

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:

«Вебсайт з оренди автомобілів»

Виконала студентка групи ЗПЗ 21-1
ЛЕВЧЕНКО Мілана Сергіївна

Керівник роботи:

ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:

ЧИЖМОТРИЯ Олена Геннадіївна

Житомир – 2025

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

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛЕВЧЕНКО Мілана Сергіївна**

Керівник роботи: **ЛОБАНЧИКОВА Надія Миколаївна**

Тема роботи: **«Вебсайт для оренди автомобілів»**, затверджена Наказом закладу вищої освіти від «14» лютого 2025 р., №62/с

Вихідні дані для роботи: процес та технології розробки веборієнтованих систем, моделі, методи та засоби розробки інтерфейсної та функціональної частин вебсайту для оренди автомобілів із використанням технологій HTML, CSS та JavaScript.

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Лобанчикова Н.М.	27.05.2025р.	27.05.2025р.
2	Лобанчикова Н.М.	27.05.2025р.	27.05.2025р.
3	Лобанчикова Н.М.	27.05.2025р.	27.05.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсайту та пояснювальної записки. Пояснювальна записка до випускної роботи містить 67 сторінок, 20 ілюстрацій, 2 таблиці та додатки, список використаних джерел містить 33 найменування.

Метою роботи є розробка вебсайту для оренди автомобілів, що забезпечує користувачам зручний доступ до функцій пошуку, фільтрації, бронювання та перегляду доступних транспортних засобів. Об'єктом дослідження є процес та технології розробки веборієнтованих систем. Предметом дослідження є моделі, методи та засоби розробки інтерфейсної та функціональної частин вебсайту для оренди автомобілів із використанням технологій HTML, CSS та JavaScript. Методами дослідження обрано системний аналіз, методи об'єктно-орієнтованого проєктування, а також моделювання за допомогою нотацій UML.

У результаті виконання проєкту розроблено багатофункціональний вебресурс, що включає механізми пошуку та бронювання авто, панель адміністратора, систему авторизації та адаптивний користувацький інтерфейс. Вебсайт має перспективи для подальшого масштабування та впровадження в реальних умовах.

КЛЮЧОВІ СЛОВА: ВЕБСАЙТ, ІНФОРМАЦІЙНА СИСТЕМА, ОРЕНДА АВТОМОБІЛІВ, CSS, HTML, JAVASCRIPT.

					ІПЗ.КР.Б.-121-25-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Левченко М.С.			Вебсайт з оренди автомобілів Пояснювальна записка	Літ.	Арк.	Акрушів
Перевір.		Лобанчикова Н.М.					3	67
Рецензент.		Чижмотря О.Г.						
Зав.заф.		Вакалюк Т.А.				Житомирська політехніка, ФІКТ, група ЗІПЗ-21-1		

ABSTRACT

The bachelor's thesis consists of a website and an explanatory report. The explanatory report contains 67 pages, 20 illustrations, 2 tables, appendices. the list of sources used contains 33 names.

The purpose of the work is to develop a website for car rental, which provides users with convenient access to the functions of searching, filtering, booking and viewing available vehicles. The object of the research is the process and technologies for developing web-based systems. The subject of the research is models, methods and tools for developing the interface and functional parts of a website for car rental using HTML, CSS and JavaScript technologies. The research methods selected are system analysis, object-oriented design methods, as well as modeling using UML notations.

As a result of the project, a multifunctional web resource was developed, which includes car search and booking mechanisms, an administrator panel, an authorization system and an adaptive user interface. The website has prospects for further scaling and implementation in real conditions.

KEYWORDS: WEBSITE, INFORMATION SYSTEM, CAR RENTAL, CSS, HTML, JAVASCRIPT.

		Левченко М.С.			ІІЗ.КР.Б.-121-25-ІІЗ	Арк.
		Лобанчикова Н.М.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ РІШЕНЬ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБСАЙТІВ .	9
1.1 Аналіз рішень та технологій розробки вебсайтів	9
1.2 Аналіз аналогів сайтів надання транспортних послуг	16
1.3 Обґрунтування вибору технологій для реалізації вебсайту	22
1.4 Технічне завдання на розробку вебсайту з оренди автомобілів	26
Висновки до першого розділу	28
РОЗДІЛ 2.МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБСАЙТУ З ОРЕНДИ АВТОМОБІЛІВ	30
2.1 Розробка UML-моделей вебсайту з оренди автомобілів	30
2.2 Розробка функціональних алгоритмів роботи вебсайтів	35
2.3 Обґрунтування архітектури та розробка модулів	38
Висновки до другого розділу	42
РОЗДІЛ 3. ПЛАН ВИКОНАННЯ ПРОЄКТУ ВЕБСАЙТУ	44
3.1 Інтерфейс програмного продукту	44
3.2 Особливості розгортання програмного продукту	52
Висновки до третього розділу	57
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТКИ.....	64

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface – інтерфейс прикладного програмування

CI/CD – Continuous Integration / Continuous Delivery – безперервна

CSS – Cascade Style Sheets

DB – Database – база даних

FTP – File Transfer Protocol – протокол передавання файлів

HTML – HyperText Markup Language

JS –JavaScript – мова програмування для інтерактивності

інтеграція / доставлення

PWA –Progressive Web Application – прогресивний вебдодаток

SQL –Structured Query Language – мова структурованих запитів

SSL – Secure Sockets Layer – технологія шифрування з'єднання

UI – User Interface – користувацький інтерфейс

UX – User Experience – досвід користувача

URL – Uniform Resource Locator – уніфікований вказівник ресурсу
програмування

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми полягає у тому, що у сучасному світі стрімкий розвиток інформаційних технологій, мобільних додатків та цифрових сервісів призвів до кардинальної трансформації процесів взаємодії користувачів із бізнесом. Однією з найбільш динамічно зростаючих галузей є сфера оренди транспортних засобів, де впровадження інноваційних ІТ-рішень значно підвищує якість та швидкість обслуговування клієнтів. У зв'язку з урбанізацією, мобільністю населення та активним використанням інтернету, користувачі очікують отримання сервісів оренди авто в декілька кліків без потреби особистого візиту чи складних телефонних процедур. Сучасні платформи оренди авто демонструють, що цифровізація цієї послуги не лише зручність, а й необхідність для ефективної взаємодії бізнесу з клієнтом. Розробка вебсайту для оренди автомобілів дозволяє задовольнити потребу у швидкому доступі до актуальної інформації про вільний транспорт, його характеристики, ціни, а також надає зручні інструменти бронювання. Тому розробка зручного, функціонального та адаптивного вебсайту оренди автомобілів є актуальною темою, яка поєднує в собі завдання з проєктування, розробки та впровадження сучасного інтерфейсу з урахуванням безпеки, масштабованості та UX-орієнтованого дизайну.

Метою цієї кваліфікаційної роботи є розробка вебсайту для оренди автомобілів із інтерфейсом, що дозволяє користувачам переглядати наявні автомобілі, здійснювати фільтрацію за параметрами, бронювати транспортні засоби та отримувати підтвердження замовлення онлайн.

Завдання роботи:

- провести аналіз методів та технологій створення вебсайтів;
- провести аналіз аналогів сайтів надання транспортних послуг;
- розробити моделі та функціональні алгоритми роботи вебсайту з оренди автомобілів;
- здійснити програмну реалізацію вебсайту з оренди автомобілів;

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

- провести тестування функціоналу;
- розробити рекомендації щодо впровадження та масштабування платформи.

Об’єктом дослідження є процес та технології розробки веборієнтованих систем.

Предметом дослідження є моделі, методи та засоби розробки інтерфейсної та функціональної частин вебсайту для оренди автомобілів із використанням технологій HTML, CSS та JavaScript.

Методами дослідження є системний аналіз, об’єктно-орієнтоване моделювання, методологія моделювання UML.

Структура та обсяг роботи. Випускна кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел із 13 найменувань, 1 додатка, 2 таблиці та 20 ілюстрацій. Повний обсяг роботи становить 67 сторінок, з них 60 сторінок основного тексту.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ РІШЕНЬ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБСАЙТІВ

1.1 Аналіз рішень та технологій розробки вебсайтів

Веб-розробка є однією з найдинамічніших галузей сучасних інформаційних технологій, яка постійно розвивається та змінюється відповідно до нових ринкових вимог, технічного прогресу та очікувань користувачів. У 2025 році, коли цифрова присутність стала обов'язковою умовою успішного функціонування будь-якого бізнесу, знання й практичне застосування основ веб-розробки набули особливої актуальності [1,2]. Цифрові платформи не лише забезпечують зручність комунікації з клієнтами, але й дозволяють оптимізувати бізнес-процеси, автоматизувати облік, продаж, бронювання, зворотний зв'язок тощо. Попри стрімку еволюцію технологій, появу нових інструментів та мов програмування, саме базові засоби веб-розробки – HTML, CSS і JavaScript – залишаються фундаментом для створення вебдодатків різної складності. На їх основі побудовано потужні сучасні фреймворки, такі як React.js, Vue.js, Angular, а також серверне середовище Node.js, що дозволяє використовувати JavaScript на стороні сервера [3–5]. Ці технології широко застосовуються при створенні інтерактивних платформ, зокрема сайтів для оренди автомобілів, де важливо забезпечити одночасно привабливий дизайн, зручний інтерфейс, швидкий відгук системи та інтеграцію з базами даних і платіжними сервісами.

HTML (HyperText Markup Language) є мовою розмітки, яка визначає логічну структуру вебсторінки. Саме за допомогою HTML задається розташування основних елементів – текстів, зображень, гіперпосилань, таблиць, форм введення, блоків мультимедіа тощо. Сучасна версія HTML5 запровадила семантичні теги, що значно покращили якість кодування, забезпечуючи краще структурування вмісту. Такі елементи, як <header>, <nav>, <main>, <section>, <article>, <footer> дозволяють логічно розподіляти контент вебсторінки, полегшуючи її сприйняття як для браузера, так і для пошукових систем [6,7]. Для прикладу, при реалізації каталогу автомобілів на сайті доцільно використовувати

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

структуру у вигляді `<section class="cars-list">`, яка містить серію блоків `<article class="car-card">`, кожен з яких представляє окрему машину. Такий підхід підвищує не лише організованість коду, а й сприяє покращенню SEO-оптимізації, доступності сайту для користувачів з інвалідністю та коректній індексації пошуковими системами.

CSS (Cascading Style Sheets) є мовою стилів, що дозволяє задавати зовнішній вигляд елементів HTML-документа. Завдяки CSS можливо реалізовувати стилізацію текстів, змінювати кольори, шрифти, відступи, позиціонування, а також створювати адаптивну верстку, яка автоматично підлаштовується під різні типи пристроїв – від комп'ютерів до мобільних телефонів. CSS дозволяє повністю відокремити структуру документа від його презентаційного оформлення, що забезпечує гнучкість розробки та полегшує подальше супроводження коду [8–10].

Сучасні можливості CSS3 суттєво розширили інструментарій вебдизайнера. Наприклад, підтримка анімацій (використання `@keyframes`, `transition`, `transform`), гнучких макетів (Flexbox, CSS Grid), змінних (`--variable`) та медіа-запитів дозволяє реалізувати інтерфейси будь-якої складності без залучення сторонніх рішень [11]. У контексті сайту для оренди автомобілів за допомогою CSS можна створити адаптивну сітку для відображення карток автомобілів, а також застосувати візуальні ефекти – такі як плавне з'явлення кнопок, наведення миші, зміна фону під час взаємодії – що покращує користувацький досвід.

JavaScript є мовою програмування, яка забезпечує динамічну поведінку вебсторінки, дозволяє реалізовувати складну логіку, а також взаємодіяти з сервером і зовнішніми API в режимі реального часу. Якщо HTML відповідає за структуру документа, а CSS – за його вигляд, то JavaScript керує поведінкою елементів на сторінці. Завдяки JavaScript реалізується валідація форм, обробка подій, таких як натискання кнопок чи зміна значень, динамічне оновлення вмісту сторінки без перезавантаження (AJAX, Fetch API), а також інтеграція з базами

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

даних, платіжними системами, картографічними сервісами тощо [3,4,11].

Наприклад, на сайті з оренди авто JavaScript дозволяє реалізувати фільтрацію автомобілів за параметрами (модель, вартість, тип кузова), здійснювати пошук у реальному часі, динамічно формувати замовлення, перевіряти правильність введених даних, а також відображати спливаючі повідомлення або інтерактивні підказки. Завдяки асинхронному виконанню функцій користувач не змушений чекати на повну перезавантаження сторінки – усі дії виконуються швидко, що значно підвищує зручність користування сайтом. Таким чином, HTML, CSS та

JavaScript є базовими технологіями сучасної веб-розробки, які забезпечують не лише функціональність, але й якісний користувацький досвід. Їх використання є основою для реалізації вебсайтів будь-якої складності, зокрема таких, що стосуються оренди автомобілів, де потрібна гнучкість, інтерактивність та адаптивність інтерфейсу до потреб різних категорій користувачів.

Використання HTML, CSS і JavaScript має низку переваг, які зумовлюють їхню широку популярність у розробці сучасних веб-сайтів, зокрема сфері оренди автомобілів. Однією з ключових переваг є гнучкість і масштабованість. CSS забезпечує централізоване управління стилями для всього сайту, що значно спрощує внесення змін і підтримку проєкту у довгостроковій перспективі. JavaScript дозволяє легко додавати нові інтерактивні елементи, такі як фільтри, сортування, калькулятори вартості оренди [12,9], не змінюючи базову структуру HTML-документів. Ще однією важливою перевагою є швидкість завантаження. Оптимізовані CSS- і JavaScript-файли зменшують обсяг коду та кількість запитів до сервера [14]. Стиснення та мінімізація цих файлів позитивно впливають на швидкість завантаження, що критично важливо для сайтів оренди автомобілів, де користувачі очікують швидкої реакції інтерфейсу. Також значну роль відіграє адаптивність. Завдяки сучасним можливостям CSS (медіа-запити, Flexbox, Grid) забезпечується коректне відображення інтерфейсу на будь-якому пристрої [15]. JavaScript дозволяє додатково підлаштовувати інтерфейс під потреби

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача, наприклад, змінювати вигляд каталогу автомобілів залежно від розміру екрана або впроваджувати мобільне меню.

Значна увага приділяється і інтерактивності вебзастосунку. JavaScript забезпечує динамічну взаємодію з користувачем: підказки, спливаючі вікна, автозаповнення форм, динамічне оновлення списку автомобілів без перезавантаження сторінки, інтеграцію карт, лічильники, слайдери тощо [5,14]. Крім того, вебтехнології забезпечують можливість розширення. Завдяки JavaScript сайт легко інтегрується з зовнішніми сервісами – платіжними системами, CRM, чат-ботами, аналітичними інструментами, що особливо важливо для сайтів оренди автомобілів.

Розвиток стандарту CSS3 значно розширив можливості верстки веб-інтерфейсів, пропонуючи засоби, які раніше вимагали використання JavaScript. Модулі Flexbox і Grid Layout забезпечують реалізацію гнучких та складних макетів без зовнішніх фреймворків. CSS3 підтримує анімаційні ефекти через властивості transition та директиву @keyframes, що дозволяє реалізовувати плавні візуальні зміни без залучення скриптів. Водночас JavaScript залишається незамінним для реалізації складної логіки, обробки подій, асинхронної взаємодії із сервером (AJAX, Fetch API), створення SPA (Single Page Application) на основі фреймворків React.js, Vue.js. Наприклад, форма бронювання автомобіля може перевіряти коректність введених даних у реальному часі, відправляти заявку без перезавантаження сторінки та одразу показувати повідомлення про успішне бронювання. Для підвищення ефективності розробки та підтримки проєктів великого масштабу застосовуються методології структуризації стилів (наприклад, BEM – Block–Element–Modifier) та CSS-препроцесори (SCSS, LESS) [8], які розширюють стандартні можливості CSS. JavaScript-код також організовується за допомогою модульних підходів (ES6 modules), що підвищує читабельність і масштабованість.

Серед сучасних тенденцій у веб-розробці слід відзначити такі напрями, як використання SSR (Server-Side Rendering) для покращення швидкості

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

початкового завантаження сторінки та SEO, впровадження Jamstack-архітектури для поєднання статичної генерації сторінок із динамічними функціями через API, а також застосування безсерверних функцій (serverless) для обробки форм, інтеграції з зовнішніми сервісами та відправлення повідомлень про замовлення.

