

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента Колбасова Антона Дмитровича
(ПІБ)

академічної групи 122-17-3
(шифр)

спеціальності 122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми Комп'ютерні науки
(назва освітньої програми)

на тему: Розробка автоматизованої інформаційної системи для агентства
продажу нерухомості

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтингово ю	інституційною	
кваліфікаційної роботи	доц. Кабак Л.В.			
розділів:				
спеціальний	доц. Кабак Л.В.			
економічний	доц. Касьяненко Л.В.			
Рецензент	Каштан В.Ю.			
Нормоконтролер	доц. Гуліна І.Г.			

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних
систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » 2021 року

ЗАВДАННЯ
на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-17-3 Колбасова Антона Дмитровича
(група) (прізвище та ініціали)

тема кваліфікаційної роботи «Розробка веб-додатку для агентства
продажу нерухомості»

затверджена наказом ректора НТУ «ДП» від 07.06.2021 р. № 317-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2021 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПО й тривалості його розробки</i>	<i>27.05.2021 р.</i>

Завдання видав

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Колбасов А.Д.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2021 р.
Термін подання кваліфікаційної роботи до ЕК: 11.06.2021 р.

РЕФЕРАТ

Пояснювальна записка: 36 с., 9 рис., 1 додатків, 21 джерел.

Об'єкт розробки: інформаційна система для агентства нерухомості.

Мета дипломного проекту: мета роботи полягає у «розробці інформаційної системи для агентства нерухомості».

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми, уточняється постановка завдання.

Перший розділ присвячено аналізу предметної області, визначенню актуальності завдання та призначення інформаційної системи, розроблена постановка завдання, технологій та програмних засобів.

Другий розділ включає у себе виконання аналізу існуючих рішень, обрання платформи для розробки, проектування та розробку програми, наведення опис алгоритму і структури функціонування програми, наведення характеристики складу параметрів технічних засобів, описаний виклик та завантаження програми, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні додатку, що допоможе клієнтам та ріелторам без перешкод сортирувати та підбирати нерухомість під запити користувача. Актуальність даного програмного продукту визначається скороченням часу, який витрачається споживачам на покупку необхідних для них типів нерухомості. Також це є досить актуальною темою на сьогоднішній час через карантинний режим.

Список ключових слів: ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, MICROSOFT OFFICE EXCEL, MICROSOFT VISUAL STUDIO, C#, WINDOWS FORMS.

ABSTRACT

Explanatory note: 36 pp., 9 figs., 1 appendices, 21 sources.

Object of development: information system for real estate agency.

The purpose of the diploma project: the purpose of the work is to "develop an information system for product delivery to consumers." The introduction considers the analysis and current state of the problem, specifies the purpose of the qualification work and the scope of its application, provides a justification for the relevance of the topic, clarifies the problem.

The first section is devoted to the analysis of the subject area, determining the relevance of the task and the purpose of the information system, developed the statement of the task, technology and software.

The second section includes analysis of existing solutions, selection of a platform for development, design and development of the program, description of the algorithm and structure of the program, characteristics of the parameters of technical means, describes the call and download of the program, describes the program.

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical significance is to create an application that will help consumers and realtors to easily sort and select the property.

The relevance of this software product is determined by the reduction of time spent on consumers to purchase the property which they need. It is also a very relevant topic today due to the quarantine regime.

Keyword list: SOFTWARE, MICROSOFT OFFICE EXCEL, MICROSOFT VISUAL STUDIO, C #, WINDOWS FORMS.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

СУБД – СИСТЕМА УПРАВЛІННЯ БАЗИ ДАНИХ

PC – PERSONAL COMPUTER

JS – JAVA SCRIPT

ТП – ТЕХНІЧНА ПІДТРИМКА

ПЗ – ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

VSC – VISUAL STUDIO CODE

PHP - HYPERTEXT PREPROCESSOR

SQL - SCRUCTURED QUERY LANGUAGE

HTML - HYPERTEXT MARKUP LANGUAGE

CSS - CASCADING STYLE SHEETS

ЗМІСТ

РЕФЕРАТ	3
АВСТРАКТ	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1. Загальні відомості з предметної галузі	10
1.2. Призначення розробки та галузь застосування.....	12
1.3. Підстава для розробки	13
1.4. Постановка завдання.....	13
1.5. Вимоги до програми або програмного виробу	15
1.5.1. Вимоги до функціональних характеристик.....	15
1.5.2. Вимоги до інформаційної безпеки	16
1.5.3. Вимоги до складу та параметрів технічних засобів	16
1.5.4. Вимоги до інформаційної та програмної сумісності.....	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	18
2.1. Функціональне призначення програми.....	18
2.2. Опис застосованих математичних методів.....	19
2.3. Опис використаних технологій та мов програмування.....	19
2.4. Опис структури системи та алгоритмів її функціонування.....	20
2.5. Обґрунтування та організація вхідних та вихідних даних програми	22
.....
2.6. Опис розробленої системи	23
2.6.1. Використані технічні засоби.....	23
2.6.2. Використані програмні засоби.....	23
.....
2.6.3. Виклик та завантаження програми.....	23
2.6.4. Опис інтерфейсу користувача.....	24
.....
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	31
3.1. Визначення трудомісткості розробки програмного забезпечення	31
3.2. Витрати на створення програмного забезпечення.....	33
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37

ВСТУП

У наш час дуже швидко розвивається ринок нерухомості. Так, як рівень життя зростає порівняно з минулим, зростає і кількість населення. Люди тяжіють до забезпечення себе основними засобами комфортного життя, одним з яких є житло. У 2018 році по всій Україні було укладено 277330 договорів купівлі-продажу жилої нерухомості, що дорівнює приблизно 760 договорам щоденно. Будь-то купівля земельної ділянки, приватного будинку чи квартири – це завжди відповідальний та трудомісткий процес. Саме тому, деякі люди віддають перевагу ріелторським організаціям для продажу чи купівлі нерухомості. Наявність вичерпної бази і систем підбору найкращого варіанту з представлених, грає вирішальну роль у налагодженні довіри клієнтів до ріелтора та фірми.

Актуальність проекту полягає у створенні каналу зв'язку між співробітниками агентства нерухомості та її клієнтами, що є необхідної складовою ефективної співпраці. Система збереження, обробки та редагування інформації знизить робочий час для пошуку серед тисяч об'єктів потрібного, що дозволяє знизити завантаженість ріелторів та підвищити якість підбору пропозицій для продажу.

Об'єктом роботи є інформаційна підтримка процесу діяльності агентства нерухомості.

Предметом дослідження є засоби забезпечення інформаційної підтримки ріелторської діяльності.

Метою роботи є розробка інформаційної системи підтримки діяльності агентства продажу нерухомості.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- провести аналіз предметної області;
- проаналізувати існуючі системи підтримки продажу нерухомості;
- встановити головні вимоги до інформаційної системи та обрати засоби реалізації;

- прискорення розвитку більш інноваційних систем для споживачів;

- розробити систему;
- провести тестування.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

З упевненістю можна заявити, що дана сфера діяльності людей стала невід'ємною частиною їхнього життя. Велика частина ріелторських агентств створила свої власні сайти з пошуку нерухомості, а також курирує окремою групою людей, яка технічно підтримує сайт і підбирає нерухомість замовникам. Також, не менш важливим пунктом є реакція ринку на глобальну пандемію, так як це питання актуальне в усьому світі вже другий рік. Домінантними представниками на ринку послуг ріелторських агентств є: Ozimka, СПД Михайлова, ЖК Оптимисто, DaSardinia, Алеанза.

Наразі глобальний ринок неможливо собі уявити без сучасних технологій, які полегшують нам життя. На жаль, у нас в Україні онлайн-послуги тільки почали завойовувати свою популярність. Мировий ринок послуг з пошуку нерухомості, що розглядається в цьому дипломі, розділений за типами на платформа (додаток) - постачання клієнту, агентство – постачання клієнту. За типом розподілу на класи, пошук нерухомості може бути відображено у вигляді веб-сайту та у вигляді мобільного додатку.

Також, якщо судити з точки зору економіки, то потреба в підборі нерухомості, тобто ріелтори, буде основним стримуючим фактором розвитку для ринку

онлайн-послуг з підбору нерухомості, але ні в якому разі не з технічної причини. Користувачі смартфонів є основними шукачами в індустрії з купівлі нерухомості, тому має сенс робити не сайт, а платформу або додаток на якому користувачі зможуть спокійно обирати товари, що їм до вподоби. Але як вже було раніше сказано, сайти є більш функціонуючим ланцюжком у сфері пошуку житла, ніж додатки, тобто в них є більше переваг. Також, збільшення числа користувачів телефонів відображає збільшення кількості пошуку і купівлі житла в Інтернеті. Глобальна кількість підписників електронної комерції досягла вже 1,5 мільярда на 2019 рік та судячи зі статистики, буде зростати ще на 800 мільйонів кожен рік. Отже, зростання кількості користувачів гаджетів і проникнення їх до Інтернету стимулює зростання онлайн-сервісів пошуку нерухомості.

Однак, перш ніж розробляти платформу чи сайт агентства нерухомості, важливо зрозуміти, який саме алгоритм дій повинен працювати. Можна взяти на прикладі будь-який додаток з пошуку нерухомості, проаналізувати його та почати власне виробництво. По-перше, ми бачимо інтерфейс, а тільки потім аналізуємо об'єм роботи, яку потрібно буде виконати.

Такі програми або сайти, що приймають замовлення на пошук нерухомості, обробляють замовлення та підбирають варіанти. Додаток надає користувачам безліч функцій, щоб зробити процеси підбору максимально зручними для всіх. Деякі з примітних розповсюджених функцій включають в себе наступне:

- Рекомендації типів нерухомості від ріелторів для споживачів.
- Фільтри пошуку категорій нерухомості по обраним критеріям користувача (наприклад, житлова нерухомість, замиська нерухомість, комерційна нерухомість тощо).
- Налаштовуемі деталі пошуку (мається на увазі готівкова плата за замовлення або безготівкова, тобто карткою у додатку чи на сайті).
- Відстеження виконання завдання є також досить важливим пунктом.

- Яскравий та інтересний інтерфейс також є невід'ємною частиною майбутнього сайту або додатку.
- Зручний функціонал. Цей пункт є основним, тому що нагромадження функцій може знижувати ефективність функціоналу вашого сайту або додатку. Мінімальна кількість взаємодій.

Також, якщо порівнювати платформу з сайтів, то сайт є більш вигідним аналогом додатку. Додаток є досить затратним не тільки з точки зору фінансів та піар кампаній але й затраченого часу на його написання та розповсюдження. Сайт є більш зручним не тільки для користувачів але й для власників агентств з продажу нерухомості.

