

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:
«ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ ТА БРОНЮВАННЯ
ПАРКОМІСЦЬ»

Виконав: студент 4 курсу, групи 2ІСТ-216
спеціальності 126 – «Інформаційні
системи та технології»

 Анна ШАПОВАЛ

Керівник: к.т.н., доц. каф. САІТ

 Сергій ЖУКОВ

«04» 06 2025 р.

Рецензент: к.т.н., ас. каф. КН

 Ілона БОГАЧ

«08» 06 2025 р.

Допущено до захисту

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

«06» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

А.Мокін д.т.н., проф. Віталій МОКІН

«28» 03 2025 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Шаповал Анні Сергіївні

1. Тема роботи: «Інформаційна система пошуку та бронювання паркомісць»
керівник роботи: Жуков С. О., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «20» 03 2025 року № 97
2. Термін подання студентом роботи 31.05.25
3. Вихідні дані до роботи:
 - 1) Набір даних про парковки та паркомісця у м.Львів:
https://www.google.com/maps/d/u/0/viewer?mid=1H0s277ON8BZqromCyEy7kiaNlbww_QQ&femb=1&ll=49.85026932274493%2C24.02263754999997&z=14
4. Зміст текстової частини:
 - 1) Аналіз об'єкта дослідження.
 - 2) ІТ-інфраструктура інформаційної системи для пошуку та бронювання паркомісць.
 - 3) Проектування та розроблення інформаційної системи.
5. Перелік ілюстративного матеріалу:
 - 1) ER діаграма бази даних системи бронювання паркомісць;
 - 2) Блок-схема роботи алгоритму бронювання паркомісць;
 - 3) Діаграма розгортання веб-системи пошуку та бронювання паркомісць;
 - 4) Діаграми USE CASE інформаційної системи;
 - 5) Інтерфейс інформаційної системи.

6. Консультанти розділів роботи:

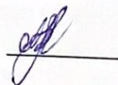
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 28.03.25

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз об'єкта дослідження	01.04	15.04	вик
2	ІТ-інфраструктура інформаційної системи для пошуку та бронювання паркомісць	15.04	30.04	вик
3	Проектування та розроблення інформаційної системи для пошуку та бронювання паркомісць	01.05	15.05	вик
4	Оформлення матеріалів до захисту БКР	15.05	30.05	вик

Студент



Анна ШАПОВАЛ

Керівник роботи



Сергій ЖУКОВ

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 50 сторінок формату А4, на яких є 32 рисунка, список використаних джерел містить 20 найменувань.

В роботі розроблено інформаційну систему пошуку та бронювання паркомісць.

Розроблено веб-систему з використанням мов програмування HTML, CSS, PHP, JavaScript. Розроблено зручний та адаптивний дизайн, функціональну частину та базу даних для взаємодії з інформацією про наявні паркомісця. Розроблена інформаційна система надає користувачам можливість пошуку, фільтрації та бронюванню паркомісць.

Ключові слова: інформаційна система, веб-сайт, JavaScript, SQL, PHP, паркування, бронювання, база даних, веб-розробка.

ABSTRACT

The bachelor's qualification work consists of 50 pages of A4 format, on which there are 32 figures, the list of used sources contains 20 names.

The work developed an information system for searching and booking parking spaces.

A web system was developed using the programming languages HTML, CSS, PHP, JavaScript. A convenient and adaptive design, a functional part and a database for interacting with information about available parking spaces were developed. The developed information system provides users with the ability to search, filter and book parking spaces.

Keywords: information system, website, JavaScript, SQL, PHP, parking, reservation, database, web development.

ЗМІСТ

ВСТУП	4
1. АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ	6
1.1 Аналіз предметної області.....	6
1.2 Аналіз аналогічних систем пошуку та бронювання паркомісць.....	10
1.3 Постановка задачі.....	16
1.4 Висновки	19
2. ІТ-ІНФРАСТРУКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПОШУКУ ТА БРОНЮВАННЯ ПАРКОМІСЦЬ	20
2.1 Вибір оптимальних технологій для реалізації	20
2.2 Формування набору даних для роботи веб-системи	25
2.3 ІТ-інфраструктура та архітектура інформаційної системи.....	28
2.4 Висновки	31
3. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ..	32
3.1 Розроблення алгоритму функціонування системи	32
3.2 Розроблення інформаційної системи	36
3.3 Розроблення панелі адміністратора	47
3.4 Висновки	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
Додаток А (обов'язковий). Технічне завдання	60
Додаток Б (обов'язковий). Протокол перевірки кваліфікаційної роботи	62
Додаток В (довідниковий). Фрагмент лістингу програми	63
Додаток Г (обов'язковий). Ілюстративна частина	73

ВСТУП

Актуальність теми. У сучасних урбанізованих центрах кількість автомобілів зростає стрімкими темпами, що створює серйозні виклики для міської інфраструктури, особливо в контексті організації паркування. Дефіцит паркувальних місць, особливо в центральних районах мегаполісів, де концентрація транспортних засобів є найбільшою, стає значною проблемою як для окремих водіїв, так і для міської громади загалом. Щоденні пошуки вільного паркомісця призводять до втрати часу, підвищення витрат енергії, заторів на дорогах, зростання викидів шкідливих речовин і погіршення екологічної ситуації. Це також негативно впливає на емоційний стан водіїв, підвищуючи рівень стресу. Одним із найперспективніших рішень є створення інформаційної системи для автоматизованого пошуку та бронювання паркомісць, яка забезпечить доступ до актуальної інформації про вільні місця, їх розташування, вартість і можливість онлайн-оплати.

Мета і задачі дослідження. Метою дослідження є забезпечення швидкості та зручності процесу пошуку та бронювання паркомісць, скоротивши час на пошук вільних місць, завдяки розробці інформаційної системи з інтерактивною фільтрацією, моніторингом завантаженості та системою реєстрації користувачів.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- провести аналіз проблематики паркування та існуючих систем управління паркомісцями;
- здійснити вибір оптимальних інформаційних технологій та ІТ-інфраструктури;
- розробити інформаційну систему для управління паркомісцями та бронюваннями.

Об'єктом дослідження є процес створення інформаційної системи для пошуку та бронювання паркомісць.

Предметом дослідження є методи і програмні засоби створення інформаційної системи для автоматизації пошуку та бронювання паркомісць.

Публікації. За результати роботи було зроблено дві доповіді. Перша доповідь на тему «Моделювання інформаційної системи пошуку та бронювання паркомісць: оптимізація паркування» на LIV Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2025) з публікацією тез [1]. Друга доповідь на тему «Інформаційна система пошуку та бронювання паркомісць» на міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» з публікацією тез [2].

1. АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Паркувальне місце — це спеціально облаштована ділянка, призначена для тимчасового або тривалого зберігання транспортного засобу. Паркувальні місця можуть бути розташовані вздовж проїзної частини, у дворах, на відкритих стоянках, у багаторівневих або підземних паркінгах. Їх розміри, розмітка, напрямок заїзду та інші технічні характеристики регулюються відповідними державними стандартами, що враховують тип транспортного засобу, безпеку руху та зручність використання [3].

Паркувальні місця та системи парковок є невід'ємною частиною сучасної міської інфраструктури. З розвитком автомобільного транспорту проблема паркування стала одним із найбільш актуальних питань у містах усього світу. Зростаюча кількість автомобілів призводить до дефіциту вільного простору для їх зберігання, що ускладнює дорожній рух, погіршує екологічну ситуацію та негативно впливає на комфорт мешканців. Водночас правильне планування, впровадження інноваційних рішень та ефективне управління паркувальними ресурсами здатне не лише поліпшити умови для автомобілістів, а й сприяти гармонійному розвитку міського середовища [4].

У світі існує велика кількість типів парковок, які відрізняються за формою власності, способом доступу, принципами роботи та технічним оснащенням. Вони можуть бути безкоштовними або платними, публічними чи приватними, автоматизованими чи звичайними. Поширеним варіантом є наземні стоянки, які використовуються як у житлових районах, так і поблизу комерційних центрів. Такі парковки прості в будівництві та обслуговуванні, проте займають значну площу, що є недоліком в умовах обмеженого простору [5].

У великих містах дедалі популярнішими стають багаторівневі парковки — це спеціальні будівлі, що дозволяють розміщувати автомобілі на декількох

поверхах, значно збільшуючи місткість стоянки без розширення її площі. Окрім того, існують підземні паркінги, які дозволяють зберігати автомобілі під будівлями або площами, звільняючи поверхню для пішоходів та озеленення.

Іншим цікавим типом є автоматизовані системи паркування, які здатні самостійно приймати автомобіль, розміщувати його на платформі або в комірці, а потім повертати власнику за запитом. Такі системи використовують ліфти, транспортери та роботизовані механізми, що мінімізує участь людини та значно економить простір. Незважаючи на високу вартість впровадження, автоматизовані паркінги стають дедалі поширенішими в мегаполісах із щільною забудовою, де вільне місце під забудову обмежене [6].

Платні парковки відіграють важливу роль в організації транспортного руху та регулюванні попиту на паркувальні місця. За допомогою цінової політики муніципалітети можуть впливати на рішення водіїв — наприклад, стимулювати пересування громадським транспортом або обмежувати в'їзд у центр міста. Ціна за паркування може змінюватися залежно від часу доби, дня тижня або завантаженості стоянки, що дозволяє досягати більш рівномірного розподілу транспортних потоків. Сучасні системи паркування передбачають використання електронних способів оплати — через мобільні додатки, банківські картки, SMS або спеціальні паркомати. Водії можуть у реальному часі бачити наявність вільних місць, бронювати їх наперед та отримувати повідомлення про закінчення оплаченого часу [7].

Зі зростанням популярності цифрових технологій дедалі частіше впроваджуються інтелектуальні системи керування паркуванням, які використовують камери, датчики, програмне забезпечення та алгоритми штучного інтелекту для контролю, обліку та аналізу даних. Такі системи здатні автоматично фіксувати заїзд та виїзд автомобілів, виявляти порушення правил паркування, розраховувати вартість послуг та формувати звіти. Вони також допомагають міським службам у прийнятті управлінських рішень, дозволяючи прогнозувати навантаження, оптимізувати розташування стоянок та коригувати тарифну політику [8].

У багатьох країнах спостерігається тенденція до впровадження екологічно орієнтованих рішень у сфері паркування. Наприклад, створюються спеціальні паркувальні місця для електромобілів із можливістю зарядки, впроваджуються зони низьких викидів, де можуть паркуватися лише транспортні засоби, що відповідають певним екологічним стандартам. Також з'являються концепції «зеленої парковки», де використовуються екологічно чисті матеріали, насадження, системи збору дощової води та сонячні панелі. У поєднанні з розвитком громадського транспорту та інфраструктури для велосипедів це дозволяє формувати більш дружнє до довкілля міське середовище [9].

У країнах із розвинутою міською культурою паркування розглядається не лише як інженерна проблема, а й як частина загальної транспортної політики. Наприклад, у Нідерландах чи Японії діють програми, які заохочують мешканців відмовлятися від особистого автомобіля на користь громадського транспорту, оренди або каршерингу. Одним із інструментів є створення паркінгів на околицях міста з пересадкою на метро або трамвай — так звані «перехоплюючі» парковки. Вони дозволяють зменшити кількість автомобілів у центрі міста, знизити рівень заторів і покращити якість повітря. Також у багатьох мегаполісах вводиться обмеження на кількість виданих паркувальних дозволів, особливо для жителів центральних районів, що сприяє зменшенню кількості автомобілів та підвищенню ефективності використання міського простору [10].

Окрему увагу слід приділяти питанням безпеки паркування. Паркувальні майданчики мають бути обладнані системами відеонагляду, освітленням, шлагбаумами, а іноді й охороною. Особливо це стосується великих багаторівневих паркінгів, де ризик викрадення або пошкодження автомобіля вищий. Водночас зручність користування паркуванням передбачає не лише технічне обладнання, а й доступність для людей з інвалідністю, наявність навігації, зрозумілої розмітки.

Паркувальна інфраструктура повинна враховувати різноманітні потреби користувачів. Наприклад, поблизу шкіл, лікарень та торговельних центрів доцільно облаштовувати місця для короткострокового паркування, які дозволяють швидко висадити або забрати пасажирів. Для мешканців житлових районів важливими є довгострокові парковки з можливістю залишати автомобіль на ніч або на декілька днів. У місцях великого скупчення туристів, аеропортах та вокзалах популярністю користуються парковки з по хвилинною оплатою або абонементом. Також існують спеціалізовані паркувальні простори для вантажного транспорту, автобусів, мотоциклів, а останнім часом і для самокатів та велосипедів [11].

Зрозуміло, що ефективна система паркування потребує не лише технічних засобів, а й чіткої нормативної бази. Законодавство має визначати правила паркування, зони дії, обов'язки водіїв, повноваження контролюючих органів, санкції за порушення. У багатьох країнах вже впроваджено системи фото- і відеофіксації порушень, які автоматично надсилають штрафи. Це підвищує дисципліну водіїв, зменшує кількість конфліктних ситуацій і дозволяє ефективніше використовувати обмежений паркувальний простір. У той же час важливо, щоб контроль не перетворювався на інструмент репресій або джерело прибутку для муніципалітету — головним завданням залишається створення зручних умов для всіх учасників дорожнього руху.

На рівні міського планування проблема паркування має розглядатися в контексті загальної мобільності. Якщо планується будівництво нового району, бізнес-центру чи торговельного комплексу, про паркування слід дбати ще на етапі проектування. Інакше існує ризик перевантаження навколишньої інфраструктури, виникнення стихійного паркування, конфліктів з пішоходами та мешканцями. Деякі міста вводять нормативи, які зобов'язують забудовника забезпечити певну кількість паркомісць у залежності від площі об'єкта чи кількості квартир. Водночас у густонаселених зонах все частіше відходять від обов'язкових вимог на користь гнучких підходів, коли паркування

розміщується централізовано, а акцент робиться на громадський транспорт, мікро мобільність та пішохідну доступність.

Новим викликом є розвиток мобільності за запитом, зокрема таксі, каршеринг та орендні сервіси. Вони зменшують потребу в особистому авто, але водночас створюють нові виклики — необхідність короткочасного паркування в зручних місцях, завантаження й розвантаження пасажирів, технічне обслуговування автопарку. У деяких містах вже впроваджено спеціальні зони для стоянки таксі та каршерингів, а також системи ідентифікації таких автомобілів через номерні знаки. Це дозволяє уникати безладу та підвищує ефективність міського простору.

1.2 Аналіз аналогічних систем пошуку та бронювання паркомісць

Parkopedia є однією з наймасштабніших та найвідоміших у світі систем пошуку та бронювання паркувальних місць. Вона була заснована у Великій Британії у 2007 році з метою вирішити глобальну проблему пошуку місця для паркування в міських умовах. Назва сервісу є поєднанням слів "parking" і "encyclopedia", що символізує її амбіції стати універсальним джерелом інформації про паркування. Сьогодні Parkopedia охоплює понад 70 мільйонів паркомісць у більш ніж 15 тисячах міст по всьому світу, включаючи Європу, Північну Америку, Азію та Австралію. Ця система інтегрується не лише у вигляді мобільного додатка чи веб-сайту, але й у бортові системи автомобілів таких брендів, як BMW, Mercedes-Benz, Audi, Jaguar та багато інших. Водії можуть безпосередньо з панелі приладів отримувати дані про вільні місця поблизу, умови паркування, тарифи, години роботи паркінгів, наявність зарядних станцій для електромобілів та навіть інформацію про рівень заповненості місць у реальному часі.

На рисунку 1.1 показано головну сторінку веб-сайту системи Parkopedia.

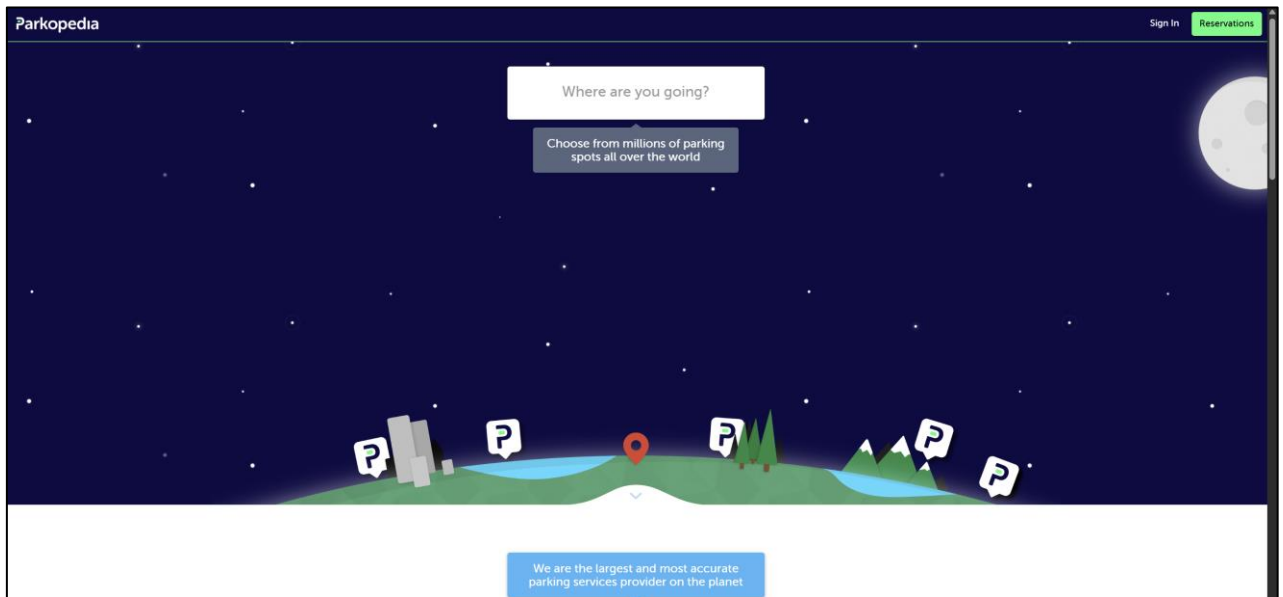


Рисунок 1.1 – Головна сторінка Parkopedia

Завдяки активному збору даних із тисяч сенсорів, відеокамер, інформаційних систем міст, а також завдяки співпраці з операторами паркінгів і муніципальними органами, Parkopedia надає точні та актуальні відомості. У деяких країнах ця платформа також дозволяє здійснювати бронювання місць заздалегідь та оплачувати паркування через додаток або вбудовану платіжну систему в авто. Завдяки штучному інтелекту та машинному навчанню сервіс навчився прогнозувати ймовірність наявності вільного місця на основі часу доби, дня тижня, історичних даних та поточних умов трафіку [12].

ParkMe є ще однією розвиненою платформою, яка спеціалізується на наданні детальної інформації про паркувальні місця. Вона належить компанії INRIX, яка займається аналітикою транспортних потоків. ParkMe має велику базу даних, що охоплює велику кількість міст по всьому світу. Ця система пропонує користувачам інтерфейс для пошуку паркінгів за різними параметрами, включаючи відстань, вартість, години роботи, тип паркування (вуличне або гаражне) та інші характеристики. ParkMe використовує прогностичні моделі, що дозволяють оцінити ймовірність наявності вільних місць у конкретний момент часу, навіть якщо для цієї парковки не встановлені

сенсори. Це стало можливим завдяки обробці великого обсягу історичних даних, даних трафіку, сезонних коливань та звичок водіїв у певній місцевості.

На рисунку 1.2 показано принцип роботи мобільного системи ParkMe.

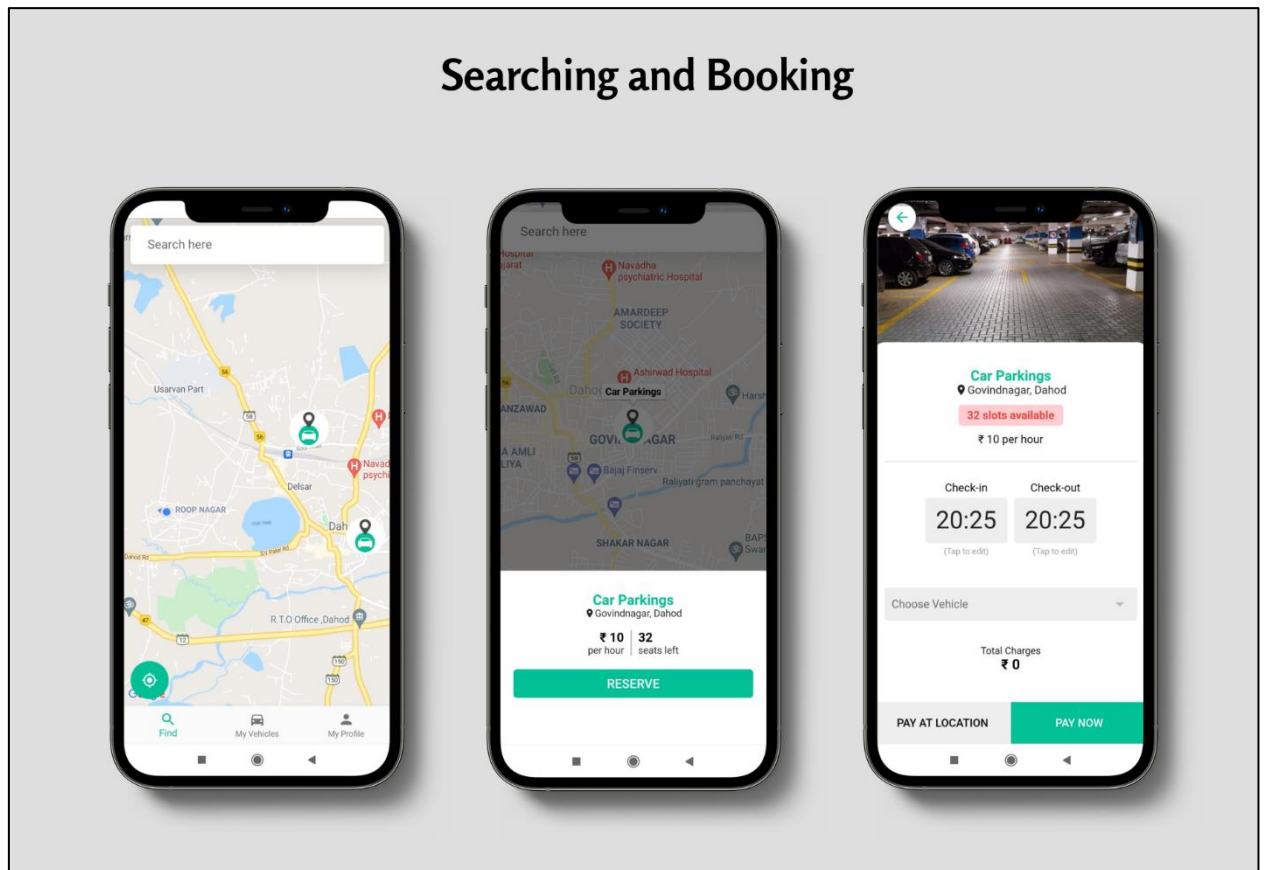


Рисунок 1.2 – Принцип роботи ParkMe

ParkMe отримує свої дані з різноманітних джерел: оператори парковок надають інформацію про свої об'єкти, міські адміністрації діляться відкритими даними про муніципальні паркінги, а користувачі також мають можливість доповнювати карту, додаючи нові місця або актуалізуючи існуючі. У деяких містах ParkMe підтримує можливість бронювання місця наперед, а також інтеграцію з навігаційними додатками для прокладання оптимального маршруту. Платформа також враховує знижки, спеціальні тарифи для мешканців, години, коли паркування є безкоштовним, і відображає все це у зручному вигляді. Завдяки зручному інтерфейсу та точності, ParkMe стала популярною серед водіїв у США, Канаді, Європі та Азії [13].

