

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розроблення системи автоматизації для оптимізації паркування  
автотранспорту на базі технології IoT  
(тема)

Виконав:

студент 4 курсу, групи АКТАКІТ-20-2

Жердєв М. В.

(прізвище, ініціали)

Спеціальності 151 Автоматизація та  
комп'ютерно-інтегровані технології

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

Освітня програма Автоматизація та  
комп'ютерно-інтегровані технології

(повна назва освітньої програми)

Керівник доц. Хрустальова С. В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАР

(підпис)

Невлюдов І. Ш.

(прізвище, ініціали)

2024 р.

# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та  
робототехніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології  
(код і повна назва)

Тип програми Освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Жердеву Михайлу Валерійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи автоматизації для оптимізації паркування  
автотранспорту на базі технології IoT

затверджена наказом університету від 03.06.2024 р. № 544 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 21.06.2024 р.

3. Вихідні дані система оптимізації процесу паркування автотранспорту на базі  
технології IoT

3.1 Призначення і цілі розробки: веб-сайт призначений для оптимізації паркування  
автотранспорту на базі технології IoT;

3.2 Середовище та мова розробки: Visual Studio Code. HTML, C++, JavaScript;

3.3 Дані мають передаватися безконтактно;

3.4 Архівування даних на локальне сховище;

3.5 Сумісність з ОС Windows XP, 7, 8, 10, 11, Android, IOS.

4. Перелік питань, що потрібно опрацювати в роботі

4.1 Аналіз існуючих систем оптимізації паркування автотранспорту;

4.2 Аналіз існуючих технологій які використовують у цій галузі: методів, передачі даних,  
систем контролю, програмних засобі.

4.3 Розробка структурної схеми системи оптимізації паркування автотранспорту;

4.4 Створення програмного забезпечення;

4.5 Розробка автоматизованої системи контролю паркувальних місць для оптимізації  
системи паркування автотранспорту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) \_\_\_\_\_  
Демонстраційний матеріал, представлений у форматі Power Point (\*.ppt) формату А4

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Огляд та аналіз існуючих систем автоматизації для оптимізації паркування автотранспорту	08.05.2024	Виконано
2	Розробка структурної схеми системи автоматизації для оптимізації паркування автотранспорту	16.05.2024	Виконано
3	Вибір компонентів системи та обґрунтування	20.05.2024	Виконано
4	Вибір програмного забезпечення	25.05.2024	Виконано
5	Розроблення програмної частини	31.05.2024	Виконано
7	Оформлення пояснювальної записки	10.06.2024	Виконано
6	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	20.06.2024	Виконано
8	Подання роботи на рецензію	20.06.2024	Виконано
9	Подання роботи на підпис зав. кафедри	21.06.2024	Виконано
10	Подання атестаційної роботи в ЕК	24.06.2024	Виконано

Дата видачі завдання 01.04.2024 р.

Студент М. Жердев Жердев М. В.  
(підпис)

Керівник роботи доц. Хрустальова С. В.  
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував незголену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

“16” червня 2024 р.



Жердев М. В.

## РЕФЕРАТ

Пояснювальна записка: 68 с., 47 рис., 2 табл., 1 дод., 12 джерел.

### УЛЬТРАЗВУКОВИЙ ДАТЧИК, ARDUINO, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПАРКУВАННЯ.

Мета роботи – підвищення ефективності управління паркувальними місцями на підземній парковці.

Об'єкт розробки – процес моніторингу та бронювання паркувальних місць у програмному виді.

Предмет розробки – програмний засіб для моніторингу та бронювання паркувальних місць.

В кваліфікаційній роботі проведено аналіз сучасних системи моніторингу паркувальних місць, методи передачі даних за допомогою Wi-Fi модулів. Розглянуто технології вимірювання заповнюваності паркувальних місць за допомогою ультразвукових датчиків.

Розроблено структурну схему системи моніторингу, що включає ультразвукові датчики для визначення наявності автомобіля на паркувальному місці, контролер Arduino для збору та обробки даних, а також Wi-Fi модуль ESP8266 для передачі даних на сервер. Проведено вибір необхідних електронних компонентів, спроектовано схему підключення, та розроблено алгоритм роботи системи.

Розроблено програмний засіб для відображення стану паркувальних місць у реальному часі та бронювання місць користувачами. Програмний засіб дозволяє користувачам бачити доступні місця на парковці, забронювати місце заздалегідь, та отримувати інформацію про зміну стану заброньованих місць.

Розглянуто питання охорони праці та безпеки при розгортанні та

експлуатації системи.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", а саме п.9.3 "Забезпечити доступність дорожньотранспортної інфраструктури, яка базується на використанні інноваційних технологій, зокрема через розширення форм участі держави у різних інфраструктурних проектах".

## ABSTRACT

Explanatory note: 68 pages, 47 figures, 2 tables, 1 appendix, 12 sources.

ULTRASONIC SENSOR, ARDUINO, SOFTWARE, PARKING.

The purpose of the work is to improve the efficiency of parking space management in the underground car park.

The object of development is the process of monitoring and booking parking spaces in software.

The subject of the development is a software tool for monitoring and booking parking spaces.

The qualification work analyses modern parking space monitoring systems and methods of data transmission using Wi-Fi modules. The technologies for measuring the occupancy of parking spaces using ultrasonic sensors are considered.

A structural diagram of the monitoring system has been developed, including ultrasonic sensors for determining the presence of a car in a parking space, an Arduino controller for data collection and processing, and an ESP8266 Wi-Fi module for data transmission to the server. We selected the necessary electronic components, designed a wiring diagram, and developed an algorithm for the system.

A software tool was developed to display the status of parking spaces in real time and to allow users to book spaces. The software tool allows users to see the available parking spaces, book a space in advance, and receive information about changes in the status of booked spaces.

The article addresses the issues of occupational health and safety during the deployment and operation of the system.

Also, the results of the work can be attributed to Sustainable Development

Goal 9 "Industry, Innovation and Infrastructure", namely, 9.3 "Ensure the availability of road transport infrastructure based on the use of innovative technologies, in particular through expanding forms of government participation in various infrastructure projects".



## ЗМІСТ

Перелік умовних скорочень.....	8
Вступ.....	9
1 Огляд та аналіз існуючих систем оптимізації паркування автотранспорту.	12
1.1 Огляд і класифікація систем паркінгу.....	12
1.2 Проблеми автомобільних парковок.....	13
1.3 Аналіз існуючих систем керування паркінгом.....	14
1.3.1 Система Park System.....	14
1.3.2 Система керування паркінгом AFAPARK.....	17
1.3.3 Автоматизована парковка FidPark.....	20
1.4 Висновки до розділу.....	26
2 Розроблення структурної схеми та алгоритму роботи системи автоматизації для оптимізації паркування автотранспорту.....	28
2.1 Загальний опис системи.....	28
2.2 Структурна схема підсистеми та алгоритм її роботи. ....	29
2.3 Вибір компонентів системи автоматизації для оптимізації паркування автотранспорту.....	32
2.3.1 Вибір ультразвукового датчика для системи, його характеристики	32
2.3.2 Вибір IoT-шлюза для забезпечення бездротового зв'язку.....	34
2.4 Розрахунок передавальних функцій компонентів системи.....	36
2.5 Висновки до розділу.....	42
3 Розроблення веб-сайту системи автоматизації для оптимізації паркування автотранспорту.....	44
3.1 Вибір середовища програмування та програмного забезпечення.....	44
3.2 Вибір мов програмування.....	45
3.3 Розроблення програмної частини.....	46
3.3.1 Функціонал сайту, який потрібно реалізувати.....	46
3.3.2 Розробка сторінки авторизації та реєстрації.....	48

3.3.3 Розробка всіх вкладок та інтеграції з картами Google .....	50
3.3.4 Розробка інформаційного розділу для загальної інформації .....	52
3.3.5 Розробка сторінок окремих паркінгів .....	53
3.3.6 Розробка системи бронювання.....	56
3.3.7 Передача даних з датчиків і відображення на сайті .....	60
3.4 Висновки до розділу.....	62
4 Охорона праці .....	64
4.1 Здоров'я та безпека при роботі за комп'ютером .....	64
Висновки .....	66
Перелік посилань.....	67
Додаток А Лістинг програми .....	69
Додаток Б Демонстраційні матеріали у вигляді презентації .....	90

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АПС – автоматизована паркувальна система

АЦП – аналого-цифровий перетворювач

ДТП – дорожньо-транспортна пригода

ОЗП – оперативна запам'ятовувальна пам'ять

СКС – структурована кабельна система

API – Application Programming Interface (Інтерфейс програмування додатків)

GLONASS – Global Navigation Satellite System (Глобальна навігаційна супутникова система)

GPS – Global Positioning System (Система глобального позиціонування)

ІоТ – Internet of Things (Інтернет речей)

LAN – Local Area Network (Локальна мережа)

TCP – Transmission Control Protocol (Протокол керування передачею)

## ВСТУП

Актуальність теми. Питання організації паркувальних місць є однією з найбільш гострих проблем у транспортних системах великих міст України. Попри щорічне зростання кількості автомобілів, кількість паркінгів збільшується значно повільнішими темпами. Крім того, більшість існуючих автостоянок не мають автоматизованих систем керування та моніторингу, що значно ускладнює процес паркування для водіїв.

