

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Проектування та розробка клієнт-серверного застосунку для
оренди авто з використанням фреймворку React js

Виконав(ла): студент(ка) 4 курсу, групи СП-42
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис)

Присяжнюк Н. В.

(прізвище та ініціали)

Керівник

(підпис)

Стефанишин В. М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Стоянов Ю. М.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М. Р.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет _____
(повна назва факультету)
Кафедра _____
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)
« »

(прізвище та ініціали)
20__р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня _____ бакалавр
(назва освітнього ступеня)
за спеціальністю _____ 121 – Інженерія програмного забезпечення
(шифр і назва спеціальності)
студенту _____ Присяжнюку Назару Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи “Проектування та розробка клієнт-серверного застосунку для оренди авто з використанням фреймворку React js”

Керівник роботи _____ Стефанишин Володимир Миколайович, ас. каф. ПІ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

[illegible]

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

(прізвище та ініціали)

Керівник роботи

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Проектування та розробка клієнт-серверного застосунка для оренди авто з використанням фреймворку ReactJS// Кваліфікаційна робота освітнього рівня «Бакалавр» // Присяжнюк Назар Вікторович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмої інженерії, група СП-42 // Тернопіль, 2024 // С. 71, рис. – 25, табл. – 4, додат. – 2, бібліогр. – 14.

Ключові слова: Оренда, авто, застосунок, користувач, адміністратор, React, NodeJS, Visual Studio Code.

У першому розділі кваліфікаційної роботи проведено детальний аналіз предметної області оренди авто. Включено загальну інформацію про ринок та аналіз переваг і недоліків основних конкурентів. Також обґрунтовано постановку задачі для розробки застосунку.

У другому розділі роботи здійснено проектування застосунку. Визначено вимоги до системи та описано її функціональні можливості. Створено діаграми прецедентів, сценарії використання та діаграми послідовності для ключових операцій.

У третьому розділі було проведено розробку застосунку та його компонентів. Вибрано відповідні технології та середовище розробки. Реалізовано основні елементи інтерфейсу, включаючи створення логотипу, головної сторінки, інтерфейсу бронювання та оренди авто. Крім того, проведено функціональне тестування застосунку для забезпечення його стабільної та надійної роботи.

ANNOTATION

Design and development of a client-server application for car rental using ReactJS framework // Qualification work of the Bachelor's degree // Prysiazhniuk Nazar Viktorovych // Ivan Puluj National Technical University of Ternopil, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, SP-42 group // Ternopil, 2024. // Pages - 71, figures - 25, tables - 4, addition - 2, literature - 14.

Keywords: Rental, car, application, user, administrator, React, NodeJS, Visual Studio Code.

The first chapter of the qualification work provides a detailed analysis of the car rental subject area. It includes general information about the market and an analysis of the advantages and disadvantages of the main competitors. The task statement for the application development is also substantiated.

The second section of the paper deals with the design of the application. The requirements for the system are defined and its functionality is described. Precedent diagrams, use cases, and sequence diagrams for key operations are created.

In the third section, we developed the application and its components. The appropriate technologies and development environment were selected. The main elements of the interface were implemented, including the creation of a logo, home page, booking and car rental interface. In addition, we conducted functional testing of the application to ensure its stable and reliable operation.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface (інтерфейс прикладного програмування, набір підпрограм, який використовується для взаємодії програмних компонентів.)

HTML – HyperText Markup Language (мова, що використовується для створення веб-застосунків).

CSS – Cascading Style Sheets (таблиці каскадних стилів, мова стилів, що використовується для опису зовнішнього вигляду документа, написаного мовою розмітки).

JS – JavaScript (мова програмування, яка використовується для створення динамічного та інтерактивного контенту на веб-сторінках).

React – JavaScript бібліотека для побудови користувацьких інтерфейсів, розроблена компанією Facebook.

VSCode – Visual Studio Code (редактор вихідного коду, розроблений компанією Microsoft, який використовується для розробки програмного забезпечення).

IDE – Integrated Development Environment (інтегроване середовище розробки, що забезпечує комплекс інструментів для написання коду в одному програмному забезпеченні).

UML – Unified Modeling Language (уніфікована мова моделювання, стандартна мова для створення схем і діаграм об'єктно-орієнтованого програмного забезпечення).

DOM – Document Object Model (об'єктна модель документа, стандарт програмного інтерфейсу для роботи з HTML та XML-документами, що представляє структуру документу у вигляді дерева).

FAQ – Frequently Asked Questions (часті запитання, розділ на веб-сайті, що містить відповіді на найбільш поширені запитання користувачів).

ЗМІСТ

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Загальна інформація про оренду авто	11
1.2 Аналіз переваг та недоліків основних конкурентів.....	13
1.2.1 Zipcar	13
1.2.2 ShareNow	14
1.2.3 Turo	15
1.3 Постановка задачі.....	17
РОЗДІЛ 2. ПРОЄКТУВАННЯ ЗАСТОСУНКУ	19
2.1 Визначення вимог	19
2.2 Опис функціоналу системи	21
2.3 Діаграми прецедентів та сценарії	23
2.4 Діаграми послідовності	30
РОЗДІЛ 3. РОЗРОБКА ЗАСТОСУНКУ	34
3.1 Вибір технологій	34
3.2 Вибір середовища розробки.....	36
3.3 Реалізація застосунка та його компонентів	38
3.3.1 Створення логотипу	38
3.3.2 Створення інтерфейсу головної сторінки застосунка.....	39
3.3.3 Створення інтерфейсу бронювання авто	44
3.3.4 Створення інтерфейсу оренди авто	45
3.3.5 Створення сторінки “Про нас”	46
3.3.6 Створення сторінки “Моделі авто”	46

3.3.7 Створення сторінки “Контакти”	47
3.4 Тестування застосунка.....	49
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	54
4.1 Принципи, способи та засоби захисту населення.....	54
4.2 Можливість виникнення статичної електрики та заходи боротьби з нею.....	55
ВИСНОВОК.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТКИ.....	61

ВСТУП

На сьогоднішній день автомобільна індустрія переживає значні зміни під впливом технологічних інновацій та зростаючих потреб споживачів. Однією з ключових тенденцій у цій сфері є перехід від власності на транспортні засоби до моделі їх спільного використання. Зростаюча популярність сервісів оренди автомобілів відображає глобальний тренд до більш ефективного використання ресурсів і зменшення негативного впливу на довкілля. У цьому контексті створення зручних та інтуїтивно зрозумілих платформ для оренди авто стає все більш актуальним.

Основною метою розробки даного застосунку є створення інноваційного рішення для оренди автомобілів, яке задовольняє потреби сучасних користувачів у швидкому, зручному та надійному доступі до транспортних засобів. Застосунок орієнтований на широке коло користувачів, які цінують свій час і бажають мати можливість орендувати автомобіль у будь-який час і в будь-якому місці. Така платформа не лише спрощує процес оренди, але й сприяє популяризації екологічно відповідального підходу до використання автомобілів.

Розробка застосунку включає в себе декілька ключових етапів, починаючи від аналізу ринку та потреб споживачів, закінчуючи технічним виконанням і тестуванням готового продукту. У процесі розробки були використані сучасні технології та підходи, що забезпечують високу продуктивність, безпеку та зручність використання. Крім того, особлива увага приділялася інтерфейсу користувача, адже саме він є важливим фактором у створенні позитивного враження про сервіс та його подальшому успіху на ринку.

Важливим аспектом реалізації проєкту було врахування різноманітних вимог та очікувань користувачів. Проведений аналіз показав, що споживачі очікують від подібних сервісів наявності широкого вибору автомобілів, прозорості умов оренди, доступності цін та можливості оперативного вирішення будь-яких проблем, що можуть виникнути під час користування послугою. Тому розробка включала

створення системи зворотного зв'язку та підтримки користувачів, що забезпечує високий рівень сервісу та задоволення потреб клієнтів.

Також слід зазначити, використання оренди автомобілів надає зручність і гнучкість в плануванні подорожей та виконанні щоденних завдань. Користувачі можуть вибрати автомобіль, що найкраще відповідає їхнім потребам у конкретний момент часу. Це особливо корисно для подорожуючих, бізнесменів, студентів та тимчасових мешканців міст.

Оренда автомобілів сприяє покращенню економіки через кілька ключових механізмів. По-перше, вона збільшує доступність транспорту для широкого кола людей, надаючи можливість переміщення навіть тим, хто не має власного автомобіля. Це стимулює економічну активність, збільшуючи обсяги туризму, комерційних операцій та інших видів діяльності, які вимагають наявності транспорту. По-друге, оренда автомобілів зменшує необхідність власності на автопарк, особливо у великих містах. Це зменшує кількість автомобілів на дорогах, сприяє зменшенню заторів та забрудненню повітря, що позитивно впливає на якість життя мешканців і знижує витрати на інфраструктуру.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна інформація про оренду авто

Каршерінг (англ. carsharing) – це сучасна форма оренди автомобілів, яка дозволяє користувачам орендувати транспортні засоби на короткий термін, зазвичай погодинно або подовово. Цей підхід до використання автомобілів отримує все більше визнання у великих містах по всьому світу завдяки своїй зручності, економічній вигоді та екологічним перевагам [1].

Каршерінг базується на кількох ключових принципах:

- **Спільне використання ресурсів:** Автомобілі не належать конкретним користувачам, а надаються у тимчасове користування багатьом людям. Це дозволяє більш ефективно використовувати транспортні засоби та зменшує кількість автомобілів на дорогах.
- **Короткострокова оренда:** На відміну від традиційної оренди автомобілів, каршерінг передбачає оренду на короткі періоди часу – від декількох хвилин до декількох годин. Це зручно для короткочасних поїздок по місту або для виконання конкретних завдань.
- **Самообслуговування:** Каршерінг-сервіси часто використовують автоматизовані системи бронювання та доступу до автомобілів. Користувачі можуть забронювати авто через мобільний додаток або веб-сайт, а доступ до автомобіля здійснюється за допомогою електронних ключів або спеціальних карток.
- **Розподіл витрат:** Вартість використання автомобіля розподіляється між всіма користувачами, що робить каршерінг економічно вигідним для індивідуальних користувачів, особливо у порівнянні з витратами на утримання власного автомобіля.

Історія та розвиток каршерінгу: концепція каршерінгу виникла у 1948 році у Швейцарії, але справжнього поширення цей сервіс набув у 1990-х роках завдяки розвитку інформаційних технологій. Перші успішні комерційні проєкти з'явилися

у Європі та Північній Америці. Згодом каршерінг став популярним у багатьох країнах світу, включаючи Азію та Австралію.

Існує кілька моделей каршерінгу, які можуть відрізнятися за умовами користування та організаційними підходами:

- Класичний каршерінг: Передбачає, що автомобіль повинен бути повернутий у певну станцію або паркувальне місце після завершення оренди.
- Free-floating каршерінг: Дозволяє користувачам залишати автомобіль у будь-якому дозволеному місці в межах визначеної зони обслуговування. Це забезпечує більшу гнучкість для користувачів.
- Peer-to-peer каршерінг: Включає у себе оренду автомобілів, що належать іншим приватним особам. Це дозволяє власникам автомобілів здавати їх в оренду під час невикористання.

Переваги каршерінгу:

Економічна вигода - зменшення витрат на володіння автомобілем, включаючи вартість покупки, страхування, обслуговування та зберігання.

Зменшення заторів - скорочення кількості автомобілів на дорогах сприяє зниженню транспортних заторів у містах.

Екологічний вплив - зниження викидів парникових газів та інших забруднювачів завдяки оптимальному використанню автомобілів та зменшенню потреби у виробництві нових транспортних засобів.

Зручність та доступність - можливість швидко та зручно орендувати автомобіль у будь-який час доби, що особливо важливо для мешканців великих міст.

Каршерінг є перспективним напрямком розвитку транспортної індустрії, який сприяє більш ефективному використанню ресурсів, зниженню витрат та позитивному впливу на довкілля. Успішна реалізація каршерінг-сервісу залежить від багатьох факторів, включаючи технологічну підтримку, нормативне середовище та потреби користувачів.

1.2 Аналіз переваг та недоліків основних конкурентів

Каршерінг-сервіси стрімко розвиваються, і на ринку існує кілька провідних компаній, які пропонують своїм користувачам зручні та доступні рішення для оренди автомобілів. У цьому розділі розглянемо основних конкурентів у сфері каршерінгу, їхні переваги та недоліки.

1.2.1 Zipcar

Zipcar – один з найвідоміших каршерінг-сервісів, який працює в багатьох країнах світу, включаючи США, Канаду та Європу. Він пропонує користувачам широкий вибір автомобілів, які можна орендувати по годинах або днях.

Переваги Zipcar:

- Широка мережа покриття: Zipcar має велику кількість автомобілів у багатьох містах по всьому світу, що робить сервіс доступним для великої кількості користувачів.
- Різноманітність автопарку: Сервіс пропонує різні типи автомобілів, включаючи компактні авто, седани, позашляховики та навіть вантажівки, що задовольняє різноманітні потреби користувачів.
- Простий процес реєстрації та бронювання: Інтуїтивно зрозумілий додаток та веб-сайт дозволяють швидко зареєструватися і забронювати автомобіль.
- Доступність: Zipcar зазвичай дешевше, ніж оренда автомобіля в традиційній компанії з прокату. Ви платите лише за час, протягом якого користуєтесь автомобілем, і немає жодних додаткових зборів за пробіг або бензин.

Недоліки Zipcar:

- Обмеження по зоні використання: Деякі міста мають обмежені зони використання, що може створювати незручності для користувачів, які хочуть подорожувати за межі цих зон.

