (Construction)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(30)	Обов'язкове поле	
Тип нерухомості	Property	S(25)	Обов'язкове поле	Зовнішній ключ (до Property)
ID	ID	S(25)	Обов'язкове поле	Первинний

Таблиця 2.10

Схема відношення Тип нерухомості (Property)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(30)	Обов'язкове поле	Первинний

Таблиця 2.11

Схема відношення Адреса (Address)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
Регіон	Region	S(50)	Обов'язкове поле	Зовнішній ключ (до Region) *
ID Району	District_id	N	Необов'язкове поле	Зовнішній ключ (до District)*
ID Населеного пункту	Locality_id	N	Обов'язкове поле	Зовнішній ключ (до Locality)*

*Регіон у поєднанні з ID району, ID населеного пункту – альтернативним ключем.

Таблиця 2.12

Схема відношення Район (District)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Примітка</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
ID Населеного пункту	Locality_id	N	Обов'язкове поле.	Зовнішній ключ (до Locality)*
Назва	Title	S(50)	Обов'язкове поле.	*

*Назва у поєднанні з ID населеного пункту утворюють альтернативний ключ.

Таблиця 2.13

Схема відношення Населений пункт (Locality)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
ID	ID	N	Обов'язкове поле	Первинний
Регіон	Region	S(50)	Обов'язкове поле	Зовнішній ключ (до Region)
Назва	Title	S(50)	Обов'язкове поле	

Таблиця 2.14

Схема відношення Регіон (Region)

<i>Зміст поля</i>	<i>Ім'я поля</i>	<i>Тип, довжина</i>	<i>Обмеження цілісності</i>	<i>Ключі</i>
Назва	Title	S(50)	Обов'язкове поле	Первинний

Таблиця 2.15

Схема відношення Обмінний курс (Exchange_rate)

Зміст поля	Ім'я поля	Тип, довжина	Обмеження цілісності	Ключі
ID	ID	N	Обов'язкове поле	Первинний
Курс покупки	buy_rate	N	Обов'язкове поле	
Курс продажу	sale_rate	N	Обов'язкове поле	
Валюта від	currency_from	S(3)	Обов'язкове поле	
Валюта до	currency_to	S(3)	Обов'язкове поле	
Дата отримання даних	effective_date	D	Обов'язкове поле	

Таблиця 2.16

Схема відношення Зображення (Image)

Зміст поля	Ім'я поля	Тип, довжина	Обмеження цілісності	Ключі
ID	ID	N	Обов'язкове поле	Первинний
Назва	title	N	Обов'язкове поле	
ID Нерухомості	Realty_id	N	Обов'язкове поле	Зовнішній ключ (до Realty)
Порядок зображень	Image_order	N	Обов'язкове поле	

Фізична модель даних визначає спосіб зберігання та організацію даних у реальній базі даних. Вона враховує особливості обраної системи управління базами даних (СУБД) та оптимізує структуру для ефективного доступу та обробки даних. Фізична модель даних веб-платформи оголошень представлена на рисунку 2.3, де

кожна таблиця перетворена в схему зберігання, враховуючи типи даних та індекси для оптимізації запитів.

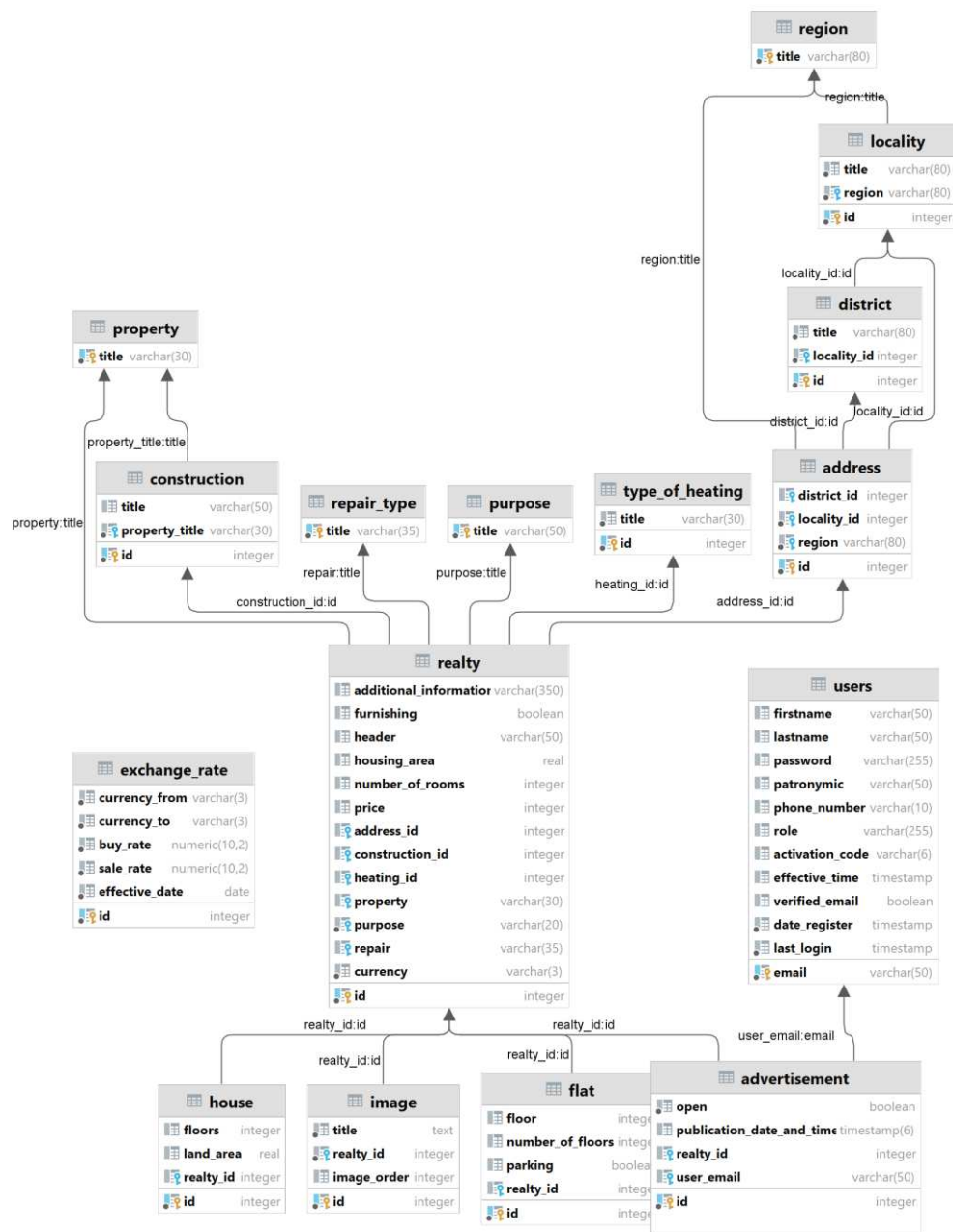


Рисунок 2.3 – Фізична модель даних

Для заповнення частини бази даних, що пов’язана з користувачами (таблиця “users”) використовується особиста інформація надана користувачем.

Для заповнення частини бази даних, що пов'язана з оголошенням (таблиця “advertisement”, “realty”, “flat”, “image” та “house”) використовується інформація надана користувачем про нерухомість.

Для заповнення частини бази даних, що пов'язана з курсом валют (таблиця “exchage_rate”) використовується відкрита інформація про актуальний курс валют.

Для заповнення частини бази даних, що пов'язана з регіонами, населеними пунктами та районами (таблиці “region”, “locality” та “district”) використовується інформація у відкритому доступі про регіони, населені пункти цих регіонів та райони.

Для заповнення частини бази даних, що пов'язана з типом опалення, типом ремонту, конструкцією будинку (таблиці “type_of_heating”, “repair_type” та “construction”) використовується інформація про всі типи опалення, всі типи ремонту та всі типи конструкції будинків.

На основі представлених вище рішень розроблені рішення з математичного забезпечення інформаційної системи.

2.2 Рішення з математичного забезпечення

Блок-схему загального алгоритму роботи додатку зображено на рисунку 2.4.