Неавтоматизовані паркінги створюють численні незручності як для відвідувачів, так і для адміністрації. Водіям доводиться кружляти темними підземними або багатоповерховими паркінгами або об'їжджати цілі квартали, шукаючи вільне місце. Це викликає роздратування та занепокоєння за збереження автомобіля. Через надмірно вузькі проїзди трапляються дрібні ДТП, що може призвести до втрати клієнтів та додаткових проблем. Адміністрації паркінгів для уникнення заторів та елементарного контролю доводиться виділяти регулювальників на в'їзді, що потребує додаткових витрат на персонал.

У Європі автоматизовані паркінги з різноманітними системами автоматизації вже тривалий час є звичним явищем і міцно утвердилися на ринку. Жоден сучасний європейський паркінг не обходиться без впровадження тих чи інших автоматизованих систем керування та моніторингу в тій чи іншій мірі.

Очевидно, що без систем інформування про наявність вільних місць на паркінгу, водіям доводиться об'їжджати його кола в пошуках місця. Не знайшовши вільного паркомісця, водій, імовірно, залишить паркінг. Після такого негативного досвіду, користувач надалі намагатиметься залишати автомобіль за межами паркінгу, щоб не витратити час на пошуки вільного місця. Неорганізована автостоянка без жодних систем управління нагадує транспортний вузол міста без світлофорів.

Впровадження цифрових інформаційних табло дозволяє водіям ще на в'їзді орієнтуватися щодо наявності та розташування вільних паркомісць. На паркінгах без контрольованого в'їзду кількість вільних місць може відстежуватися за допомогою камер відеоспостереження та відповідного програмного забезпечення для відображення цієї інформації на картах.

Для розкриття повного потенціалу паркінгової інфраструктури в Україні необхідно її технологічне та інформаційне вдосконалення, впровадження сучасних систем контролю. Це значно полегшить роботу адміністрації автостоянок та зробить процес паркування більш зручним для користувачів.

Впровадження систем позиціонування та локалізації є одним з ключових і актуальних напрямків технологічного та процесного вдосконалення в таких галузях, як безпілотні транспортні засоби та автоматизовані системи паркування автомобілів.

У повсякденному житті ми постійно користуємося глобальними навігаційними системами, такими як GPS та GLONASS. Їх функціонування забезпечують спеціальні супутники на геостаціонарній орбіті, прив'язані до певної точки в глобальній системі координат. Пристрої, наприклад навігатори або смартфони з підтримкою цих систем, отримують сигнал від супутників і визначають своє місцезнаходження за географічними координатами на основі завантажених карт місцевості. Точність глобальних навігаційних систем достатня для великих територій – похибка вимірюється кілометрами або сотнями метрів в масштабах кварталів чи міст. Проте іноді вона може сягати 5-20 метрів залежно від перешкод та якості сигналу.

На відміну від глобальних навігаційних систем, системи локального позиціонування позбавлені їхніх недоліків і мають очевидні переваги. Їх доцільно використовувати для побудови систем визначення місцезнаходження та зв'язку в межах обмежених територій і приміщень, зокрема підземних паркінгів, які є предметом нашого інтересу. Саме в таких

умовах локальні системи позиціонування демонструють найбільшу ефективність.

Системи локального позиціонування призначені для визначення місцезнаходження людей, техніки та інших фізичних об'єктів в межах обмежених територій або всередині будівель з високою точністю – від 5 метрів до 0,1 метра. Такі комплекси, як правило, складаються з стаціонарних базових станцій, пристроїв на рухомих об'єктах, блоків керування, серверів та спеціалізованого програмного забезпечення. Конкретна структура системи може варіюватись залежно від специфічних потреб та вимог застосування.

Тому тема проекту, яка присвячена розробленню системи автоматизації для оптимізації паркування автотранспорту на базі технології IoT є актуальною.

Мета роботи – підвищення ефективності управління паркувальними місцями на підземній парковці.

Об'єкт розробки – процес моніторингу та бронювання паркувальних місць у програмному виді.

Предмет розробки – програмний засіб для моніторингу та бронювання паркувальних місць.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести огляд та аналіз існуючих методів, засобів та систем оптимізації паркування автотранспорту;
- розробити структурну схему системи;
- обрати компоненти для реалізації системи;
- обрати програмне забезпечення;
- реалізувати систему у вигляді програмного засобу;
- оформити пояснювальну записку, використовуючи методичні вказівки [1] та ДСТУ 3008-15 [2].

# 1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ОПТИМІЗАЦІЇ ПАРКУВАННЯ АВТОТРАНСПОРТУ

## 1.1 Огляд і класифікація систем паркінгу

Автомобільна парковка – це спеціально відведена ділянка на проїжджій частині дороги, вулиці населеного пункту або прилеглий до них території, призначена для тимчасової стоянки транспортних засобів.

Наявність паркувальних місць є невід'ємним компонентом житлових комплексів, офісних і адміністративних будівель, торгово-розважальних центрів тощо. При проектуванні практично будь-якого нового об'єкта забудови обов'язково вирішується питання організації місць для розміщення автомобілів відвідувачів.

Автоматизована паркувальна система (АПС) складається із спеціалізованого обладнання та програмного забезпечення. Вона дозволяє автомобілям в'їжджати та виїжджати з паркінгу, а також здійснює збір оплати за послуги паркування в автоматичному режимі з мінімальним залученням людського ресурсу. До спеціалізованого обладнання відносяться в'їзні та виїзні термінали, шлагбауми для пропуску транспорту, автоматичні або ручні каси для оплати, а також система розпізнавання номерних знаків на базі відеокамер та плат відеозахоплення.

Інтелектуальна система паркування (IoT Parking) – складається з спеціалізованого обладнання та програмного забезпечення. Вона дозволяє автомобілям в'їжджати на паркінг, після чого здійснює повністю автоматичне паркування без участі водія. Система автоматично розраховує вартість за час перебування автомобіля на парковці. До спеціалізованого обладнання відносяться: в'їзні та виїзні термінали, шлагбауми, автоматичні каси (інтернет-банкінг, термінали, електронні гаманці), модулі бездротового зв'язку (BLE-маяки для локального позиціонування, Wi-Fi, Bluetooth, 4G/5G

модулі), різноманітні датчики (руху, освітлення, зайнятості місць), контролери керування рухом авто, сервери (для даних, додатків, сервісів).

Специфіка інтелектуальних паркінгів полягає у тому, що пересування автомобілів відбувається без участі водіїв, тому до транспортних засобів висуваються наступні основні вимоги:

- наявність систем автономного безпілотного керування;
- обов'язкове обладнання системами допомоги при паркуванні;
- оснащення різноманітними датчиками, такими як лідари, парктроніки тощо;
- можливість обміну даними між автомобілем та системою паркінгу в обох напрямках.

## 1.2 Проблеми автомобільних парковок

У містах Київ та Харків, де кількість автомобілів досягла рівня найбільших європейських мегаполісів, гостро відчувається нестача паркувальних місць, що створює загрозу транспортного колапсу. Для вирішення цієї проблеми міська влада впроваджує концепцію перехоплюючих паркінгів.

Перехоплюючі паркінги призначені для того, щоб водії залишали свої авто на околиці міста і пересідали на міський громадський транспорт, зменшуючи тим самим навантаження на дорожню мережу в центральній частині міста.

Перехоплюючий паркінг – це автостоянка, розташована на під'їзді до центральної частини міста або в передмістях, що має зручний зв'язок із міським громадським транспортом. Він дозволяє мешканцям приміських зон та іншим особам залишати свої автомобілі на стоянці та пересідати на автобус, маршрутне таксі, трамвай, метро чи інший транспорт для подальшого руху до центру міста. Автомобіль зберігається на перехоплюючому паркінгу протягом дня, а власник забирає його після



повернення. Така схема покликана зменшити транспортне навантаження на центральні вулиці міста.

Протягом багатьох років Харків потерпає від хронічних транспортних заторів та гострої нестачі паркувальних місць. Пошук вільного місця для паркування перетворюється для водіїв на нездійсненне завдання. Коріння проблеми полягає в тому, що міська інфраструктура Харкова спочатку не проектувалася з розрахунку на таку величезну кількість приватних автомобілів, якою вона є сьогодні.

Паркування автомобілів сьогодні є однією з найскладніших задач для автовласника. Часто важко відразу знайти вільне паркувальне місце через велике скупчення машин. Крім того, ситуація на парковці постійно змінюється, і зайняті місця можуть швидко звільнитися. Проте складність полягає не лише у пошуку вільного місця – майже кожен водій, особливо з невеликим досвідом, стикається з проблемою нестачі простору для маневрування під час паркування. Іноді важко відразу визначити, чи достатньо місця для паркування чи ні.

### 1.3 Аналіз існуючих систем керування паркінгом

#### 1.3.1 Система Park System

Park System – це система автоматизації паркування, яка передбачає програмно-апаратний комплекс, призначений для автоматизації в'їзду та виїзду транспортних засобів на територію платного паркінгу (рисунок 1.1). Напівавтоматичне паркування передбачає наявність касира, який здійснює взаєморозрахунки з клієнтами паркувального комплексу. Місце касира може бути облаштовано в будь-якому зручному місці, сервер і паркомати пов'язані по мережі LAN [4].



Рисунок 1.1 – Приклад автоматизованої системи від бренду Park System [4]

Основні функції цієї системи автоматизації паркування:

- розподіл потоків транспортних засобів відвідувачів;
- здійснення контролю доступу автотранспортних засобів на територію паркувального простору;
- автоматизований облік часу проведеного автотранспортним засобом на території паркінгу;
- автоматизація оплати часу проведеного на платному паркінгу, за допомогою спеціального терміналу самообслуговування;
- забезпечення прискореного проїзду для постійних клієнтів які мають спеціальні перепустки (постійні відвідувачі/службові карти);
- збір, зберігання та обробка даних про транспортні засоби які здійснили в'їзд на паркувальний майданчик (включаючи фото та відео) та про оплату паркування;
- автоматизований контроль за роботою обладнання для паркування, та своєчасне інформування обслуговуючого персоналу про стан обладнання;
- звітність про всі події та кошти що стосуються паркувального майданчику.