- Додаткові збори: Існують додаткові збори за пізнє повернення автомобіля, відсутність пального та перевищення обмеження пробігу, що може збільшити загальну вартість оренди.
- Високі ціни Вартість оренди у Zipcar може бути вищою, ніж у деяких конкурентів, особливо для тривалих поїздок.

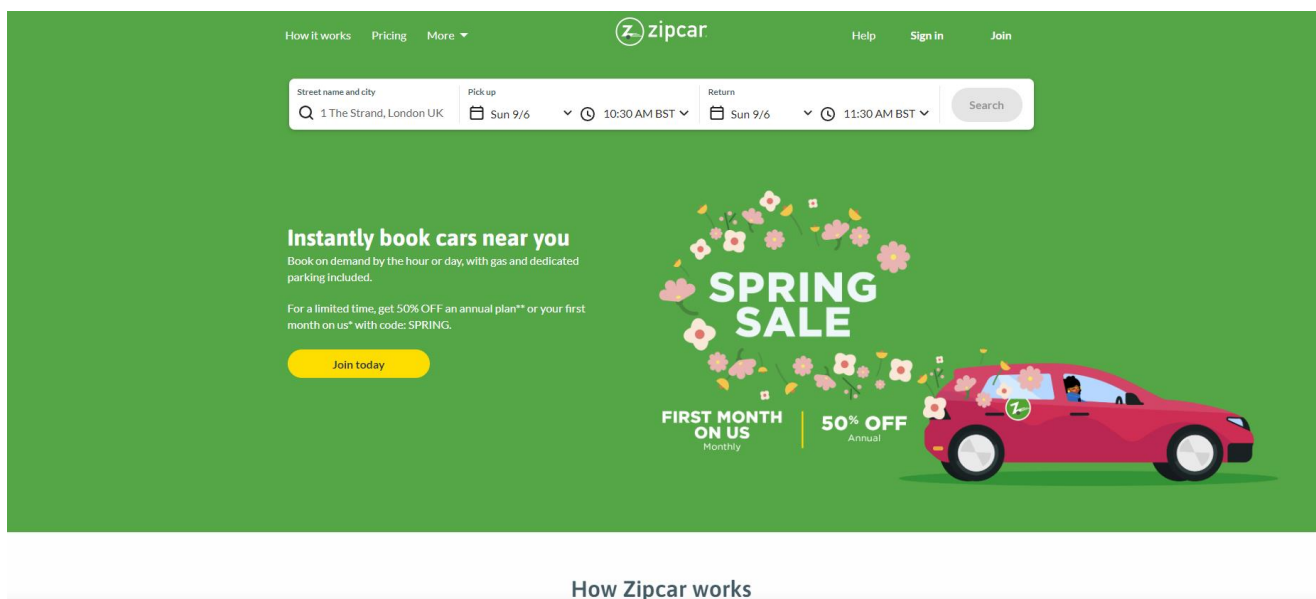


Рисунок 1.1 – Приклад роботи застосунка Zipcar

1.2.2 ShareNow

ShareNow— це один з провідних free-floating каршерінг-сервісів, що пропонує оренду автомобілів без необхідності повернення до конкретної станції.

Переваги ShareNow:

- Гнучкість у використанні: Можливість залишати автомобіль у будь-якому дозволеному місці в межах зони обслуговування, що робить поїздки більш зручними.
- Простота користування: Легкий процес реєстрації та доступ до автомобілів за допомогою мобільного додатку.
- Прозорі тарифи: Чітка структура цін, що включає погодинну та поминутну оплату, робить витрати передбачуваними для користувачів.

Недоліки ShareNow:

- Обмежена зона обслуговування: Не всі міста мають покриття ShareNow, а зона обслуговування може бути обмеженою навіть у містах, де сервіс доступний.
- Менший вибір автомобілів: У порівнянні з деякими іншими сервісами, ShareNow пропонує обмежений вибір моделей автомобілів, переважно компактних.
- Вартість парковки: У деяких містах вартість паркування може бути високою, що додає додаткові витрати для користувачів.
- Необхідність членства: Щоб користуватися Zipcar, вам потрібно буде стати членом. Членські внески варіюються залежно від вашого місця проживання.

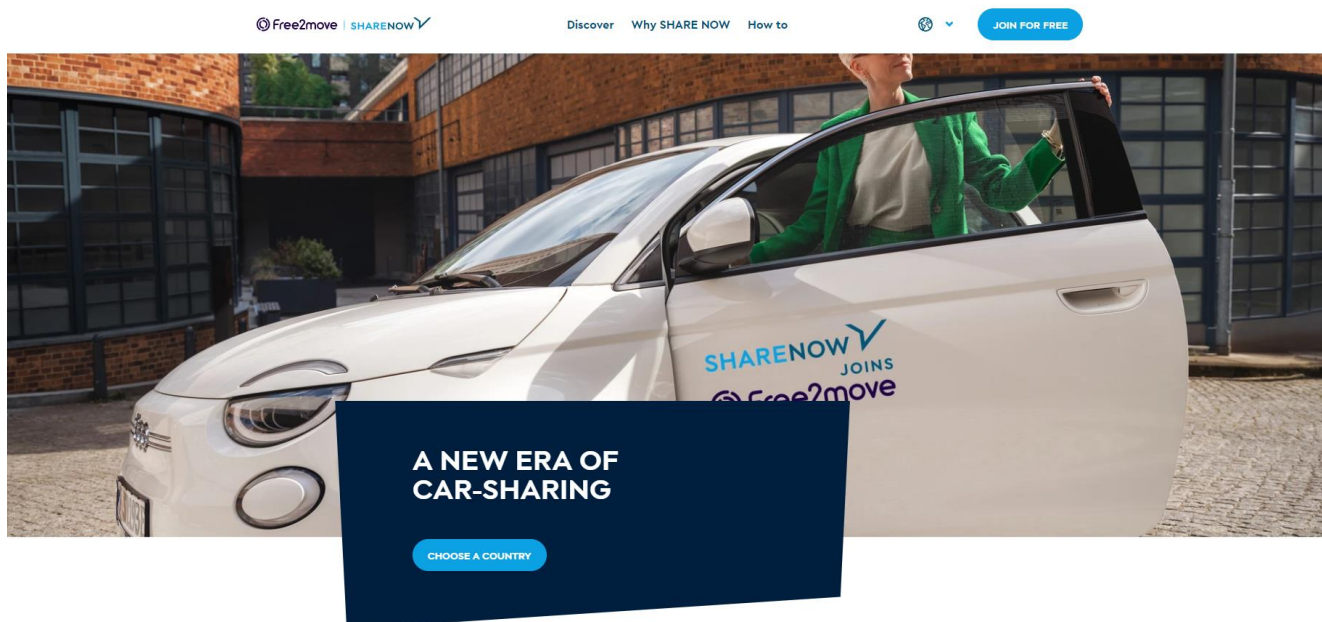


Рисунок 1.2 – Приклад роботи застосунка ShareNow

1.2.3 Turo

Turo – це peer-to-peer каршерінг-платформа, яка дозволяє приватним власникам здавати свої автомобілі в оренду іншим користувачам.

Переваги Turo:

- Широкий вибір автомобілів: Завдяки моделі peer-to-peer, Turo пропонує велику різноманітність автомобілів, включаючи розкішні моделі, спортивні авто, а також класичні та рідкісні автомобілі.
- Гнучкі умови оренди: Користувачі можуть домовлятися про умови оренди безпосередньо з власниками, що дозволяє знайти найкращі пропозиції.
- Конкурентні ціни: Можливість знайти автомобілі за більш вигідними цінами, ніж у традиційних каршерінг-сервісах.

Недоліки Turo:

- Залежність від власників: Якість та стан автомобілів можуть варіюватися залежно від власників, що може впливати на досвід користувачів.
- Обмеження по доступності: Доступність автомобілів може бути обмеженою в певних регіонах, особливо у менш населених або віддалених районах.
- Ризики та страхування: Хоча Turo пропонує страхові пакети, ризики, пов'язані з орендою автомобіля у приватних осіб, можуть бути вищими, ніж у випадку з традиційними каршерінг-компаніями.

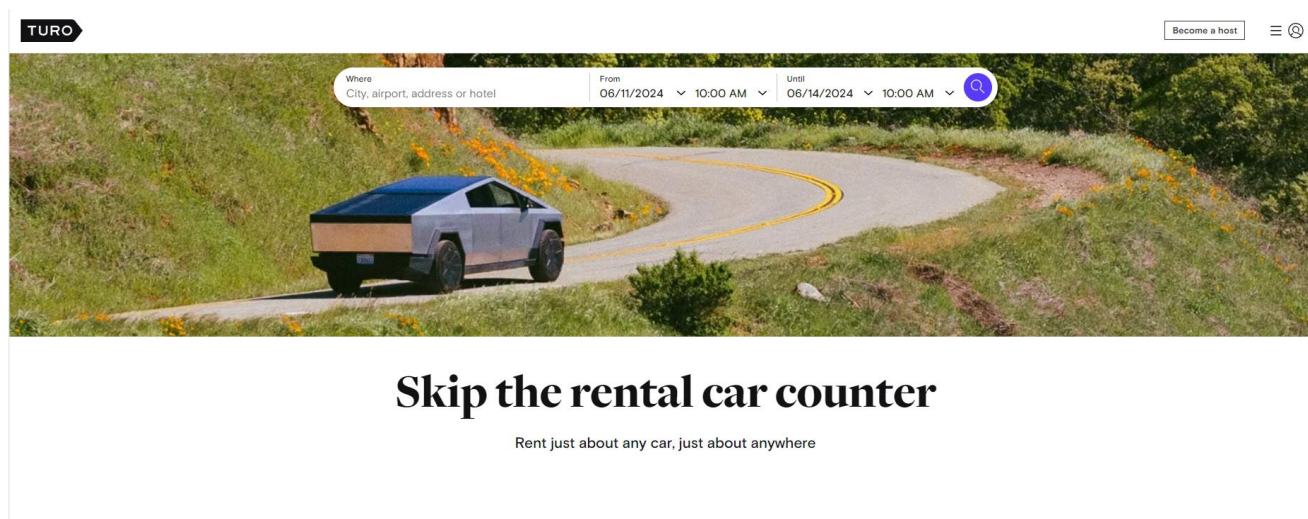


Рисунок 1.3 – Приклад роботи застосунка Turo

Кожен з розглянутих конкурентів має свої унікальні переваги та недоліки, що робить їх привабливими для різних груп користувачів. Zipcar приваблює своєю великою мережею та різноманітністю автопарку, але може бути дорожчим для тривалих поїздок. ShareNow пропонує гнучкість завдяки моделі free-floating, проте має обмежену зону обслуговування. Turo надає користувачам великий вибір автомобілів за конкурентними цінами, але залежить від якості обслуговування приватних власників.

1.3 Постановка задачі

Аналіз конкурентів, таких як Zipcar, Car2Go (тепер частина Share Now) та Turo, показав, що кожен з них має свої унікальні переваги та недоліки.

На основі цього аналізу можна зробити висновок, що для успішного впровадження каршерінг-сервісу необхідно поєднати переваги існуючих рішень та мінімізувати їх недоліки. Важливими аспектами є широка мережа покриття, різноманітність автопарку, гнучкі умови оренди, простота користування та прозорість тарифів.

Метою даної роботи є розробка інноваційного застосунку для каршерінгу, який забезпечуватиме зручність, доступність та ефективність оренди автомобілів для користувачів. Застосунок має відповідати сучасним вимогам ринку та враховувати виявлені сильні сторони і недоліки конкурентів. Тому для успішності застосунку потрібно вирішити наступні завдання:

1. Аналіз потреб користувачів - проведення опитувань та досліджень для визначення основних потреб і очікувань користувачів каршерінг-сервісів. Визначення ключових функціональних вимог до застосунку на основі зібраних даних.
2. Розробка архітектури застосунку - проєктування масштабованої та надійної архітектури, яка забезпечуватиме високу продуктивність та

стійкість до навантажень. Вибір технологій та інструментів, що забезпечать ефективну розробку та підтримку застосунку.

3. Розробка основних функціональних модулів - створення модулів для реєстрації користувачів, бронювання автомобілів, перегляду доступних авто та підтримки зв'язку з користувачами. Впровадження інтерфейсу користувача, що забезпечуватиме зручність та інтуїтивну зрозумілість у користуванні застосунком.
4. Тестування та оптимізація - проведення всебічного тестування застосунку для виявлення та усунення помилок, забезпечення високої якості продукту. Оптимізація продуктивності та швидкодії застосунку для забезпечення безперебійної роботи під час високих навантажень.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ЗАСТОСУНКУ

2.1 Визначення вимог

Перед початком розробки будь-якого програмного продукту важливо чітко визначити вимоги до нього. Це дозволяє створити ефективну, надійну та зручну систему, що відповідатиме очікуванням користувачів і забезпечить високий рівень задоволеності клієнтів. У контексті застосунку, який передбачає оренду автомобілів, необхідно врахувати широкий спектр вимог.

Функціональні вимоги визначають, що саме має робити система, які функції та можливості вона повинна надавати користувачам. Вони включають базові операції, такі як реєстрація користувачів, пошук автомобілів, бронювання, а також додаткові функції, що покращують користувацький досвід, такі як підтримка, рейтингова система та інші [2].

Нефункціональні вимоги, визначають, як система повинна виконувати свої функції. Вони охоплюють аспекти продуктивності, безпеки, надійності, масштабованості, юзабіліті та мобільності. Ці вимоги важливі для забезпечення високої якості сервісу та гарантування, що система буде працювати стабільно і ефективно навіть при значних навантаженнях [3].

Таким чином, розробка застосунку повинна базуватися на детально опрацьованих вимогах, які забезпечать його конкурентоспроможність та задоволення потреб користувачів. Нижче наведені функціональні та нефункціональні вимоги, які необхідно врахувати при створенні цього програмного продукту.