У деяких аспектах агентств нерухомості можна сказати, що технічні протиріччя є мінімальним для сайтів та додатків. Технічні протиріччя – це така ситуація, при якій спроба покращити одну з характеристик об'єкта, погіршується інша його характеристика або показник. Між цими показниками існує співвідношення, яке й називають технічним протиріччям. Але у сайтів для пошуку нерухомості ці протиріччя невеликі, через простий дизайн та стислий функціонал.

1.2. Призначення розробки та галузь застосування

Призначення розробки та галузь застосування – це застосування інформаційної системи для агентства продажу нерухомості. На разі, така СУБД є досить актуальною у наш час та може використовуватись як з РС, так і з ваших смартфонів. Ця система застосовується для сортирування, розподілення об'єктів нерухомості. По-перше, будь-який користувач сайту або платформи буде спроможний викростати таку систему з будь-якої точки країни, або, навіть світу. По-друге, для використання такої системи, не потрібно мати потужний гаджет через те, що функціонал дуже мінімальний.

Система з розподілення нерухомості може використовуватись не тільки серед звичайних користувачів, але й мати спеціальне призначення. Наприклад, такі системи можуть використовуватись в Комітеті з питань прав людини, деокупації та реінтеграції тимчасово окупованих територій, національних меншин і міжнаціональних відносин для розподілення об'єктів нерухомості між переселенцями серед невеликих груп людей.

1.3. Підстава для розробки

Відповідно до ОКХ та ОПП, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою та затверджується наказом ректора.

Отже підставами для розробки (виконання кваліфікаційної роботи) є:

- ОКХ та ОПП за напрямом підготовки «Комп'ютерні науки»;
- Навчальний план та графік навчального процесу;
- Наказ ректора Національного технічного університету «Дніпровська політехніка» №317-с від 07.06.2021 року;
- Завдання на дипломний проект на тему: «Розробка інформаційної системи для доставки продуктів споживачам».

1.4. Постановка завдання

Метою даної кваліфікаційної роботи є створення інформаційної системи для агентства продажу нерухомості.

Система для агентства продажу нерухомості повинна реалізувати наступні показники:

- Мета дипломного проектування – це створення інформаційної системи для підтримки інформаційних процесів, які супроводжують процес продажу житлових приміщень. Основне завдання полягає в написанні функціонального простого сайту на мові програмування JS.
- Інформаційна система для агентства продажу нерухомості може бути використана як зі сторони незалежних розробників, які не мають власної компанії, також зі сторони таких закладів, як ріелторські компанії. Інформаційна система буде розроблена і призначена спеціально як для замовників, так і для користувачів. Однак, якщо ріелторські компанії не мають власного сайту, вони можуть скористатися послугами таких платформ за додаткову комісію або платню.

Впровадження комп'ютеризованих технологій у сферу діяльності продажу має на меті полегшити організацію роботи співробітників з клієнтами. Надання клієнтам доступу до системи дозволяє виключити ріелтора агентства з процесу внесення змін до побажання клієнта, тим самим, усунення посередництва зберігає час, який потрібний для забезпечення якісного надання послуг клієнтам.

- По-перше, сайт повинен складатися з яскравого та зрозумілого інтерфейсу, перед користувачем повинні бути основні функції сайту на головній сторінці. Також, важливе значення має коректне відображення інтерфейсу, як на телефоні, так і на РС. По-друге, можна відстежувати стан додатку або сайту судячи з отриманих завдань.
- Чіткий правильний структурований код, який дозволяє приймати інформацію від користувача, тобто завдань, обробляти цю інформацію, тобто фіксувати замовлення та відсилати до агентств нерухомості. Обробка замовлень повинна бути у порядку живої черги, тобто обробляти замовлення на пошук нерухомості від одного користувача до останнього, але робити це без жодних затримок, тобто автоматично коригувати неправильні замовлення. Також є важливим факт контролю замовлень, які надходять до сайту або платформи.

- У разі, якщо код був написаний структуровано невірно, припиняється автоматична обробка замовлень на сайті або платформі. Сайт є більш гнучким матеріалом, тому зафіксувати та виправити помилки легше, ніж зробити це на платформі. Тому, у такому разі, деякі з платформ мають навіть власну технічну підтримку, яка буде обробляти запити від користувачів, які мають індивідуальні збої в програмі.
- Така інформаційна система є досить гнучкою, тому вона може бути пов'язана наприклад з такими задачами, як розподіл замовлень індивідуально кожному з вільних ріелторів. Як вже було сказано раніше, система може розподіляти між собою замовлення, які йдуть окремим ріелторам. Також, треба додати функцію написання листа до ТП, такі листи будуть автоматично сортуватися по темі написання та надходити до відповідних відділів, які займаються конкретними справами.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Вимоги до сайту полягають в обробці, розподіленню, контролю інформації, яка надходить до нього від користувачів. Головне меню користувача або замовника повинно складатися лише з корисних та діючих опцій, що відповідають ходу їх процесу. У них повинні бути наступні варіанти:

- Реєстрація на сайті або в додатку.
- Список ресторанів або продуктових магазинів, які також можуть займатись доставкою. Також повинні бути фільтри, які допомагають користувачеві здійснити вибір нерухомості, які йому до смаку.
- Декілька варіантів платежів, тобто готівковий платіж або безготівковий.
- Пропозиції та компейн промокоди.
- Відстеження в реальному часі і відомості про відповідальне за доставку.

1.5.2 Вимоги до інформаційної безпеки

Для захисту ПЗ в першу чергу необхідно обмежити доступ до коду усіх, окрім системних адміністраторів та технічної підтримки, яка також працює над виправленням помилок. Також, у кожного користувача сайту або додатку буде свій власний унікальний ідентифікаційний код, який допоможе захистити облікові записи.

1.5.3 Вимоги до складу та параметрів технічних засобів

Почати треба з того, що сайту може працювати на будь-якій операційній системі, тобто:

- Windows 10
- Windows 7
- Windows 8
- Linux
- Unix

Для коректного відображення сайту потрібно мати один з перелічених браузерів:

- Internet Explorer 8.0.0
- Opera 5.0
- Google Chrome 10.0
- Mozilla Firefox 13.0

Мінімальні системні вимоги:

- Процесор класу Intel Core
- Не менше 2 ГБ ОЗУ
- Вільного місця на жорсткому диску приблизно від 100 МБ.

Вимоги до обладнання:

- Монітор

- Миша
- Навушники
- Працюючий РС або ноутбук.

1.5.4 Вимоги до інформаційної та програмної сумісності

Технічні вимоги до комп'ютера, який відповідає мінімальним системним вимогам для розробки і розгортання створюваної програми, інсталяції інструментальних засобів розробки, що враховує особливості експлуатації програмного забезпечення, приведені для всіх використовуваних програмних засобів. У такому випадку для розробки сайту буде використовуватись Visual Studio Code. Також для розробки інформаційної системи будуть використанні такі мови програмування:

- Java Script
- PHP
- SQL
- HTML
- CSS

Також, написання коду може бути на будь-якій операційній системі, наприклад:

- Windows 10
- Windows 7
- Windows 8
- Linux
- Unix

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Функціональне призначення програми складається з автоматизації інформаційної системи для агентства продажу нерухомості. Функціонал даної програми хоч і обмежений, однак, такі функції як вхід в обліковий запис адміністратора, клієнта, ріелтора і реєстрація разом з збереженнями даних, передбачена.

- Обліковий запис адміністратора дозволяє нам додавати потрібний тип нерухомості, вивантажувати картинки, опис до типу нерухомості, кількість одиниць нерухомості, змінювати написи і оформлення сторінки і сайти в режимі реального часу.
- Обліковий запис ріелтора дозволяє додавати тип нерухомості, які потрібно виконати, в свою власну чергу, щоб інші ріелтори не змогли повторно взяти той же тип нерухомості, також профіль ріелтора має додаткову панель на якій можна відстежувати виконані або поточні завдання. Також ріелтор може переглядати всі свої виконані завдання за останній час.
- Обліковий запис клієнта дозволяє заходити в додаток і переглядати вивантажені типи нерухомості на сайті, вибрати потрібний тип виходячи з категорій, також додатково можна вибрати тип нерухомості, який клієнт хоче розглянути.

Функціонал сайту і платформи достатньо простий, але ефективний, так як в роботі він ідеально підійде з точки зору замовника, так як є функції, які дозволяють редагувати сайт в теперішньому часі, так і з точки зору споживача.

Дана платформа забезпечує необхідними ресурсами не тільки замовника і споживача, але і виконавців теж, тобто ріелторів.

2.2. Опис застосованих математичних методів

Оскільки особливості предметної області інформаційної системи, що проектується, не передбачають застосування математичних методів, при розробці інформаційної системи для агентства нерухомості математичні методи не використовувались.

2.3. Опис використаних технологій та мов програмування

Для створення структури всієї програми використовувалося інтегроване середовище розробки програмного забезпечення Microsoft Visual Studio, а також такі мови програмування як: JS, PHP, SQL, HTML і CSS. Також нам варто позначити кожну мову програмування окремо:

- JS (JavaScript) - це мова програмування, який використовується в першу чергу для більш складних цілей в документі HTML. JS - це мова, яка дозволяє змінювати інтерфейси веб додатки, додавати або змінювати різні функціональні можливості сайту. Сторонні API можуть дозволити нам додавати різні функції, які вже існують на інших сайтах.

Ви можете застосувати до вашого HTML документу сторонні фреймворки і бібліотеки, що дозволить вам прискорити створення сайтів і додатків.