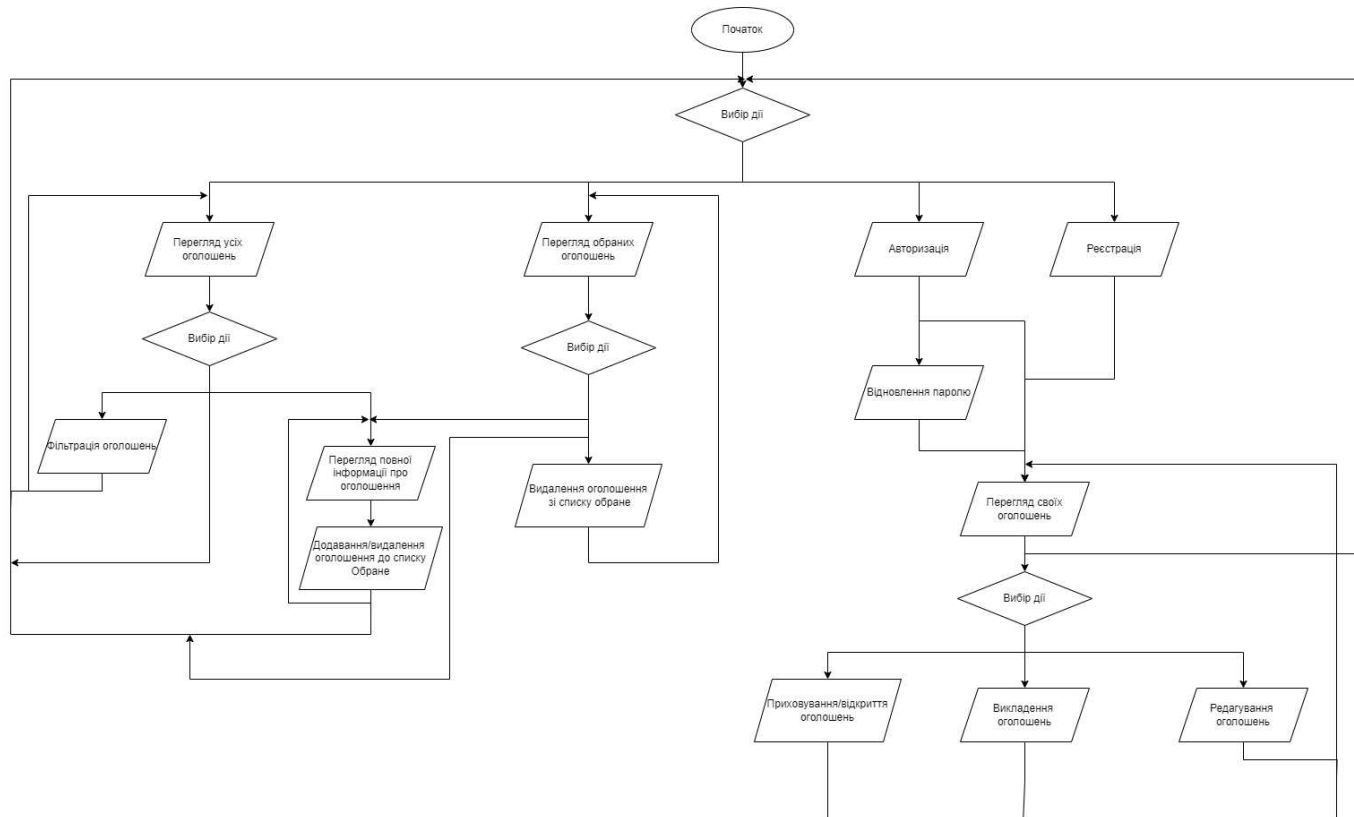


Рисунок 2.4 - Блок-схема загального алгоритму роботи додатку

Користувач заходить на сайт та бачить всі актуальні оголошення. У верхній частині екрана знаходиться навігаційна частина через яку користувач може перейти на сторінку авторизації, реєстрації, перегляд власних оголошень (якщо він авторизований) та на сторінку з обраними оголошеннями.

Через головну сторінку користувач переходить на сторінку фільтрації, де може ввести дані, за якими потрібно відфільтрувати оголошення. Після цього йому відкриється та сама головна сторінка, але вже з відфільтрованими оголошеннями.

На головній сторінці знаходиться тільки основна частина інформації про оголошення. Щоб відобразити всю інформацію потрібно натиснути відповідну кнопку.

На сторінці з повною інформацією про оголошення користувач може додавати та видаляти оголошення з списку обране, переглядати зображення нерухомості.

На сторінці з власними оголошеннями користувач може видалити оголошення, приховати або відкрити, подивитись повну інформацію та перейти на сторінки редагування та викладення оголошень.

На сторінці авторизації можна відновити пароль.

На основі представлених рішень розробляються рішення з програмного забезпечення інформаційної системи

2.3 Рішення щодо програмного забезпечення

База даних розробляється на основі СУБД PostgreSQL. PostgreSQL - це потужна система керування базами даних (СУБД), яка володіє високою надійністю, масштабованістю та функціональністю.

Порівняно з іншими СУБД, PostgreSQL має ряд переваг, що зробили його привабливим вибором для проекту. Перш за все, він відкритий та має активну спільноту розробників, що гарантує швидку підтримку та вирішення проблем. Крім того, PostgreSQL має розширену підтримку стандартів SQL, що робить розробку додатків більш гнучкою та сумісною.

Ще однією перевагою PostgreSQL є його висока продуктивність та швидкодія, особливо при оптимізації запитів та роботі з великими обсягами даних. Він також має широкий спектр функцій, включаючи підтримку геоданих та розширені можливості роботи з JSON, що дозволяє ефективно використовувати базу даних для різних потреб проекту.

У проєкті PostgreSQL використовувався для зберігання інформації про користувачів, їх оголошення.

Інформація про PostgreSQL підготовлена на основі матеріалу [21].

Розроблено програмне забезпечення для бази даних, з використанням, приведених вище, інструментів, проведено налагодження з формування та ведення бази даних.

Під час розробки програмного забезпечення використовувалися наступні методи та засоби. Основною мовою програмування для вибрано Java. Java є мовою програмування, що відома своєю стабільністю та надійністю. Вона володіє високим рівнем читабельності коду та широким спектром фреймворків для розробки серверних додатків. Крім того, Java має велику спільноту розробників та багато інструментів та бібліотек, які полегшують процес розробки та підтримки проєктів [22].

Java використовувався для реалізації бізнес-логіки та взаємодії з базою даних. Використання Java дозволило створити надійний та масштабований код, який легко супроводжувати та розширювати.

Також для розробки використовувався фреймворк Spring, який включає в себе модуль Spring Security. Spring є одним з найпопулярніших і розширених фреймворків для розробки веб-додатків на Java.

Spring надає широкий спектр функціональності та інструментів для створення надійних та ефективних серверних додатків. Він включає в себе інструменти для управління залежностями, що дозволяє легко і зручно керувати компонентами додатку. Також Spring забезпечує потужні засоби для конфігурації, що дозволяє гнучко налаштовувати різні аспекти додатку з використанням анотацій або XML-конфігурацій.

Однією з ключових особливостей Spring є його підтримка інверсії управління та впровадження залежностей (IoC/DI), що спрощує створення та управління об'єктами в додатку. Це дозволяє підвищити модульність та переносимість коду, а також полегшує тестування додатку.

Інформацію про Spring було підготовлено на основі матеріалу [8].

Spring було використано для створення різних компонентів додатку, таких як контролери для обробки HTTP-запитів, сервіси для виконання бізнес-логіки та репозиторії для взаємодії з базою даних.

Spring Security - це розширення фреймворка Spring, яке забезпечує потужні засоби для автентифікації та авторизації веб-додатків. Він дозволяє забезпечити захист ресурсів додатку, контролюючи доступ користувачів до різних функцій та ресурсів. Spring Security надає можливість налаштовувати рівні доступу, використовувати різні стратегії аутентифікації (наприклад, основані на формах, токенах або стандартні протоколи автентифікації, такі як OAuth), а також забезпечує захист від різних видів атак, таких як перехоплення сесій або введення шкідливого коду (injection attacks) [18].

Spring Security був використаний для забезпечення безпеки веб-додатку, зокрема автентифікації та авторизації користувачів, контролю доступу до ресурсів та захисту від різних видів атак. Використання цього фреймворку дозволило ефективно налаштувати механізми безпеки та забезпечити захист конфіденційності та цілісності даних.

Для розробки веб-сторінок використовувалися мови розмітки та стилів, такі як HTML5, CSS та фреймворк Bootstrap.

HTML5 (HyperText Markup Language 5) - це остання основна версія мови розмітки, що використовується для структурування та відображення гіпертекстових документів у всесвітній мережі Інтернет. Вона є застарілою рекомендацією Консорціуму Всесвітньої павутини (W3C), і її поточна специфікація, відома як HTML Living Standard, підтримується Робочою групою з веб-технологій гіпертекстових застосунків (WHATWG), спільно з провідними виробниками браузерів, такими як Apple, Google, Mozilla та Microsoft.

HTML5 включає ряд нововведень, які сприяють більш сумісним та ефективним імплементаціям. Вона розширює, поліпшує та раціоналізує розмітку, доступну для

документів, і вводить нові розміткові та програмні інтерфейси застосунків (API) для розробки складних веб-додатків.