Функціональні вимоги:

1. Реєстрація та авторизація користувачів:

- Можливість створення облікового запису користувача, ввівши свої дані.
- Підтвердження електронної пошти.
- Авторизація через електронну пошту.

2. Профіль користувача:

- Перегляд та редагування особистих даних.
- Історія орендованих авто.

3. Перегляд авто:

- Можливість переглянути автопарк.
- Перегляд детальної інформації кожного авто.

4. Бронювання авто:

- Можливість вибору авто після перегляду або у автопарку.
- Можливість вибрати потрібне місце, дату та час початку і кінця оренди.
- Введення персональних даних та підтвердження бронювання через пошту.

5. Підтримка користувачів:

- Сторінка для надсилання детальних питань.
- FAQ, тобто найпоширеніші питання.

6. Рейтинг та відгуки:

- Можливість залишати відгуки про авто та сервіс.
- Перегляд відгуків інших користувачів.

Нефункціональні вимоги:

1. Безпека:

- Захист особистих даних користувачів.
- Шифрування приватної інформації (паролі, платіжні дані).
- Захист від несанкціонованого доступу та використання.

2. Продуктивність:

- Швидка обробка запитів користувачів.
- Висока продуктивність під час високих навантажень.
- Система повинна використовувати мінімальні ресурси.

3. Надійність:

- Застосунок повинен бути надійним та працювати без збоїв.
- Можливість швидко відновлюватися в випадку збою системи.

- Застосунок повинен мати резервні копії даних на випадок збою.

4. Масштабованість:

- Можливість масштабування системи для обробки зростаючої кількості користувачів без значних змін архітектури.
- Можливість підтримувати велике число одночасних користувачів.

5. Зручність:

- Інтуїтивно зрозумілий інтерфейс для користувачів.
- Адаптивність під різні розміри моніторів чи девайси (ПК, мобільні пристрої, планшети).

6. Ефективність обслуговування:

- Мінімальний час відгуку на запити користувачів.
- Швидке вирішення проблем через службу підтримки.

2.2 Опис функціоналу системи

Наш застосунок пропонує різноманітні функції для задоволення потреб користувачів. Залежно від рівня доступу, користувачі можуть виконувати різні дії та отримувати доступ до різних розділів застосунку. Ознайомтесь з функціоналом, доступним для кожного типу користувача, щоб дізнатися більше про можливості нашого сервісу.

Функціонал неавторизованого користувача:

1. Реєстрація у системі. Користувач повинен вказати свою електронну адресу, яка буде слугувати логіном у системі, пароль та підтвердити пароль.
2. Авторизація. Користувачу потрібно буде ввести логін та пароль, які він вказував під час реєстрації. Якщо все вірно – користувач увійде до системи.
3. Перегляд головної сторінки з загальною інформацією.
4. Перегляд автопарку.

5. Ознайомлення з відгуками клієнтів.
6. Перегляд сторінки “Про нас” для детальнішим ознайомленням сервісу.
7. Перегляд FAQ, тобто найпоширеніших питань.
8. Можливість підписатися на отримання новин застосунку.
9. Можливість надсилати розгорнуте запитання на яке у найкоротший час відповідь кваліфікований консультант.

Функціонал авторизованого користувача:

1. Перегляд автопарку.
2. Вибір авто для бронювання.
3. Бронювання авто. Потрібно обрати авто, вказати місце та дату початку та кінця оренди. Система найде вільний варіант та надасть можливість користувачу підтвердити бронювання.
4. Підтвердження оренди після бронювання. Потрібно ввести детальну інформацію про користувача (ім’я, прізвище, номер телефону, вік, адресу та місто проживання, поштовий індекс) та підтвердити оренду.
5. Перегляд історії оренди.
6. Ознайомлення з відгуками клієнтів.
7. Залишення відгука.
8. Перегляд сторінки “Про нас” для детальнішим ознайомленням сервісу.
9. Перегляд FAQ, тобто найпоширеніших питань.
10. Можливість підписатися на отримання новин застосунку.
11. Можливість надсилати розгорнуте запитання на яке у найкоротший час відповідь кваліфікований консультант.

Функціонал адміністратора:

1. Додавання нових автомобілів.
2. Перегляд автопарку.
3. Редагування інформації про існуючі автомобілі (ціна, характеристики, фото тощо).
4. Видалення автомобілів.
5. Перегляд історії оренд користувачів.

6. Блокування або видалення облікових записів користувачів у разі порушень правил користування.

2.3 Діаграми прецедентів та сценарії

Прецедент - це техніка моделювання в системному аналізі та проєктуванні програмного забезпечення, що використовується для опису функціональності системи з точки зору її користувачів.

Кожен прецедент описує конкретний сценарій взаємодії між користувачем та системою, який включає у себе послідовність подій та дій, що відбуваються в процесі досягнення певної мети.

Вони допомагають уточнити потреби користувачів та відобразити ключові сценарії взаємодії, що дозволяє краще розуміти функціональність системи та її можливості [4].

Наступні діаграми прецедентів стануть основою для подальшого аналізу та проєктування системи, сприяючи успішному розробленню програмного забезпечення, яке відповідає потребам та очікуванням користувачів.

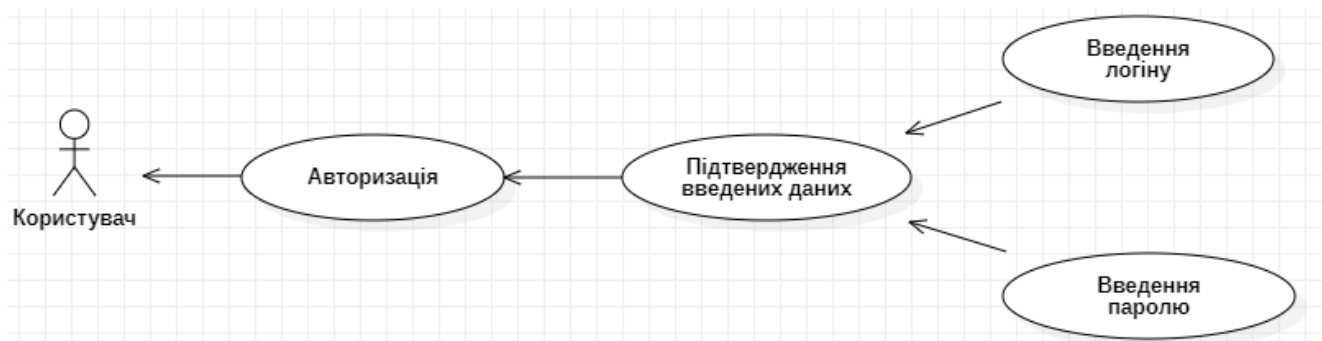


Рисунок 2.3 – Авторизація користувача

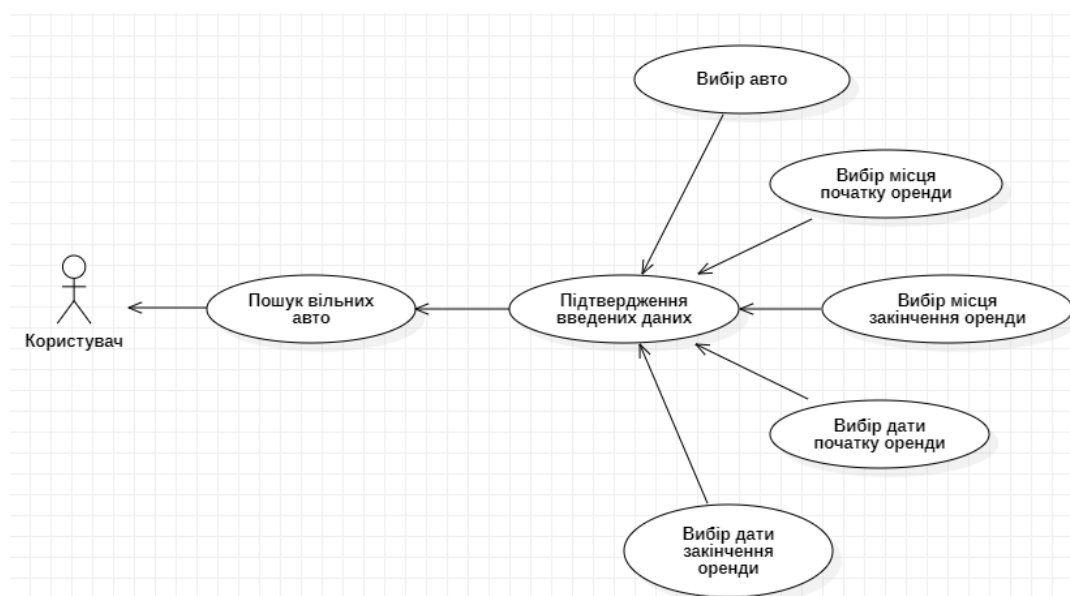


Рисунок 2.4 – Бронювання вільного авто

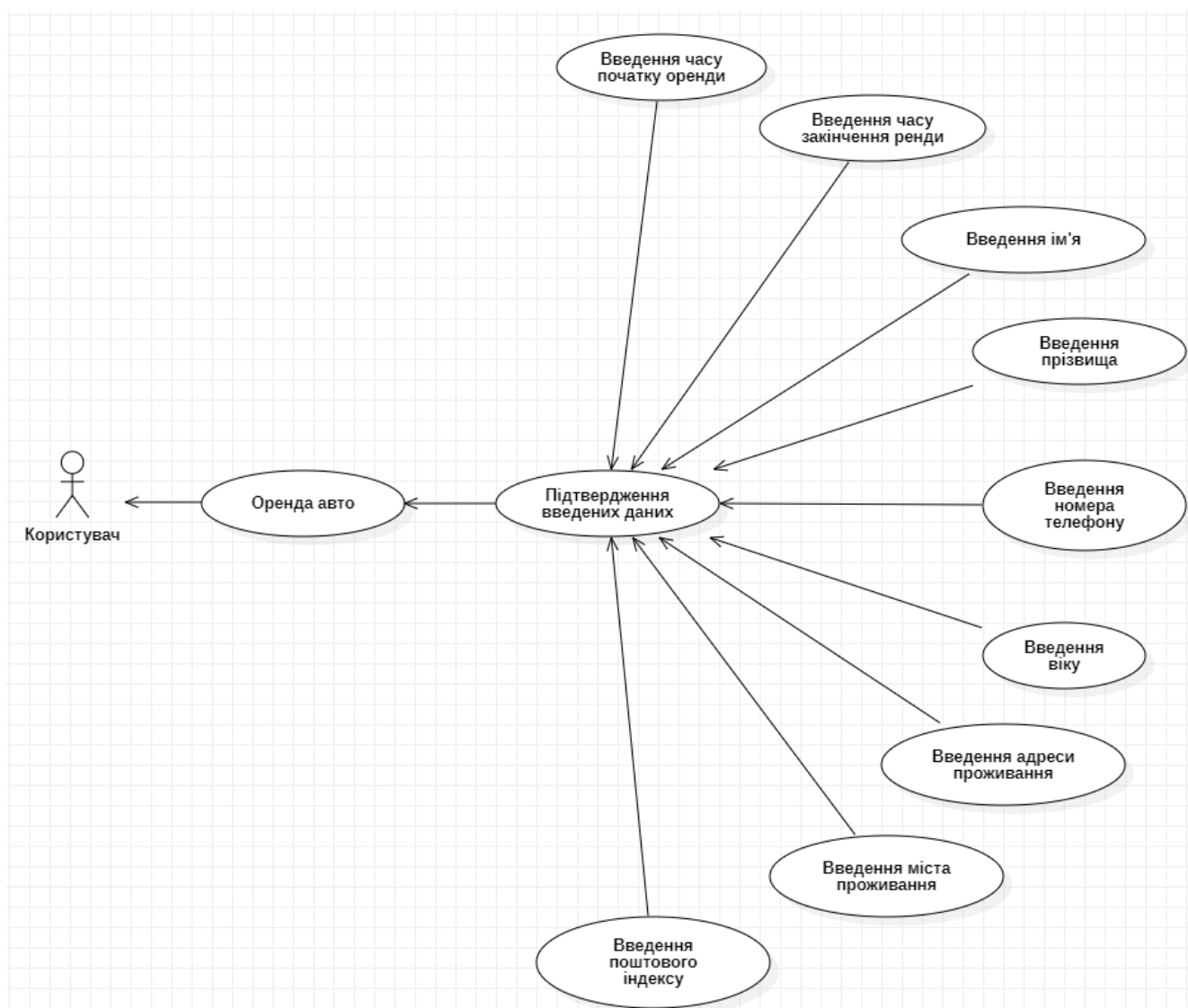


Рисунок 2.5 – Підтвердження оренди авто

Для успішного проєктування та розробки системи оренди автомобілів важливо розглянути основні сценарії взаємодії користувачів із застосунком. Вони описують типові послідовності дій, які виконують користувачі для досягнення конкретних цілей у системі.

Перед початком користування системою оренди автомобілів, новий користувач повинен створити обліковий запис. Реєстрація дозволяє користувачу отримати доступ до всіх функцій застосунку, таких як залишення відгуків, бронювання та оренда автомобілів. Нижче наведений сценарій у таблиці описує процес реєстрації користувача у системі, включаючи всі необхідні кроки та перевірки.

Таблиця 2.1

Назва	Реєстрація користувача
Актори	Користувач
Передумови	1. Користувач повинен мати доступ до інтернету. 2. Користувач не має існуючого облікового запису в системі.
Основний сценарій	1. Користувач відкриває сторінку реєстрації у браузері. 2. Система запитує у користувача введення необхідних даних (електронна пошта, пароль). 3. Користувач заповнює форму реєстрації та натискає кнопку "Реєстрація". 4. Система перевіряє введені дані на коректність та унікальність електронної пошти. 5. У разі вірності всіх даних, система активує обліковий запис користувача та повідомляє його про успішну реєстрацію.
Альтернативний сценарій	Якщо дані введені некоректно або електронна пошта вже використовується, система відображає відповідне повідомлення про помилку та пропонує виправити дані.
Післяумови	Користувач зареєстрований у системі та може увійти до свого облікового запису.