- PHP (Hypertext Preprocessor) - це мова програмування, який потрібен для статистичного відображення сторінки, а також для "довголіття" вашого сайту. Тобто, дана мова використовується для щоденного оновлення

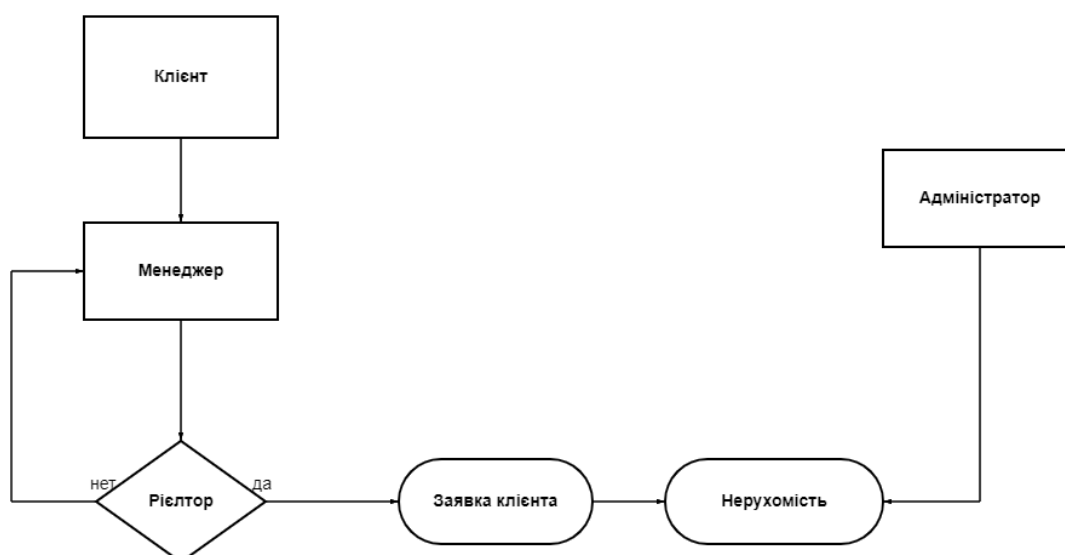
сайту, надходження нової свіжої інформації, яка буде актуальна кожен день для користувачів. Дана мова програмування використовується для написання "сценаріїв", які завантажуються в HTML документ, щоб відобразити нестатичність картинки. Справа в тому, що документ HTML здатний відобразити тільки статичну інформацію на сайті, а PHP показує "свіжу".

- SQL (Structured Query Language) - це мова програмування, яка може використовуватися як ефективний спосіб для таких функцій, як: збереження в БД, пошуку їх частин, поновлення, вилучення з бази даних і видалення звітти. Взаємодія з БД відбувається швидко навіть в ситуаціях, коли обсяги даних при завантаженні досить великі. SQL – це досить зручна мова для великої кількості інформації, так як дозволяє миттєво розробнику перерозподіляти інформацію.
- HTML (HyperText Markup Language) - це мова програмування, яка дозволяє розробнику створювати і структурувати безліч розділів, заголовків, посилань і цілі блоки для веб-сторінок і додатків. Це мова, яка дозволяє створювати структуру web-сторінок.
- CSS (Cascading Style Sheets) - це скоріше не мова програмування, а правила оформлення HTML документа. CSS дозволяє розробнику встановлювати розмітку сторінки, розмір шрифтів, колір, заголовки, а також змінювати шрифт. Можна сказати, що CSS безпосередньо відповідає за візуальну складову сайту, так як жоден HTML документ не може обійтися без CSS.

2.4. Опис структури системи та алгоритмів її функціонування

А) Алгоритм заданої системи полягає в тому, що кожен з користувачів взаємопов'язаний один з одним. Наприклад, адміністратор завантажує на сервер типи нерухомості з обмеженою кількістю, поміщаючи в окрему групу нерухомості. Користувач бачить, що новий тип був доданий в обмеженою кількості і виходячи зі своїх потреб, вибирає тип нерухомості з тієї чи іншої категорії. Тобто користувачем здійснюється вибір типу нерухомості, далі дане замовлення формується і надходить в базу даних ріелторів, між ними дані обраного покупцем типу нерухомості розподіляються. Тобто, кожен ріелтор вибирає собі замовлення виходячи з типу нерухомості.

Наочний приклад одного з алгоритмів можна побачити в даному малюнку:



1.2. Алгоритм системи

Також дана система спрямована на те, щоб алгоритм виконувався у виключній послідовності, тобто так, щоб не виникало збоїв, які заважають циклу повторення алгоритму.

1.3. Структура бази даних

В) Структура системи складається з сполучних компонентів, які виражені у вигляді облікових записів. У даній інформаційній системі кожен користувач нерозривно пов'язаний з іншим. Наприклад, ріелтор пов'язаний з клієнтом, а клієнт з ріелтором і адміністратором, в свою чергу адміністратор пов'язаний з клієнтом і ріелтором. Таким чином дані користувачі утворюють цикл, тобто періодичність повторюваних дій. Клієнт подає заявку, заявка утворюється на сайті і приймається ріелтором, адміністратор відповідає за вивантаження нерухомості і за коригування на сайті.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Дані користувача збираються в формі у користувача, відправляються на сервер і далі обробляються (незалежно від типу користувача); також по заздалегідь створеному шаблону виводяться всі типи нерухомості і користувач може їх вибрати в потрібній кількості і розглядати, під час вибору типу нерухомості відбувається запис в базу даних, тобто всі типи нерухомості, які були обрані користувачем, не будуть загублені, навіть при зміні пристрою; потім створюється заявка, яка передається вільному ріелтору.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для написання даної інформаційної системи використовувався персональний комп'ютер, мишка, клавіатура і гарнітура.

2.6.2. Використані програмні засоби

Створення даної інформаційної системи було створено на операційній системі Windows 10.

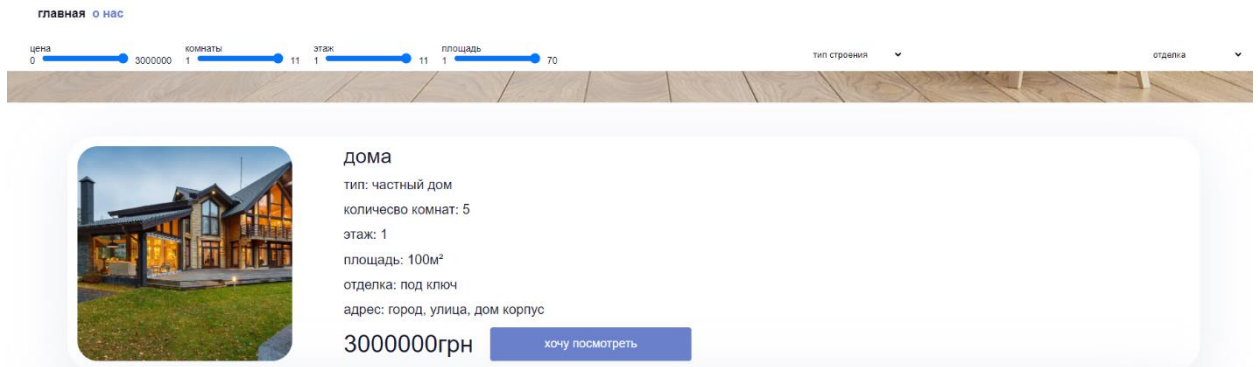
Також для розробки даної інформаційної системи використовувалася платформа Visual Studio Code.

2.6.3. Виклик та завантаження програми

Програма з розподілу нерухомості може бути викликана як через накопичувач, тобто Флейком або диск, так і через перехід по посиланню. Тобто, поширення посилання відбувається безпосередньо через браузер, який використовується користувачами. Дана програма також може бути завантажена на платформу через яку буде виконуватися вхід в мобільний додаток через яке можна буде здійснити вхід. Також програму можна завантажити з браузера Google Chrome, так як була додана подібна опція.

2.6.4. Опис інтерфейсу користувача

Перше, що бачить користувач, коли заходить на веб сайт - це панель на якій можна увійти в свою обліковий запис або зареєструватися на сайті:



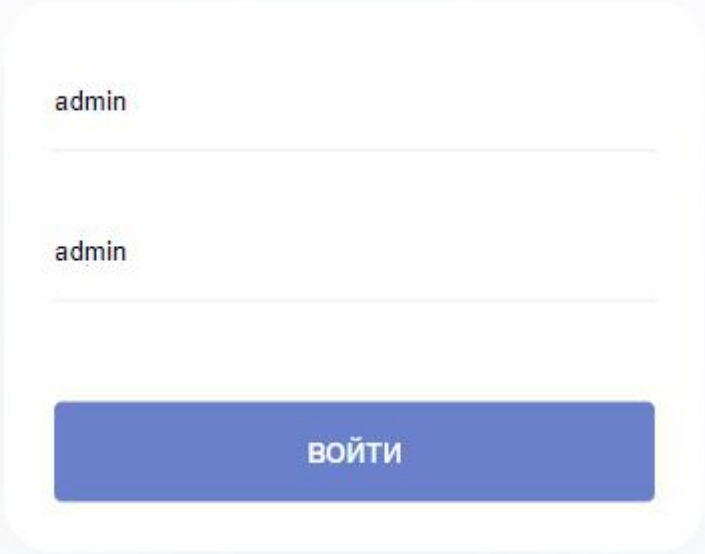
1.4. Початкове меню

У вкладці реєстрації нам доступні стандартні дані користувача:

The image shows a registration form with three input fields and a submit button. The first field is labeled "customer" and contains the text "customer". The second field contains the phone number "0668364646". The third field is labeled "почта". Below the fields is a blue button labeled "отправить".

1.5. Меню реєстрації

Також користувачам доступні певні функції в залежності від їх акаунтів:



The image shows a login form with two input fields. The first field contains the text 'admin'. The second field also contains the text 'admin'. Below the input fields is a blue button with the text 'ВОЙТИ' (Log In) in white capital letters. The form is centered on a light gray background.

1.6.

Акаунт адміністратора

[главная](#) [о нас](#)


выйти

добавить недвижимость

удалить недвижимость

Активные заявки:

имя: 1 телефон: 1 почта: 1



тип: лофт
количество комнат: 11
этаж: 11
площадь: 11м²
отделка: под ключ
адрес: 11, 11, 11


11грн

завершить заявку

1.7. Акаунт менеджера

[главная](#) [о нас](#)

цена 0 11 комнаты 1 11 этаж 1 11 площадь 1 11 тип строения отделка



11

тип: лофт
количество комнат: 11
этаж: 11
площадь: 11м²
отделка: под ключ
адрес: 11, 11, 11

11грн хочу посмотреть

Property Estate
менеджерам

Данный сайт носит информационно-справочный характер и ни при каких условиях не является публичной офертой

1.8. Акаунт покупателя

Коли ми реєструємося і входимо в обліковий запис, як споживач, перше, що ми можемо помітити - це вікно, яке пропонує нам усі товари на вибір:



ДОМА

тип: частный дом
количество комнат: 5
этаж: 1
площадь: 100м²
отделка: под ключ
адрес: город, улица, дом корпус

3000000грн

[хочу посмотреть](#)



11

тип: лофт
количество комнат: 11
этаж: 11
площадь: 11м²
отделка: под ключ
адрес: 11, 11, 11