У 2025 році простежується тенденція до поєднання перевірених підходів із сучасними технологіями, що дозволяє досягати високих результатів у створенні вебплатформ нового покоління. Зростаючий попит на швидкі, зручні, безпечні та масштабовані вебресурси зумовлює інтеграцію традиційних методів із інноваційними рішеннями в межах загальної архітектури сайтів. Серед ключових напрямів розвитку вебтехнологій вирізняються кілька стратегічно важливих рішень, які формують сучасні практики веброзробки. Одним із таких рішень є відновлення інтересу до SSR (Server-Side Rendering), що дозволяє формувати HTML-контент на стороні сервера перед його передачею браузеру користувача. Це забезпечує пришвидшене завантаження початкового вмісту сторінки, зменшує час до взаємодії (Time to Interactive) та покращує SEO-індексацію, що має особливу вагу для проєктів у висококонкурентному середовищі, зокрема сайтів для оренди автомобілів, які мають на меті охопити широку аудиторію через пошукові системи та соціальні платформи. SSR особливо ефективний у поєднанні з сучасними фреймворками, такими як Next.js або Nuxt.js, які забезпечують гнучке налаштування рендерингу та оптимізацію ресурсів [1].

Паралельно з SSR набуває популярності Jamstack-підхід (JavaScript, APIs, Markup), який активно впроваджується як стратегія створення високопродуктивних, захищених і масштабованих вебрішень. Jamstack-архітектура базується на попередній генерації статичних сторінок з подальшою інтеграцією динамічних можливостей через API, що дозволяє ефективно відокремлювати фронтенд від бекенду. У випадку сайтів з оренди автомобілів це означає, що загальна структура сторінок може бути згенерована заздалегідь, а такі функції, як фільтрація авто, система онлайн-бронювання чи обробка замовлень, реалізуються через асинхронні запити до сторонніх сервісів або

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

власного API. Такий підхід значно покращує швидкодію сайту, підвищує рівень безпеки завдяки відсутності прямого доступу до серверної логіки та дозволяє легко масштабувати функціонал без суттєвого впливу на основну архітектуру. Завдяки широкій підтримці хмарних платформ (наприклад, Netlify, Vercel, Cloudflare Pages) розгортання Jamstack-проектів відбувається швидко й з мінімальними витратами. Ще одним перспективним напрямом є використання безсерверних обчислень (serverless computing), які дозволяють виконувати логіку на запит без потреби в постійному утриманні сервера. Безсерверні функції, які зазвичай реалізуються за допомогою сервісів, таких як AWS Lambda, Google Cloud Functions або Azure Functions, забезпечують обробку подій, що виникають у результаті дій користувача, зокрема заповнення форм, підтвердження бронювання, надсилання електронних листів або push-сповіщень, взаємодію з CRM або платіжними системами. Їх використання в контексті сайту з оренди авто дозволяє зменшити витрати на підтримку інфраструктури, підвищити гнучкість проекту та швидко інтегрувати нові функції, зберігаючи високу продуктивність та масштабованість. Ефективна реалізація сучасного вебсайту для оренди автомобілів на основі HTML, CSS та JavaScript вимагає дотримання ряду базових принципів, які гарантують як технічну якість рішення, так і комфорт для кінцевого користувача. Передусім необхідно забезпечити чітку структуру HTML-документа за допомогою семантичної розмітки. Використання елементів HTML5, таких як `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`, не лише полегшує сприйняття коду розробниками, а й підвищує доступність сайту, сприяє кращій індексації сторінок пошуковими системами та дозволяє використовувати програми читання з екрана для людей із порушенням зору [3,8].

Не менш важливою є реалізація адаптивного дизайну, що дозволяє інтерфейсу коректно відображатися на пристроях з різними розмірами екрана, включно з мобільними телефонами, планшетами та настільними комп'ютерами. Для цього доцільно застосовувати сучасні технології CSS, зокрема Flexbox і CSS

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Grid, які дозволяють створювати гнучкі та масштабовані макети, а також медіа-запити (media queries), що адаптують стилі залежно від параметрів пристрою користувача. Важливим аспектом є модульність стилів, яка передбачає розділення стилів на логічно обґрунтовані частини або компоненти, що спрощує підтримку, масштабування та повторне використання коду. Такий підхід забезпечується як через класичне структурування CSS-файлів, так і завдяки використанню CSS-препроцесорів (SASS, LESS) або CSS-модулів у фреймворках. Оптимізація продуктивності вебресурсу є ключовим завданням для забезпечення стабільної роботи в умовах великого трафіку та високих очікувань користувачів. Серед основних підходів до оптимізації можна виділити стиснення зображень, використання сучасних форматів (WebP, AVIF), мініфікацію CSS і JavaScript-файлів, застосування систем кешування, lazy-loading для зображень та зовнішніх компонентів, а також скорочення кількості HTTP-запитів [16]. Комплексна оптимізація дозволяє значно зменшити час завантаження сторінки, покращити показники Core Web Vitals та забезпечити високу продуктивність навіть за умов повільного з'єднання [3,5,8].

Окремої уваги заслуговує питання доступності (accessibility), що передбачає створення таких інтерфейсів, які є зручними для користувачів із фізичними або когнітивними обмеженнями. Це включає використання коректної розмітки для форм, альтернативних текстів для зображень, логічної структури заголовків, підтримку клавіатурної навігації, а також сумісність із програмами читання з екрана. Дотримання стандартів доступності (наприклад, WCAG 2.1) є не лише етичним зобов'язанням, а й вимогою для участі у міжнародних тендерах та грантах, а також фактором покращення загального юзабіліті сайту. Окрім базової функціональності, сучасний вебсайт з оренди автомобілів повинен мати можливість гнучкого розширення. Це передбачає впровадження таких компонентів, як інтерактивні форми бронювання з перевіркою даних, вбудовані фотогалереї з підтримкою перегляду зображень у повноекранному режимі, інтеграцію з Google Maps для вибору місця видачі/повернення авто, системи

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

онлайн-оплати з використанням платіжних шлюзів (Stripe, LiqPay, Wayforpay) та особистого кабінету користувача для перегляду історії замовлень і керування бронюванням. Усі ці елементи мають бути реалізовані з урахуванням сучасних вимог до безпеки, зокрема захисту даних користувачів, шифрування запитів, захисту від CSRF і XSS-атак.

У сукупності зазначені тенденції та технології формують комплексний підхід до створення конкурентоспроможного вебсайту оренди автомобілів, який поєднує високу продуктивність, гнучкість, масштабованість і зручність для користувача. Це дає змогу забезпечити ефективну взаємодію з цільовою аудиторією, підвищити рівень конверсії та забезпечити стабільний розвиток цифрової платформи.

1.2 Аналіз аналогів сайтів надання транспортних послуг

Розробка програмного забезпечення для оренди автомобілів є етапом у створенні сучасного цифрового продукту, що забезпечує комфортний інтерфейс, ефективну функціональність та відповідність реальним потребам користувачів. Водночас це не просто технічний процес реалізації ідеї – це стратегічний крок, який вимагає глибокого аналізу поточної ситуації на ринку, за рахунок уже наявних рішень і продуктів у цій галузі. Такий аналіз є невід’ємною складовою початкового етапу планування та моделювання майбутнього програмного забезпечення. Він дозволяє нам більш усвідомлено підходити до формування архітектурних систем, логіки її роботи, а також підбору технологій, що найкраще відповідають вимогам сучасного користувача. Завдяки аналізу ринку з’являються можливості більш розібратися у вже впроваджених цифрових продуктах, що надають послуги оренди автомобілів, та оцінити, як саме провідні компанії реалізують бізнес-ключ-процеси. Вивчення конкурентного середовища дозволяє не лише розмістити сили провідних гравців ринку, а й побачити їхні недоліки та слабкі місця, які можуть бути покращені або реалізовані інакше у власному програмному рішенні. вибір того, в процесі аналізу виявляються

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

актуальні тенденції розвитку індустрії, інноваційні функції, що покращують досвід використання (наприклад, інтелектуальний підбір авто, інтеграція з платіжними сервісами, підтримка електромобілів, адаптивний інтерфейс тощо), а також популярні підходи до дизайну, безпеки даних і систем масштабування. Ретельне дослідження конкурентів дозволяє врахувати реальні очікування цільової аудиторії, уникнути повторення типових помилок, адаптувати найкращі практики та, що містяться, створити унікальну ціннісну пропозицію – що таке програмне забезпечення, які мають конкурентні переваги і вдається успішно конкурувати з уже існуючими платформами.

Оцінка конкурентів дає можливість проаналізувати свої сильні та слабкі сторони, зосередитися на недоліках, які потребують покращення, та застосувати найкращі практики при створенні власного програмного продукту. Водночас важливі характеристики та конкурентні переваги майбутньої системи, які дозволяють їй успішно виділитися на фоні інших пропозицій і краще відповідати зростаючим потребам і очікуванням цільової аудиторії – зокрема сучасних користувачів, які цінують зручність, швидкість і персоналізований підхід. У межах дослідження ринку доцільно відзначити провідні глобальні компанії, які надають послуги оренди автомобілів, та проаналізувати їх ключові особливості, функціональні можливості програмного забезпечення, досвід взаємодії з користувачами, а також інфраструктурне обслуговування. Серед проаналізованих конкурентів виділяються такі компанії як Rentalcars [17], Sixt [18], 7cars [19], Hertz [20], Avis [21], Eurocar [22].

Однією з найвідоміших компаній у цій галузі є Hertz – світовий лідер у сфері прокату автомобілів. Компанія має багаторічний досвід понад, а її мережа включає 10 000 відділень у більш ніж 150 країнах світу [20]. Hertz пропонує широкий вибір транспортних засобів – від економ-класу до преміум-авто – що дозволяє охопити різні категорії клієнтів із ефективними фінансовими можливостями. без цього компанія активно впроваджує цифрові рішення, зокрема мобільні застосунки та онлайн-сервіси для бронювання автомобілів, що

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує доступність та зручність користування її послугами. Серед основних переваг Hertz варто відзначити високу швидкість даними між клієнтськими пристроями та серверами компанії: середній час відповіді становить лише 0,3 секунди, що показує якісну оптимізацію обміну внутрішньої інфраструктури. Також компанія пропонує спеціальні програми лояльності, знижки для корпоративних клієнтів і тривалу оренду, що робить її привабливою для бізнесу. Проте, як і в багатьох масштабних сервісах, у діяльності компанії спостерігаються певні недоліки. Частина клієнтів висловлює невдоволення якістю обслуговування, зокрема труднощами зі зворотним зв'язком та рішенням проблемних ситуацій. Крім того, мобільний додаток Hertz іноді критикують за надійність навігації та обмежену інформативність щодо автомобілів. Серед інших недоліків – поява автоматичних рекомендацій щодо вибору транспортного засобу відповідно до категорійного посвідчення, що може ускладнити процес бронювання для менш досвідчених користувачів. Варто також зазначити що у травні 2023 року компанія завершила процес фінансової реструктуризації та офіційно вийшла з процедури банкрутства, що відкриває нові можливості для її подальшого розвитку та інвестування в інноваційні рішення (рис. 1.1).

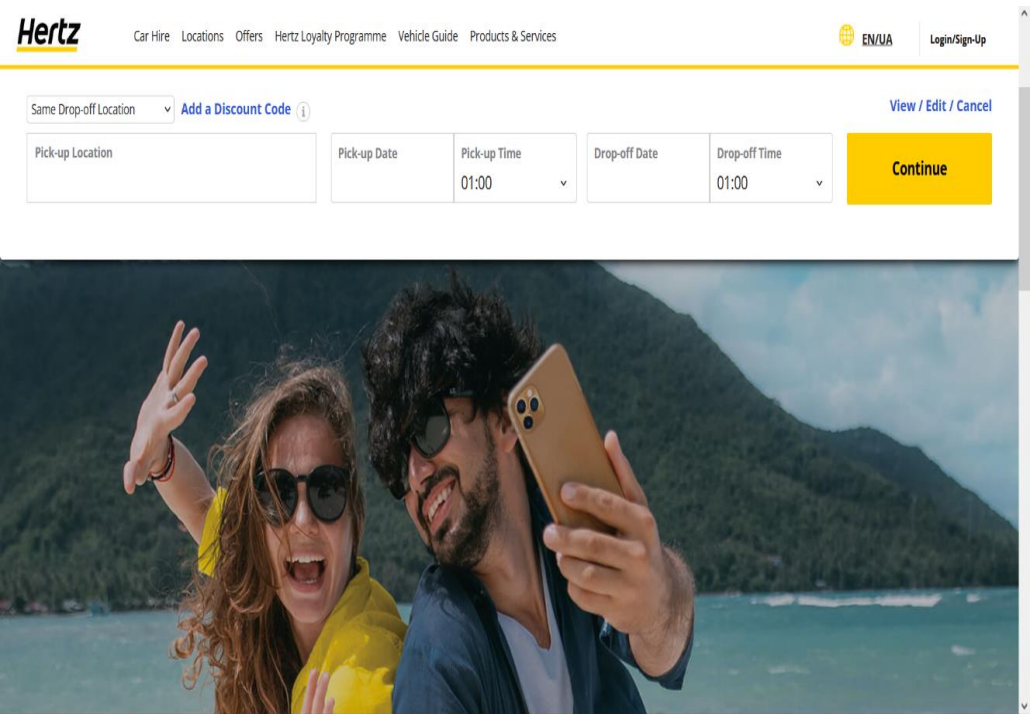


Рис. 1.1 Домашня сторінка Hertz

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Компанія Avis є міжнародним постачальником послуг прокату автомобілів, що працює більш ніж у 110 країнах [21]. Вона спеціалізується на оренді автомобілів преміумкласу та пропонує вигідні умови для корпоративних клієнтів, включаючи спеціальні програми лояльності. Серед переваг Avis – доступність у багатьох країнах світу та фокус на якісному обслуговуванні бізнес-клієнтів. Проте вартість послуг цієї компанії може бути вищою, ніж у її конкурентів. Як і в інших випадках, немає функціоналу з рекомендаціями автомобілів відповідно до категорії водійського посвідчення. Додатково, у першому кварталі 2023 року компанія продемонструвала зростання чистого прибутку більш ніж на 100% у порівнянні з аналогічним періодом 2022 року (рис. 1.2).

Рис. 1.2. Домашня сторінка Avis

Компанія Europcar є одним із ключових гравців на міжнародному ринку оренди автомобілів, маючи понад 6000 відділень у більш ніж 140 країнах світу [22]. Вона активно позиціонує себе як бренд, орієнтований на сталий розвиток і турботу про довкілля, зокрема завдяки широкій наявності в автопарку екологічно

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

чистих транспортних засобів, включаючи гібридні та повністю електричні моделі. Такий підхід відповідає сучасним глобальним трендам, спрямованим на скорочення викидів парникових газів і підтримку ініціатив «зеленої» мобільності.

Проте деякі технічні проблеми, такі як затримка відповіді сервера в середньому на 1 секунду, можуть знижувати зручність користування. Як і в інших сервісах, відсутня система рекомендацій за типом водійського посвідчення. У першому кварталі 2023 року компанія відзвітувала про зростання чистого прибутку більш ніж на 50% порівняно з відповідним періодом попереднього року (рис. 1.3).

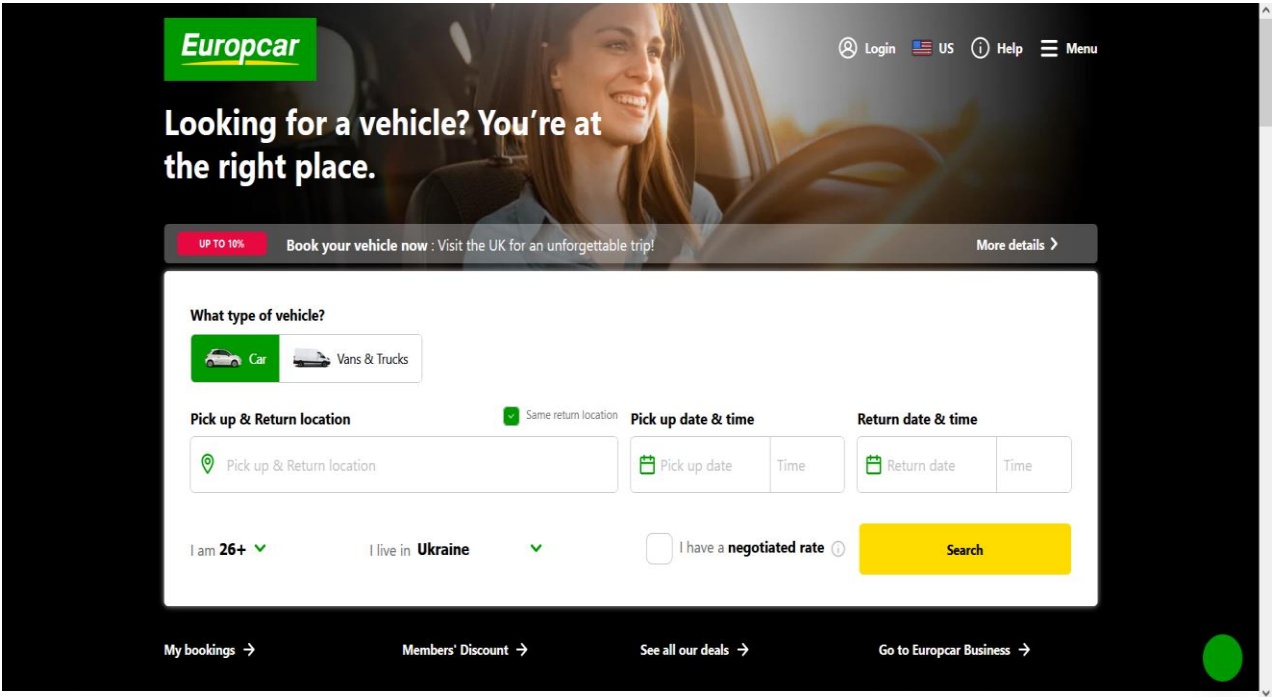


Рис. 1.3 Домашня сторінка Europcar

Порівняльний аналіз показує, що кожна з існуючих платформ має як сильні сторони, так і обмеження. Наш програмний продукт LuxuryCars вирізняється зручним інтерфейсом, розширеними функціями пошуку та фільтрації, а також високим рівнем захисту даних.