Після успішної реєстрації користувачі можуть увійти до своїх облікових записів для доступу до персоналізованих функцій системи. Авторизація забезпечує безпеку та конфіденційність даних користувача. Сценарій авторизації у таблиці нижче описує процес входу в систему, включаючи перевірку даних та можливі помилки.

Таблиця 2.2

Назва	Авторизація користувача
Актори	Користувач
Передумови	Користувач повинен бути зареєстрований у системі
Основний сценарій	1. Користувач відкриває сторінку входу у браузері. 2. Система запитує у користувача введення електронної пошти та паролю. 3. Користувач вводить електронну пошту та пароль і натискає кнопку "Увійти". 4. Система перевіряє введені дані на коректність. 5. Система дозволяє користувачу доступ до облікового запису та перенаправляє його на головну сторінку.
Альтернативний сценарій	Якщо дані введені некоректно або обліковий запис не знайдений, система відображає відповідне повідомлення про помилку та пропонує повторити спробу.
Післяумови	Користувач успішно увійшов у свій обліковий запис.

Користувачі можуть забронювати автомобіль на певний період через систему оренди автомобілів. Бронювання включає вибір автомобіля, вказання місця та часу оренди, а також заповнення особистих даних. Цей сценарій описує детальний процес бронювання автомобіля у системі.

Таблиця 2.3

Назва	Бронювання авто
Актори	Користувач
Передумови	Користувач повинен бути авторизований у системі.

Продовження таблиці 2.3

Основний сценарій	1. Користувач вибирає автомобіль зі списку. 2. Система запитує місто початку та кінця оренди, дату початку та кінця оренди у відповідних полях. 3. Користувач заповнює поля та натискає кнопку "Пошук". 4. Система шукає вільне авто та дає можливість користувачу орендувати його.
Альтернативний сценарій	Якщо місто або дати не вибрані, система відображає відповідне повідомлення про помилку та пропонує заповнити всі поля.
Післяумови	Автомобіль успішно заброньований.

Після бронювання автомобіля користувач має можливість орендувати його на обраний період. Цей сценарій у описує процес оренди автомобіля, включаючи взаємодію з системою для отримання доступу до автомобіля та завершення оренди.

Таблиця 2.4

Назва	Оренда авто
Актори	Користувач
Передумови	Користувач повинен бути авторизований у системі.
Основний сценарій	1. Після закінчення бронювання система дає можливість орендувати авто. 2. Користувач заповнює потрібні поля (час початку та закінчення оренди, ім'я, прізвище, номер телефону, вік, адреса проживання, місто проживання та поштовий індекс) 3. Користувач натискає кнопку "Орендувати"

Продовження таблиці 2.4

Альтернативний сценарій	Якщо дані введені некоректно, система відображає відповідне повідомлення про помилку та пропонує повторити спробу.
Післяумови	Авто успішно орендовано, користувач отримав підтвердження.

2.4 Діаграми послідовності

Для більш детального розуміння функціонування системи оренди автомобілів та взаємодії між різними її компонентами використовуються діаграми послідовності. Діаграми послідовності є важливим інструментом у проектуванні програмного забезпечення, оскільки вони наочно демонструють, як об'єкти системи взаємодіють між собою в певних сценаріях.

Ці діаграми відображають послідовність повідомлень, що обмінюються між об'єктами, і допомагають зрозуміти динамічний аспект системи. Вони показують хронологічний порядок виконання операцій та взаємодій, дозволяючи виявити потенційні проблеми та покращити архітектуру системи [5].

Тут будуть представлені діаграми послідовності для ключових сценаріїв використання застосунка. Ці діаграми допоможуть детально вивчити процеси, що відбуваються в системі, і забезпечити ефективну розробку програмного забезпечення, яке відповідатиме всім функціональним та нефункціональним вимогам.



Рисунок 2.6 – Діаграма послідовності реєстрації

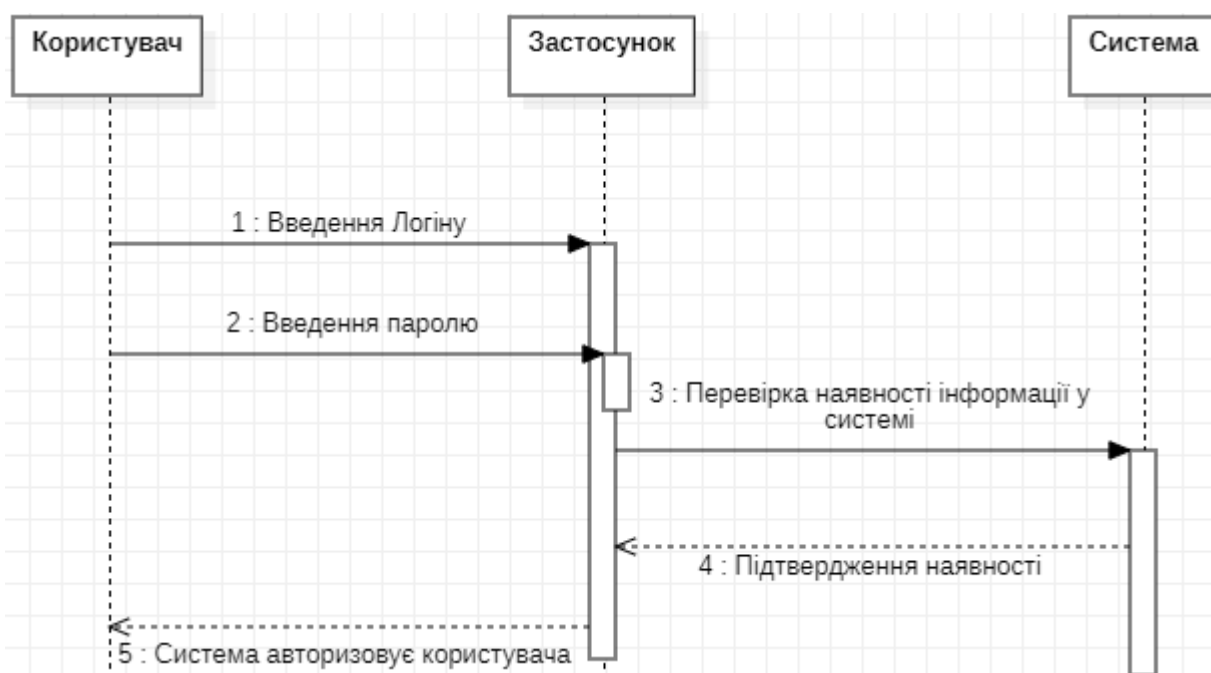


Рисунок 2.7 – Діаграма послідовності авторизації



Рисунок 2.8 – Діаграма послідовності бронювання



Рисунок 2.9 – Діаграма послідовності оренди



Рисунок 2.10 – Діаграма послідовності залишення відгуку



Рисунок 2.11 – Діаграма послідовності перегляду автопарку

РОЗДІЛ 3. РОЗРОБКА ЗАСТОСУНКУ

3.1 Вибір технологій

У процесі розробки застосунку для оренди автомобілів було обрано ряд сучасних веб-технологій, які забезпечують ефективність, зручність та високу продуктивність роботи системи. Основними технологіями, які використовуються у проєкті є HTML, CSS, JavaScript та React. Нижче наведено детальний опис кожної з цих технологій та обґрунтування їх вибору.

HTML є основною мовою розмітки для створення веб-сторінок. Вона використовується для структурування контенту на веб-сторінках, визначення елементів, таких як заголовки, абзаци, списки, посилання, зображення та інші мультимедійні об'єкти [6].

Переваги:

- Простота та зрозумілість: HTML легко засвоюється та використовується, що дозволяє швидко створювати базову структуру веб-сторінок.
- Сумісність: HTML підтримується всіма веб-браузерами, забезпечуючи стабільне відображення контенту на різних платформах.
- Стандартизація: HTML є стандартизованою мовою, що дозволяє створювати веб-сторінки, які відповідають міжнародним стандартам.

HTML забезпечує основу для створення веб-сторінок застосунку, дозволяючи структуровано розмістити весь необхідний контент та інтерактивні елементи.

CSS використовується для опису вигляду та форматування веб-сторінок, включаючи розташування елементів, кольори, шрифти, відступи та інші стилі [7].

Переваги:

- Розділення контенту та стилів: CSS дозволяє відокремити візуальні стилі від HTML-коду, спрощуючи підтримку та зміну дизайну.

- Гнучкість та контроль: CSS забезпечує високий рівень контролю над зовнішнім виглядом веб-сторінок, дозволяючи створювати адаптивний дизайн для різних пристроїв.
- Оптимізація продуктивності: CSS сприяє оптимізації продуктивності веб-сторінок завдяки використанню кешування та зменшенню обсягу HTML-коду.

CSS дозволяє створювати привабливий та професійний дизайн для застосунку, забезпечуючи при цьому адаптивність та сумісність з різними пристроями.

JavaScript є мовою програмування, що використовується для додавання інтерактивності та динамічних елементів до веб-сторінок. Вона дозволяє створювати складні функції, обробляти події та взаємодіяти з сервером без необхідності перезавантаження сторінки [8].

Переваги:

- Динамічність: JavaScript дозволяє створювати динамічні та інтерактивні веб-додатки, забезпечуючи кращий користувацький досвід.
- Широке застосування: JavaScript підтримується всіма сучасними веб-браузерами та має велику спільноту розробників, що забезпечує наявність великої кількості бібліотек та інструментів.
- Розширені можливості: JavaScript дозволяє реалізовувати складні функції, такі як анімації, обробка форм, взаємодія з API та багато іншого.

JavaScript забезпечує необхідну інтерактивність та динамічність застосунку, дозволяючи створювати багатофункціональний та зручний інтерфейс користувача.

React є популярною бібліотекою для створення користувацьких інтерфейсів. Вона дозволяє розробникам створювати компоненти, які можуть бути повторно використані в різних частинах застосунку, що спрощує розробку та підтримку коду [9].

Переваги:

- Компонентна архітектура: React дозволяє створювати компоненти, які можуть бути легко повторно використані та комбіновані для створення складних інтерфейсів.
- Висока продуктивність: React використовує віртуальний DOM, що забезпечує високу продуктивність та ефективне оновлення інтерфейсу.
- Широка екосистема: React має велику екосистему бібліотек та інструментів, що дозволяє швидко розширювати функціональність застосунку.

Обрані технології HTML, CSS, JavaScript та React забезпечують потужний та гнучкий набір інструментів для створення сучасного та ефективного застосунку для оренди автомобілів. Вони дозволяють створювати структуровані, привабливі та інтерактивні веб-сторінки, які відповідають потребам користувачів та забезпечують високий рівень продуктивності та зручності.

3.2 Вибір середовища розробки

Для розробки застосунку для оренди автомобілів було обрано середовище розробки Visual Studio Code (VSCode). VSCode є популярним редактором коду, який надає розробникам потужний набір інструментів для ефективної роботи з різними мовами програмування та технологіями. Нижче наведено детальний опис цього середовища розробки, а також обґрунтування його вибору та переваги.

Visual Studio Code – це редактор коду від компанії Microsoft. Він відомий своєю швидкістю, легкістю, потужними функціями та підтримкою широкого спектру мов програмування та технологій. VSCode доступний для Windows, macOS та Linux, що робить його універсальним вибором для розробників [10].

Переваги:

Підтримка різних мов програмування: VSCode підтримує багато мов програмування "з коробки" і має можливість розширювати підтримку через розширення. Це дозволяє використовувати один інструмент для роботи з HTML, CSS, JavaScript, React та іншими технологіями.

Інтелектуальне автодоповнення коду: завдяки функціям IntelliSense, VSCode надає контекстне автодоповнення коду, що прискорює написання коду та зменшує кількість помилок.

Інтеграція з системами контролю версій: VSCode має вбудовану підтримку Git, що дозволяє розробникам легко виконувати операції з системою контролю версій безпосередньо з редактора.

Підтримка розширень: велика кількість доступних розширень дозволяє розширювати функціональність редактора відповідно до потреб проєкту. Наприклад, розширення для React, Prettier для автоматичного форматування коду, ESLint для аналізу коду тощо.

Налагодження: VSCode має вбудовані інструменти для налагодження коду, що дозволяє розробникам знаходити та виправляти помилки швидко і ефективно.

Термінал: вбудований термінал дозволяє розробникам виконувати командні операції без необхідності залишати редактор.

Живий сервер: розширення Live Server дозволяє запускати локальний сервер для перегляду змін у реальному часі, що значно прискорює процес розробки.

Visual Studio Code було обрано як середовище розробки для проєкту оренди автомобілів через його універсальність, потужні функції, підтримку різних мов програмування та можливість розширення функціональності через розширення. Це середовище забезпечує ефективний та зручний інструмент для роботи з кодом, що сприяє підвищенню продуктивності та якості розробки програмного забезпечення.

3.3 Реалізація застосунка та його компонентів

3.3.1 Створення логотипу

Логотип є важливим елементом будь-якого бренду, оскільки він допомагає формувати візуальну ідентичність компанії та підвищує впізнаваність серед користувачів. Для проєкту оренди автомобілів було розроблено унікальний логотип, який символізує основні цінності та послуги, що надаються компанією. Нижче наведено опис процесу створення логотипа та його ключові особливості.

Логотип складається з графічного зображення автомобіля з символом бездротового зв'язку над ним та текстового елемента з назвою "ОРЕНДА Авто". Основні кольори логотипа - червоний та чорний.