11грн

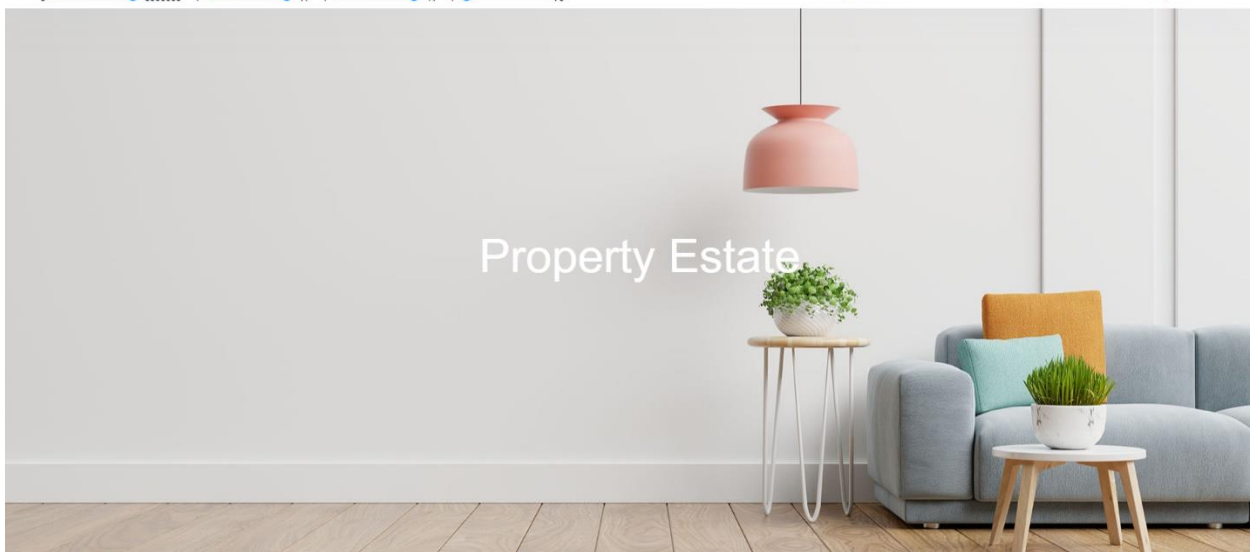
[хочу посмотреть](#)

[главная](#) [о нас](#)

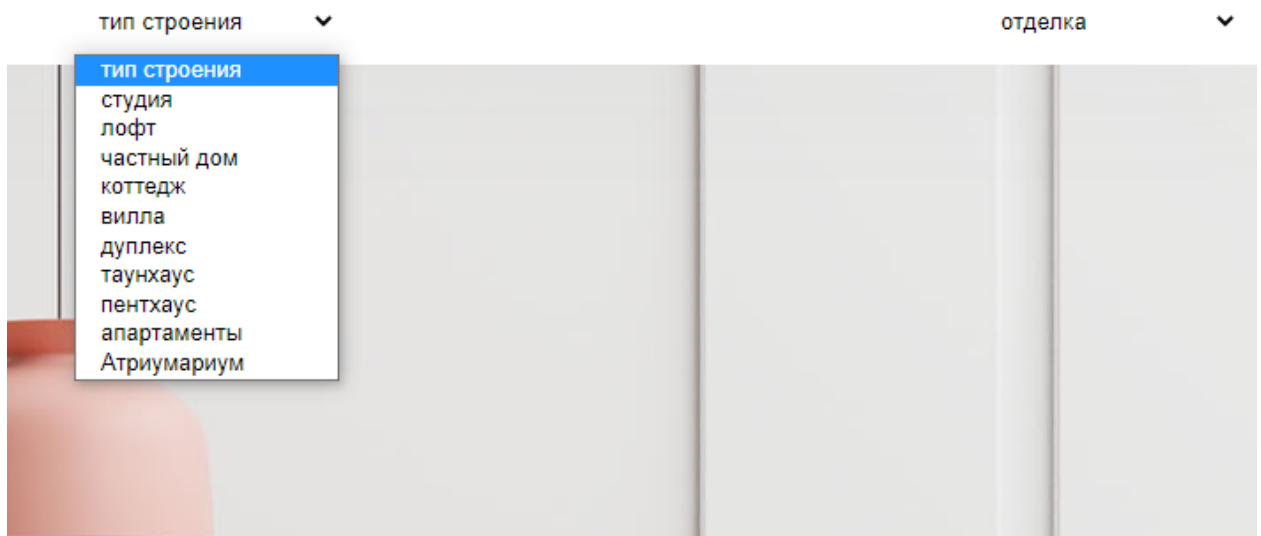
цена 0 3000000 комнаты 1 11 этаж 1 11 площадь 1 70

тип строения

отделка

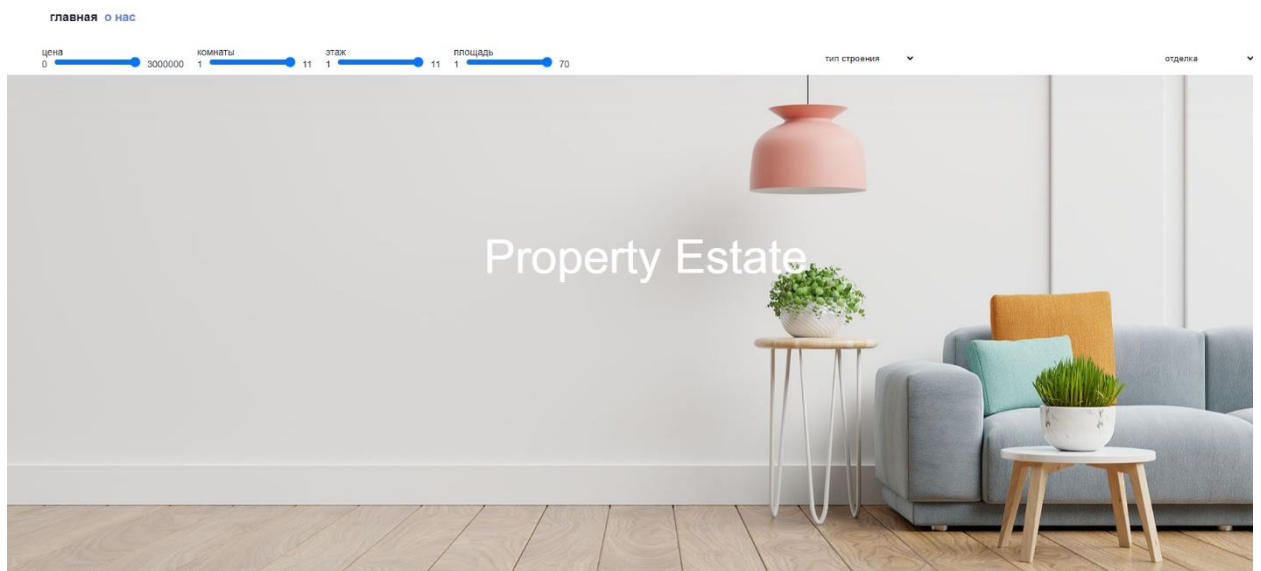


1.9. Початкове меню додатку



2.0. Меню

Є також розділ профілю, який дозволяє користувачеві змінювати необхідні йому дані. Зверху зліва також є кнопка виходу з облікового запису, після виходу з облікового запису користувач автоматично потрапляє в початкове меню веб сайту:



2.1. Меню профіля

Через аккаунт адміністратора можна впливати на всіх користувачів, які підключені до сервера, тобто можна додавати інших користувачів, а також видаляти.

Також існує вкладка, яка дозволяє додавати товар користувачам, вказуючи назву, вагу в грамах, вартість продукту, а також кількість товарів, також можна вказувати фільтри і завантажувати картинку для відображення товару:

2.2. Меню додатку нерухомості для споживачів

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Визначення трудомісткості розробки програмного забезпечення

При розробці програмного забезпечення (ПЗ) важливими етапами є визначення трудомісткості розробки і розрахунок витрат на створення програмного продукту.

Задані дані:

- 1) Передбачуване число операторів – 600;
- 2) Коефіцієнт складності програми – 1,5;
- 3) Коефіцієнт корекції програми в ході її розробки – 0,1;
- 4) Годинна заробітна плата програміста, грн/год – 250;
- 5) Коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,1;
- 6) Коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,0;

7) Вартість машино-години ЕОМ, грн/год – 45.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки може бути розрахована на основі системи моделей з різною точністю оцінки. Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_\delta, \text{ людино-годин.} \quad (1.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

t_{oml} - витрати праці на налагодження програми на ЕОМ;

t_δ - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (1.2)$$

Q - передбачуване число операторів;

C - коефіцієнт складності програми;

P - коефіцієнт кореляції програми в ході її розробки.

Звідси умовне число операторів в програмі:

$$Q = 1,5 \cdot 600 \cdot (1 + 0,1) = 990$$

Витрати праці на вивчення опису задачі та визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = Q * B(75..85) * k, \text{ людино-годин.} \quad (1.3)$$

B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;
k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

$$t_u = \frac{(990 \cdot 1,2)}{(75 \cdot 1,2)} = 13,2 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a \cdot = Q / (20..25) * k, \text{ людино-годин.} \quad (1.4)$$

Витрати на складання програми по готовій блок-схемі:

$$t_n \cdot = Q / (20..25) * k, \text{ людино-годин.} \quad (1.5)$$

$$t_a = \frac{1165}{(20 \cdot 1,2)} = 49 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{отл} \cdot = Q / (4..5) * k, \text{ людино-годин.} \quad (1.6)$$

- за умови комплексного налагодження завдання:

$$T_{к отл} = 1,5 * t_{отл}, \text{ людино-годин.} \quad (1.7)$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,} \quad (1.8)$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{\partial p} = Q / (15..20) * k, \text{ людино-годин} \quad (1.9)$$

$t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації.

$$T_k \partial o = 0,75 * t_{\partial p}, \text{ людино-годин.} \quad (1.10)$$

дставляючи відповідні значення, отримаємо:

$$t_{\partial p} = \frac{990}{(18 \cdot 1,2)} = 45 \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 316 = 41 \text{ людино-годин.}$$

$$t_{\partial} = 45 + 41 = 86 \text{ людино-годин.}$$

$$t = 50 + 22 + 49 + 58 + 194 + 86 = 459 \text{ людино-годин.}$$

3.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{no} = 3 z_n + 3 m_v, \text{ грн.} \quad (1.11)$$

Заробітна плата виконавців визначається за формулою:

$$З_{зп} = t \cdot C_{пр}, \text{ грн.} \quad (1.12)$$

$$З_{зп} = 459 \cdot 250 = 114,750 \text{ грн.}$$

t - загальна трудомісткість, людино-годин;

$C_{пр}$ - середня годинна заробітна плата програміста, грн/година.

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \cdot C_{мч}, \text{ грн.} \quad (1.13)$$

$T_{отл}$ - трудомісткість налагодження програми на ЕОМ, год.

