

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Інтелектуальна система рекомендацій для
платформи пошуку автомобілів з використанням
алгоритмів машинного навчання»**

Виконав студент групи ПЗм-23-1
РУДИК Денис Олександрович

Керівник роботи:
ЧИЖМОТРЯ Олена Геннадіївна

Рецензент:
ЄФРЕМОВ Юрій Миколайович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«1» жовтня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **РУДИК Денис Олександрович**

Керівник роботи: **ЧИЖМОТРЕ Олена Геннадіївна**

Тема роботи: **«Інтелектуальна система рекомендацій для платформи пошуку автомобілів з використанням алгоритмів машинного навчання»**,

затверджена Наказом закладу вищої освіти від «1» жовтня 2024 р. № 520/с

Вихідні дані для роботи: **система рекомендацій, використання для аналізу асортименту, оцінка цін, огляд якості, оцінка процесу замовлення та доставки.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О. Г.	2.10.2024р.	2.10.2024р.
2	Чижмотря О. Г.	2.10.2024р.	2.10.2024р.
3	Чижмотря О. Г.	2.10.2024р.	2.10.2024р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена для полегшення пошуку та вибору авто для покупця. Пояснювальна записка до кваліфікаційної роботи містить 67 сторінок, 40 ілюстрацій та 9 таблиць, 2 додатки та 49 посилань на зовнішні джерела.

Метою даної роботи є розробка системи рекомендацій для платформи пошуку автомобілів на основі алгоритмів машинного навчання. У рамках проєкту було реалізовано модуль для підбору автомобілів, додано функціонал для публікації нових оголошень, створено сторінку гарячої лінії з контактною інформацією, а також модуль для відстеження транспортних засобів.

У процесі виконання роботи визначено ключові завдання для розробки додатку, проведено аналіз аналогічних програмних рішень, обґрунтовано вибір архітектурного патерна для реалізації. Описано сценарії використання програмного забезпечення, загальну структуру додатку, базу даних (Supabase), алгоритми взаємодії між модулями, а також функціонування основних модулів, реалізованих за допомогою мови програмування JavaScript.

КЛЮЧОВІ СЛОВА: СИСТЕМА РЕКОМЕНДАЦІЙ, ОГЛОШЕННЯ, ПОШУК, ГАРЯЧА ЛІНІЯ, БАЗА ДАНИХ.

ABSTRACT

The qualification work for the Master's degree is dedicated to simplifying the process of searching and selecting cars for buyers. The explanatory note for the qualification work comprises 67 pages, 40 illustrations, 9 tables, 2 appendices, and 49 references to external sources.

The purpose of this work is to develop a recommendation system for a car search platform based on machine learning algorithms. Within the framework of the project, a module for car selection was implemented, functionality for publishing new advertisements was added, a hotline page with contact information was created, and a module for tracking vehicles was developed.

During the implementation of the project, the key tasks for application development were identified, an analysis of similar software solutions was conducted, and the choice of an architectural pattern for implementation was justified. The document describes the use scenarios of the software, the overall application structure, the database (Supabase), algorithms for module interaction, as well as the operation of the main modules developed using the JavaScript programming language.

KEYWORDS: RECOMMENDATION SYSTEM, ADVERTISEMENTS, SEARCH, HOTLINE, DATABASE.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1. Постановка задачі	10
1.2. Аналіз аналогів програмного продукту.....	12
1.3. Вибір архітектури обміну інформацією	17
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	22
Висновки до першого розділу	24
РОЗДІЛ 2. РОЗРОБКА ТА ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ДЛЯ ПЛАТФОРМИ ПОШУКУ АВТОМОБІЛІВ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ	26
2.1 Визначення варіантів використання та об’єктно-орієнтованої структури системи	26
2.2 Розробка бази даних системи.....	31
2.3 Проєктування та реалізація алгоритмів роботи	33
2.4 Реалізація функціоналу в додаток.....	34
Висновки до другого розділу	35
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ.....	37
3.1 Порядок встановлення та налаштування параметрів системи	37
3.2 Тестування роботи програмного комплексу	50
Висновки до третього розділу	50
ВИСНОВКИ	53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ.....	60
<i>ДОДАТОК А</i>	<i>61</i>
<i>ДОДАТОК Б</i>	<i>65</i>

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – це прикладний програмний інтерфейс.

БД – база даних.

ПЗ – програмне забезпечення.

CRUD – це набір операцій (create, read, update, delete), що перекладається як "створити, прочитати, оновити, видалити".

VIN (Vehicle Identification Number) – унікальний ідентифікаційний номер автомобіля, який складається з 17 символів. Він містить інформацію про марку, характеристики авто та рік випуску.

ВСТУП

Актуальність теми. Купівля автомобілів через онлайн-платформи стає дедалі популярнішою завдяки стрімкому розвитку цифрових технологій. Інтернет відкриває доступ до широкого асортименту транспортних засобів, дозволяє порівнювати ціни, технічні характеристики та вибрати найкращий варіант. Це значно полегшує процес купівлі, але водночас створює виклики, пов'язані з ефективним пошуком, фільтрацією та персоналізованими рекомендаціями для користувачів.

Сучасна потреба в інтелектуальних рекомендаційних системах, які можуть допомогти швидко знайти автомобіль відповідно до заданих критеріїв (ціна, марка, модель, рік випуску тощо), є надзвичайно актуальною. Використання методів машинного навчання та штучного інтелекту дає змогу створювати алгоритми, які враховують індивідуальні вподобання користувачів, їхню історію взаємодії та характеристики автомобілів для надання точних і релевантних рекомендацій.

Метою випускної роботи полягає у створенні інтелектуальної системи рекомендацій для онлайн-платформи пошуку автомобілів на основі алгоритмів машинного навчання. Система має забезпечувати ефективний пошук, фільтрацію та персоналізовані рекомендації, враховуючи індивідуальні потреби користувачів, їхній бюджет та технічні вимоги.

З мети можна визначити наступні завдання:

- Провести аналіз існуючих платформ для пошуку автомобілів, визначити їхні переваги та недоліки.
- Розробити концепцію системи рекомендацій для платформи.
- Реалізувати алгоритм підбору автомобілів з використанням методів машинного навчання.
- Створити модуль для додавання нових оголошень.
- Розробити сторінку з контактною інформацією для гарячої лінії.
- Обґрунтувати вибір архітектурного патерна для реалізації додатку.

- Спроекувати базу даних для зберігання інформації про автомобілі, оголошення та користувачів.
- Описати сценарії взаємодії користувачів із додатком.
- Провести тестування та оптимізацію роботи основних модулів системи.

Об'єктом дослідження процес автоматизованого пошуку та рекомендацій автомобілів на платформі.

Предметом дослідження розробка інтелектуальної рекомендаційної системи на основі алгоритмів машинного навчання для веб-платформи пошуку автомобілів.

Для роботи використано методи об'єктно-орієнтованого програмування та проєктування, а також принципи створення інтуїтивного інтерфейсу користувача.

Наукова новизна роботи полягає у застосуванні сучасних методів машинного навчання для створення гібридної системи рекомендацій, яка поєднує collaborative filtering та content-based filtering. Це дозволяє підвищити точність і релевантність рекомендацій. Система враховує як індивідуальні вподобання користувачів, так і технічні характеристики автомобілів. Крім того, розроблено механізм обробки даних із інтеграцією системи у веб-додаток із сучасним і зручним інтерфейсом.

Практичне значення полягає у оптимізації процесу пошуку автомобілів, допомагаючи користувачам швидше знаходити транспортні засоби, що відповідають їхнім вимогам. Рекомендації, сформовані на основі машинного навчання, автоматизують підбір релевантних оголошень і покращують користувацький досвід. Можливість редагування, видалення та публікації оголошень спрощує керування даними. Інтеграція гарячої лінії забезпечує швидкий доступ до підтримки. База даних (Supabase) гарантує масштабованість і надійність платформи, а модульна архітектура дозволяє легко розширювати функціонал системи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Постановка задачі

Розробка даного додатку являє собою перш за все створення компонентів: простір для опублікування пропозицій, рекомендації авторизованого користувача, калькулятор розмитнення, пошукова система за VIN номером, фільтрація оголошень та додаткова контактна інформація для оформлення замовлення замовлення.

Додаток має реалізовувати наступні можливості:

- Опубліковувати нові оголошення;
- фільтрувати оголошення;
- відображати рекомендації користувача;
- відстеження вто по VIN номеру;
- обрахування повну суму розмитнення;
- ознайомство користувача з оформленням замовлення.

Сервіс представлятиме собою вебдодаток, який буде отримувати інформацію про наявність автомобілів. У даному додатку повинні бути два типи користувачів: користувач та користувач з правами адміністратора.

Користувач повинен:

- мати змогу ознайомлюватись з інформацією сайту;
- перегляд оголошень;
- користуватися відстеженням по VIN;
- фільтрувати оголошення;
- отримувати додаткову інформацію за потреби скориставшись телефонами гарячих ліній.

Для користувача з правами адміністратора повинен бути реалізований наступний функціонал:

- мати змогу ознайомлюватись з інформацією сайту;
- перегляд оголошень;
- користуватися відстеженням по VIN;

- фільтрувати оголошення;
- переглядати рекомендації;
- отримувати додаткову інформацію за потреби скориставшись телефонами гарячих ліній.

- опубліковувати оголошення;
- редагувати та видаляти їх.

Основними етапами розробки описаного додатку:

1. Розробка дизайну.

1.1. Розробка дизайну Figma.

1.2. Реалізація в коді HTML/CSS .

1.3. Перенесення даного коду в бібліотеку React.

1.4. Реалізація функціоналу додатку.

1.4.1. Створення структури для проекту.

1.4.2. Створення компонентів.

1.4.3. Реалізація функціоналу для навігації.

1.4.4. Створення авторизації через Google акаунт.

1.5. Реалізація генерації посилання на зображення.

2. Розробка клієнтської частини майбутнього сервісу.

2.1. Проектування концепції дизайну та користувацького інтерфейсу для створення навігаційних та інтерактивних елементів.

2.2. Розробка клієнтської частини та реалізація взаємодії з БД.

3. Розробка вебсервісу.

3.1.Реалізація API [1].

3.1.1. Створення програмних модулів для:

3.1.1.1. Авторизації користувачів.

3.1.1.2. Перегляду рекомендацій користувачів.

3.1.1.3. Пошуку авто.

3.1.1.3.1. Можливості перегляду інформації про автомобіль.

3.1.1.3.2. Редагування інформації характеристик.

4. Генерації id для користувача.

5. Створення компоненту розпізнавання користувача по id.
6. Демонстрація створених прикладів для тестового контенту.
7. Розробка плану тестування системи:
 - 7.1. Мануальне тестування.
 - 7.2. Тестування графічного інтерфейсу.

Результатом виконання поставлених завдань є розробка системи, яка автоматично ідентифікує користувачів під час реєстрації або авторизації за допомогою попередньо згенерованого ID користувача та передає відповідні дані до бази даних. Розроблений вебсервіс є застосунком, що дозволяє як звичайним користувачам, так і адміністраторам переглядати оголошення через браузер. Крім того, вебдодаток забезпечує зручний функціонал для створення та оновлення даних про автомобілі. Інтеграція компонентів забезпечує автоматизацію процесу купівлі-продажу транспортних засобів, а також надійне зберігання відповідної інформації в базі даних.

1.2. Аналіз аналогів програмного продукту

Для забезпечення конкурентоспроможності та функціональності розроблюваного додатку необхідно провести аналіз його аналогів. У ході дослідження існуючих вебдодатків не вдалося знайти повністю ідентичних рішень, проте були розглянуті інструменти, які надають можливість реалізувати необхідний функціонал. Під час аналізу були досліджені такі продукти, як Boss Auto [2], Auto Terminal [3] і Komis Auto [4].

Boss Auto (рис.1.1) – це компанія, що спеціалізується на продажу автомобілів. Її місія полягає у створенні умов, за яких процес купівлі авто буде максимально простим і надійним. Завдяки цьому клієнти зможуть уникнути ризиків, пов'язаних із підrobкою пробігу, прихованими пошкодженнями чи недобросовісною ціною автомобіля.

Серед основних переваг Boss Auto можна виділити: зручний інтерфейс, який забезпечує легке використання платформи навіть для новачків; автоматизацію рутинних процесів, що дозволяє ефективніше вести базу клієнтів,

управляти продажами та обробляти замовлення; підтримку мобільних пристроїв, яка надає доступ до системи з будь-якої точки світу у зручний час; масштабованість, що робить платформу підходящою як для невеликих автосалонів, так і для великих дилерських мереж; а також якісну технічну підтримку, яка допомагає вирішувати проблеми та адаптувати систему під потреби бізнесу.

Однак, Boss Auto має й недоліки. До них належить висока вартість, яка може бути завеликою для малих компаній і стартапів; складність налаштування системи, що вимагає додаткових ресурсів та залучення спеціалістів при інтеграціях; обмежений функціонал для компаній із нестандартними вимогами, що потребує додаткової кастомізації; а також необхідність витратити час на навчання персоналу, що тимчасово може впливати на продуктивність роботи.



Рисунок 1.1 - Логотип інструменту Boss Auto

Auto Terminal (рис. 1.2) – це компанія, яка надає можливість ознайомитися зі своєю діяльністю та скористатися послугами купівлі автомобілів. Вона пропонує різноманітні варіанти доставки, включаючи транспортування з країн ЄС і США.

Серед головних переваг Auto Terminal: автоматизація основних процесів, таких як управління автопарком і обробка замовлень, що дозволяє значно скоротити час на виконання рутинних завдань; простий і зрозумілий інтерфейс, який робить платформу доступною навіть для співробітників із мінімальними технічними навичками; мобільність завдяки доступу через мобільні додатки чи веб-версії, що дозволяє працювати в будь-якому місці; підтримка багатомовності, що є ідеальним рішенням для компаній із міжнародною діяльністю; а також гнучкі налаштування, які дозволяють адаптувати систему до специфіки бізнесу, забезпечуючи її ефективне використання автосалонами, дилерами та сервісними центрами.

Однак система Auto Terminal має і певні недоліки. Висока вартість впровадження, налаштування та інтеграції може бути недоступною для малих підприємств. Обмежений функціонал для специфічних бізнес-потреб може вимагати додаткових витрат на доопрацювання. Програма інколи може піддаватися технічним збоям чи проблемам із оновленнями, що здатні тимчасово зупинити роботу. Процес навчання персоналу може займати багато часу, особливо якщо функціонал платформи складний. Крім того, при обробці великого обсягу конфіденційних даних, таких як інформація про клієнтів, необхідний високий рівень захисту, адже недостатня безпека може створити ризик витоку даних.



Рисунок 1.2 - Логотип інструменту Auto Terminal

Komis Auto (рис. 1.3) – це платформа, яка дозволяє купувати автомобілі та реалізовувати власні рішення щодо їх доставки.

Серед ключових переваг Komis Auto можна виділити її адаптацію до особливостей бізнесу, пов'язаного з продажем вживаних автомобілів. Система враховує специфіку роботи з документами, оцінкою технічного стану авто та формуванням цін. Вона дозволяє ефективно керувати базою транспортних засобів, додаючи нові пропозиції, оновлюючи інформацію, завантажуючи фотографії, технічні характеристики та описи. Завдяки автоматизації таких процесів, як оцінка автомобілів, створення договорів купівлі-продажу та розрахунок комісійних, користувачі заощаджують час. Мобільний доступ забезпечує можливість роботи з платформою будь-де і будь-коли. Додатково

система пропонує юридичну підтримку у вигляді вбудованих шаблонів документів, таких як договори, акти огляду автомобілів та інші, що відповідають місцевим законодавчим нормам.

Попри значні переваги, Komis Auto має й певні недоліки. Висока вартість ліцензування, впровадження та інтеграції може бути суттєвим фінансовим бар'єром для невеликих автосалонів. Хоча інтерфейс системи зручний, співробітникам може знадобитися час для повного освоєння її функціоналу, що може тимчасово вплинути на продуктивність. Залежність від інтеграцій також може стати проблемою, особливо якщо платформа не підтримує популярні сервіси та інструменти, що ускладнює робочі процеси. Крім того, обмеження у можливостях масштабування може створювати труднощі для компаній, що активно розвиваються, оскільки платформа може не відповідати їхнім зростаючим потребам, що змусить шукати інші програмні рішення.



Рисунок 1.3 - Логотип платформи Komis Auto

Відштовхуючись від отриманих даних, які були наведені вище, складемо порівняльну характеристику (табл. 1.1).

Порівняльна таблиця характеристик

Характеристика	Boss Auto	Auto Terminal	Komis Auto	Власний вебдодаток
Зручність навігації	Чітка та зрозуміла	Потребує доопрацювання	Відносно зрозуміла, але потребує часу щоб розібратися	Все знаходиться в зручному місці. Містяться підказки.
Зручність інтерфейсу	+	-	+/-	+
Можливість авторизуватися	+	-	-	+
Наявність підтримки	+	-	+	+
Наявність рекомендацій	+/-	-	-	-
Конфіденційність даних	+	-	-	+
Незалежність від сторонніх сервісів	+/-	-	-	+

Розробка додатку з таким функціоналом є досить унікальною, адже більшість досліджених рішень не повністю відповідають критеріям, визначеним для розробки.

Створений додаток надасть користувачам з усього світу можливість легко взаємодіяти з процесом вибору автомобілів. Вони зможуть переглядати доступні пропозиції, отримувати детальну інформацію про історію та стан транспортних засобів, встановлювати контакт із продавцями та здійснювати покупки у зручний спосіб.

Особливу увагу буде приділено конфіденційності, адже система передбачає обробку фотографій, особистих даних та іншої чутливої інформації.

Витік таких даних може призвести до серйозних наслідків, тому передача конфіденційної інформації стороннім сервісам або особам буде виключена.