У таблиці 1.1 наведено порівняння основних характеристик програмних рішень: Hertz, Avis, Europcar та розробленої нами системи LuxuryCars. Для кожної функції відображено, чи підтримується вона на відповідній платформі.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Для кожної функції вказано параметри, які пропонує кожна платформа знак «+» або «-» (табл.1.1).

На основі проведеного порівняльного аналізу вебсайтів провідних компаній у сфері оренди автомобілів (Hertz, Avis, Europcar) можна зробити висновок, що розроблена система LuxuryCars має низку переваг, що забезпечують її конкурентоспроможність на ринку. Зокрема, платформа пропонує розширений асортимент транспортних засобів, включаючи преміум-, спортивні та комерційні авто, що перевищує пропозицію деяких конкурентів приблизно на 20%.

Таблиця 1.1

Таблиця порівняльний аналіз програмних аналогів

Характеристика	Hertz	Avis	Eurocar	LuxuryCars
Реєстрація користувачів та перевірка особистих даних	+	+	+	+
Вибір авто за категорією водійського посвідчення	—	—	—	планується
Розширені фільтри (тип палива, коробка тощо)	—	—	—	+
Адаптивність під мобільні пристрої	+	+	+	+
Підтримка онлайн-платежів	+	+	+	+
Підтримка української мови	—	—	—	+
Темна тема	—	—	—	планується
Чат-бот	—	—	—	планується

Інтерфейс системи реалізовано із застосуванням розширених фільтрів, які дозволяють користувачам швидко і зручно знаходити відповідні авто за численними критеріями. Однією з унікальних функцій є валідація водійських прав, що забезпечує безпечне надання доступу до автомобілів лише тим користувачам, які мають відповідну категорію посвідчення.

Окрім цього, архітектура системи LuxuryCars є модульною і масштабованою, що дозволяє у перспективі інтегрувати додаткові види транспорту (мотоцикли, човни) та розширити функціональність без значних змін у базовій структурі. Таким чином, платформа має потенціал до гнучкого розширення, надає персоналізований досвід користувача і демонструє високий рівень технологічної готовності до впровадження на ринку.

1.3 Обґрунтування вибору технологій для реалізації вебсайту

Вибір технологій для розробки вебсайту з оренди автомобілів є одним із ключових етапів проектування, оскільки саме від обраного технологічного стеку залежить функціональність, продуктивність, безпека, масштабованість, а також зручність підтримки та подальшого розвитку програмного продукту. Під час вибору технологій було проведено комплексний аналіз сучасних тенденцій веброзробки, вимог до користувацького досвіду, специфіки цільового ринку та завдань, які має вирішувати система.

Для побудови інтерфейсу користувача обрано основні стандарти вебу – HTML5 і CSS3 [18,19,23]. Вони є фундаментальними технологіями, які підтримуються усіма сучасними браузерами і не вимагають додаткового серверного оточення для рендерингу сторінок. Це значно спрощує розгортання сайту на популярних хостингах, таких як GitHub Pages або Netlify, що особливо важливо для студентських проєктів та невеликих комерційних продуктів.

HTML5 забезпечує семантичну структуру документа, що позитивно впливає на доступність сайту для користувачів з інвалідністю, а також покращує індексацію сайту пошуковими системами. Використання семантичних тегів,

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

таких як <header>, <nav>, <article>, <section>, сприяє кращій організації контенту, що полегшує розуміння структури сайту як людьми, так і автоматизованими пошуковими роботами [10,8].

CSS3 використовується для стилізації веб-сторінок, включно з адаптивним дизайном, анімаціями, переходами та сітками. Особливо важливими є медіазапити (media queries), які дозволяють забезпечити коректне відображення сайту на різноманітних пристроях – від смартфонів і планшетів до десктопів [19]. Адаптивність є критичною, оскільки більшість сучасних користувачів заходить на сайти саме з мобільних пристроїв [23,25].

Для покращення інтерактивності та динамічності інтерфейсу було використано JavaScript – мови програмування, що дозволяє створювати динамічні елементи, такі як калькулятор вартості оренди, фільтри в реальному часі, а також змінювати контент без необхідності перезавантаження сторінки. JavaScript також підтримує роботу з API, що є важливим для інтеграції фронтенду з бекендом і сторонніми сервісами.

Сучасним трендом є застосування JavaScript-фреймворків та бібліотек, зокрема React або Vue.js, які дають змогу створювати компонентно-орієнтовані інтерфейси, що легко підтримувати, масштабувати та тестувати [14]. Наприклад, React має велику екосистему та активну спільноту, що забезпечує наявність готових рішень для типових завдань, що суттєво прискорює розробку і покращує якість кінцевого продукту [2].

Для серверної частини вебсайту важливо обрати технології, які забезпечать високу швидкість обробки запитів, надійність, масштабованість, а також простоту інтеграції з базами даних та іншими системами. Одним із найбільш актуальних виборів є Node.js – середовище виконання JavaScript на сервері, що дозволяє використовувати одну мову програмування для розробки як фронтенду, так і бекенду. Це значно спрощує розробку, тестування та підтримку коду.

У поєднанні з популярним фреймворком Express.js Node.js забезпечує легке створення RESTful API, яке реалізує функціонал аутентифікації

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів, обробки бронювань, платежів і взаємодії з базами даних. Для зберігання даних використовується комбінація баз даних.

Реляційні бази даних (PostgreSQL, MySQL) ідеально підходять для структурування інформації про користувачів, бронювання, автомобілі, платежі. Вони забезпечують надійність зберігання, підтримують складні запити і транзакції. Для зберігання неструктурованих даних або кешування доцільно застосувати NoSQL бази (MongoDB) [6], що підвищує продуктивність системи і дозволяє гнучко масштабувати сховище даних. Оскільки вебсайт працюватиме з персональними даними користувачів та фінансовою інформацією, забезпечення безпеки є пріоритетом.

До основних заходів безпеки, реалізованих у межах вебплатформи для оренди автомобілів, належить застосування SSL-сертифікатів, що забезпечують шифрування всього трафіку між клієнтом і сервером. Це унеможливорює перехоплення конфіденційних даних зловмисниками. Важливою складовою є також захист від SQL-ін'єкцій, що реалізується шляхом використання параметризованих запитів, а також застосуванням ORM (Object-Relational Mapping), що мінімізує прямий доступ до бази даних і підвищує загальну безпечність.

Для запобігання XSS-атакам, які пов'язані з впровадженням шкідливих скриптів у форми вводу, реалізується фільтрація і екранування введених користувачем даних. Крім того, задля посилення захисту облікових записів користувачів передбачене впровадження багатофакторної аутентифікації, що значно знижує ризики несанкціонованого доступу [3].

У сфері обробки фінансових операцій застосовується інтеграція з перевіреними платіжними шлюзами, такими як Stripe і PayPal. Вони відповідають міжнародним стандартам безпеки PCI DSS, що забезпечує захист платіжних транзакцій та конфіденційної інформації користувачів. Для підвищення зручності користувачів і покращення продуктивності передбачається реалізація прогресивного веб-додатку (PWA). PWA дозволяє

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

сайту працювати офлайн, надсилати push-повідомлення, а також бути встановленим на мобільні пристрої як нативний додаток, що значно покращує користувацький досвід.

Планується також інтеграція чат-бота на базі штучного інтелекту [11], який автоматизує обробку запитів клієнтів, скорочуючи час відповіді служби підтримки та підвищуючи рівень взаємодії з користувачами. Використання безсерверної архітектури (serverless) для окремих сервісів (наприклад, обробка зображень, відправка email) дозволить оптимізувати витрати на інфраструктуру, забезпечити автоматичне масштабування системи під навантаження та підвищити надійність роботи.

Обраний стек технологій гарантує гнучкість і масштабованість проєкту, адже дає змогу оперативно розширювати функціональні можливості, впроваджувати нові сервіси та обслуговувати велику кількість одночасних користувачів. Завдяки уніфікації мови програмування (JavaScript) як для фронтенду, так і для бекенду, істотно спрощується процес розробки, зменшуються витрати часу й полегшується командна робота.

Інтерфейс платформи характеризується високою якістю завдяки використанню сучасних фреймворків, що забезпечують інтерактивність, плавність функціонування та зручність користування [27]. Безпека користувацьких даних досягається завдяки реалізації багаторівневих заходів захисту та відповідності міжнародним стандартам у галузі інформаційної безпеки [28–30]. Крім того, використання відкритих технологій, безкоштовних рішень та хмарних сервісів із підтримкою масштабування забезпечує оптимальне співвідношення між продуктивністю й витратами на інфраструктуру.

Тому, обраний стек технологій відповідає сучасним вимогам веб-розробки, дозволяє створити сучасний, надійний і зручний у використанні вебсайт з оренди автомобілів, який легко адаптуватиметься до майбутніх змін і розширень.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4 Технічне завдання на розробку вебсайту з оренди автомобілів

Розробка інформаційної системи для онлайн-бронювання автомобілів передбачає створення функціонального, масштабованого та безпечного вебресурсу, орієнтованого на кінцевого користувача. При визначенні архітектури системи та виборі програмно-апаратного забезпечення враховувалися сучасні вимоги до продуктивності, стійкості до навантажень, зручності обслуговування, а також швидкості розробки та адаптації до ринкових змін[2]. Основу клієнтської частини проєкту складають мови розмітки HTML5 та каскадні таблиці стилів CSS3. Таке рішення обумовлено їхньою сумісністю з усіма сучасними браузерами, простотою інтеграції на стороні клієнта, а також можливістю швидкого розгортання на платформах типу GitHub Pages без потреби в окремому серверному середовищі.

Застосування семантичних HTML-елементів, таких як header, main, section, nav, article та інші, покращує структуру контенту, полегшує його індексацію пошуковими системами, а також забезпечує кращу доступність для користувачів з обмеженими можливостями. Крім того, через використання метатегів досягається базова оптимізація сайту для пошукових систем та мобільних пристроїв. Стилзація інтерфейсу реалізується за допомогою CSS3, де ключовим компонентом є адаптивна верстка, заснована на медіазапитах, що гарантує коректне відображення сайту на різних типах пристроїв, включно з мобільними телефонами, планшетами та десктопами.

Для забезпечення динамічності вебінтерфейсу була реалізована інтеграція JavaScript. Основними завданнями мови виступає реалізація логіки калькулятора вартості оренди, валідації форм, взаємодії з API та оновлення вмісту без перезавантаження сторінки. З огляду на потребу масштабування функціоналу й підвищення інтерактивності, передбачається подальше впровадження сучасних JavaScript-фреймворків, таких як React або Vue.js. Вони дозволяють будувати багаторазово використовувані компоненти інтерфейсу, забезпечують високу продуктивність та зменшують обсяг ручного коду. Зокрема, React, завдяки

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

розвинутій екосистемі, широкій підтримці спільноти та зручній інтеграції з бібліотеками керування станом, такими як Redux або Context API, є доцільним вибором для даного класу проєктів. У випадках, де важливим є швидке налаштування й редагування контенту без втручання в код, можлива інтеграція з CMS, наприклад, WordPress, через REST API, що спрощує адміністрування ресурсу.

Серверна частина вебресурсу реалізується на основі платформи Node.js з використанням фреймворку Express.js. Такий вибір зумовлений перевагами асинхронної обробки запитів, високою масштабованістю та уніфікацією стеку технологій, адже і клієнтська, і серверна частини розробляються на JavaScript. Node.js дозволяє ефективно обробляти одночасні підключення та забезпечує гнучку інтеграцію з базами даних. У якості системи керування базами даних передбачається використання як реляційних СУБД (зокрема PostgreSQL або MySQL), так і документно-орієнтованих NoSQL-баз, таких як MongoDB. Реляційні бази даних забезпечують цілісність і надійне збереження транзакційних даних, що включають інформацію про користувачів, автомобілі, бронювання та оплату. У свою чергу, NoSQL-рішення доцільні для обробки неструктурованих даних, зберігання журналів, кешування або реалізації елементів швидкого доступу. Забезпечення належного рівня інформаційної безпеки є одним з основних пріоритетів у розробці даного вебсайту. Планується впровадження шифрування за протоколом SSL/TLS, що дозволяє гарантувати захист передаваних даних між клієнтом та сервером. Також враховуються механізми захисту від атак типу SQL-ін'єкцій, міжсайтового скриптингу (XSS) та CSRF-атак. З метою підвищення захищеності акаунтів користувачів можлива реалізація багатофакторної аутентифікації, зокрема з використанням SMS або email-кодів підтвердження. Платіжні операції планується реалізовувати через інтеграцію з платіжними системами Stripe і PayPal, що відповідають міжнародним стандартам безпеки обробки платіжних даних, зокрема PCI DSS. У регіональному контексті доцільним є додаткове підключення API платіжної

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

системи Privat24, що є популярним платіжним засобом в Україні.

Для підвищення ефективності користувацької взаємодії з сайтом розглядається можливість реалізації прогресивного вебдодатку (Progressive Web App, PWA), що дозволяє працювати з ресурсом в офлайн-режимі, додавати його на головний екран пристрою та отримувати push-сповіщення. Такий підхід особливо ефективний у контексті мобільного користування. У довгостроковій перспективі заплановано впровадження чат-бота з елементами штучного інтелекту для автоматизації первинної взаємодії з клієнтами.

Такий компонент здатен розпізнавати типові запити, надавати відповіді в режимі реального часу та зменшувати навантаження на операторів служби підтримки. Окремі модулі функціоналу можуть бути реалізовані на основі безсерверної архітектури, зокрема із застосуванням функцій AWS Lambda, Firebase Cloud Functions або Vercel Serverless Functions, що дозволяє динамічно масштабувати обчислювальні ресурси відповідно до реального навантаження, оптимізувати витрати на інфраструктуру та підвищити загальну доступність сервісу. Тому, комплексне використання сучасних технологій веброзробки забезпечить створення ефективної, безпечної та зручної у користуванні інформаційної системи оренди автомобілів, яка відповідатиме сучасним вимогам до якості цифрових сервісів і матиме потенціал для подальшого розвитку та масштабування.

Висновки до першого розділу

У першому розділі було проаналізовано основні аспекти, що впливають на розробку сучасного вебсайту для оренди автомобілів. Було розглянуто сучасні технології веброзробки. На основі аналізу було визначено, що HTML5, CSS3 і JavaScript залишаються базовими інструментами, які забезпечують гнучкість, адаптивність, інтерактивність і швидкодію вебсайтів. Особливу увагу приділено їх інтеграції з фреймворками, що розширює функціональні можливості.

Було проведено порівняльний аналіз аналогічних платформ таких як Hertz,

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Avis, Europcar. Визначено основні переваги та недоліки конкурентів, а також виявлено відсутні функції, які враховано в розробці власного продукту LuxuryCars. Обґрунтовано вибір технологічного стеку для реалізації вебсайту. Обрані рішення – HTML5, CSS3, JavaScript, Node.js з Express.js, PostgreSQL, MongoDB – забезпечують надійність, масштабованість і безпеку, а також дозволяють гнучко розширювати платформу.

Також було сформульовано технічне завдання на створення вебресурсу. Визначено цільову аудиторію, вимоги до інтерфейсу, функціональні модулі, середовище розробки та умови розгортання. Це стало основою для наступних етапів моделювання та реалізації системи.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2.МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБСАЙТУ З ОРЕНДИ АВТОМОБІЛІВ

2.1 Розробка UML-моделей вебсайту з оренди автомобілів

Важливим етапом у процесі створення веб-ресурсу для оренди машин є моделювання функціональності системи. Для цього використовуються нотації UML (Unified Modeling Language), які дають змогу детально описати структуру, поведінку та взаємозв'язки між компонентами системи. Основна ціль моделювання – формалізувати функціональні та нефункціональні вимоги до системи, а також налагодити ефективну комунікацію між учасниками розробки. UML-діаграми дають змогу спроектувати логіку взаємодії користувачів із системою, моделі даних, послідовність дій, інфраструктуру та інші важливі складові. У процесі моделювання системи для розробки вебсайту оренди автомобілів застосовуються різні типи UML-діаграм, кожна з яких відіграє важливу роль у візуалізації певного аспекту архітектури чи функціональності. Діаграма варіантів використання (Use Case Diagram) дозволяє наочно продемонструвати функціональні можливості, доступні кожній ролі користувача, зокрема адміністратору, клієнту та неавторизованому відвідувачу.