Автомобіль виконаний у стилізованій формі, що символізує мобільність, динамічність та сучасність. Символ бездротового зв'язку над автомобілем підкреслює цифровий характер послуг компанії, що надаються через онлайн-застосунок. Назва "ОРЕНДА Авто" написана простим, але елегантним шрифтом, який забезпечує чіткість та легкість читання. Чорний колір тексту створює контраст з яскравим червоним графічним елементом, що робить логотип візуально привабливим та збалансованим.

Процес створення логотипа:

1. Дослідження: На першому етапі було проведено дослідження ринку та аналіз конкурентів для визначення основних тенденцій у дизайні логотипів для сервісів оренди автомобілів. Було важливо створити логотип, який виділяється серед конкурентів і водночас відображає унікальні характеристики нашої компанії.
2. Концептуалізація: На основі зібраної інформації були розроблені кілька концепцій логотипа. Основна ідея полягала в тому, щоб поєднати елементи, що асоціюються з автомобілями та цифровими технологіями, підкреслюючи сучасний та інноваційний підхід компанії.

3. Ескізи та прототипи: Було створено декілька ескізів та прототипів логотипа, з яких обрано найвдаліший варіант. У процесі роботи над ескізами враховувалися такі фактори, як простота, зчитуваність, впізнаваність та відповідність загальній концепції бренду.
4. Тестування та внесення змін: Перед остаточним вибором логотип був протестований у різних масштабах та на різних носіях (веб-сайт, мобільний застосунок, друковані матеріали). На основі отриманих результатів було внесено необхідні корективи для забезпечення оптимальної якості та впізнаваності логотипа.

Логотип застосунка зображений на рис. 3.1.



Рисунок 3.1 – Логотип застосунка

3.3.2 Створення інтерфейсу головної сторінки застосунка

Головна сторінка веб-застосунку для оренди автомобілів відіграє ключову роль у залученні користувачів та представленні основних послуг компанії. Вона повинна бути зрозумілою, привабливою та зручною для навігації, щоб забезпечити позитивний досвід користувача та стимулювати подальші дії, такі як реєстрація або бронювання автомобіля. Нижче наведено опис процесу створення головної сторінки, зображеної на рис. 3.2.

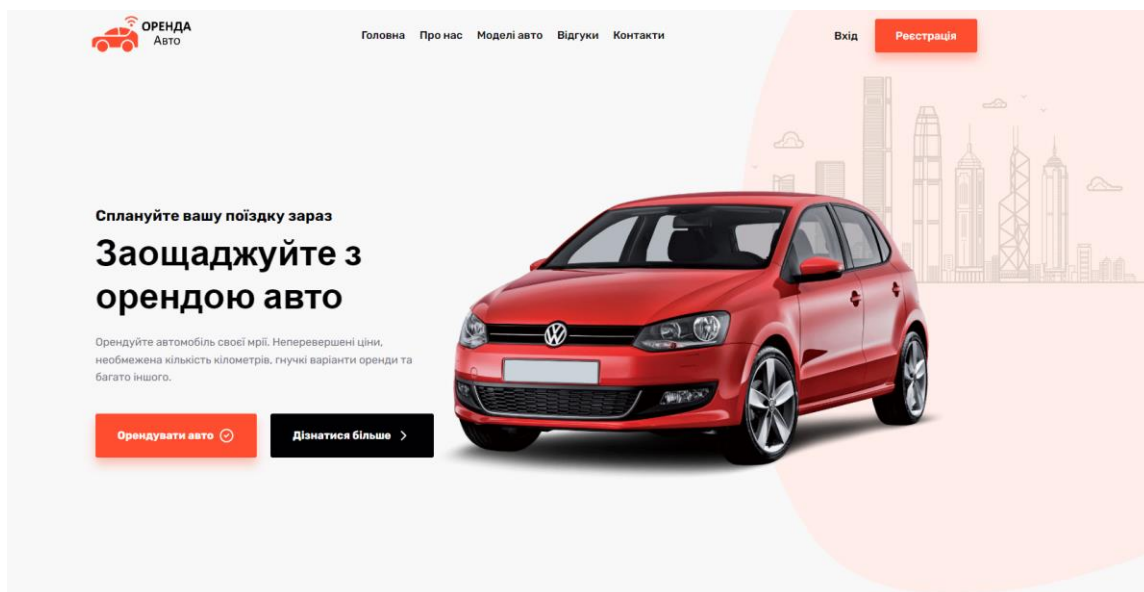


Рисунок 3.2 – Головна сторінка

Опис дизайну головної сторінки:

В лівому верхньому куті розміщено логотип компанії, який складається з графічного елемента (автомобіль з символом бездротового зв'язку) та тексту "Оренда Авто".

Праворуч від логотипу розміщено навігаційну панель з посиланнями на наступні розділи: "Головна", "Про нас", "Моделі авто", "Відгуки", "Контакти".

У верхньому правому куті знаходяться кнопки "Вхід" та "Реєстрація" для користувачів.

Основний блок складається з великого зображення автомобіля на тлі міського пейзажу. Ліворуч від автомобіля розташовано заголовок та підзаголовок, які привертають увагу користувачів та повідомляють про можливості економії з орендою авто. Під заголовком знаходиться короткий опис послуг, які надає компанія, що включає оренду автомобілів за вигідними цінами, необмежену кількість кілометрів, гнучкі варіанти оренди та інші переваги.

Під текстовим блоком розташовані дві кнопки дій: "Орендувати авто" та "Дізнатися більше". Кнопка "Орендувати авто" виділена яскравим червоним кольором для привернення уваги та стимулювання користувача до дії. Вона перенаправить користувача до розділу бронювання, де користувач зможе обрати

авто, заповнити необхідні поля та забронювати авто. Кнопка "Дізнатися більше" перенаправить користувача на сторінку "Про нас", де користувач зможе дізнатися більше інформації про сервіс застосунка.

Прогорнувши головну сторінку вниз, користувач може побачити трохи більше інформації про наш сервіс. Це зображено на рис. 3.3.

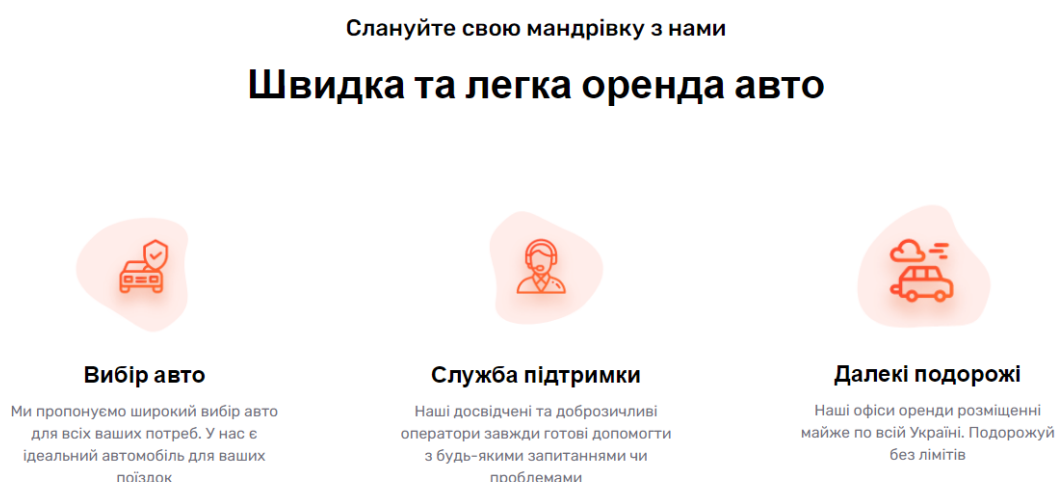


Рисунок 3.3 – Елемент з інформацією про сервіс

Нижче інформації про сервіс, на головній сторінці розташований інтерактивний елемент перегляду автопарку. Він зображений на рис. 3.4.

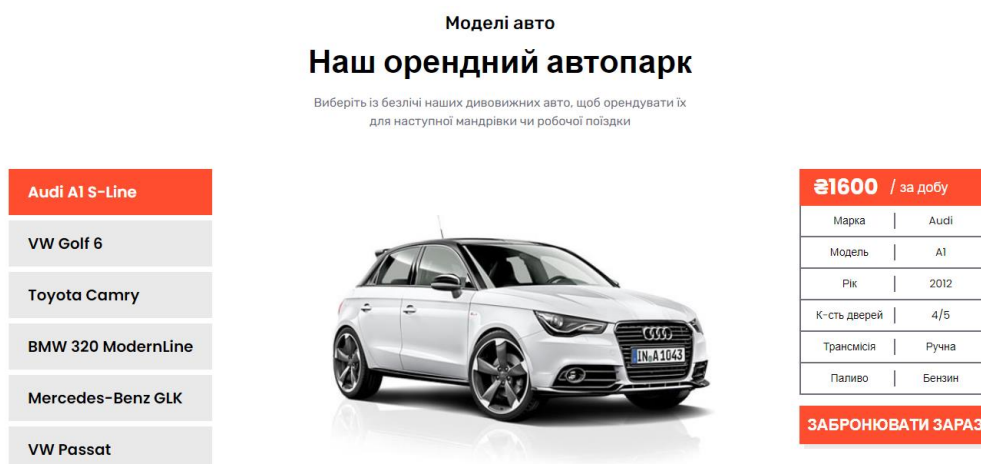


Рисунок 3.4 – Елемент перегляду автопарку

Користувач може обрати потрібний автомобіль із списку, переглянути його фото та детальну інформацію:

- Вартість авто за добу
- Марка авто
- Модель авто
- Кількість дверей
- Трансмісія або коробка передач
- Паливо на якому їздить авто

Ще нижче на головній сторінці розташовані приємні для перегляду елементи з цікавою інформацією про наш сервіс та наші переваги, для заохочення орендувати авто саме у нас, а кнопка “Детальніше” перенаправить користувача на сторінку “Про нас”. Це зображено на рис. 3.5.

Заощаджуйте з нашим сервісом оренди авто!

Офіси у всіх великих містах України. Багато пропозицій. **24/7** Підтримка.

Чому Вам слід вибрати нас

Найкращі пропозиції, які ви коли-небудь знайдете

Відкрийте для себе найкращі пропозиції з нашим неперевершеним сервісом. Ми прагнемо надати вам найкращу якість за ваші кошти, щоб ви могли насолоджуватися нашими послугами, не розбиваючи копійку. Наші пропозиції спрямовані надати вам найкращий досвід оренди, тож не пропустіть ваш шанс.

Детально >

Поїздка країною

Підніміть свої враження від водіння на новий рівень з нашими першокласними автомобілями для ваших пригод по Україні.

Приємна ціна

Отримайте все необхідне за однією зручною та прозорою ціною.

Без зайвих платежів

Насолоджуйтеся спокоєм завдяки нашій політиці відсутності прихованих платежів. Ми віримо в прозорість та чесне ціноутворення.

Рисунок 3.5 – Елементи заохочення

Також на головній сторінці розташовані поширені запитання (FAQ), де користувач може ознайомитись із питаннями які можливо його цікавили.

Натиснувши на потрібне питання, з'являється відповідь на нього. Елемент FAQ зображений на рис. 3.6.

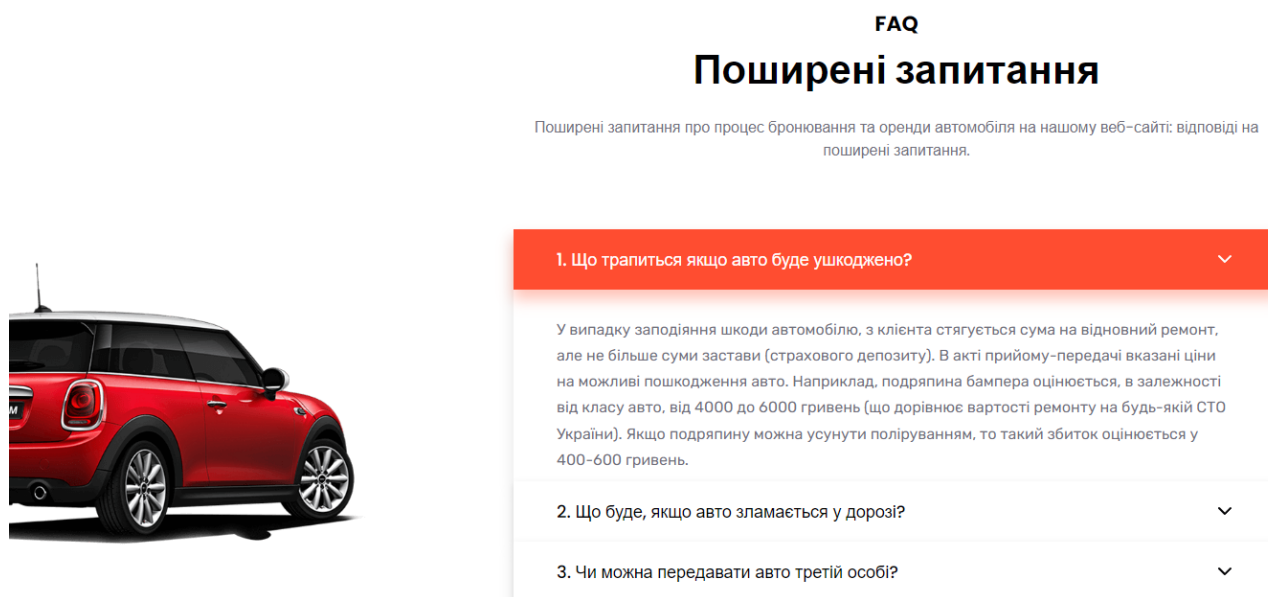


Рисунок 3.6 – Елемент “Поширені запитання”

Головна сторінка закінчується футером, де розташована назва сервісу “ОРЕНДА Авто”, текст інформації, контакти для зворотного зв’язку, дні та години роботи, а також можливість ввести свою пошту для отримання новин сервісу. Футер зображений на рис. 3.7.