Таким чином, за результатами аналізу наявних платформ, було прийнято рішення створити власний додаток. Основними пріоритетами є забезпечення зручності у використанні та стабільної роботи продукту.

1.3. Вибір архітектури обміну інформацією

Застосунок складається з двох ключових складових: клієнтської та серверної частин. Серверна частина реалізована у вигляді API, який відповідає за передачу даних до клієнтської частини. API (Application Programming Interface) – це сукупність правил та протоколів, що забезпечують взаємодію між програмами. Він регламентує, які функції можна викликати, які дані передаються і в якому форматі.

У процесі роботи з API формується запит до сервера, що виконує певну функцію. Запит передається через HTTP або інші протоколи, а сервер у відповідь надсилає дані, необхідні для роботи програми.

До ключових компонентів API належать точка доступу (Endpoint), яка представляє собою URL-адресу для звернення до сервера з метою виконання функції або отримання ресурсу. Крім того, API використовує HTTP-методи: GET – для отримання даних, POST – для їх передачі, PUT – для оновлення, і DELETE – для видалення. Параметри запиту додають додатковий контекст, наприклад, ідентифікатор користувача чи фільтри для вибору інформації. Обмін даними між системами здійснюється у форматах JSON [6] або XML [7], які є найбільш поширеними. Для забезпечення безпеки API може вимагати автентифікацію за допомогою токенів, ключів API або інших механізмів.

Основна архітектура включає клієнтську частину, яка відповідає за відображення інформації користувачеві, контролер, що обробляє запити від клієнта, і API, який забезпечує взаємодію з базою даних і передачу необхідних даних. Такий підхід дозволяє досягти ефективності, масштабованості та забезпечити безпеку системи.

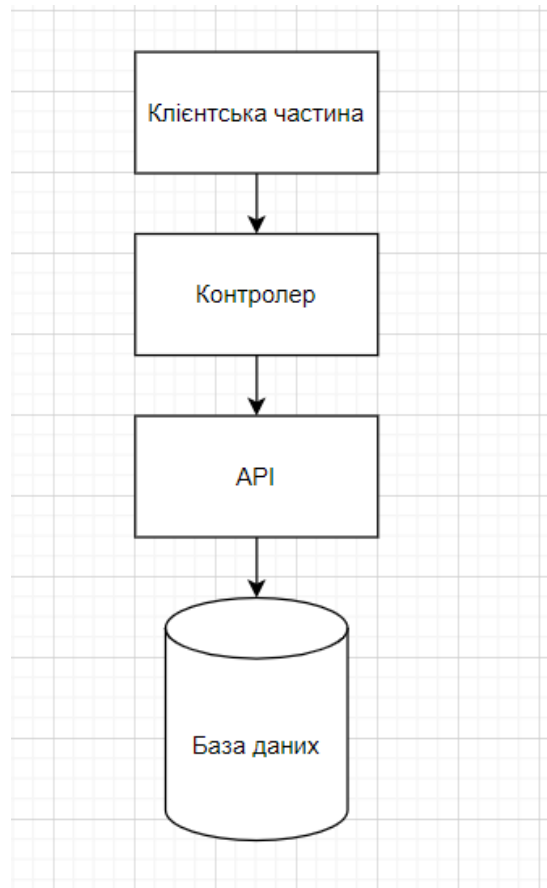


Рисунок 1.4 - Взаємодія компонентів з БД

При виборі архітектурного рішення було враховано переваги й недоліки декількох підходів. Для розробки вебсервісу можливим є використання таких архітектур: на основі компонентів, із застосуванням контейнерів і компонентів, або із впровадженням стейт-менеджерів, таких як Redux.

Архітектура, що базується на компонентах, передбачає, що компоненти React [8] відповідають за відображення даних і взаємодію з користувачем. Вони здійснюють запити до API для отримання необхідної інформації, а також надсилають дані на сервер через API, наприклад, для створення чи оновлення ресурсів.

Архітектура з контейнерами і компонентами передбачає розподіл обов'язків між двома типами компонентів. Контейнерні компоненти відповідають за логіку роботи з API, виконують серверні запити, отримують дані та передають їх презентаційним компонентам через властивості (props). Презентаційні компоненти, своєю чергою, не мають власного стану та відповідають лише за відображення отриманих даних, забезпечуючи простий і зрозумілий інтерфейс.

Архітектура із використанням Redux або інших стейт-менеджерів, таких як MobX, забезпечує централізоване управління станом додатка за допомогою Redux Store. Усі отримані через API дані зберігаються в Store. React-компоненти підписуються на зміни в Store і автоматично оновлюють свої дані, синхронізуючи стан програми з її інтерфейсом.

Після порівняльного аналізу згаданих підходів і вимог додатку було обрано архітектуру з контейнерами та компонентами. Основна ідея такого підходу полягає у відокремленні бізнес-логіки від представлення даних. Контейнерні компоненти відповідають за отримання даних із зовнішніх джерел (наприклад, через API), їх обробку та передачу презентаційним компонентам.

Наприклад, контейнерний компонент може виконати запит до API за допомогою таких інструментів, як Fetch [10], отримати дані, виконати їх обробку (наприклад, фільтрацію чи сортування) та передати їх презентаційним компонентам через властивості (props [11]). Презентаційні компоненти відповідають за відображення отриманих даних і взаємодію з користувачем, використовуючи передані через props дані для побудови інтерфейсу.

Оскільки ця архітектура базується на використанні React-компонентів, додаток отримує додаткові переваги від цієї бібліотеки:

React забезпечує чітку й структуровану архітектуру, що спрощує читання та підтримку коду.

Компоненти, створені для конкретних завдань, легко інтегруються в інші частини додатку, що знижує витрати часу на розробку нових функцій.

Архітектура з використанням контейнерів і компонентів (рис. 1.5) має значні переваги, зокрема позитивний вплив на продуктивність і надійність додатка.

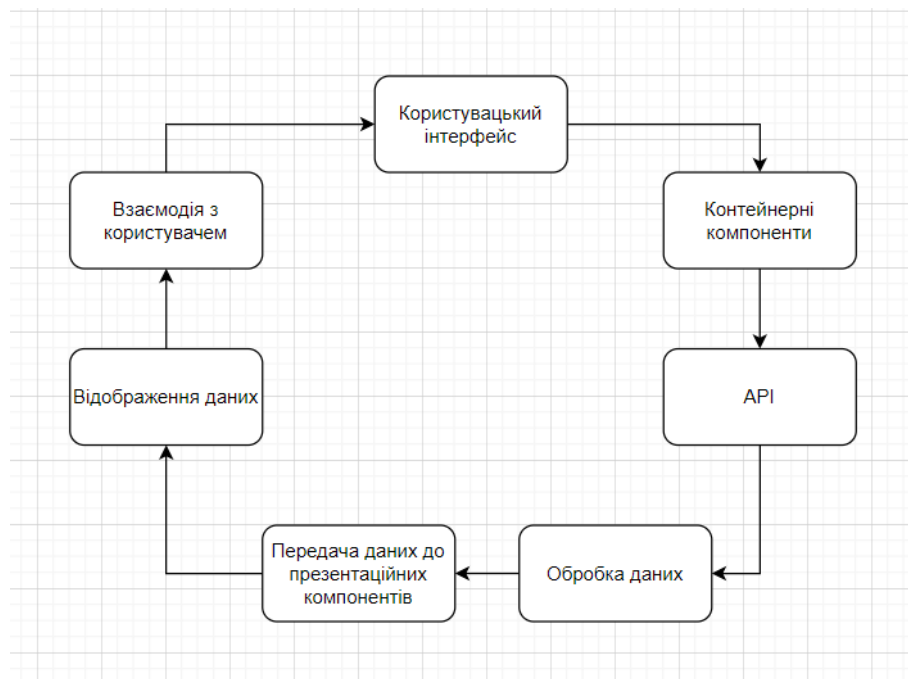


Рисунок 1.5 - Архітектура з використанням контейнерів і компонентів

Процес передачі даних складається з кількох ключових етапів. Спершу створюються презентаційні компоненти, які відповідають за відображення інформації та взаємодію з користувачем. Потім розробляються контейнерні компоненти, що забезпечують отримання даних через API, їх обробку та передачу до презентаційних елементів.

Для отримання даних застосовується API, який взаємодіє із зовнішніми джерелами. У цьому процесі використовуються такі бібліотеки, як Axios або Fetch, які дозволяють виконувати HTTP-запити. Отримані дані обробляються контейнерними компонентами: їх можна фільтрувати, сортувати чи групувати залежно від потреб застосунку.

Після обробки дані передаються до презентаційних компонентів через властивості (props), де вони відображаються у зручному для користувача форматі. Крім того, презентаційні компоненти забезпечують взаємодію з користувачем, обробляючи події, такі як натискання кнопок, введення тексту чи

вибір пунктів меню. Ці події викликають відповідні функції або методи для виконання необхідних операцій.

Архітектура, що поєднує контейнери та компоненти (рис. 1.6), працює як система послідовно з'єднаних модулів, що забезпечує ефективну взаємодію користувача із додатком. Користувач ініціює дії, взаємодіючи із застосунком, а події передаються контейнерним компонентам, які викликають потрібні функції або методи.

Контейнерний компонент надсилає запит до зовнішнього API, використовуючи спеціалізовані бібліотеки. API обробляє запит і повертає відповідь із необхідними даними. Контейнерний компонент оновлює стан системи, передаючи отриману інформацію презентаційним компонентам, які відображають ці дані на екрані, роблячи їх доступними для користувача.

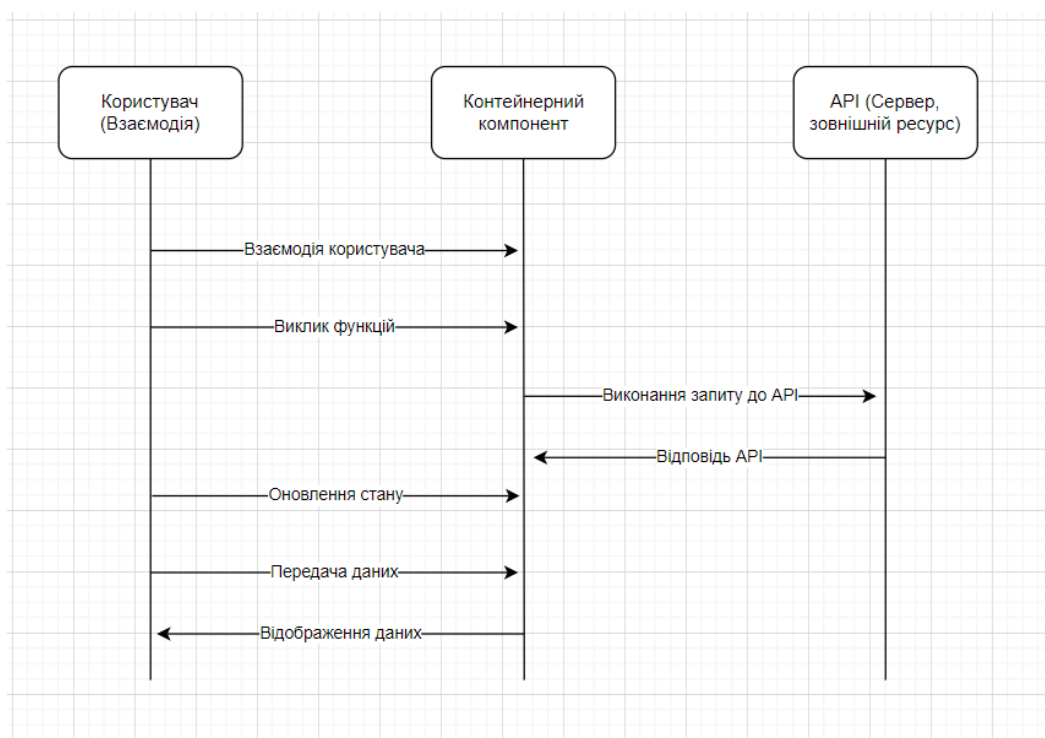


Рисунок 1.6 - Архітектура з використанням контейнерів і компонентів взаємодія з користувачем

Описані компоненти: вебсервер, клієнтський додаток (React додаток) та база даних з'єднуються між собою за допомогою двонаправленого зв'язку між вебсервером та іншими частинами системи (рис. 1.7).

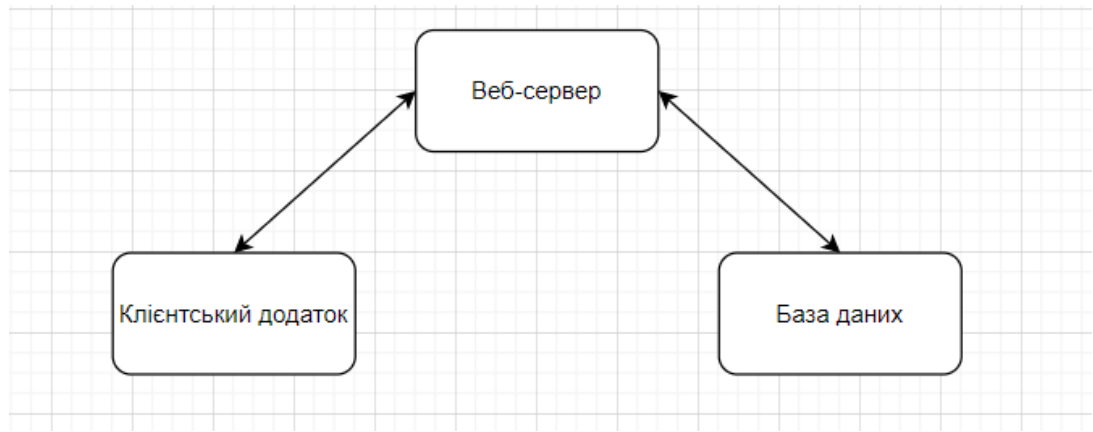


Рисунок 1.7 - Взаємодія компонентів

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

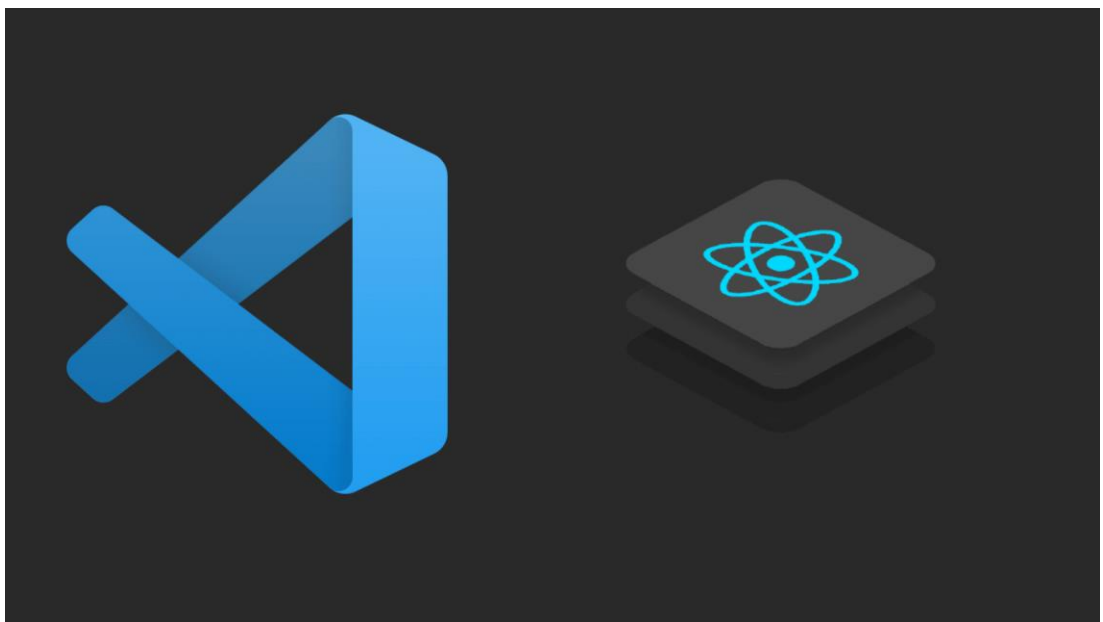


Рисунок 1.8 – Середовище розробки та бібліотека React

Розробка кожного компонента додатку потребує використання певних інструментів. Створення клієнтської частини здійснюється в інтегрованому середовищі розробки Visual Studio Code [12]. Для реалізації клієнтської сторони були обрані наступні інструменти:

1. React – JavaScript-бібліотека, яка використовується для розробки інтерфейсів користувача [10].

2. React Router DOM[13] - розширення бібліотеки React, що забезпечує функціонал маршрутизації для вебдодатків. Воно дозволяє створювати односторінкові додатки (Single-Page Applications, SPA) із динамічною багатосторінковою навігацією без необхідності перезавантаження сторінки.

3. Google-map-react [14] – бібліотека, що надає можливості стилізації інтерфейсу в стилі Material Design для застосунків, створених на основі React.

4. @supabase/auth-helpers-react [15] бібліотека, яка містить набір інструментів і компонентів для автентифікації в React-додатках за допомогою платформи Supabase.

Обрані інструменти дозволяють створювати сучасні й зручні клієнтські додатки, використовуючи передові технології, що гарантують високу якість програмного продукту та його надійність.

Розробка серверної частини виконувалася в середовищі Visual Studio з використанням локального сервера.

Запити до бази даних обробляються за допомогою Supabase:

Supabase [16] - це відкрита платформа, яка забезпечує просте розгортання й управління базами даних. Вона надає інфраструктуру для швидкої та зручної розробки серверних додатків із високою функціональністю та безпекою.