EasyPark — це європейська система, яка спеціалізується на мобільному управлінні паркуванням. Її головна відмінність полягає у фокусі не лише на пошуку, але й на повному контролі процесу паркування. Заснована у Швеції, ця система сьогодні охоплює понад 2 200 міст у таких країнах, як Швеція, Норвегія, Данія, Німеччина, Італія, Іспанія, Австрія, Швейцарія та багато інших. Користувачі через мобільний додаток можуть знайти доступне місце, переглянути всю необхідну інформацію про нього, а також розпочати сесію паркування. Унікальність EasyPark полягає у можливості дистанційного продовження або завершення сесії — користувач, наприклад, може вийти з магазину раніше і закінчити паркування, не переплачуючи за зайвий час [14].

На рисунку 1.3 показано сторінку завантаження системи EasyPark.

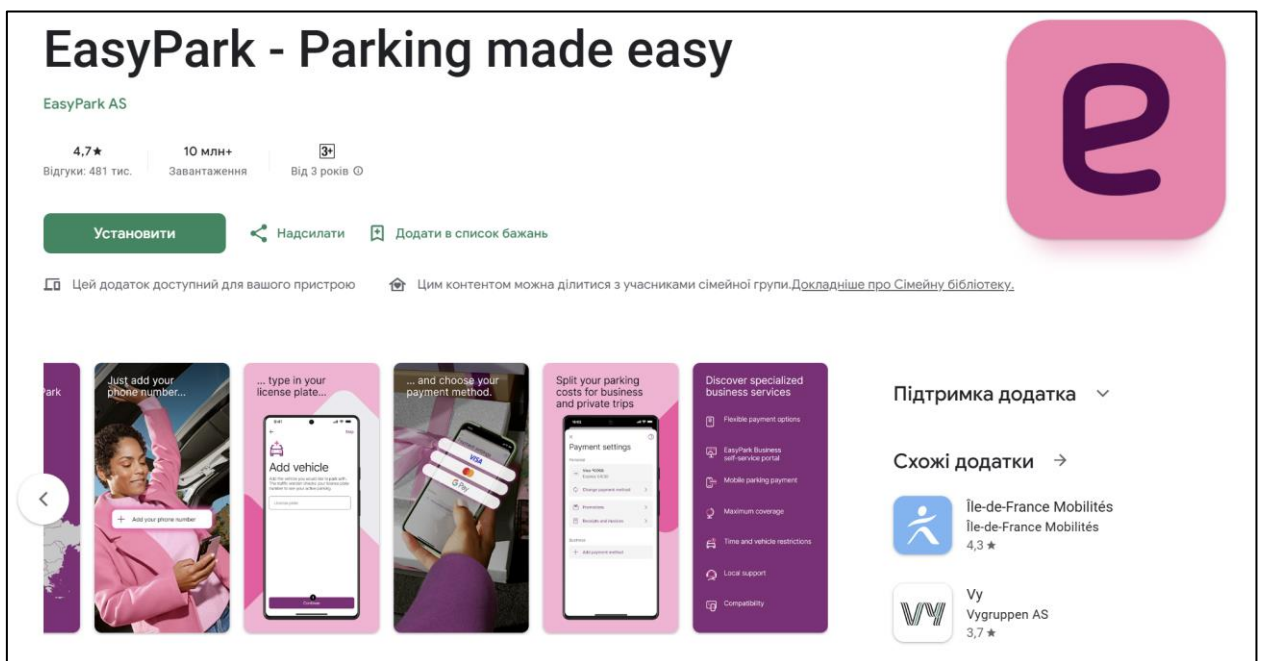


Рисунок 1.3 – Сторінка завантаження системи EasyPark

Оплата здійснюється безпосередньо з системи через банківські картки або електронні гаманці. EasyPark також інтегрується з системами автопарків великих підприємств, міських адміністрацій та навіть з деякими новими автомобілями. Сервіс надає візуалізацію зон паркування на карті, дозволяє фільтрувати їх за ціною або тривалістю, а також дає змогу швидко знайти

місце відповідно до заданих параметрів. Крім того, EasyPark має спеціальні інструменти для бізнесу — компанії можуть контролювати паркування своїх автопарків, вести звітність, спрощувати адміністрування витрат на паркування співробітників. У деяких країнах система також взаємодіє з паркувальними камерами та дозволяє автоматично завершувати сесію, коли авто виїжджає з місця.

SpotHero — популярна американська платформа для бронювання паркомісць у великих містах США та Канади. Вона була заснована в 2011 році та швидко стала відомою завдяки зручному інтерфейсу, можливості заздалегідь бронювати місця та гнучкій ціновій політиці. SpotHero дозволяє водіям знайти доступне місце у великих містах, таких як Нью-Йорк, Чикаго, Лос-Анджелес, Сан-Франциско, Вашингтон, Торонто та інших. На відміну від деяких систем, які орієнтуються здебільшого на муніципальні паркінги, SpotHero активно співпрацює з приватними стоянками, гаражами та навіть окремими власниками приватних паркомісць. Таким чином, користувач отримує ширший вибір і часто може знайти місце ближче до потрібної локації за нижчою ціною.

На рисунку 1.4 показано систему SpotHero.

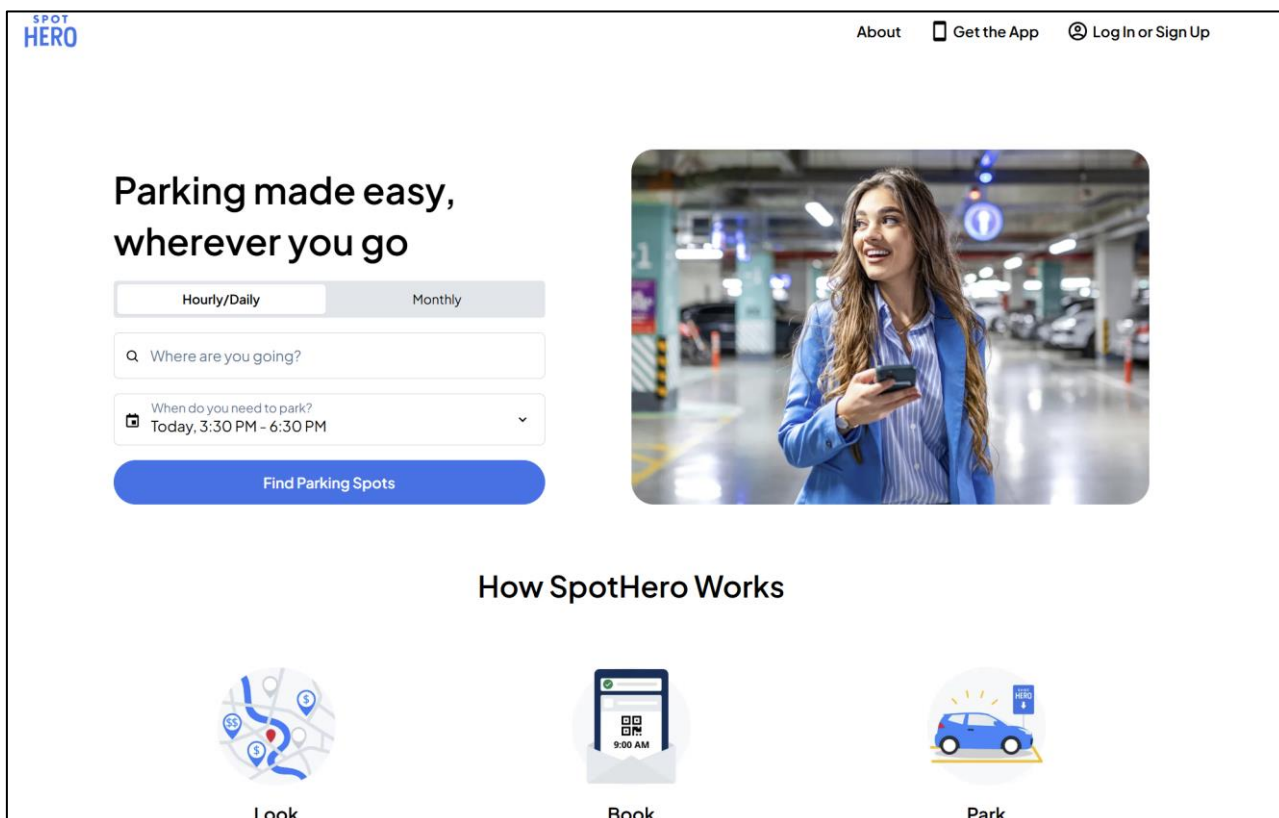


Рисунок 1.4 – Система пошуку та бронювання паркомісць SpotHero

Основна функція SpotHero — це попереднє бронювання. Водій обирає дату та час, бачить доступні варіанти, порівнює ціни та умови, і одразу оплачує послугу. Після цього система видає QR-код або цифрову перепустку, яка сканується на в'їзді до паркінгу. Багато об'єктів працюють повністю в автоматичному режимі, без необхідності контакту з персоналом. SpotHero також пропонує вигідні пропозиції для постійних користувачів, має систему знижок та індивідуальних тарифів. У деяких містах платформа тестує інтеграцію з електромобілями та можливість бронювання зарядних станцій разом із місцем [15].

Усі ці системи мають як сильні сторони, так і певні обмеження. Parkopedia є найбільш глобальною, з великою кількістю інтеграцій, але не завжди дозволяє забронювати місце заздалегідь. ParkMe зосереджується на аналітиці й прогнозуванні, але не у всіх країнах підтримує оплату або бронювання. EasyPark вирізняється зручністю та фокусом на повному

контролі паркування, проте обмежена переважно європейським ринком. SpotHero спеціалізується на заздалегідь заброньованих паркувальних місцях і зручна у використанні в Північній Америці, але поза її межами практично не представлена.

Порівняльно можна виділити такі плюси й мінуси: Parkopedia має найширшу географію покриття, вбудовується в системи авто та надає багато деталей, але бронювання доступне не всюди. ParkMe пропонує точні аналітичні оцінки й інформативність, однак може мати затримки в оновленні даних. EasyPark дозволяє повністю контролювати процес паркування через смартфон, включаючи оплату та дистанційне завершення сесії, але вимагає реєстрації та не скрізь підтримується. SpotHero зручний для бронювання наперед і економії часу, проте його сервіс обмежений США та Канадою. В загальному, невелика кількість подібних систем пристосовані до України.

1.3 Постановка задачі

Мета проєкту полягає у розробці веб-системи для пошуку та бронювання паркомісць в Україні, що забезпечує зручний пошук, візуалізацію та резервування паркомісць для користувачів. Система має надавати актуальну інформацію про доступні парковки, їх розташування, зони, ціни, кількість вільних місць та можливість бронювання конкретного паркомісця з урахуванням часу і тривалості. Проєкт спрямований на підвищення ефективності використання паркувальних зон, зменшення часу пошуку вільних місць та забезпечення комфортного досвіду для водіїв.

Для реалізації системи необхідно виконати наступні задачі:

1. Проєктування структури бази даних:
 - Розробити реляційну базу даних для зберігання інформації про парковки, окремі паркомісця та бронювання;

- Створити таблицю для зберігання даних про парковки (ідентифікатор, адреса, зона, сектор, ціна, загальна кількість місць, вільні місця, координати для карти);
- Створити таблицю для управління окремими паркомісцями (ідентифікатор, номер місця, ідентифікатор парковки, статус зайнятості, час закінчення бронювання);
- Створити таблицю для зберігання інформації про бронювання (ідентифікатор, ідентифікатор парковки, ідентифікатор паркомісця, ім'я користувача, номер телефону, час бронювання, тривалість);
- Забезпечити унікальність паркомісць через ключ UNIQUE та зв'язки між таблицями через зовнішні ключі.

2. Розробка серверної логіки:

- Налаштувати підключення до бази даних з безпечним зберіганням облікових даних;
- Реалізувати обробку запитів для відображення списку парковок, їхньої інформації та ініціалізації карти;
- Розробити функції для відображення плиток паркомісць, обробки бронювання з транзакційною логікою (оновлення статусу місця, зменшення вільних місць, додавання бронювання);
- Створити функції для асинхронного завантаження даних про паркомісця;
- Забезпечити валідацію вхідних даних (GET і POST) для захисту від SQL-ін'єкцій і некоректних даних (наприклад, формат телефону, позитивна тривалість).

3. Розробка клієнтської частини:

- Реалізувати інтерактивну карту з використанням бібліотеки Leaflet для відображення маркерів парковок із координатами (latitude, longitude);

- Додати спливаючі вікна (popups) до маркерів із інформацією про парковку (адреса, зона, сектор, ціна, вільні місця) та кнопкою «Забронювати», що перенаправляє на сторінку бронювання з вказанням конкретної парковки;
- Розробити таблицю з даними про парковки, підтримкою сортування за колонками (адреса, зона, сектор, ціна, кількість місць, вільні місця);
- Реалізувати фільтри пошуку за адресою, зоною, сектором і наявністю вільних місць, а також пагінацію для зручного перегляду;
- Забезпечити адаптивний дизайн із використанням Tailwind CSS для коректного відображення на різних пристроях.

4. Сторінка бронювання:

- Реалізувати випадаючий список для вибору парковки, якщо `spot_id` не задано через GET-параметри;
- Відобразити паркомісця у вигляді плиток (зелені для вільних, червоні для зайнятих із вказівкою часу зайнятості);
- Додати форму для бронювання з полями: ім'я, телефон, час бронювання, тривалість;
- Забезпечити асинхронне оновлення плиток при виборі парковки;
- Реалізувати інтерактивність плиток (вибір вільного місця активує кнопку «Забронювати»).

5. Розгортання:

- Розгорнути систему на сервері (Apache із PHP та MariaDB) із правильними правами доступу;
- Налаштувати логування помилок для моніторингу роботи системи.

Виконання зазначених задач дозволить створити функціональну, безпечну та зручну систему бронювання паркомісць, яка відповідає сучасним вимогам веб-додатків. Реалізація проєкту передбачає інтеграцію карти, фільтрів, асинхронного завантаження даних і транзакційної логіки для

забезпечення надійності та актуальності інформації. Система стане корисним інструментом для водіїв і може бути розширена додатковими функціями, такими як інтеграція платежів або мобільний додаток.

1.4 Висновки

У розділі було проведено комплексний аналіз предметної області, що дозволив визначити ключові аспекти функціонування паркувальних систем. Огляд аналогічних систем пошуку та бронювання паркомісць виявив їхні переваги й недоліки, що стало основою для формування вимог до розробки нової інформаційної системи. На основі цього сформульовано задачу дослідження, яка в подальшому стане опорою для розробки.

2. ІТ-ІНФРАСТРУКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПОШУКУ ТА БРОНЮВАННЯ ПАРКОМІСЦЬ

2.1 Вибір оптимальних технологій для реалізації

Для реалізації веб-системи бронювання паркомісць у Львові було використано набір мов програмування, які забезпечують серверну логіку, клієнтський інтерфейс, обробку даних і стилізацію. Кожна мова була обрана з урахуванням її сильних сторін і специфіки завдань проєкту. Нижче наведено огляд використаних мов програмування, їхньої ролі та конкретики застосування в системі.

PHP для серверної логіки, обробка запитів, взаємодії з базою даних. PHP (Hypertext Preprocessor) є основною мовою для серверної частини системи, оскільки вона забезпечує динамічне генерування веб-сторінок, обробку HTTP-запитів і взаємодію з базою даних MariaDB. PHP обрано через його широку підтримку в веб-розробці, простоту інтеграції з Apache і надійність для обробки транзакцій. Крім того, PHP має велику спільноту розробників, що гарантує наявність численних бібліотек і фреймворків, таких як Laravel чи Symfony, які значно спрощують створення складних систем з високою продуктивністю [16].

У контексті систем пошуку та бронювання паркомісць PHP відповідає за реалізацію бізнес-логіки, включаючи обробку запитів користувачів на пошук вільних місць, бронювання та скасування броні. Серверна частина обробляє дані, отримані з клієнтської частини, проводить валідацію, взаємодіє з базою даних для збереження або оновлення інформації, а також формує відповіді у форматі JSON або HTML для відправки назад клієнту.

PHP також відповідає за інтеграцію із зовнішніми сервісами, наприклад, API карт або платіжних систем, що дозволяє забезпечити навігацію до паркомісця та безпечну оплату бронювання. Використання PHP дозволяє легко масштабувати систему, розділяючи навантаження на кілька серверів або

впроваджуючи кешування за допомогою Redis чи Memcached, що значно підвищує швидкодію при великій кількості користувачів.

Завдяки своїй гнучкості PHP дозволяє швидко розробляти нові функції, адаптувати систему під різні вимоги клієнтів, а також забезпечує стабільність роботи навіть при пікових навантаженнях. Таким чином, вибір PHP як серверної мови програмування є оптимальним балансом між продуктивністю, надійністю та швидкістю розробки у сфері створення систем пошуку та бронювання паркомісць [17].

JavaScript було обрано для реалізації клієнтської логіки, інтерактивності, роботи з картою, асинхронні запити.

JavaScript забезпечує динамічну поведінку веб-інтерфейсу, обробку подій користувача, взаємодію з картою Leaflet і асинхронне завантаження даних через fetch. Мова обрана через її універсальність у браузерях і підтримку сучасних API. За допомогою JavaScript користувачі можуть в режимі реального часу переглядати доступні паркомісця на інтерактивній карті, отримувати миттєві оновлення про їх статус і легко здійснювати бронювання без необхідності перезавантаження сторінки.

Взаємодія з Leaflet дозволяє ефективно відображати географічні дані, створювати кастомізовані мітки, шари та маршрути, що значно покращує користувацький досвід при пошуку паркувальних місць. Асинхронні запити через fetch забезпечують безперервний обмін даними між клієнтською частиною і сервером, що дозволяє миттєво оновлювати інформацію про наявність паркомісць, тарифні ставки або час дії броні [18].

Завдяки своїй асинхронній природі, JavaScript допомагає оптимізувати продуктивність системи, розвантажуючи сервер і забезпечуючи плавну роботу користувацького інтерфейсу навіть при великій кількості одночасних користувачів. Також ця мова відкриває можливості для інтеграції з іншими сервісами через REST або GraphQL API, що робить систему більш гнучкою і розширюваною.

Отже, JavaScript є ключовим елементом у побудові сучасних веб-додатків для пошуку і бронювання паркомісць, забезпечуючи інтерактивність, швидкість реакції та комфорт користувачів, що значно підвищує загальну ефективність системи.

SQL (Structured Query Language) використано для створення структури бази даних MariaDB, вставки даних, вибірки інформації та оновлення статусів. SQL забезпечує ефективну роботу з реляційними даними і транзакціями, що є критично важливим для систем пошуку та бронювання паркомісць, де важливо зберігати точну і актуальну інформацію про стан кожного місця, час бронювання, користувачів і оплату [19].

За допомогою SQL реалізуються складні запити, які дозволяють швидко знаходити вільні паркомісця в заданій локації, фільтрувати їх за параметрами, такими як ціна, тип паркінгу або час доступності. Важливою частиною роботи є підтримка цілісності даних, що забезпечується за допомогою механізмів транзакцій і обмежень цілісності, таких як первинні ключі, зовнішні ключі і унікальні індекси.

Також SQL дає змогу ефективно оновлювати статуси паркомісць у режимі реального часу, наприклад, позначати місце як зайняте при бронюванні або вивільнене після завершення паркування. Завдяки потужним засобам оптимізації запитів, база даних здатна обробляти велику кількість одночасних операцій, що важливо для підтримки продуктивності системи при великій кількості користувачів.

Використання SQL дозволяє реалізувати звітність та аналітику, наприклад, підрахунок популярності певних паркомісць, аналіз пікових годин навантаження або оцінку прибутковості різних паркувальних зон. Це допомагає адміністраторам системи приймати обґрунтовані рішення щодо управління паркінгами і покращення сервісу.

Таким чином, SQL є основою для надійного зберігання, швидкого доступу і точного оновлення даних у системах пошуку і бронювання

паркомісць, забезпечуючи стабільність, масштабованість і гнучкість у роботі з інформацією.