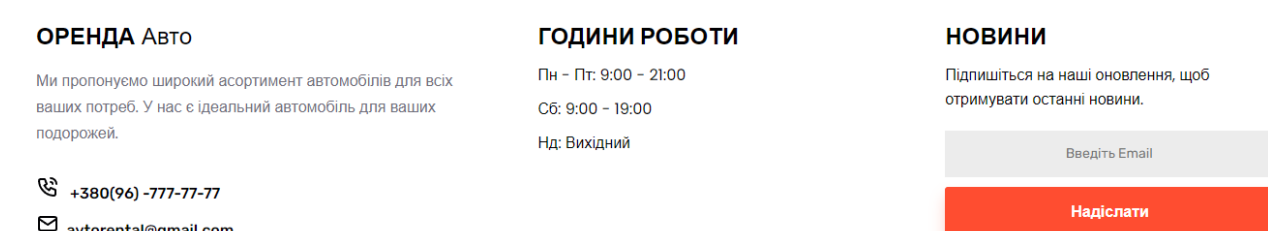
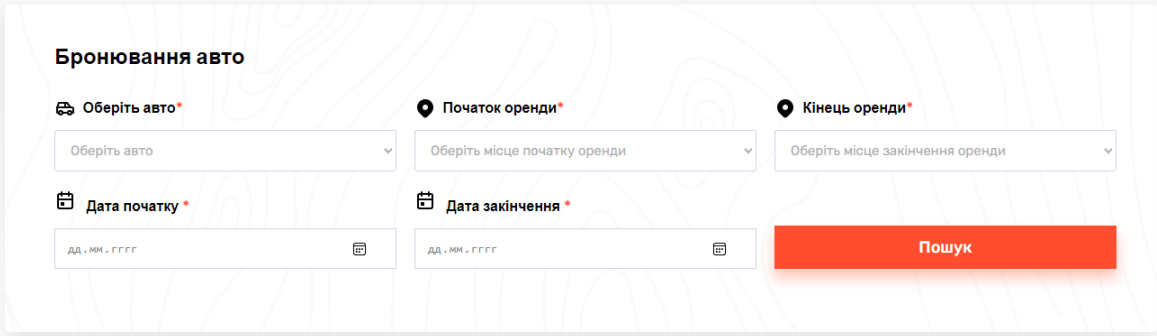


Рисунок 3.7 – Футер сторінки

3.3.3 Створення інтерфейсу бронювання авто

Інтерфейс бронювання автомобіля також є одним з ключових елементів веб-застосунку, що дозволяє користувачам зручно і швидко вибрати та забронювати автомобіль для оренди. В цьому розділі описано структуру та елементи інтерфейсу бронювання, представлені на рис 3.8.



The screenshot shows a web form for car rental booking. The title is "Бронювання авто". The form contains six input fields arranged in two rows. The first row has three dropdown menus: "Оберіть авто*" (with a car icon), "Початок оренди*" (with a location pin icon), and "Кінець оренди*" (with a location pin icon). The second row has two date input fields: "Дата початку*" and "Дата закінчення*" (both with calendar icons), and a red "Пошук" button on the right. Each input field has a placeholder text indicating what to select or enter.

Рис 3.8 – Елемент “Бронювання авто”

Опис елементу “Бронювання авто”:

Інтерфейс складається з кількох основних полів та кнопки для виконання пошуку.

Заголовок "Оренда авто" розташований у верхньому лівому куті форми, цей заголовок чітко вказує на призначення інтерфейсу.

Поле вибору автомобіля - це випадаюче меню з підказкою "Оберіть авто". Користувач може вибрати потрібний автомобіль зі списку доступних.

Поле вибору "Початок оренди" та "Кінець оренди" – це також випадаючі поля де користувач може обрати місце початку та закінчення оренди, тобто міста.

Поля вибору "Дата початку" та "Дата закінчення" – це поля, в якому користувачу потрібно обрати потрібну дату. Натиснувши на картинку календаря, з'являється інтерактивний календар де користувач може обрати дату.

Кнопка "Пошук" ця кнопка виконує пошук вільного авто після заповнення всіх необхідних полів користувачем.

3.3.4 Створення інтерфейсу оренди авто

Інтерфейс оренди авто дозволяє користувачу орендувати авто після його бронювання, ввівши усі необхідні поля. Повна структура зображена у додатку А.

Опис елементу “Оренда авто”:

Інтерфейс складається з кількох основних полів та кнопки для підтвердження бронювання.

Заголовок "Завершення бронювання авто" вказує що користувач вже завершує бронювання, і ввівши всі необхідні дані та натиснувши кнопку “Орендувати”, зможе орендувати авто.

Під заголовком розміщено повідомлення про те, що після заповнення форми клієнт отримає на пошту талон на оренду, який потрібно пред'явити після прибуття в пункт прокату, а також безкоштовний номер служби підтримки клієнтів.

Поле "Дата та час початку оренди": поле де користувач може перевірити введenu дату початку оренди та ввести потрібний час.

Поле "Дата та час закінчення оренди": поле де користувач може перевірити введenu дату закінчення оренди та ввести потрібний.

Поля "Місце початку" та "Місце закінчення" також дають користувачу змогу упевнитись у вірності введених даних.

Поле “Інформація про автомобіль” показує користувачу автомобіль, який він обрав.

Елемент “Інформація про клієнта” містить поля, які обов’язково потрібно заповнити, щоб завершити оренду авто.

Чекбокс "Я погоджуюсь на отримання останніх новин" дає користувачу, за його згодою, отримувати останні новини сервісу.

Кнопка “Орендувати” перевіряє та підтверджує усі вхідні дані користувача, у разі їх валідності система орендує авто.

3.3.5 Створення сторінки “Про нас”

Після переходу на сторінку “Про нас”, користувач побачить під навігацію застосунка, назву сторінки на якій він перебуває. Ще нижче буде розміщена картинка одного з офісів та детальна інформація про наш сервіс. Вигляд сторінки “Про нас” зображена на рис. 3.9.

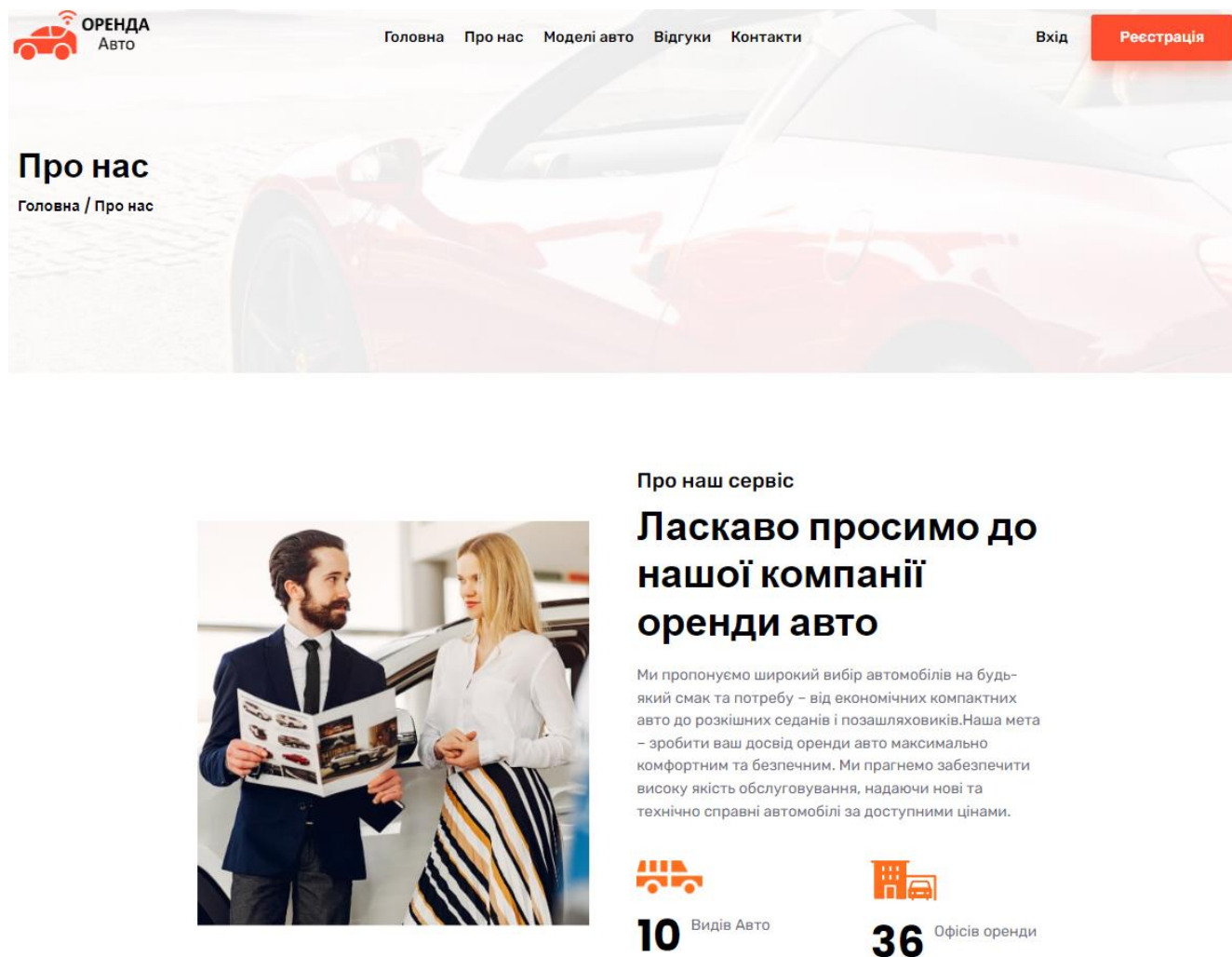


Рисунок 3.9 – Сторінка “Про нас”

3.3.6 Створення сторінки “Моделі авто”

Сторінка перегляду доступних автомобілів є важливим елементом застосунку, який дозволяє користувачам швидко ознайомитися з наявними варіантами та вибрати автомобіль, який відповідає їхнім потребам. Сторінка зображена у додатку Б.

Інтерфейс складається з декількох ключових полів та віджетів, які надають користувачам повну інформацію про доступні для оренди автомобілі та їх деталі.

Список автомобілів складається із фото авто та їхніх повних характеристик нижче фото. Основні характеристики:

- Марка та модель: вказує виробника і модель автомобіля (наприклад, «Volkswagen Golf 6»).
- Ціна за добу: відображає вартість оренди автомобіля за один день (наприклад, «700 грн/доба»).
- Кількість дверей: зазначає, скільки дверей має автомобіль (наприклад, «4 двері»).
- Трансмісія: вказує тип трансмісії автомобіля (наприклад, «Автоматична» або «Механічна»).
- Паливо: зазначає тип палива, який використовує автомобіль (наприклад, «Бензин», «Дизель» або «Гібрид»).

Також під кожним авто розміщена кнопка «Забронювати», яка перенаправить користувача до елемента бронювання авто.

3.3.7 Створення сторінки «Контакти»

Інтерфейс сторінки контактів призначений для зручного зв'язку користувачів із службою підтримки. Ця сторінка містить необхідні контактні дані компанії та форму для відправки повідомлень. Інтерфейс сторінки зображений на рис. 3.10.

Потрібна додаткова інформація?

Надішліть своє запитання повідомленням і наш спеціаліст з служби підтримки надасть Вам розгорнуту відповідь.

☎ 0(96) 777-7777

✉ avtorental@gmail.com

📍 Тернопіль, Україна

Ім'я та прізвище *

Напр. "Назар Присяжнюк"

Email *

youremail@example.com

Опишіть ваше питання *

Пишіть тут..

✉ Надіслати повідомлення

Рисунок 3.10 – Сторінка “Контакти”

Нижче наведено детальний опис елементів інтерфейсу, представлених на зображенні.

Під заголовком розташований текст, який пропонує користувачам надіслати свої запитання повідомленням, та обіцяє, що спеціаліст з служби підтримки надасть розгорнуту відповідь. Ліворуч під інструкцією знаходяться контактні дані компанії, де кожен елемент контактної інформації супроводжується відповідною іконкою для полегшення візуального сприйняття.

Праворуч розташована форма, яка дозволяє користувачам надіслати повідомлення безпосередньо зі сторінки. Форма складається з наступних полів:

Текстове поле з підказкою «Напр. Назар Присяжнюк», де користувач може ввести своє ім'я та прізвище. Це поле є обов'язковим для заповнення, про що свідчить зірочка (*).

Наступне текстове поле – місце введення електронної пошти з підказкою «youremail@example.com». Користувач вводить свою пошту на яку хоче отримати відповідь від служби підтримки.

Останнє велике текстове поле для введення тексту та надання можливості користувачу ввести своє розгорнуте запитання.

Особливості та переваги інтерфейсу:

- Зручність використання: Чітке розміщення елементів та інтуїтивно зрозумілі підказки роблять форму зручною для користувачів.
- Повна контактна інформація: Користувачі мають змогу швидко знайти всі необхідні контактні дані для зв'язку з компанією.
- Естетичний дизайн: Використання іконок та привабливих кольорів покращує візуальне сприйняття сторінки та робить її привабливою для користувачів.
- Форма для повідомлень: Можливість надіслати запитання безпосередньо зі сторінки збільшує зручність користування сайтом та знижує бар'єри для зв'язку з підтримкою.