Для забезпечення стабільної роботи системи, високої швидкодії та ефективного використання апаратних ресурсів було визначено низку ключових вимог:

1. Вимоги до локального серверу:

- 1.1. Оперативна пам'ять - 8192 MB
- 1.2. Жорсткий диск - 512 GB
- 1.3. Процесор з частотою від 2 GHz
- 1.4. Браузер

2. Робочої станція розробника:

- 2.1. Оперативна пам'ять - 8192 MB
- 2.2. Операційна система – Windows 8/10
- 2.3. Процесор з частотою від 2 GHz

2.4. Браузер

2.5. Жорсткий диск (бажано ssd або M2) - 256 GB

3. Робоча станція користувача:

3.1. Гаджет з браузером, який відповідає наступним вимогам:

3.1.1. Chrome 60+

3.1.2. Safari 10+ / iOS Safari 10+

3.1.3. Edge 12+

3.1.4. Firefox ESR+

3.1.5. Internet Explorer 11

3.1.6. Opera

Термін реакції програми на дії користувача 0,3 с.

Час виконання команд меню не більше 4 с.

Відображення даних за запитом не більше 10 секунд.

Доступність БД – 90% цілодобово.

Операції з'єднання з БД не більше 15 секунд.

Програма повинна функціонувати при безперебійній роботі ПК.

Висновки до першого розділу

Використання React надає можливість створювати гнучкий, ефективний і легко масштабований інтерфейс додатку. Завдяки компонентному підходу, React дозволяє розділяти інтерфейс на окремі незалежні компоненти, що спрощує управління кодом і його підтримку. Крім того, широка спільнота розробників React забезпечує доступ до великої кількості документації та готових рішень для різноманітних завдань.

API (Application Programming Interface) є ключовою складовою будь-якого сервісу, оскільки забезпечує взаємодію між фронтендом (побудованим на React) і бекендом (серверною частиною). Це дозволяє обмінюватися даними, здійснювати керування замовленнями й інтегрувати функції між різними частинами додатку.

Supabase — це сучасна, готова до використання база даних, яка значно спрощує управління даними. Її використання забезпечує швидке виконання запитів і операцій з даними. Supabase також надає вбудовані інструменти для автентифікації та авторизації користувачів, що є важливим для додатків, які потребують обмеженого доступу до певних функцій або конфіденційної інформації.

Загалом, побудова архітектури додатку на основі React, API та Supabase дозволяє створити потужний і масштабований магазин із сучасним і ефективним інтерфейсом. Ці інструменти забезпечують зручне управління базою даних і взаємодію з зовнішніми сервісами, що робить додаток конкурентоспроможним і здатним задовольнити потреби як розробників, так і кінцевих користувачів.

РОЗДІЛ 2. РОЗРОБКА ТА ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ДЛЯ ПЛАТФОРМИ ПОШУКУ АВТОМОБІЛІВ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Додаток об'єднує в єдиній системі взаємодію звичайних користувачів та користувачів з правами адміністратора. Для наочного представлення інформаційного обміну між елементами системи була створена діаграма варіантів використання (рис. 2.1).

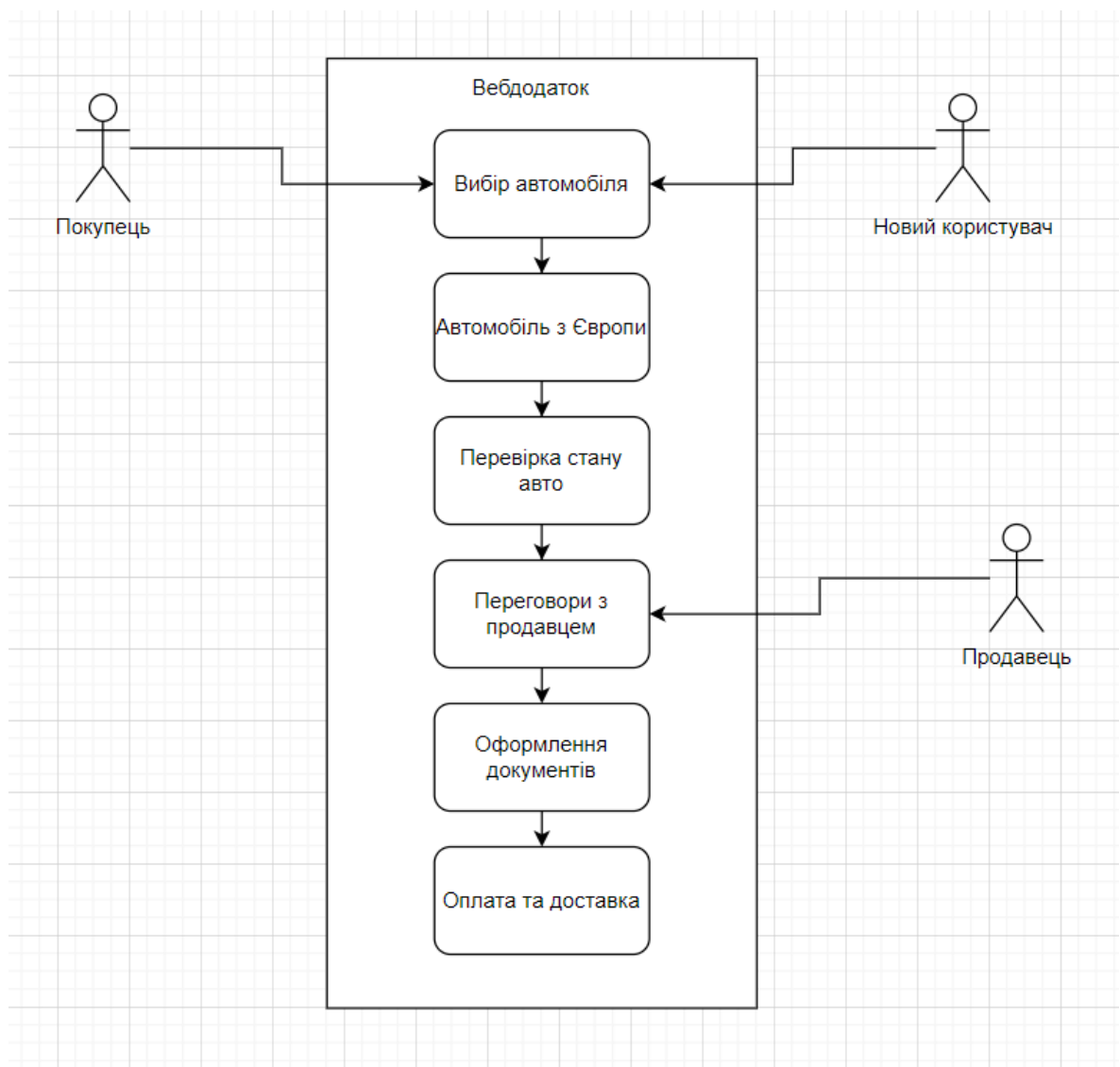


Рисунок 2.1 - Діаграма варіантів використання

Опис кожного варіанта використання наведено у відповідних таблицях (табл. 2.1 – 2.12).

Таблиця 2.1

Опис варіанту - вибір авто

Назва варіанту використання	Вибір автомобіля
Основні актори	Покупець, вебдодаток
Короткий опис	Покупець розглядає варіанти автомобілів з країн Європи

Таблиця 2.2

Опис варіанту - автомобіль з Європи

Назва варіанту використання	Автомобіль з Європи
Основний актор	Покупець, вебдодаток
Короткий опис	Кожен автомобіль представляє окремий актор інформації про автомобілі з певної країни ЄС.

Таблиця 2.3

Опис варіанту - перевірка стану авто

Назва варіанту використання	Перевірка стану авто
Основний актор	Покупець, вебдодаток
Короткий опис	Покупець проводить перевірку технічного стану автомобіля

Таблиця 2.4

Опис варіанту - прееговори з продавцем

Назва варіанту використання	Переговори з продавцем
Основний актор	Покупець, вебдодаток, продавець
Короткий опис	Покупець вступає в переговори з продавцем.

Таблиця 2.5

Опис варіанту - оформлення документів

Назва варіанту використання	Оформлення документів
Основні актори	Покупець, вебдодаток, продавець
Короткий опис	Після досягнення угоди з продавцем, покупець займається оформленням необхідних документів.

Таблиця 2.6

Опис варіанту - оплата та доставка

Назва варіанту використання	Оплата та доставка
Основні актори	Покупець, вебдодаток, продавець
Короткий опис	Покупець здійснює оплату автомобіля та організовує його доставку.

Діаграма варіантів ілюструє послідовність дій, які покупець виконує під час придбання автомобіля. Процес починається з вибору автомобіля, а завершується оплатою та доставкою. Кожен етап представлений окремим актором, що виконує конкретну роль у процесі продажу.

Щоб система відповідала необхідним потребам, для продукту були визначені та сформульовані відповідні вимоги.

Бізнес-вимоги:

1. Основні цілі: створення додатку для спрощення процесу купівлі чи продажу автомобілів.
2. Консультації: користувачі зможуть звернутися до гарячої лінії для отримання допомоги у використанні додатку чи з інших питань.
3. Представлення проекту: проект буде розроблений у вигляді вебсайту.

Вимоги користувачів:

1. Користувачі:
 - 1.1. Переглядати оголошення автомобілів.

- 1.2. Фільтрувати оголошення за різними критеріями.
- 1.3. Переглядати рекомендовані оголошення.
- 1.4. Відстежувати авто за VIN-номером [17].
- 1.5. Використовувати калькулятор для розрахунку загальної вартості розмитнення авто.

- 1.6. Звертатися на гарячу лінію та ознайомлюватися з інформацією про компанію.

2. Користувачі з правами адміністратора:

- 2.1. Переглядати оголошення автомобілів.
- 2.2. Додавати вподобання автомобілів для полегшення формування рекомендацій для конкретного користувача.

- 2.3. Фільтрувати оголошення за різними критеріями.

- 2.4. Відстежувати авто за VIN-номером.

- 2.5. Використовувати калькулятор для розрахунку загальної вартості розмитнення авто.

- 2.6. Звертатися на гарячу лінію та переглядати інформацію про компанію.

- 2.7. Додавати нові оголошення, ввівши відповідний код у поле для цієї послуги.

- 2.8. Редагувати та видаляти існуючі оголошення.

Вхідні дані: дані про автомобілі (характеристики, VIN-номер), інформація про користувача та зображення автомобілів.

Нефункціональні вимоги:

1. Сприйняття:

- 1.1. Час ознайомлення з додатком має становити до 15 хвилин для новачків та до 10 хвилин для досвідчених користувачів.

- 1.2. Система повинна відповідати на більшість запитів за 5 секунд, а на складні запити — до 7 секунд.

- 1.3. Інтерфейс має бути сучасним та інтуїтивно зрозумілим.

2. Надійність

2.1. Кількість помилок у роботі системи не повинна перевищувати 1 на 1000 запитів.

3. Продуктивність

3.1. Додаток має стабільно працювати навіть при одночасному використанні великою кількістю користувачів.

Для розробки системи обрано багатoshарову архітектуру, основними структурними елементами якої є: моделі-сутності, моделі-прив'язки, менеджери, API, допоміжні методи та контролери.

Діаграма класів, яка ілюструє роботу додатку, представлена на рисунку 2.2.

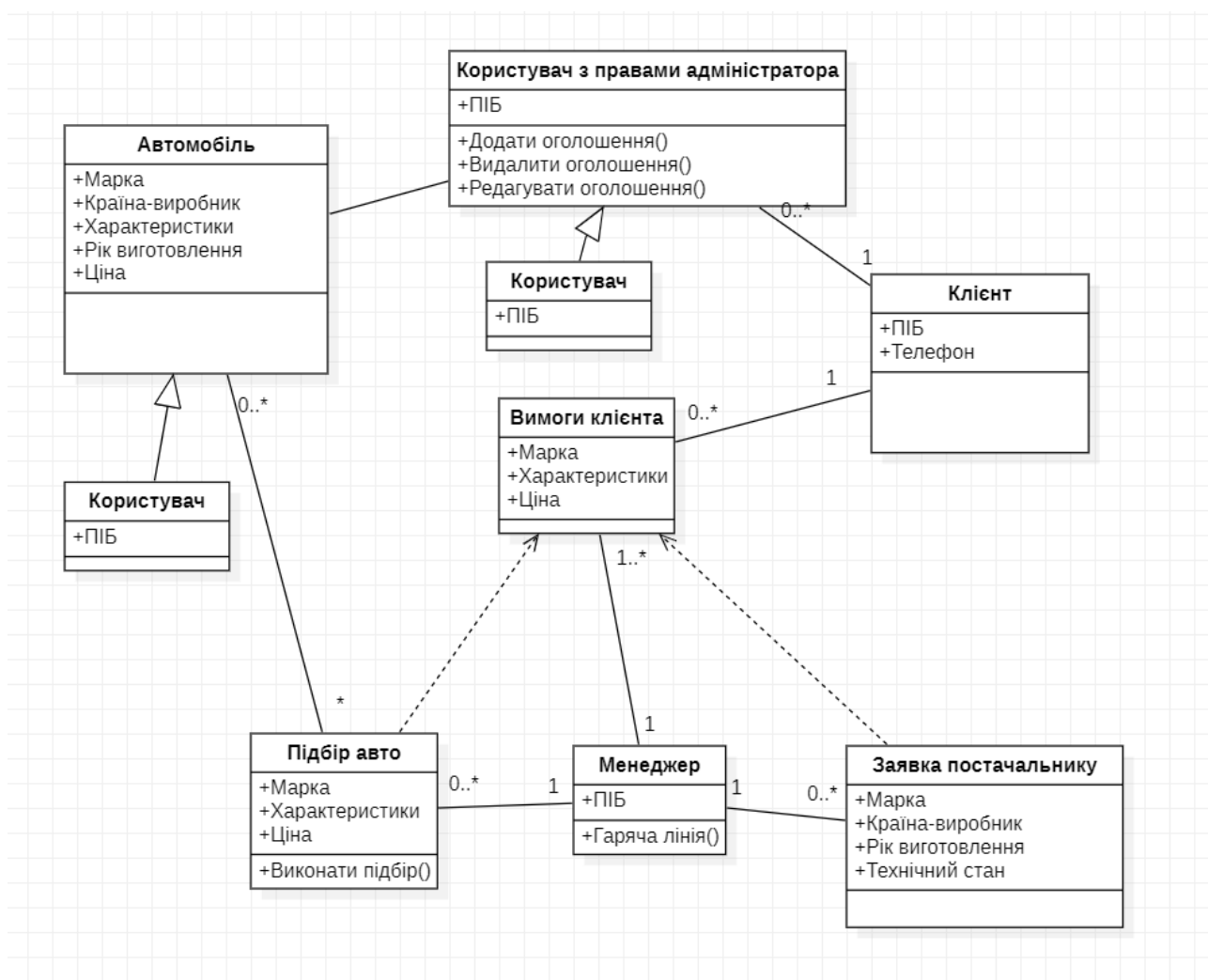


Рисунок 2.2 - Діаграма класів моделей-сутностей

Автомобіль – всі оголошення автомобілів, зібрані в базі даних, які відображаються користувачам при перегляді наявних авто. Кожен автомобіль пов'язаний із продавцем, який є користувачем додатку.

Користувач та користувач з правами адміністратора – схожі типи користувачів, основна відмінність полягає в можливостях. Користувач з правами адміністратора має додаткові функції: додавання оголошень, їх редагування та видалення.

Клієнт – це особа, яка пред’являє вимоги до пошуку автомобіля і очікує зворотного зв’язку від компанії для подальшої співпраці.

Підбір авто – це процес, який виконується під час фільтрування оголошень автомобілів за визначеними критеріями.

Менеджер – забезпечує взаємодію з клієнтами через гарячу лінію для надання консультацій та вирішення питань.

Заявка постачальнику – передається менеджером, який відповідає за зв’язок із постачальником для уточнення деталей чи замовлення автомобіля.

2.2 Розробка бази даних системи

Для збереження сутностей та організації взаємодії з даними в системі вебдодатку було розроблено базу даних. Як платформу для управління базою даних обрано **Supabase**, яка забезпечує зручну інтеграцію з вебдодатком, швидке виконання запитів та управління даними.

Відповідно до архітектури додатку, взаємодія між вебдодатком і базою даних здійснюється через API, який виконує запити до БД і повертає оброблені дані в зручному форматі.

База даних системи складається з 2 таблиць:

1. Cars – таблиця, яка зберігає дані про пости в системі.
2. user_interactions – таблиця, яка зберігає дані вподобань

Нижче наведений опис кожної з наведених таблиць бази даних (див. табл. 2.13 – 2.14)

Таблиця 2.7

Структура бази даних Cars

Cars		
Поле	Тип	Призначення

Id	int	Id оголошення
model	varchar	Модель авто
year	int	Рік
brand	varchar	Бренд
transmission	varchar	Привід автомобіля
type_of_transport	varchar	Тип транспорту
origin	varchar	Регіон
city	varchar	Місто
car_price	numeric	Ціна
color	varchar	Колір авто
user_id	text	Порядковий номер користувача
vin_num	text	Номер авто
img_link	text	Посилання на вільний доступ
range	numeric	Пробіг
engine	varchar	Двигун
fuel	varchar	Паливо
comfort	varchar	Комфорт
multimedia	varchar	Мультимедія
security	varchar	Сигналізація
wheel_drive	varchar	Привід
customs_cleared	varchar	Розмитнення
injured	varchar	Пошкодження
car_description	text	Опис до автомобіля
created_at	varchar	Дата створення
img_url	text	Посилання на картинку

Таблиця 2.8

Структура бази даних user_interactions

user_interactions		
Поле	Тип	Призначення
id	int	Порядковий номер вподобання
user_id	varchar	Порядковий номер користувача

car_id	int	Порядковий номер оголошення
interaction_type	text	Вподобання
timestamp	timestamp	Дата додавання

2.3 Проєктування та реалізація алгоритмів роботи

Оскільки наш вебдодаток складається з трьох основних компонентів (клієнтський додаток, сервер та база даних), критично важливо забезпечити ефективну взаємодію між ними. На рисунку 2.10 представлено діаграму активності, яка відображає послідовність дій при створенні оголошення та подальшій роботі з ним. Цей процес включає наступні кроки:

1. Продавець або менеджер ініціюють створення оголошення, заповнюючи відповідну форму в клієнтському додатку.
2. Після заповнення дані валідуються на стороні клієнтського додатку.
3. Після заповнення форми дані перевіряються на відповідність базовим вимогам безпосередньо в клієнтському додатку.
4. Успішно валідовані дані зберігаються в базі даних. Створене оголошення додається до таблиці.
5. Після збереження оголошення клієнтський додаток отримує підтвердження від сервера й відображає користувачу успішно створену пропозицію.
6. Інші користувачі можуть переглядати створене оголошення в своїх рекомендаціях. Вони мають можливість додати оголошення до списку вподобань або відмовитися від нього.