$C_{мч}$ - вартість машино-години ЕОМ, грн/год.

$$З_{мв} = 194 \cdot 45 = 8730 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{по} = 114,750 + 8,730 = 123,480 \text{ грн.}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = t / B_k * F_p, \text{ міс.}$$

$$(1.14)$$

B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідси витрати на створення програмного продукту:

$$T = \frac{459}{(1 \cdot 176)} \approx 2,5 \text{ міс.}$$

ВИСНОВКИ

Метою дипломного проекту є розробка програмного забезпечення для підбору нерухомості для клієнтів, а також для користування додатком агентствами нерухомості чи власного агентства. Програма являє собою систему, що реалізує запис інформації про замовлення користувачів, обробку та зберігання даних, відображення інформації у зрозумілій впорядкованій формі.

Мета даної дипломної роботи складалася з декількох частин, наприклад:

- 1) Написання плану дій дипломної роботи.
- 2) Здійснення задуманої концепції, яка відображається коректно, а також має подальше застосування в безпосередньо важливих сферах життя людини. Також додавання декількох функцій в процесі роботи, наприклад, таких як:

зміна облікових даних, додавання або зміна будь-яких продуктів, включаючи їжу, але і також товари щоденного користування.

Тема дипломної роботи є актуальною на сьогоднішній день, а також, провівши певні аналітичні спостереження за розвитком сфери діяльності людини по пошуку нерухомості, дана тема набирає свою актуальність і популярність. Існують емпіричні дані згідно з якими послуги з підбору нерухомості будуть приносити більший прибуток в найближчому майбутньому. Агентство нерухомості хоч і не є новою і унікальною ідеєю, однак, допускає безліч нових поправок, а також коригувань і фільтрів, які дійсно можуть зробити продукт більш унікальним і універсальним, згідно смакам користувачів.

З практичної точки зору, агентство нерухомості буде займати в майбутньому лідируючі позиції, так як велика частина людей шукає нерухомість.

Також, дана програма не є дорогою в плані її підтримки, проте, якщо пізніше враховувати кількість витрат на підтримку програми, вони можуть зрости. У додатку є кілька пунктів економічного застосування, тобто воно буде застосовуватися від імені незалежної компанії, або ж використовуватися агентствами нерухомості. В кожному окремому випадку, даний вид додатка буде приносити величезний прибуток, вимагаючи мінімальні витрати. Використання системи дозволяє скоротити час на пошук потрібного варіанту нерухомості серед тисяч доступних для надання якісних послуг.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Джон Джеккет, Javascript и jQuery. Интерактивная веб-разработка, URL: <https://monster-book.com/javascript-i-jquery-interaktivnaya-veb-razrabotka>
2. Mega Realtor. Электронный ресурс. URL: <http://mega-realtor.com.ua/>
3. Стефанов С, 124с. 75с. 89с, javascript. Шаблоны, URL: <https://ru.pdfdrive.com/javascript-%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D1%8B-e186422699.html>
4. Лоусон Б. Шарп Р, 68с, Изучаем HTML 5. Библиотека специалиста, URL: <http://padabum.com/d.php?id=41242>
5. П. Лабберс - HTML 5 для профессионалов URL: https://proklondike.net/books/html/labbers_html5.html
6. Дронов В. А. - HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов, URL:

- https://proklondike.net/books/webdesign/dronov_html5_css3_web20_2011.html
7. Бен Хеник - HTML и CSS Путь к совершенству, URL: https://proklondike.net/books/html/ben_henik_HTML_CSS_2011.html
 8. Мэтт Стаффер, Laravel. Полное руководство, URL: <https://monster-book.com/laravel-polnoe-rukovodstvo>
 9. Дмитрий Котеров, PHP 7 в подлиннике, URL: <https://monster-book.com/php-7-v-podlinnike>
 10. Игорь Симдянов, Максим Кузнецов, Самоучитель PHP 7, URL: <https://monster-book.com/samouchitel-php-7>
 11. Бретт Маклафлин, PHP и MySQL. Исчерпывающее руководство, URL: <https://monster-book.com/php-i-mysql-ischerpyvayushchee-rukovodstvo>
 12. Яков Файн, Антон Моисеев, Angular и TypeScript. Сайтостроение для профессионалов, URL: <https://monster-book.com/angular-i-typescript>
 13. Джон Дакетт, HTML и CSS. Разработка и дизайн веб-сайтов, URL: <https://monster-book.com/html-i-css-razrabotka-i-dizayn-veb-saytov>
 14. Александр Чипижко, Прибыльная веб-студия. Пошаговое руководство, URL: <https://monster-book.com/pribylnaya-veb-studiya-chipizhko>
 15. Кайл Симпсон, Типы и грамматические конструкции {Вы не знаете JS}, URL: <https://monster-book.com/typy-i-grammaticheskie-konstrukcii-js>
 16. Дейт К.Дж. Введение в системы баз данных 8-ое издание. — М.: Вильямс, 2005. — 1328 с., URL: <https://www.twirpx.com/file/196117/>
 17. Джим Р. Уилсон, Эрик Редмонд, Семь баз данных за семь недель, URL: <https://monster-book.com/7-baz-dannyh-za-7-nedel>
 18. Евгений Моргунов PostgreSQL. Основы языка SQL, URL: <https://monster-book.com/postgresql-osnovy-yazyka-sql>
 19. Томас Коннолли, Каролин Бегг, Базы данных. Проектирование, реализация и сопровождение, URL: <https://monster-book.com/bazy-dannyh-proektirovanie-realizaciya-i-soprovozhdenie>

20. Джон Резиг, знания основ JavaScript. URL: <http://null-book.com/?rout=article&article=941>
21. Мартин Фаулер, Рефакторинг кода на JavaScript: улучшение проекта существующего кода, URL: <https://revall.info/refactoring-koda-na-javascript-uluchshenie-proekta-sushhestvuyushhego-koda.html>

КОД ПРОГРАМИ

main.js

```
import { createApp } from 'vue'
import App from './App.vue'
import './registerServiceWorker'
import router from './router'
import store from './store'
import axios from 'axios'
import VueAxios from 'vue-axios'

createApp(App).use(VueAxios, axios).use(store).use(router).mount('#app')
```

app.vue

```
<template>
  <div id="nav">
    <router-link to="/">главная</router-link>
    <router-link to="/about">о нас</router-link>
  </div>
  <div class="content">
    <router-view/>
  </div>
  <div class="footer">
    <div>
      <p class="logo">property estate</p>
      <router-link to="/admin">менеджерам</router-link>
    </div>
    <p class="footer__description">Данный сайт носит информационно-справочный характер и ни при каких
    условиях не является публичной офертой</p>
  </div>
</template>

<style lang="scss" >
*{padding: 0;margin: 0;border: 0; scroll-behavior: smooth;}
*,*:before,*:after{-moz-box-sizing: border-box;-webkit-box-sizing: border-box;box-sizing: border-box;}
:focus, :active{outline: none;}
a:focus,a:active{outline: none;}
nav,footer,header,aside{display: block;}
html,body{height:100%;width:100%;font-size:100%;line-height:1;font-size:14px;-ms-text-size-adjust:100%;-moz-
text-size-adjust:100%;-webkit-text-size-adjust:100%;}
input,button,textarea{font-family:inherit;}
input::-ms-clear{display: none;}
button{cursor: pointer;user-select: none;}
button::-moz-focus-inner{padding:0;border:0;}
a,a:visited{text-decoration: none;}
```

```

a:hover{text-decoration: none;}
ul li{list-style: none;}
img{vertical-align: top;}
h1,h2,h3,h4,h5,h6{font-size:inherit;font-weight: inherit;}
@media (min-width: 500px) {
  ::-webkit-scrollbar{
    width: 10px;
  }
  ::-webkit-scrollbar-track{
    border-radius: 100px;
    background-clip: border-box;
  }
  ::-webkit-scrollbar-thumb{
    background: rgba(0, 0, 0, 0.2);
    background-clip: padding-box;
    border: 4.5px solid rgba(0, 0, 0, 0.0);
    border-radius: 100px;
  }
  ::-webkit-scrollbar-track-piece{
    background: rgba(0, 0, 0, 0.0);
  }
  ::-webkit-scrollbar-button{
    background: rgba(0, 0, 0, 0.0);
  }
  ::-webkit-scrollbar-corner {
    background: rgba(0, 0, 0, 0.0);
  }
}
@media (max-width: 600px) {
  .footer__description{
    font-size: 14px !important;
  }
  .footer{
    height: 300px !important;
  }
}
body{
  min-height: 100vh;
  width: 100vw;
  overflow-y: overlay;
  overflow-x: hidden;
}
#app {
  font-family: Helvetica;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #11142d;
  min-height: 100vh;
  width: 100vw;
  display: flex;

```

```

    flex-direction: column;
  }
  .logo{
    font-size: 24px;
    color: #6a80cb;
    text-transform: capitalize;
  }
  #nav {
    width: 100vw;
    position: fixed;
    top: 0;
    left: 0;
    height: 72px;
    padding: 0 72px;
    display: flex;
    align-items: center;
    background: #ffffff;
    box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
    z-index: 999;
    a {
      font-size: 18px;
      color: #6a80cb;
      margin-right: 10px;
      font-weight: 600;
      transition: 0.2s;
      &:hover{
        color: #e2e2e2;
      }
      &.router-link-exact-active {
        color: #11142d;
        text-shadow: 0 0 5px rgba(255, 224, 51, 0.13);
        // box-shadow: 0 0 10px rgba(255, 224, 51, 0.13);
      }
    }
  }
  .content{
    width: 100vw;
    height: 100%;
    padding: 16px 16px 0 16px;
    flex: 1 0 auto;
    display: flex;
    flex-direction: column;
  }
  .footer{
    flex: 0 0 auto;
    height: 200px;
    width: 100vw;
    margin-top: 50px;
    box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
    padding: 0 72px;
    display: flex;

```

```

align-items: center;
justify-content: space-between;
.footer__description{
  font-size: 16px;
  width: 40%;
}
a {
  font-size: 18px;
  color: #11142d;
  margin-top: 10px;
  transition: 0.2s;
  &:hover{
    color: #e2e2e2;
  }
}
.button{
  height: 50px;
  width: 300px;
  background: #6a80cb;
  box-shadow: 0 0 10px rgba(255, 224, 51, 0.13);
  display: grid;
  place-items: center;
  font-size: 18px;
  color: #fff;
  border-radius: 4px;
  transition: 0.1s;
  &:hover{
    background: #1b1d21;
    color: #fff;
  }
}
</style>

```

Home.vue

```

<template>
<div class="home">
  <section class="section section__main">
    Property Estate
  </section>
  <div class="filter">
    <ul class="filter__list">
      <li class="filter__list-item">
        <p>цена</p>
        <div>
          <span>0</span><input type="range" id="inputcost" @change="filterChange()" v-model="cost" min="0"
:max="maxcost" :step="10000"><span>{{ Math.floor(cost) }}</span>
        </div>
      </li>
      <li class="filter__list-item">