Ця система автоматизації паркування – це варіант, який має на увазі автоматичний в'їзд та виїзд на територію паркінга, але оплату послуг паркування через касу. Комплект включає в себе шлагбауми для в'їзда та виїзда з території паркінгу, в'їздний та виїздний паркомати, інше обладнання необхідне для оснащення проїздів та обладнання необхідне для роботи касира.

На в'їзді на територію парковки встановлюється шлагбаум та в'їздний паркомат. Водій автомобіля повинен натиснути кнопку на ліцевій панелі паркомата, після чого він отримає в'їздний квиток та шлагбаум відчинеться. Якщо водій має RFID перепустку та має право на безкоштовний проїзд, він також повинен піднести її до зчитувача який знаходиться на панелі паркомата (рисунок 1.2).

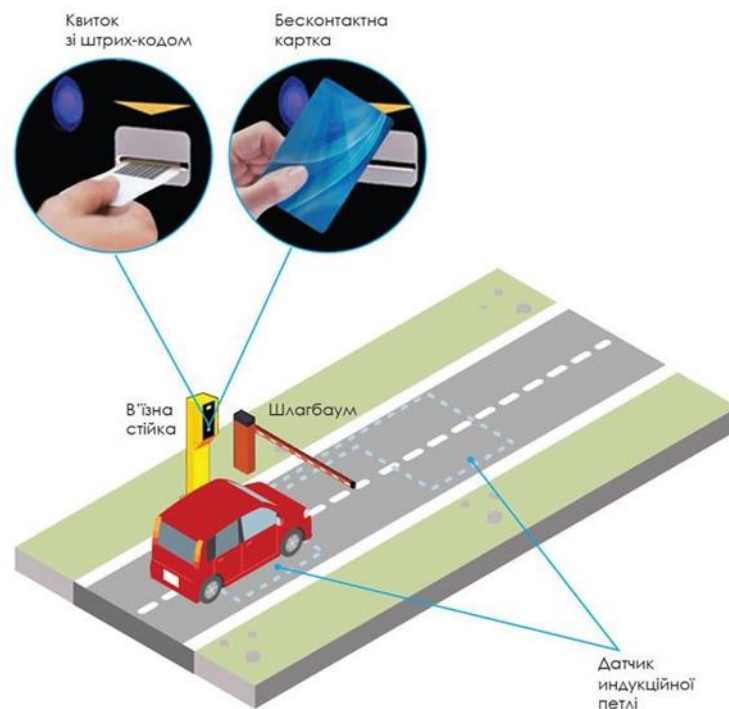


Рисунок 1.2 – Схема в'їзду на парковку [4]

Комплектація системи включає в'їзну та виїзну стійки, шлагбауми ZkTeco BG1045L з телескопічною стрілою 2,5-4,5 метри, фотоелементи СОМРАСТА А20-80, дистанційні пульти керування шлагбаумом, контролери

індукційної петлі, індукційні петлі, програмне забезпечення "SPSOFT", мультіформатний рідер для запису карт та сканер талонів. У комплект постачання входить серверне обладнання. Для роботи системи необхідний комп'ютер для адміністрування та окремий комп'ютер для серверу. Система працює в онлайн-режимі, тому важливо забезпечити безперебійну роботу серверу для підтримання функціональності паркінгу.

Система напівавтоматичного паркування від Park System є ефективним рішенням для автоматизації паркувальних процесів, що включає контроль доступу, облік часу перебування та автоматизацію оплати. Комплексне обладнання та програмне забезпечення забезпечують надійну роботу та високу ефективність управління паркувальним простором.

### 1.3.2 Система керування паркінгом AFAPARK

Визначення та інтуїтивна індикація вільних паркувальних місць у реальному часі, динамічні і інтуїтивні інформаційні табло з відео і аудіо інформацією дозволяють автомобілістам вільно рухатися по парковці. Рух користувачів проходить в комфортній обстановці починаючи з в'їзду на парковку до місця стоянки (рисунок 1.3) [6].



Рисунок 1.3 – Система керування паркінгом AFAPARK [6]

При в'їзді на паркінг динамічний дисплей показує загальну кількість вільних місць для паркування і запрошує водія в'їжджати, інформуючи його про наявність доступних місць.

У середині парковки динамічні стрілкові покажчики полегшують рух автомобілістам, вказуючи, на якому рівні і в якій зоні знаходяться вільні паркувальні місця.

У кожній зоні динамічні покажчики індикують місця, що вільні для паркування.

Після вибору водієм автомобіля алеї, миготлива сигналізація зеленого кольору чітко показує знаходження вільного місця. У момент паркування автомобіля, датчик, який зазвичай розташований зверху на стелі або в центрі паркомісця, визначає наявність автомобіля і відразу змінює колір із зеленого на червоний.

Дисплеї в алеї та відповідній зоні відразу ж відзначають це місце як зайняте, зменшуючи загальну кількість вільних місць.

Як додаткова послуга, в корпус датчика може бути вбудований звуковий гучномовець, який автоматично включається після визначення наявності автомобіля. Він програє персоналізоване звукове повідомлення, яке допомагає власнику автомобіля швидко запам'ятати номер паркомісця. Після цього звукового повідомлення може відтворюватися інша інформація або реклама, персоналізована за допомогою спеціальної програми вашого відділу маркетингу. Цей звуковий сигнал також може бути використаний для спеціалізованих місць для паркування, таких як інвалідні, сімейні, для електромобілів і т.д.

Як додаткова послуга, в сигналізацію може бути вбудований звуковий гучномовець, який автоматично включає приємну музику, створюючи гармонійну обстановку на паркінгу. В той же час гучномовець може поширювати повідомлення про заходи евакуації у разі пожежі.

AFAPARK постійно розробляє і вдосконалює свій асортимент продукції, щоб адаптуватися до різних архітектурних обмежень парковок, що

дозволяє оптимізувати витрати на впровадження системи.

Система працює таким чином: при в'їзді на паркінг клієнт завдяки світловим табло вздовж проїзду швидко дізнається про наявність вільних місць. У секторі яскрава світлова сигналізація вказує на всі доступні паркувальні місця, забезпечуючи швидку і комфортну парковку, причому сигналізатор добре помітний незалежно від напрямку руху. Після зупинки автомобіля система озвучує місцезнаходження пасажирам, допомагаючи знайти машину або попереджаючи про місце, призначене для інвалідів.

Крім комфорту для клієнтів, візуально-звукова система адміністрування та управління рухом на автостоянці AFAPARK сприяє ефективному управлінню та скороченню витрат завдяки графічній та статичній інформації. В аварійних ситуаціях адміністрація може швидко повідомити про евакуацію, а також система AFAPARK може стати додатковим джерелом доходу через рекламні покази. Використання цієї системи сприяє підвищенню рентабельності.

Використання системи організації, керівництва та управління паркінгом AFAPARK приносить економічну вигоду декількома способами. Спочатку, оптимізований рух та скорочення переміщень по паркінгу зменшують час, витрачений на паркування, у середньому на 6 хвилин. Крім того, система забезпечує максимальне використання паркувальної площі – 100%, у порівнянні з традиційними паркінгами, де навіть при оптимальному заповненні не використовується до 2% площі. Це досягається завдяки точній візуалізації та направленню водіїв на вільні місця.

З екологічної точки зору, система також має значну цінність. По-перше, вона допомагає знизити витрати палива та забруднення повітря, оскільки автомобілі направляються безпосередньо на вільні місця, а час на пошук парковки скорочується. Це може призвести до зниження рівня забруднення до 20%, що є досить значною цифрою.

Економічна вигода від використання системи організації та управління паркінгом AFAPARK відчутна навіть на етапі проектування.

Система AFAPARK це:

- гарантована 100% зайнятість стоянкових місць, що максимізує використання доступної площі;
- зниження інвестиційних витрат при будівництві автостоянки на до 10%, завдяки оптимізації простору та раціональному розміщенню місць;
- економія часу паркування на 75%, завдяки візуальній та звуковій наводці на парковочне місце;
- полегшений рух по території автостоянки завдяки оптимізованій організації руху транспорту;
- зменшений ризик втрати автомобіля та підвищена безпека завдяки контролю та моніторингу на паркінгу;
- ідентифікація місць стоянок, призначених для інвалідів та іншого призначення, що сприяє соціальній відповідальності та комфорту користувачів;
- зниження витрат експлуатації паркінгу, зокрема електроенергії, що використовується для вентиляції і освітлення, що веде до зменшення експлуатаційних витрат;
- зниження емісії вихлопних газів (CO<sub>2</sub>) на майже 4 рази, що призводить до значного зменшення забруднення довкілля та покращення якості повітря для працівників і користувачів паркінгу.

### 1.3.3 Автоматизована парковка FidPark

FidPark-CAR призначена для автоматизації платних і службових автостоянок. Система надає водіям різні способи оплати за стоянку – передоплата, депозити, квитки заходів, різні абонементи [6]. Програмне забезпечення FidPark дозволяє налаштувати параметри автостоянки для досягнення заданої бізнес-цілі:

- отримання максимального прибутку автостоянки;
- максимальна пропускна здатність об'єкта, який вона обслуговує;

- постійна наявність вільних місць.