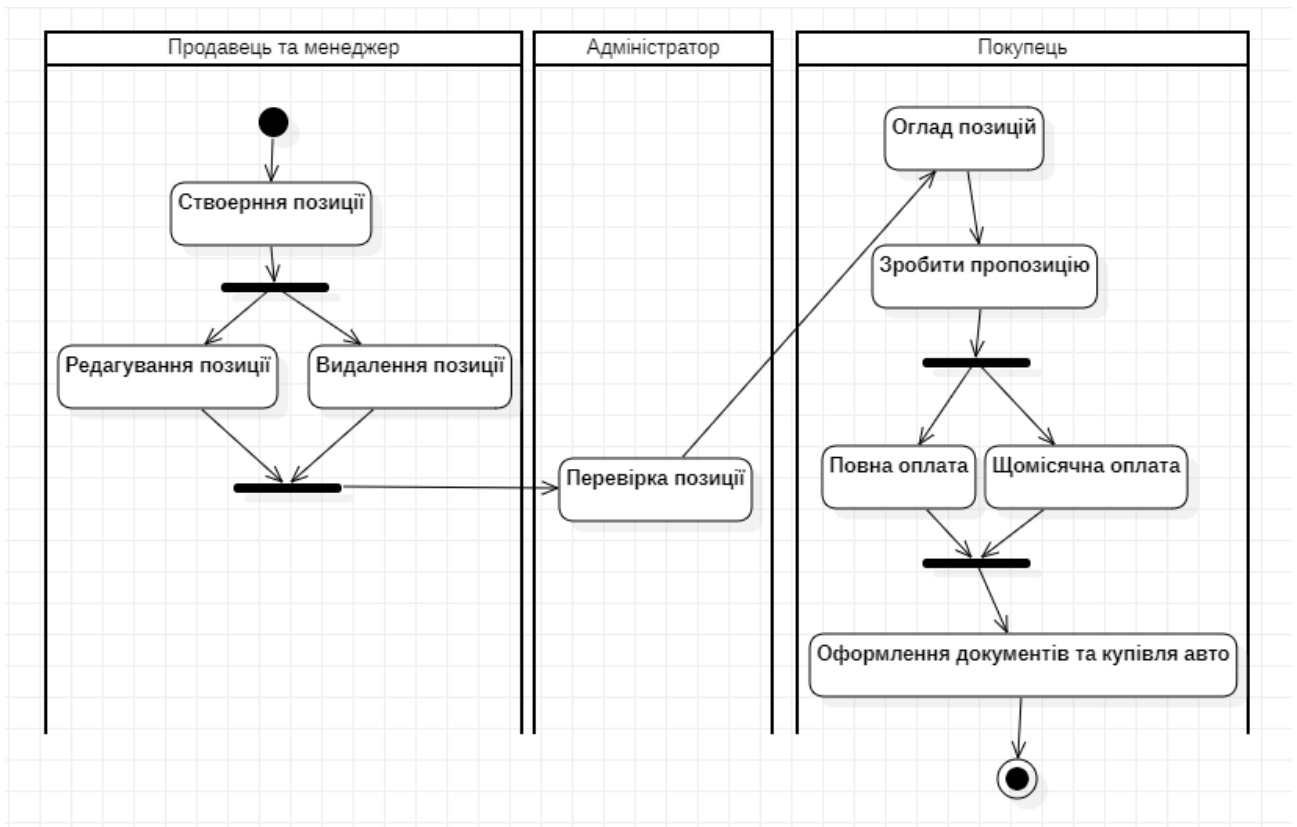


Рисунок 2.3 - Діаграма активності взаємодія користувача з вебдодатком

Отже, отримані результати виконання описаних кроків взаємодія всіх компонентів системи дозволяє досягти пришвидшення процесу пошуку за допомогою фільтрування та покупки авто.

2.4 Реалізація функціоналу в додаток

Розробка цього додатка передбачає створення функціоналу для пошуку та додавання нових оголошень..

Процес визначення параметрів фільтрації починається з вибору характеристик автомобілів, які використовуватимуться для пошуку. Серед основних параметрів: марка, модель, рік випуску, ціновий діапазон, тип кузова, об'єм двигуна, вид палива, пробіг та інші ключові показники. Після цього розробляється інтерфейс фільтра, який дозволяє користувачам задавати ці параметри через випадаючі списки, перемикачі чи текстові поля. Наприклад,

користувач може вибрати марку "BMW", встановити ціну в межах від 10 000 до 20 000 євро та рік випуску автомобіля після 2015 року.

Після введення критеріїв система виконує пошук, звертаючись до бази даних або іншого джерела, для отримання відповідних автомобілів. Відфільтровані результати відображаються у вигляді списку або інтерактивної карти. Також користувач може скинути всі налаштування фільтра, щоб переглянути весь доступний каталог автомобілів. Для розширення можливостей додатку можна передбачити фільтрацію за додатковими характеристиками, такими як країна виробництва, наявність систем безпеки, додаткові функції чи колір кузова.

Функція відстеження за VIN-номером дозволяє інтеграцію з базою даних VIN. Користувач вводить номер через текстове поле, після чого програма перевіряє його формат і здійснює пошук у базі. Якщо номер знайдено, користувач отримує дані про автомобіль, зокрема його марку, модель, рік випуску та технічні характеристики.

Для формування рекомендацій додаток збирає інформацію про вподобання користувача, виключає автомобілі, які не були обрані, і аналізує спільні характеристики моделей, що сподобалися. На основі цих даних обчислюється ступінь схожості інших автомобілів із вподобаними, враховуючи такі критерії, як ціна, рік випуску та вид палива. Автомобілі сортуються за рівнем релевантності, а користувач отримує персоналізований список рекомендацій.

Функція розрахунку загальної суми розмитнення включає обчислення акцизного збору з урахуванням типу палива та об'єму двигуна, митних платежів, які залежать від вартості чи ваги автомобіля, а також підсумовування цих витрат. Підсумкова сума або детальна розбивка витрат відображається для користувача у зручному форматі.

Висновки до другого розділу

Додання нового функціоналу до застосунку для пошуку автомобілів з має значний потенціал для підвищення зручності та ефективності користувацького

досвіду. Пропоновані можливості, зокрема пошук за різними параметрами, підбір актуальних оголошень, перевірка за VIN-кодом та калькулятор розмитнення, надають користувачам ширший вибір інструментів для пошуку та аналізу автомобілів.

Функція пошуку за параметрами дозволяє користувачам швидко знаходити автомобілі, що відповідають їхнім потребам, використовуючи фільтри за брендом, моделлю, роком випуску та іншими характеристиками. Це спрощує процес вибору та допомагає зосередитися на найбільш відповідних варіантах.

Можливість пошуку за VIN-кодом забезпечує зручність для користувачів, які вже мають унікальний ідентифікатор автомобіля. Це особливо актуально для перевірки конкретної моделі або історії транспортного засобу.

Калькулятор розмитнення допомагає оцінити витрати на оформлення автомобіля, враховуючи фактори, такі як вік, вартість, об'єм двигуна та митні ставки. Завдяки цьому користувачі можуть краще планувати бюджет до покупки автомобіля.

Функція рекомендацій на основі індивідуальних уподобань дозволяє персоналізувати пошук, аналізуючи взаємодії користувача із застосунком. Алгоритми враховують вподобання, такі як ціна, тип пального чи рік випуску, щоб пропонувати найбільш релевантні варіанти. Це значно скорочує час на пошук і сприяє більш ефективному вибору.

Усі ці функції спрямовані на підвищення інформованості покупців, забезпечення точності та оперативності, а також полегшення планування витрат. Завдяки цьому процес вибору та придбання автомобілів стає більш зручним та ефективним.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ДОДАТКОМ

3.1 Порядок встановлення та налаштування параметрів системи

Налаштування та запуск системи передбачає розгортання всіх її компонентів та їх подальшу конфігурацію як окремо, так і в рамках цілісної системи. Для ілюстрації архітектури програмного комплексу та опису процесу розгортання створено діаграму розгортання (рис. 3.1).

Ця діаграма дозволяє детально описати компоненти системи та процес їх інтеграції.

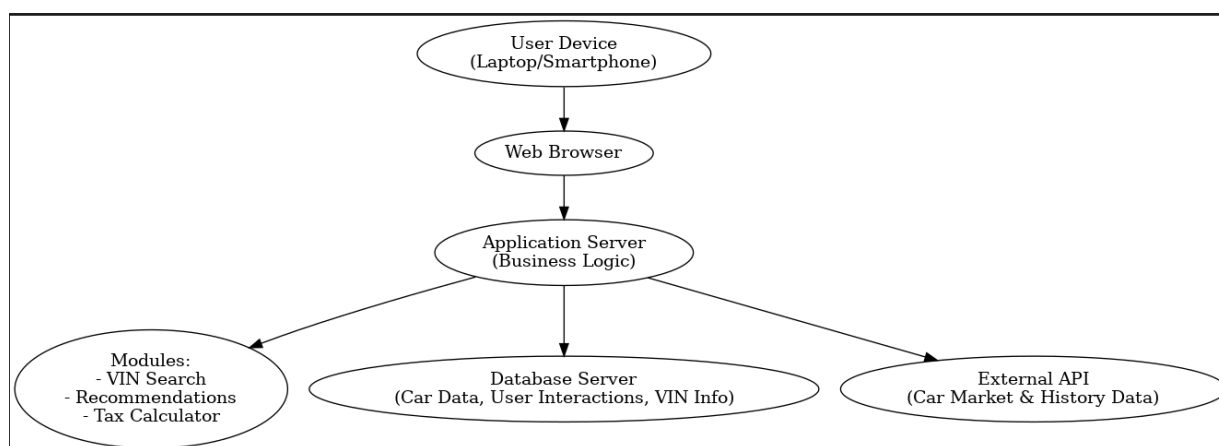


Рисунок 3.1 - Діаграма розгортання додатку

За допомогою створеної діаграми можна детально проаналізувати програмний комплекс компонентів і описати процес його розгортання.

Короткий опис діаграми:

Система складається з кількох взаємопов'язаних компонентів. User Device (Laptop/Smartphone) — це пристрій кінцевого користувача, наприклад, ноутбук або смартфон, на якому функціонує додаток. Користувач взаємодіє із додатком через Web Browser, що завантажує інтерфейс користувача та виконує роль клієнтської частини, яка передає запити до сервера. Запити, надіслані користувачем, обробляються на Application Server (Business Logic) — центральному сервері, де розміщена бізнес-логіка системи. Цей сервер керує даними, обробляє запити та повертає відповіді назад до браузера. Крім того, сервер здійснює взаємодію з іншими компонентами, включаючи спеціалізовані модулі, базу даних і зовнішні API.

Модулі виконують певні спеціалізовані функції, такі як VIN Search, що дозволяє здійснювати пошук інформації за VIN-кодом автомобіля, Recommendations, який генерує персоналізовані рекомендації на основі даних користувача, і Tax Calculator, що забезпечує розрахунок податків для транспортних засобів. Уся інформація про автомобілі, взаємодії користувачів та VIN-коди зберігається на Database Server, який забезпечує збереження та доступ до необхідних даних для функціонування додатка.

Система також використовує External API для отримання ринкової інформації про автомобілі та історії транспортних засобів, що дозволяє розширити обсяг доступних даних. Взаємодія між компонентами забезпечує єдиний робочий процес, коли запит, надісланий через браузер, обробляється сервером, після чого за потреби залучаються відповідні модулі, база даних або зовнішні API, і результат передається назад у браузер для відображення.

Структура інтерфейсу

Взаємодія користувача із системою здійснюється через клієнтський додаток. Після відкриття сервісу у браузері користувач бачить головну сторінку (рис. 3.2), яка є початковим інтерфейсом для доступу до функціоналу системи.

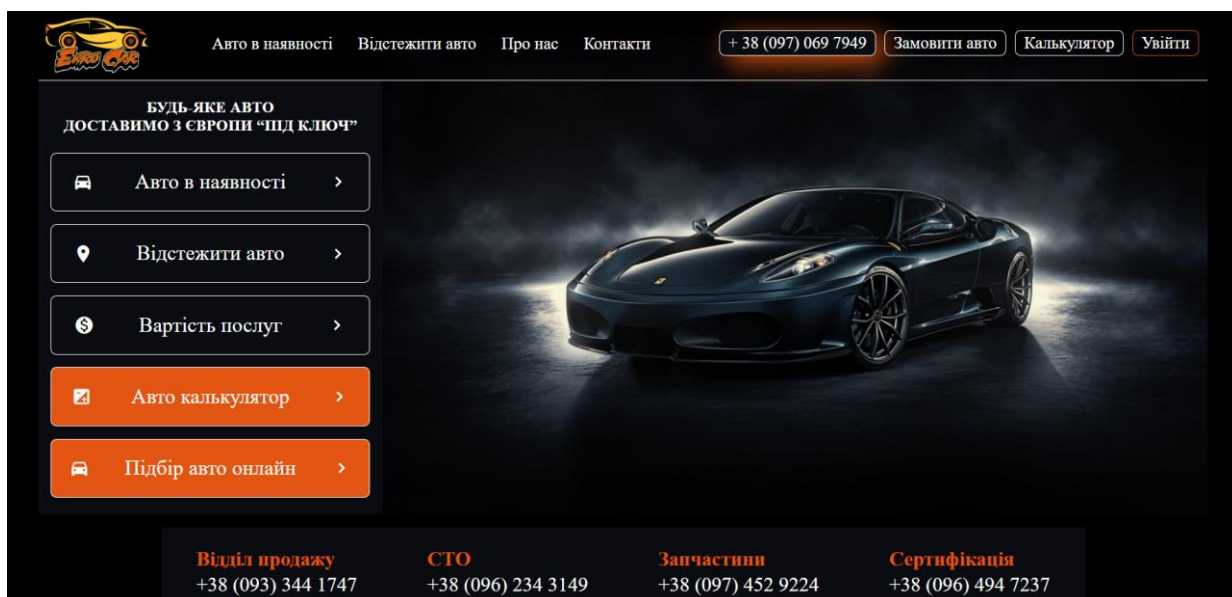


Рисунок 3.2 – Головна сторінка

Після перегляду наданої інформації користувач має можливість скористатися послугами, доступними без авторизації. До таких послуг належать

калькулятор, перевірка автомобіля за VIN-кодом, а також фільтрація наявних автомобілів за різними параметрами (рис. 3.3 – рис. 3.5).

Калькулятор розрахунку авто з Європи

Наш калькулятор допоможе зорієнтуватися, який автомобіль з Європи вибрати для доставки.
Під час доставки з країн Америки, наприклад, береться умовний каталог, де ціни завищені і не враховують невеликого зносу машини. Так ціна розмитнення завищується, і вам неприємно мати справу з такими операціями. Більше й не потрібно – доставка з Європи буде прозорою та простою, тим більше, ми все зробимо самі.

Калькулятор

Ціна авто (USD)

Рік випуску

Тип двигуна

Об'єм двигуна (см3)/Ємність батареї (кВт)

Ціна

Ціна авто	0\$
Ввізне мито	0\$
Акциз	0\$
ПДВ	0\$
Загальна вартість	0\$

Зателефонувати

Даний розрахунок є орієнтовним

Рисунок 3.3 – Калькулятор розмитнення авто

Перевірити автомобіль по VIN

Держ. номер
VIN

В тренді сьогодні

DAEWOO LANOS
2008
KB9999KB

Ваша історія пошуку

DAEWOO LANOS
2008
KB9999KB

Рисунок 3.4 – Відстеження авто по VIN номеру

VIN-код повинен містити 17 символів, тому додаток обмежує введення користувачем, не дозволяючи ввести більше, ніж потрібно (рис. 3.5).