Діаграма класів (Class Diagram) відображає об'єкти системи, їхні атрибути, методи та взаємозв'язки, що дозволяє сформулювати базову структуру програмної моделі. У свою чергу, діаграма розгортання (Deployment Diagram) описує інфраструктурне середовище, у якому працює система, та демонструє розміщення програмних компонентів на фізичних або віртуальних пристроях.

Діаграма активності (Activity Diagram) ілюструє покрокову логіку виконання процесів, таких як пошук автомобіля або оформлення замовлення, та дозволяє виявити можливі гілки або винятки в логіці взаємодії. Діаграма послідовності (Sequence Diagram) фіксує порядок обміну повідомленнями між об'єктами під час реалізації конкретного сценарію, що дає змогу формалізувати взаємодію між фронтендом, бекендом та базою даних.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

На наступній діаграмі використання показано дії, доступні зареєстрованим користувачам і гостям (рис. 2.1).

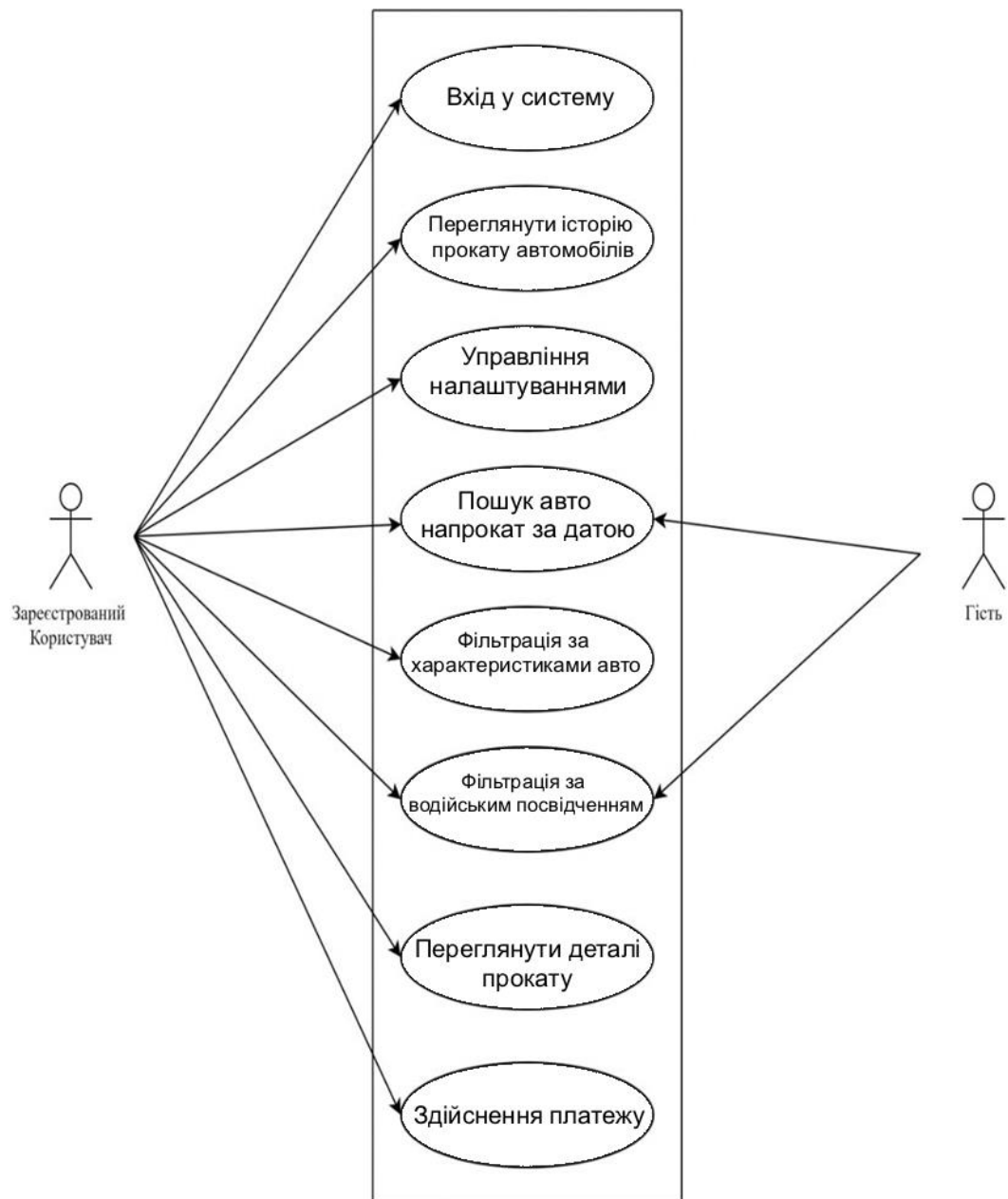


Рис. 2.1. Діаграма варіантів використання для користувача "Клієнт"

Зареєстрований користувач: повинен мати можливість входити в систему, переглядати свою історію оренди, створити профіль, переглянути список орендованих автомобілів за датою, фільтрувати за типом і характеристиками водійських прав, переглядати інформацію про оренду та здійснювати платежі.

Гостьовий користувач: повинен мати можливість перегляду автомобілів за датою та фільтрації за характеристиками.

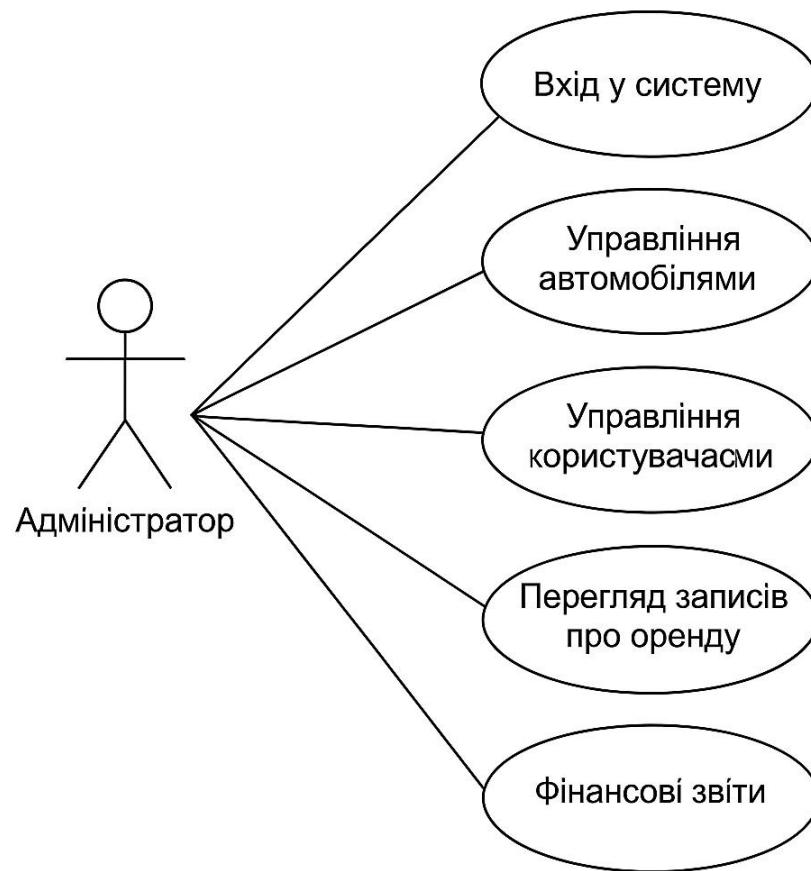


Рис. 2.2. Діаграма варіантів використання для користувача "Адміністратор"

Ця діаграма є основою для реалізації структури бази даних та логіки програмної системи (рис. 2.2). Адміністратор має змогу: додавати нові автомобілі до каталогу, редагувати або видаляти інформацію про авто, переглядати аналітику (кількість замовлень, фінанси), підтверджувати або відхиляти бронювання, модерація відгуків та контенту, керування користувачами (деактивація, блокування).

Діаграма класів (рис. 2.3) описує основні об'єкти (класи), які функціонують у системі, та їхні зв'язки між собою. Кожен клас має атрибути (властивості) та методи (функції), що реалізують логіку роботи. У системі передбачено кілька основних класів. Клас user (користувач) містить такі атрибути: ID, ім'я, email, пароль, роль (адміністратор або клієнт).

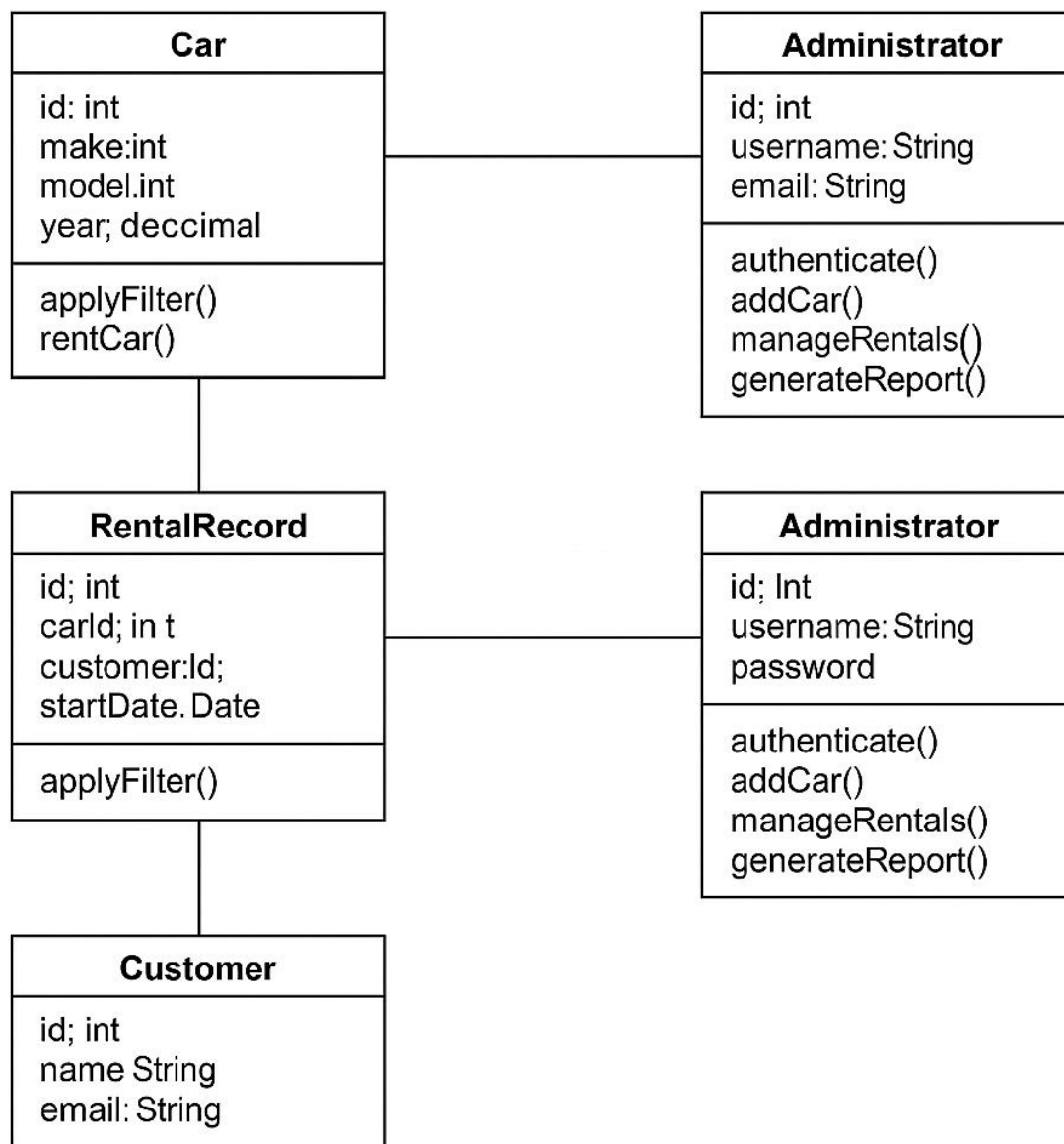


Рис. 2.3. Діаграма класів вебсайту з оренди автомобілів

Основні методи цього класу—реєстрація(), авторизація() та перегляд Історії(). Клас car (автомобіль) включає атрибути: ID, марка, модель, рік, ціна, доступність, а також методи: редагувати(), додати(), видалити(). Клас booking (бронювання) описується такими атрибутами: ID, ID користувача, ID авто, дата початку, дата завершення, статус, і має методи підтвердити() та скасувати(). Клас review (відгук) містить атрибути ID, ID користувача, текст, рейтинг, дата, а також методи залишити() і видалити().

Використання UML-діаграм суттєво покращує процес розробки, оскільки

дозволяє візуалізувати структуру системи, взаємодію її компонентів та логіку функціонування. Це забезпечує чіткість технічного завдання, спрощує комунікацію між учасниками проєкту та дає можливість уникнути багатьох помилок на етапі проєктування.

Однією з важливих UML-діаграм є діаграма розгортання (рис. 2.4), яка відображає фізичну архітектуру програмної системи, її інфраструктуру та компоненти, що взаємодіють у реальному середовищі.

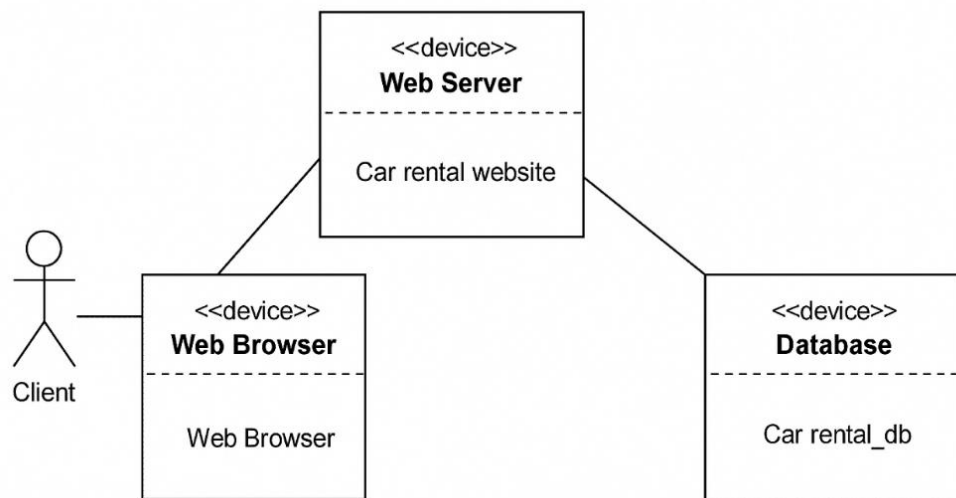


Рис. 2.4. Діаграма розгортання вебсайту з оренди автомобілів

У контексті розробки вебсайту для оренди автомобілів діаграма розгортання показує апаратне та програмне середовище, в якому функціонує система. Зокрема, вона демонструє взаємозв'язки між такими ключовими компонентами, як клієнтські пристрої (зазвичай веббраузери користувачів, через які здійснюється доступ до сайту), вебсервер (на якому розгорнуто фронтенд – наприклад, на платформах Netlify або GitHub Pages), а також серверна логіка, що виконується на платформі Node.js або PHP.

Крім цього, важливим компонентом є база даних, яка зберігає інформацію про користувачів, автомобілі, бронювання, відгуки тощо. У якості СУБД може використовуватися MySQL або MongoDB, залежно від обраної технологічної платформи. Для обробки онлайн-платежів інтегрується платіжний шлюз, зокрема Stripe або PayPal, які забезпечують захищене проведення фінансових

транзакцій. Також система може використовувати хмарне сховище для зберігання зображень автомобілів, наприклад, Firebase Storage або інші сервіси, що дозволяють зберігати та швидко завантажувати мультимедійні ресурси.

Загалом така архітектура забезпечує гнучкість, масштабованість та високу доступність системи, дозволяючи ефективно обслуговувати запити користувачів, зберігати великі обсяги даних і підтримувати сучасні функціональні можливості вебдодатку.

2.2 Розробка функціональних алгоритмів роботи вебсайтів

Після того, як структура та взаємодії в системі будуть формалізовані, наступним кроком є розробка алгоритмів, які забезпечують її функціонування. Основна мета цього етапу – детально описати логіку взаємодії користувачів з веб-ресурсом та обробки запитів у вигляді чітко визначених послідовностей дій, використовуючи для цього відповідні UML-діаграми (зокрема, діаграми активності та послідовності).