HTML5 використовувався для створення структури та маркування контенту веб-додатку. За допомогою HTML5 було створено різноманітні веб-сторінки, включаючи сторінку авторизації, головну сторінку, сторінку реєстрації, сторінку фільтрації. HTML5 дозволив визначити структуру документів та їхній зовнішній вигляд, що забезпечило коректне відображення вмісту на різних пристроях та браузерях.

Інформація про HTML5 підготовлена на основі матеріалу [23].

Каскадні таблиці стилів (CSS) - це мова таблиць стилів, яка використовується для визначення оформлення і стилізації документа, написаного на мові розмітки, такі як HTML або XML (включаючи діалекти XML, такі як SVG, MathML або XHTML). CSS є ключовою технологією всесвітньої павутини, разом з HTML і JavaScript [14].

CSS використовувався для стилізації і візуального оформлення веб-сторінок в проєкті. За допомогою CSS було визначено кольори, шрифти, розміри, відступи та інші властивості елементів HTML, що дозволило забезпечити єдність стилю та зовнішній вигляд всієї системи. Також CSS було використано для розробки адаптивного дизайну, щоб забезпечити коректне відображення сторінок на різних пристроях та роздільних здатностях екрану.

Bootstrap є потужним фреймворком для розробки веб-інтерфейсів, який базується на HTML, CSS та JavaScript. Він містить готові компоненти, такі як кнопки, форми, картки, навігаційні панелі тощо, які можна легко використовувати для побудови стильного та адаптивного веб-дизайну. Використання Bootstrap дозволяє розробникам ефективно створювати зручні та сучасні інтерфейси для веб-сторінок, що добре працюють на різних пристроях і розмірах екранів.

У проєкті Bootstrap використовувався для швидкої і зручної стилізації веб-елементів та компонентів, таких як кнопки, форми, таблиці, навігаційні панелі та багато іншого. Його гнучка сіткова система також дозволила створювати адаптивний

дизайн, що підлаштовується під різні розміри екранів та пристроїв. Використання Bootstrap значно спростило розробку і забезпечило єдність стилю та вигляду усього веб-додатку [13].

Для розробки програмного забезпечення використовувалися також різні інструменти та сервіси, зокрема Amazon S3, IntelliJ IDEA та Git.

Amazon S3 (Simple Storage Service) є обліковим сховищем від Amazon Web Services, яке надає можливість зберігання та керування великими обсягами даних у хмарному середовищі. Використання Amazon S3 дозволяє забезпечити надійне та масштабоване зберігання різноманітних типів даних, таких як зображення, відео, аудіофайли тощо, і забезпечує швидкий та безперебійний доступ до них для користувачів [10].

Amazon S3 використано для зберігання зображень нерухомості. Цей сервіс надає можливість безпечно зберігати великі обсяги фотографій, пов'язаних з нерухомістю, у хмарному середовищі. Використання Amazon S3 забезпечило надійне та масштабоване зберігання цих даних, а також швидкий та безперебійний доступ до них для користувачів веб-додатку.

IntelliJ IDEA є потужним інтегрованим середовищем розробки (IDE) для роботи з різними мовами програмування, включаючи Java, JavaScript та інші. IntelliJ IDEA надає розробникам ряд корисних функцій, таких як автодоповнення коду, вбудовані інструменти для аналізу та відлагодження коду, інтеграцію з системами керування версіями тощо. Використання IntelliJ IDEA дозволяє розробникам підвищити продуктивність та якість свого коду під час розробки програмного забезпечення [25].

Вся програмна частина розроблена за допомогою середовища розробки IntelliJ IDEA. Це середовище надає зручність та ефективність під час написання коду на різних мовах програмування, включаючи Java, JavaScript та інші. Завдяки автодоповненню коду, вбудованим інструментам для аналізу та відлагодження коду, а також інтеграції з системами керування версіями, було підвищено продуктивність та якість програмного забезпечення.

Git є розподіленою системою керування версіями, яка використовується для керування та відстеження змін вихідного коду програмного забезпечення. Використання Git дозволяє розробникам ефективно спільно працювати над проектом, зберігати різні версії коду, вносити та відстежувати зміни, а також вирішувати конфлікти між різними гілками розробки. Git також надає можливість зберігати вихідний код в хмарному репозиторії, що дозволяє зручно зберігати та спільно використовувати код між різними членами команди розробників.

Інформація про Git підготовлена на основі матеріалу [24].

GIT використовувався для керування версіями програмного коду проекту. Кожен розробник міг клонувати репозиторій проекту, створювати та переходити між гілками для внесення змін, а також об'єднувати різні гілки для збереження та розподілу коду. GIT дозволив ефективно керувати розробкою проекту, забезпечуючи можливість відслідковувати зміни, вносити правки та забезпечувати спільну роботу розробників над кодом.

Веб-платформа розроблена для роботи на різних операційних системах, включаючи Windows, macOS та Linux.

Для розробки програмного забезпечення будуть використані технічні засоби, які описані в наступному підрозділі.

2.4 Рішення щодо технічного забезпечення

Для взаємодії з інформаційною системою необхідний браузер. Це означає, що пристрої, з яких можна взаємодіяти з інформаційною системою, повинні мати достатні технічні характеристики для запуску браузера.

З урахуванням цієї інформації визначимо технічні вимоги до пристроїв користувача (встановлені на основі системних вимог браузера Google Chrome):

Для Windows:

- Процесор Intel Pentium 4 або новіший з підтримкою SSE3;
- Мінімум 512 МБ оперативної пам'яті;
- Відеопам'ять не менше 64 МБ;
- Вільне місце на жорсткому диску не менше 350 МБ;
- Роздільна здатність екрану не менше 800 × 600;
- Клавіатура;
- Миша.

Для Linux:

- Технічні характеристики аналогічні до Windows.

Для MacOS:

- Пристрій працює на базі ОС X El Capitan (10.11) і новіших версій.

Для Android:

- Пристрій працює на базі Android 5.0 (Lollipop) і новіших версій.

Для iOS:

- Пристрій працює на базі iOS 12.2 (iPadOS 12.2) і новіших версій.

Сервер, на якому працюватиме вся розроблювана інформаційна система, буде хоститися на платформі Amazon EC2. Amazon EC2 (Elastic Compute Cloud) - це веб-сервіс обчислювальних можливостей в хмарному середовищі, наданий компанією Amazon Web Services (AWS). EC2 дозволяє користувачам орендувати віртуальні сервери із заздалегідь визначеними характеристиками або масштабувати ресурси в залежності від потреб. Технічні характеристики та налаштування сервера відповідатимуть веб-платформи оголошень.

На основі розробленого технічного забезпечення реалізується проект інформаційної системи.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОЕКТУ ІС

Для використання системи була розроблена робоча документація, що містить інструкції для користувачів щодо використання функціоналу інформаційної системи. Усю необхідну інформацію про використання системи можна знайти у додатку А.

Користувацька частина системи містить багато форм для заповнення, кожна зі своїми полями. Для полегшення використання системи кожне поле містить підказку для коректного заповнення.

Кожна кнопка користувацької частини інформаційної системи має невелику довідку щодо функціоналу, яка розташована біля неї. Також є модуль відстеження заповнених форм та коректності заповнень, який перевіряє правильність заповнення форм та не дозволяє продовжити процес викладення, редагування або фільтрації оголошень, якщо форма не заповнена або заповнена неправильно.

Інформаційну систему протестовано на коректність роботи всіх модулів. Також була перевірена коректність обміну даних між сервером та користувачами. інтеграція та робота сторонніх сервісів, використаних при розробці системи. Всі дані для заповнення бази даних також були перевірені на коректність.