```



```

    <p>комнаты</p>
    <div>
      <span>1</span><input type="range" id="inputroom" @change="filterChange()" v-model="room" min="0"
:max="maxroom" step="1"><span>{{ room }}</span>
    </div>
  </li>
  <li class="filter__list-item">
    <p>этаж</p>
    <div>
      <span>1</span><input type="range" id="inputfloor" @change="filterChange()" v-model="floor" min="0"
:max="maxfloor" step="1"><span>{{ floor }}</span>
    </div>
  </li>
  <li class="filter__list-item">
    <p>площадь</p>
    <div>
      <span>1</span><input type="range" id="inputsize" @change="filterChange()" v-model="size" min="0"
:max="maxsize" step="1"><span>{{ size }}</span>
    </div>
  </li>
</ul>
<select @change="filterChange()" v-model="selectcategory">
  <option value="0">тип строения</option>
  <option :value="category.id" v-for="category in categories" :key="category.id">{{ category.name }}</option>
</select>
<select @change="filterChange()" v-model="selectfinish">
  <option value="0">отделка</option>
  <option :value="finish.id" v-for="finish in finishing" :key="finish.id">{{ finish.name }}</option>
</select>
</div>
<section class="section section__store">
  <ul class="store__list">
    <li class="store__list-item" v-for="property in propertys" :key="property.id">
      <ul class="pictures">
        <li class="picture" :style="'background-image: url(https://property-
estate.website/'+loadPisture(property.id)+')"></li>
      </ul>
      <div class="description">
        <p class="name" :title="'id недвижимости: '+property.id">{{ property.name }}</p>
        <p>тип: {{ property.category }}</p>
        <p>количество комнат: {{ property.rooms }}</p>
        <p>этаж: {{ property.floor }}</p>
        <p>площадь: {{ property.size }}м²</p>
        <p>отделка: {{ property.finishing }}</p>
        <p>адрес: {{ property.city }}, {{ property.street }}, {{ property.streetinfo }}</p>
        <div class="wrapp">
          <span class="cost">{{ property.cost }}грн</span>
          <button class="button" @click="openPopup(); selectedId = property.id">хочу посмотреть</button>
        </div>
      </div>
    </li>
  </ul>

```

```

</ul>
</section>
<div class="popup">
  <div class="cross" @click="closePopup">
  </div>
  <div class="form">
    <input class="pname" v-model="name" type="text" placeholder="имя">
    <input class="pphone" v-model="phone" type="tel" placeholder="номер телефона">
    <input v-model="email" type="email" placeholder="почта">
    <button class="button" @click="sendProperty()">отправить</button>
  </div>
</div>
</div>
</template>

<script>
export default {
  name: 'Home',
  data () {
    return {
      image: "https://storage.pik-broker.ru/public-
catalog/upload/images/5a/7c/30/49/5a7c3049d3bfa2a6f594ac4b49430671_pik_image_494_279.webp",
      propertys: {},
      images: {},
      finishing: {},
      categories: {},
      maxcost: 0,
      maxroom: 0,
      maxfloor: 0,
      maxsize: 0,
      cost: 0,
      room: 0,
      floor: 0,
      size: 0,
      selectfinish: 0,
      selectcategory: 0,
      name: "",
      phone: "",
      email: "",
      selectedId: 0,
    }
  },
  mounted() {
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 2 }, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data);
      this.images = response.data
    })
  }
}

```

```

this.loadCategories()
this.loadFinishing()
this.loadPropetys()
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 5 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data);
  this.maxcost = response.data
  this.cost= this.maxcost
})
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 6 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data);
  this.maxfloor = response.data
  this.floor= this.maxfloor
})
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 7 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data);
  this.maxroom = response.data
  this.room= this.maxroom
})
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 8 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data);
  this.maxsize = response.data
  this.size= this.maxsize
})
},
methods:{
  loadPisture(idproperty){
    for (let i = 0; i < this.images.length; i++) {
      const image = this.images[i];
      if(image.idproperty == idproperty){
        return image.imagepath
      }
    }
  },
  loadPropetys(){

```

```

this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 1 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data);
  this.property = response.data
})
},
loadCategories(){
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 3 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data)
  this.categories = response.data
})
},
loadFinishing(){
this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 4 }, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
}).then((response) => {
  console.log(response.data)
  this.finishing = response.data
})
},
sendProperty(){
  if(this.name != "" && this.phone != ""){
    this.closePopup()
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 10, name: this.name, phone:
this.phone, email: this.email, id: this.selectedId }, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data)
    })
  }
  if(this.name == ""){
    document.querySelector('.pname').style.borderBottom = "1px solid red"
  }
  if(this.phone == ""){
    document.querySelector('.pphone').style.borderBottom = "1px solid red"
  }
},
filterChange(){
  this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 9, cost: this.cost, room: this.room, floor:
this.floor, size: this.size, finishing: this.selectfinish, category: this.selectcategory }, {

```

```

headers: {
  'Content-Type': 'multipart/form-data'
}
}).then((response) => {
  console.log(this.selectcategory);
  console.log(response.data);
  this.property = response.data
})
},
openPopup(){
  document.querySelector('.popup').style.display = "grid"
  document.querySelector('.pname').style = "none"
  document.querySelector('.pphone').style = "none"
  this.name = ""
  this.phone = ""
  this.email = ""
  this.selectedId= ""
},
closePopup(){
  document.querySelector('.popup').style.display = "none"
}
}
}
</script>
<style lang="scss" scoped>
.popup{
  width: 100vw;
  height: 100vh;
  position: fixed;
  z-index: 9999;
  background: #fff;
  top: 0;
  left: 0;
  display: none;
  place-items: center;
  .cross{
    position: absolute;
    width: 35px;
    height: 35px;
    transform: translate(-50%,-50%);
    top: 50px;
    left: 50%;
    cursor: pointer;
    &:before{
      position: absolute;
      transform: rotateZ(-45deg);
      content: "";
      height: 2px;
      width: 35px;
      background: rgba(151, 165, 193, 1);
    }
  }
}

```

```

&:after{
  position: absolute;
  transform: rotateZ(45deg);
  content: "";
  height: 2px;
  width: 35px;
  background: rgba(151, 165, 193, 1);

}
}
.form{
  width: 350px;
  box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
  padding: 25px;
  border-radius: 25px;
  input{
    width: 300px;
    height: 50px;
    margin-bottom: 25px;
    border-bottom: 1px solid rgba(151, 165, 193, 0.2);
  }
  button{
    margin-top: 25px;
  }
}
}
.section{
  width: 100%;
}
.section__main{
  position: relative;
  left: -16px;
  width: 100vw;
  height: calc(100vh - 16px);
  background-image: url('https://property-estate.website/img/icons/main.jpg');
  background-repeat: no-repeat;
  background-position: center center;
  background-size: cover;
  display: grid;
  place-items: center;
  color: #fff;
  font-size: 70px;
}
.filter{
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 50px;
  height: 50px;
  position: fixed;
  z-index: 999;
}

```

```

top: 72px;
left: 0;
width: 100vw;
background: #fff;
.filter__list{
  justify-content: space-between;
  align-items: center;
  display: flex;
  .filter__list-item{
    margin: 0 10px;
  }
  input{
    margin: 0 10px;
    cursor: pointer;
  }
}
select{
  width: 150px;
  padding: 10px;
  cursor: pointer;
}
}
.store__list{
  margin-top: 50px;
  padding: 0 100px;
  .store__list-item{
    height: 350px;
    box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
    width: 100%;
    border-radius: 40px;
    display: grid;
    grid-template-columns: 350px 620px;
    .pictures{
      align-self: center;
      justify-self: center;
      height: 320px;
      width: 320px;
      position: relative;
      .picture{
        height: 320px;
        width: 320px;
        background-repeat: no-repeat;
        background-position: center center;
        background-size: cover;
        border-radius: 30px;
      }
    }
  }
  .description{
    margin-left: auto;
    margin-right: 25px;
    display: flex;

```

```

flex-direction: column;
justify-content: space-between;
padding: 15px;
font-size: 20px;
.button{
  display: inline;
  margin-left: 25px;
}
.wrapp{
  display: flex;
  align-items: center;
}
.name{
  font-size: 30px;
}
.cost{
  font-size: 36px;
}
}
}
}
}
</style>

```

index.vue\about

```

<template>
<div class="about">
  <h3>Агенство недвижимости Property Estate занимается продажей недвижимости с 1149г до н.э.</h3>
  <p>у нас <span>1 025 345</span> отзывов</p>
  <p>наше агенство <span>3</span> раза продало <span>Биг-Бен</span></p>
  <p>Выбирайте нас и сможете купить <span>Пирамиду Хеопса</span> в 3 клика</p>
</div>
</template>
<style lang="scss" scoped>
.about{
  padding: 100px;
  margin-top: 72px;
  font-size: 24px;
  line-height: 50px;
  span{
    color: #6a80cb;
  }
  h3{
    font-size: 30px;
  }
}
}
</style>

```

index.vue\admin


```

<template>
<div class="admin">
  <section class="section" v-if="token == "">
    <div class="login">
      <input class="phone" type="text" v-model="phone" placeholder="телефон">
      <input class="pass" type="text" v-model="password" placeholder="пароль">
      <button class="button" @click="login()">войти</button>
    </div>
  </section>
  <section class="section" v-else>
    <div class="admin__header">
      <button class="button" @click="exit()">выйти</button>
      <button class="button" @click="openPopup(1)">добавить недвижимость</button>
      <button class="button" @click="openPopup(2)">удалить недвижимость</button>
    </div>
    <div class="bids">
      <h3>Активные заявки:</h3>
      <ul class="bids__list">
        <li class="bids__list-item" v-for="bid in bids" :key="bid.id">
          <div class="bids__header">
            <span>имя: {{bid.cname}}</span>
            <span>телефон: {{bid.cphone}}</span>
            <span>почта: {{bid.cemail}}</span>
          </div>
          <div class="property-wrapp">
            <ul class="pictures">
              <li class="picture" :style="`background-image: url(https://property-estate.website/'+loadPisture(bid.idproperty)+')`"></li>
            </ul>
            <div class="description">
              <p class="name" :title="'id недвижимости: '"></p>
              <p>тип: {{bid.category}}</p>
              <p>количество комнат: {{bid.rooms}}</p>
              <p>этаж: {{bid.floor}}</p>
              <p>площадь: {{bid.size}} м²</p>
              <p>отделка: {{bid.finishing}}</p>
              <p>адрес: {{bid.city}}, {{bid.street}}, {{bid.streetinfo}}</p>
              <div class="wrapp">
                <span class="cost">{{bid.cost}} грн</span>
              </div>
            </div>
            <button class="button" @click="closeProperty(bid.idproperty)">завершить заявку</button>
          </div>
        </li>
      </ul>
    </div>
  </section>
  <div class="popup">
    <div class="cross" @click="closePopup()">
    </div>
    <div class="form form1">