Діаграми активності (Activity Diagrams) (рис. 2.5) належать до класу поведінкових діаграм і слугують для моделювання алгоритмів, робочих процесів (workflow) та сценаріїв взаємодії. Вони є аналогами блок-схем і наочно ілюструють, як система реагує на дії користувача, з урахуванням гілок, умов, виключень і паралельних процесів. Наприклад, процес пошуку автомобіля починається з входу користувача на головну сторінку, після чого він здійснює вибір параметрів фільтрації, таких як місто, дата, марка автомобіля та категорія водійських прав. Після надсилання запиту система обробляє введені дані та формує список доступних автомобілів. Користувач може обрати конкретне авто та перейти до етапу бронювання. У разі некоректних даних або відсутності відповідних варіантів система інформує користувача про помилку або пропонує змінити параметри. Також реалізована перевірка відповідності категорії водійських прав класу вибраного автомобіля.

Ще одним важливим алгоритмом є процес оформлення бронювання. Він

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

передбачає авторизацію або реєстрацію користувача, вибір автомобіля, введення терміну оренди та подальшу перевірку наявності необхідних документів і категорії водійських прав. Після успішної перевірки користувач підтверджує замовлення, переходить до оплати (через платіжні сервіси Stripe або PayPal), після чого система генерує підтвердження бронювання. Передбачено також обробку виняткових ситуацій, наприклад: відмова у бронюванні через невідповідність категорії прав, відмова у випадку незаповнених або некоректних даних, а також обробка помилки оплати, яка передбачає перенаправлення користувача на сторінку повторної спроби.

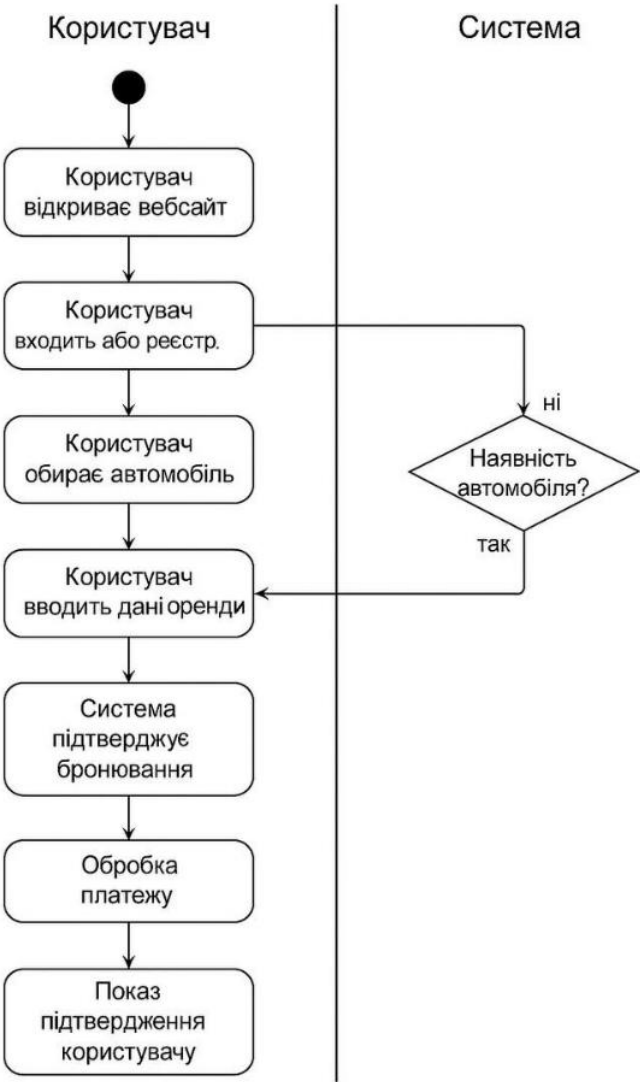


Рис. 2.5. Діаграма активності процесу оренди авто

Усі ці процеси можуть бути формалізовані в UML-діаграмах активності та послідовності (рис . 2.6), що дозволяє точно описати логіку роботи вебсайту та забезпечити надійне функціонування системи.

Діаграма активності відображає послідовність дій, які виконує користувач або система для реалізації конкретного сценарію взаємодії. Один із ключових сценаріїв—це «Оренда автомобіля». У рамках цього процесу користувач спочатку заходить на сайт і проходить авторизацію, або ж користується гостьовим доступом. Далі він обирає бажаний автомобіль із доступного переліку, встановлює дати початку та завершення оренди, після чого підтверджує своє замовлення. На наступному етапі користувач обирає зручний спосіб оплати (наприклад, через платіжний шлюз Stripe або PayPal). Після успішної оплати система генерує підтвердження та надсилає його користувачеві на електронну пошту.

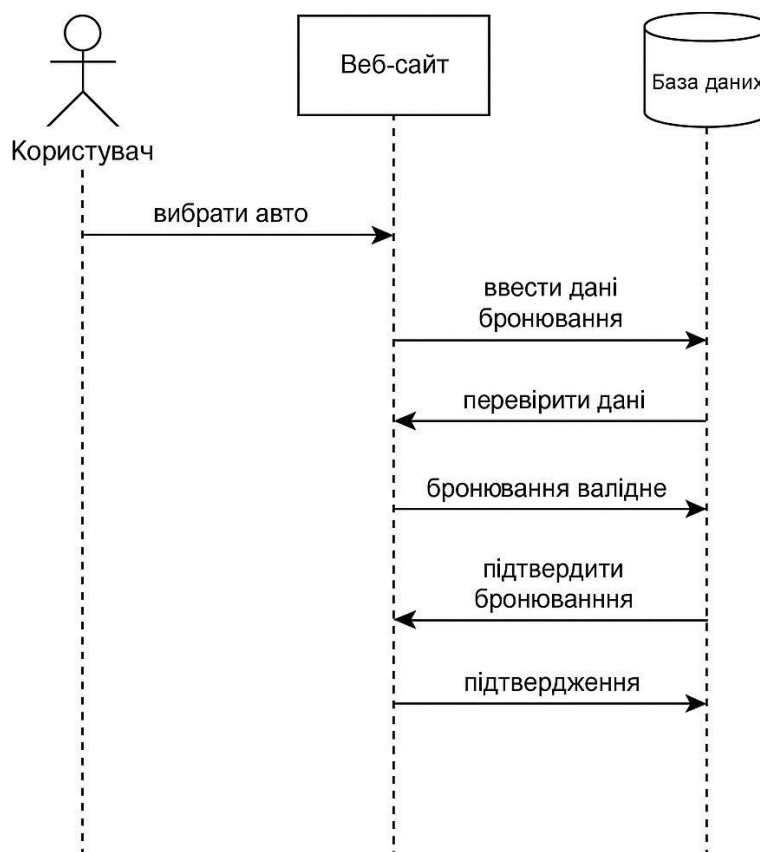


Рис. 2.6. Діаграма послідовності оформлення бронювання

Ця діаграма описує взаємодію між основними компонентами системи під

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

час процесу бронювання автомобіля. На початку користувач надсилає запит на оренду обраного транспортного засобу. Система, у свою чергу, перевіряє наявність відповідного автомобіля у задані дати та створює тимчасове бронювання. Після цього користувачу надсилається запит на підтвердження дії. У разі згоди, система обробляє оплату за допомогою інтегрованого платіжного шлюзу, змінює статус бронювання на активний і надсилає повідомлення про успішне завершення процесу оренди. Ці алгоритми відображають бізнес-логіку, що лежить в основі функціонування вебсайту, та забезпечують формалізоване інженерне представлення ключових дій, які виконуються як користувачами, так і системою. Такий підхід значно спрощує реалізацію ефективних механізмів обробки запитів, перевірки коректності введених даних та забезпечення стабільної роботи системи в цілому.

2.3 Обґрунтування архітектури та розробка модулів

Архітектура вебсайту з оренди автомобілів повинна забезпечувати високий рівень масштабованості, зручності обслуговування, ефективності взаємодії між компонентами та безпеки обробки користувацьких даних. З огляду на ці вимоги, доцільним є застосування клієнт-серверної архітектури, що дозволяє розділити обов'язки між фронтенд- та бекенд-частинами системи. Такий підхід сприяє модульності, гнучкості в обслуговуванні та майбутньому масштабуванню проекту.

Клієнтська частина (frontend) відповідає за візуальне відображення інтерфейсу користувача, обробку взаємодії та забезпечення доступу до основних функцій сайту. Для реалізації інтерфейсу використано мови розмітки HTML5, стилізації CSS3 та скриптову мову JavaScript, що забезпечує адаптивність, кросбраузерність, швидкодію і взаємодію з динамічними компонентами [5,10,26]. Також передбачена можливість розширення функціоналу шляхом впровадження сучасних JavaScript-фреймворків, зокрема React або Vue.js, що базуються на компонентному підході до побудови UI. Компонентна структура

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволяє спростити повторне використання елементів, підвищити швидкість розробки та забезпечити ефективне управління станом застосунку.

Серверна частина (backend) реалізує бізнес-логіку застосунку, обробку запитів, зберігання даних, автентифікацію користувачів і взаємодію з платіжними системами. Для реалізації обрано середовище Node.js у поєднанні з фреймворком Express.js, що дозволяє створювати RESTful API з ефективною обробкою асинхронних запитів. Завдяки такому підходу досягається висока продуктивність при мінімальних затратах ресурсів сервера.

Уніфікація мови програмування (JavaScript) як на фронтенді, так і на бекенді значно спрощує процес розробки, обслуговування та тестування системи [14]. Це дозволяє залучати менше ресурсів на команду розробки, зменшує кількість помилок, пов'язаних з міжмовною взаємодією, та забезпечує узгоджене використання бібліотек і стандартів кодування. Для зберігання структурованих даних (користувачі, транспортні засоби, бронювання) застосовано реляційні системи управління базами даних, зокрема MySQL або PostgreSQL. У перспективі для обробки неструктурованих даних або тимчасового кешування можливе використання NoSQL-сховищ (наприклад, MongoDB або Redis). Такий гібридний підхід дозволяє адаптувати зберігання даних відповідно до специфіки кожного модуля, покращуючи загальну продуктивність та відмовостійкість системи. На основі визначених функціональних вимог було спроектовано декілька логічних модулів, які забезпечують ефективну реалізацію всіх ключових процесів вебсайту: Модуль авторизації виконує функції реєстрації нових користувачів, входу до системи, відновлення паролю, а також підтримує базову автентифікацію з можливістю впровадження багатофакторної перевірки. Передбачено використання токенів доступу (наприклад, JWT) для реалізації безпечної сесійної моделі. Модуль керування автомобілями надає адміністраторам можливість додавати нові транспортні засоби, редагувати існуючу інформацію, змінювати технічні характеристики, категорії, зображення, а також здійснювати видалення застарілих або неактуальних

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

записів. Модуль включає перевірку валідності введених даних та управління медіа-файлами.

Модуль бронювання відповідає за створення замовлень на оренду, перевірку доступності автомобілів у зазначені дати, підтвердження або скасування бронювання, а також відображення історії замовлень у профілі користувача. У випадку конфлікту бронювань система автоматично попереджає користувача про накладання дат. Модуль обробки платежів забезпечує інтеграцію з платіжними сервісами, зокрема Stripe та PayPal. Він виконує перевірку платіжних даних, ініціює проведення транзакцій, здійснює обробку помилок та формує відповідне підтвердження успішної оплати. Усі транзакції передаються по захищених каналах із застосуванням HTTPS і TLS [16]. Модуль управління користувачами забезпечує доступ до персонального кабінету, редагування персональних даних, перегляд історії бронювань, а також реалізацію інших функцій, пов'язаних із взаємодією між користувачем та системою. Передбачено можливість зміни пароля, видалення облікового запису та керування сповіщеннями. Модуль аналітики призначений для збору статистичних даних, аналізу ефективності роботи сервісу, а також формування управлінських звітів, що використовуються адміністрацією для прийняття обґрунтованих рішень. Він також дозволяє виявляти пікові години навантаження, популярні категорії автомобілів та джерела трафіку.

У контексті сучасної розробки веборієнтованих програмних рішень модульна клієнт-серверна архітектура демонструє високу ефективність завдяки своїй гнучкості та здатності до масштабування. Завдяки чіткому розмежуванню логіки представлення, обробки даних та доступу до сховищ інформації, така архітектура забезпечує ефективну взаємодію між клієнтською частиною та серверною логікою. Це особливо важливо для систем, які мають динамічно зростаючу аудиторію користувачів, змінні вимоги до функціональності або плануються до подальшого розвитку у вигляді мультиплатформних застосунків.

Кожен модуль в обраній архітектурі реалізує окрему бізнес-логіку та має

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

чітко визначені інтерфейси взаємодії, що мінімізує взаємозалежності між складовими системи. Це, у свою чергу, сприяє підвищенню надійності, оскільки відмова одного модуля не призводить до критичного порушення функціонування всієї системи. Крім того, такий підхід полегшує тестування, оскільки кожен модуль може перевірятися ізольовано, що відповідає принципам модульного та інтеграційного тестування. Для обґрунтування вибору клієнт-серверної архітектури було здійснено порівняльний аналіз основних підходів до побудови вебсистем: монолітної, клієнт-серверної та мікросервісної. Основні характеристики наведено в (табл. 2.1), де визначено ключові відмінності між архітектурами за критеріями складності, масштабованості, керованості та витрат на підтримку.

Таблиця 2.1

Ключові відмінності між архітектурами за критеріями

Параметр	Монолітна архітектура	Клієнт-серверна архітектура	Мікросервісна архітектура
Складність реалізації	Низька	Середня	Висока
Масштабованість	Низька	Середня	Висока
Підтримка та оновлення	Ускладнено	Помірно	Гнучка, ізольовані модулі
Швидкість розробки	Висока на старті	Середня	Повільна на початку
Розділення відповідальностей	Відсутнє	Фронт/бек окремо	Повне, за кожен сервіс
Вимоги до інфраструктури	Мінімальні	Стандартні	Складні (Docker, CI/CD тощо)
Використання в проєкті	Ні	Так	Ні

Використання REST API в якості основного механізму взаємодії між клієнтом і сервером обумовлюється його простотою, легкістю впровадження, а також широкою підтримкою серед фреймворків і бібліотек. REST-орієнтований підхід дозволяє створювати незалежні клієнти, які можуть взаємодіяти із серверною частиною незалежно від платформи, що відкриває широкі можливості для подальшого розвитку вебсайту у вигляді мобільного застосунку або PWA (Progressive Web Application).

Проведене дослідження архітектурних підходів засвідчило, що хоча мікросервісна архітектура забезпечує найвищу гнучкість і масштабованість, її впровадження потребує значних ресурсів, складних інструментів оркестрації та високого рівня технічної експертизи. З іншого боку, монолітна архітектура має низький поріг входу, проте не дозволяє ефективно масштабувати систему та супроводжувати її в умовах зростання функціонального навантаження. У зв'язку з цим, клієнт-серверна модель була обрана як оптимальний компроміс між простотою реалізації, гнучкістю, стабільністю та перспективами розвитку. Тому чином, обрана архітектурна модель не лише задовольняє поточні вимоги до побудови вебсайту з оренди автомобілів, а й створює надійний фундамент для подальшого розширення функціоналу, інтеграції з іншими сервісами та адаптації до нових технологічних викликів у сфері цифрових сервісів.

Висновки до другого розділу

У процесі моделювання та проєктування вебсайту для оренди автомобілів було побудовано повноцінну логічну структуру програмного продукту із застосуванням сучасних засобів візуального моделювання на основі UML. Основною метою цього етапу було формалізувати функціональні вимоги, визначити взаємодію між користувачами й системою, а також забезпечити уніфіковану структуру об'єктів.

Зокрема, було створено діаграми варіантів використання, які відображають усі можливі сценарії взаємодії клієнта та адміністратора з платформою. Це

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволило чітко визначити основні ролі користувачів та доступні для них функції, включно з переглядом, фільтрацією, бронюванням і керуванням транспортними засобами.

Окрему увагу приділено проектуванню структури бази даних і реалізації об'єктно-орієнтованої моделі системи за допомогою діаграми класів. Вона відображає ключові сутності, їх атрибути, методи, а також зв'язки між об'єктами, що забезпечує логічну цілісність системи.

Було також побудовано діаграми активності та послідовності, які описують покрокову взаємодію між компонентами під час здійснення ключових операцій, таких як фільтрація автомобілів, оформлення замовлення, перевірка користувача, обробка платежу.

У межах архітектурного проектування обґрунтовано вибір клієнт-серверної архітектури з чітким розподілом функціоналу між фронтендом та бекендом. Запропонована архітектура забезпечує розділення відповідальності, полегшує масштабування та подальший розвиток.