Головна сторінка веб-додатку інформаційної системи представлена на рисунку 3.1

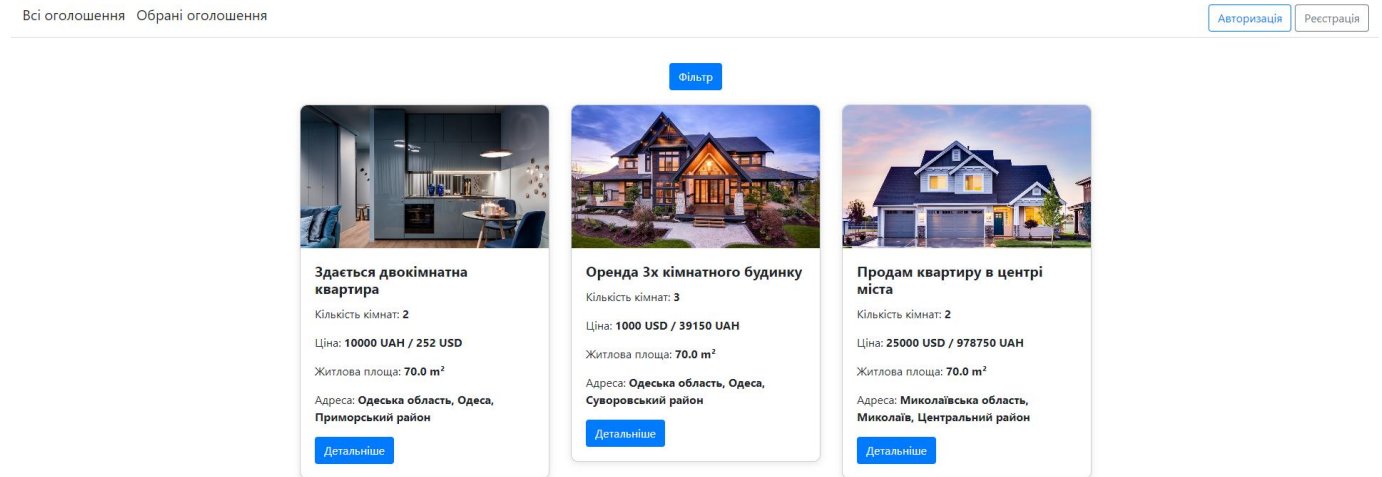


Рисунок 3.1 – Головна сторінка веб-додатку

Сторінка авторизації представлена на рисунку 3.2

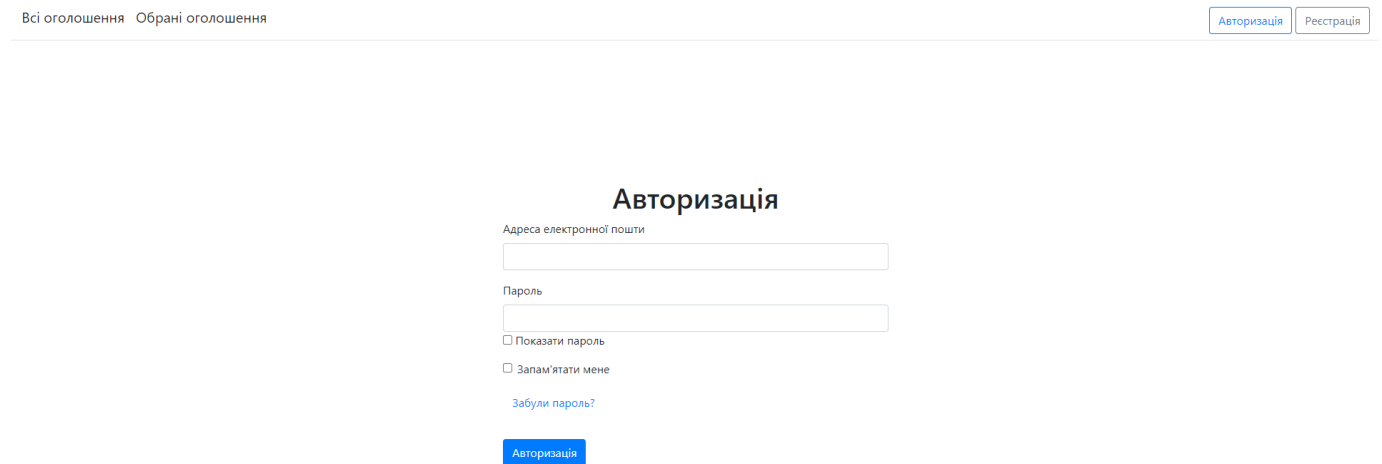


Рисунок 3.2 – Сторінка авторизації користувачів

Сторінка реєстрації користувачів представлена на рисунку 3.3

Всі оголошення
Обрані оголошення

Авторизація
Реєстрація

Реєстрація

Ім'я

По-батькові

Прізвище

Електронна пошта

Номер телефону

Пароль

☐ Показати пароль

Реєстрація

Рисунок 3.3 – Сторінка реєстрації користувачів

Сторінка з вибраними оголошеннями представлена на рисунку 3.4.

Всі оголошення
Обрані оголошення

Авторизація
Реєстрація

Обрані оголошення



Здається двокімнатна квартира

Кількість кімнат: 2

Ціна: 10000 UAH / 252 USD

Житлова площа: 70.0 m²

Адреса: Одеська область, Одеса, Приморський район

Детальніше
Видалити

Рисунок 3.4 - Сторінка з вибраними оголошеннями

Сторінка з формою для фільтрації оголошень представлена на рисунку 3.5.

Рисунок 3.5 - Сторінка з формою для фільтрації оголошень

Сторінка з усіма оголошеннями, що виклав користувач, представлена на рисунку 3.6.

Мої оголошення

Продам квартиру в центрі міста
Кількість кімнат: 2
Ціна: 25000 USD / 978750 UAH
Житлова площа: 70.0 m²
Адреса: Миколаївська область, Миколаїв, Центральний район

Оренда 3х кімнатного будинку
Кількість кімнат: 3
Ціна: 1000 USD / 39150 UAH
Житлова площа: 70.0 m²
Адреса: Одеська область, Одеса, Суворовський район

Здається двокімнатна квартира
Кількість кімнат: 2
Ціна: 10000 UAH / 252 USD
Житлова площа: 70.0 m²
Адреса: Одеська область, Одеса, Приморський район

Викласти оголошення

Рисунок 3.6 - Сторінка для ведення сайту

Сторінка з формою для викладення оголошення представлена на рисунку 3.7.

Всі оголошення Мої оголошення Обрані оголошення **Вихід**

Викладення оголошення

Для чого викладаєте оголошення?
Надання в оренду

Заголовок до вашого оголошення

Саме за цим заголовком користувачі будуть шукати ваше оголошення

Валюта
☐ UAH ☐ USD

Ціна

Житлова площа

Кількість кімнат

Стан ремонту
Обрати

Тип будинку

Рисунок 3.7 - Сторінка з формою для викладення оголошення

Сторінка з формою для редагування даних оголошення представлена на рисунку 3.8.

Всі оголошення Мої оголошення Обрані оголошення **Вихід**

Редагування оголошення

Для чого викладаєте оголошення?
Надання в оренду

Заголовок до вашого оголошення
Оренда 3х кімнатного будинку

Саме за цим заголовком користувачі будуть шукати ваше оголошення

Валюта
☐ UAH ☒ USD

Ціна
1000

Житлова площа
70.0

Площа земельної ділянки (в сотках)
10.0

Кількість кімнат
3

Стан ремонту

Рисунок 3.8 - Сторінка з формою для редагування оголошення

Сторінка з повною інформацією про нерухомість представлена на рисунку 3.9.

Всі оголошення
Мої оголошення
Обрані оголошення

Вихід

Продам квартиру в центрі міста

Опис

Продажа дома

Власник оголошення

Олександр Токаренко

Призначення	Продаж
Ціна	25000 USD
Адреса	Миколаївська область, Миколаїв, Центральний район
Кількість кімнат	2
Житлова площа	70.0 м²
Стан ремонту	Без ремонту
Тип будинку	Сталінка
Меблі	Ні
Тип опалення	Централізоване
Парковка	Ні
Поверх	1 з 3

Додати в обране

Рисунок 3.9 - Сторінка з повною інформацією про нерухомість

Апробація веб-платформи показали позитивні результати. Проект досягнув поставленої цілі, розроблена інформаційна система полегшує пошук та розміщення оголошень користувачами.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ І ТЕХНІКА БЕЗПЕКИ

Вимоги безпеки та захисту здоров'я працівників при роботі з екранними пристроями затверджено наказом Міністерства соціальної політики від 14.02.2018 № 207. Ці вимоги поширюються на всіх суб'єктів господарювання незалежно від форм власності, організаційно-правової форми та видів діяльності та встановлюють мінімальні вимоги безпеки та захисту здоров'я при здійсненні роботи, пов'язаної з використанням екранних пристроїв незалежно від їх типу та моделі.

Вимоги не поширюються на:

- Робочі місця здобувачів освіти у комп'ютерних класах (кабінетах, аудиторіях тощо) навчальних закладів;
- Робочі місця пілотів, водіїв або операторів транспортних засобів, обладнані екранними пристроями у системах обробки даних на борту засобів сполучення, екранні пристрої у складі машин та обладнання, що переміщуються у процесі роботи;
- Робочі місця працівників, які займаються обслуговуванням, ремонтом та налагодженням екранних пристроїв;
- Портативні системи обробки даних, якщо вони не завжди використовуються на робочому місці;
- Обчислювальні машини (калькулятори), касові апарати та пристрої з невеликими пристроями індикації даних або результатів вимірювання;
- Цифрові друкарські машини, які обладнані візуальними дисплейними терміналами (дисплейні друкарські машини);
- Планшети, смартфони, мобільні телефони.