Перевірити автомобіль по VIN

Держ. номер
VIN

Держ. реєстрація

Україна

Chevrolet Camaro
2020

≈ 20000 \$

Рисунок 3.5 – Відстеження авто по VIN номеру

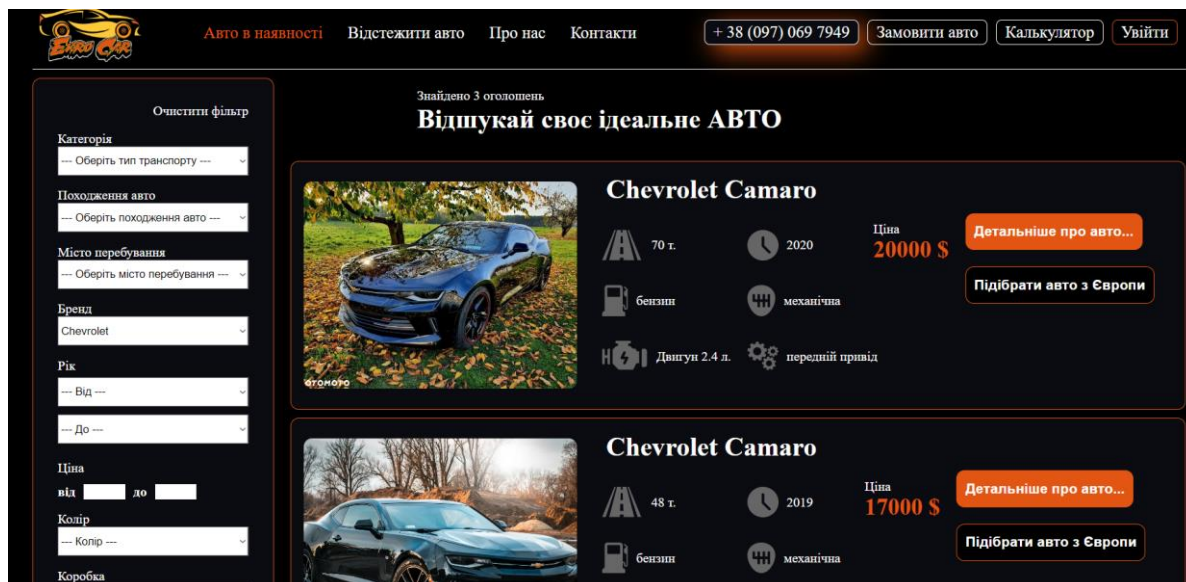


Рисунок 3.6 – Фільтрація наявних автомобілей

У нижній частині сторінки можна переглянути рекомендовані автомобілі, які були відібрані з бази даних на основі їхньої популярності (рис. 3.7).

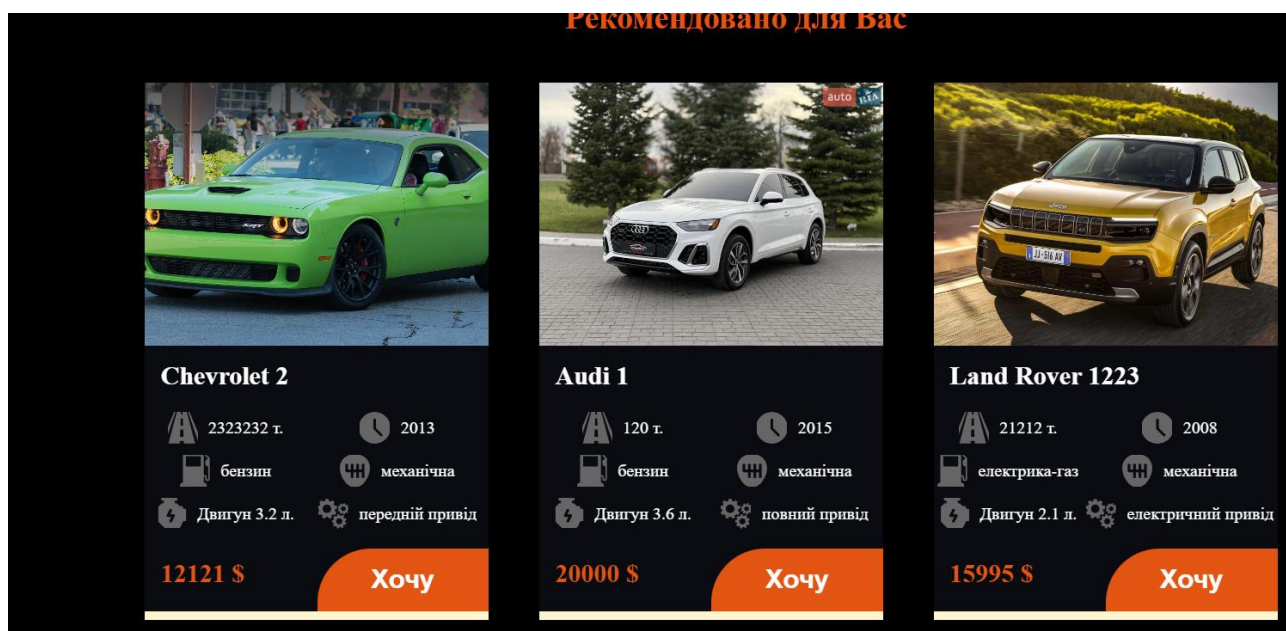


Рисунок 3.7 – Сторінка про компанію

На цій сторінці також доступна інформація про компанію (рис. 3.8 – 3.9).

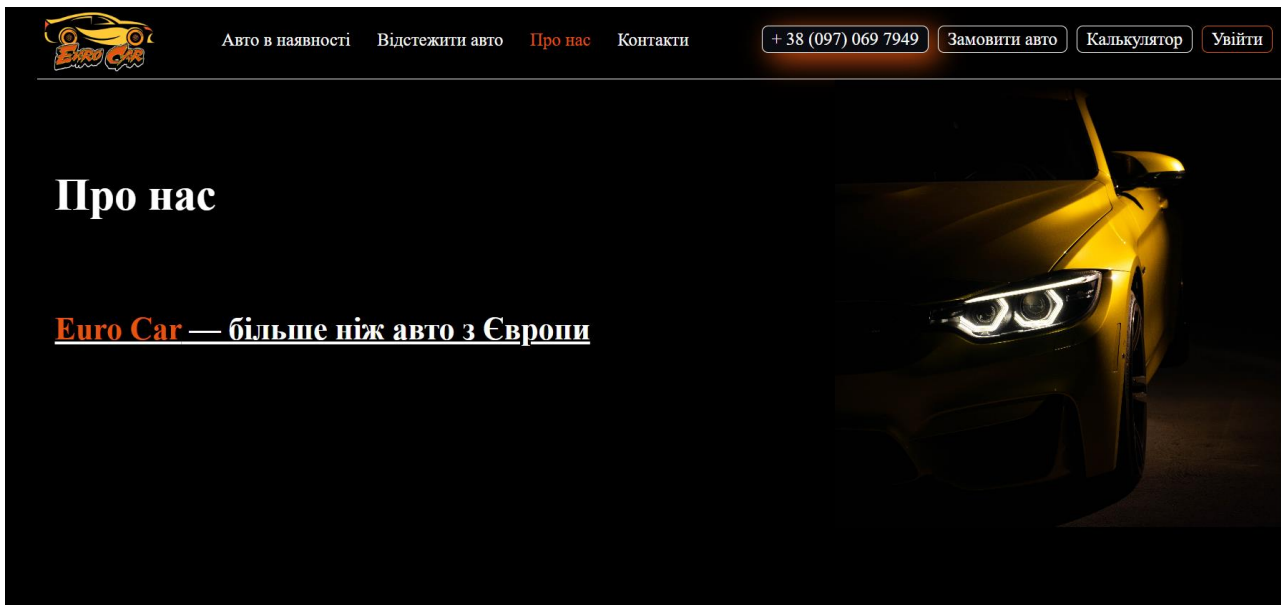


Рисунок 3.8 – Сторінка про компанію

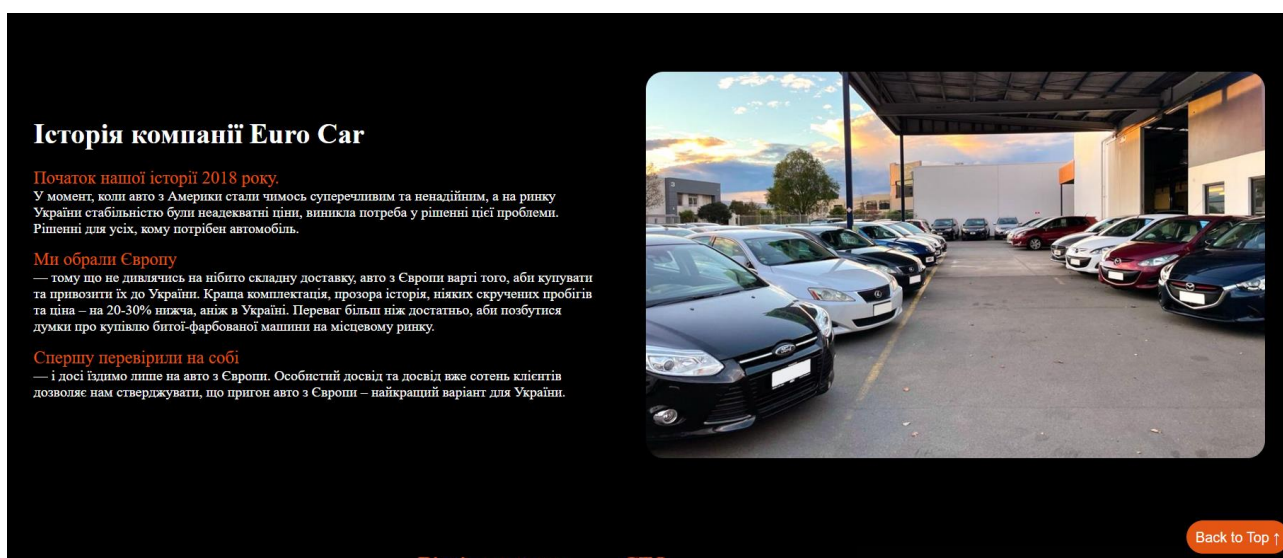


Рисунок 3.9 – Сторінка про компанію

Для перегляду детальної інформації про автомобіль користувач має обрати потрібний елемент зі списку та натиснути кнопку "Детальніше про авто". Після цього відкриється сторінка з детальними характеристиками вибраного автомобіля (рис. 3.10 – 3.11).



[Авто в наявності](#)
[Відстежити авто](#)
[Про нас](#)
[Контакти](#)

[+ 38 \(097\) 069 7949](#)
[Замовити авто](#)
[Калькулятор](#)
[Увійти](#)



Chevrolet Camaro

20000 \$

70т. км
Бензинбензин
Об'єм 2.4л
2020

механічна
передній привід

Опис:

Виробництво нового Сатаго розпочалося 16 березня 2009 року після кількох років перерви з попереднім поколінням, виробництво якого закінчилося в 2002 році. Сатаго V надійшов у продаж в 2009 році. Автомобіль пропонується з двома модифікаціями двигунів: на Сатаго LT — V6 304 к.с., Сатаго SS — V8 на 400 та 427 к.с.

[Отримати консультацію](#)

Рисунок 3.10 - Детальна інформація про авто

[Отримати консультацію](#)

ТЕХНІЧНИЙ ПАСПОРТ:

Походження:	Європа
Місто:	Київ
Марка:	Chevrolet
Модель:	Сатаго
Рік:	2020
Колір:	чорний
Пошкоджений:	Ні
Розмитнений:	Так
Привід:	передній привід
Коробка:	механічна
Паливо:	бензин

КОМПЛЕКТАЦІЯ:

Безпека:	датчик руху
Мультимедіа:	дисплей заднього виду
Комфорт:	клімат-контроль



Euro Car

Відділ продажу
СТО

+38 (093) 344 1747
+38 (096) 234 3149

[Приєднуйся:](#)
[Back to Top ↑](#)

Рисунок 3.11 - Детальна інформація про авто

Сторінка контактної інформації містить номери телефонів і адресу компанії. Користувач може зв'язатися з компанією за допомогою електронної пошти, щоб уникнути тривалого очікування на телефонній лінії. Для зручності передбачено активні посилання, які переадресовують користувача безпосередньо до поштового сервісу або контактної інформації (рис. 3.12 – 3.13).

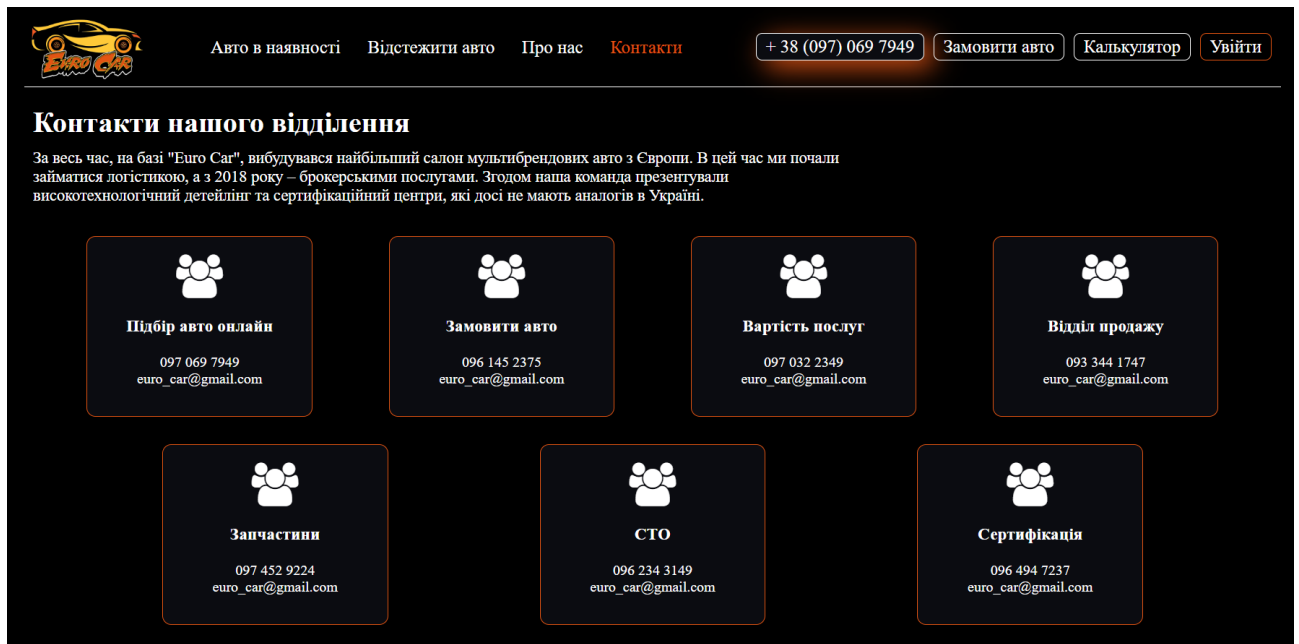


Рисунок 3.12 – Сторінка з контактною інформацією

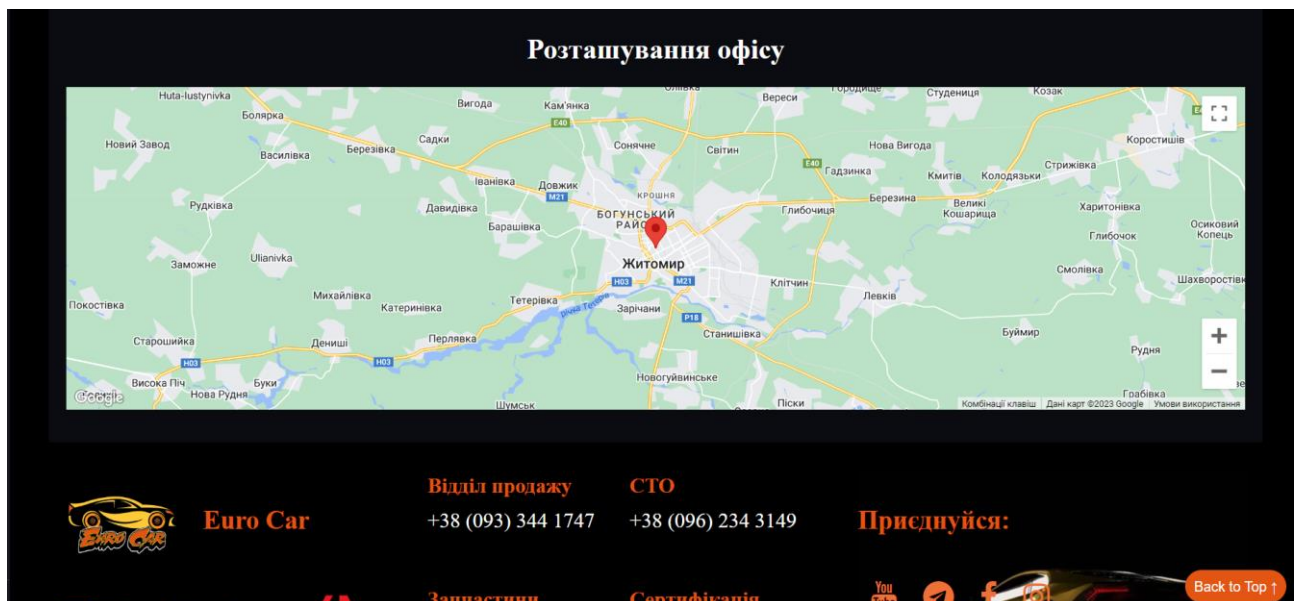


Рисунок 3.13 – Сторінка з контактною інформацією

У разі, якщо користувач бажає ознайомитися з детальною інформацією щодо оформлення замовлення, необхідно виконати авторизацію та увійти до особистого кабінету. Авторизація здійснюється через стандартну сторінку входу (рис. 3.14 – 3.17). Якщо обліковий запис ще не створений, користувач повинен пройти процедуру реєстрації (рис. 3.14 – 3.17). На цих сторінках передбачені підказки для полегшення процесу авторизації та реєстрації.