#### Характеристики та можливості FidPark:

- необмежена кількість контрольних пунктів (в'їздів і виїздів);
- необмежена кількість паркувальних зон з різними тарифами та контролем доступу;
- можливість розширення функціональності системи та її інтеграції з системами обліку;
- необмежена кількість користувачів;
- необмежена тривалість паркування автомобілів;
- необмежена кількість постійних клієнтів;
- необмежена кількість тарифних планів;
- різні тарифи на кожен день тижня;
- установка цін для кожного інтервалу тривалістю 5 хв. – 24 години;
- перелік святкових днів і робочих днів-винятків;
- збереження локальної функціональності при розриві зв'язку з сервером;
- захист від вандалізму і злому програмного забезпечення;
- відповідність фіскального законодавства;
- зручна інтеграція з зовнішніми WEB платіжними системами;
- система знижок, яка працює через WEB;
- можливість інтеграції з системою контролю доступу.

У версії FidPark-ANPR з автоматичним розпізнаванням номерів (Automatic Number Plate Recognition) автомобільний номер зчитується спеціальними відеокамерами і може бути використаний як додатковий засіб ідентифікації або як альтернатива іншим методам, таким як RFID, штрих-код і т.п. ANPR значно розширює можливості систем FidPark і FidPark +, а також підвищує зручність використання системи водіями, забезпечуючи можливість автоматичної оплати кожної стоянки з банківської картки.

Функцію ANPR можна використовувати як додаткову опцію. Приклад роботи цієї функції зображений на рисунку 1.4.





Рисунок 1.4 – Приклад роботи функції ANPR

Якщо ANPR використовується як додаткова опція, FidPark набуває таких можливостей:

- якщо клієнт з одноразовим штрих-кодовим талоном заплатив за паркування, то система FidPark може випустити цю машину без необхідності сканування талона на виїзді, за винятком рідкісних випадків помилкового розпізнавання номера;
- адміністратор може легко визначити дату/час заїзду машини з конкретним номером і відповідну вартість парковки, що вирішує проблеми розрахунків з клієнтами, що заявили про втрату паркувального талона довготривалої стоянки;
- порівняння номерів на в'їзді і виїзді запобігає виїзду автомобіля за чужу перепустку;
- на багатозонних парковках, камери ANPR між зонами дають можливість контролювати фактичне перебування автомобіля в кожній зоні без додаткових терміналів і бар'єрів;
- платежі зареєстрованих клієнтів, прив'язаних до номера свого автомобіля, автоматично списуються з їх картки або мобільного рахунку, що звільняє їх від необхідності платити касиру або в касовому автоматі;
- клієнт може забронювати місце на стоянці, ввівши на смартфоні реєстраційний номер свого автомобіля замість отримання і використання талона зі штрих-кодом;

– з'являється можливість побудувати службову автостоянку без в'їзних і виїзних терміналів, що має велике значення в містах з жорсткими правилами будівництва та землекористування.

FidPark з ANPR як єдиною технологією ідентифікації є найбільш зручним варіантом для службових автостоянок. За ростом популярності сприяє зниження цін на відеокамери.

Функцію ANPR використовують для постійних клієнтів. Приклад роботи функції представлений на рисунку 1.5.

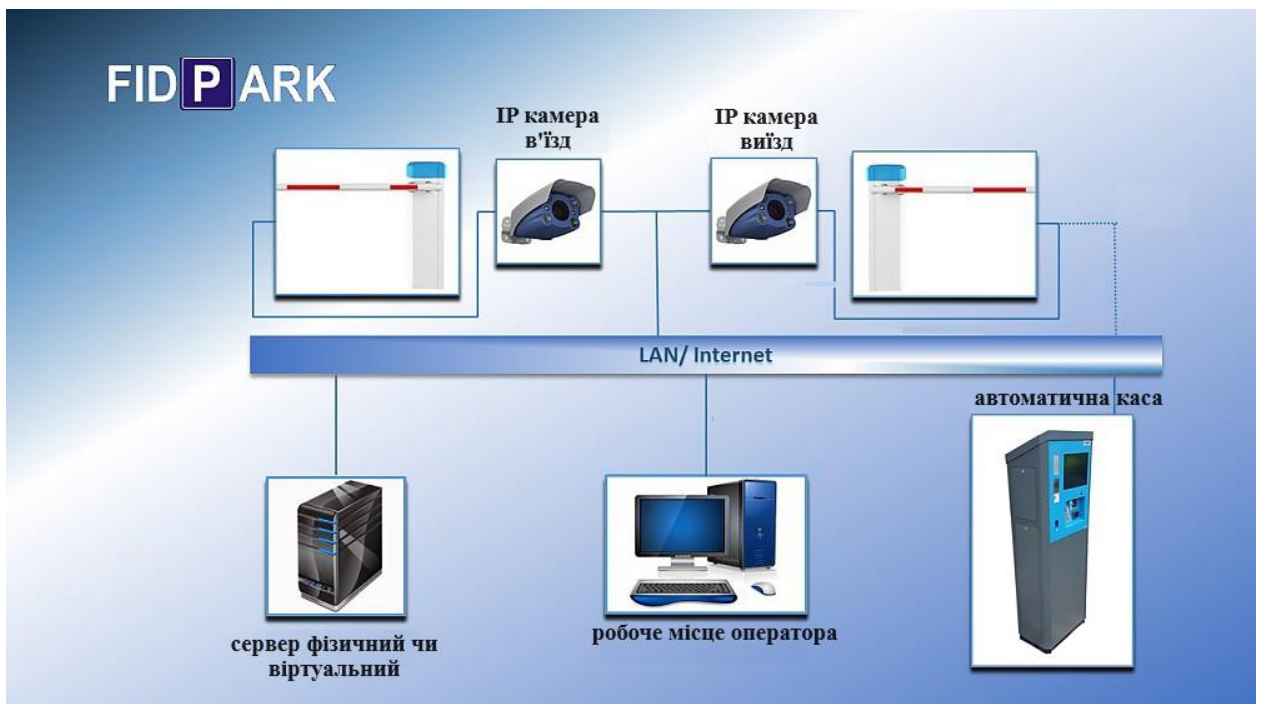


Рисунок 1.5 – Приклад роботи функції ANPR для постійних клієнтів [7]

Нижче наведено перелік можливостей FidPark-ANPR:

- всі клієнти звільняються від необхідності будь-яких дій з використанням талонів, карток і таке подібне;
- усунення паперових талонів зі штрих-кодами для разових клієнтів дозволяє знизити витрати на підтримку та технічне обслуговування системи.
- усувається можливість зловживань з багаторазовим використанням одного пропуску (antipassback);
- зареєстровані клієнти з відомими реєстраційними номерами автомашини можуть використовувати стоянку, не витрачаючи час на оплату,

яка при виїзді буде зніматися автоматично з рахунку клієнта;

- клієнт може забронювати місце для паркування через Інтернет, просто ввівши реєстраційний номер свого автомобіля;

- у разі паркування без шлагбаумів (on-street) забезпечується автоматична реєстрація часу в'їзду / виїзду і генерації відповідних звітів / рахунків для справляння оплати з клієнтів.

Приклад використання функції ANPR разовими клієнтами наведений на рисунку 1.6.

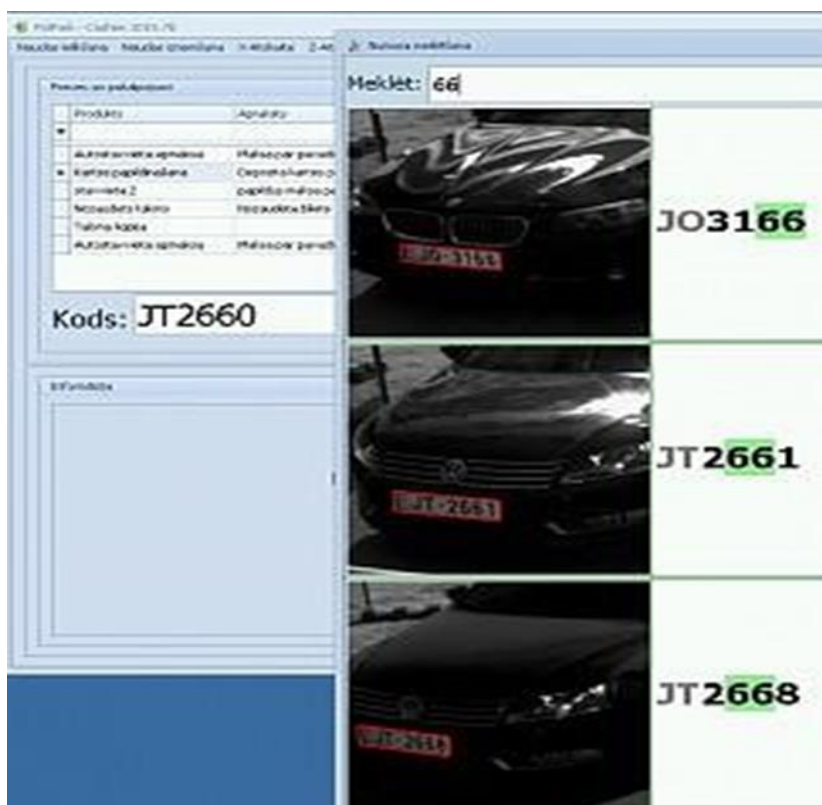


Рисунок 1.6 – Приклад роботи функції ANPR для разових користувачів

Особлива увага в системі FidPark-ANPR для платних стоянок була приділена зробленню всіх операцій більш доброзичливими для клієнтів, навіть у випадку помилкового розпізнавання одного або декількох символів номерного знаку. Це зроблено з метою уникнення стресу для клієнтів, особливо для тих, хто не має досвіду роботи з сенсорними екранами. У випадку автоматичної каси клієнту достатньо ввести лише кілька цифр номерного знаку, щоб вибрати свою автомашину зі списку для оплати

стоянки. Аналогічна процедура доступна і в програмі для касира.