HTML (HyperText Markup Language) використано для створення структури сторінок `index.php` і `reserve.php`, включаючи навігацію, таблиці, форми, плитки та спливаючі вікна. HTML забезпечує семантичну основу для контенту, що дозволяє браузерам і допоміжним технологіям правильно інтерпретувати і відображати інформацію. Завдяки чіткій структурі HTML коду користувачі можуть легко орієнтуватися на сторінках, знаходити потрібні елементи та взаємодіяти з інтерфейсом, наприклад, заповнювати форми бронювання або переглядати список доступних паркомісць [20].

CSS (Cascading Style Sheets) використано для оформлення сторінок, створення адаптивного дизайну та анімацій. У проєкті CSS застосовується через фреймворк Tailwind CSS, який дозволяє швидко і гнучко задавати стилі за допомогою утилітних класів, що прискорює процес розробки і підтримує консистентність зовнішнього вигляду. Крім того, власні стилі доповнюють фреймворк, забезпечуючи унікальний дизайн специфічних елементів, наприклад, кнопок у спливаючих вікнах, які привертають увагу користувача і покращують взаємодію.

Завдяки використанню CSS реалізовано адаптивність інтерфейсу — сторінки коректно відображаються на різних пристроях, від великих моніторів до мобільних телефонів, що дуже важливо для сучасних користувачів, які часто користуються мобільними додатками або браузерами. Анімації та плавні переходи підвищують приємність користування системою, роблячи взаємодію більш інтуїтивною і комфортною.

Поєднання HTML і CSS формує візуальний і функціональний каркас системи, дозволяючи створювати зрозумілий, сучасний та привабливий інтерфейс, який підтримує всі необхідні функції пошуку, відображення і бронювання паркомісць. Таким чином, ці технології є фундаментальними для побудови зручного користувацького досвіду у веб-системі.

Інтеграція мов програмування у проєкті забезпечує злагоджену роботу різних компонентів системи пошуку та бронювання паркомісць, що підвищує ефективність і зручність користування. PHP і SQL працюють у тісній взаємодії: PHP виконує логіку обробки запитів користувачів, формує відповідні SQL-запити для вибірки, вставки або оновлення даних у базі MariaDB. Завдяки цьому серверна частина здатна оперативно реагувати на дії користувача, наприклад, шукати вільні місця або реєструвати бронювання, забезпечуючи цілісність і актуальність даних [21].

PHP і HTML інтегруються через динамічну генерацію контенту. PHP вставляє отримані з бази даних дані безпосередньо в HTML-шаблони сторінок, таких як `index.php` і `reserve.php`. Це дозволяє створювати сторінки, які адаптуються під конкретного користувача і контекст, відображаючи актуальну інформацію про паркомісця, тарифи та інші параметри.

JavaScript у взаємодії з HTML маніпулює DOM-структурою, динамічно оновлюючи вміст сторінок без їх повного перезавантаження. Він відповідає за оновлення таблиць, плиток з інформацією про паркомісця та спливаючих вікон, що значно покращує користувацький досвід, роблячи його більш інтерактивним і швидким. Через JavaScript також здійснюється асинхронний обмін даними із сервером за допомогою `fetch`, що дозволяє отримувати нові дані в реальному часі.

Взаємодія JavaScript і CSS відбувається через динамічне додавання або видалення CSS-класів, наприклад, класу `selected` для підсвічування обраної плитки або кнопки. CSS, у свою чергу, через Tailwind і власні стилі визначає зовнішній вигляд елементів, їх розташування, кольори, шрифти та анімації, що забезпечує привабливий та зрозумілий інтерфейс [22].

CSS і HTML працюють спільно для створення адаптивного дизайну, який коректно відображається на різних пристроях і екранах. HTML формує структуру, а CSS — зовнішній вигляд і поведінку елементів, включно з адаптацією до розміру екрану, що важливо для мобільних користувачів.

Загалом, тісна інтеграція цих мов програмування і технологій забезпечує високу продуктивність, гнучкість і зручність системи, роблячи її ефективним інструментом для пошуку і бронювання паркомісць у реальному часі.

2.2 Формування набору даних для роботи веб-системи

Для розроблення та тестування системи пошуку та бронювання паркомісця було використано відкриті дані про парковки від сервісу ЛеоПарковка, який адмініструється ЛКП "Львівавтодор". Ці дані були отримані з офіційного відкритого ресурсу, що надає доступ до інформації про паркувальні майданчики, які перебувають в управлінні міського оператора. Зокрема, в наборі містилася структурована інформація про розташування паркінгів, їхні адреси, кількість місць, тип паркувального майданчика, режим роботи та інші супровідні атрибути.

Дані були надані у форматі KML (Keyhole Markup Language) — це стандарт XML-подібного представлення геопросторової інформації, який часто використовується для обміну даними між геоінформаційними системами, такими як Google Earth або картографічні модулі типу Leaflet і OpenStreetMap. Формат KML дозволяє точно відображати координати точок, ліній та полігонів, що ідеально підходить для візуалізації паркувальних майданчиків на карті.

На рисунку 2.1 показано завантаження даних у форматі KML.

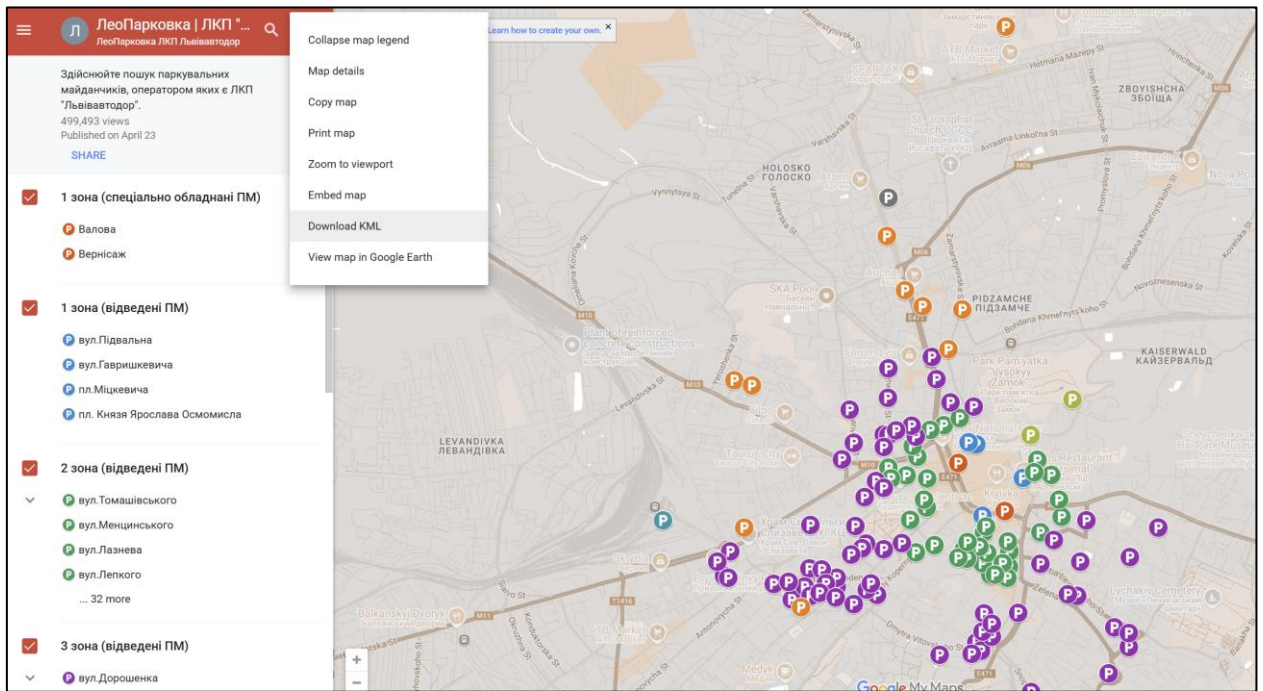


Рисунок 2.1 – Завантаження даних про парковки у форматі KML

Оскільки дані саме в такому форматі не несуть великої цінності та важко опрацьовуються в форматі розроблення системи, за допомогою скрипта написаного на Python їх було конвертовано в формат CSV. На рисунку 2.2 показано отримані після конвертації дані.

	A	B	C	D	E
1	name	description	latitude	longitude	address
2	Валова	1 зона (спеціально обладнані паркувальні майданчик), 45 грн/год Кількість паркувальних місць: 49.8394206 24.0330654	49.8394206	24.0330654	Валова
3	Вернісаж	1 зона (спеціально обладнані паркувальні майданчик), 45 грн/год Кількість паркувальних місць: 49.8428797 24.0279346	49.8428797	24.0279346	Вернісаж
4	вул.Підвальна	1 зона, 1 сектор 35 грн/год Способи оплати: Безготівково: через застосунок 49.8418114 24.0350976	49.8418114	24.0350976	вул.Підвальна
5	вул.Гавришкевича	1 зона, 1 сектор 35 грн/год Способи оплати: Безготівково: через застосунок 49.8441927 24.0299007	49.8441927	24.0299007	вул.Гавришкевича
6	пл.Міцкевича	1 зона, 1 сектор 35 грн/год Способи оплати: Безготівково: через застосунок 49.8391168 24.0360216	49.8391168	24.0360216	пл.Міцкевича
7	пл.Князя Ярослава Осмомисла	1 зона, 1 сектор 35 грн/год Способи оплати: Безготівково: через застосунок 49.8443361 24.0291036	49.8443361	24.0291036	пл.Князя Ярослава Осмомисла
8	вул.Томашівського	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8367042 24.0285441	49.8367042	24.0285441	вул.Томашівського
9	вул.Менцинського	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8419836 24.0221894	49.8419836	24.0221894	вул.Менцинського
10	вул.Лазнева	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8455604 24.0263583	49.8455604	24.0263583	вул.Лазнева
11	вул.Лепкого	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8425182 24.0220261	49.8425182	24.0220261	вул.Лепкого
12	вул.Гребінки	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8417444 24.0205019	49.8417444	24.0205019	вул.Гребінки
13	вул.Сянська	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8460835 24.0280956	49.8460835	24.0280956	вул.Сянська
14	вул.Драгоманова	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8348849 24.0314465	49.8348849	24.0314465	вул.Драгоманова
15	вул.Стецька	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8354106 24.0335178	49.8354106	24.0335178	вул.Стецька
16	вул.Саксаганського	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8346525 24.033395	49.8346525	24.033395	вул.Саксаганського
17	вул.Григорівича	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8356972 24.0306189	49.8356972	24.0306189	вул.Григорівича
18	вул.Фредра	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.836568 24.0336092	49.836568	24.0336092	вул.Фредра
19	вул.Вагова	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8452178 24.0248378	49.8452178	24.0248378	вул.Вагова
20	вул.Волошина	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.836438 24.0311134	49.836438	24.0311134	вул.Волошина
21	вул.Пекарська	2 зона, 2-3 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8379313 24.0370941	49.8379313	24.0370941	вул.Пекарська
22	вул.Білозіра	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.835725 24.0328861	49.835725	24.0328861	вул.Білозіра
23	вул.Грушевського	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8348979 24.0319578	49.8348979	24.0319578	вул.Грушевського
24	пл.Маланюка	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8379387 24.0308245	49.8379387	24.0308245	пл.Маланюка
25	вул.Ковчуна	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8368749 24.0302652	49.8368749	24.0302652	вул.Ковчуна
26	вул.Костюшка	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8397183 24.0244638	49.8397183	24.0244638	вул.Костюшка
27	вул.Князя Романа	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.837284 24.0334192	49.837284	24.0334192	вул.Князя Романа
28	вул.Наймвайка	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8432926 24.023434	49.8432926	24.023434	вул.Наймвайка
29	вул.Дудаса	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.836064 24.0289141	49.836064	24.0289141	вул.Дудаса
30	вул.Стефаніка	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8369839 24.0253216	49.8369839	24.0253216	вул.Стефаніка
31	вул.Стефаніка (від пер. Скорика до пер. вул. Дудаса)	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8369091 24.0279647	49.8369091	24.0279647	вул.Стефаніка (від пер. Скорика до пер. вул. Дудаса)
32	вул.Січових Стрільців	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8402714 24.0241894	49.8402714	24.0241894	вул.Січових Стрільців
33	вул.Скорика	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8372258 24.0290079	49.8372258	24.0290079	вул.Скорика
34	вул.Словацького	2 зона, 2-4 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8388219 24.0226587	49.8388219	24.0226587	вул.Словацького
35	вул.Просвіти	2 зона, 2-3 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8420296 24.0382373	49.8420296	24.0382373	вул.Просвіти
36	вул.Коперника	2 зона, 2-5 сектор 25 грн/год Способи оплати: Безготівково: через застосунок 49.8364623 24.0232407	49.8364623	24.0232407	вул.Коперника

Рисунок 2.2 – Отримані дані у форматі CSV

Для коректної роботи з цими даними необхідно їх правильно структурувати та розбити на окремі таблиці. Також, дані що містяться зараз в стовпці «description» потрібно розбити на декілька окремих стовпців. Для цього було написано окремий Python-скрипт, який дозволив отримати CSV файл з необхідними стовпцями.

Отримавши CSV файл з необхідними даними, було розроблено структуру бази даних, яка б відповідала поточним потребам системи. Структуру розробленої бази даних показано на рисунку 2.3.

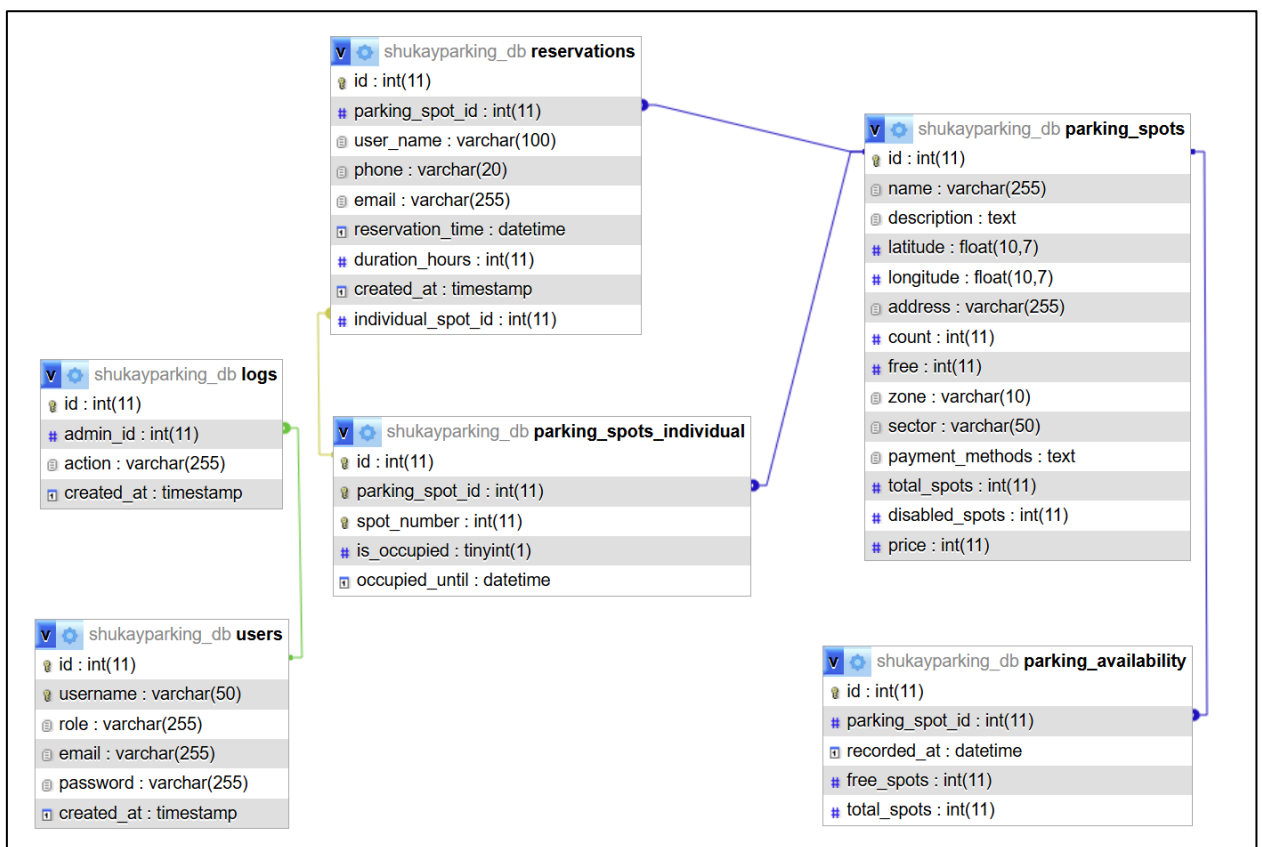


Рисунок 2.3 – ER діаграма бази даних системи бронювання паркомісць

База даних системи бронювання паркомісць складається з п'яти таблиць (parking_spots, parking_spots_individual, reservations, logs, users), які пов'язані через зовнішні ключі для забезпечення цілісності.

Кожна з таблиць виконує наступні функції:

- parking_spots: Зберігає інформацію про парковки (адреса, зона, кількість місць, вільні місця, ціна);
- parking_spots_individual: Містить дані про окремі паркомісця (номер, статус зайнятості, час до якого зайнято).
- reservations: Зберігає бронювання (ID парковки, ID місця, ім'я, телефон, email, час, тривалість);
- logs: Логування дій системи (дії, деталі, час) для дебагування та аудиту;
- users: Містить дані користувачів (email, роль, пароль) для авторизації;
- parking_availability: Зберігає історичні дані про зайнятість парковок (ID парковки, час, вільні/загальні місця) для аналітики.

В результаті було отримано базу даних що відповідає поставленим вимогам та є оптимальною для розроблення системи.

2.3 IT-інфраструктура та архітектура інформаційної системи

Система пошуку та бронювання паркомісць побудована за клієнт-серверною архітектурою з трирівневою моделлю: рівень інтерфейсу (веб-інтерфейс), рівень обробки (бізнес-логіка та запити), рівень даних (база даних). Клієнтська частина — це веб-інтерфейс у браузері, серверна — веб-сервер і база даних. Такий підхід забезпечує розподіл навантаження та можливість масштабування.

Клієнтська частина — це веб-інтерфейс, створений із використанням HTML, CSS (з фреймворком Tailwind CSS) і JavaScript. Вона забезпечує інтерактивну роботу: створення та редагування бронювань, управління паркомісцями, перегляд статистики. Доступ здійснюється через HTTPS із будь-якого пристрою з браузером.

Серверна частина працює на веб-сервері, який обробляє запити та взаємодіє з базою даних MySQL. База даних містить таблиці для зберігання інформації про паркомісця, бронювання, користувачів і логи. Розгортання здійснюється на хмарному сервері (наприклад, AWS EC2) або локальному сервері організації з цілодобовим доступом.

На рисунку 2.4 показано блок-схему архітектури інформаційної системи пошуку та бронювання паркомісць.

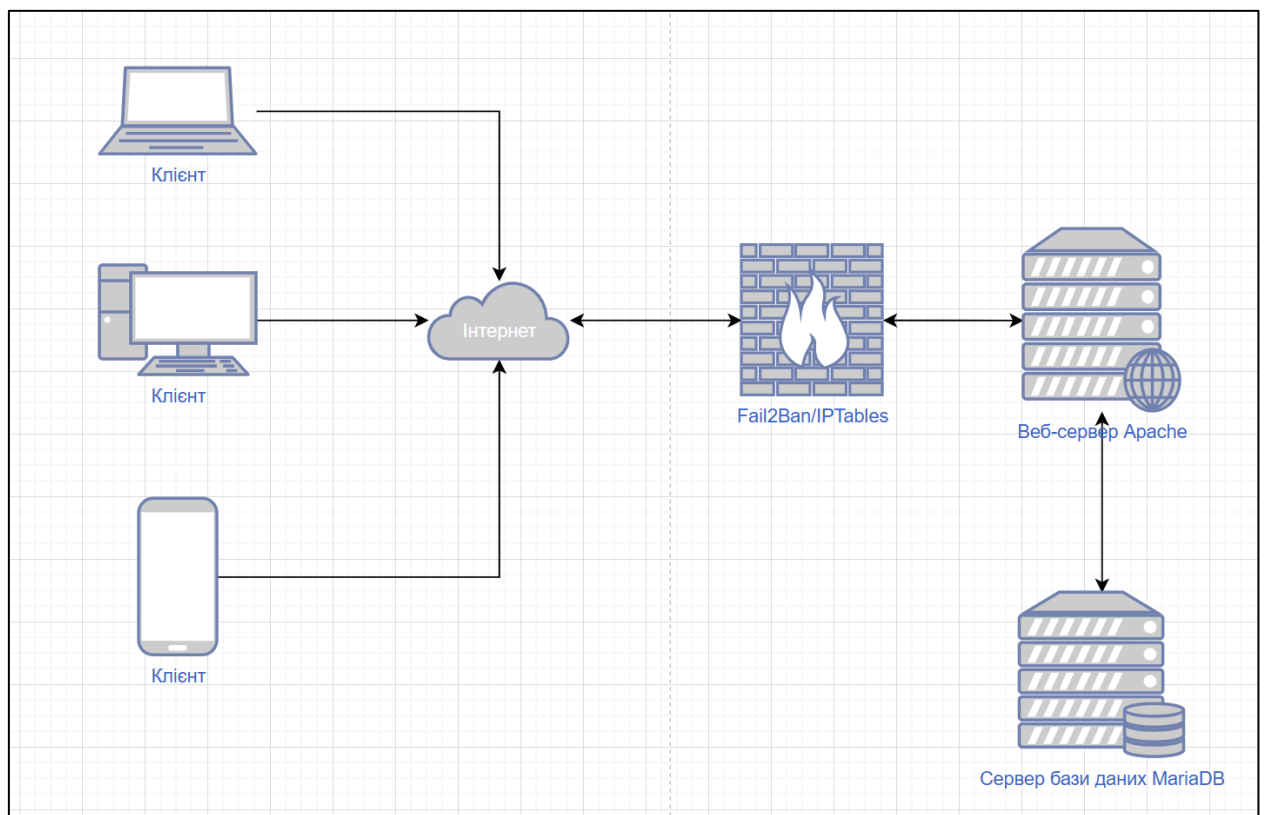


Рисунок 2.4 – Блок-схема архітектури клієнт-серверної системи пошуку та бронювання паркомісць

Було прийнято рішення в тестовому форматі розгорнути систему на віртуальному сервері. На сервер було виділено такі ресурси:

- CPU: 4 ядра;
- RAM: 4 гігабайти;
- SSD: 100 гігабайт дискового простору.