Роботодавець повинен проінформувати працівників під розписку про умови праці та наявність на їх робочих місцях небезпечних та шкідливих виробничих факторів (фізичних, хімічних, біологічних, психофізіологічних), які виникають при

роботі з екранними пристроями та ще не усунуті, а також про можливі наслідки їх впливу на здоров'я працівників.

Роботодавець повинен забезпечити навчання та перевірку знань працівників з питань охорони праці та безпечного використання екранних пристроїв до початку роботи з ними, а також у випадках модифікації та організації роботи обладнання.

Роботодавець повинен вжити відповідних заходів, щоб забезпечити відповідність робочого місця працівника цим вимогам.

Під час облаштування робочого місця працівника з екранними пристроями необхідно вибирати таке обладнання, яке не створює надмірного шуму і не виділяє надмірного тепла. Рівні шуму на робочих місцях осіб, які працюють з екранними пристроями, повинні відповідати вимогам Санітарних норм виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених ухвалою Головного державного санітарного лікаря України від 01.12.1999 № 37.

Роботодавець повинен за рахунок тривалості робочої зміни організувати внутрішні регламентовані перерви для відпочинку відповідно до Державних санітарних правил та норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря від 10.12.1998 № 7.

Роботодавець повинен забезпечити власним коштом проведення медичних оглядів працівників відповідно до вимог Порядку проведення медичних оглядів працівників певних категорій, затвердженим наказом Міністерства охорони здоров'я України від 21.05.2007 № 246.

За результатами цих оглядів роботодавець у разі потреби повинен забезпечити виконання відповідних оздоровчих заходів.

Роботодавець зобов'язаний за необхідності проводити лабораторні дослідження умов праці працівників з метою виявлення шкідливих та небезпечних факторів виробничого середовища, тяжкості та напруженості трудового процесу (зокрема щодо

виявлення ризиків, пов'язаних із погіршенням зору, порушенням фізичного стану, стресом) та здійснювати заходи щодо усунення виявлених ризиків .

Робочі місця працівників з екранними пристроями мають бути спроектовані і мати такі розміри, щоб працівники мали простір зміни робочого становища і рухів.

Для забезпечення безпеки та захисту здоров'я працівників все випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня (вплив на людину факторів середовища - шуму, вібрації, забруднювачів, температури тощо, що не викликає соматичних або психічних розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій з точки зору безпеки та охорони здоров'я працівників.

Організація робочого місця працівника з екранними пристроями повинна забезпечувати відповідність всіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характером виконуваних робіт.

Висвітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном та навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.

Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями повинен підтримуватися на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 0.12.1999 №42 (далі - ДСП 3.3.6.042-99).

Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість при розміщенні екрана, клавіатури, документів та відповідного обладнання.

Робоче крісло має бути стійким і дозволяти працівникові з екранними пристроями легко рухатися та займати зручне положення.

Сидіння має регулюватися по висоті, спинка сидіння як по висоті, так і по нахилу. Слід передбачати підніжку для тих, кому це потрібно для зручності.

Мінімальні вимоги безпеки під час роботи з екранними пристроями:

- Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень;
- Після завершення роботи екранні пристрої слід відключати від електричної мережі;
- У разі виникнення аварійної ситуації необхідно негайно відключити пристрій від електричної мережі.

Не допускається:

- Виконувати технічне обслуговування, ремонт та налагодження екранних пристроїв безпосередньо на робочому місці працівника під час роботи з екранними пристроями;
- Відключати захисні пристрої, самовільно проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;
- Працювати з екранними пристроями, де під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності.

При виконанні робіт операторського типу, пов'язаних з нервово-емоційною напругою, у приміщеннях при роботі з екранними пристроями, на пультах та постах управління технологічними процесами та в інших приміщеннях повинні дотримуватися оптимальних умов мікроклімату відповідно до вимог ГСН 3.3.6.042 - 99.

Мінімальні вимоги безпеки до екранних пристроїв:

- Екранні пристрої не повинні бути джерелом ризику для працівників;
- Все випромінювання, крім видимої частини електромагнітного спектра, має бути зведено до незначного рівня з погляду безпеки та охорони здоров'я працівників;
- Символи на екранних пристроях повинні бути чіткими, відповідного розміру. Між символами та рядками символів має бути належна відстань;

- Зображення на екрані має бути стабільним, без миготінь чи інших видів нестабільності;
- Яскравість та/або контрастність символів повинні легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов;
- Вибираючи екрани, слід віддавати перевагу таким екранам, які легко та вільно обертаються та нахиляються відповідно до потреби працівника;
- При необхідності може використовуватися окрема підставка або регульований стіл для розміщення екрана;
- Екран не повинен відблискувати або відображати світло, щоб не викликати дискомфорту у працівника під час роботи з екранними пристроями;
- Вибираючи клавіатуру, слід віддавати перевагу такій клавіатурі, яка відкидається і є автономною (відокремленою від екрану), щоб працівник міг вибрати зручну робочу позу та уникнути втоми рук (кисті та верхньої частини руки);
- Поверхня клавіатури має бути матовою, щоб уникнути відображення. Розташування клавіш і клавіші повинні полегшувати роботу з клавіатурою. Позначення клавіш має бути досить контрастним та розбірливим;
- Обладнання, що входить до робочої станції, не повинно виділяти надлишкового тепла, що може заподіяти незручності працівникам при роботі з екранними пристроями;
- При розробці, виборі, замовленні та модифікації програмного забезпечення, а також при розробці задач, що передбачають використання обладнання з екранними пристроями, роботодавець повинен керуватися таким програмним забезпеченням, яке відповідає розв'язуванню завданням і просто у використанні, а де необхідно адаптованим до рівня знань та досвіду працівника.

Відповідно до статті 43 Закону України «Про охорону праці» за порушення законодавства про охорону праці та невиконання приписів (розпоряджень) посадових осіб органів виконавчої влади з нагляду за охороною праці юридичні та фізичні особи,

які згідно із законодавством використовують найману працю, залучаються органами виконавчої влади з нагляду за охороною праці до сплати штрафу у порядку, встановленому законом.

Сплата штрафу не звільняє юридичну чи фізичну особу, яка відповідно до законодавства використовує найману працю, від усунення виявлених порушень у визначені терміни.

Максимальний розмір штрафу не може перевищувати п'яти відсотків середньомісячного фонду заробітної плати за попередній рік юридичної або фізичної особи, яка відповідно до законодавства використовує найману працю.

Несплата або неповна сплата юридичними або фізичними особами, які відповідно до законодавства використовують найману працю, штрафу тягне за собою нарахування пені на несплачену суму штрафу (його частини) з розрахунку 120% річних облікової ставки Національного банку України, що діяла в період такої несплати, за кожен день прострочення.

Інформація про охорону праці та техніку безпеки підготовлено на основі матеріалу [26].

ВИСНОВОК

У даній кваліфікаційній роботі розроблено інформаційну систему веб-платформи оголошень про нерухомість. Система призначена для полегшення процесів пошуку, купівлі, продажу та оренди нерухомості для користувачів.

Процес розробки системи розпочався з формулювання концепції та аналізу вимог до неї. На основі цього була розроблена концепція, яка визначила функціонал інформаційної системи. Технічний проект був визначений, у тому числі інформаційне та програмне забезпечення, а також технічне забезпечення системи.

Основними перевагами розробленої інформаційної системи є інтуїтивний та зручний інтерфейс, швидка фільтрація оголошень, можливість розміщення повної інформації про нерухомість, захист та безпека даних користувачів.

Впровадження даної інформаційної системи дозволить підвищити ефективність процесів просування нерухомості на ринку, задовольнити потреби користувачів у зручному пошуку та розміщенні оголошень, а також забезпечити належний рівень безпеки та надійності при роботі з системою.

Перспективами подальшого розвитку проекту можуть бути розширення функціоналу платформи, інтеграція з додатковими сервісами, а також постійне оновлення та удосконалення системи відповідно до нових тенденцій та вимог ринку нерухомості.