FidPark підтримує різноманітні технології ідентифікації клієнтів, серед яких:

- безконтактні мітки: вони можуть бути у формі карток, браслетів тощо і працюють на різних частотах (125 кГц, 13,56 МГц, ISO/IEC 14443a/b або ISO/IEC 15693). Деякі з них мають зчитування на відстань до 10 см, а інші – до 1 метрів;
- мітки з великою відстанню зчитування: ці мітки працюють на частоті UHF і можуть бути зчитані на відстані до 10 метрів;
- NFC-картки або мобільні телефони: вони можуть бути зчитані на відстань до 2-3 см і забезпечують можливість швидкої ідентифікації через смартфон;
- 2D штрих-коди: вони можуть бути розміщені на разових талонах або екрані смартфона і зчитані з відстані;
- державні номери автомобіля: використовуються для ідентифікації транспортних засобів;
- номер або IMEI мобільного телефону: вони можуть бути використані для ідентифікації користувачів за їхніми мобільними пристроями;
- біологічні методи ідентифікації: наприклад, відбитки пальців, які можуть бути використані для безпечної ідентифікації особи.

Вибір конкретних пристроїв і технологій залежить від особливостей об'єктів та вимог замовників. Можливо використовувати декілька ідентифікаторів для одного клієнта, наприклад, картку і браслет.

У системі FidPark-Phone для ідентифікації водія використовується мобільний додаток. При в'їзді водій повинен здійснити дзвінок на певний номер, зазвичай вказаний на стрілі шлагбаума. Цей дзвінок є безкоштовним. Якщо власник номера телефону має право на в'їзд, то шлагбаум автоматично відкривається. У протилежному випадку, протягом кількох секунд водій отримує SMS з вказівкою причини відмови.

FidPark-Phone найбільш зручно використовувати на службових автостоянках, де дозволяється в'їзд лише постійним клієнтам або відвідувачам за попереднім погодженням. Це дозволяє зберігати функціональність системи FidPark, зокрема – групові пропуски, тайм-зони та захист від багаторазового в'їзду.

Для управління системою можна використовувати будь-який веб-браузер. Орендарі автостоянки мають можливість в межах виділеної квоти (кількість місць, тайм-зони) самостійно редагувати список номерів телефонів водіїв, яким надано право на в'їзд на автостоянку. Також вони можуть реєструвати "своїх" відвідувачів, вводячи їхні мобільні номери та час початку візиту (в'їзду на стоянку). Система автоматично розраховує вартість стоянки відвідувача з урахуванням тривалості, а отримані кошти зараховуються на рахунок користувача автостоянки.

FidPark-Phone може бути ефективним варіантом для використання на платних автостоянках як альтернатива системі FidPark-LPR, яка базується на розпізнаванні номерів.

#### 1.4 Висновки до розділу

Система Park System оптимізує використання простору, забезпечуючи максимально ефективне розміщення транспортних засобів та зменшуючи кількість незайнятих місць. Завдяки цим функціям система керування паркінгом від Park System підвищує загальну ефективність паркувального процесу та забезпечує зручність для користувачів.

Основні функції системи керування паркінгом AFAPARK: виявлення вільних місць за допомогою датчиків у кожній зоні та на паркомісцях; індикація зайнятості місць за допомогою світлових сигналів; динамічні дисплеї та стрілкові покажчики для навігації; персоналізовані звукові повідомлення про місце паркування; оптимізація використання паркувальної площі; звукові повідомлення та музика для створення гармонійної

атмосфери.

Основні функції автоматизованої парковки FidPark: автоматичне розпізнавання номерних знаків; використання різних методів ідентифікації (RFID, штрих-коди); різні способи оплати: передоплата, депозити, абонементи; налаштування гнучких тарифів для різних зон та часу; захист від вандалізму та злому програмного забезпечення; автоматичні повідомлення для водіїв про статус їх паркування.

Було досліджено завдання, для вирішення яких використовується автоматизація паркування, зокрема:

- ускладнення пошуку вільних місць на парковці;
- уявна завантаженість паркінгу при наявності вільних місць;
- ДТП та транспортні затори, викликані хаосом у зайнятості паркувальних місць;
- мінімізація часу пошуку паркувальних місць.

Крім того, розглядалася можливість візуалізації вільних місць у реальному часі.

Динамічні та інтуїтивні інформаційні табло, які дозволяють легко та безперешкодно орієнтуватися на парковці.

Отже, швидкодія системи залежить від різноманітних факторів:

- максимальна пропускна здатність об'єктів, які система обслуговує;
- постійна наявність вільних місць для паркування;
- велика кількість паркомісць;
- ширина проїзної частини;
- різноманітність типів паркомісць;
- збереження локальної функціональності при відключенні від мережі;
- зручна інтеграція з зовнішніми платіжними системами.

Це свідчить про широкі можливості розширення функціональності системи та її інтеграції з системами обліку.

Все це демонструє, наскільки різноманітними можуть бути паркінги.

## 2 РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ ТА АЛГОРИТМУ РОБОТИ СИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ ОПТИМІЗАЦІЇ ПАРКУВАННЯ АВТОТРАНСПОРТУ

### 2.1. Загальний опис системи

Функціонування системи автоматизації паркування на базі IoT спрямоване на забезпечення постійної доступності інформації про наявність вільних паркувальних місць, полегшуючи пошук для користувачів і підвищуючи ефективність управління паркувальним простором. Система відстежує, коли паркувальне місце стає вільним або зайнятим, забезпечуючи точний і актуальний облік місць у реальному часі.

Основним компонентом цієї системи є ультразвукові датчики, які встановлюються на кожному паркувальному місці. Вони вимірюють відстань до найближчого об'єкта, що дозволяє визначити, чи зайняте паркувальне місце. Ці дані передаються до контролерів Arduino, які обробляють сигнали від датчиків і визначають статус паркувальних місць. Контролери передають зібрані дані до IoT-шлюзів, які забезпечують передачу даних на сервер через Wi-Fi-з'єднання.

Однією з ключових особливостей цієї системи є можливість автоматичної передачі даних про статус паркувальних місць на віддалений сервер, що дозволяє користувачам отримувати актуальну інформацію в будь-який момент. Сервер обробляє отримані дані і зберігає їх у базі даних, забезпечуючи можливість доступу до них через API для різних клієнтських додатків.

Для зручності користувачів створено веб-сайт, де вони можуть переглядати актуальний статус паркувальних місць, бронювати вільні місця та отримувати сповіщення про зміни. Веб-сайт розроблений з використанням сучасних технологій, таких як HTML, CSS і JavaScript, що забезпечує

зручний і інтуїтивно зрозумілий інтерфейс.

Система автоматизації паркування на базі IoT має значні переваги. Вона покращує користувацький досвід, знижуючи час на пошук вільного паркувального місця і надаючи можливість бронювання. Крім того, система забезпечує ефективне управління паркувальним простором, автоматично оновлюючи статус паркувальних місць і дозволяючи управляти завантаженістю. Також важливим аспектом є масштабованість системи, яка дозволяє легко інтегрувати додаткові датчики і розширювати покриття.

Впровадження такої системи автоматизації паркування значно підвищує ефективність використання паркувальних ресурсів, забезпечуючи зручний доступ до інформації про статус паркувальних місць у реальному часі. Це дозволяє зменшити навантаження на паркувальні майданчики, підвищити зручність для користувачів і забезпечити ефективне управління паркувальним простором. Завдяки використанню сучасних технологій, система є надійною і гнучкою платформою для впровадження у різних умовах і може бути легко адаптована до різних потреб.

## 2.2 Структурна схема системи автоматизації та алгоритм її роботи.

Система автоматизації паркування включає можливості моніторингу зайнятості паркувальних місць, передачі даних у режимі реального часу на сервер, а також відображення інформації для користувачів.

Як тільки в радіусі дії ультразвукового датчика з'являється автомобіль, він відразу визначає статус паркувального місця та надсилає дані до сервера для подальшої обробки і відображення на сайті, де цю інформацію може побачити користувач після авторизації.

Структурну схему наведено на рисунку 2.1.