```

```

<input v-model="propertyToAdd.name" type="text" placeholder="название">
<input v-model="propertyToAdd.size" type="number" placeholder="площадь">
<input v-model="propertyToAdd.room" type="number" placeholder="количество комнат">
<input v-model="propertyToAdd.cost" type="number" placeholder="цена">
<input v-model="propertyToAdd.description" type="text" placeholder="описание">
<input v-model="propertyToAdd.floor" type="text" placeholder="этаж">
<input v-model="propertyToAdd.city" type="text" placeholder="город">
<input v-model="propertyToAdd.street" type="text" placeholder="улица">
<input v-model="propertyToAdd.streetinfo" type="text" placeholder="дом корпус квартира">
<select id="selectcategory">
  <option v-for="category in categories" :key="category.id" :value="category.id">{{ category.name }}</option>
</select>
<select id="selectfinishing">
  <option v-for="finish in finishing" :key="finish.id" :value="finish.id">{{ finish.name }}</option>
</select>
<input id="fileinput" type="file" placeholder="картинка" accept="image/jpeg,image/png" required>
<button class="button" @click="addProperty()">добавить</button>
</div>
<div class="form form2">
  <input v-model="selectedid" type="number" placeholder="id">
  <button class="button" @click="deleteProperty()">удалить</button>
</div>
</div>
</div>
</template>

<script>
export default {
  name: 'Admin',
  data(){
    return{
      token: "",
      phone: "",
      password: "",
      bids: {},
      images: {},
      selectedid: "",
      categories: {},
      finishing: {},
      propertyToAdd: {
        name: "",
        street: "",
        city: "",
        streetinfo: "",
        size: "",
        room: "",
        description: "",
        cost: "",
        floor: ""
      }
    }
  }
}

```

```

    }
  },
  mounted(){
    if(localStorage.estatetoken){
      this.token = localStorage.estatetoken
    }
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: "12"}, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data)
      this.bids = response.data
    })
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 2}, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data);
      this.images = response.data
    })
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 3}, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data)
      this.categories = response.data
    })
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: 4}, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log(response.data)
      this.finishing = response.data
    })
  },
  methods: {
    loadPisture(idproperty){
      for (let i = 0; i < this.images.length; i++) {
        const image = this.images[i];
        if(image.idproperty == idproperty){
          return image.imagepath
        }
      }
    },
  },
  exit(){
    localStorage.clear('estatetoken')
    this.token = ""
  }
}

```

```

},
login(){
  if(this.password != "" && this.phone != ""){
    this.axios.post('https://property-estate.website/api/api.php', { apiexecute: "11", phone: this.phone, password:
this.password}, {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then((response) => {
      console.log({ apiexecute: "11", phone: this.phone, password: this.password});
      console.log(response.data)
      if(response.data.token){
        localStorage.estatetoken = 1
        this.token = localStorage.estatetoken
        document.querySelector('.pass').style = "none"
        document.querySelector('.phone').style = "none"
      }else{
        alert('проверьте введенные данные')
      }
    })
  }
  if(this.password == ""){
    document.querySelector('.phone').style.borderBottom = "1px solid red"
  }
  if(this.phone == ""){
    document.querySelector('.pass').style.borderBottom = "1px solid red"
  }
},
deleteProperty(){
  this.axios.post('https://property-estate.website/api/api.php', { apiexecute: "13", idproperty: this.selectedid}, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  }).then((response) => {
    console.log(response.data)
    this.closePopup()
  })
},
closeProperty(idproperty){
  this.axios.post('https://property-estate.website/api/api.php', { apiexecute: "13", idproperty: idproperty}, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  }).then((response) => {
    console.log(response.data)
    this.closePopup()
  })
},
openPopup(npopup){
  if(npopup == 1){
    document.querySelector('.form1').style.display = "block"
  }
}

```

```

    document.querySelector('.form2').style.display = "none"
    document.querySelector('.popup').style.display = "grid"
  }else if(npopup == 2){
    document.querySelector('.form1').style.display = "none"
    document.querySelector('.form2').style.display = "block"
    document.querySelector('.popup').style.display = "grid"
  }
},
closePopup(){
  document.querySelector('.popup').style.display = "none"
  this.axios.post('https://property-estate.website/api/api.php', { apiexecute: "12"}, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  }).then((response) => {
    console.log(response.data)
    this.bids = response.data
  })
},
addProperty(){
  this.propertyToAdd.category = document.querySelector('#selectcategory').value
  this.propertyToAdd.finishing = document.querySelector('#selectfinishing').value
  let formdata = new FormData()
  formdata.append('apiexecute', 15)
  formdata.append('name',this.propertyToAdd.name)
  formdata.append('cost',this.propertyToAdd.cost)
  formdata.append('room',this.propertyToAdd.room)
  formdata.append('description',this.propertyToAdd.description)
  formdata.append('size',this.propertyToAdd.size)
  formdata.append('city',this.propertyToAdd.city)
  formdata.append('street',this.propertyToAdd.street)
  formdata.append('floor',this.propertyToAdd.floor)
  formdata.append('streetinfo',this.propertyToAdd.streetinfo)
  formdata.append('category', this.propertyToAdd.category)
  formdata.append('finishing', this.propertyToAdd.finishing)
  formdata.append('image', document.getElementById('fileinput').files[0],
document.getElementById('fileinput').files[0].name)
  this.axios.post('https://property-estate.website/api/api.php', formdata, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  }).then((response) => {
    console.log(response.data)
    this.closePopup()
  })
}

// changeSlide(){
// if(this.page == 1){
//   document.querySelectorAll('.page__list-item')[0].style.display = 'flex'
//   document.querySelectorAll('.page__list-item')[1].style.display = 'none'

```

```

// document.querySelectorAll('.page__list-item')[2].style.display = 'none'
// document.querySelectorAll('.nav__list-item')[0].style.fontSize = '24px'
// document.querySelectorAll('.nav__list-item')[1].style.fontSize = '18px'
// document.querySelectorAll('.nav__list-item')[2].style.fontSize = '18px'
// this.loadusers()

// }else if(this.page == 2){
// document.querySelectorAll('.page__list-item')[0].style.display = 'none'
// document.querySelectorAll('.page__list-item')[1].style.display = 'flex'
// document.querySelectorAll('.page__list-item')[2].style.display = 'none'
// document.querySelectorAll('.nav__list-item')[0].style.fontSize = '18px'
// document.querySelectorAll('.nav__list-item')[1].style.fontSize = '24px'
// document.querySelectorAll('.nav__list-item')[2].style.fontSize = '18px'
// this.loadProducts()

// }else if(this.page == 3){
// document.querySelectorAll('.page__list-item')[0].style.display = 'none'
// document.querySelectorAll('.page__list-item')[1].style.display = 'none'
// document.querySelectorAll('.page__list-item')[2].style.display = 'flex'
// document.querySelectorAll('.nav__list-item')[0].style.fontSize = '18px'
// document.querySelectorAll('.nav__list-item')[1].style.fontSize = '18px'
// document.querySelectorAll('.nav__list-item')[2].style.fontSize = '24px'
// this.loadPostmans()
// }
// },
// loadPostmans(){
// this.axios.post('https://product-delivery.website/api/api.php', {apiexecute: 12, token: localStorage.token}, {
// headers: {
// 'Content-Type': 'multipart/form-data'
// }
// }).then((response) => {
// this.postmans = response.data
// })
// },
// loadusers(){
// this.axios.post('https://product-delivery.website/api/api.php', {apiexecute: 13, token: localStorage.token}, {
// headers: {
// 'Content-Type': 'multipart/form-data'
// }
// }).then((response) => {
// this.users = response.data
// })
// },
// loadProducts(){
// this.axios.post('https://product-delivery.website/api/api.php', {apiexecute: 14, token: localStorage.token}, {
// headers: {
// 'Content-Type': 'multipart/form-data'
// }
// }).then((response) => {
// this.products = response.data
// })

```

```

// },
// popupClose(){
// document.querySelectorAll('.popUpProduct')[0].style.display = "none"
// document.querySelectorAll('.popUpProduct')[1].style.display = "none"
// },
// popupAddProduct(){
// document.querySelectorAll('.popUpProduct')[0].style.display = "grid"
// },
// popupRemoveProduct(){
// document.querySelectorAll('.popUpProduct')[1].style.display = "grid"
// },
// addProduct(){
// this.productToAdd.category = document.getElementById('select').value
// let formdata = new FormData()
// formdata.append('apiexecute', 15)
// formdata.append('name',this.productToAdd.name)
// formdata.append('cost',this.productToAdd.cost)
// formdata.append('weight',this.productToAdd.weight)
// formdata.append('category',this.productToAdd.category)
// formdata.append('qtyonstorage',this.productToAdd.qtyOnStorage)
// formdata.append('image', document.getElementById('fileinput').files[0],
document.getElementById('fileinput').files[0].name)
// this.axios.post('https://product-delivery.website/api/api.php', formdata, {
// headers: {
// 'Content-Type': 'multipart/form-data'
// }
// }).then((response) => {
// this.popupClose()
// this.loadProducts()
// })
// },
// removeProduct(){
// this.axios.post('https://product-delivery.website/api/api.php', { apiexecute: 16, product: this.selectedProduct }, {
// headers: {
// 'Content-Type': 'multipart/form-data'
// }
// }).then((response) => {
// this.popupClose()
// this.loadProducts()
// })
// }
}

}
</script>
<style lang="scss" scoped>
.section{
position: relative;
min-height: calc(100vh - 16px);
width: 100%;
}

```

```

.admin__header{
  margin-top: 72px;
  width: 100%;
  display: flex;
  justify-content: space-evenly;
}
.login{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 350px;
  box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
  padding: 25px;
  border-radius: 25px;
  input{
    width: 300px;
    height: 50px;
    margin-bottom: 25px;
    border-bottom: 1px solid rgba(151, 165, 193, 0.2);
  }
  button{
    margin-top: 25px;
  }
}
.pictures{
  align-self: center;
  justify-self: center;
  height: 320px;
  width: 320px;
  position: relative;
  padding: 15px;
  .picture{
    height: 320px;
    width: 320px;
    background-repeat: no-repeat;
    background-position: center center;
    background-size: cover;
    border-radius: 30px;
  }
}
.description{
  margin: 0 25px;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  padding: 15px;
  font-size: 20px;
  .button{
    display: inline;
    margin-left: 25px;
  }
}