На віртуальному сервері було розгорнуто операційну систему Ubuntu Server та розгорнуто LAMP (Linux, Apache, MySQL, PHP). Для покращення безпеки, було також встановлено фаєрволи iptables та fail2ban, які будуть автоматично блокувати IP-адреси користувачів, які намагаються підібрати пароль до будь-якого з сервісів на сервісі, включно з розгорнутою інформаційною системою.

На рисунку 2.5 показано діаграму, що відображає принцип роботи системи fail2ban та iptables.

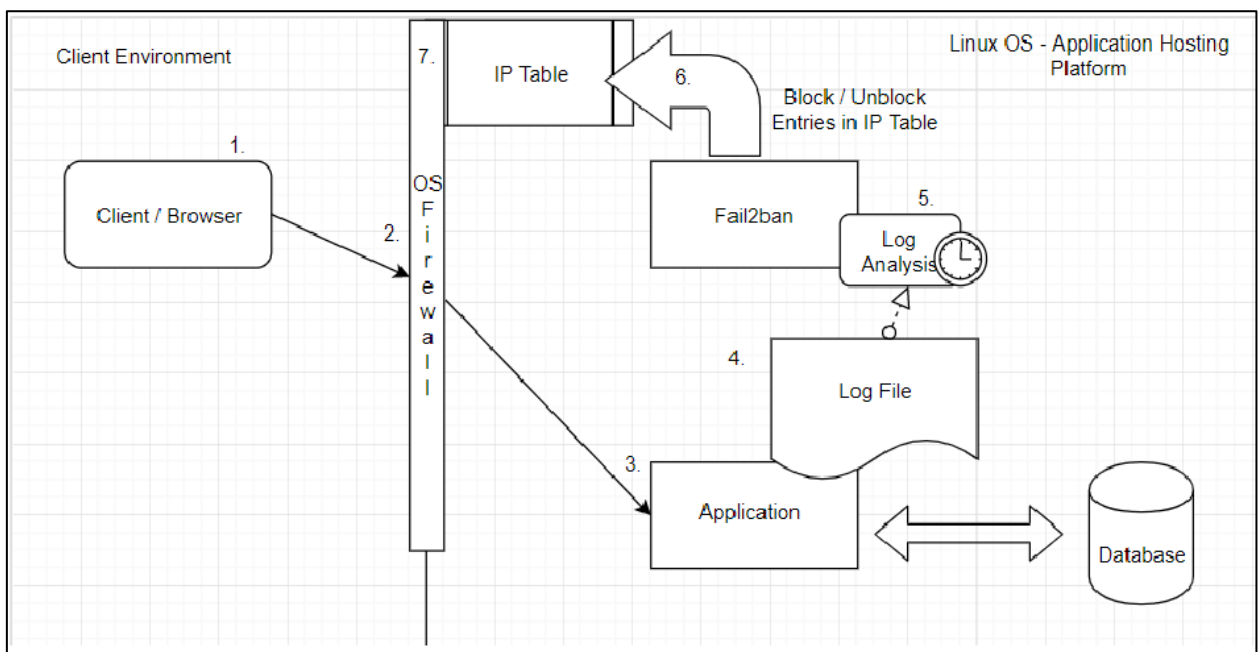


Рисунок 2.5 – Схема роботи алгоритму блокування зловмисних IP-адрес

Система реалізує базові, але важливі заходи безпеки, які забезпечують надійний захист від найпоширеніших загроз. Автентифікація адміністратора через перевірку ідентифікатора в сесії обмежує доступ до панелі лише авторизованим користувачам. Використання HTTPS гарантує шифрування даних при їх передачі, що забезпечує конфіденційність інформації.

Функція логування дозволяє фіксувати всі дії адміністраторів із зазначенням часу, ідентифікатора та опису подій, що сприяє прозорості та підвищує контроль за роботою користувачів із правами доступу. Санітація

введених даних захищає систему від SQL-ін'єкцій та підвищує загальну стійкість до поширених атак.

2.4 Висновки

У розділі було визначено оптимальні технології для реалізації веб-системи, зокрема JavaScript, PHP та MySQL, що забезпечують ефективну обробку даних і надійність. Проведено підготовку середовища для роботи системи, створено віртуальний сервер та встановлено ОС Ubuntu Server. Також на сервері було налаштовано Apache для обробки веб-запитів, MariaDB в якості СКБД та реалізовано розроблену структуру бази даних.

Також сформовано набір даних, який включає інформацію про паркомісця, бронювання та користувачів, забезпечуючи основу для тестування та подальшої експлуатації системи.

3. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розроблення алгоритму функціонування системи

Схема роботи алгоритму — це графічне або логічне подання кроків, які виконує алгоритм для розв’язання задачі. Вона показує послідовність дій, умовні переходи, цикли та інші логічні операції, які виконує алгоритм під час роботи. На рисунку 3.1 показано схему роботи алгоритму інформаційної системи бронювання паркомісць.



Рисунок 3.1 – Схема роботи алгоритму бронювання паркомісць

Блок-схема на рисунку 3.1 ілюструє послідовність дій для обробки запиту на бронювання паркомісця, від ініціалізації системи та отримання даних до завершення.

Процес бронювання в системі починається з того, що користувач відкриває сторінку бронювання і надсилає POST-запит із необхідними даними, такими як ім'я, телефон, email, час бронювання, тривалість та ID паркомісця. Система отримує ці дані, фільтрує та валідує їх, наприклад, перевіряє правильність email чи чи є значення числовим.

Далі виконується перевірка, чи вибране місце не зайняте, за допомогою запиту до бази даних. Якщо місце вільне, система починає транзакцію, оновлює інформацію в базі (відмічає місце як зайняте, зменшує кількість вільних місць, додає запис про бронювання), і зберігає деталі в сесію. Після успішного завершення транзакції користувача перенаправляють на сторінку підтвердження з деталями бронювання.

Якщо ж місце вже зайняте або виникла помилка під час транзакції, система фіксує це в логах, виконує відкат змін і повідомляє користувача про помилку, повертаючи форму для повторної спроби. У підсумку користувач або бачить підтвердження бронювання, або повідомлення про невдачу. На рисунку 3.2 показано діаграму розгортання веб-системи, яка ілюструє компоненти які є необхідними для розгортання системи.

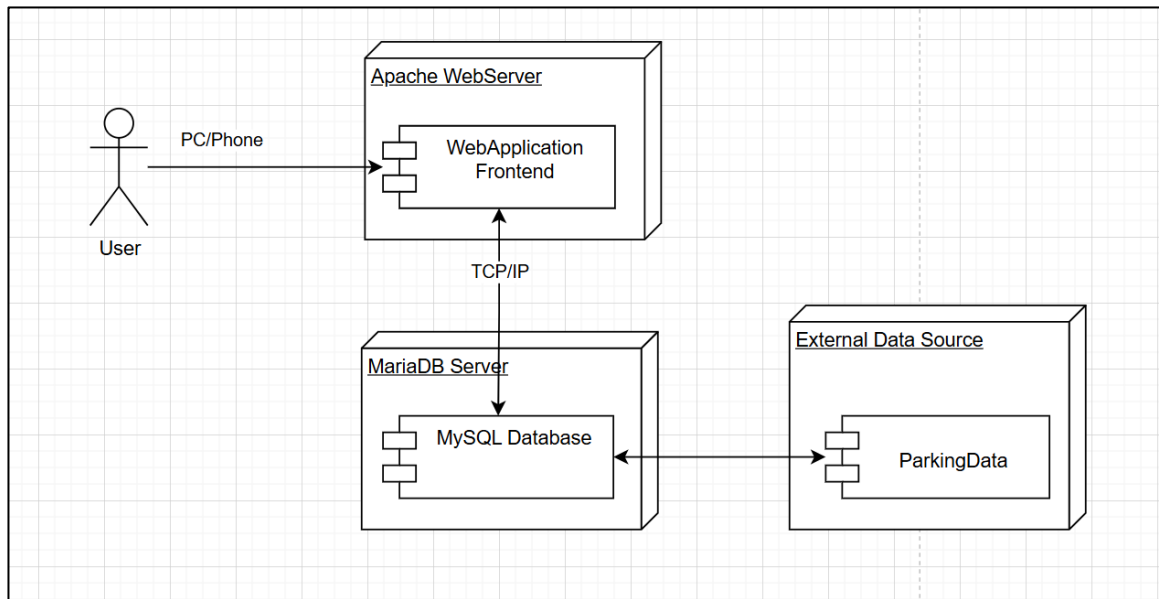


Рисунок 3.2 – Діаграма розгортання веб-системи пошуку та бронювання паркомісць

Діаграма розгортання показує, як система бронювання розміщена на фізичних вузлах і як між ними відбувається обмін даними. Користувач взаємодіє із веб-додатком через браузер на ПК або телефоні, надсилаючи запити до веб-сервера Apache, де працює інтерфейс сайту. Веб-сервер, у свою чергу, обробляє ці запити, звертаючись до бази даних на сервері MariaDB для зчитування чи запису інформації про бронювання. Зовнішні джерела даних (наприклад, сенсори або API) можуть автоматично передавати інформацію про наявність місць, що оновлюється в базі даних. Уся комунікація між компонентами відбувається через протокол TCP/IP. Така архітектура забезпечує чіткий поділ обов'язків між інтерфейсом, логікою та даними, дозволяючи масштабувати систему та легко інтегрувати нові джерела інформації.

На рисунку 3.3 показано діаграму варіантів використання системи для зареєстрованого користувача та гостя.

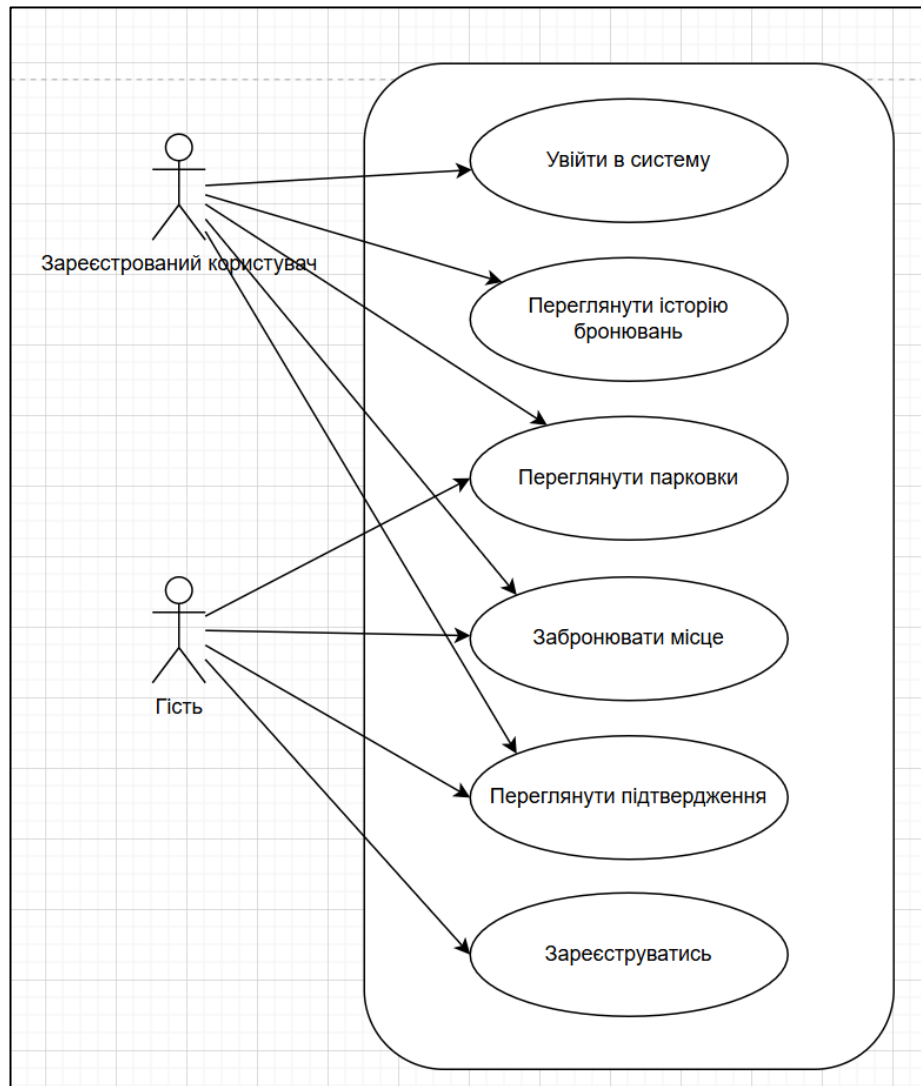


Рисунок 3.3 – Діаграма варіантів використання для гостя та зареєстрованого користувача

Для Гостя забезпечено базові функції: перегляд парковок, бронювання місця з подальшим підтвердженням і можливість реєстрації, що покриває потреби некоеєстрованих користувачів. Зареєстрований користувач розширює ці можливості, додаючи доступ до історії бронювань і авторизацію, що підвищує зручність для повторних відвідувачів.

На рисунку 3.4 показано діаграму варіантів використання для адміністратора системи.



Рисунок 3.4 – Діаграма варіантів використання для адміністратора

Адміністратор має повний контроль: від авторизації та аналізу зайнятості парковок до керування парковками, користувачами та логами, а також функцію виходу з системи, що забезпечує безпеку та адміністрування.

Діаграма чітко розмежовує ролі, відображаючи, як система адаптується до різних типів користувачів.

3.2 Розроблення інформаційної системи

На рисунку 3.5 можна бачити головну сторінку веб-сайту. Сторінка сайту призначена для відображення всіх паркомісць у місті Львів. Вона має сучасний інтерфейс із шапкою, що містить навігаційне меню з посиланнями на головну, бронювання та інформацію про компанію. Основна частина

сторінки складається з інтерактивної карти, реалізованої за допомогою Leaflet.js, на якій позначені всі паркомісця, отримані з бази даних.

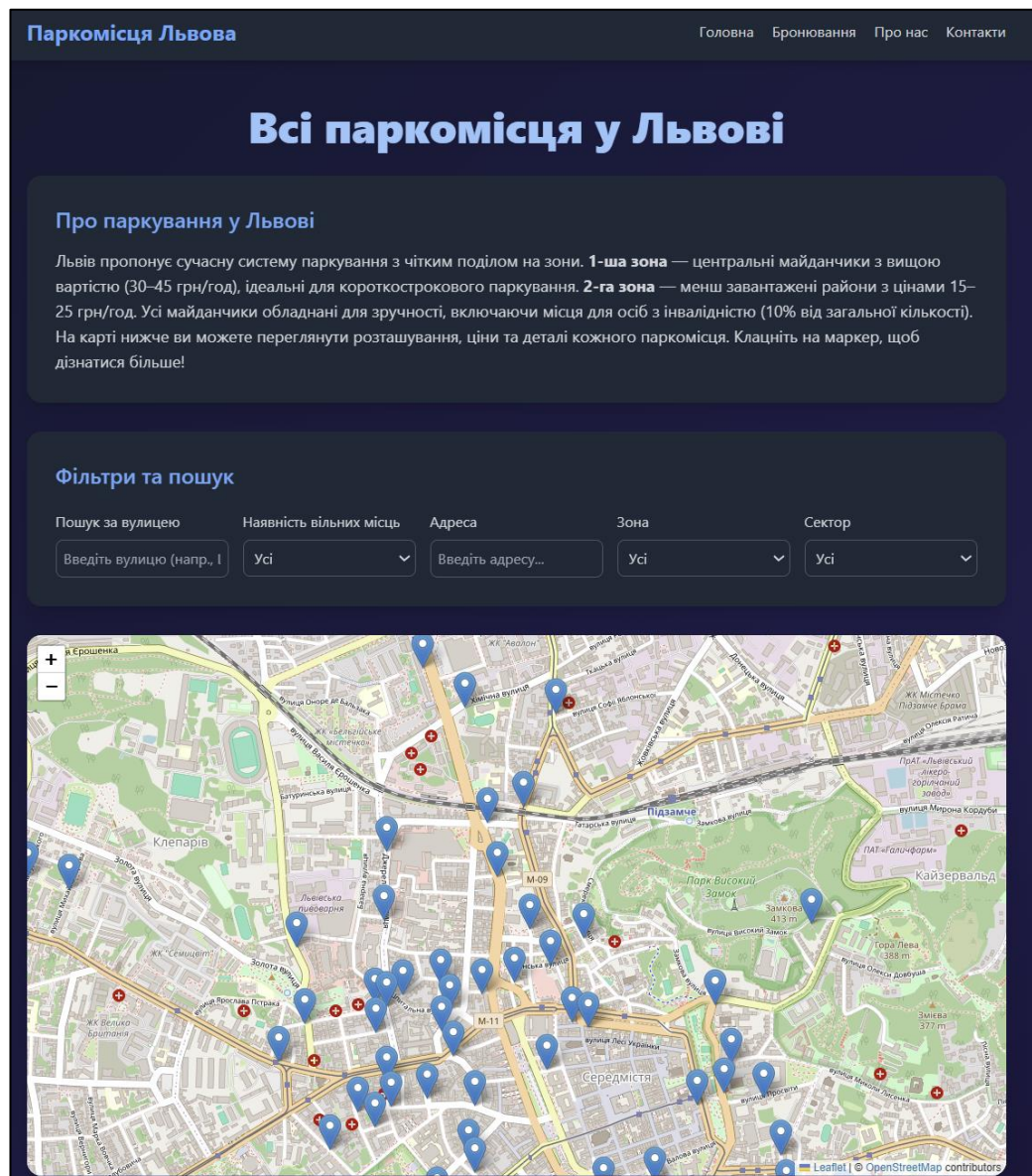


Рисунок 3.5 – Перша частина головної сторінки

Користувач може змінювати масштаб, переглядати розташування паркомісць і фільтрувати їх за зонами, які динамічно витягуються з таблиці бази даних. На рисунку 3.6 можна бачити другу частину головної сторінки.

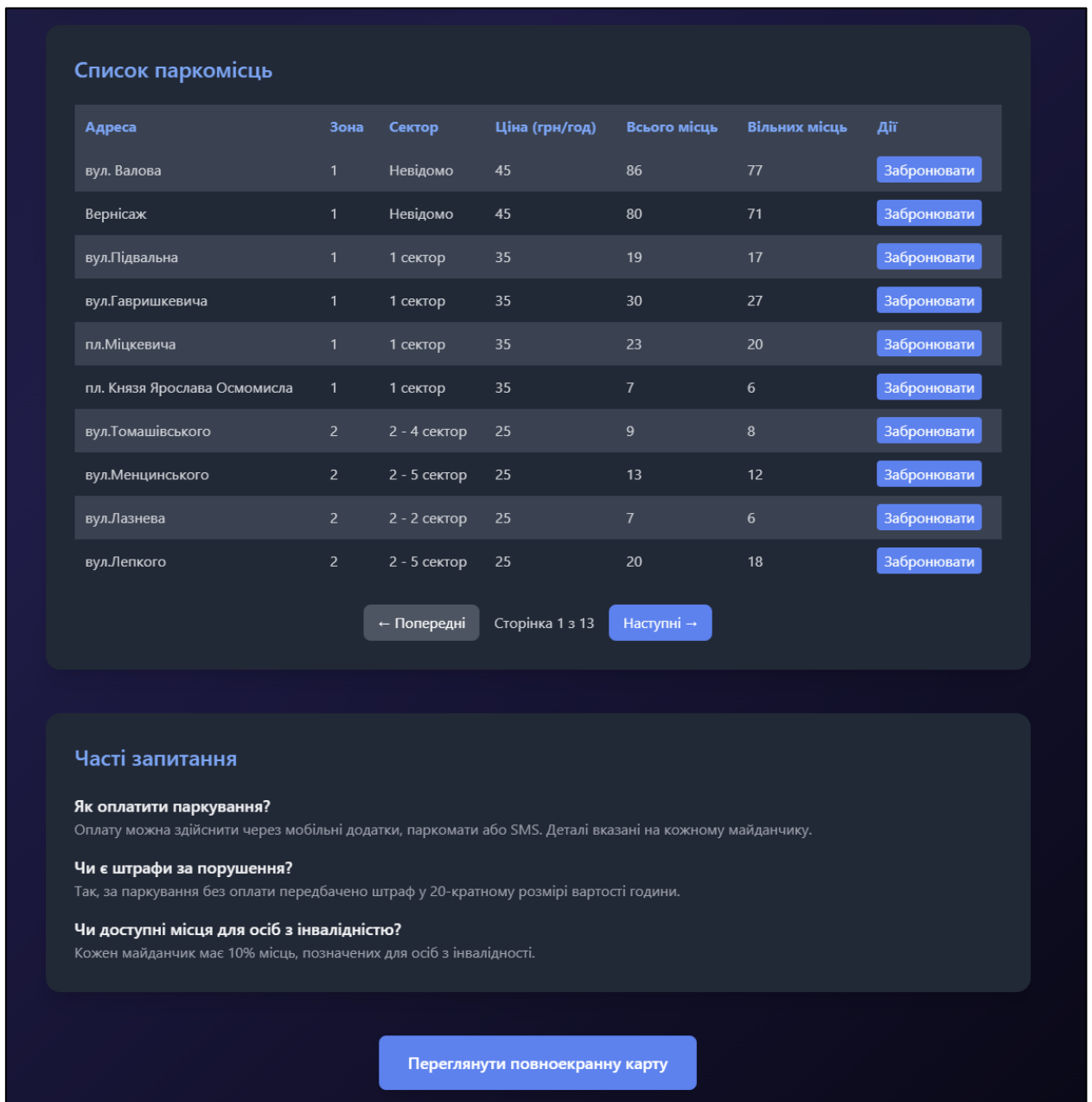


Рисунок 3.6 – Друга частина головної сторінки

Окрім міток з парковками на карті, під нею також розташоване меню в якому можна бачити текстову інформацію про парковки. Окрім того, одразу можна обрати потрібну та забронювати її.