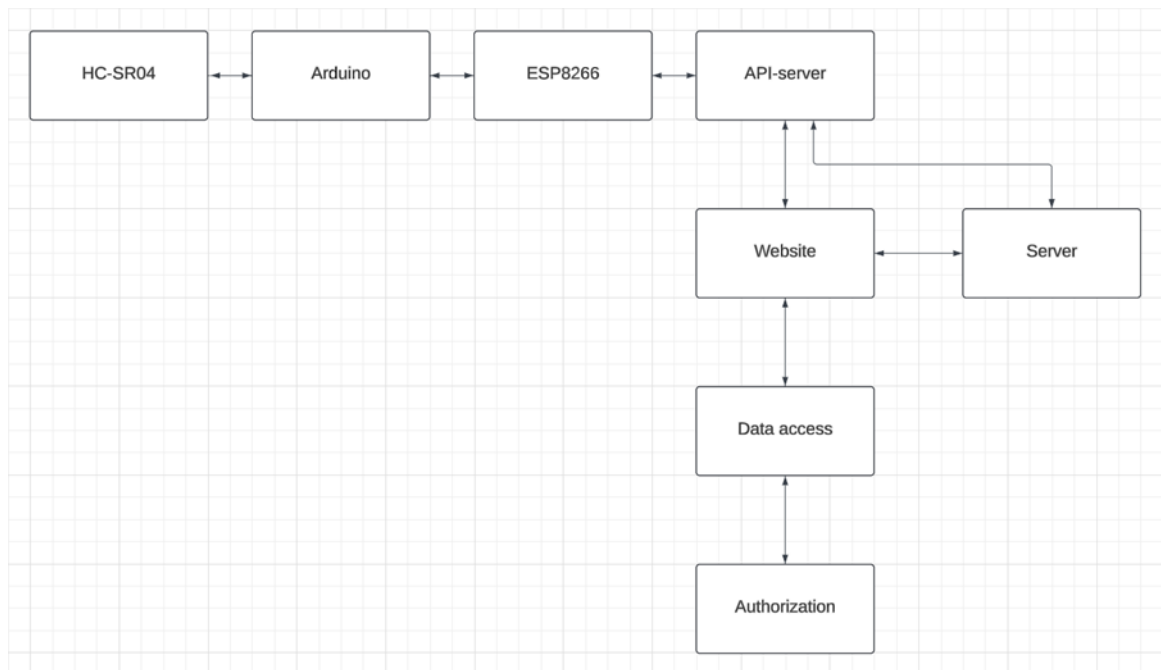


Рисунок 2.1 – Структурна схема системи

Структурна схема системи автоматизації паркування складається з наступних компонентів:

- ультразвуковий датчик HC-SR04: вимірює відстань до об'єкта (автомобіля) і передає дані на контролер;
- контролер Arduino: обробляє дані з датчика, визначає статус паркувального місця (вільне або зайняте) і передає цю інформацію на IoT-шлюз;
- IoT-шлюз ESP8266: приймає дані від контролера Arduino і передає їх на веб-сервер через Wi-Fi;
- веб-сервер: обробляє отримані дані, зберігає їх у базі даних і надає доступ до них через веб-сайт;
- веб-сайт: відображає інформацію про статус паркувальних місць для користувачів.

Опис алгоритму системи графічним способом у вигляді блок-схеми зображено на рисунку 2.2.

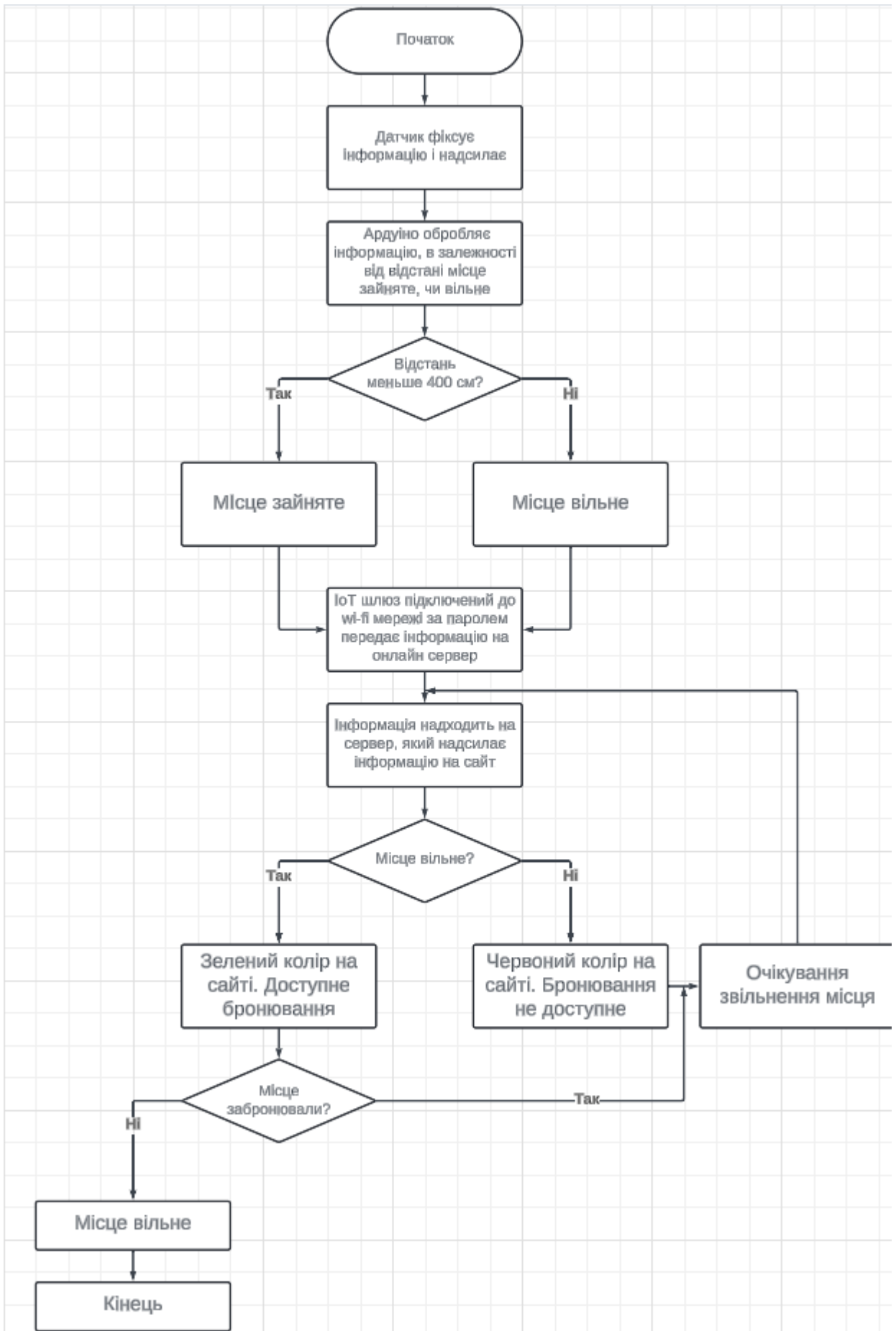


Рисунок 2.2 – Алгоритм системи автоматизації у графічному виді

Алгоритм роботи системи моніторингу та бронювання паркувальних місць:

- початок;
- ультразвуковий датчик фіксує наявність або відсутність автомобіля на паркувальному місці;
- контролер Arduino обробляє дані від датчика та визначає, чи зайняте місце;
- якщо відстань менше 400 см, місце зайняте; інакше місце вільне;
- дані передаються через Wi-Fi модуль ESP8266 на сервер;
- сервер оновлює інформацію про стан місць на веб-сайті;
- на сайті відображається зелений колір, якщо місце вільне, або червоний, якщо зайняте;
- якщо місце вільне, користувач може забронювати його через сайт, або зайняти його;
- якщо місце заброньоване, воно відображається як зайняте і очікує час бронювання, після якого датчик оновлює інформацію про стан місця;
- кінець.

## 2.3 Вибір компонентів системи автоматизації для оптимізації паркування автотранспорту

### 2.3.1 Вибір ультразвукового датчика для системи, його характеристики

Ультразвуковий датчик відстані HC-SR04 [9] – це простий далекомір з великим діапазоном вимірюваних відстаней (рисунок 2.3). Ультразвуковий далекомір генерує звукові імпульси на частоті 40 кГц і слухає відлуння. За часом поширення звукової хвилі туди і назад можна однозначно визначити відстань до об'єкта. Цей далекомір може слугувати датчиком для робота, завдяки якому він зможе визначати відстані до об'єктів або об'їжджати перешкоди. Для того, щоб вмонтувати ультразвуковий датчик HC-SR04 в

експериментальну версію системи, важливо знати його реальні фізичні розміри, це допоможе при проектуванні кріплень та корпусу для датчика. Реальні розміри та діаграма спрямованості датчика відображено на рисунку 2.4.



Рисунок 2.3 – Ультразвуковий датчик відстані HC-SR04 [10]

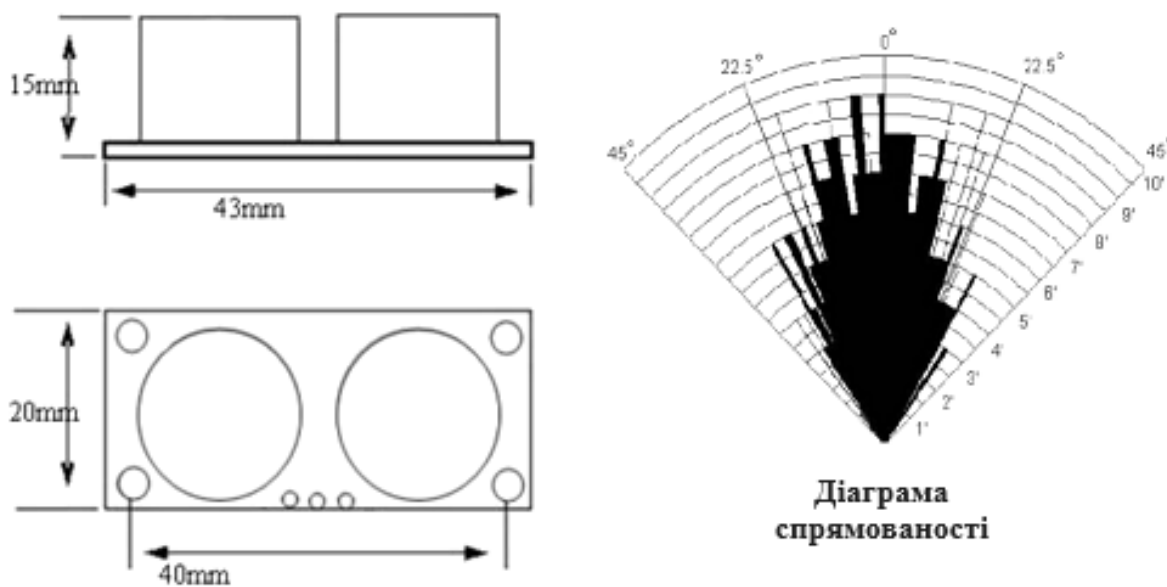


Рисунок 2.4 – Реальні розміри та діаграма спрямованості датчика [10]

Принцип роботи:

- відправлення сигналу: датчик надсилає ультразвуковий імпульс

тривалістю 10 мікросекунд через пін TRIG;

- відображення сигналу: імпульс відбивається від об'єкта і повертається до датчика;

- отримання сигналу: датчик отримує відбитий імпульс через пін ЕСНО;

- вимірювання часу: контролер вимірює час між відправленням і отриманням імпульсу;

- обчислення відстані: відстань обчислюється за формулою: відстань = (час x швидкість звуку) / 2.