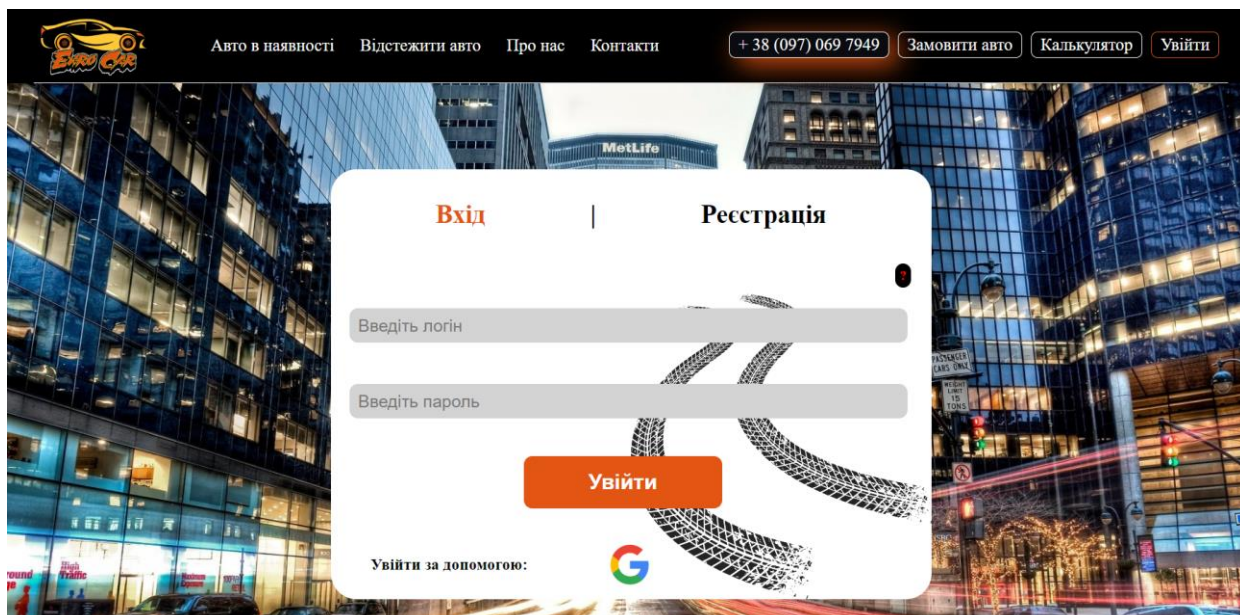


Рисунок 3.14 – Сторінка авторизації

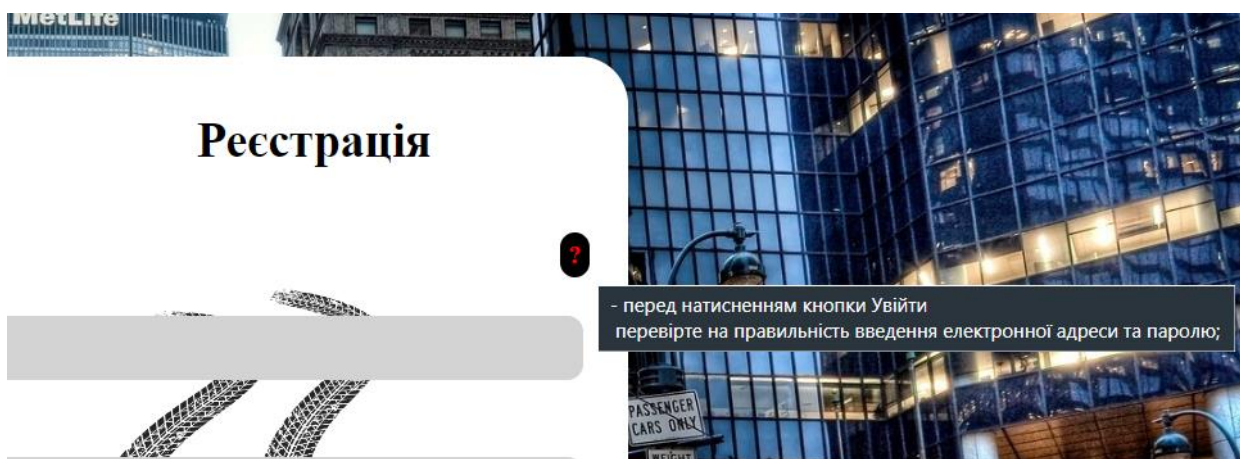


Рисунок 3.15 – Сторінка авторизації з підказкою для користувача

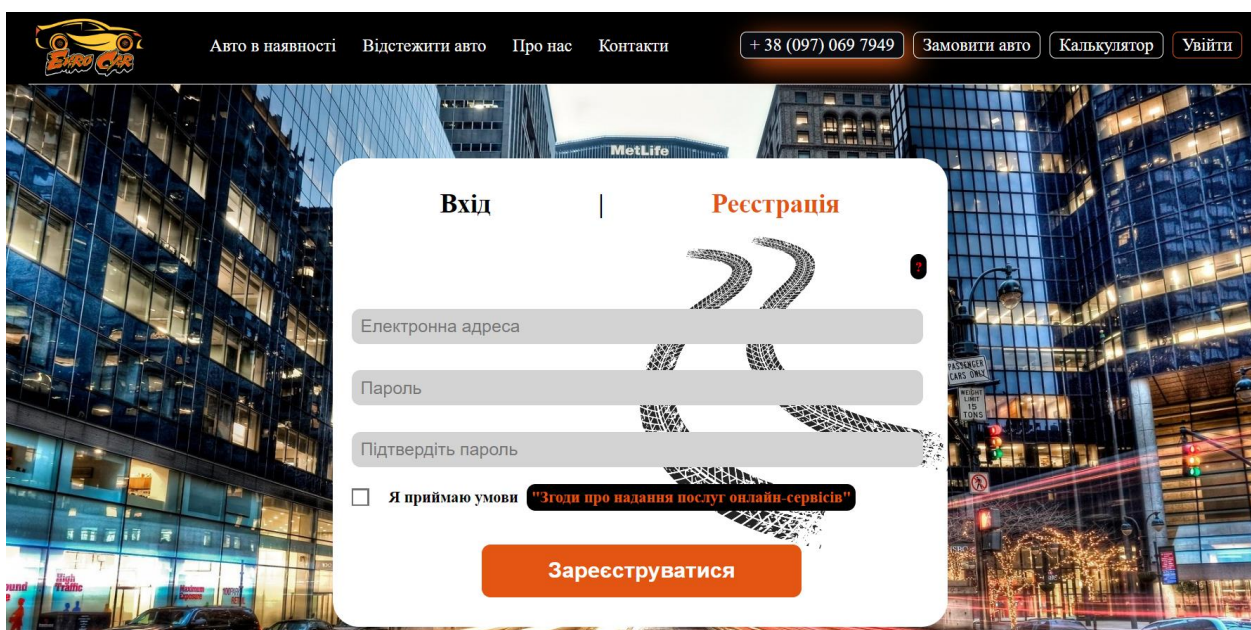


Рисунок 3.16 – Сторінка реєстрації



Рисунок 3.17 – Сторінка реєстрації з підказкою для користувача

Також додаток дозволяє виконати авторизацію через Google. Ця функція є дуже зручною та користується популярністю (рис. 3.18 – 3.19).

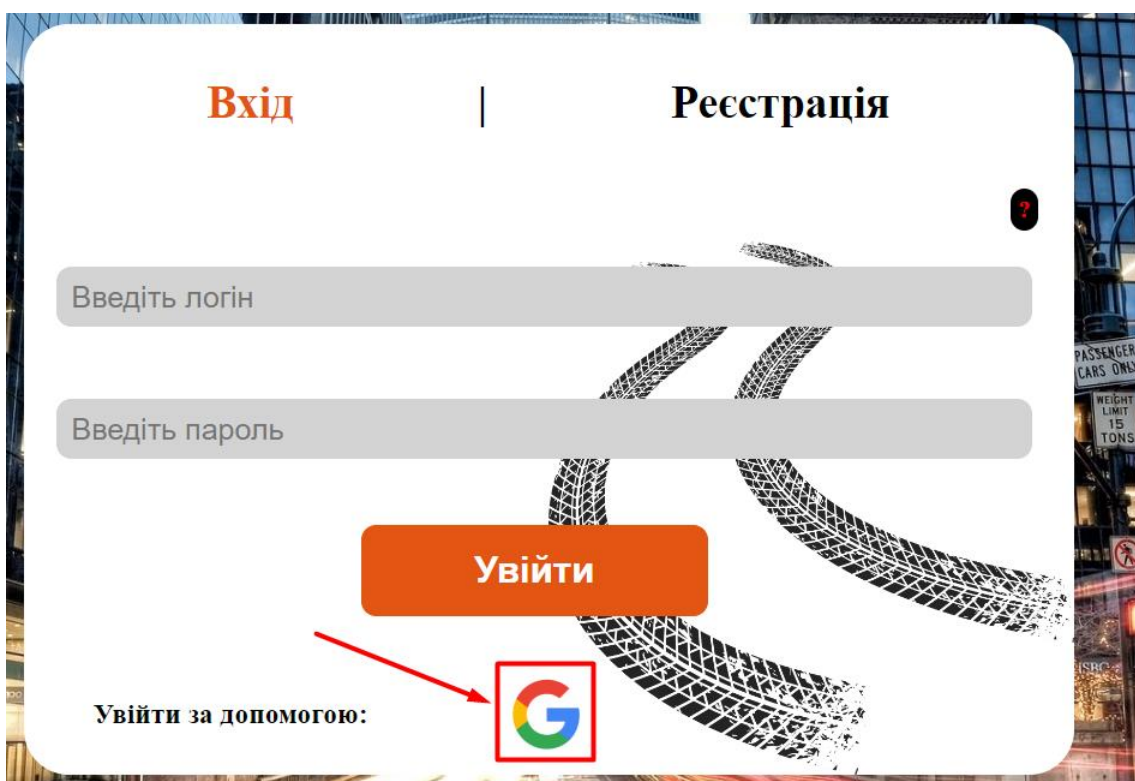


Рисунок 3.18 – Авторизація через Google

Денис Рудик
ipz193_rdo@student.ztu.edu.ua

Рисунок 3.19 – Авторизація через Google

Після успішного входу до системи у верхньому правому куті вебдодатку з'являється аватар користувача (рис. 3.20).

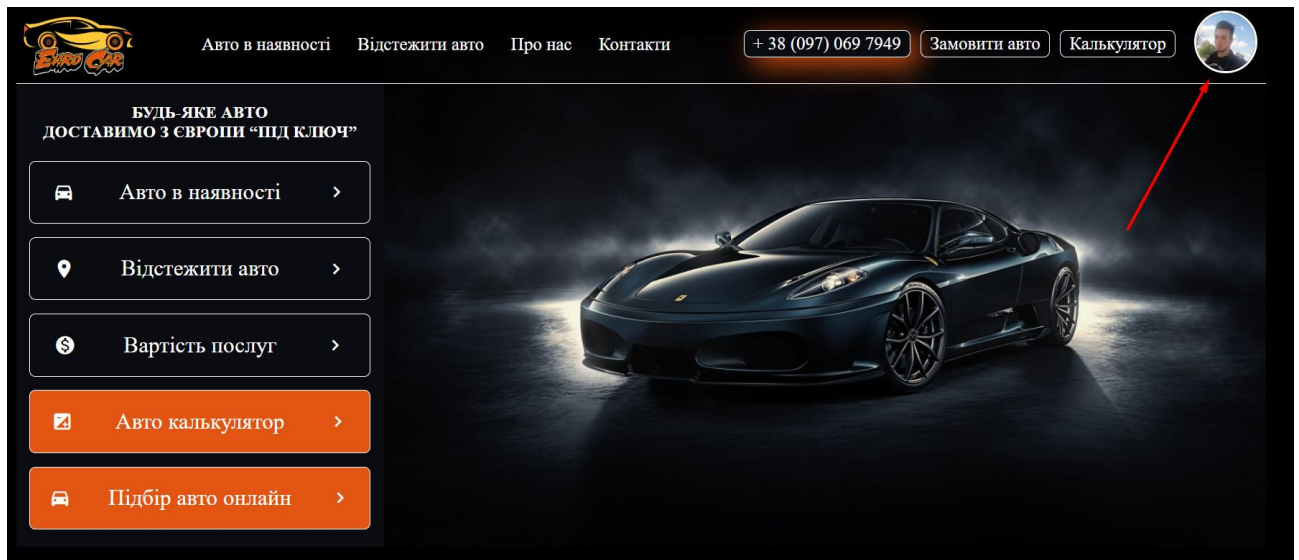


Рисунок 3.20 – Відображення аватара після авторизації

Авторизований користувач отримує доступ до особистого кабінету (рис. 3.21), де можна ознайомитися з інформацією про купівлю або продаж автомобілів (рис. 3.22).

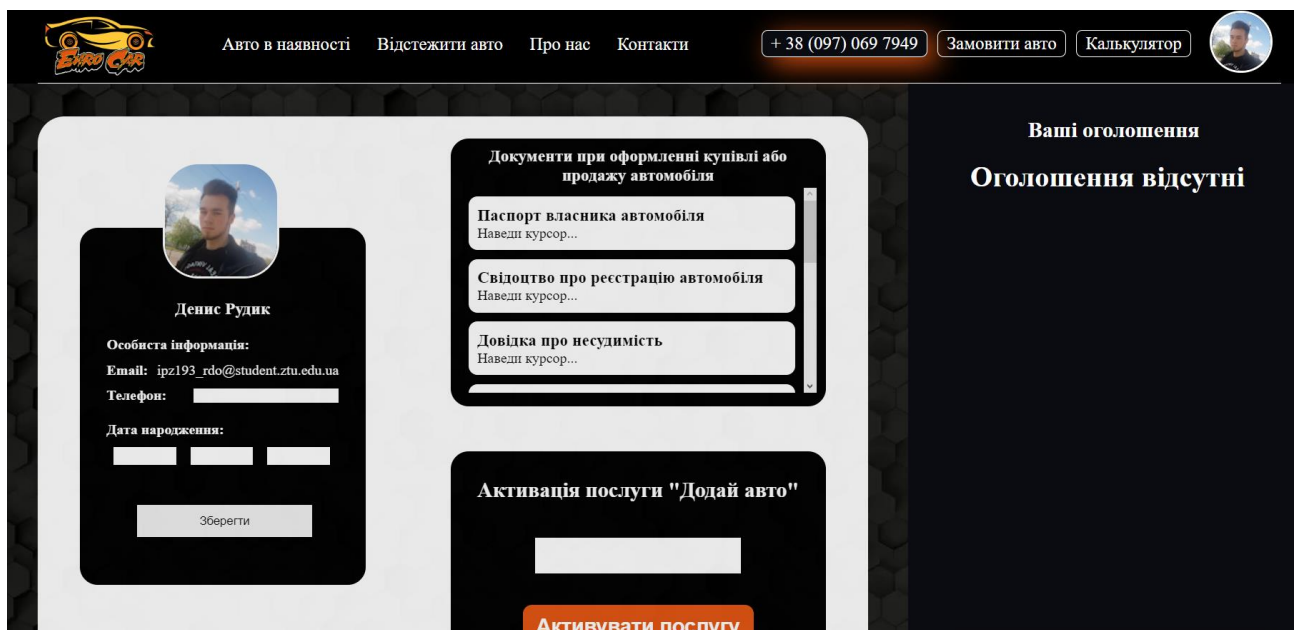


Рисунок 3.21 – Особистий кабінет

В особистому акаунті можна ознайомитись з інформацією щодо купівлі або продажу авто (рис. 3.22).

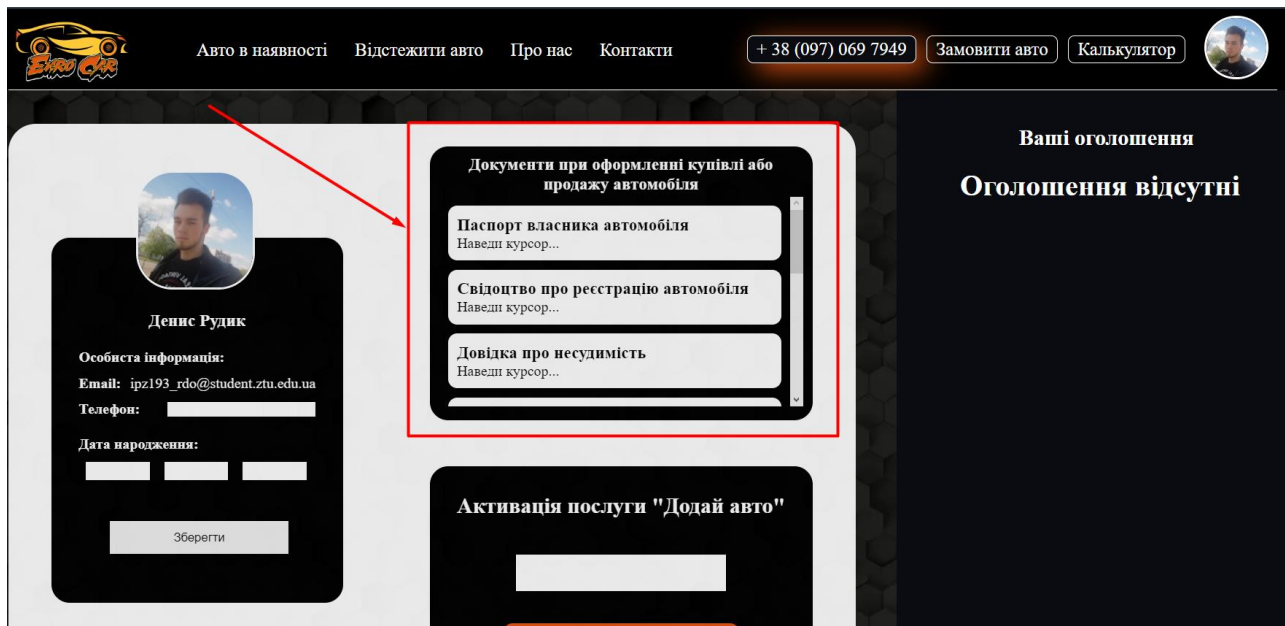


Рисунок 3.22 – Інформація щодо купівлі або продажу авто

Процес створення оголошення розпочинається з активації послуги “Додай авто”. Для її використання користувач повинен мати код доступу (рис. 3.23). Після введення коду активується кнопка “Додати авто” (рис. 3.23 – 3.25), що дозволяє перейти до створення оголошення.