Кнопка «Забронювати» також міститься на кожній мітці на карті, включно з інформацією про парковку. Це показано на рисунку 3.7.

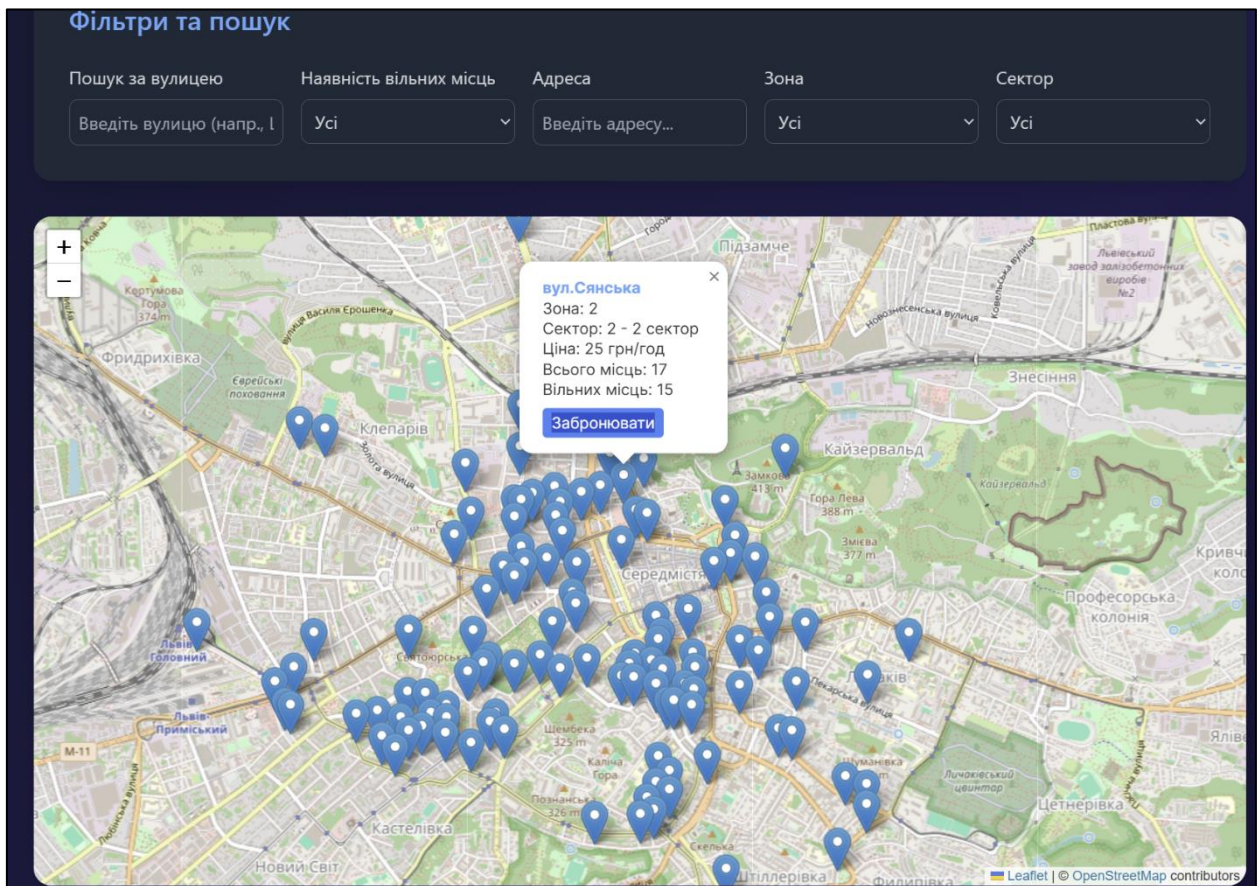


Рисунок 3.7 – Можливість обрати парковку для бронювання на мапі

Окрім того, система має систему пошуку та фільтрації парковок за назвою, наявністю вільних паркомісць, Повною адресою, вулицею, зоною та сектором. Для наглядності, на рисунку 3.8 продемонстровано приклад пошуку парковок у радіусі кілометра від вулиці «Тадеуша Костюшка».

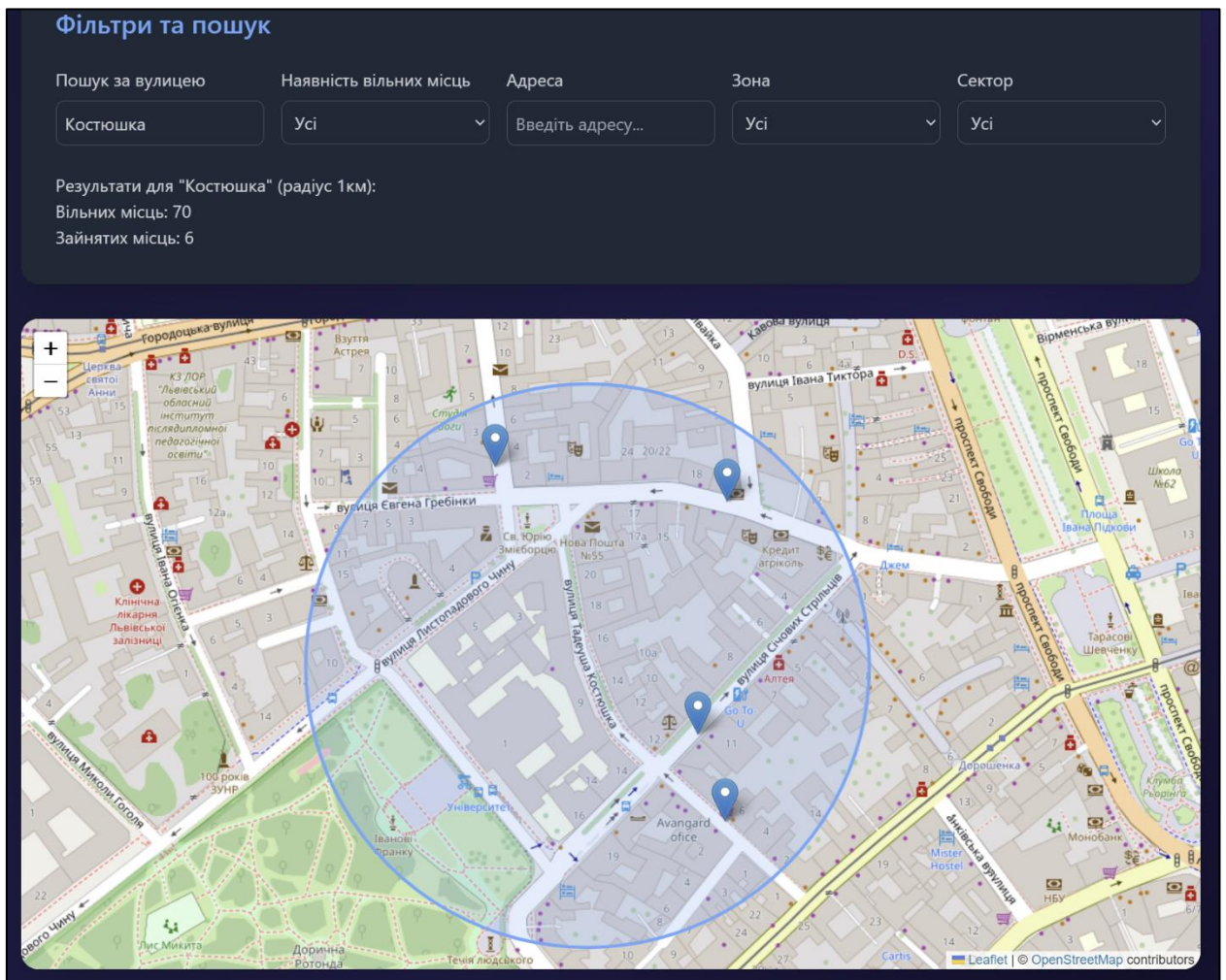


Рисунок 3.8 – Пошук парковок за вулицею

Як можна бачити на рисунку 3.8, пошук працює правильно та за вказаним запитом було знайдено 4 парковки. Обравши необхідну парковку, користувач потрапляє на сторінку бронювання, яку показано на рисунку 3.9.

Паркомісія Львова

Головна Бронювання Про нас Контакти

Бронювання паркомісць

Парковка: вул.Менцинського (Зона: 2)

Оберіть паркомісце

Місце №1	Місце №2	Місце №3	Місце №4	Місце №5	Місце №6
Місце №7	Місце №8	Місце №9	Місце №10	Місце №11	Місце №12

Ваше ім'я

Номер телефону

Email

Час бронювання

Тривалість (години)

Рисунок 3.9 – Сторінка «Бронювання паркомісць»

Обравши парковку та місце, користувачу також необхідно ввести додаткові дані, такі як ім'я, номер, поштову скриньку, дату, час і тривалість бронювання. Після введення всіх даних користувач отримує підтвердження що паркомісце за ним заброньоване, а інформація в базі даних оновлюється і на період бронювання місце стає недоступним для бронювання (рис. 3.10-3.11).

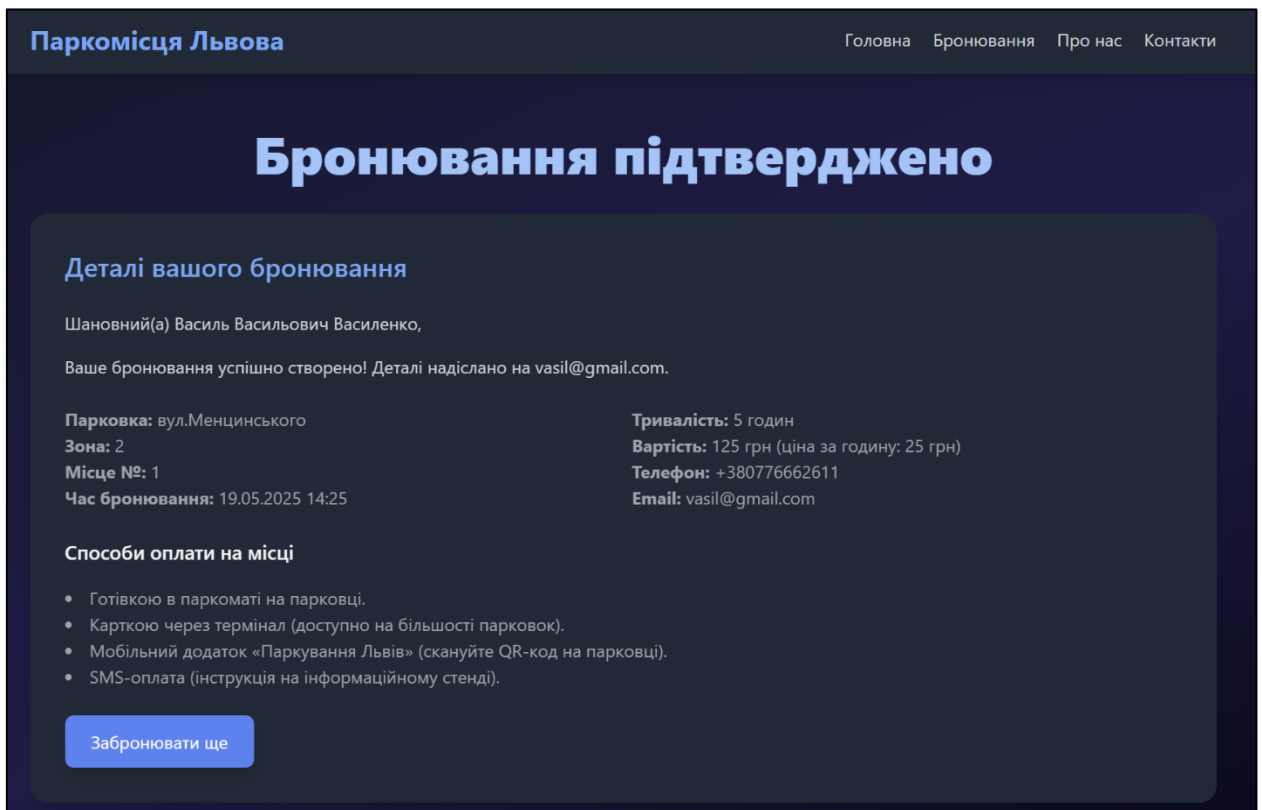


Рисунок 3.10 – Підтвердження бронювання

На сторінці підтвердження бронювання користувач отримує детальну інформацію про своє бронювання. Вона включає ім'я користувача, електронну адресу, на яку надіслано додаткові відомості, загальні дані про обрану парковку, вартість послуги, тривалість бронювання, доступні способи оплати та точний час оформлення.

Система також автоматично фіксує що конкретне паркомісце на конкретній парковці було заброньовано та в подальшому буде відображати користувачам що це місце недоступне для бронювання (рис. 3.11).

Паркомісця Львова Головна Бронювання Про нас Контакти

Бронювання паркомісць

Парковка: вул.Менцинського (Зона: 2)

Оберіть паркомісце

Місце №1 Зайнято до: 19.05.2025 19:25	Місце №2	Місце №3	Місце №4	Місце №5	Місце №6
Місце №7	Місце №8	Місце №9	Місце №10	Місце №11	Місце №12

Ваше ім'я

Номер телефону

Email

Час бронювання

Тривалість (години)

Забронювати

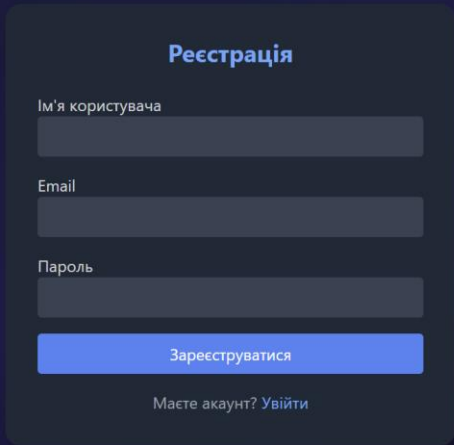
Рисунок 3.11 – Перевірка вже заброньованого паркомісця

Після успішного бронювання паркомісця, при повторному перегляді сторінки цієї ж парковки, заброньоване місце відображається з відповідною позначкою, що свідчить про його недоступність для інших користувачів. Це дозволяє уникнути повторного бронювання одного й того ж місця та забезпечує наочне інформування про актуальний стан паркомісць. Така функціональність підвищує зручність користування системою та сприяє кращому плануванню паркування.

Подальше вивільнення заброньованих паркомісць по закінченню строку бронювання виконується певним скриптом.

Окрім того, що користувачі можуть просто бронювати паркомісця використовуючи розроблену систему, в них також є можливість створення власних акаунтів для зберігання інформації та керування своїми

бронюваннями. Для цього було зроблено систему реєстрації, авторизації та особистого кабінету. На рисунку 3.12 показано сторінку реєстрації.



The image shows a registration form titled "Реєстрація" (Registration) centered on a dark blue background. The form is a light gray rounded rectangle containing three input fields: "Ім'я користувача" (Username), "Email", and "Пароль" (Password). Below these fields is a blue button labeled "Зареєструватися" (Register). At the bottom of the form, there is a link that says "Маєте акаунт? Увійти" (Have an account? Log in).

Рисунок 3.12 – Сторінка реєстрації користувачів

Після введення ім'я користувача, пошти та паролю, якщо всі дані є коректними, користувач отримує повідомлення про успішну реєстрацію та автоматично переспрямовується на сторінку входу. Після входу в акаунт, користувач потрапляє в кабінет користувача, який показано на рисунку 3.13.

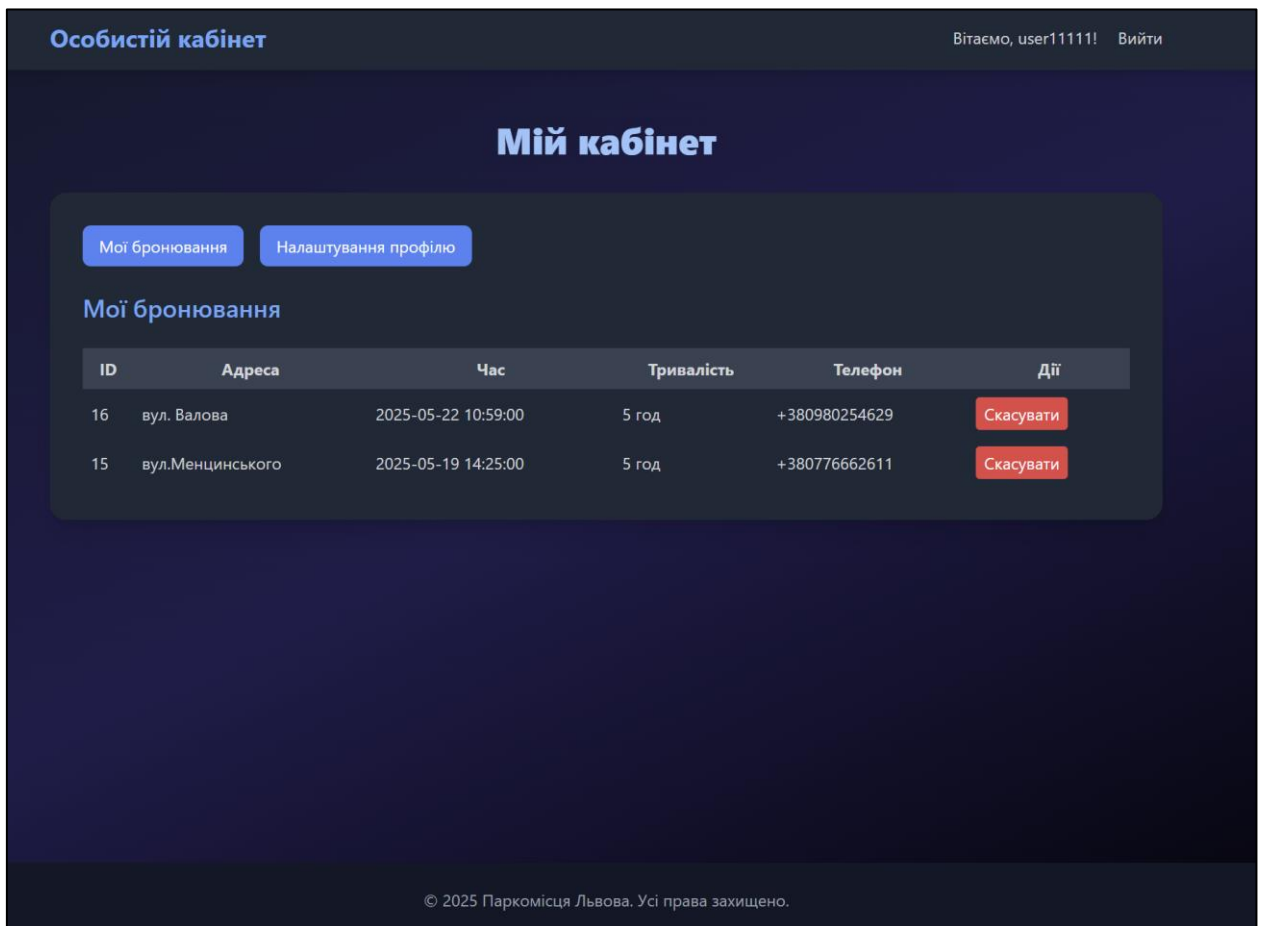


Рисунок 3.13 – Особистий кабінет користувача

Основний інтерфейс складається з двох вкладок: "Мої бронювання" та "Налаштування профілю". У вкладці "Мої бронювання" показано таблицю з усіма активними та минулими бронюваннями користувача, де для кожного запису вказано ID, адресу парковки, час бронювання, тривалість у годинах, номер телефону та кнопка для скасування бронювання, якщо воно ще актуальне — при натисканні з'явиться підтвердження дії, а після скасування кількість вільних місць на парковці автоматично оновиться.

На рисунку 3.14 показано розділ «Налаштування профілю» особистого кабінету.

Особистий кабінет

Вітаємо, user11111! Вийти

Мій кабінет

Мої бронювання Налаштування профілю

Мій профіль

Ім'я користувача

user11111

Телефон

+380123456789

Зберегти

Зміна пароля

Старий пароль

Новий пароль

Підтвердити новий пароль

Змінити пароль

Рисунок 3.14 – Вкладка «Налаштування профілю»

На сторінці налаштування профілю можна змінити ім'я користувача або номер телефону, для цього достатньо ввести нові значення в відповідні поля та натиснути кнопку "Зберегти". У цій же вкладці доступна форма для зміни пароля — потрібно ввести старий пароль, новий пароль та його підтвердження, після чого натиснути "Змінити пароль". Важливо, що новий пароль має містити щонайменше 6 символів, а якщо введені дані не збігаються чи старий пароль невірний, система повідомить про помилку. Усі дії, як скасування бронювання чи зміна пароля, супроводжуються повідомленнями про успіх або помилку, які з'являються у верхній частині сторінки.

Для перевірки функціональності було змінено ім'я користувача на Denis Newman. На рисунку 3.15 показано результат зміни.