В таблиці 2.1 наведено технічні характеристики ультразвукового датчика відстані HC-SR04.

Таблиця 2.1 – Технічні характеристики датчика HC-SR04

Діапазон вимірювання	від 2 см до 400 см
Точність	до 3 мм
Кут виявлення	15°
Кут вимірювань	30°
Робоча напруга	5 В
Сила струму спокою	2 мА
Ширина імпульсу тригера	10 мікросекунд
Розміри	43 мм x 20 мм x 15 мм

### 2.3.2 Вибір IoT-шлюза для забезпечення бездротового зв'язку

ESP8266 – це невеликий мікроконтролерний модуль, який розробляється компанією Espressif Systems (рисунок 2.5). Його основними характеристиками є вбудований Wi-Fi і можливість програмування за допомогою мови програмування C++ або Arduino IDE. ESP8266 широко використовується для розробки Інтернету речей (IoT) та проектів з підключенням до Інтернету через Wi-Fi. Він дуже популярний серед хобістів

і професійних розробників завдяки своїй низькій вартості, потужним можливостям і простоті використання. Існує безліч модифікацій плат, іменованих зазвичай від ESP-01 до ESP-12. Відмінності в платах полягають здебільшого в портах введення-виведення, кількості флеш-пам'яті, виду конекторів тощо. Процесор – один і той самий, тож з погляду програмування не має значення, яку плату програмувати [11].



Рисунок 2.5 – Мікроконтролерний модуль ESP8266 [11]

В таблиці 2.2 наведено технічні характеристики мікроконтролерного модуля ESP8266.

Таблиця 2.1 – Характеристики мікроконтролерного модуля ESP8266

Робоча напруга	3,3 В
Енергоспоживання	10 мкА...170 мА
Флеш-пам'ять	до 16 МБ максимум (зазвичай 512 кБ)
Процесор	Tensilica L106, 32 біта
Швидкість процесора	80...160 МГц
ОЗП	32 кБ + 80 кБ
Максимальне число підключень TCP	5
Тип антени	вбудована
Підтримка безпеки	WPA / WPA2

ESP8266 володіє оптимізованим споживанням енергії, що дозволяє йому працювати на батарейках або інших джерелах живлення довгий час. Також, він має можливість працювати в режимі глибокого сну, що дозволяє мінімізувати споживання енергії в періоди, коли він не активний.

ESP8266 виступає як мікроконтролер, який отримує дані від ультразвукового датчика HC-SR04 та передає їх на веб-сервер. Послідовність дій, які будуть виконуватись:

- прийом даних від Arduino: ESP8266 підключений до контролера Arduino за допомогою інтерфейсу UART. Він отримує від Arduino інформацію про відстань до перешкоди, яку виміряв ультразвуковий датчик;
- підключення до мережі Wi-Fi: ESP8266 підключається до доступної Wi-Fi мережі для передачі даних на веб-сервер;
- відправлення даних на сервер: після отримання даних від Arduino, ESP8266 упакує ці дані в форматі HTTP або MQTT та відправляє їх на сервер через Wi-Fi зв'язок;
- обробка даних на сервері: сервер отримує дані від ESP8266 і виконує їх обробку. Це може включати збереження даних в базі даних, аналіз та подальшу обробку для відображення на веб-сторінці;
- взаємодія з веб-інтерфейсом: ESP8266 може також взаємодіяти з веб-інтерфейсом, який дозволяє користувачам відслідковувати статус паркувальних місць, робити бронювання та отримувати сповіщення про доступність місць.

#### 2.4 Розрахунок передавальних функцій компонентів системи

Ультразвуковий датчик HC-SR04 вимірює відстань до об'єкта, випромінюючи ультразвуковий сигнал і приймаючи його відбиття від об'єкта. Час проходження сигналу від датчика до об'єкта і назад використовується для обчислення відстані. Відстань розраховується за формулою:

$$d = \frac{v \cdot t}{2}, \quad (1)$$

де  $d$  – відстань до об'єкта;

$v$  – швидкість звуку (343 м/с);

$t$  – час проходження сигналу.

У реальних умовах потрібно враховувати нелінійні ефекти, такі як затухання сигналу чи шум. Змоделюємо вихідний сигнал з додаванням нелінійної складової:

$$d = k \cdot t + \alpha \cdot t^2, \quad (2)$$

де  $k = \frac{v}{2} \cdot 10^6$  (константа пропорційності, з урахуванням переведу в мікросекунди),

$\alpha$  – коефіцієнт нелінійності.

Нелінійна характеристика ультразвукового датчика HC-SR04 зображена на рисунку 2.6.

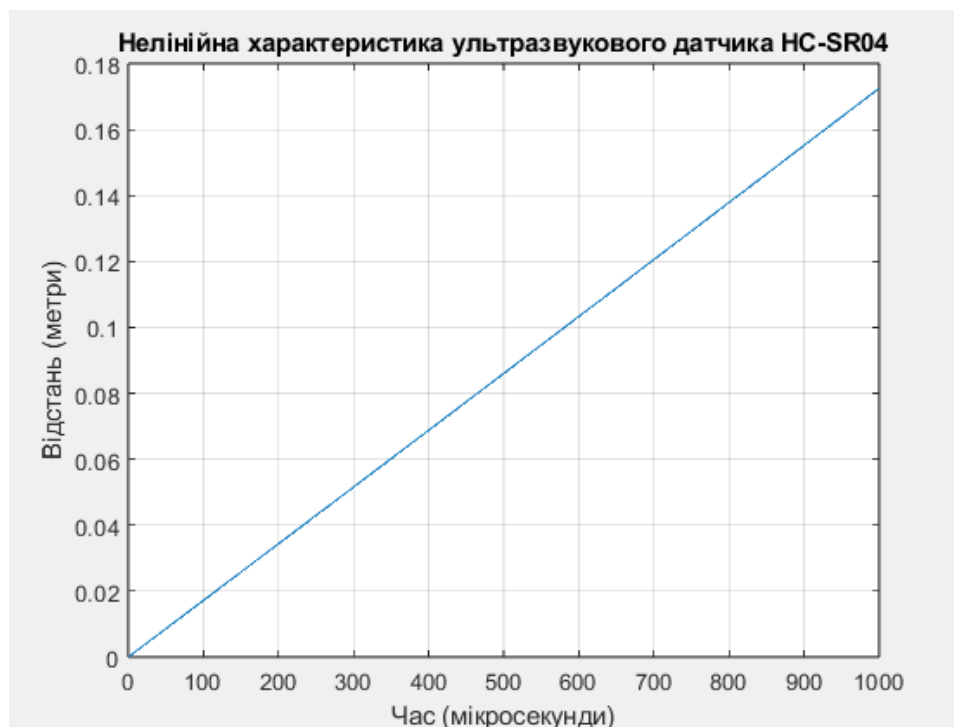


Рисунок 2.6 – Нелінійна характеристика ультразвукового датчика HC-SR04



Передавальна функція ESP8266.

Мікроконтролер ESP8266 використовується для обробки сигналу від датчика HC-SR04 і передачі оброблених даних. Припустимо, що система містить низькочастотний фільтр першого порядку (RC-фільтр), що додає інерційність до обробки сигналу. Передавальна функція такого фільтра має вигляд:

$$H_{ESP8266}(s) = \frac{1}{\tau s + 1}, \quad (3)$$

де  $\tau = RC$  – постійна часу фільтра;

$R$  – опір;

$C$  – ємність.

Якщо  $R = 1$  кОм і  $C = 1$  мкФ, тоді:

$$\tau = (1 \cdot 10^3) \cdot (1 \cdot 10^{-6}) = 10^{-3} \text{ с}. \quad (4)$$

Таким чином, передавальна функція ESP8266 з RC-фільтром буде:

$$H_{ESP8266}(S) = \frac{1}{10^3 s + 1}. \quad (5)$$

Воде діаграми для ESP8266 з RC-фільтром зображено на рисунку 2.7.