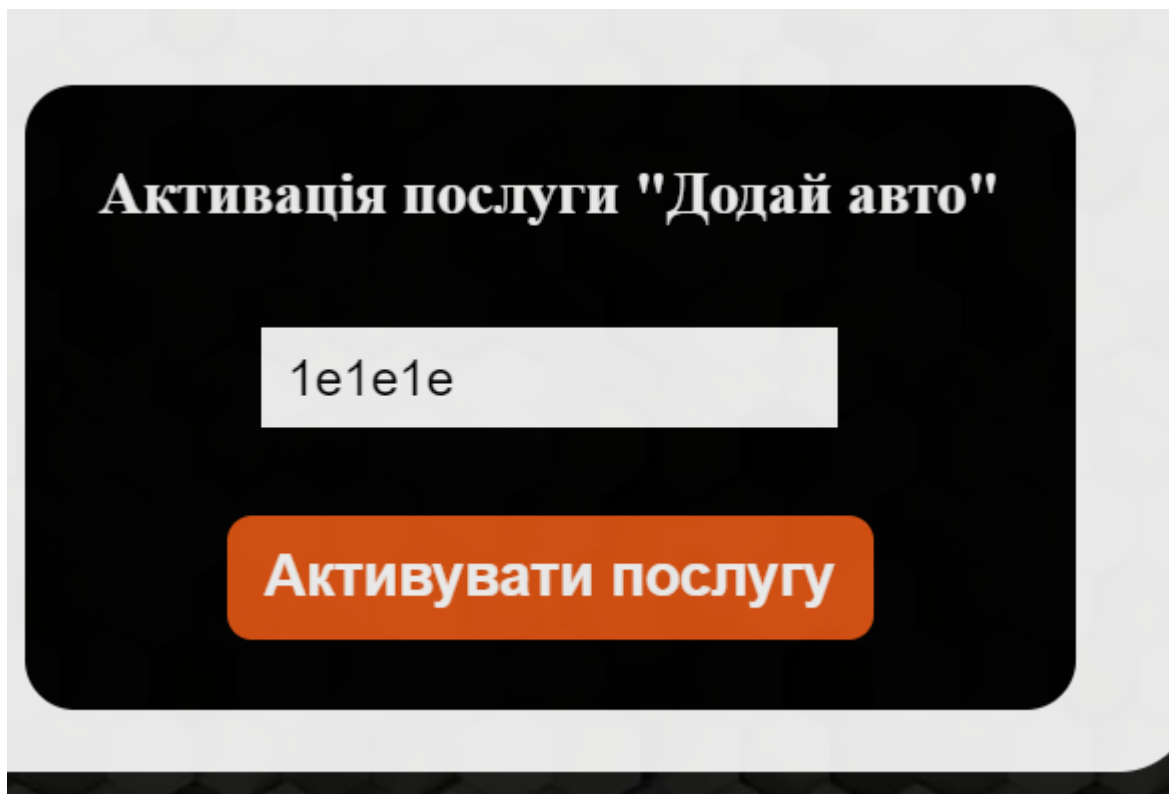


Рисунок 3.23 – Доступ до послуги

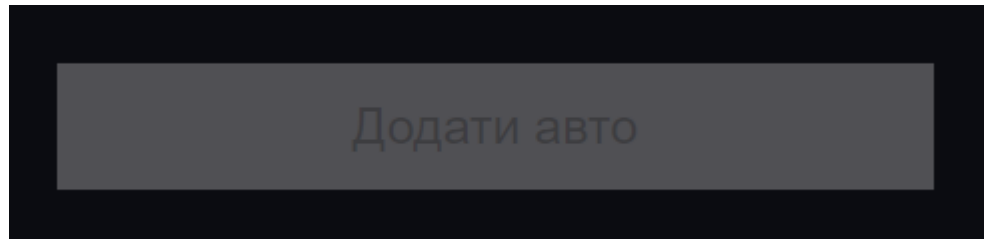


Рисунок 3.24 – Кнопка до активації

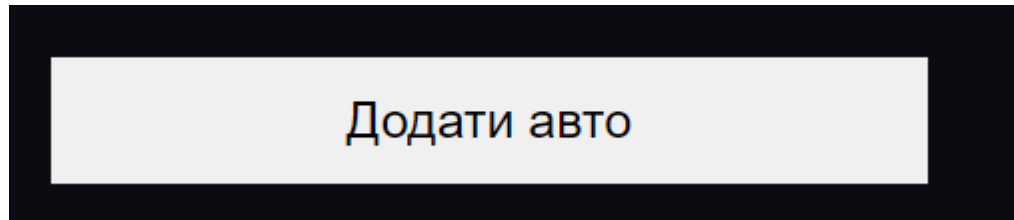


Рисунок 3.25 – Кнопка після активації

Користувач отримує права адміністратора, що дає змогу додавати власні оголошення. Для цього потрібно натиснути кнопку “Додати авто” і перейти на відповідну сторінку (рис. 3.26 – 3.27), де здійснюється створення та публікація оголошення.

A screenshot of a web application interface. At the top is a navigation bar with a logo on the left, menu items "Авто в наявності", "Відстежити авто", "Про нас", and "Контакти" in the center, and contact information "+ 38 (097) 069 7949", "Замовити авто", and "Калькулятор" on the right, along with a user profile picture. The main content area is titled "Створити оголошення" (Create advertisement) with a subtitle "Опишіть у подробицях" (Describe in detail). Below the title is a green button labeled "Виберіть файл" (Select file). The form contains three main sections: "Ціна" (Price) with a text input field containing "20000"; "Опис" (Description) with a text area containing the placeholder "Опишіть своє авто" (Describe your car); and "Категорія" (Category) with a dropdown menu showing "--- Оберіть категорію ---". At the bottom, the text "Походження авто" (Origin of the car) is visible.

Рисунок 3.26 – Форма для заповнення

Походження авто
--- Оберіть походження ---

Місто перебування
--- Оберіть місто ---

Марка
--- Оберіть марку ---

Модель

Рік виготовлення
--- Оберіть рік ---

Колір
--- Оберіть колір ---

Пошкодження
--- Авто перебувало в ДТП ---

Розмитнено
--- Розмитнене авто ---

Привід

Back to Top ↑

Рисунок 3.27 – Форма для заповнення

Після створення оголошення користувача автоматично перенаправляють до його профілю. У профілі можна побачити, що створене оголошення успішно з'явилося у списку (рис. 3.28).

Документи при оформленні купівлі або продажу автомобіля

Паспорт власника автомобіля
Наведи курсор...

Свідоцтво про реєстрацію автомобіля
Наведи курсор...

Довідка про несудимість
Наведи курсор...

Активація послуги "Додай авто"

Активувати послугу

Ваші оголошення

Редагувати Видалити

Рисунок 3.28 – Відображення створеного оголошення

Продавець має можливість редагувати або видалити оголошення, якщо воно стало неактуальним. Для редагування необхідно натиснути на кнопку

“Редагувати”, після чого здійснюється перехід на сторінку редагування (рис. 3.29). На цій сторінці можна змінити потрібні поля, а після внесення змін зберегти оновлену інформацію.

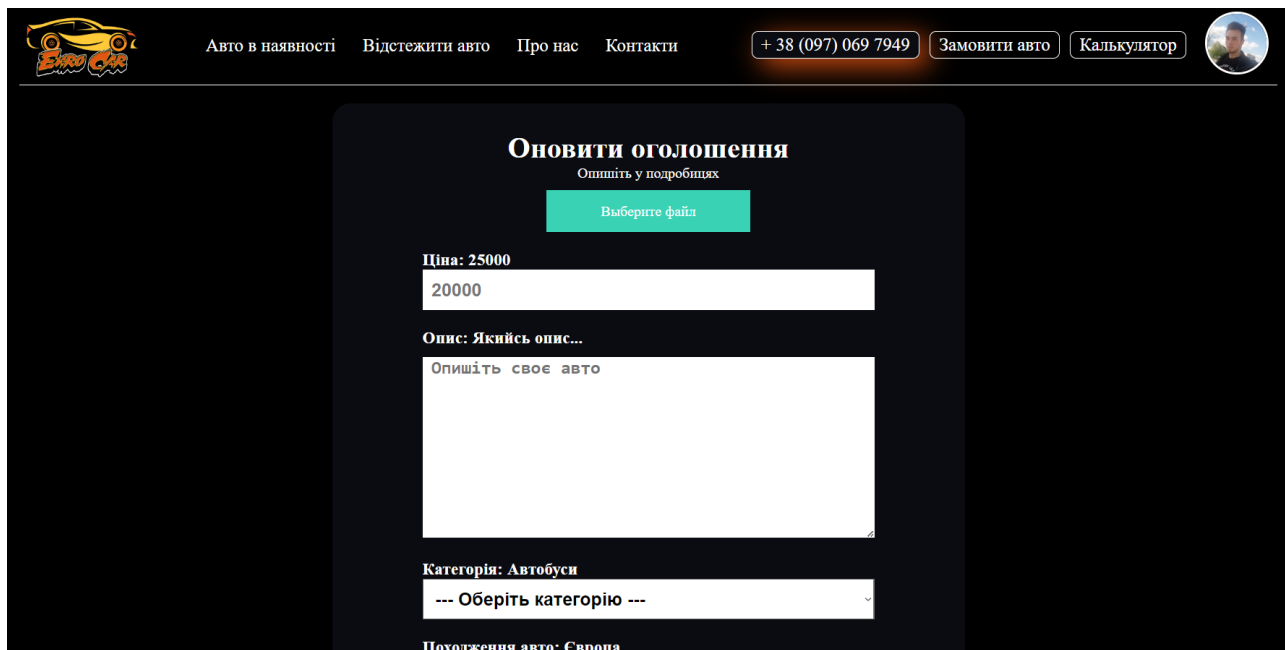


Рисунок 3. 29– Редагування оголошення

3.2 Тестування роботи програмного комплексу

Тестування програмного забезпечення є ключовим етапом для перевірки його функціональності та стабільності. У межах цієї роботи було проведено ручне тестування основних функцій додатку.

Під час авторизації користувач починає із форми входу. Після введення коректних облікових даних та натискання кнопки "Увійти" система успішно перенаправляє користувача на головну сторінку, що підтверджено результатами тесту. Після цього, для створення оголошення, користувач натискає на пункт "Додати авто", що відкриває форму створення оголошення. Тест показав, що ця функція працює правильно.

Перегляд створених оголошень здійснюється на сторінці "Авто в наявності". Натискання на обране оголошення дозволяє побачити детальну інформацію про нього, що було успішно перевірено під час тестування. Редагування оголошення відбувається через особистий кабінет. Користувач

обирає потрібне оголошення, натискає "Редагувати" та отримує доступ до форми оновлення, яка дозволяє вносити зміни.

Функція фільтрації оголошень перевіряє здатність системи звузити результати пошуку відповідно до заданих параметрів, таких як ціна, марка чи інші характеристики. Тести підтвердили правильність її виконання.

Висновки до третього розділу

У цьому розділі розглянуто структуру системи та процес розгортання кожного її компонента. Було детально описано всі структурні елементи, а також їхню взаємодію між собою.

Розглянуто функціонал редагування та видалення оголошень, що є ключовим елементом додатку для автомобілів. Ця функція дозволяє користувачам оперативно оновлювати чи змінювати інформацію про транспортні засоби, розміщену на платформі.

Функція фільтрування оголошень є важливим інструментом для покупців, адже вона дозволяє звузити пошук за заданими параметрами, такими як бренд, модель, ціна, розмір двигуна тощо. Це значно спрощує процес вибору автомобіля, який найбільше відповідає потребам користувача.

Система рекомендацій додає персоналізований підхід до використання платформи. Вона надає користувачам індивідуальні пропозиції на основі їхніх інтересів, покращуючи досвід використання додатку та допомагаючи швидше знайти оптимальний варіант автомобіля.

Функція відстеження автомобіля за VIN номером є цінним інструментом для перевірки історії транспортного засобу. Вона забезпечує доступ до важливої інформації, сприяючи прозорості та підвищенню рівня довіри під час покупки.

Довіра до компанії, яка керує цим додатком, відіграє важливу роль. Репутація компанії, якість сервісу та достовірність наданої інформації є ключовими факторами успіху.

Гаряча лінія забезпечує ефективну комунікацію між сервісом і клієнтами. Вона дозволяє покупцям швидко отримати відповіді на питання, консультації чи роз'яснення стосовно процесу покупки, умов гарантії чи інших аспектів.

У цілому, додаток демонструє свою надійність і зручність, пропонуючи можливості редагування та видалення оголошень, фільтрації пропозицій, персоналізованих рекомендацій і перевірки за VIN номером. Це робить його практичним і корисним інструментом для покупців автомобілів.

ВИСНОВКИ

У цій роботі розроблено програмний додаток, що відзначається зручністю у використанні для кінцевого користувача. Для досягнення цієї мети було виконано такі завдання:

Проведений аналіз предметної області дав змогу визначити ключові вимоги до розробки. На основі аналізу сформовано перелік функцій, що забезпечують автоматизацію процесу продажу автомобілів.

Оскільки створення конкурентного веб-додатка неможливе без вивчення аналогів, здійснено порівняння з найбільш подібними рішеннями на ринку. У результаті цього було сформовано набір важливих функцій для реалізації в нашому продукті.

Система складається з кількох компонентів, якість кожного з яких суттєво впливає на загальну функціональність. Для кожного модуля було обрано архітектуру, яка найкраще відповідає його специфіці.

Залежно від поставлених вимог і обраної архітектури підібрано відповідні засоби реалізації проекту.

Описано алгоритм взаємодії компонентів системи та логіку виконання операцій з даними.

Проаналізовано процес реалізації системи, обрано оптимальні архітектурні рішення та методи обробки даних для підвищення ефективності роботи.

Надано детальний опис структури системи, її компонентів та процесу розгортання.

Розроблено опис інтерфейсу додатка, включно з його структурними елементами, функціоналом відповідно до ролей користувачів, а також інструментами для роботи з даними.

Здійснено тестування програми для перевірки її коректності та стабільності роботи.

Розроблений програмний комплекс готовий до використання та може застосовуватися для продажу автомобілів.

У майбутньому продукт може бути вдосконалений шляхом додавання функцій, таких як збереження історії пошуку, виведення трендових автомобілів тощо.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “AJAX and APIs”, Reactjs.org, 2023 [Електроний ресурс]. Доступ за посиланням: <https://legacy.reactjs.org/docs/faq-ajax.html#gatsby-focus-wrapper> [Дата звернення: 07.04.2023]
2. “Boss Auto”, Bossautoukraine.com.ua, 2023 [Електроний ресурс]. Доступ за посиланням: <https://bossautoukraine.com.ua> [Дата звернення: 09.04.2023]
3. “Auto Terminal”, Bossautoukraine.com.ua, 2023 [Електроний ресурс]. Доступ за посиланням: <https://aterminal.com.ua> [Дата звернення: 10.04.2023]
4. “Komis Auto”, Komis.com.ua, 2023 [Електроний ресурс]. Доступ за посиланням: <https://www.komis.com.ua> [Дата звернення: 11.04.2023]
5. “Найкращі практики проєктування REST API”, Codeguida.com, 2023 [Електроний ресурс]. Доступ за посиланням: <https://codeguida.com/post/2462> [Дата звернення: 14.04.2023]
6. “AJAX та запити до API”, Reactjs.org, 2023 [Електроний ресурс]. Доступ за посиланням: <https://uk.legacy.reactjs.org/docs/faq-ajax.html> [Дата звернення: 17.04.2023]
7. “Посібник: знайомство з React”, Reactjs.org, 2023 [Електроний ресурс]. Доступ за посиланнями: <https://uk.legacy.reactjs.org/tutorial/tutorial.html> [Дата звернення: 19.04.2023]
8. “React JavaScript-бібліотека для створення користувацьких інтерфейсів”, Reactjs.org, 2023 [Електроний ресурс]. Доступ за посиланням: <https://uk.legacy.reactjs.org> [Дата звернення: 21.04.2023]
9. “MobX”, 2023 Доступ за посиланням: <https://mobx.js.org/README.html> [Дата звернення: 22.04.2023]
10. “How to Fetch Data in React: Cheat Sheet + Examples”, Reencodecamp.org, 2023 [Електроний ресурс]. Доступ за посиланнями: <https://www.freecodecamp.org/news/fetch-data-react/> [Дата звернення: 23.04.2023]

11. “Components and Props”, Reactjs.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://legacy.reactjs.org/docs/components-and-props.html> [Дата звернення: 02.04.2023]
12. “Code editing. Redefined.”, Visualstudio.com, 2023 Доступ за посиланням: <https://code.visualstudio.com> [Дата звернення: 27.04.2023]
13. “React Router DOM”, Reactrouter.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://reactrouter.com/en/main> [Дата звернення: 30.04.2023]
14. “Google-map-react”, Npmjs.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.npmjs.com/package/google-map-react> [Дата звернення: 2.05.2023]
15. “@supabase/auth-helpers-react”, Npmjs.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.npmjs.com/package/@supabase/auth-helpers-nextjs> [Дата звернення: 04.05.2023]
16. “Supabase is an open source Firebase alternative for building secure and performant Postgres backends with minimal configuration.”, Supabase.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://supabase.com> [Дата звернення: 07.05.2023]
17. “Що таке VIN номер і як він допоможе врятуватися від автошахраїв”, Youcontrol.com.ua, 2023 [Електронний ресурс]. Доступ за посиланням: <https://youcontrol.com.ua/topics/shcho-take-vin-nomer-iyak-vin-dopomozhe-vriatuvatysia-vid-avtoshakhrayiv/> [Дата звернення: 10.05.2023]
18. “React – A JavaScript library for building user interfaces”, Reactjs.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://reactjs.org> [Дата звернення: 01.05.2023]
19. “Redux – A predictable state container for JavaScript apps”, Redux.js.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://redux.js.org> [Дата звернення: 02.05.2023]
20. “React Router – Declarative routing for React.js”, Reactrouter.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://reactrouter.com> [Дата звернення: 02.05.2023]