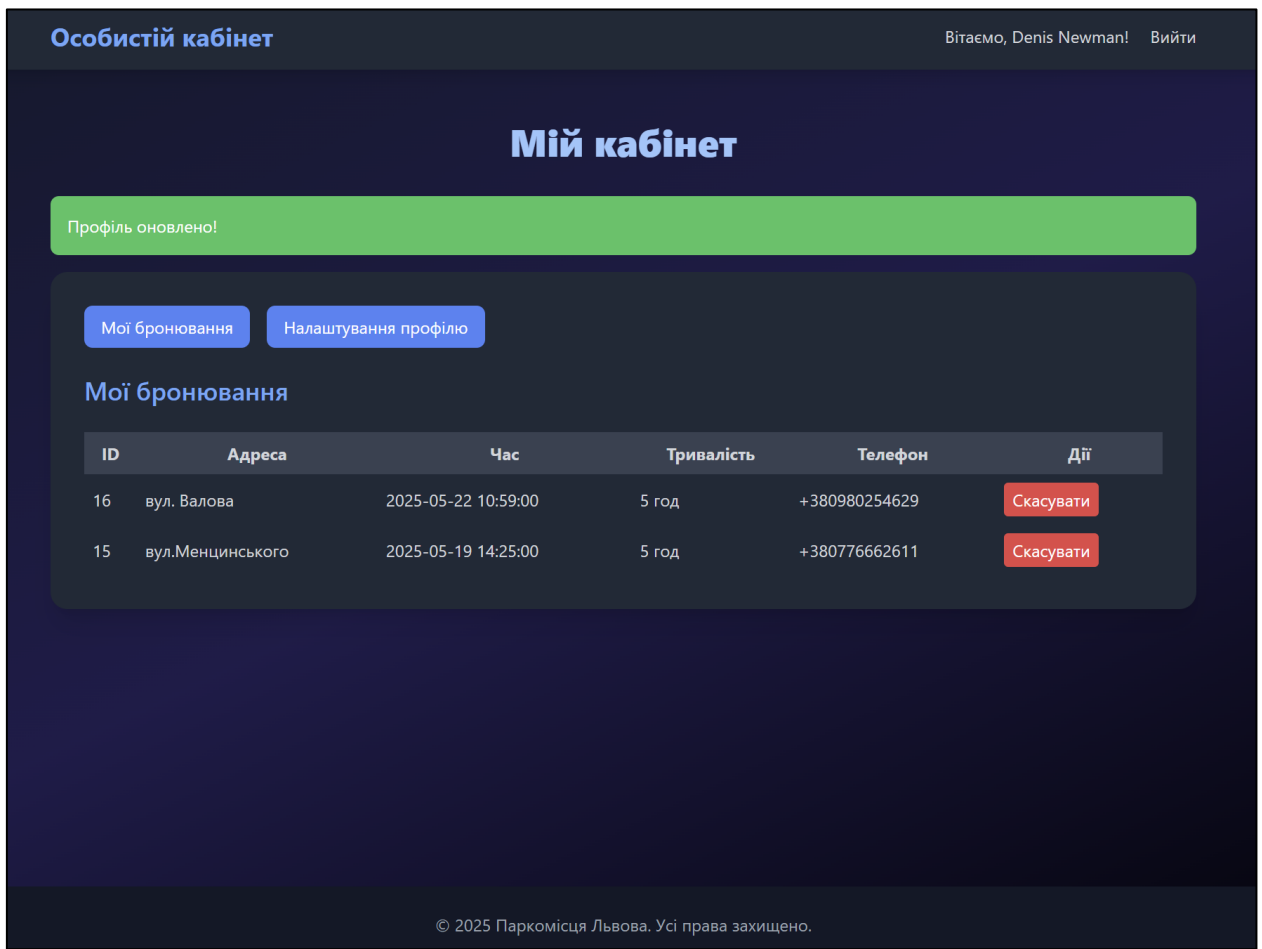


Рисунок 3.15 – Результат зміни імені користувача

Як можна бачити на рисунку 3.15, користувачеві було виведено повідомлення про успішне оновлення профілю, та у верхній частині сторінки було автоматично змінено ім'я користувача у повідомленні вітання.

3.3 Розроблення панелі адміністратора

Крім панелі користувача, для системи також було розроблено панель адміністратора для керування системою. Для цього було зроблено сторінку входу, окрему від користувацької. На рисунку 3.16 показано сторінку входу в панель адміністратора.

The screenshot shows a web application interface for 'Паркомісця Львова' (Lviv Parking Spaces). The header includes the site name on the left and a 'Головна' (Home) link on the right. The main content area is titled 'Вхід адміністратора' (Administrator Login). It contains a login form with two input fields: 'Логін' (Login) and 'Пароль' (Password). Below these fields is a blue button labeled 'Увійти' (Login). The footer contains a copyright notice: '© 2025 Паркомісця Львова. Усі права захищено.' (© 2025 Lviv Parking Spaces. All rights reserved.)

Рисунок 3.16 – Вхід в панель адміністратора

Сторінка входу адміністратора в системі призначена для авторизації адміністраторів, щоб вони могли отримати доступ до панелі керування. Після вводу коректних даних, користувача буде направлено на сторінку панелі адміністратора. Панель адміністратора показано на рисунку 3.17.

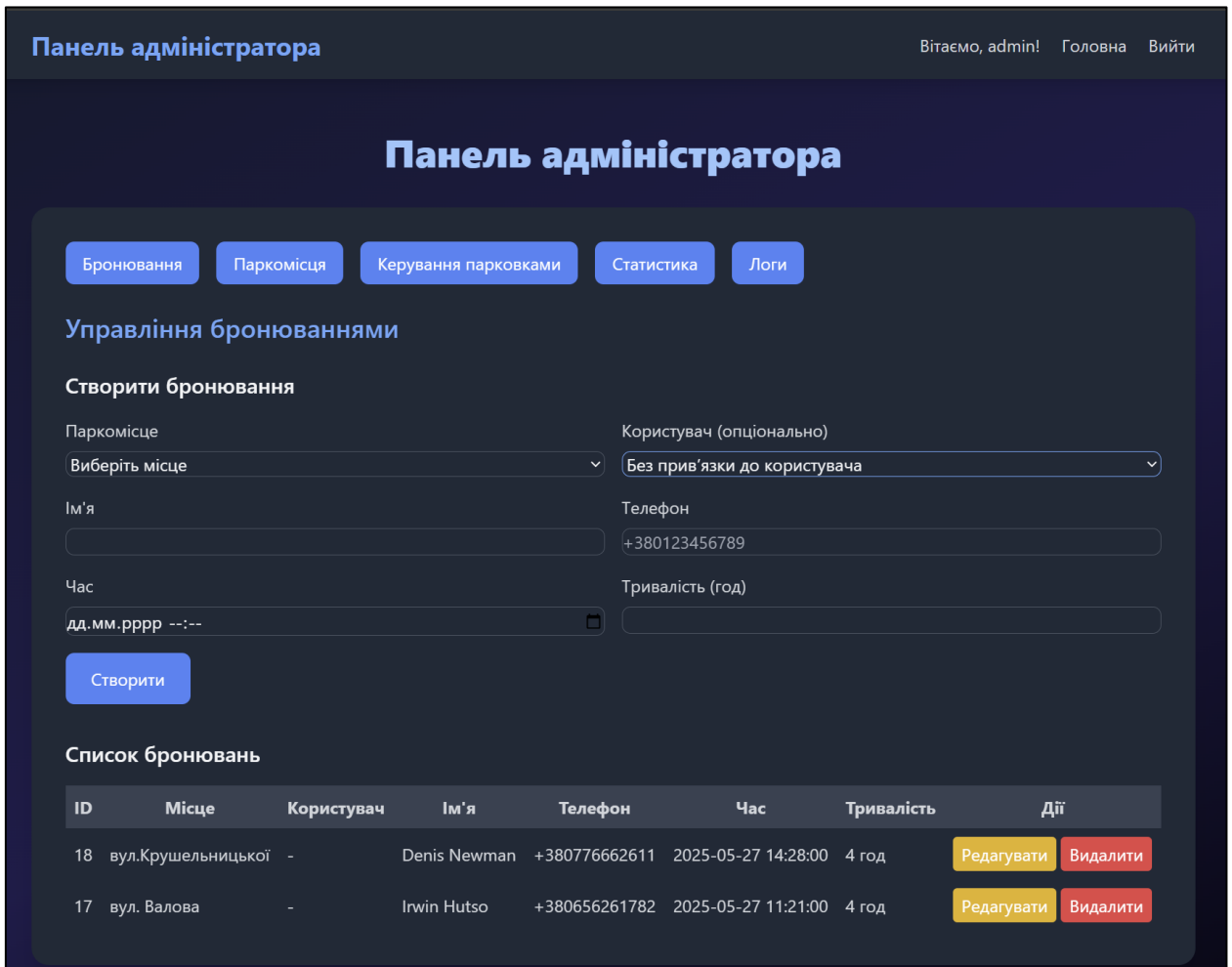


Рисунок 3.17 – Панель адміністратора

Основний інтерфейс складається з п'яти вкладок: "Бронювання", "Паркомісця", "Керування парковками", "Статистика" та "Логи". Вкладка "Бронювання" дозволяє створювати, редагувати та видаляти бронювання: у формі створення потрібно обрати паркомісце зі списку доступних, можна прив'язати бронювання до користувача, вказавши його ID, ім'я, телефон, час бронювання та тривалість, після чого натиснути "Створити", і якщо місце доступне, бронювання додається, а кількість вільних місць оновлюється.

Нижче форми відображається таблиця всіх бронювань із можливістю редагування через спливне вікно або видалення з підтвердженням.

На рисунку 3.18 показано вкладку «Паркомісця» панелі керування.

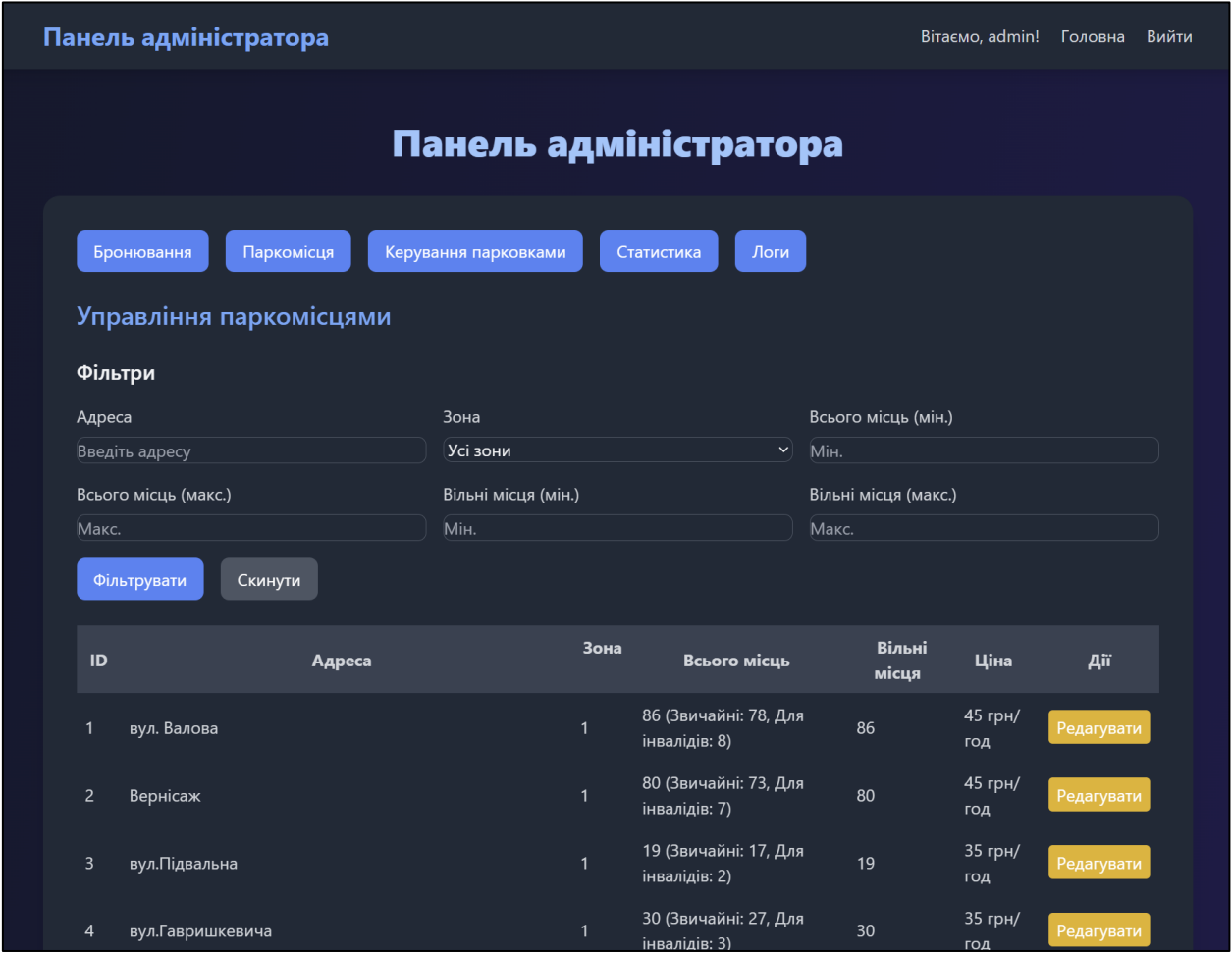


Рисунок 3.18 – Вкладка «Паркомісія»

Вкладка "Паркомісія" показує таблицю всіх парковок із фільтрами за адресою, зоною, кількістю місць та їхньою доступністю, а також дозволяє сортувати дані, натискаючи на заголовки колонок, і редагувати кількість місць через спливне вікно. Приклад редагування паркомісць показано на рисунку 3.19.

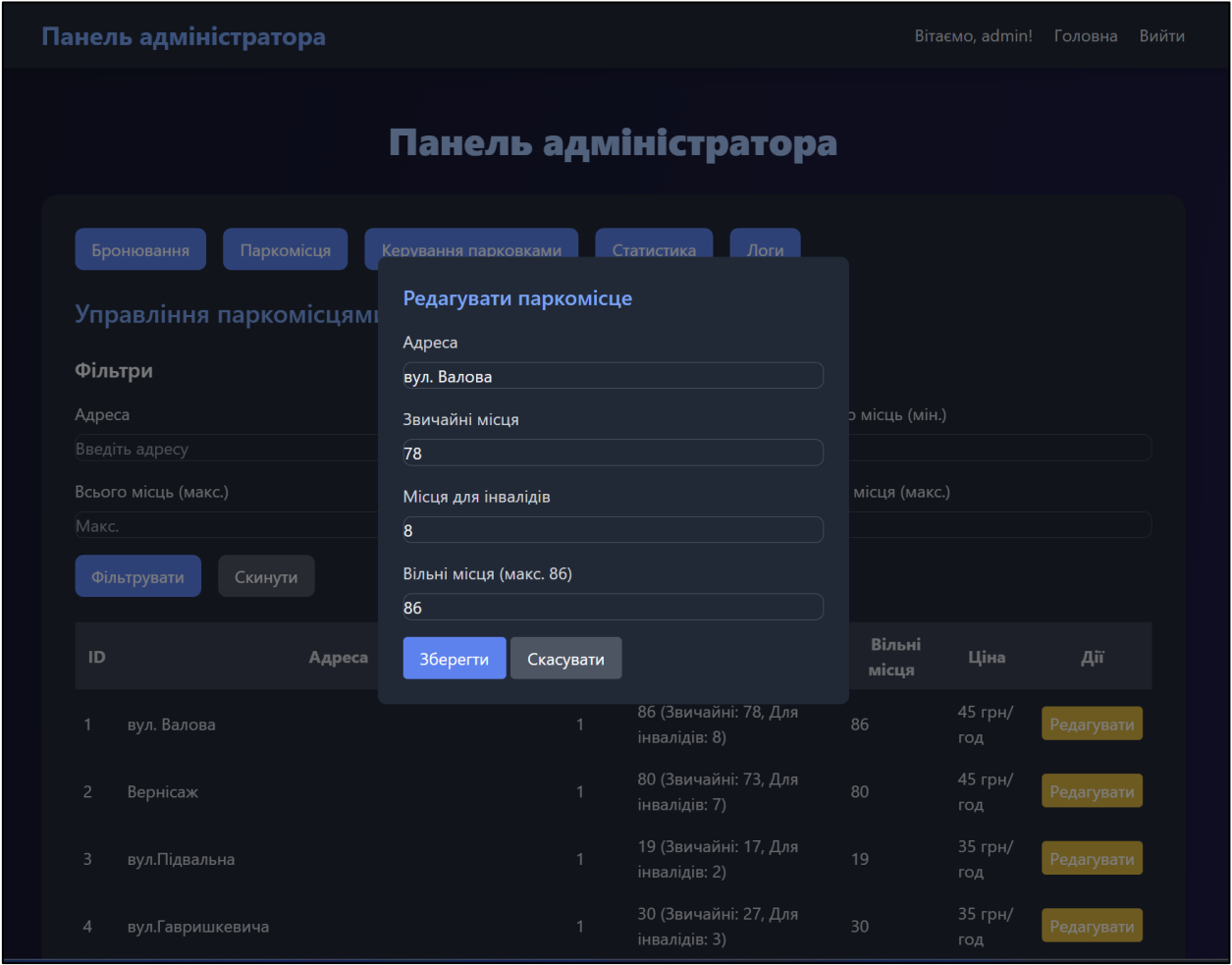


Рисунок 3.19 – Редагування паркомісця

Таким чином у адміністраторів є можливість редагувати всю інформацію для існуючих парковок, на випадок змін кількості вільних паркомісць, місць для інвалідів або адреси. Важливо, що не можна встановити більшу кількість вільних місць, ніж є наявних з врахуванням звичайних місць та місць для інвалідів.

На рисунку 3.20 показано вкладку «Керування парковками».

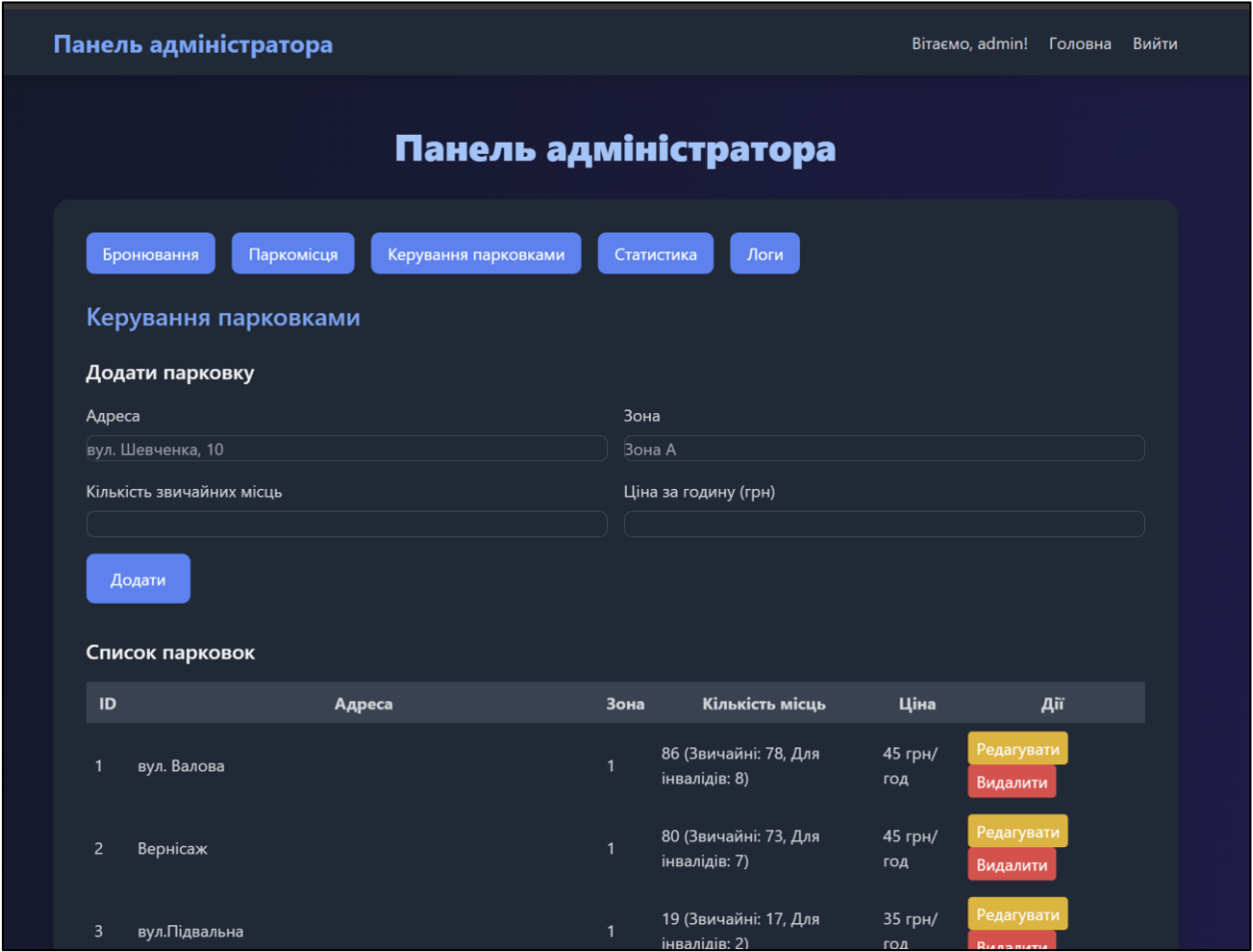


Рисунок 3.20 – Вкладка «Керування парковками»

У вкладці "Керування парковками" адміністратор може додати нову парковку, вказавши адресу, зону, кількість місць і ціну за годину, або редагувати чи видалити існуючі парковки, при цьому видалення потребує підтвердження, адже воно також видаляє всі пов’язані бронювання.

На рисунку 3.21 показано вкладку «Статистика».