Було визначено основні функціональні модулі вебплатформи: авторизації, керування користувачами, управління транспортними засобами, модуль бронювання, модуль оплати та адміністрування. Передбачено реалізацію особистого кабінету користувача, адміністративної панелі, а також можливість інтеграції чат-бота на базі штучного інтелекту. Результати моделювання створили міцну основу для реалізації ефективної та масштабованої системи оренди автомобілів, що відповідає як технічним, так і функціональним вимогам сучасних вебресурсів.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ПЛАН ВИКОНАННЯ ПРОЄКТУ ВЕБСАЙТУ

3.1 Інтерфейс програмного продукту

Інтерфейс користувача вебсайту з оренди автомобілів є одним із ключових елементів, що визначають якість взаємодії кінцевого користувача з програмною системою. Його роль полягає не лише у візуальному представленні інформації, а й у забезпеченні логічної, послідовної та комфортної взаємодії з функціоналом платформи. В умовах високої конкуренції на ринку цифрових сервісів, інтерфейс повинен відповідати сучасним вимогам до UI/UX-дизайну [31,32], бути адаптивним до різних типів пристроїв (мобільних, планшетів, десктопів), підтримувати принципи доступності (accessibility), а також відповідати загальноприйнятим патернам поведінки користувача в інтернет-середовищі. Основною вимогою до дизайну інтерфейсу є його інтуїтивна зрозумілість – користувач повинен без додаткового навчання розуміти, яким чином здійснюється пошук автомобіля, процес бронювання, авторизація, перегляд інформації про послуги та умови оренди. Крім того, важливу роль відіграє візуальна привабливість – використання гармонійної кольорової палітри, контрасту, типографіки та іконографіки, що покращують сприйняття та зменшують когнітивне навантаження.

Головна сторінка (рис. 3.1) виконує функцію початкової точки входу до системи та орієнтує користувача в її структурі. На цій сторінці зазвичай розміщуються елементи привітання, короткий опис компанії, головні переваги сервісу, актуальні пропозиції, а також доступ до основних функціональних модулів: перегляд автопарку, бронювання, контактна інформація, вхід у профіль. Вона також виконує роль навігаційного хабу, забезпечуючи легкий доступ до всіх ключових розділів сайту. З огляду на адаптивність, інтерфейс реалізовано з використанням технологій HTML5, CSS3 та фреймворків, що підтримують респонсивну верстку (наприклад, Bootstrap або Flexbox Grid). Це забезпечує коректне відображення елементів на екранах різної роздільної здатності, що є

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

критично важливим в умовах широкого використання мобільних пристроїв.

Якісно спроектований інтерфейс користувача відіграє вирішальну роль у залученні, утриманні та задоволенні користувача, а також безпосередньо впливає на комерційну успішність та конкурентоспроможність вебресурсу.

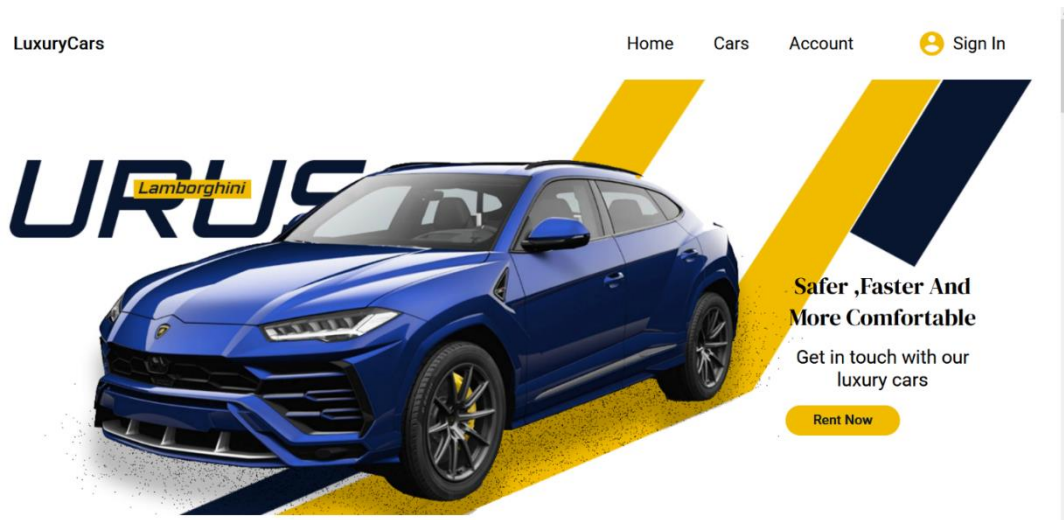


Рис. 3.1. Головна сторінка

Одним із ключових елементів головної сторінки є фільтр для пошуку автомобілів (рис. 3.2), який забезпечує швидкий і зручний доступ до релевантних пропозицій, орієнтованих на конкретні запити користувача. Фільтр реалізовано у вигляді інтерактивної форми, що дозволяє вводити такі параметри, як місто, конкретна адреса або локація отримання автомобіля, а також вибір дат оренди – початку й завершення. Це дозволяє системі одразу формувати список доступних транспортних засобів відповідно до введених критеріїв, що значно економить час користувача та підвищує рівень задоволеності сервісом.

Рис. 3.2. Фільтр для пошуку авто

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк. 45
		Лобанчикова Н.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізація подібного функціоналу ґрунтується на принципах персоналізації, що є важливою складовою сучасного UX-дизайну. Користувач отримує не загальний перелік усіх доступних авто, а лише ті, що відповідають його індивідуальним потребам та обставинам. Крім того, наявність динамічного фільтра пошуку сприяє зменшенню навантаження на інтерфейс, оскільки усуває потребу в переході на додаткові сторінки або виконанні зайвих дій.

Сторінка «Сервіси та послуги компанії» (рис.3.3) надає детальну інформацію про послуги, які пропонує компанія. Вона структурована так, щоб кожен користувач міг легко знайти потрібну інформацію та зрозуміти переваги співпраці.

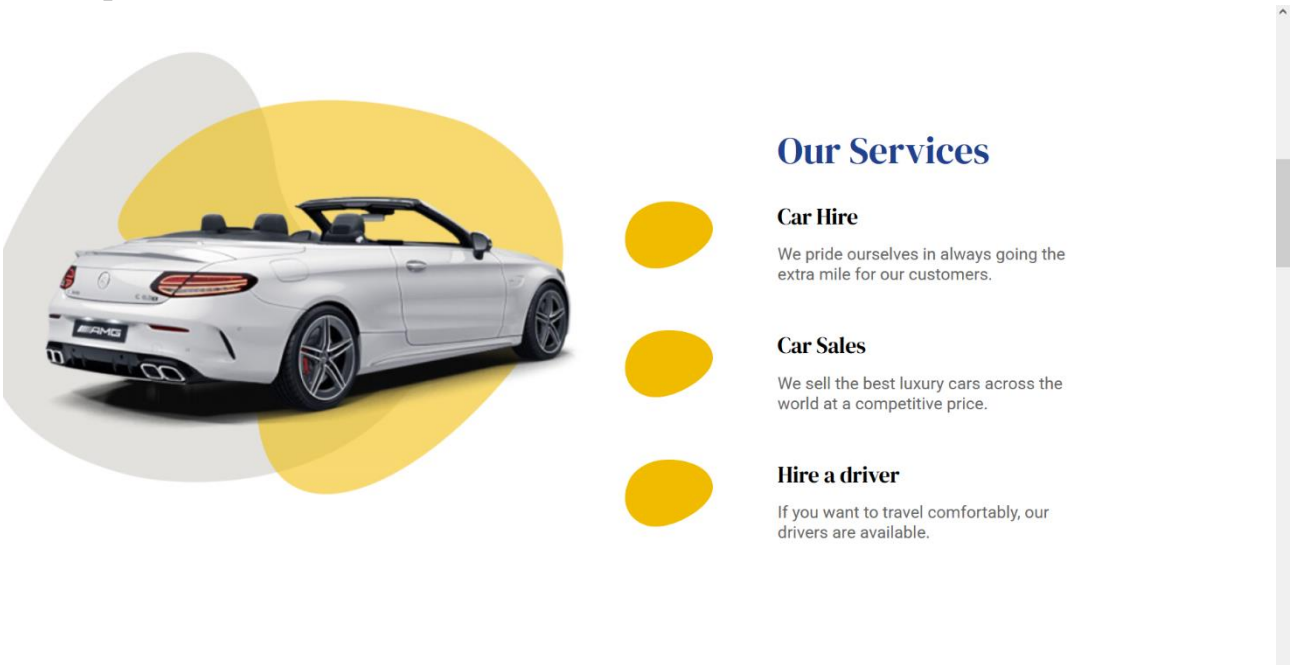


Рис. 3.3. Сторінка «Сервіси та послуги компанії»

Сторінка каталогу автомобілів (рис. 3.4, рис. 3.5) виконує ключову роль у взаємодії користувача з вебресурсом, оскільки саме через неї реалізується основна бізнес-функція платформи – оренда або купівля транспортного засобу. Інтерфейс цієї сторінки побудований за принципами зручності, інтуїтивності та функціональної насиченості. Користувач має змогу переглядати повний перелік доступних автомобілів, який представлено у вигляді інтерактивної сітки або списку з короткою інформацією про кожен автомобіль – марку, модель, клас, вартість оренди, доступність, основні технічні характеристики та фото.

We Have Everything You Need

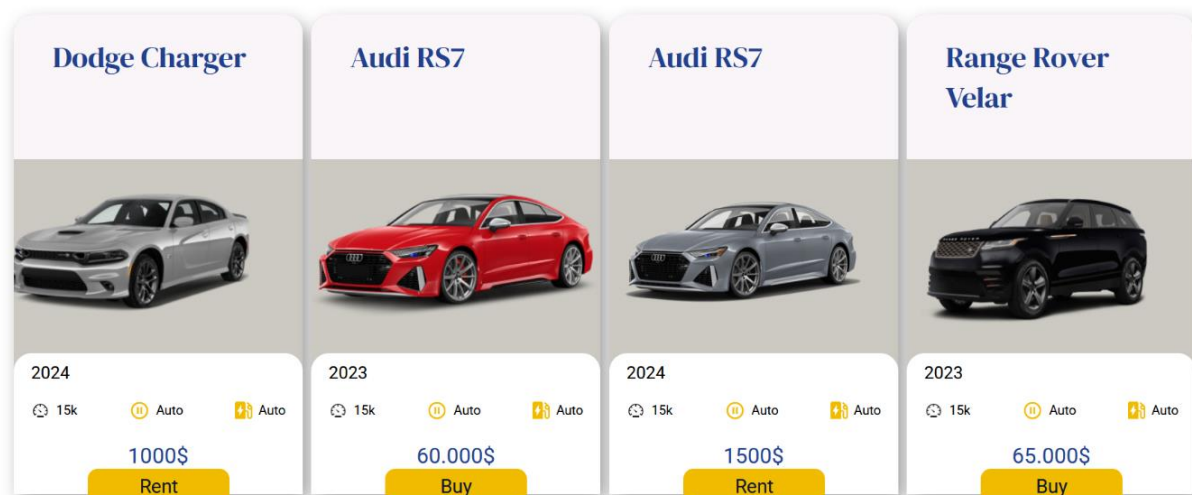


Рис.3.4. Сторінка каталогу автомобілів

Особливістю реалізації каталогу є можливість багатокритеріальної фільтрації за такими параметрами, як тип авто (седан, позашляховик, кабріолет тощо), марка, ціновий діапазон, тип трансмісії, паливна ефективність, а також в майбутньому наявність певних опцій (GPS, кондиціонер, автоматична коробка передач тощо).

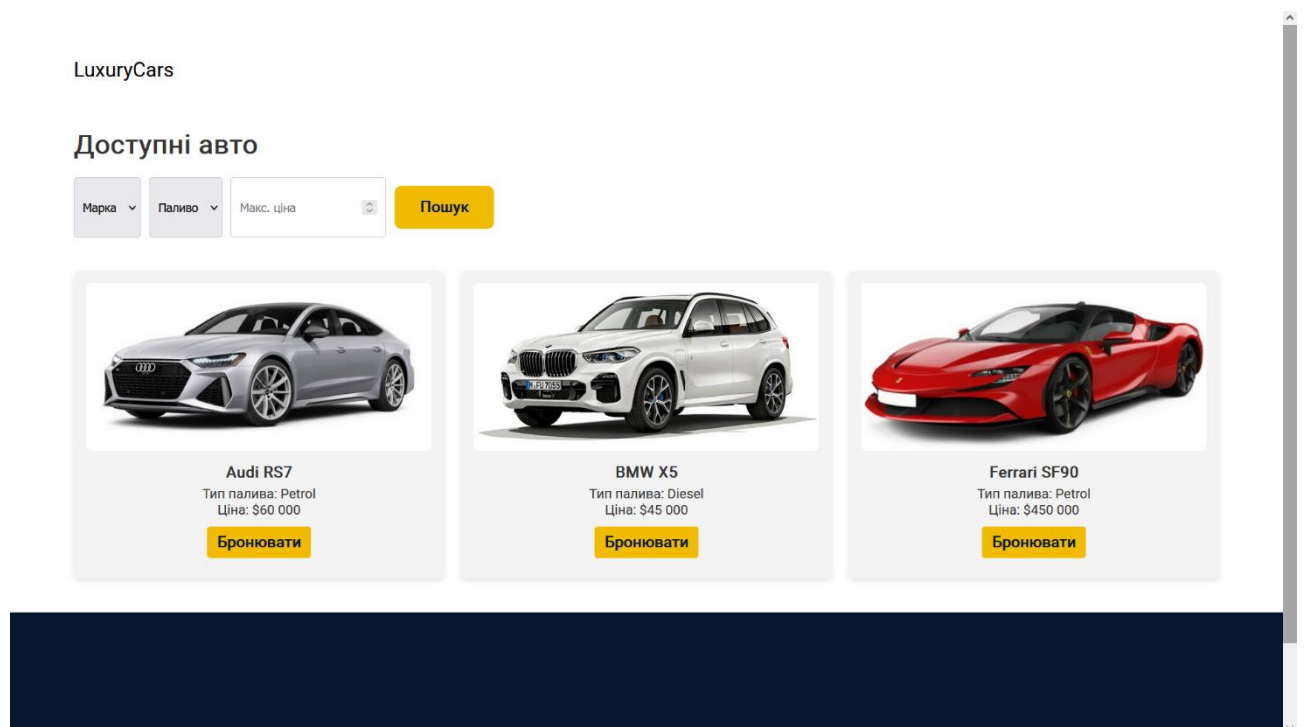


Рис.3.5. Сторінка каталогу автомобілів

Ця особливість дозволяє користувачу швидко знайти оптимальний варіант відповідно до особистих потреб. Крім того, кожен елемент списку містить

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

кнопку або посилання для переходу на сторінку з розгорнутою інформацією про конкретне авто та можливістю ініціювати процес бронювання.

Інтерактивність каталогу реалізовано з використанням сучасних вебтехнологій – зокрема, JavaScript-фреймворків (React або Vue) та асинхронної взаємодії з сервером (через REST API), що забезпечує динамічне оновлення вмісту без потреби в перезавантаженні сторінки. Саме тому, сторінка каталогу авто є не лише вітриною платформи, а й критично важливим функціональним вузлом, який формує досвід користувача, впливає на швидкість ухвалення рішення та сприяє ефективній реалізації бізнес-цілей сайту.

Для підвищення залученості користувачів і формування стабільної клієнтської бази на вебсайті реалізовано функціональність підписки на інформаційну розсилку (рис. 3.6).

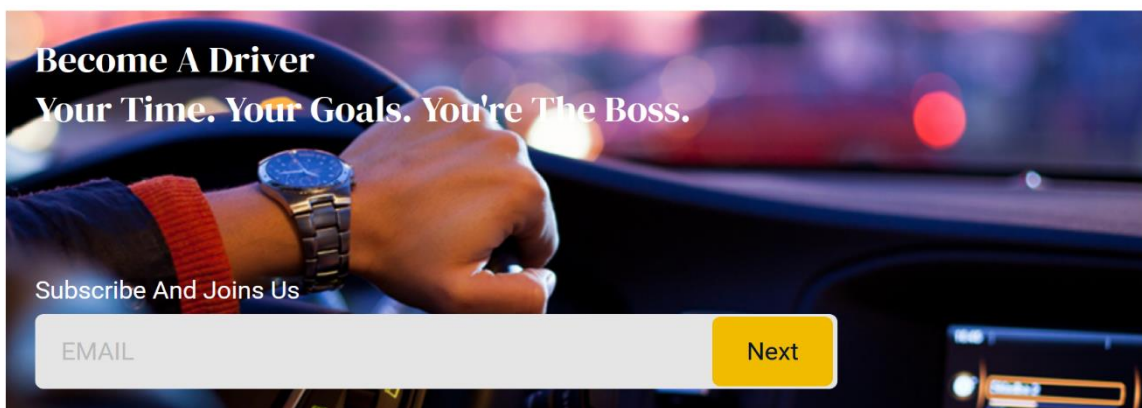


Рис. 3.6. Сторінка підписки на оновлення

Цей інструмент є ефективним каналом комунікації між платформою та її користувачами, що дозволяє оперативно інформувати аудиторію про новини компанії, оновлення автопарку, акційні пропозиції, знижки та інші маркетингові ініціативи. Підписка реалізована у вигляді інтерактивної форми, яка розміщується в нижній частині головної сторінки або у вигляді спливаючого віджета з можливістю введення електронної пошти. З технічної точки зору, механізм підписки побудовано з урахуванням принципів безпеки та захисту персональних даних. Зокрема, реалізовано підтвердження підписки через email (double opt-in), що відповідає вимогам GDPR та іншим сучасним стандартам

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

конфіденційності. Зібрані контактні дані зберігаються в окремій базі даних, інтегрований з системою автоматичної email-розсилки (наприклад, SendGrid або Mailchimp), що дозволяє забезпечити регулярність та персоналізацію розсилок. Впровадження цієї функції сприяє підвищенню рівня лояльності користувачів, покращенню інформованості клієнтів щодо сервісів компанії, а також виступає додатковим маркетинговим інструментом для просування нових послуг і розширення охоплення аудиторії.