21. “Material-UI: React components for faster and easier web development”, Mui.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://mui.com> [Дата звернення: 03.05.2023]
22. “Firebase – App success made simple”, Firebase.google.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://firebase.google.com> [Дата звернення: 03.05.2023]
23. “Next.js – The React Framework for Production”, Nextjs.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://nextjs.org> [Дата звернення: 04.05.2023]
24. “Supabase – The Open Source Firebase Alternative”, Supabase.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://supabase.com> [Дата звернення: 04.05.2023]
25. “PostgreSQL – The World’s Most Advanced Open Source Relational Database”, Postgresql.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.postgresql.org> [Дата звернення: 05.05.2023]
26. “Tailwind CSS – Rapidly build modern websites without ever leaving your HTML”, Tailwindcss.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://tailwindcss.com> [Дата звернення: 05.05.2023]
27. “Framer Motion – A production-ready motion library for React”, Framer.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.framer.com/motion> [Дата звернення: 06.05.2023]
28. “Axios – Promise based HTTP client for the browser and node.js”, Axios-http.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://axios-http.com> [Дата звернення: 06.05.2023]
29. “JSON Web Tokens – Open, industry standard RFC 7519 method for representing claims securely between two parties”, Jwt.io, 2023 [Електронний ресурс]. Доступ за посиланням: <https://jwt.io> [Дата звернення: 07.05.2023]
30. “Chart.js – Simple yet flexible JavaScript charting for designers & developers”, Chartjs.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.chartjs.org> [Дата звернення: 07.05.2023]

31. “Google-map-react”, Npmjs.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.npmjs.com/package/google-map-react> [Дата звернення: 02.05.2023]
32. “@supabase/auth-helpers-react”, Npmjs.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.npmjs.com/package/@supabase/auth-helpers-nextjs> [Дата звернення: 04.05.2023]
33. “Supabase is an open source Firebase alternative for building secure and performant Postgres backends with minimal configuration.”, Supabase.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://supabase.com> [Дата звернення: 07.05.2023]
34. “Що таке VIN номер і як він допоможе врятуватися від автошахраїв”, Youcontrol.com.ua, 2023 [Електронний ресурс]. Доступ за посиланням: <https://youcontrol.com.ua/topics/shcho-take-vin-nomer-iyak-vin-dopomozhe-vriatuvatysia-vid-avtoshakhrayiv/> [Дата звернення: 10.05.2023]
35. “REST API Tutorial”, Restapitutorial.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://restapitutorial.com> [Дата звернення: 11.05.2023]
36. “GraphQL – A query language for your API”, GraphQL.org, 2023 [Електронний ресурс]. Доступ за посиланням: <https://graphql.org> [Дата звернення: 11.05.2023]
37. “Docker – Empowering App Development for Developers”, Docker.com, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.docker.com> [Дата звернення: 12.05.2023]
38. “Kubernetes – Production-Grade Container Orchestration”, Kubernetes.io, 2023 [Електронний ресурс]. Доступ за посиланням: <https://kubernetes.io> [Дата звернення: 13.05.2023]
39. “Jest – Delightful JavaScript Testing Framework with a focus on simplicity”, Jestjs.io, 2023 [Електронний ресурс]. Доступ за посиланням: <https://jestjs.io> [Дата звернення: 14.05.2023]
40. “Cypress – Fast, easy and reliable testing for anything that runs in a browser”, Cypress.io, 2023 [Електронний ресурс]. Доступ за посиланням: <https://www.cypress.io> [Дата звернення: 15.05.2023]

41. “ESLint – Pluggable JavaScript Linter”, [Eslint.org](https://eslint.org), 2023 [Електронний ресурс].
Доступ за посиланням: <https://eslint.org> [Дата звернення: 16.05.2023]
42. “Prettier – An opinionated code formatter”, [Prettier.io](https://prettier.io), 2023 [Електронний ресурс].
Доступ за посиланням: <https://prettier.io> [Дата звернення: 17.05.2023]
43. “Git – Distributed Version Control System”, [Git-scm.com](https://git-scm.com), 2023 [Електронний ресурс].
Доступ за посиланням: <https://git-scm.com> [Дата звернення: 18.05.2023]
44. “GitHub – Where the world builds software”, [Github.com](https://github.com), 2023 [Електронний ресурс].
Доступ за посиланням: <https://github.com> [Дата звернення: 19.05.2023]
45. “Visual Studio Code – Code Editing. Redefined.”, [Code.visualstudio.com](https://code.visualstudio.com), 2023 [Електронний ресурс].
Доступ за посиланням: <https://code.visualstudio.com> [Дата звернення: 20.05.2023]
46. “OpenAI – Artificial Intelligence Research and Deployment”, [Openai.com](https://openai.com), 2023 [Електронний ресурс].
Доступ за посиланням: <https://openai.com> [Дата звернення: 21.05.2023]
47. “GPT-3: Language Models are Few-Shot Learners”, [Arxiv.org](https://arxiv.org/abs/2005.14165), 2023 [Електронний ресурс].
Доступ за посиланням: <https://arxiv.org/abs/2005.14165> [Дата звернення: 22.05.2023]
48. “Node.js – JavaScript runtime built on Chrome's V8 JavaScript engine”, [Nodejs.org](https://nodejs.org), 2023 [Електронний ресурс].
Доступ за посиланням: <https://nodejs.org> [Дата звернення: 23.05.2023]
49. “Express – Fast, unopinionated, minimalist web framework for Node.js”, [Expressjs.com](https://expressjs.com), 2023 [Електронний ресурс].
Доступ за посиланням: <https://expressjs.com> [Дата звернення: 24.05.2023]

ДОДАТКИ

1. Index.jsx:

```

2. import React from "react";
3. import "./index.css";
4. import MainCar from "../../assets/1579155590_31-39.jpg";
5. import ShortInfo from "../../assets/short_info_img.jpg";
6. import BanerCars from "../../assets/1579155590_31-39.jpg";
7. import { NavLink, useNavigate } from "react-router-dom";
8.
9. import MultiCarousel from "../../components/MultiCarousel/MultiCarousel";
10.
11. const Index = () => {
12.   const navigate = useNavigate();
13.   return (
14.     <>
15.       <section>
16.         <div className="container">
17.           <header>
18.             <div className="left-main_menu">
19.               <h3>
20.                 БУДЬ-ЯКЕ АВТО
21.               <br />
22.                 ДОСТАВИМО З ЄВРОПИ “ПІД КЛЮЧ”
23.             </h3>
24.             <ul>
25.               <a href="#">
26.                 <NavLink to="/stock">
27.                   <li>
28.                     <i className="large material-
29. icons">directions_car</i>Авто
30.                     в наявності
31.                     <i className="large material-icons">chevron_right</i>
32.                   </li>
33.                 </NavLink>
34.               </a>
35.               <a href="#">
36.                 <NavLink to="/tracking">
37.                   <li>
38.                     <i className="large material-icons">place</i>Відстежити
39.                     авто
40.                     <i className="large material-icons">chevron_right</i>
41.                   </li>
42.                 </NavLink>
43.               </a>
44.               <a href="#">
45.                 <NavLink to="/contacts">
46.                   <li>
47.                     <i className="large material-icons">monetization_on</i>
48.                     Вартість послуг

```

```

48.         <i className="large material-icons">chevron_right</i>
49.     </li>
50. </NavLink>
51. </a>
52. <a href="#">
53.     <NavLink to="/calculator">
54.         <li className="last_btn">
55.             <i className="large material-icons">exposure</i>Авто
56.             калькулятор
57.             <i className="large material-icons">chevron_right</i>
58.         </li>
59.     </NavLink>
60. </a>
61. <a href="#">
62.     <NavLink to="/contacts">
63.         <li className="last_btn">
64.             <i className="large material-icons">directions_car</i>
65.             Підбір авто онлайн
66.             <i className="large material-icons">chevron_right</i>
67.         </li>
68.     </NavLink>
69. </a>
70. </ul>
71. </div>
72. <img className="img_main-car" src={MainCar} />
73. </header>
74. <div className="contacts_header">
75.     <ul>
76.         <li>
77.             Відділ продажу
78.             <br />
79.             <span>+38 (093) 344 1747</span>
80.         </li>
81.         <li>
82.             СТО
83.             <br />
84.             <span>+38 (096) 234 3149</span>
85.         </li>
86.         <li>
87.             Запчастини
88.             <br />
89.             <span>+38 (097) 452 9224</span>
90.         </li>
91.         <li>
92.             Сертифікація
93.             <br />
94.             <span>+38 (096) 494 7237</span>
95.         </li>
96.     </ul>
97. </div>
98. </div>

```

```

99.     </section>
100.     <section>
101.         <div className="container">
102.             <div className="short_info">
103.                 <div className="choose_btn-div">
104.                     <h2>ТВОЯ МРІЯ СТАЛА ЩЕ ДОСТУПНІШОЮ!</h2>
105.                     <p>
106.                         АДЖЕ EURO CAR
107.                         <br />
108.                         ПОКРИВАЄ УВЕСЬ ДІАПАЗОН КУПІВЛІ,
109.                         <br />
110.                         ДОСТАВКИ ТА ОФОРМЛЕННЯ АВТО
111.                     </p>
112.                     <NavLink to="/contacts">
113.                         <button>Підібрати авто в один клік</button>
114.                     </NavLink>
115.                 </div>
116.                 <div className="short_info-li">
117.                     <ul>
118.                         <li>
119.                             АВТОМОБІЛІ З ЄВРОПИ
120.                             <span className="large material-icons">check</span>
121.                         </li>
122.                         <li>
123.                             ДОСТАВКА<span className="large material-
124.                                 icons">check</span>
125.                         </li>
126.                         <li>
127.                             РОЗМИТНЕННЯ<span className="large material-
128.                                 icons">check</span>
129.                         </li>
130.                         <li>
131.                             РЕМОНТ<span className="large material-
132.                                 icons">check</span>
133.                         </li>
134.                         <li>
135.                             СЕРТИФІКАЦІЯ
136.                             <span className="large material-icons">check</span>
137.                         </li>
138.                     </ul>
139.                 </div>
140.             </div>
141.             <div>
142.                 <img src={ShortInfo} alt="short_info_img" />
143.             </div>
144.         </div>
145.     </section>
146.     <section>
147.         <div className="container">
148.             <div className="div_offers">
149.                 <div className="info_offers">
150.                     <h2 className="main_h2">
151.                         <span>EURO CAR</span>

```

```

147.         <br /> НАДІЙНА ДОСТАВКА АВТО З ЄВРОПИ ТА ДОСТАВКА У
        ВАШЕ МІСТО
148.     </h2>
149.     <div className="blocks_offers">
150.         <div className="offer">
151.             <h2 className="offer_h2_main">
152.                 КУПИТИ АВТО
153.                 <br />З ЄС
154.             </h2>
155.             <p>ЗАМОВЛЯЮЧИ АВТО ЧЕРЕЗ EURO CAR, ВИ ЕКОНОМИТЕ
                25%.</p>
156.             <h2>У ВАРТІСТЬ ВКЛЮЧЕНО:</h2>
157.             <ul>
158.                 <li>- АУКЦІОННА ЦІНА АВТО</li>
159.                 <li>- ДОСТАВКА АВТО</li>
160.                 <li>- РОЗМИТНЕННЯ ТА СЕРТИФІКАЦІЯ АВТО</li>
161.                 <li>- РЕМОНТ АВТО ТА ХІМЧИСТКА</li>
162.             </ul>
163.             <button>Отримати консультацію</button>
164.         </div>
165.         <div className="offer">
166.             <h2 className="offer_h2_main">
167.                 КУПИТИ РОЗМИТНЕНЕ
168.                 <br />
169.                 АВТО
170.             </h2>
171.             <p>
172.                 ЗПРОДАЄМО РОЗМИТНЕНІ АВТО ІЗ ЄВРОПИ ЗА ДОСТУПНИМИ
                ЦІНАМИ.
173.             </p>
174.             <h2>У ВАРТІСТЬ ВКЛЮЧЕНО:</h2>
175.             <ul>
176.                 <li>- АУКЦІОННА ЦІНА АВТО</li>
177.                 <li>- ДОСТАВКА АВТО</li>
178.                 <li>- РОЗМИТНЕННЯ ТА СЕРТИФІКАЦІЯ АВТО</li>
179.                 <li>- РЕМОНТ АВТО ТА ХІМЧИСТКА</li>
180.             </ul>
181.             <button>Отримати консультацію</button>
182.         </div>
183.         <div className="offer">
184.             <h2 className="offer_h2_main">
185.                 ПРОДАЙ СВОЄ АВТО З
186.                 <br />
187.                 ЄВРОПИ
188.             </h2>
189.             <p>У ВАС Є МОЖЛИВІСТЬ ПРОДАТИ СВОЄ ПРИВЕЗЕНЕ АВТО З
                ЄС</p>
190.             <h2>У ВАРТІСТЬ ВКЛЮЧЕНО:</h2>
191.             <ul>
192.                 <li>- ВАРТІСТЬ ПОЛУГ EURO CAR</li>
193.                 <li>- ДОСТАВКА АВТО</li>

```

```

194.             <li>- РОЗМИТНЕННЯ ТА СЕРТИФІКАЦІЯ АВТО</li>
195.             <li>- РЕМОНТ АВТО ТА ХІМЧИСТКА</li>
196.         </ul>
197.         <button>Отримати консультацію</button>
198.     </div>
199. </div>
200. </div>
201. </div>
202. </div>
203. </section>
204. <section>
205.     <div className="container">
206.         <div className="car_slider_box">
207.             <h2 className="main_h2">ОТРИМАЙ БАЖАНИЙ АВТОМОБІЛЬ В ОДИН
                КЛИК</h2>
208.             <div className="slider">
209.
210.                 <MultiCarousel />
211.
212.                 <button className="more_cars" onClick={() =>
                    navigate("/stock")}>
213.                     Більше авто{" "}
214.                 </button>
215.             </div>
216.         </div>
217.     </div>
218. </section>
219. </>
220. );
221. };
222.
223. export default Index;
224.

```

ДОДАТОК Б

SmartBlock:

```

4. import { supabase } from "../../components/SupabaseConf/Supabase";
5.
6. async function fetchRecommendationsAPI(userId) {
7.     try {
8.         console.log("Fetching recommendations for user:", userId);
9.
10.        const { data: favoriteCars, error: favError } = await supabase
11.            .from("user_interactions")
12.            .select("car_id")
13.            .eq("user_id", userId)
14.            .eq("interaction_type", "favorite");
15.

```

```

16.   const { data: dislikedCars, error: disError } = await supabase
17.     .from("user_interactions")
18.     .select("car_id")
19.     .eq("user_id", userId)
20.     .eq("interaction_type", "dislike");
21.
22.   if (favError || disError) {
23.     console.error("Error fetching user interactions:", favError ||
disError);
24.     throw new Error("Failed to fetch user interactions.");
25.   }
26.
27.   const favoriteCarIds = favoriteCars.map((car) => car.car_id);
28.   const dislikedCarIds = dislikedCars.map((car) => car.car_id);
29.   console.log("Favorite cars:", favoriteCarIds);
30.   console.log("Disliked cars:", dislikedCarIds);
31.
32.   const { data: allCars, error: carsError } = await supabase
33.     .from("cars")
34.     .select("*");
35.
36.   if (carsError) {
37.     console.error("Error fetching all cars:", carsError);
38.     throw new Error("Failed to fetch all cars.");
39.   }
40.
41.   const filteredCars = allCars.filter(
42.     (car) => !dislikedCarIds.includes(car.id)
43.   );
44.   console.log("Filtered cars (excluding dislikes):", filteredCars);
45.
46.   const { data: favoriteAttributes, error: favAttrError } = await supabase
47.     .from("cars")
48.     .select("*")
49.     .in("id", favoriteCarIds);
50.
51.   if (favAttrError) {
52.     console.error("Error fetching favorite attributes:", favAttrError);
53.     throw new Error("Failed to fetch favorite car attributes.");
54.   }
55.
56.   console.log("Favorite car attributes:", favoriteAttributes);
57.
58.   const carRatings = {};
59.
60.   const weightConfig = {
61.     price: 0.3,
62.     year: 0.3,
63.     fuel_type: 0.4,
64.   };
65.

```

```

66. function calculateSimilarity(car, favoriteAttributes) {
67.   let rating = 0;
68.   favoriteAttributes.forEach((favCar) => {
69.     const priceSimilarity =
70.       1 - Math.abs(car.price - favCar.price) / favCar.price;
71.     rating += priceSimilarity * weightConfig.price;
72.
73.     const yearSimilarity = 1 - Math.abs(car.year - favCar.year) / 10;
74.     rating += yearSimilarity * weightConfig.year;
75.
76.     const fuelSimilarity = car.fuel_type === favCar.fuel_type ? 1 : 0;
77.     rating += fuelSimilarity * weightConfig.fuel_type;
78.   });
79.
80.   return rating;
81. }
82.
83. filteredCars.forEach((car) => {
84.   carRatings[car.id] = calculateSimilarity(car, favoriteAttributes);
85. });
86.
87. console.log("Car ratings:", carRatings);
88.
89. const recommendedCars = Object.keys(carRatings)
90.   .sort((a, b) => carRatings[b] - carRatings[a])
91.   .slice(0, 10)
92.   .map((carId) => filteredCars.find((car) => car.id ===
    parseInt(carId)));
93.
94. console.log("Recommended cars:", recommendedCars);
95.
96. return recommendedCars;
97. } catch (error) {
98.   console.error("Error in fetchRecommendationsAPI:", error);
99.   return [];
100. }
101. }
102. export default fetchRecommendationsAPI;
103.

```