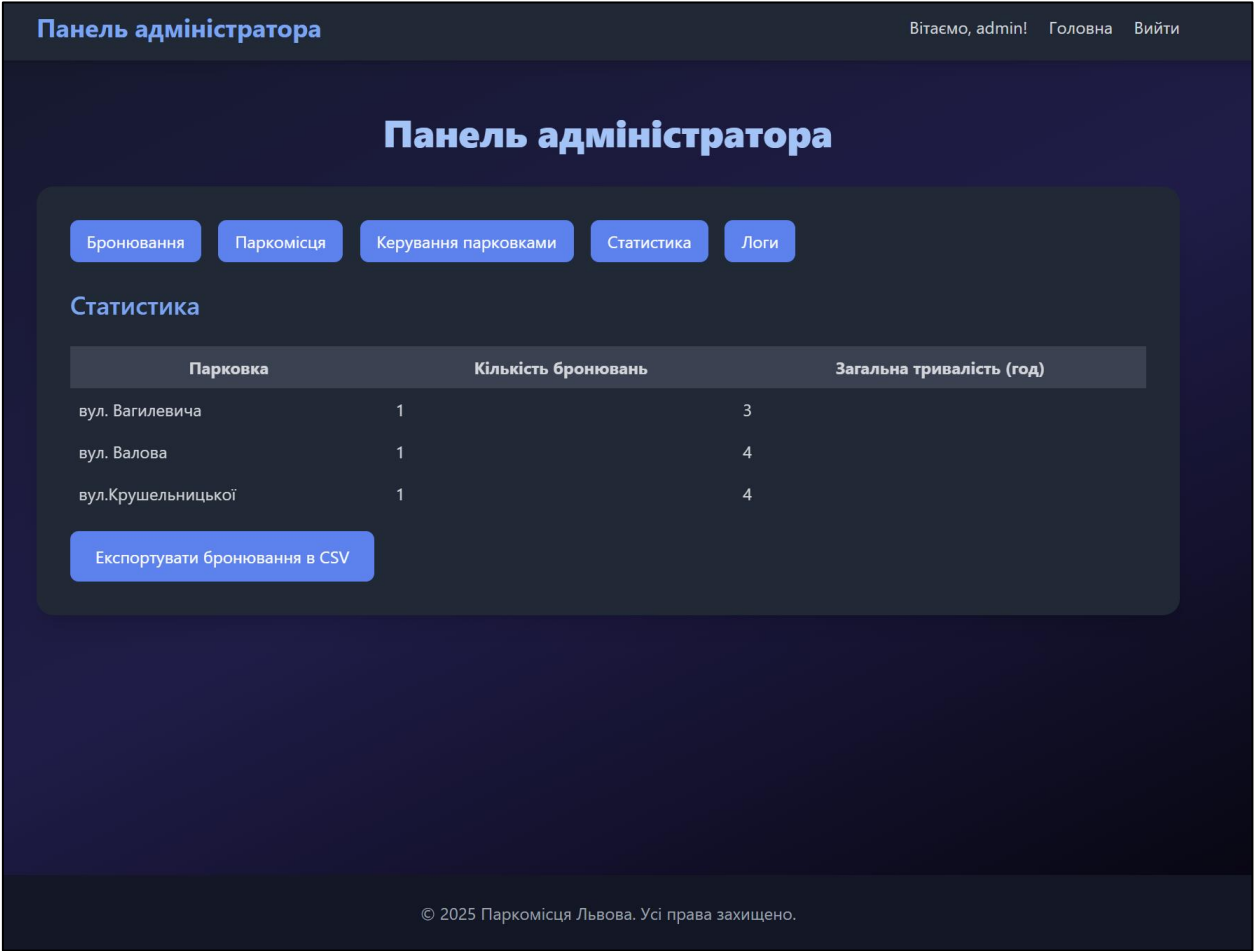


Рисунок 3.21 – Вкладка «Статистика»

Вкладка "Статистика" відображає таблицю з інформацією про кількість наявних бронювань і загальну тривалість по кожній зоні, а також пропонує кнопку для експорту бронювань у CSV-файл для подальшого аналізу. На рисунку 3.22 показано вкладку «Логи» яка дозволяє збирати інформацію про діяльність адміністраторів панелі.

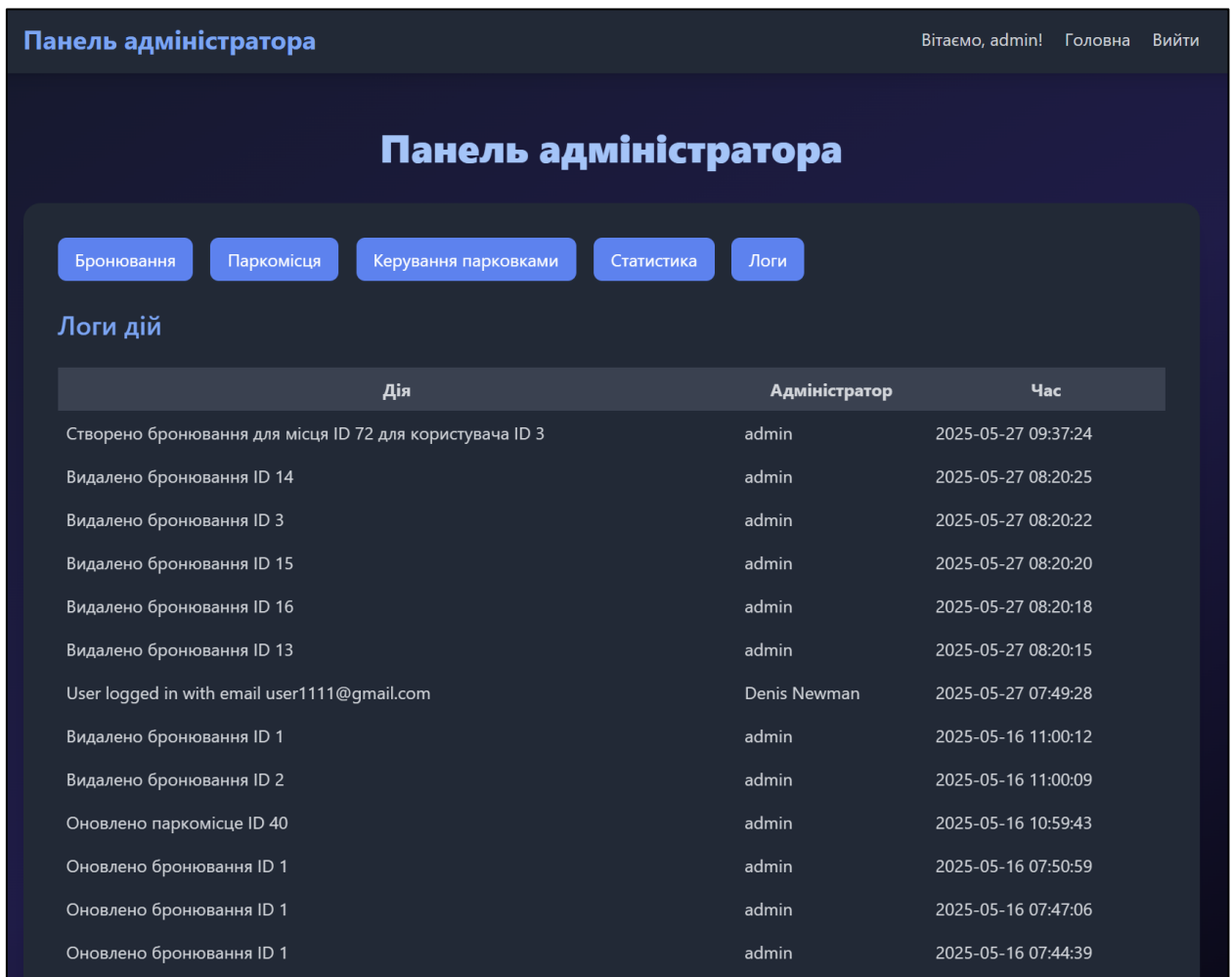


Рисунок 3.22 – Вкладка «Логи»

На вкладці "Логи" відображено останні 50 дій адміністраторів, таких як створення чи видалення бронювань, із зазначенням дії, імені адміністратора та часу. Усі дії, як-от створення чи видалення, супроводжуються повідомленнями про успіх або помилку у верхній частині сторінки, які автоматично зникають через кілька секунд завдяки анімації.

3.4 Висновки

У розділі було описано створення алгоритму функціонування системи, який забезпечує логічну послідовність операцій, таких як створення, редагування та видалення бронювань. Розроблено інформаційної системи з урахуванням обраних технологій, що включає реалізацію модулів для обробки

даних і взаємодії з базою даних. Також розроблено панель адміністратора, яка надає зручний інтерфейс для управління паркомісцями, бронюваннями та перегляду статистики, що сприяє ефективному адмініструванню системи. В результаті було отримано функціонуючу інформаційну систему для пошуку та бронювання паркомісць. Система пройшла тестування, яке підтвердило її стабільність і відповідність поставленим вимогам. Крім того, враховано можливість подальшого розширення функціоналу, таких як інтеграція з платіжними системами чи картами для зручнішого використання.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи було виконано всі поставлені завдання, що дозволило досягти основної мети дослідження — забезпечення швидкості та зручності процесу пошуку та бронювання паркомісць, скоротивши час на пошук вільних місць, завдяки розробці інформаційної системи з інтерактивною фільтрацією, моніторингом завантаженості та системою реєстрації користувачів.

Проведено аналіз проблематики паркування та існуючих систем управління паркомісцями. Під час аналізу вже існуючих систем для пошуку та бронювання паркомісць було виявлено основні вимоги до системи. До них можна віднести зручний та інтуїтивно зрозумілий користувацький інтерфейс, моніторинг зайнятості паркомісць, зручні інструменти для пошуку та фільтрації паркомісць.

Здійснено вибір оптимальних інформаційних технологій та IT-інфраструктури. Проведено підготовку IT-інфраструктури, створено віртуальний сервер та встановлено ОС Ubuntu Server. Також на сервері було налаштовано Apache для обробки веб-запитів, MariaDB в якості СКБД та реалізовано розроблену раніше структуру бази даних.

Структура бази даних, яку було розроблено в рамках бакалаврської кваліфікаційної роботи забезпечує ефективне зберігання даних про парковки, паркомісця, бронювання, користувачів і логи, а також реалізовано модуль статистики, що дозволяє аналізувати кількість бронювань і загальну тривалість використання парковок для стратегічного планування. Безпека даних забезпечена завдяки автентифікації адміністраторів і логуванню всіх дій у системі.

Адміністративна панель, розроблена в рамках роботи, дозволяє створювати, редагувати та видаляти бронювання, керувати паркомісцями (додавання, редагування, видалення парковок), а також переглядати

статистику та логи дій для аналізу ефективності використання паркувальних ресурсів.

Розроблена інформаційна система повністю виконала поставлені задачі, надаючи користувачам можливість зручного пошуку, фільтрації та бронювання паркомісць в реальному часі. Окрім того, також було розроблено систему реєстрації, авторизації та особистого кабінету для користувачів, та зберігання інформації про минулі бронювання.

Результатом дослідження стала інформаційна система пошуку та бронювання паркомісць, що виконує всі поставлені задачі та надає найкращий досвід користування.

За результатами даної роботи були зроблені доповіді на двох конференціях: LIV Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2025) та міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» з публікацією тез [1, 2].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шаповал А.С., Жуков С.О. Моделювання інформаційної системи пошуку та бронювання паркомісць: оптимізація паркування. *Всеукраїнська науково-технічна конференція підрозділів Вінницького національного технічного університету* (2025). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/author/submission/24160> (дата звернення: 22.05.2025).
2. Шаповал А.С., Жуков С.О. Інформаційна система пошуку та бронювання паркомісць. *Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)»*. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/25401> (дата звернення: 04.06.2025)
3. Що означає паркувальне місце? URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/25401> (дата звернення: 06.06.2025)
4. Intelligent parking systems URL: <https://www.sciencedirect.com/science/article/abs/pii/S0377221705002304> (дата звернення: 09.04.2025)
5. A reservation-based smart parking system URL: https://scholar.google.com/scholar?output=instlink&q=info:MSI91RbNRRAJ:scholar.google.com/&hl=en&as_sdt=0,5&scillfp=17675425605933664539&oi=lle (дата звернення: 09.04.2025)
6. Smart parking systems: comprehensive review based on various aspects URL: https://scholar.google.com/scholar?output=instlink&q=info:QXurG1GmFfgJ:scholar.google.com/&hl=en&as_sdt=0,5&as_vis=1&scillfp=17299812704242636571&oi=lle (дата звернення: 11.04.2025)
7. A survey of intelligent car parking system URL: <https://dovidka.biz.ua/yaka-riznitsya-mizh-parkovka-ta-stoyanka/> (дата звернення:

13.04.2025)

8. IOT Parking: Все про розумні системи паркування. URL: <https://hub.kyivstar.ua/articles/rozumni-systemy-parkuvannya-tendenciyi-ta-mozhlyvosti-dlya-ukrayinskyh-mist> (дата звернення: 14.04.2025)
9. Екологічна паркова як ефективне рішення для сталого розвитку міста URL: <https://eir.nuos.edu.ua/items/5792872b-f6a0-4465-9c85-5ecd28dc8336>
10. Класифікація автомобільних стоянок URL: <https://surl.li/grdpng> (дата звернення: 17.04.2025)
11. Вимоги до розміщення автомобільних стоянок. URL: <https://epl.org.ua/human-posts/vymogy-do-rozmishhennya-avtomobilnyh-stoyanok/> (дата звернення: 18.04.2025)
12. Система пошуку та бронювання паркомісць Parkopedia. URL: <https://www.parkopedia.com/> (дата звернення: 19.04.2025)
13. Система пошуку та бронювання паркомісць Parkme. URL: <https://www.parkme.com/accounts/login/?next=/> (дата звернення: 19.04.2025)
14. Система пошуку та бронювання паркомісць EasyPark URL: <https://easyparkgroup.com/> (дата звернення: 19.04.2025)
15. Система пошуку та бронювання паркомісць SpotHero. URL: <https://spothero.com/> (дата звернення: 22.04.2025)
16. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 : монографія. 6-те вид. Sebastopol : O'Reilly Media, 2022. 50–55 с.
17. Tatroe K., MacIntyre P., Lerdorf R. Programming PHP : монографія. 4-те вид. Sebastopol : O'Reilly Media, 2023. 80–85 с.
18. Freeman E., Freeman E. Head First JavaScript Programming: A Brain-Friendly Guide : монографія. 2-ге вид. Sebastopol : O'Reilly Media, 2022. 100–108 с.
19. Bailey L. SQL for Data Scientists: A Beginner's Guide for Building Datasets for Analysis : монографія. Hoboken : Wiley, 2021. 50–57 с.
20. Duckett J. HTML and CSS: Design and Build Websites : монографія. Hoboken : John Wiley & Sons, 2011. 120–127 с.

Додаток А
(обов'язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«__» _____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на бакалаврську кваліфікаційну роботу
ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ ТА БРОНЮВАННЯ
ПАРКОМІСЦЬ
08-34.БКР.026.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Сергій ЖУКОВ

«__» _____ 2025 р.

Розробив студент гр. 2ІСТ-216

_____ Анна ШАПОВАЛ

«__» _____ 2025 р.

Вінниця 2025

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____ 2025р., та індивідуальне завдання на БКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2025р.

2. Джерела розробки:

1) Що означає паркувальне місце? URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/25401>

2) Intelligent parking systems URL: <https://www.sciencedirect.com/science/article/abs/pii/S0377221705002304>

3. Мета і призначення роботи.

Метою дослідження є забезпечення швидкості та зручності процесу пошуку та бронювання паркомісць, скоротивши час на пошук вільних місць завдяки розробці інформаційної системи з інтерактивною фільтрацією, моніторингом завантаженості та системою реєстрації користувачів.

4. Вихідні дані для проведення робіт:

Набір даних про парковки та паркомісця у м.Львів:

https://www.google.com/maps/d/u/0/viewer?mid=1H0s277ON8BZqromCyEy7kiaNIbww_QQ&femb=1&ll=49.85026932274493%2C24.02263754999997&z=14

5. Методи дослідження:

Аналіз існуючих систем, синтез вимог до інформаційної системи, проектування схеми бази даних, проектування архітектури системи, розроблення інформаційної системи, тестування.

6. Етапи роботи і терміни їх виконання:

а) Аналіз об'єкта дослідження _____ – _____

б) ІТ-інфраструктура інформаційної системи для пошуку та бронювання паркомісць _____ – _____

в) Проектування та розроблення інформаційної системи для пошуку та бронювання паркомісць _____ – _____

г) Оформлення матеріалів до захисту БКР _____ – _____

7. Очікувані результати та порядок реалізації

Розроблена інформаційна система для пошуку та бронювання паркомісць.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)).

9. Порядок приймання роботи

Публічний захист «__» _____ 2025 р.

Початок розробки «__» _____ 2025 р.

Граничні терміни виконання БКР «__» _____ 2025 р.

Розробила студентка групи 2ІСТ-216 _____ Анна ШАПОВАЛ

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Інформаційна система пошуку та бронювання паркомісць»

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ: кафедра САІТ, ФІІТА, гр. 2ІСТ-216

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 0.60%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне):

- Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Віталій МОКІН, зав. каф. САІТ

_____ (підпис)

Євгеній КРИЖАНОВСЬКИЙ, доц. каф. САІТ

_____ (підпис)

Особа, відповідальна за перевірку _____ (підпис)

Сергій ЖУКОВ

З висновком експертної комісії ознайомлений(-на)

Керівник _____ (підпис)

Сергій ЖУКОВ, к.т.н., доц. каф. САІТ

Здобувач _____ (підпис)

Анна Шаповал

Додаток В

(довідниковий)

Фрагмент лістингу програми

Код файлу додатку map.js

```
window.ParkingApp = window.ParkingApp || {};
```

```
ParkingApp.initMap = function() {
```

```
  console.log('Initializing map...');
```

```
  try {
```

```
    ParkingApp.map = L.map('map').setView([49.8397, 24.0297], 14);
```

```
    console.log('Map initialized:', ParkingApp.map);
```

```
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
```

```
      attribution: '© <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>  
      contributors'
```

```
    }).addTo(ParkingApp.map);
```

```
    ParkingApp.parkingIcon = L.icon({
```

```
      iconUrl: 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-  
      icon-2x-blue.png',
```

```
      iconSize: [25, 41],
```

```
      iconAnchor: [12, 41],
```

```
      popupAnchor: [1, -34],
```

```
      shadowUrl: 'https://unpkg.com/leaflet@1.9.4/dist/images/marker-shadow.png',
```

```
      shadowSize: [41, 41]
```

```
    });
```

```
    console.log('Icon initialized:', ParkingApp.parkingIcon);
```

```
    ParkingApp.markers = [];
```

```
  } catch (error) {
```

```
    console.error('Error initializing map:', error);
```

```
    ParkingApp.map = null;
```

```
    ParkingApp.parkingIcon = null;
```

```
ParkingApp.markers = [];
```

Код файлу додатку filters.js

```
window.ParkingApp = window.ParkingApp || {};
```

```
ParkingApp.applyFilters = function() {
```

```
    const freeFilter = document.getElementById('free-filter').value;
```

```
    const addressFilter = document.getElementById('address-filter').value.toLowerCase();
```

```
    const zoneFilter = document.getElementById('zone-filter').value;
```

```
    const sectorFilter = document.getElementById('sector-filter').value;
```

```
    const street = document.getElementById('street-search').value.trim();
```

```
    // Якщо є пошук за вулицею, він має пріоритет
```

```
    if (street) {
```

```
        ParkingApp.searchByStreet();
```

```
        return;
```

```
    }
```

```
    // Стандартна фільтрація
```

```
    ParkingApp.filteredData = ParkingApp.parkingData.filter(row => {
```

```
        const hasFree = freeFilter === 'all' || (freeFilter === 'available' && parseInt(row.free) > 0);
```

```
        const matchesAddress = !addressFilter || (row.address || row.name ||  
        "").toLowerCase().includes(addressFilter);
```

```
        const matchesZone = zoneFilter === 'all' || row.zone === zoneFilter;
```

```
        const matchesSector = sectorFilter === 'all' || (row.sector || "").match(new  
        RegExp(`\\b${sectorFilter}\\b`));
```

```
        return hasFree && matchesAddress && matchesZone && matchesSector;
```

```
    });
```

```
    // Очистка результатів пошуку
```

```
    const resultsDiv = document.getElementById('search-results');
```

```
    resultsDiv.classList.add('hidden');
```

```
    if (ParkingApp.searchCircle) {
```

```
        ParkingApp.map.removeLayer(ParkingApp.searchCircle);
```

```
        ParkingApp.searchCircle = null;
```

```
    }
```

```

ParkingApp.currentPage = 1; // Скинути на першу сторінку
ParkingApp.renderTableAndMarkers();
};

ParkingApp.searchByStreet = async function() {
  const street = document.getElementById('street-search').value.trim();
  const resultsDiv = document.getElementById('search-results');
  const message = document.getElementById('search-message');
  const freeSpots = document.getElementById('free-spots');
  const occupiedSpots = document.getElementById('occupied-spots');

  // Очистка попередніх результатів
  resultsDiv.classList.add('hidden');
  if (ParkingApp.searchCircle) {
    ParkingApp.map.removeLayer(ParkingApp.searchCircle);
    ParkingApp.searchCircle = null;
  }

  if (!street) {
    ParkingApp.applyFilters();
    return;
  }

  const coords = await ParkingApp.geocodeStreet(street);
  if (!coords) {
    resultsDiv.classList.remove('hidden');
    message.innerHTML = 'Вулицю не знайдено. Спробуйте ще раз.';
    freeSpots.innerHTML = '';
    occupiedSpots.innerHTML = '';
    ParkingApp.applyFilters();
    return;
  }

  // Фільтрація паркомісць у радіусі 200 м

```

```

let totalFree = 0;
let totalOccupied = 0;
ParkingApp.filteredData = ParkingApp.parkingData.filter(row => {
  const lat = parseFloat(row.latitude);
  const lon = parseFloat(row.longitude);
  if (isNaN(lat) || isNaN(lon)) return false;
  const distance = ParkingApp.haversineDistance(coords.lat, coords.lon, lat, lon);
  if (distance <= 200) {
    totalFree += parseInt(row.free) || 0;
    totalOccupied += (parseInt(row.count) || 0) - (parseInt(row.free) || 0);
    return true;
  }
  return false;
});

// Оновлення результатів
resultsDiv.classList.remove('hidden');
message.innerHTML = `Результати для "${street}" (радіус 200 м):`;
freeSpots.innerHTML = `Вільних місць: ${totalFree}`;
occupiedSpots.innerHTML = `Зайнятих місць: ${totalOccupied}`;

// Оновлення карти
ParkingApp.map.setView([coords.lat, coords.lon], 16);
ParkingApp.searchCircle = L.circle([coords.lat, coords.lon], {
  radius: 200,
  color: '#60a5fa',
  fillColor: '#60a5fa',
  fillOpacity: 0.2
}).addTo(ParkingApp.map);

ParkingApp.currentPage = 1; // Скинути на першу сторінку
ParkingApp.renderTableAndMarkers();
};