Футер сайту (рис. 4.7) містить корисні посилання (на політику конфіденційності, умови використання тощо), а також іконки соцмереж, що сприяє побудові довіри та дозволяє користувачам залишатися на зв'язку з компанією. А також кнопку переходу на панель адміністратора.

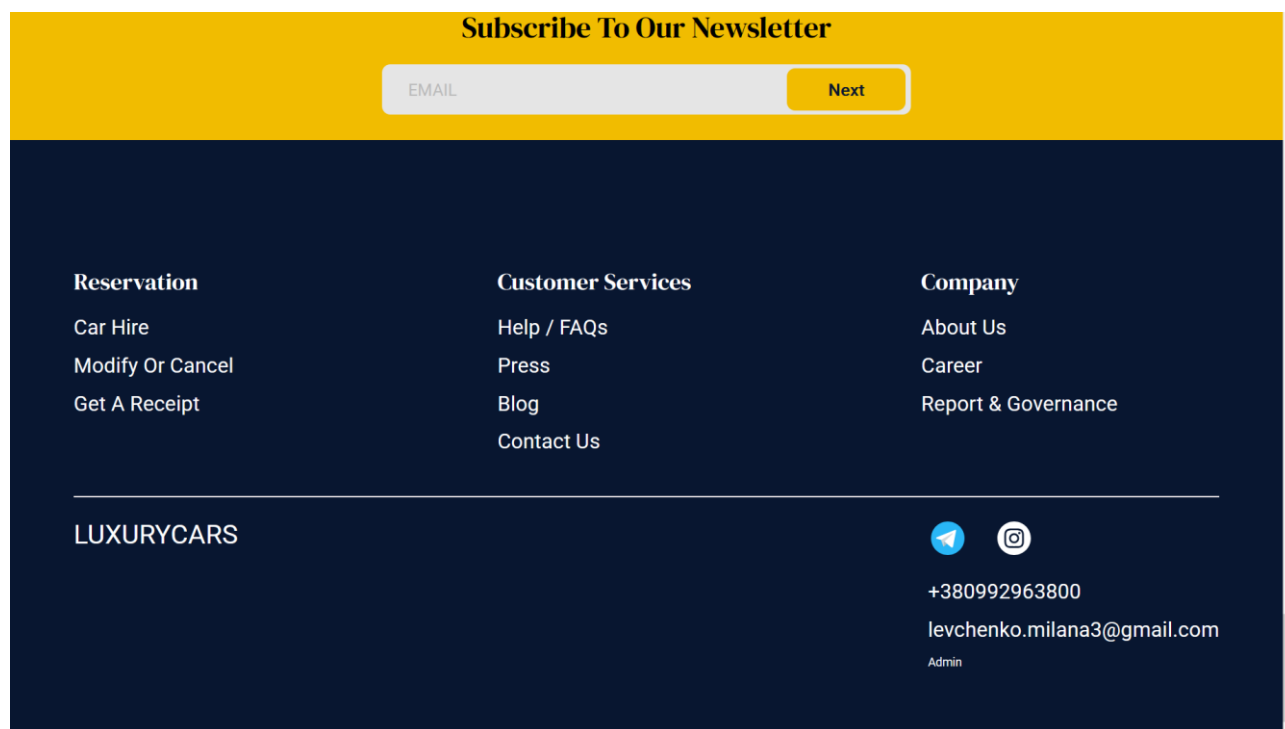


Рис. 3.7. Футер із посиланнями на соцмережі

Особистий кабінет користувача (рис. 3.8) є ключовим елементом персоналізації взаємодії з платформою. У цьому розділі користувач має доступ до історії оренди, збережених замовлень, персональних даних та налаштувань облікового запису. Також планується реалізувати можливість залишати відгуки, оцінювати послуги, переглядати активні та завершені бронювання. Передбачено опцію редагування профілю, зміну пароля, а також перегляд інформації щодо

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

оплати. Інтерфейс кабінету розроблено з урахуванням принципів юзабіліті – проста навігація, логічна структура, візуально виокремлені блоки інформації.

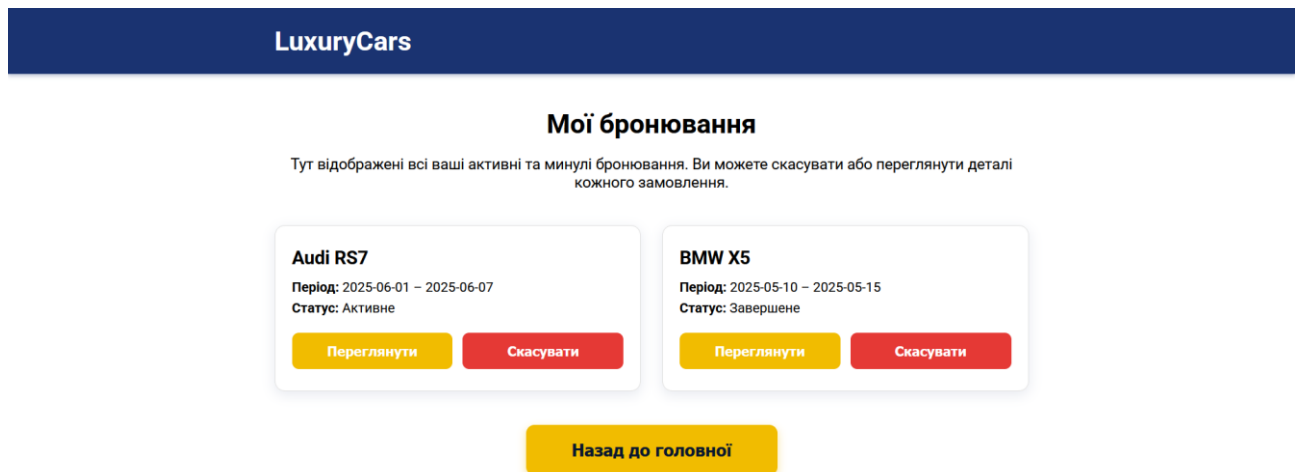


Рис. 3.8. Особистий кабінет користувача

Панель адміністратора (рис. 3.9) дозволяє внутрішнім користувачам системи (менеджерам, адміністраторам) здійснювати управління контентом сайту, контролювати процеси бронювання, оновлювати дані про автопарк, переглядати статистику та аналітику.



Рис. 3.9. Панель адміністратора

Передбачено інтерфейс для модерації відгуків, обробки звернень, а також формування звітності щодо завантаженості сервісу, доходів тощо. Усі дії адміністратора логуються для забезпечення прозорості змін. Панель має обмежений доступ із системою авторизації.

Сторінка оплати бронювання (рис. 3.10) забезпечує користувачам зручний і безпечний інтерфейс для завершення транзакцій при оренді автомобілів. Вона інтегрує два популярні платіжні сервіси – Приват24 та PayPal, що дає змогу клієнтам обрати найбільш зручний та знайомий спосіб оплати. Завдяки цьому

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечується широка географія підтримки, різноманітність валют та високий рівень безпеки фінансових операцій. Інтерфейс сторінки спрощує введення платіжних даних, автоматично перевіряє їх коректність та забезпечує шифрування інформації для захисту від несанкціонованого доступу.

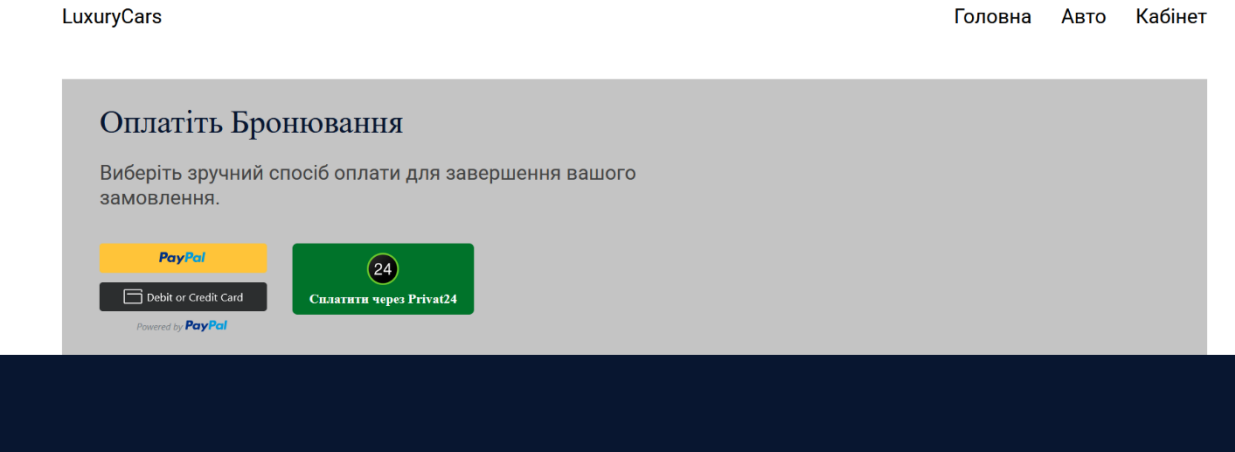


Рис. 3.10. Сторінка оплати

Використання цих сервісів також дозволяє здійснювати швидку обробку платежів із підтвердженням успішної транзакції у режимі реального часу. Крім того, інтеграція з платіжними шлюзами передбачає можливість подальшого розширення підтримки інших популярних систем оплати, що підвищує гнучкість і адаптивність вебсайту. Сторінка оплати бронювання має інтуїтивний дизайн, оптимізований для різних пристроїв, що покращує користувацький досвід і сприяє зростанню конверсії замовлень.

Інтеграція чат-бота планується у наступних версіях системи. Вона дозволить значно покращити комунікацію з клієнтами, автоматизувавши відповіді на типові запити (про ціни, терміни оренди, документи, наявність авто), допомогу в навігації сайтом, а також первинне оформлення замовлень. В основі чат-бота передбачається використання технологій штучного інтелекту, з можливістю навчання на основі історії звернень. Це знизить навантаження на службу підтримки та підвищить загальну ефективність взаємодії з клієнтами.

Інтерфейс створено з використанням HTML5 і CSS3, реалізовано адаптивну верстку за допомогою медіа-запитів. Особливу увагу приділено доступності: забезпечено достатній контраст елементів, можливість

масштабування тексту та альтернативний текст для зображень. Це робить вебсайт зручним для всіх категорій користувачів.

3.2 Особливості розгортання програмного продукту

Розгортання веб-застосунку є комплексом операцій, спрямованих на перенесення програмного продукту з локального середовища розробки у продуктивне середовище, де він стає доступним для кінцевих користувачів. У випадку вебсайту оренди люксових автомобілів цей процес має забезпечити стабільність, безпеку та ефективність функціонування системи в умовах реального використання. Передусім необхідно здійснити створення та конфігурування бази даних. На цьому етапі розробляється структура бази даних і виконується її ініціалізація згідно з вимогами предметної області. У базі даних зберігається інформація про користувачів, автомобілі, бронювання та фінансові транзакції. Важливою умовою є забезпечення належної взаємодії між інтерфейсом користувача та серверною логікою, яка виконує обробку запитів та маніпуляції з даними.

Далі слідує розміщення вебзастосунку на хостингу. Для забезпечення безперервного доступу користувачів до сервісу вебсайт розміщується на віддаленому сервері або у хмарному середовищі. У межах студентського проєкту доцільним є застосування безкоштовних або доступних хостинг-платформ, таких як Netlify, GitHub Pages [33] чи Render, які підтримують автоматизоване розгортання через механізми CI/CD і дозволяють швидко вивести проєкт у публічний доступ. Після цього виконується налаштування серверного середовища. У залежності від технологічного стеку, використаного під час розробки (наприклад, Node.js, PHP, Django або Flask), налаштовується середовище виконання, встановлюються необхідні бібліотеки та залежності, а також конфігурується серверне програмне забезпечення для стабільної та ефективної роботи вебзастосунку.

Особлива увага приділяється забезпеченню інформаційної безпеки.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки система оперує персональними даними користувачів і фінансовою інформацією, важливо впровадити ефективні засоби захисту від типових кіберзагроз, зокрема SQL-ін'єкцій, міжсайтових скриптових атак (XSS), CSRF-атак та інших векторів злому. Одним з обов'язкових заходів є впровадження HTTPS-протоколу з використанням сертифікатів безпеки, наприклад, безкоштовних рішень від Let's Encrypt, які забезпечують шифрування переданих даних і підвищують довіру користувачів до сервісу[16].

Невід'ємним компонентом розгортання є впровадження систем моніторингу та логування. Такі інструменти, як Sentry або LogRocket, дозволяють розробникам оперативно виявляти помилки, аналізувати поведінку користувачів і контролювати працездатність системи. Це сприяє підтриманню високого рівня якості сервісу та оперативному реагуванню на можливі проблеми. Крім того, важливим заходом є організація системи резервного копіювання. Регулярне створення резервних копій бази даних і файлової структури системи дає змогу зберегти цілісність інформації та забезпечити швидке відновлення функціонування сервісу у разі збоїв, помилок користувачів або атаки злоумисників.

Ще одним етапом є тестування продуктивності та навантаження. Використання інструментів Google Lighthouse, GTmetrix або PageSpeed Insights допомагає оцінити загальну швидкість завантаження сторінок, виявити вузькі місця в реалізації клієнтської або серверної частини застосунку, а також визначити напрямки оптимізації. Такий підхід дозволяє підвищити якість взаємодії користувача з системою і знизити ризики перевантаження.

У результаті, програмний продукт розгортається в хмарному середовищі з можливістю масштабування, резервного копіювання та цілодобового моніторингу. Така архітектура дозволяє забезпечити гнучкість, швидке оновлення версій, стабільну доступність сервісу та мінімізувати витрати на утримання. Для студентських або демонстраційних проєктів використання безкоштовних хостингових платформ, таких як Render, є доцільним і

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

ефективним рішенням, яке дозволяє забезпечити швидке та зручне розгортання вебзастосунків без необхідності фінансових витрат. Render надає можливість автоматизованого деплоюменту безпосередньо з репозиторіїв на GitHub або GitLab, що значно спрощує процес розміщення проєкту в мережі. Крім того, платформа підтримує середовище безперервної інтеграції та доставки (CI/CD), що дозволяє автоматизувати оновлення застосунку при зміні вихідного коду.

Render забезпечує базову конфігурацію SSL-сертифікатів, дозволяє працювати з бекенд-сервісами, має підтримку баз даних та просту систему управління середовищами (environments), що робить її універсальним інструментом навіть у рамках навчальних або дослідницьких проєктів. Платформа також надає гнучкі можливості конфігурації залежностей, перемінних середовища та резервного копіювання, що наближає її до повнофункціональних рішень, застосовуваних у реальних умовах розробки. З огляду на обмежені ресурси студентів, використання Render дозволяє уникнути витрат на оренду серверів або налаштування локального хостингу, зберігаючи при цьому повноцінну функціональність і доступність застосунку з будь-якого пристрою. Такий підхід не лише сприяє ефективному навчальному процесу, але й формує практичні навички, що є актуальними для сучасного ринку ІТ-послуг. Грамотно організований процес розгортання на платформі Render є ключовим фактором забезпечення стабільної, надійної та безпечної експлуатації вебсайту з оренди люксових автомобілів у реальних або навчальних умовах. Застосування Render у даному випадку дозволяє студентським проєктам швидко виходити в онлайн-простір, тестувати функціональність на реальних користувачах та збирати зворотний зв'язок, не витрачаючи зайвих ресурсів (рис. 3.10).

У процесі розгортання вебзастосунку також важливо враховувати аспекти керованості та масштабованості інфраструктури. Зокрема, у випадках, коли очікується зростання кількості користувачів або обсягу даних, слід передбачити можливість горизонтального масштабування сервісу. Для цього можуть застосовуватись контейнеризовані рішення на основі Docker або оркестраційні

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

платформи на кшталт Kubernetes. Такі підходи дозволяють автоматизувати розгортання, балансувати навантаження, управляти оновленнями без зупинки роботи застосунку, а також ефективно відновлювати систему після збоїв. Іншою важливою складовою є налаштування середовищ розгортання. У типовому життєвому циклі програмного забезпечення використовуються три основні середовища: розробницьке (development), тестове (staging) та продуктивне (production).