СПИСОК ЛІТЕРАТУРИ

1. Ажищев В., Морозова А. Методические указания к выполнению дипломного проектирования со специальности 6.050101 “Компьютерные науки”. Миколаїв, 2024.
2. Марченко А. В. Проектування інформаційних систем. Київ, 2016.
3. Литвин В. В., Пасічник В. В. Проектування інформаційних систем. Навчальний посібник (затв. МОН України). Львів : Магнолія 2006, 2023.
4. Берт Бейтс. Head First. Java. Київ : Ранок, 2022.
5. Ігор Рогов. PostgreSQL 15 зсередини. Харків : Print2print, 2022.
6. OLX. URL: <https://www.olx.ua/uk/> (дата звернення: 05.03.2024)
7. DIM.RIA. URL: <https://dom.ria.com/uk/> (дата звернення: 05.03.2024)
8. Spring Framework. URL: <https://spring.io/projects/spring-framework> (дата звернення: 05.03.2024)
9. Spring Framework. URL: https://uk.wikipedia.org/wiki/Spring_Framework (дата звернення: 07.03.2024)
10. Amazon S3. URL: https://uk.wikipedia.org/wiki/Amazon_S3 (дата звернення: 06.03.2024)
11. Apache Log4j. URL: <https://logging.apache.org/log4j/2.x/> (дата звернення 25.03.2024)
12. Структура HTML. URL: <http://weblib.com.ua/blog/struktura-html> (дата звернення: 25.03.2024)
13. Bootstrap. URL: <https://uk.wikipedia.org/wiki/Bootstrap> (дата звернення: 25.03.2024)
14. CSS. URL: <https://uk.wikipedia.org/wiki/CSS> (дата звернення: 25.03.2024)
15. pgAdmin 4. URL: <https://www.pgadmin.org/docs/pgadmin4/8.4/index.html> (дата звернення: 25.03.2024)

16. What is Entity Relationship Diagram (ERD). URL: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/> (дата звернення: 25.03.2024)
17. В чем разница между логической моделью данных и физической моделью данных. URL: <https://aws.amazon.com/ru/compare/the-difference-between-logical-and-physical-data-model/> (дата звернення: 31.03.2024)
18. Spring Security. URL: <https://docs.spring.io/spring-security/reference/index.html> (дата звернення: 31.03.2024)
19. Sightengine Documentation. URL: <https://sightengine.com/docs/> (дата звернення: 06.04.2024)
20. Требования безопасности при работе с компьютером. URL: <https://hrliga.com/index.php?module=news&op=view&id=20212> (дата звернення: 08.04.2024)
21. Чем PostgreSQL лучше других SQL баз данных с открытым исходным кодом. URL: <https://habr.com/ru/articles/282764/> (дата звернення: 15.04.2024)
22. Java. URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення: 29.04.2024)
23. HTML5. URL: <https://uk.wikipedia.org/wiki/HTML5> (дата звернення: 29.04.2024)
24. Git. URL: <https://uk.wikipedia.org/wiki/Git> (дата звернення: 29.04.2024)
25. IntelliJ IDEA. URL: https://uk.wikipedia.org/wiki/IntelliJ_IDEA (дата звернення: 29.04.2024)
26. Організація системи керування виробничою безпекою. URL: <https://crust.ust.edu.ua/server/api/core/bitstreams/09fdc246-5ef4-49ab-8c75-940b971b7cb7/content> (дата звернення: 15.05.2024)

ДОДАТОК А – Інструкція користувачу

На головній сторінці представлені всі актуальні оголошення та основна інформація про них. Натиснувши на поле Детальніше, що зображене на рисунку А.1, можна побачити всю інформацію про дану нерухомість.



Рисунок А.1 – Поле для переходу на сторінку з повною інформацією про оголошення

На сторінці з інформацією про оголошення можна дізнатись повну інформацію про нерухомість. Також там представлена інформація про її власника. Щоб додати оголошення до списку обране, потрібно натиснути кнопку Додати в обране. Ця кнопка представлена на рисунку А.2.

Призначення	Продаж
Ціна	25000 USD
Адреса	Миколаївська область, Миколаїв, Центральний район
Кількість кімнат	2
Житлова площа	70.0 m²
Стан ремонту	Без ремонту
Тип будинку	Сталінка
Меблі	Ні
Тип опалення	Централізоване
Парковка	Ні
Поверх	1 з 3

Додати в обране

Рисунок А.2 – Кнопка для додавання оголошення до списку Обране

Кнопки авторизації та реєстрації зображені у правому верхньому куту. Ці кнопки представлені на рисунку А.3.

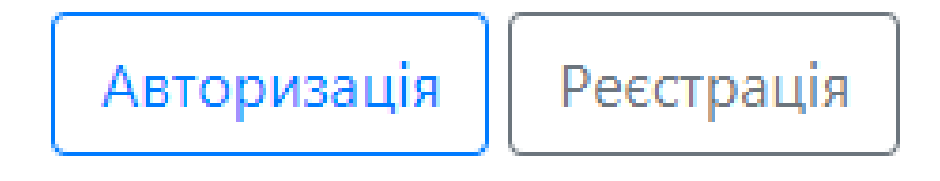


Рисунок А.3 – Кнопки авторизації та реєстрації

Щоб авторизуватись користувач повинен ввести свою електронну адресу та пароль. Якщо одне з полів введено не вірно, користувач отримує про це повідомлення

з помилкою. Ця помилка зображена на рисунку А.4. Система авторизує користувача тільки тоді, коли користувач ввів всі поля вірно.

The image shows a web form for authentication. At the top, the title 'Авторизація' is centered. Below it is a red error message box that says 'Неправильна електронна адреса або пароль'. The form has two input fields: 'Адреса електронної пошти' (Email address) containing 'test@gmail.com' and 'Пароль' (Password) containing masked characters. Below the password field are two checkboxes: 'Показати пароль' (Show password) and 'Запам'ятати мене' (Remember me). There is a link 'Забули пароль?' (Forgot password?) and a blue 'Авторизація' (Login) button at the bottom.

Рисунок А.4 – Помилка при авторизації

При реєстрації користувач вводить всю особисту інформацію, яку просить система. При введенні неправильної інформації користувач також отримує про це помилки. Ці помилки зображені на рисунку А.5. Після введення правильної інформації користувач отримає на пошту повідомлення для підтвердження електронної адреси.

Реєстрація

Ім'я

По-батькові

Прізвище

Введіть прізвище

Електронна пошта

Неправильний формат

Номер телефону

Номер телефону повинен мати формат 0XXXXXXXX, де X - цифри

Пароль

☒ Показати пароль

Пароль повинен містити латинські літери та цифри

Реєстрація

Рисунок А.5 – Помилки при реєстрації

Навігаційна частина знаходиться у верхній частині екрану. Натиснувши на поля відкриваються відповідні сторінки. Навігаційна частина зображена на рисунку А.6.



Рисунок А.6 – Навігаційна частина

На сторінці Мої оголошення знаходяться всі оголошення, що виклав користувач. Біля кожного з оголошень є кнопки для редагування та видалення оголошень. Для викладення оголошення користувач натискає відповідну кнопку у правій нижній частині екрану. Ці кнопки зображені на рисунку А.7.

Мої оголошення

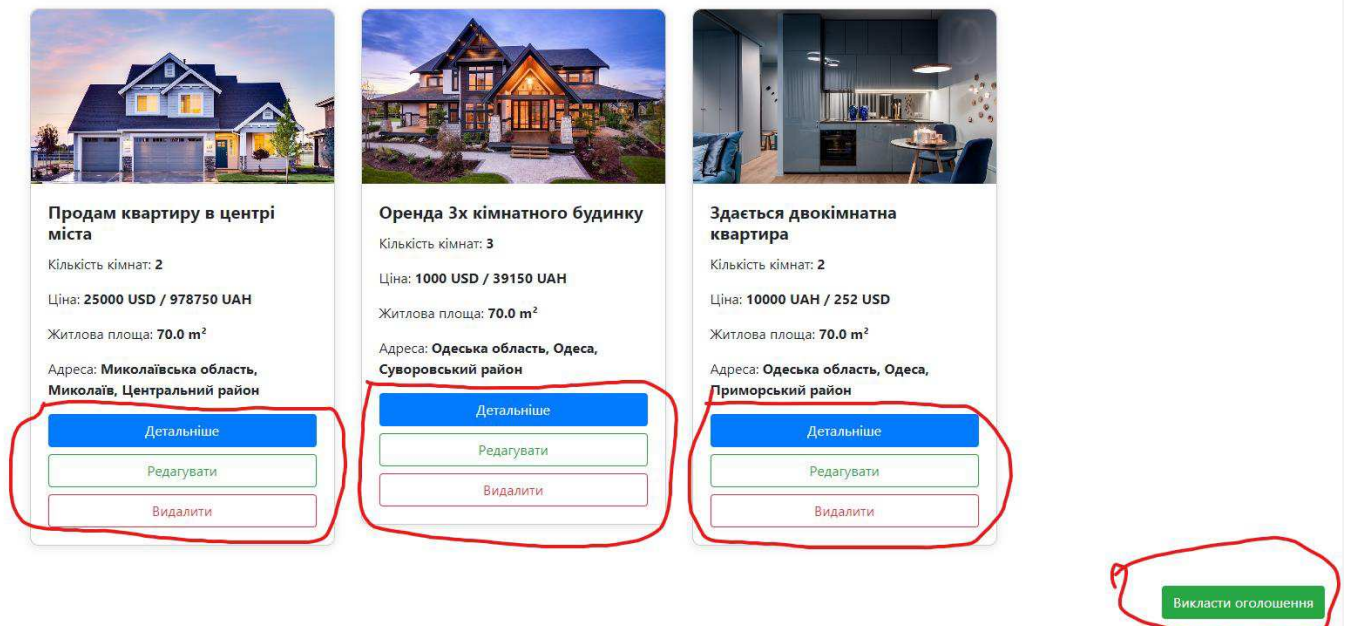


Рисунок А.7 – Сторінка Мої оголошення

При викладенні оголошень користувач вводить всю інформацію, що просить система. Всі поля є обов'язковими. Якщо якесь з полів введено не правильно, користувач отримує повідомлення з помилкою. Сторінка для викладення оголошення та можливі помилки, які може отримати користувач, представлені на рисунку А.8.