```

Код файлу додатку map.js:

```
window.ParkingApp = window.ParkingApp || {};
```

```
ParkingApp.init = function() {
```

```
  // Ініціалізація даних
```

```
  ParkingApp.parkingData = window.parkingData || [];
```

```
  ParkingApp.filteredData = [...ParkingApp.parkingData];
```

```
  ParkingApp.errorMessage = window.errorMessage || '';
```

```
  ParkingApp.sortField = 'address';
```

```
  ParkingApp.sortDirection = 'asc';
```

```
  // Ініціалізація карти
```

```
  ParkingApp.initMap();
```

```
  // Ініціалізація пагінації
```

```
  ParkingApp.initPagination();
```

```
  // Початковий рендеринг
```

```
  try {
```

```
    if (!Array.isArray(ParkingApp.parkingData) || ParkingApp.parkingData.length === 0) {
```

```
      console.warn('Дані порожні або некоректні');
```

```
      document.getElementById('parking-table').innerHTML = '<tr><td colspan="7" class="text-center text-gray-400">Дані відсутні</td></tr>';
```

```
      throw new Error('Дані відсутні');
```

```
    }
```

```
    ParkingApp.renderTableAndMarkers();
```

```
  } catch (error) {
```

```
    console.error('Помилка ініціалізації:', error);
```

```
  }
```

```
  // Додавання слухачів для фільтрів
```

```
  document.getElementById('free-filter').addEventListener('change', ParkingApp.applyFilters);
```

```
  document.getElementById('address-filter').addEventListener('input', ParkingApp.applyFilters);
```

```
  document.getElementById('zone-filter').addEventListener('change', ParkingApp.applyFilters);
```

```

document.getElementById('sector-filter').addEventListener('change', ParkingApp.applyFilters);
document.getElementById('street-search').addEventListener('input', () => {
    clearTimeout(window.searchTimeout);
    window.searchTimeout = setTimeout(ParkingApp.searchByStreet, 500);
});

// Додавання слухачів для сортування
document.querySelectorAll('th[data-sort]').forEach(th => {
    th.addEventListener('click', () => {
        const field = th.getAttribute('data-sort');
        ParkingApp.sortTable(field);
    });
});

// Додавання слухачів для пагінації
document.getElementById('prev-page').addEventListener('click', ParkingApp.prevPage);
document.getElementById('next-page').addEventListener('click', ParkingApp.nextPage);
});

// Запуск ініціалізації після завантаження DOM
document.addEventListener('DOMContentLoaded', ParkingApp.init);

```

Код файлу додатку render.js:

```

window.ParkingApp = window.ParkingApp || {};

ParkingApp.renderTableAndMarkers = function() {
    console.log('Rendering, filteredData length:', ParkingApp.filteredData.length);

    // Очистка маркерів, якщо карта ініціалізована
    if (ParkingApp.map && Array.isArray(ParkingApp.markers)) {
        ParkingApp.markers.forEach(marker => {
            try {
                ParkingApp.map.removeLayer(marker);
            } catch (e) {

```



```

        console.warn('Error removing marker:', e);
    }
});
ParkingApp.markers.length = 0;
} else {
    console.warn('Map not initialized, skipping marker cleanup');
    ParkingApp.markers = [];
}

const tableBody = document.getElementById('parking-table');
console.log('Table body:', tableBody);
tableBody.innerHTML = "";

const totalPages = Math.ceil(ParkingApp.filteredData.length / ParkingApp.itemsPerPage);
const startIndex = (ParkingApp.currentPage - 1) * ParkingApp.itemsPerPage;
const endIndex = startIndex + ParkingApp.itemsPerPage;
const paginatedData = ParkingApp.filteredData.slice(startIndex, endIndex);
console.log('Paginated data length:', paginatedData.length, 'Page:', ParkingApp.currentPage, 'Total pages:', totalPages);

const prevButton = document.getElementById('prev-page');
const nextButton = document.getElementById('next-page');
const pageInfo = document.getElementById('page-info');
prevButton.disabled = ParkingApp.currentPage === 1;
nextButton.disabled = ParkingApp.currentPage === totalPages || totalPages === 0;
pageInfo.textContent = `Сторінка ${ParkingApp.currentPage} з ${totalPages || 1}`;

try {
    if (!Array.isArray(ParkingApp.filteredData) || ParkingApp.filteredData.length === 0) {
        console.warn('No data to render');

        tableBody.innerHTML = '<tr><td colspan="7" class="text-center text-gray-400">Дані відсутні</td></tr>';

        return;
    }
}

```

```

let validMarkers = 0;
ParkingApp.filteredData.forEach(row => {
  const hasCoords = row.latitude && row.longitude;
  const lat = hasCoords ? parseFloat(row.latitude) : null;
  const lng = hasCoords ? parseFloat(row.longitude) : null;
  const isValid = hasCoords && !isNaN(lat) && !isNaN(lng);

  console.log('Row:', {
    address: row.address,
    latitude: row.latitude,
    longitude: row.longitude,
    hasCoords: hasCoords,
    isValid: isValid
  });

  if (isValid && ParkingApp.map && ParkingApp.parkingIcon) {
    try {
      const marker = L.marker([lat, lng], { icon: ParkingApp.parkingIcon
    }).addTo(ParkingApp.map);

      const popupContent = `
        <b>${row.address || row.name || 'Без назви'}</b><br>
        Зона: ${row.zone || 'Невідомо'}<br>
        Сектор: ${row.sector || 'Невідомо'}<br>
        Ціна: ${row.price || 'Невідомо'} грн/год<br>
        Всього місць: ${row.count || 'Невідомо'}<br>
        Вільних місць: ${row.free || '0'}<br>
        ${row.free > 0 ?
          `<a
href="reserve.php?spot_id=${encodeURIComponent(row.id)}&address=${encodeURIComponent(row.a
ddress || row.name || ")}&zone=${encodeURIComponent(row.zone || ")}" class="inline-block mt-2 px-2
py-1 bg-blue-500 text-white rounded hover:bg-blue-600">Забронювати</a>` :
          `<span style="color: red;">Місць немає</span>`
        }
      `;

      marker.bindPopup(popupContent);
      ParkingApp.markers.push(marker);
    }
  }
});

```

```

        validMarkers++;
    } catch (e) {
        console.warn('Error adding marker:', e);
    }
}

});

console.log('Valid markers added:', validMarkers);

let validRows = 0;
paginatedData.forEach((row, index) => {
    const hasCoords = row.latitude && row.longitude;
    const lat = hasCoords ? parseFloat(row.latitude) : null;
    const lng = hasCoords ? parseFloat(row.longitude) : null;
    const isValid = hasCoords && !isNaN(lat) && !isNaN(lng);

    if (isValid) {
        tableBody.innerHTML += `
            <tr class="${index % 2 === 0 ? 'bg-gray-700' : 'bg-gray-800'} hover:bg-gray-600
transition">
                <td>${row.address || row.name || 'Без назви'}</td>
                <td>${row.zone || 'Невідомо'}</td>
                <td>${row.sector || 'Невідомо'}</td>
                <td>${row.price || 'Невідомо'}</td>
                <td>${row.count || 'Невідомо'}</td>
                <td>${row.free || '0'}</td>
                <td>
                    ${row.free > 0 ?
                        `<a
href="reserve.php?spot_id=${encodeURIComponent(row.id)}&address=${encodeURIComponent(row.a
ddress || row.name || "")}&zone=${encodeURIComponent(row.zone || "")}" class="px-2 py-1 bg-blue-500
text-white rounded hover:bg-blue-600">Забронювати</a>` :
                        `<span class="text-red-500">Місць немає</span>`}
                </td>
            </tr>
        `;
    }
});

```

```

        validRows++;
    }
});
console.log('Valid rows added:', validRows);

if (ParkingApp.filteredData.length > 0 && ParkingApp.map) {
    const bounds = ParkingApp.filteredData
        .filter(row => row.latitude && row.longitude && !isNaN(parseFloat(row.latitude)) &&
!isNaN(parseFloat(row.longitude)))
        .map(row => [parseFloat(row.latitude), parseFloat(row.longitude)]);
    if (bounds.length > 0) {
        console.log('Fitting bounds:', bounds);
        ParkingApp.map.fitBounds(bounds, { padding: [50, 50] });
    } else {
        console.warn('No valid bounds to fit');
    }
} else {
    console.warn('Map not available for bounds fitting');
}
} catch (error) {
    console.error('Помилка обробки даних:', error);
    tableBody.innerHTML = `<tr><td colspan="7" class="text-center text-gray-400">Помилка
завантаження даних: ${ParkingApp.errorMessage || 'Невідома помилка'}</td></tr>`;
}
};

```

Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ ТА БРОНЮВАННЯ
ПАРКОМІСЦЬ**

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«__» _____ 2025 р.

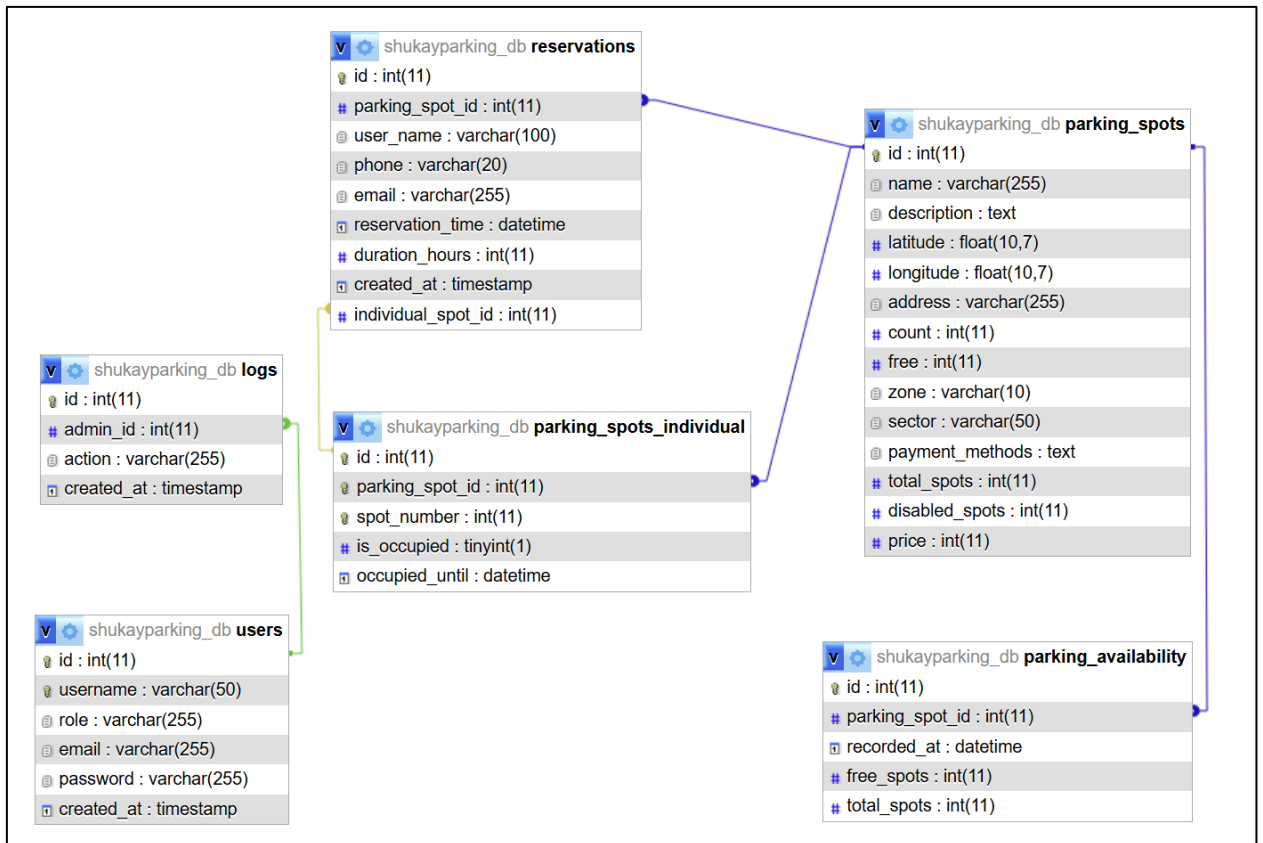


Рисунок Г.1 – ER діаграма бази даних системи бронювання паркомісць



Рисунок Г.2 – Блок-схема роботи алгоритму бронювання паркомісць

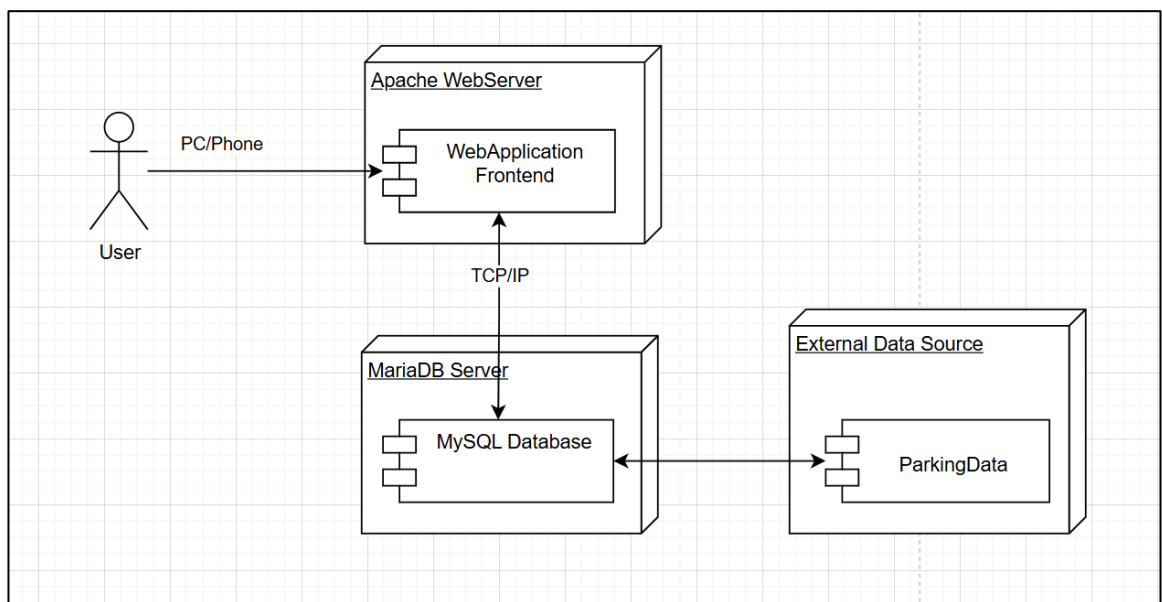


Рисунок Г.3 – Діаграма розгортання веб-системи пошуку та бронювання паркомісць

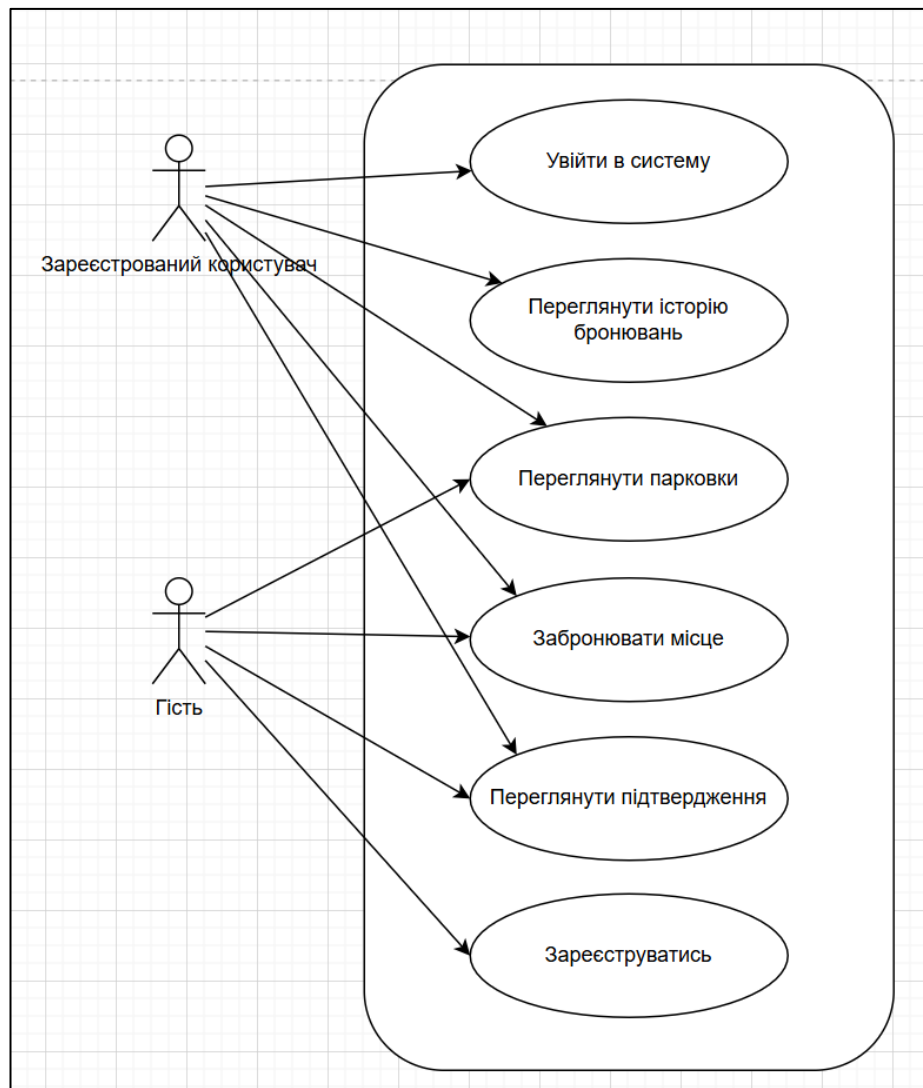


Рисунок Г.4 – Діаграма варіантів використання для гостя та зареєстрованого користувача



Рисунок Г.5 – Діаграма варіантів використання для адміністратора

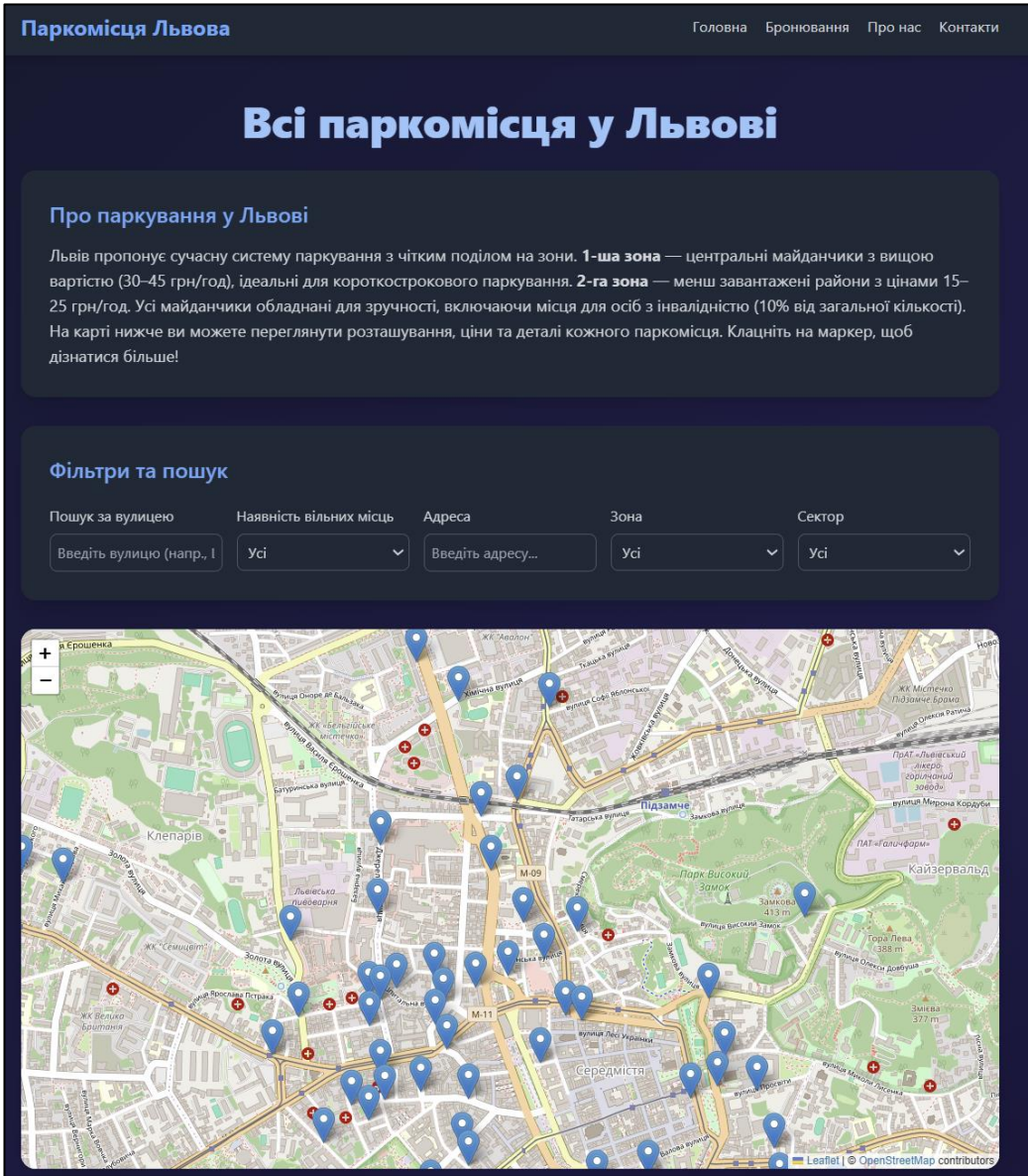


Рисунок Г.6 – Інтерфейс інформаційної системи