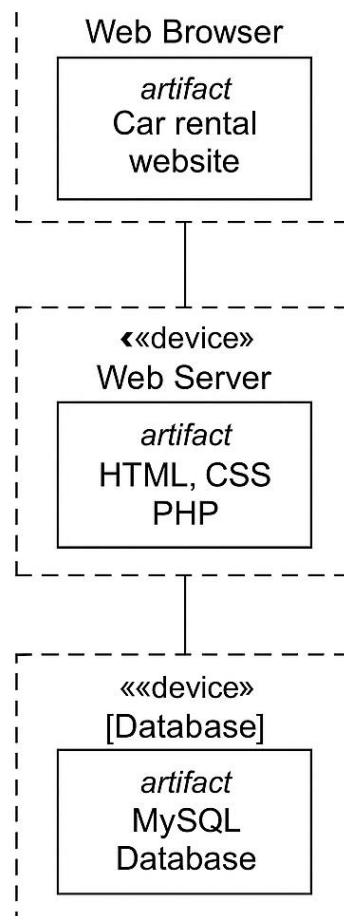


Рис. 3.11. Схема розгортання вебсайту в робочому середовищі

У розробницькому середовищі здійснюється початкове створення функціоналу, в тестовому – перевірка працездатності системи без ризику для реальних користувачів, а в продуктивному – остаточне розгортання для кінцевих користувачів. Дотримання цього підходу дозволяє уникнути неочікуваних помилок, протестувати всі оновлення перед публікацією, а також забезпечити контроль якості на кожному етапі життєвого циклу проєкту.

Особливу увагу необхідно приділяти реалізації автоматизованого розгортання (continuous deployment, CD) у поєднанні з безперервною інтеграцією (continuous integration, CI). Ці практики дозволяють інтегрувати зміни в коді, автоматично запускати тести, перевіряти відповідність стандартам коду та без втручання людини розгортати застосунок на хостинг-платформі. Наприклад, при зміні коду в репозиторії GitHub може автоматично запускатися робочий процес GitHub Actions або сервіс Vercel, який миттєво збирає нову версію сайту, перевіряє її і, у разі успішного проходження всіх перевірок, оновлює продуктивну версію вебресурсу. Це значно скорочує час на доставку нових функціональностей, покращує зручність підтримки та знижує ймовірність людських помилок.

Ще одним важливим аспектом є ведення документації щодо процесів розгортання, конфігурації інфраструктури, залежностей і процедур аварійного відновлення. Документація дозволяє не лише систематизувати знання про систему, але й забезпечити легкість передачі проєкту іншим членам команди або новим розробникам. Крім того, при появі критичних збоїв або необхідності масштабування на нове середовище, наявність чіткої документації дозволяє заощадити час та уникнути зайвих труднощів.

Не менш важливою є організація системи управління версіями застосунку. Впровадження семантичного версіонування (наприклад, формат MAJOR.MINOR.PATCH) дозволяє ефективно координувати оновлення, контролювати сумісність між модулями та фіксувати всі зміни в кодовій базі. Це також дає можливість швидко повернутися до попередньої стабільної версії у разі виникнення критичних помилок у новій версії. У контексті безпеки слід враховувати й захист API-інтерфейсів, які забезпечують обмін даними між клієнтом і сервером. Важливо обмежувати доступ до ресурсів за допомогою механізмів автентифікації та авторизації (наприклад, через JSON Web Token або OAuth2), а також реалізувати контроль допустимих запитів і обмеження частоти (rate limiting), щоб уникнути зловживань або атак типу denial-of-service (DoS).

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Доцільно також впровадити механізми аудитування дій користувачів, що дозволить виявляти підозрілу активність та реагувати на несанкціоновані дії.

Для розробника важливо також враховувати кросбраузерність і адаптивність інтерфейсу. Перед релізом застосунку проводиться тестування відображення вебсайту на різних пристроях і браузерах (Chrome, Firefox, Safari, Edge) та адаптація його до різних розмірів екранів. Це дозволяє гарантувати зручність користування сервісом незалежно від того, з якого пристрою до нього звертається користувач – зі смартфона, планшета чи ПК. Розгортання також включає підготовку до SEO-оптимізації (пошукової оптимізації), зокрема налаштування метатегів, карти сайту (sitemap.xml), файлу robots.txt, канонічних URL-адрес, що дозволяє підвищити видимість сайту у пошукових системах. Це особливо актуально для проєктів, орієнтованих на кінцевого користувача, які мають залучати трафік з відкритого Інтернету. Тому процес розгортання вебсайту є не лише технічним завершенням розробки, а й комплексною операційною задачею, яка охоплює аспекти безпеки, керованості, масштабованості, продуктивності, документування та взаємодії з кінцевим користувачем. Успішне розгортання забезпечує не лише стабільну роботу програмного продукту, а й формує позитивний користувацький досвід, який є ключовим чинником для подальшого розвитку системи.

Висновки до третього розділу

У третьому розділі випускної кваліфікаційної роботи було реалізовано практичну частину кваліфікаційної роботи, пов'язану з проєктуванням інтерфейсу користувача та процесом розгортання вебсайту з оренди автомобілів. Здійснено розробку інтерфейсних компонентів, які забезпечують інтуїтивно зрозумілу, зручну та візуально привабливу взаємодію кінцевого користувача з вебзастосунком.

Особливу увагу приділено побудові головної сторінки, каталогу автомобілів, сторінок із розширеною фільтрацією, розділу сервісів компанії,

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

футера з посиланнями та формою підписки. Крім того, спроектовано елементи персоналізованої взаємодії: панель адміністратора, особистий кабінет користувача, що дозволяють здійснювати авторизацію, перегляд бронювань та керування контентом.

Планується впровадження чат-бота на основі штучного інтелекту, який автоматизує обробку типових запитів, підвищуючи швидкість реагування та зменшуючи навантаження на службу підтримки. Це дозволить надалі покращити користувацький досвід і розширити функціональні можливості системи.

У ході дослідження було також описано особливості розгортання вебресурсу. Проведено аналіз технічних етапів перенесення програмного продукту з локального середовища до продуктивного: налаштування серверного оточення, вибір хостингу, забезпечення безпеки (включно з впровадженням HTTPS, захистом від XSS та SQL-ін'єкцій), створення резервних копій, тестування продуктивності [16].

Серед обґрунтованих платформ для розміщення сайту – Netlify, GitHub Pages, Render. Зазначено, що вибір хмарного розгортання дає змогу забезпечити масштабованість, доступність 24/7, а також мінімізувати витрати на інфраструктуру, що особливо важливо для студентських і MVP-проектів. Тому, у результаті реалізації третього розділу було досягнуто повної готовності вебінтерфейсу до публічного використання, забезпечено оптимальні умови розгортання вебзастосунку, визначено напрямки подальшого розвитку функціоналу та удосконалення платформи.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано вебсайт для оренди автомобілів, який забезпечує повний цикл взаємодії користувача з системою – від ознайомлення з автопарком до оформлення бронювання та оплати. Робота охоплює повний спектр розробки сучасного вебзастосунку – від аналізу ринку та вибору технологій до моделювання, архітектурного проектування, реалізації інтерфейсу та розгортання продукту в хмарному середовищі.

Під час дослідження проведено аналіз сучасних вебтехнологій, зокрема HTML5, CSS3, JavaScript, а також фреймворків і платформ, таких як React.js, Node.js і Express.js. Було обґрунтовано вибір технологічного стеку, що відповідає вимогам до продуктивності, масштабованості, безпеки та підтримки інтерактивності вебплатформи. Також проведено порівняльний аналіз провідних сервісів з оренди авто (Hertz, Avis, Europcar) для визначення конкурентних переваг майбутнього продукту.

На основі результатів аналізу було розроблено повний набір UML-діаграм, які описують функціональність, структуру, алгоритми та взаємодію між компонентами. Зокрема, діаграми варіантів використання дозволили окреслити ролі клієнта й адміністратора, а діаграми класів і послідовності – формалізували логіку функціонування платформи.

Було реалізовано архітектуру вебсайту з чітким поділом на фронтенд і бекенд, визначено основні модулі системи: авторизації, бронювання, управління транспортними засобами, платежів, адміністративного контролю, а також особистий кабінет користувача. Окрему увагу приділено безпеці даних, зокрема впровадженню захисту від XSS- і SQL-ін'єкцій, використанню HTTPS та інтеграції з платіжними шлюзами. В рамках реалізації інтерфейсу користувача було створено повноцінну візуальну структуру сайту: головна сторінка, каталог автомобілів із фільтрацією, інформаційні блоки, форма замовлення, особистий кабінет та панель адміністратора. Запропоновано план впровадження додаткових

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

функцій, зокрема інтеграції чат-бота для автоматизованої взаємодії з користувачами.

Було також описано процес розгортання продукту на хмарному хостингу, включно з етапами налаштування середовища, безпеки, моніторингу та резервного копіювання. Обрано платформи, що дозволяють безкоштовне публічне розміщення сайту, з можливістю масштабування у разі зростання аудиторії.

У результаті поставлені завдання було повністю реалізовано. Сформовано функціональний, безпечний і масштабований вебресурс, який відповідає сучасним вимогам як з боку користувача, так і з боку розробника. Практична цінність розробки полягає в універсальності, гнучкості та можливості подальшого вдосконалення системи, включно з розширенням автопарку, підключенням нових сервісів та підтримкою багатомовного інтерфейсу. Отримані результати можуть бути використані як основа для розгортання повноцінної комерційної платформи з оренди авто.

Отже, поставлені в роботі цілі досягнуто, виконано всі ключові завдання, що підтверджується результатами тестування та демонстрації функціоналу. Розроблений продукт може бути впроваджено у комерційне використання з подальшим розширенням функціоналу та інтеграцією з іншими сервісами.

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Найновіші технології веб-розробки 2025: як створюються сучасні сайти[Електронний ресурс].Wezom.URL:<https://wezom.com.ua/ua/blog/naynovishi-tehnologiyi-veb-rozrobki-2025-yak-stvoryuyutsya-suchasni-sayti>(дата звернення: 16.05.2025).
2. Smashing Magazine. Best Practices for Modern Web Design [Електронний ресурс].URL: <https://www.smashingmagazine.com> (дата звернення: 16.05.2025).
3. Duckett J. HTML and CSS: Design and Build Websites.Chichester : John Wiley and Sons Ltd.
4. Farley D. Modern Software Engineering: Doing What Works to Build Better Software Faster. Pearson Education, Limited, 2022.
5. Курінний С. Розробка веб-сайтів для початківців: HTML, CSS, JavaScript[Електронний ресурс]. 2023. URL: https://chtyvo.org.ua/authors/Kurinnyi_Serhii/Rozrobka_vebsaitiv_dlja_pochatkivtsiv_HTML_CSS_JavaScript/ (дата звернення: 16.05.2025).
6. HTML Living Standard/ WHATWG[Електронний ресурс]. 2025. URL: <https://html.spec.whatwg.org/> (дата звернення: 16.05.2025).
7. W3Schools. HTML5 Tutorial [Електронний ресурс].URL: <https://www.w3schools.com/html/>(дата звернення: 16.05.2025).
8. Daber J. HTML, CSS, Website Development and Design.London : Exemo, 2022.480 p.
9. CSS-Tricks. Актуальні техніки адаптивної верстки [Електронний ресурс]. URL: <https://css-tricks.com> (дата звернення: 16.05.2025).
10. Ніколс Р. HTML5 і CSS3. Розробка сучасних веб-сайтів. Київ : Фабула, 2021. 432 с.
11. W3Schools. CSS3 Tutorial [Електронний ресурс]. URL: <https://www.w3schools.com/css/> (дата звернення: 16.05.2025).
12. Google Developers. Web Fundamentals [Електронний ресурс]. URL:

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

<https://developers.google.com/web/fundamentals> (дата звернення: 16.05.2025).

13. ТОП технологій для розробки сайту у 2025 році [Електронний ресурс]. Megasite.ua, 2025. URL: <https://megasite.ua/blog/top-tehnologiy-dlya-rozrobki-saytu-2025> (дата звернення: 16.05.2025).

14. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol : O'Reilly Media, 2020. 1150 p.

15. WebAIM. Accessibility Principles [Електронний ресурс]. URL: <https://webaim.org/principles/> (дата звернення: 16.05.2025).

16. Google Search Central. SEO Starter Guide [Електронний ресурс]. 2023. URL: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide> (дата звернення: 16.05.2025).

17. Rentalcars.com – Платформа онлайн-бронювання авто [Електронний ресурс]. URL: <https://www.rentalcars.com> (дата звернення: 16.05.2025).

18. Sixt.com. Офіційний сайт. Аналіз UX/UI [Електронний ресурс]. 2024. URL: <https://www.sixt.com> (дата звернення: 16.05.2025).

19. 7cars.ua. Офіційний сайт. Аналіз UX/UI [Електронний ресурс]. 2024. URL: <https://7cars.ua> (дата звернення: 16.05.2025).

20. Hertz. Офіційний сайт компанії з оренди автомобілів [Електронний ресурс]. URL: <https://www.hertz.com> (дата звернення: 16.05.2025).

21. Avis Car Rental. Офіційний сайт [Електронний ресурс]. URL: <https://www.avis.com> (дата звернення: 16.05.2025).

22. Eurocar. Офіційний сайт [Електронний ресурс]. URL: <https://www.eurocar.ua> (дата звернення: 16.05.2025).

23. Freeman A., Robson E. Head First HTML and CSS. 2nd ed. Sebastopol : O'Reilly Media, 2022. 768 p.

24. MDN Web Docs. CSS Reference [Електронний ресурс]. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 16.05.2025).

25. MDN Web Docs. HTML Reference [Електронний ресурс]. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 16.05.2025).

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

26. CSS: Cascading Style Sheets / W3C [Електронний ресурс]. URL: <https://www.w3.org/Style/CSS/> (дата звернення: 16.05.2025).

27. Розробка сайту з оренди автомобілів [Електронний ресурс]. Stfalcon.com, 2024. URL: <https://stfalcon.com/uk/blog/post/rent-car-site> (дата звернення: 16.05.2025).

28. Закон України «Про захист персональних даних», № 2297-VI від 01.06.2010 р. Відомості Верховної Ради України. 2010. № 34. С. 481.

29. ISO/IEC 27001:2022. Інформаційна технологія. Методи захисту. Системи управління інформаційною безпекою. Вимоги. Женева : ISO, 2022.

30. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016. General Data Protection Regulation (GDPR). Official Journal of the European Union. 2016.

31. Webflow University. Responsive Web Design [Електронний ресурс]. URL: <https://university.webflow.com/lesson/responsive-web-design> (дата звернення: 16.05.2025).

32. Figma. Design System Best Practices [Електронний ресурс]. URL: <https://www.figma.com/resources/learn-design/design-systems/> (дата звернення: 16.05.2025).

33. GitHub Pages Documentation. Hosting static websites [Електронний ресурс]. URL: <https://docs.github.com/en/pages> (дата звернення: 16.05.2025).

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М				64
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг А.1 – JavaScript-код фільтрації автомобілів

```

const allCars = [
  {
    id: 1,
    brand: "Audi",
    model: "RS7",
    fuel: "Petrol",
    price: 60000,
    image: "../assets/images/audi-rs7.jpg",
  },
  {
    id: 2,
    brand: "BMW",
    model: "X5",
    fuel: "Diesel",
    price: 45000,
    image: "../assets/images/bmw-x5.jpg",
  },
  {
    id: 3,
    brand: "Ferrari",
    model: "SF90",
    fuel: "Petrol",
    price: 450000,
    image: "../assets/images/ferrari-sf90.jpg",
  },
];

function renderCars(cars, targetId = "carList") {
  const container = document.getElementById(targetId);
  if (!container) return;

```

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

container.innerHTML = "";
cars.forEach((car) => {
    const div = document.createElement("div");
    div.className = "car-card";

    div.innerHTML = `
        
        <h3>${car.brand} ${car.model}</h3>
        <p>Тип палива: ${car.fuel}</p>
        <p>Ціна: ${car.price.toLocaleString()}</p>
        <button onclick="openBooking(${car.id})">Бронювати</button>
    `;

    container.appendChild(div);
});
}

document
.getElementById("filterForm")
?.addEventListener("submit", function (e) {
    e.preventDefault();

    const { brand, fuel, priceMax } = Object.fromEntries(new
FormData(this));

    const filtered = allCars.filter(
        (c) =>
            (!brand || c.brand === brand) &&
            (!fuel || c.fuel === fuel) &&
            (!priceMax || c.price <= +priceMax)
    );

    renderCars(filtered);
});

```

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

function openBooking(carId) {
    const modal = document.getElementById("bookingModal");
    const bookingForm = document.getElementById("bookingForm");
    if (!modal || !bookingForm) return;

    bookingForm.carId.value = carId;
    modal.classList.remove("hidden");
}

document.getElementById("closeModal")?.addEventListener("click", ()
=> {
    document.getElementById("bookingModal")?.classList.add("hidden");
});

renderCars(allCars);

```

		Левченко М.С.			ІПЗ.КР.Б.-121-25-ІПЗ	Арк.
		Лобанчикова Н.М.				67
Змн.	Арк.	№ докум.	Підпис	Дата		