Викладення оголошення

Для чого викладаєте оголошення?

Продаж

Заголовок до вашого оголошення

Саме за цим заголовком користувачі будуть шукати ваше оголошення

Заголовок не повинен бути порожнім
Мінімальна довжина заголовку 10 символів, максимальна - 50 символів

Валюта

☐ UAH ☒ USD

Ціна

20000

Житлова площа

Поле не повинно бути порожнім

Кількість кімнат

2

Стан ремонту

Рисунок А.8 – Викладення оголошення

При редагуванні оголошення користувач вводить ті поля, які хоче змінити. Всі інші не змінені поля залишаються старими. Сторінка для редагування оголошення представлена на рисунку А.9.

Редагування оголошення

Для чого викладаєте оголошення?

Надання в оренду

Заголовок до вашого оголошення

Оренда 3х кімнатного будинку

Саме за цим заголовком користувачі будуть шукати ваше оголошення

Валюта

☐ UAH ☒ USD

Ціна

1500

Житлова площа

70.0

Площа земельної ділянки (в сотках)

10.0

Кількість кімнат

3

Стан ремонту

Рисунок А.10 – Сторінка редагування оголошень

Для знаходження оголошень за фільтром, потрібно на головній сторінці натиснути кнопку Фільтр. Після цього відкриється сторінка з формою для пошуку. Кнопка фільтр представлена на рисунку А.10.

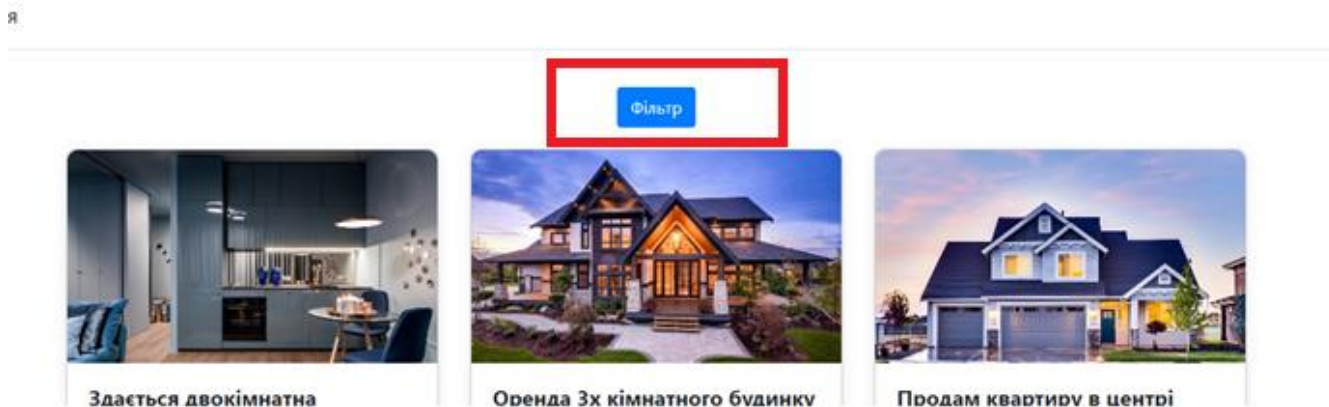


Рисунок А.10 – Кнопка для пошуку оголошень за фільтром

При фільтрації оголошень, користувач вводить тільки ті поля, за якими він хоче знайти нерухомість. Не потрібно заповнювати всі поля. Наприклад, якщо користувач шукає квартиру для купівлі і йому важливо тільки те, щоб ціна була від 20000 до 25000 доларів, то він вводить тільки це, а інші поля пропускає. Такий приклад заповнення фільтру зображено на рисунку А.11.

Фільтр

Підкатегорія

Продаж

Валюта

☐ UAH ☒ USD

Ціна

20000 25000

Кількість кімнат

від: до:

Житлова площа

від: до:

Стан ремонту

Всі оголошення

Тип будинку

Всі оголошення

Рисунок А.11 – Фільтрація оголошень

За потреби, якщо користувач забув свій пароль, на сторінці авторизації він натискає Забули пароль і після цього він має ввести свою електронну адресу, на яку зареєструвався. Якщо дана електронна адреса не буде знайдена, він отримає помилку. Якщо така адреса є, то користувачу на цю пошту надходить спеціальний код, який він повинен ввести. Після цього він може змінити пароль на новий. На рисунку А.12 представлено повідомлення, яке надійде користувачу на пошту. На рисунку А.13 представлено форму, куди він має вписати цей код.

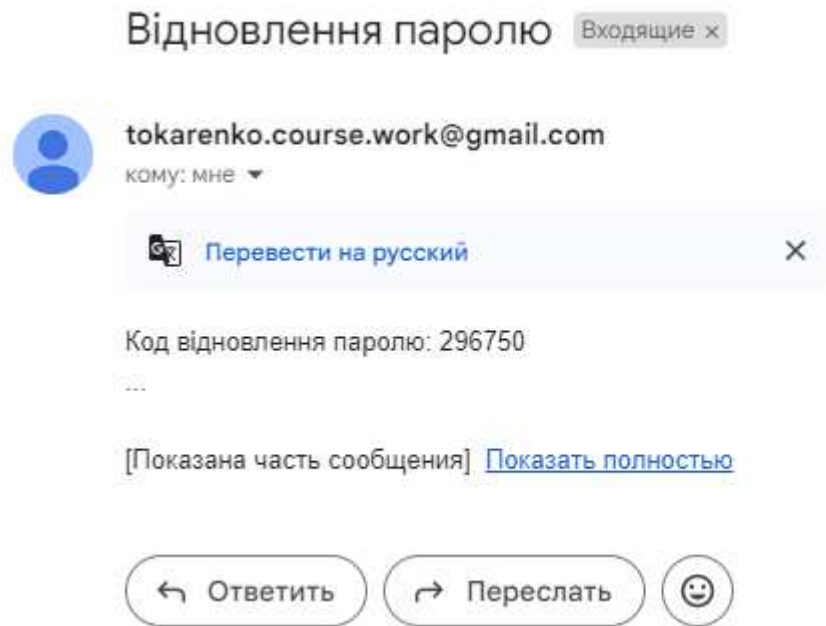


Рисунок А.12 – Код відновлення паролю.

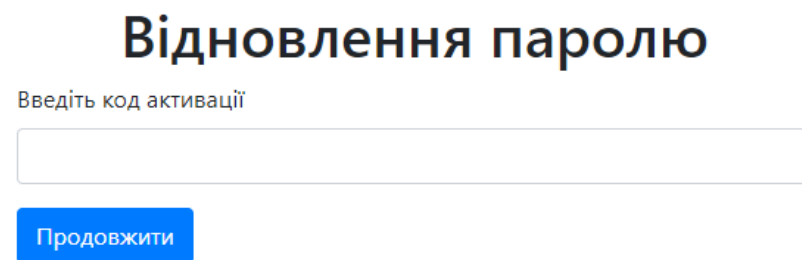


Рисунок А.13 – Поле для введення коду відновлення паролю

ДОДАТОК Б – Текст програми

AuthController.java:

```
package com.example.realestateads.controller.auth;

import com.example.realestateads.dto.UserRegisterDTO;
import com.example.realestateads.entity.User;
import com.example.realestateads.exception.UserAlreadyExistException;
import com.example.realestateads.repository.UserRepository;
import com.example.realestateads.service.dataServices.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/auth")
@RequiredArgsConstructor
public class AuthController {

    private final UserService userService;
    private final AuthenticationManager authenticationManager;
    private final UserRepository userRepository;

    @GetMapping("/login")
    public String login( Model model) {
        return "user/auth/login";
    }

    @GetMapping("/register")
    public String register(@ModelAttribute UserRegisterDTO user, Model model) {

        model.addAttribute("user",user);

        return "user/auth/register";
    }

    @PostMapping("/register")
    public String register(@Validated @ModelAttribute("user") UserRegisterDTO user,
        BindingResult result, Model model) {

        if (result.hasErrors()) {
            return "user/auth/register";
        }

        try{
            userService.save(User.fromUserDTO(user));
        } catch (UserAlreadyExistException uaeEx) {
            return "user/auth/register";
        }
    }
}
```

```

        return "user/auth/login";
    }
}