3.4 Тестування застосунка

Функціональне тестування є одним з найважливіших етапів розробки програмного забезпечення, спрямованих на перевірку правильності роботи функцій системи відповідно до встановлених вимог. Його головна мета – забезпечити, щоб кожна функція програмного продукту працювала згідно з очікуваннями і відповідала вимогам специфікації. Це тестування охоплює різноманітні аспекти, включаючи перевірку користувацьких інтерфейсів, бізнес-логіки, інтеграції між компонентами системи та сценаріїв використання.

Одним з ключових аспектів функціонального тестування є перевірка функціональності програмного забезпечення. Це включає тестування всіх функцій, які програмний продукт повинен виконувати, від основних операцій до специфічних задач. Важливо переконатися, що всі функції працюють правильно, незалежно від умов їх використання [11].

Користувацький інтерфейс також є важливим аспектом функціонального тестування. Зручність і коректність роботи інтерфейсу безпосередньо впливають

на досвід користувачів. Функціональне тестування перевіряє, чи всі елементи інтерфейсу відображаються правильно і чи взаємодія з ними відповідає очікуванням користувача.

Тест-кейс 1: Успішна реєстрація

Попередні умови: Email користувача ще не зареєстрований.

Кроки:

1. Відкрити сторінку реєстрації.
2. Ввести коректні дані (email та пароль).
3. Натиснути кнопку «Зареєструватися».

Очікуваний результат: Користувач успішно реєструється, відкривається головна сторінка.

Тест-кейс 2: Реєстрація з вже існуючим email

Попередні умови: Email вже зареєстрований в системі.

Кроки:

1. Відкрити сторінку реєстрації.
2. Ввести дані (email, пароль).
3. Натиснути кнопку «Зареєструватися».

Очікуваний результат: Показується повідомлення про помилку: «Користувач з таким email вже існує».

Тест-кейс 3: Успішна авторизація

Попередні умови: Користувач зареєстрований в системі.

Кроки:

1. Відкрити сторінку авторизації.
2. Ввести правильні email та пароль.
3. Натиснути кнопку «Увійти».

Очікуваний результат: Користувач успішно входить в систему, відкривається головна сторінка.

Тест-кейс 4: Успішна оренда авто

Попередні умови: Користувач авторизований в системі.

Кроки:

1. Відкрити сторінку бронювання.
2. Вибрати автомобіль, місто початку та кінця оренди, дати початку та кінця оренди.
3. Натиснути кнопку «Пошук».
4. Заповнити детальну інформацію про себе та обрати точний час початку та кінця оренди.
5. Натиснути кнопку «Орендувати».
6. Підтвердити оренду у своїй пошті.

Очікуваний результат: Автомобіль успішно орендовано, користувач отримує підтвердження на пошту.

Тест-кейс 5: Бронювання авто без вибору всіх обов'язкових полів

Попередні умови: Користувач авторизований в системі.

Кроки:

1. Відкрити сторінку бронювання.
2. Вибрати тільки деякі обов'язкові поля (наприклад, тільки автомобіль та місто початку оренди).
3. Натиснути кнопку «Пошук».

Очікуваний результат: Показується повідомлення про необхідність заповнити всі обов'язкові поля.

Тест-кейс 6: Успішне залишення відгуку

Попередні умови: Користувач авторизований в системі.

Кроки:

1. Відкрити сторінку відгуків.
2. Заповнити поле для відгуку.
3. Натиснути кнопку «Відправити».

Очікуваний результат: Відгук успішно збережено і відображається в списку відгуків.

Таким чином, наведені тест-кейси забезпечують всебічне функціональне тестування основних функцій нашого застосунку для оренди автомобілів. Виконання цих тестів допоможе впевнитися у коректній роботі процесів реєстрації,

авторизації, перегляду автопарку, бронювання автомобілів та залишення відгуків. У разі виявлення будь-яких помилок або недоліків, вони будуть вчасно виправлені, що сприятиме підвищенню загальної якості та надійності системи, а також покращенню користувацького досвіду.

Автоматизоване тестування є невід’ємною частиною процесу забезпечення якості програмного забезпечення, оскільки воно дозволяє швидко та ефективно перевіряти функціональність системи на різних етапах розробки. Застосування автоматизованого тестування допомагає зменшити час, необхідний на тестування, підвищити точність перевірки та забезпечити повторюваність тестів. За допомогою автоматизованих тестів можна легко і швидко виявляти регресії та недоліки, що виникають після внесення змін до коду.

Для автоматизації тестування нашого застосунку для оренди автомобілів ми використали Selenium IDE. Selenium IDE є потужним інструментом для автоматизації веб-додатків, який дозволяє створювати та виконувати тест-кейси за допомогою простого графічного інтерфейсу. Цей інструмент надає можливість записувати дії користувача на веб-сторінці, зберігати їх у вигляді сценаріїв та повторно виконувати ці сценарії на різних версіях браузерів. Використання Selenium IDE значно спрощує процес створення та підтримки автоматизованих тестів, що, у свою чергу, сприяє підвищенню якості нашого застосунку [12].

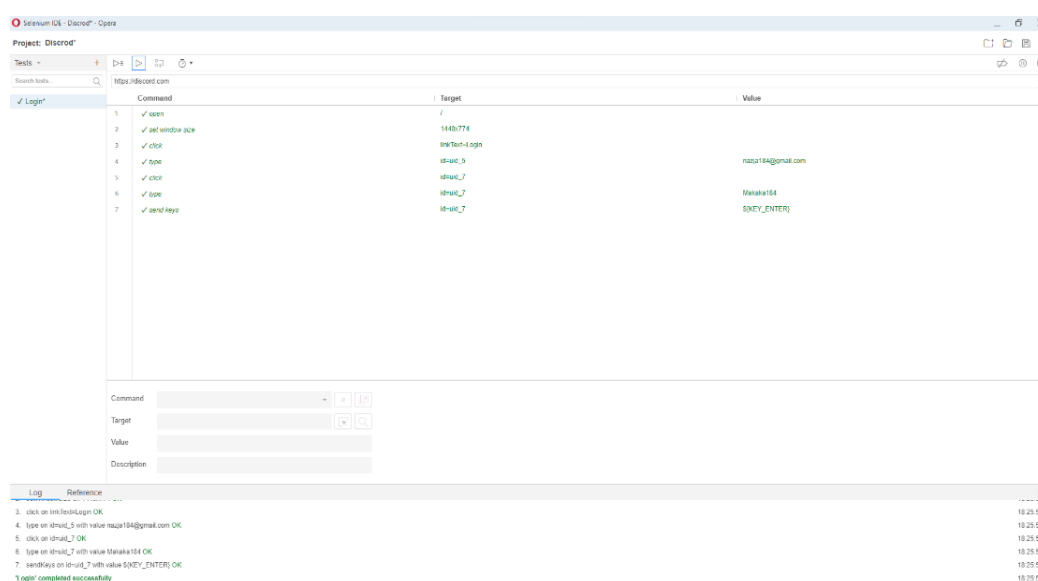


Рисунок 3.11 – Результат автоматизованого тестування

Усі функції нашого застосунку для оренди автомобілів пройшли ретельне тестування. Автоматизоване тестування за допомогою Selenium IDE підтвердили коректну роботу таких ключових можливостей, як реєстрація, авторизація, перегляд автопарку, бронювання автомобілів та залишення відгуків. Це свідчить про те, що наш застосунок відповідає встановленим вимогам якості, забезпечуючи стабільність та надійність роботи для користувачів.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Принципи, способи та засоби захисту населення.

Основні принципи щодо захисту населення:

- захист населення планується і здійснюється диференційовано, залежно від економічного та природного характеру його розселення, виду і ступеня небезпеки можливих надзвичайних ситуацій;
- усі заходи щодо життєзабезпечення населення готуються заздалегідь і здійснюються відповідно до законів держави;
- при захисті населення використовують усі наявні засоби захисту (захисні споруди, індивідуальні засоби захисту, евакуацію із небезпечних районів тощо);
- поінформованість громадян про свої основні обов'язки щодо безпеки життєдіяльності, дотримання ними установлених правил поведінки під час надзвичайних ситуацій.

Основні заходи щодо забезпечення захисту населення в надзвичайних ситуаціях:

- повідомлення населення про загрозу і виникнення надзвичайних ситуацій та постійне його інформування про наявні обставини;
- навчання населення застосуванню засобів індивідуального захисту і діям у надзвичайних ситуаціях;
- укриття людей у сховищах, медичний, радіаційний та хімічний захист, евакуація з небезпечних районів;
- спостереження та контроль за ураженістю навколишнього середовища, продуктів харчування та води радіоактивними, отруйними, сильнодіючими отруйними речовинами та біологічними препаратами;
- організація і проведення рятувальних та інших робіт у районах лиха й осередках ураження.

Засоби індивідуального захисту:

Засоби індивідуального захисту поділяють на засоби індивідуального захисту органів дихання та засоби індивідуального захисту шкіри.

За принципом захисної дії засоби індивідуального захисту органів дихання поділяють на фільтруючі та ізолюючі.

До засобів індивідуального захисту органів дихання фільтруючого типу відносять фільтруючі протигази, респіратори, ватно-марлеві пов'язки.

Фільтруючі протигази призначені для захисту органів дихання, очей та обличчя від отруйних і радіоактивних речовин та бактеріальних засобів.

Принцип дії фільтруючого протигаза ґрунтується на ізоляції органів дихання від забрудненого навколишнього середовища й очищення повітря, що вдихається, від токсичних аерозолів і парів у фільтруючо-поглинаючій системі.

Протигаз складається із шолом-маски та фільтруючо-поглинаючої коробки, з'єднаних між собою безпосередньо або за допомогою трубки.

Підготовка протигаза до використання починається з вибору необхідного розміру шолом-маски, який визначається за обсягом голови шляхом вимірювання її по лінії, що проходить через маківку, щоки та підборіддя.

Ізолюючі засоби індивідуального захисту органів дихання призначені для захисту органів дихання, обличчя та очей від шкідливих речовин у повітрі в умовах ізоляції органів дихання від навколишнього середовища [13].

До засобів захисту шкіри відносять:

- загальновійськовий захисний комплект (ЗЗК);
- легкий захисний костюм (Л-1);

4.2 Можливість виникнення статичної електрики та заходи боротьби з нею.

Статична електрика виникає внаслідок накопичення електричних зарядів на поверхні матеріалів, що може статися через тертя, контакт або відокремлення різних матеріалів. Зазвичай це спостерігається у виробничих процесах, де

взаємодіють різні типи матеріалів, таких як пластмаси, текстиль, папір та інші. В умовах низької вологості повітря електричні заряди не можуть легко розсіюватися, що призводить до їх накопичення на поверхнях. Це явище може бути небезпечним, особливо у середовищах з легкозаймистими матеріалами або вибухонебезпечними речовинами.

У промислових умовах, таких як виробництво, зберігання і транспортування хімічних речовин, нафтопродуктів та інших легкозаймистих матеріалів, виникнення статичної електрики може створювати значні ризики. Навіть невеликий іскровий розряд може стати причиною пожежі або вибуху, що загрожує життю і здоров'ю працівників, а також може призвести до значних матеріальних збитків. Наприклад, у нафтохімічній промисловості, де зберігаються та обробляються великі обсяги легкозаймистих рідин, контроль за статичною електрикою є критично важливим для безпеки.

Крім того, статична електрика може впливати на роботу електронних пристроїв та обладнання. Накопичення зарядів на поверхні електронних компонентів може призвести до їх пошкодження або виходу з ладу, що впливає на стабільність і надійність технологічних процесів. Високочутливі електронні компоненти можуть бути пошкоджені навіть малими розрядами статичної електрики, що може призвести до значних фінансових втрат через необхідність ремонту або заміни обладнання.

Для зниження ризиків, пов'язаних зі статичною електрикою, застосовуються різноманітні заходи. Основними методами боротьби з нею є:

Підвищення вологості повітря. В умовах підвищеної вологості електричні заряди легше розсіюються, що знижує ймовірність накопичення статичної електрики. Використання зволожувачів повітря або контролю рівня вологості в приміщеннях є ефективним заходом.

Заземлення обладнання та робочих місць. Заземлення металевих конструкцій, обладнання та робочих місць дозволяє забезпечити відведення статичних зарядів у землю, що запобігає їх накопиченню. Використання

антистатичних підлогових покриттів і заземлених робочих столів є поширеною практикою.

Використання антистатичних матеріалів. Матеріали з антистатичними властивостями, такі як спеціальні тканини, пластики та інші полімери, мають здатність запобігати накопиченню статичних зарядів. Використання таких матеріалів для виготовлення одягу, пакування та інших виробів допомагає знизити ризик утворення статичної електрики.

Іонізація повітря. Використання іонізаторів повітря дозволяє створювати умови, за яких електричні заряди нейтралізуються за допомогою позитивних і негативних іонів, що генеруються пристроями. Це особливо ефективно в середовищах, де важливо підтримувати низький рівень електричних зарядів.

Регулярне обслуговування і контроль. Проведення регулярного технічного обслуговування обладнання, перевірка стану заземлення та інших антистатичних заходів дозволяє своєчасно виявляти та усувати потенційні джерела статичної електрики. Важливо також навчати персонал правилам роботи в умовах підвищеного ризику виникнення статичної електрики та забезпечувати їх відповідним захисним обладнанням.

Використання антистатичних засобів індивідуального захисту. Одяг та взуття з антистатичними властивостями, а також спеціальні антистатичні браслети та інші засоби індивідуального захисту допомагають зменшити ризик накопичення зарядів на тілі працівника [14].

ВИСНОВОК

Дипломна робота присвячена розробці веб-застосунку для оренди автомобілів. Проведене дослідження, розробка та тестування дозволили створити функціональний, зручний та надійний інструмент для оренди автомобілів, що відповідає сучасним вимогам ринку.