```



```

}
.wrapp{
  display: flex;
  align-items: center;
}
.name{
  font-size: 30px;
}
.cost{
  font-size: 36px;
}
}
.property-wrapp{
  height: 350px;
  width: 100%;
  display: flex;
}
.bids{
  h3{
    font-size: 24px;
    line-height: 50px;
    padding: 0 100px
  }
}
.bids__list{
  padding: 0 100px;
  .bids__list-item{
    position: relative;
    margin-bottom: 50px;
    padding: 25px;
    border-radius: 40px;
    box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
    .bids__header{
      display: flex;
      justify-content: space-evenly;
      font-size: 24px;
      height: 50px;
      align-items: center;
    }
    button{
      position: absolute;
      right: 25px;
      bottom: 25px;
    }
  }
}
.popup{
  width: 100vw;
  height: 100vh;
  position: fixed;
  z-index: 9999;
}

```

```

background: #fff;
top: 0;
left: 0;
display: none;
place-items: center;
.cross{
  position: absolute;
  width: 35px;
  height: 35px;
  transform: translate(-50%,-50%);
  top: 50px;
  left: 50%;
  cursor: pointer;
  &:before{
    position: absolute;
    transform: rotateZ(-45deg);
    content: "";
    height: 2px;
    width: 35px;
    background: rgba(151, 165, 193, 1);
  }
  &:after{
    position: absolute;
    transform: rotateZ(45deg);
    content: "";
    height: 2px;
    width: 35px;
    background: rgba(151, 165, 193, 1);
  }
}
}
.form{
  width: 350px;
  box-shadow: 0 14px 86px rgba(151, 165, 193, 0.2);
  padding: 25px;
  border-radius: 25px;
  input{
    width: 300px;
    height: 50px;
    margin-bottom: 25px;
    border-bottom: 1px solid rgba(151, 165, 193, 0.2);
  }
  button{
    margin-top: 25px;
  }
}
}
.form1{
  width: 725px;
  display: grid;
  grid-template-columns: 350px 350px;
}

```

```
}  
</style>
```

index.js

```
import { createRouter, createWebHistory } from 'vue-router'  
import Home from '../views/Home.vue'
```

```
const routes = [  
  {  
    path: '/',  
    name: 'Home',  
    component: Home  
  },  
  {  
    path: '/admin',  
    name: 'Admin',  
    component: () => import('../views/admin/')  
  },  
  {  
    path: '/about',  
    name: 'About',  
    component: () => import('../views/about/')  
  },  
]
```

```
const router = createRouter({  
  mode: "history",  
  history: createWebHistory(process.env.BASE_URL),  
  routes  
})
```

```
export default router
```

api.php

```
<?php  
header("Access-Control-Allow-Origin: *");  
header('Content-Type: application/json;charset=utf-8');  
header('Accept: application/json');  
require 'rb-mysql.php';  
  
R::setup('mysql:host=81.177.139.61;dbname=9295608607_estate', '047245193_estate', 'passtodb');  
  
if( !R::testConnection() ){  
    echo json_encode('База данных недоступна');  
}  
$postData = file_get_contents('php://input');  
$data = json_decode($postData, true);  
if(json_decode($postData, true)['apiexecute'] == null){
```

```

$data = $_POST;
}
switch ($data['apiexecute']){
case 1:
    $response = R::getAll('SELECT property.id AS id,property.name AS name, property.cost AS cost, property.description AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS finishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.id = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = finishing.id WHERE property.isactive = 1 AND property.idsale = 1');
    echo json_encode($response);
    break;
case 2:
    $response = R::getAll('SELECT * FROM `images` ORDER BY id');
    echo json_encode($response);
    break;
case 3:
    $response = R::getAll('SELECT * FROM `category` ORDER BY id');
    echo json_encode($response);
    break;
case 4:
    $response = R::getAll('SELECT * FROM `finishing` ORDER BY id');
    echo json_encode($response);
    break;
case 5:
    $response = R::getAll('SELECT * FROM `property` ORDER BY cost DESC');
    echo json_encode($response[0]['cost']);
    break;
case 6:
    $response = R::getAll('SELECT * FROM `property` INNER JOIN `house` ON house.id =property.idhouse ORDER BY floor DESC');
    echo json_encode($response[0]['floor']);
    break;
case 7:
    $response = R::getAll('SELECT * FROM `property` INNER JOIN `house` ON house.id =property.idhouse ORDER BY rooms DESC');
    echo json_encode($response[0]['rooms']);
    break;
case 8:
    $response = R::getAll('SELECT * FROM `property` INNER JOIN `house` ON house.id =property.idhouse ORDER BY size DESC');
    echo json_encode($response[0]['size']);
    break;
case 9:
    if($data['category'] == 0){
        if($data['finishing'] == 0){
            $response = R::getAll('SELECT property.id AS id,property.name AS name, property.cost AS cost, property.description AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS finishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.id = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = finishing.id');
        }
    }
}

```

```

shing.id WHERE house.size <= '.$data['size'].' AND house.floor <= '.$data['floor'].' AND property.cost <= '.$data['
cost'].' AND house.rooms <= '.$data['room'].' AND property.isactive = 1 AND property.idsale = 1');
}else{
    $response = R::getAll('SELECT property.id AS id,property.name AS name, property.cost AS cost, property.de
scription AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS f
inishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.i
d = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.
id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = fini
shing.id WHERE house.size <= '.$data['size'].' AND house.floor <= '.$data['floor'].' AND property.cost <= '.$data['
cost'].' AND house.rooms <= '.$data['room'].' AND house.idfinishing = '.$data['finishing'].' AND property.isactive =
1 AND property.idsale = 1');
}
}else{
    if($data['finishing'] == 0){
        $response = R::getAll('SELECT property.id AS id,property.name AS name, property.cost AS cost, property.de
scription AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS f
inishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.i
d = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.
id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = fini
shing.id WHERE house.size <= '.$data['size'].' AND house.floor <= '.$data['floor'].' AND property.cost <= '.$data['
cost'].' AND house.rooms <= '.$data['room'].' AND house.idcategory = '.$data['category'].' AND property.isactive =
1 AND property.idsale = 1');
    }else{
        $response = R::getAll('SELECT property.id AS id,property.name AS name, property.cost AS cost, property.de
scription AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS f
inishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.i
d = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.
id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = fini
shing.id WHERE house.size <= '.$data['size'].' AND house.floor <= '.$data['floor'].' AND property.cost <= '.$data['
cost'].' AND house.rooms <= '.$data['room'].' AND house.idcategory = '.$data['category'].' AND house.idfinishing =
'.$data['finishing'].' AND property.isactive = 1 AND property.idsale = 1');
    }
}

echo json_encode($response);
break;
case 10:
    $response = R::dispense('customer');
    $response->name = $data['name'];
    $response->phone = $data['phone'];
    $response->email = $data['email'];
    $response->idproperty = $data['id'];
    R::store($response);
    echo json_encode($response);
    break;
case 11:
    $response = R::getAll('SELECT * FROM `user` WHERE phone = "'.$data['phone'].'"');
    if(count($response) == 0){
        echo json_encode($response);
    }else{
        $response = array(

```

```

        "token" => $response[0]['id']
    );
    echo json_encode($response);
}
break;
case 12:
    $response = R::getAll('SELECT property.id AS idproperty, customer.id AS id, customer.name AS cname, customer.phone AS cphone, customer.email AS cemail,property.name AS name, property.cost AS cost, property.description AS description, city.name AS city, street.name AS street, category.name AS category, finishing.name AS finishing, house.streetinfo, house.floor, house.rooms, house.size FROM `property` INNER JOIN `house` ON house.id = property.idhouse INNER JOIN `city` ON house.idcity = city.id INNER JOIN `street` ON house.idstreet = street.id INNER JOIN `category` ON house.idcategory = category.id INNER JOIN `finishing` ON house.idfinishing = finishing.id INNER JOIN `customer` ON property.id = customer.idproperty WHERE property.isactive = 1 AND property.idsale = 1');
    echo json_encode($response);
    break;
case 13:
    if($data['idproperty'] != ""){
        $response = R::load('property', $data['idproperty']);
        $response->isactive = 0;
        R::store($response);
        echo json_encode($response);
    }
    break;
case 14:
    $response = R::load('property', $data['idproperty']);
    $response->idsale = 2;
    R::store($response);
    echo json_encode($response);
    break;
case 15:
    $dir = "/img";
    $fieldname = 'image';
    $filename = $_FILES[$fieldname]['name'];
    $tmpname = $_FILES[$fieldname]['tmp_name'];
    $response = R::getAll('SELECT * FROM `city` WHERE `name` = "'.$data['city'].'"');
    if(count($response) == 0){
        $response = R::dispense('city');
        $response->name = $data['city'];
        R::store($response);
    }else{
        $response = $response[0];
    }
    $response2 = R::getAll('SELECT * FROM `street` WHERE `name` = "'.$data['street'].'"');
    if(count($response2) == 0){
        $response2 = R::dispense('street');
        $response2->name = $data['street'];
        R::store($response2);
    }else{
        $response2 = $response2[0];
    }
}

```

```

$response3 = R::dispense('house');
$response3->idcity = $response['id'];
$response3->idstreet = $response2['id'];
$response3->streetinfo = $data['streetinfo'];
$response3->idcategory = $data['category'];
$response3->floor = $data['floor'];
$response3->rooms = $data['room'];
$response3->size = $data['size'];
$response3->idfinishing = $data['finishing'];
$response3->isactive = 1;
R::store($response3);

$response4 = R::dispense('property');
$response4->name = $data['name'];
$response4->cost = $data['cost'];
$response4->idhouse = $response3['id'];
$response4->description = $data['description'];
$response4->idsale = 1;
$response4->isactive = 1;
R::store($response4);

$response5 = R::dispense('images');
$response5->idproperty = $response4['id'];
$id = $response4['id'];
if(!is_dir('..' . $dir . '/' . $id . "")) {
    mkdir('..' . $dir . '/' . $id . "", 0777, true);
}
move_uploaded_file ( $tmpname , '..' . $dir . '/' . $id . '/' . $filename . "" );
$response5->imagepath = " . $dir . '/' . $id . '/' . $filename . "" ;
R::store($response5);
echo json_encode($response5);
break;
}
exit;
?>

```

ДОДАТОК Б

Відгук керівника економічного розділу

ДОДАТОК В**Перелік файлів на диску****ПЕРЕЛІК ДОКУМЕНТІВ НА МАГНІТНОМУ НОСІЇ**

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота Колбасов.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота Колбасов.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Колбасов.rar	Архів. Містить коди програми і скомпільовану програму
Презентація	
Колбасов.ppt	Презентація кваліфікаційної роботи