```

AppConfig.java:

```

package com.example.realestateads.config;

import com.google.gson.Gson;
import org.modelmapper.ModelMapper;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.JavaMailSenderImpl;

import java.util.Properties;

@Configuration
public class AppConfig {

    @Bean
    public ModelMapper modelMapper() {
        return new ModelMapper();
    }

    @Bean
    public Gson gson() {
        return new Gson();
    }

    @Value("${mail.sender}")
    private String senderUsername;

    @Value("${mail.password}")
    private String password;

    @Bean
    public JavaMailSender getJavaMailSender() {
        JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
        mailSender.setHost("smtp.gmail.com");
        mailSender.setPort(587);

        mailSender.setUsername(senderUsername);
        mailSender.setPassword(password);

        Properties props = mailSender.getJavaMailProperties();
        props.put("mail.transport.protocol", "smtp");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.debug", "true");

        return mailSender;
    }
}

```

House.java:

```
package com.example.realestateads.entity;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import org.hibernate.annotations.OnDelete;
import org.hibernate.annotations.OnDeleteAction;

@Getter
@Setter
@Entity
@Table(name = "house")
@NoArgsConstructor
@AllArgsConstructor
public class House {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Integer id;

    @OneToOne(fetch = FetchType.LAZY)
    @OnDelete(action = OnDeleteAction.CASCADE)
    @JoinColumn(name = "realty_id")
    private Realty realty;

    @Column(name = "land_area")
    private Float landArea;

    @Column(name = "floors")
    private Integer floors;
}
```

FlatRepository.java:

```
package com.example.realestateads.repository;

import com.example.realestateads.dto.FlatRequestDTO;
import com.example.realestateads.entity.Flat;
import com.example.realestateads.entity.Realty;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import java.util.Optional;

@Repository
public interface FlatRepository extends JpaRepository<Flat, Integer> {
    @Transactional
    @Modifying
    @Query("update Flat f set f.parking = ?1, f.floor = ?2, f.numberOfFloors = ?3\nwhere f.realty.id = ?4")
```

```

    void update(Boolean parking, Integer floor, Integer numberOfFloors, int
    realtyId);
    Optional<Flat> findByRealty_Id(Integer id);

    @Modifying
    @Transactional
    @Query(value = "insert into Flat(floor,number_of_floors,parking,realty_id)
    values(:#{#flat.floor},:#{#flat.numberOfFloors},:#{#flat.parking}," +
        ":realtyId)",nativeQuery = true)
    void saveFlatRequestDTO(@Param("flat") FlatRequestDTO flat, @Param("realtyId")
    int realtyId);
}

```

Login.html:

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml"
xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
<head>
    <meta charset="UTF-8">
    <title>Авторизація</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="/css/styles.css">
    <link rel="icon" href="/img/favicon.ico">
</head>
<body>
<div class="fixed-top">
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white
border-bottom box-shadow mb-3">
        <a class="navbar-brand" th:href="@{/}">Всі оголошення</a>
        <a class="navbar-brand" sec:authorize="isAuthenticated()"
th:href="@{/nedvizhimost/my-ads}">Мої оголошення</a>
        <a class="navbar-brand" th:href="@{/nedvizhimost/bookmarks}">Обрані
оголошення</a>
        <!-- <a class="navbar-brand" th:href="@{/nedvizhimost/filter}">Фільтр</a>-->
        <div class="navbar-nav">
            <form th:action="@{/auth/login}" class="nav-item">
                <input type="submit" class="btn btn-outline-primary"
value="Авторизація">
            </form>
            <form th:action="@{/auth/register}" class="nav-item">
                <input type="submit" class="btn btn-outline-secondary"
value="Реєстрація">
            </form>
        </div>
    </nav>
</div>

<div class="container">
    <div class="row justify-content-center align-items-center" style="height:
100vh;">
        <div class="col-md-6">
            <div class="text-center">
                <h1>Авторизація</h1>

```

```

        </div>
        <h6 th:if="${param.error}" class="text-center alert alert-
danger">Неправильна електронна адреса або пароль</h6>
        <form action="/auth/login" method="post">
            <div class="mb-3">
                <label for="username" class="form-label">Адреса електронної
пошти</label>
                <input type="email" class="form-control" id="username"
name="username" required aria-describedby="emailHelp">
            </div>
            <div class="mb-3">
                <label for="password" class="form-label">Пароль</label>
                <input type="password" class="form-control" name="password"
required id="password">
                <input type="checkbox" id="eyeicon" >
                <label class="form-check-label" for="eyeicon">Показати
пароль</label>
            </div>
            <div class="form-group form-check">
                <input type="checkbox" class="form-check-input" id="remember-me"
name="remember-me">
                <label for="remember-me" class="form-check-label">Запам'ятати
мене</label>
            </div>
            <div class="form-group">
                <a href="/auth/forgot-password" class="btn btn-link mb-3">Забули
пароль?</a>
            </div>
            <button type="submit" class="btn btn-primary">Авторизація</button>
        </form>
    </div>
</div>
<script>
    let eyeicon = document.getElementById("eyeicon");
    let password = document.getElementById("password");

    eyeicon.onclick = function () {
        if (password.type === "password") {
            password.type = "text";
        } else {
            password.type = "password";
        }
    }
</script>
</body>
</html>

```

UserRegisterDTO.java:

```

package com.example.realestateads.dto;

import com.example.realestateads.valid.Email;
import com.example.realestateads.valid.Password;
import com.example.realestateads.valid.PhoneNumber;
import jakarta.validation.constraints.*;
import lombok.AllArgsConstructor;
import lombok.Data;

```

```

import lombok.NoArgsConstructor;
import org.hibernate.validator.constraints.Length;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UserRegisterDTO {

    @Email(message = "Неправильний формат")
    @NotBlank(message = "Введіть електронну пошту")
    private String email;

    @NotBlank(message = "Введіть прізвище")
    private String firstname;

    @NotBlank(message = "Введіть по-батькові")
    private String patronymic;

    @NotBlank(message = "Введіть ім'я")
    private String lastname;

    @NotBlank(message = "Введіть номер телефону")
    @Length(min = 10, max = 10, message = "Номер телефону повинен містити 10
СИМВОЛІВ")
    @PhoneNumber(message = "Номер телефону повинен мати формат 0XXXXXXXXX, де X -
цифри")
    private String phoneNumber;

    @NotBlank(message = "Введіть пароль")
    @Password(message = "Пароль повинен містити латинські літери та цифри")
    @Length(min = 8, max = 15, message = "Пароль повинен мати довжину від 8 до 15
СИМВОЛІВ ВКЛЮЧНО")
    private String password;
}

```

EmailServiceImpl.java:

```

package com.example.realestateads.service.email;

import com.example.realestateads.entity.User;
import com.example.realestateads.service.dataServices.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.mail.SimpleMailMessage;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.stereotype.Component;
import org.springframework.stereotype.Service;

import java.sql.Timestamp;

@Service
public class EmailServiceImpl implements EmailService{

    @Autowired
    private JavaMailSender emailSender;

    @Autowired
    private PasswordResetUtil passwordResetUtil;
}

```



```

@Autowired
private UserService userService;

@Value("${mail.sender}")
private String senderEmail;

@Override
public void sendSimpleMessage(String to, String subject, String text) {

    SimpleMailMessage message = new SimpleMailMessage();
    message.setFrom(senderEmail);
    message.setTo(to);
    message.setSubject(subject);
    message.setText(text);
    emailSender.send(message);
}

public void sendActivationCodeToUser(User user) {

    String activationCode = passwordResetUtil.generateResetCode();
    userService.updateActivationCode(user, activationCode, new
Timestamp(System.currentTimeMillis()));
    sendSimpleMessage(user.getEmail(), "Відновлення паролю", "Код відновлення
пароллю: " + activationCode);
}
}

```

PhoneNumberValidator.java:

```

package com.example.realestateads.valid.validator;

import com.example.realestateads.valid.PhoneNumber;
import jakarta.validation.ConstraintValidator;
import jakarta.validation.ConstraintValidatorContext;

import java.util.regex.Pattern;

public class PhoneNumberValidator implements ConstraintValidator<PhoneNumber, String>
{

    private Pattern pattern = Pattern.compile("^0[0-9]{9}$");

    @Override
    public void initialize(PhoneNumber constraintAnnotation) {
    }

    @Override
    public boolean isValid(String email, ConstraintValidatorContext context) {
        return pattern.matcher(email).find();
    }
}

```

Повний код програми можна переглянути за посиланням:

<https://github.com/AlexandrTokarenko/Real-Estate-Ads>