У першому розділі роботи був проведений детальний аналіз предметної області. Спочатку було розглянуто загальну інформацію про ринок оренди авто, його розвиток та значення в сучасному світі. Далі було проведено аналіз переваг та недоліків основних конкурентів у цій сфері. Цей аналіз дозволив виявити сильні та слабкі сторони існуючих сервісів, що стало основою для визначення ключових вимог до нашого застосунку. Після цього була сформульована постановка задачі, яка включала визначення основних цілей і вимог до функціональності та зручності використання майбутнього продукту.

У другому розділі було зосереджено увагу на проектуванні застосунку. Спочатку були визначені функціональні та нефункціональні вимоги до системи, які включали основні функції для користувачів. Далі був описаний функціонал системи та створені діаграми прецедентів, які ілюструють взаємодію користувачів із системою. Також були розроблені сценарії для ключових функцій застосунку, що допомогло зрозуміти можливі варіанти використання системи. В кінці цього розділу були представлені діаграми послідовності, які показують покрокову взаємодію між різними компонентами системи під час виконання основних функцій.

Третій розділ роботи був присвячений безпосередньо розробці застосунку. Спочатку було обрано технології, що використовувалися для розробки, описано переваги цих технологій та причини їх вибору. Після цього було детально розглянуто процес створення компонентів застосунку. В кінці розділу було описано процес тестування застосунку, яке включало як функціональне, так і

автоматизоване тестування. Результати тестування підтвердили, що всі основні функції застосунку працюють коректно і відповідають вимогам якості.

На основі проведених досліджень та розробки можна зробити висновок, що створений застосунок для оренди авто є конкурентоспроможним продуктом, який забезпечує зручний і надійний сервіс для користувачів. Розроблений інтерфейс і функціонал дозволяють користувачам легко здійснювати всі необхідні дії, від перегляду автопарку до завершення бронювання. Впровадження сучасних технологій та ретельне тестування гарантують високу якість і стабільність роботи застосунку. У майбутньому можливе розширення функціональності та інтеграція додаткових сервісів для подальшого підвищення зручності та ефективності використання нашого продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Carsharing [Електронний ресурс] – Режим доступу до ресурсу: [https://carsharing.org/what-is-car-sharing/].
2. Функціональні вимоги [Електронний ресурс] – Режим доступу до ресурсу: [https://visuresolutions.com/uk/blog/functional-requirements/].
3. Нефункціональні вимоги [Електронний ресурс] – Режим доступу до ресурсу: [https://visuresolutions.com/uk/blog/non-functional-requirements/].
4. Діаграма прецедентів [Електронний ресурс] – Режим доступу до ресурсу: [https://evergreens.com.ua/ua/articles/uml-diagrams.html].
5. Діаграма послідовності [Електронний ресурс] – Режим доступу до ресурсу: [https://www.maxzosim.com/sequence-diagrams/].
6. Довідник HTML [Електронний ресурс] – Режим доступу до ресурсу: [https://css.in.ua/html/tags].
7. Довідник CSS [Електронний ресурс] – Режим доступу до ресурсу: [https://css.in.ua/].
8. Посібник JavaScript [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.javascript.info/].
9. Посібник React [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.legacy.reactjs.org/tutorial/tutorial.html].
10. VS Code [Електронний ресурс] – Режим доступу до ресурсу: [https://code.visualstudio.com/docs].
11. Функціональне тестування [Електронний ресурс] – Режим доступу до ресурсу: [https://qalight.ua/baza-znaniy/funktsionalne-testuvannya/].
12. Selenium IDE [Електронний ресурс] – Режим доступу до ресурсу: [https://www.selenium.dev/selenium-ide/].
13. Лапін В. М. Безпека життєдіяльності людини. Навч. Посіб. — 6-те вид., перероб. І доп. — К.:Знання, 2013, — 332 с.
14. Запорожець О. І. Основи охорони праці. Підручник. — 2-те вид. — К.: Центр учбової літератури, 2019. — 264 с.

ДОДАТКИ

ЗАВЕРШЕННЯ БРОНЮВАННЯ АВТО

Заповнивши цей запит на бронювання, ви отримаєте:

Ваш талон на оренду, який потрібно пред'явити після прибуття в пункт прокату і безкоштовний номер служби підтримки клієнтів.

Місце & Дата

Авто - VW Golf 6

Дата & Час початку оренди

2024-06-01 / 08:28

Дата & Час закінчення оренди

2024-06-23 / 23:00

Місце початку

Тернопіль

Місце закінчення

Київ



ІНФОРМАЦІЯ ПРО КЛІЄНТА

Ім'я *

Введіть ім'я

Обов'язкове поле.

Прізвище *

Введіть прізвище

Обов'язкове поле.

Номер телефону *

Введіть номер телефону

Обов'язкове поле.

Вік *

18

Обов'язкове поле.

Адреса *

Введіть свою адресу

Обов'язкове поле.

Місто *

Введіть місто проживання

Обов'язкове поле.

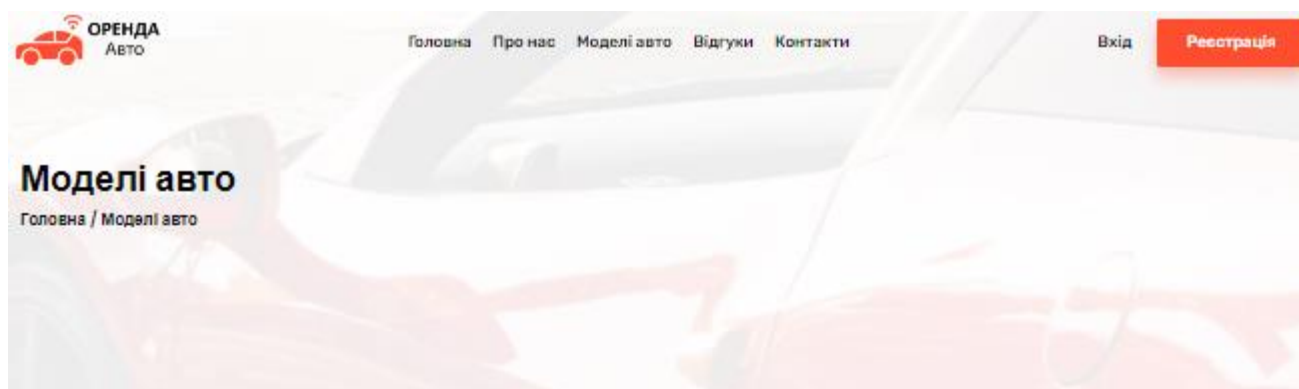
Поштовий індекс *

Введіть поштовий індекс

Обов'язкове поле.

☐ Я погоджуюсь на отримання останніх новин

Орендувати



 Audi ☆☆☆☆ 1600 € за добу ⚙ A1 4/5 🚗 ⚙ Ручна Бензин 🚗 Забронювати	 VW ☆☆☆☆ 1100 € за добу ⚙ Golf 6 4/5 🚗 ⚙ Ручна Дизель 🚗 Забронювати	 Toyota ☆☆☆☆ 900 € за добу ⚙ Camry 4/5 🚗 ⚙ Автомат Гібрид 🚗 Забронювати
 BMW ☆☆☆☆ 1200 € за добу ⚙ 320 4/5 🚗 ⚙ Автомат Дизель 🚗 Забронювати	 Mercedes ☆☆☆☆ 1800 € за добу ⚙ GLK 4/5 🚗 ⚙ Автомат Дизель 🚗 Забронювати	 VW ☆☆☆☆ 800 € за добу ⚙ Passat 4/5 🚗 ⚙ Автомат Бензин 🚗 Забронювати

Додаток В – Лістинг основних функцій

```

import { useState } from "react";

function CarBox({ data, carID }) {
  const [carLoad, setCarLoad] = useState(true);
  return (
    <>
      {data[carID].map((car, id) => (
        <div key={id} className="box-cars">
          {/* car */}
          <div className="pick-car">
            {carLoad && <span className="loader"></span>}
            <img
              style={{ display: carLoad ? "none" : "block" }}
              src={car.img}
              alt="car_img"
              onLoad={() => setCarLoad(false)}
            />
          </div>
          {/* description */}
          <div className="pick-description">
            <div className="pick-description__price">
              <span>€{car.price}</span>/ за добу
            </div>
            <div className="pick-description__table">
              <div className="pick-description__table__col">
                <span>Мапка</span>
                <span>{car.model}</span>
              </div>

              <div className="pick-description__table__col">
                <span>Модель</span>
                <span>{car.mark}</span>
              </div>

              <div className="pick-description__table__col">
                <span>Pik</span>
              </div>
            </div>
          </div>
        </div>
      ))}
    </>
  );
}

```



```

        <span>{car.year}</span>
    </div>

    <div className="pick-description__table__col">
        <span>К-сть дверей</span>
        <span>{car.doors}</span>
    </div>

    <div className="pick-description__table__col">
        <span>Трансмiсія</span>
        <span>{car.transmission}</span>
    </div>

    <div className="pick-description__table__col">
        <span>Паливо</span>
        <span>{car.fuel}</span>
    </div>
</div>
{ /* btn cта */}
<a className="cta-btn" href="#booking-section">
    Забронювати зараз
</a>
</div>
</div>
    )})
</>
);
}

export default CarBox;

```

```

import { Link } from "react-router-dom";
import BgShape from "../images/hero/hero-bg.png";
import HeroCar from "../images/hero/main-car.png";
import { useEffect, useState } from "react";
import { IconChevronRight, IconCircleCheck } from "@tabler/icons-react";

function Hero() {
  const [goUp, setGoUp] = useState(false);

  const scrollToTop = () => {
    window.scrollTo({ top: (0, 0), behavior: "smooth" });
  };

  const bookBtn = () => {
    document
      .querySelector("#booking-section")
      .scrollIntoView({ behavior: "smooth" });
  };

  useEffect(() => {
    const onPageScroll = () => {
      if (window.pageYOffset > 600) {
        setGoUp(true);
      } else {
        setGoUp(false);
      }
    };
    window.addEventListener("scroll", onPageScroll);

    return () => {
      window.removeEventListener("scroll", onPageScroll);
    };
  }, []);

  return (
    <>
      <section id="home" className="hero-section">
        <div className="container">
          <img className="bg-shape" src={BgShape} alt="bg-shape" />

```

```

<div className="hero-content">
  <div className="hero-content__text">
    <h4>Сплануйте вашу поїздку зараз</h4>
    <h1>
      Заощаджуйте з орендою авто
    </h1>
    <p>
      Орендуйте автомобіль своєї мрії. Неперевершені ціни,
необмежена кількість кілометрів,
      гнучкі варіанти оренди та багато іншого.
    </p>
    <div className="hero-content__text__btns">
      <Link
        onClick={bookBtn}
        className="hero-content__text__btns__book-ride"
        to="/"
      >
        Орендувати авто &nbsp; <IconCircleCheck />
      </Link>
      <Link
        className="hero-content__text__btns__learn-more"
        to="/"
      >
        Дізнатися більше &nbsp; <IconChevronRight />
      </Link>
    </div>
  </div>

  { /* img */ }
  <img
    src={HeroCar}
    alt="car-img"
    className="hero-content__car-img"
  />
</div>

{ /* page up */ }
<div
  onClick={scrollToTop}

```

```

        className={`scroll-up ${goUp ? "show-scroll" : ""}`}
      >
        ^
      </div>
    </section>
  </>
);
}
export default Hero;

import { Link } from "react-router-dom";
import Logo from "../images/logo/logo.png";
import { useState } from "react";
import { IconMenu2, IconX } from "@tabler/icons-react";

function Navbar() {
  const [nav, setNav] = useState(false);

  const openNav = () => {
    setNav(!nav);
  };

  return (
    <>
      <nav>
        {/* mobile */}
        <div className={`mobile-navbar ${nav ? "open-nav" : ""}`}>
          <div onClick={openNav} className="mobile-navbar__close">
            <IconX width={30} height={30} />
          </div>
          <ul className="mobile-navbar__links">
            <li>
              <Link onClick={openNav} to="/">
                Головна
              </Link>
            </li>
            <li>
              <Link onClick={openNav} to="/about">
                Про нас

```

```

        </Link>
    </li>
    <li>
        <Link onClick={openNav} to="/models">
            Моделі авто
        </Link>
    </li>
    <li>
        <Link onClick={openNav} to="/testimonials">
            Відгуки
        </Link>
    </li>
    <li>
        <Link onClick={openNav} to="/contact">
            Контакти
        </Link>
    </li>
</ul>
</div>
{ /* desktop */}
<div className="navbar">
    <div className="navbar__img">
        <Link to="/" onClick={() => window.scrollTo(0, 0)}>
            <img src={Logo} alt="logo-img" />
        </Link>
    </div>
    <ul className="navbar__links">
        <li>
            <Link className="home-link" to="/">
                Головна
            </Link>
        </li>
        <li>
            {" "}
            <Link className="about-link" to="/about">
                Про нас
            </Link>
        </li>
    </ul>

```

```

    <li>
      {" "}
      <Link className="models-link" to="/models">
        Моделі авто
      </Link>
    </li>
    <li>
      {" "}
      <Link className="testi-link" to="/testimonials">
        Відгуки
      </Link>
    </li>
    <li>
      {" "}
      <Link className="contact-link" to="/contact">
        Контакти
      </Link>
    </li>
  </ul>
  <div className="navbar__buttons">
    <Link className="navbar__buttons__sign-in" to="/">
      Вхід
    </Link>
    <Link className="navbar__buttons__register" to="/">
      Реєстрація
    </Link>
  </div>
  { /* mobile */ }
  <div className="mobile-hamb" onClick={openNav}>
    <IconMenu2 width={30} height={30} />
  </div>
</div>
</nav>
</>
);
}

export default Navbar;

```