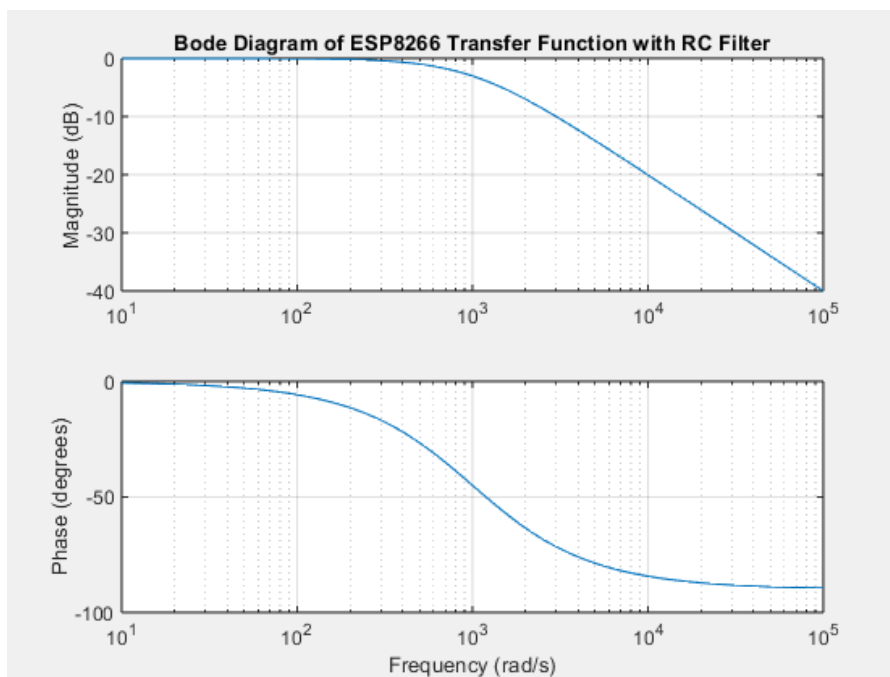


Рисунок 2.7 – Bode діаграми для ESP8266 з RC-фільтром

Імпульсна характеристика для ESP8266 з RC-фільтром зображена на рисунку 2.8.

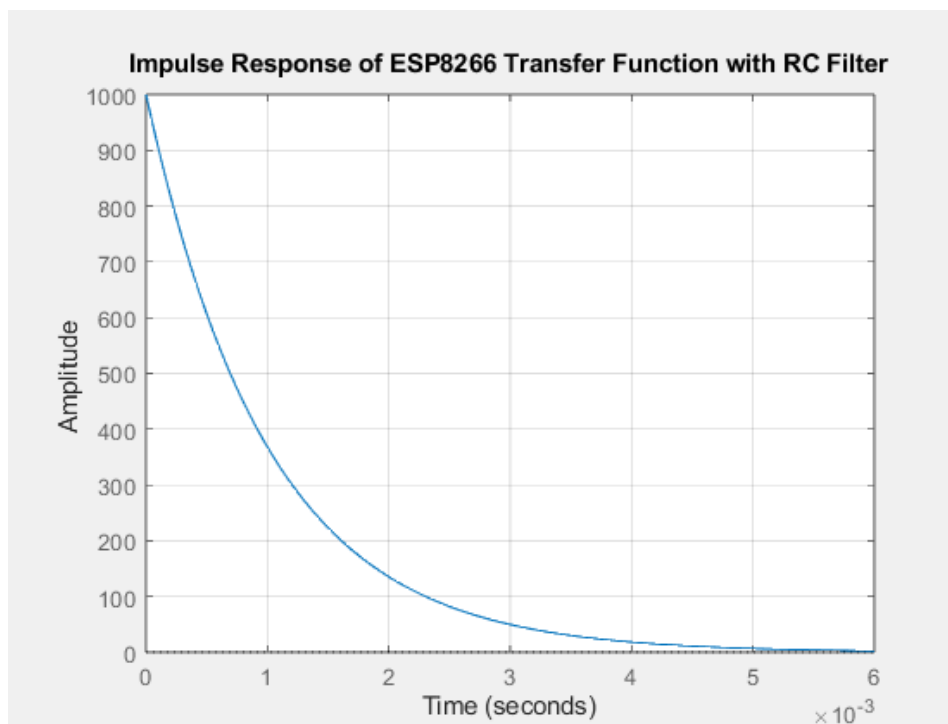


Рисунок 2.8 – Імпульсна характеристика для ESP8266 з RC-фільтром

Перехідна характеристика для ESP8266 з RC-фільтром зображена на рисунку 2.9.

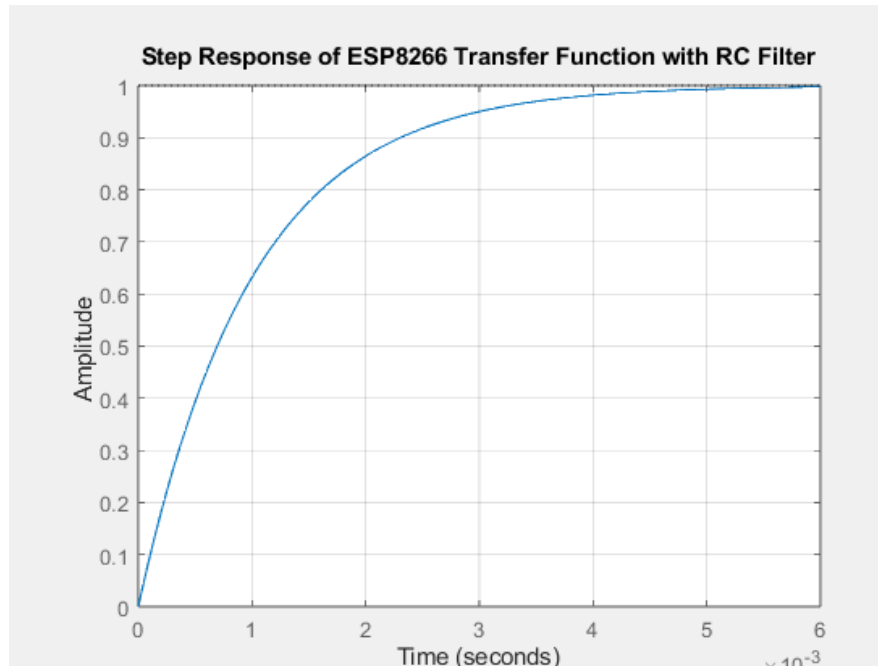


Рисунок 2.9 – Перехідна характеристика для ESP8266 з RC-фільтром

Загальна передавальна функція системи.

Загальна передавальна функція системи, яка складається з ультразвукового датчика HC-SR04 і мікроконтролера ESP8266 з RC-фільтром, може бути представлена як послідовне з'єднання передавальних функцій датчика і фільтра:

$$H_{system}(s) = H_{HC-SR04}(s) \cdot H_{RSP8266}(s), \quad (6)$$

де  $H_{HC-SR04}(s) = 0,0001715$ ;

$$H_{ESP8266}(s) = \frac{1}{10^{-3}s+1}.$$

Воде діаграми загальної передавальної функції систем зображено на рисунку 2.10.

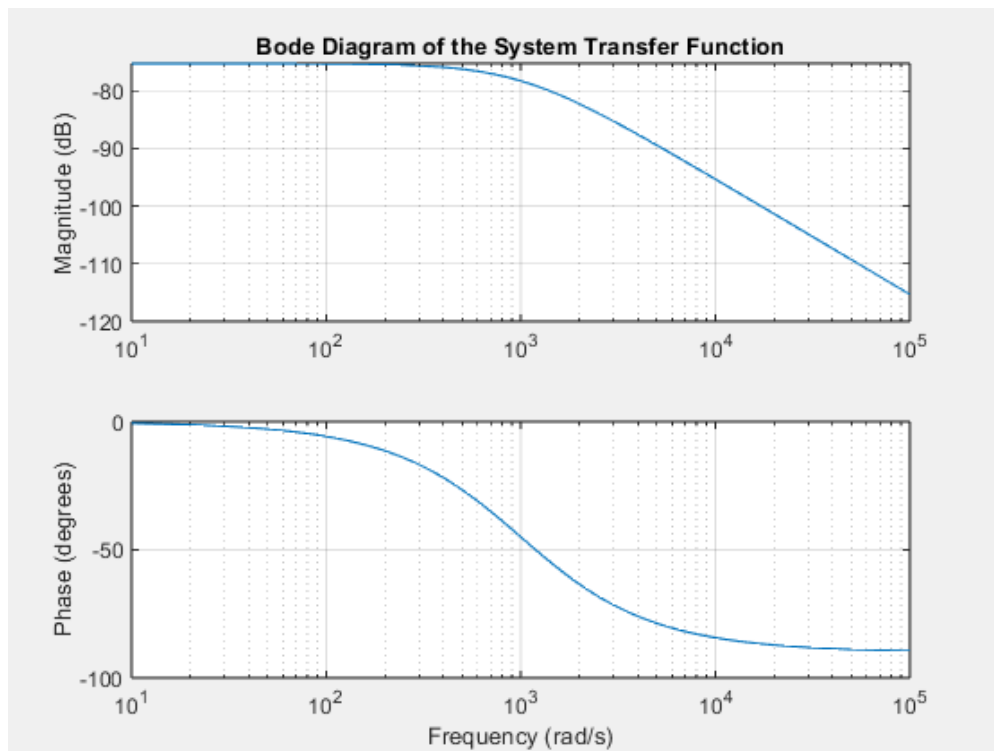


Рисунок 2.10 – Bode діаграми загальної передавальної функції системи

Імпульсна характеристика загальної передавальної функції системи зображена на рисунку 2.11.

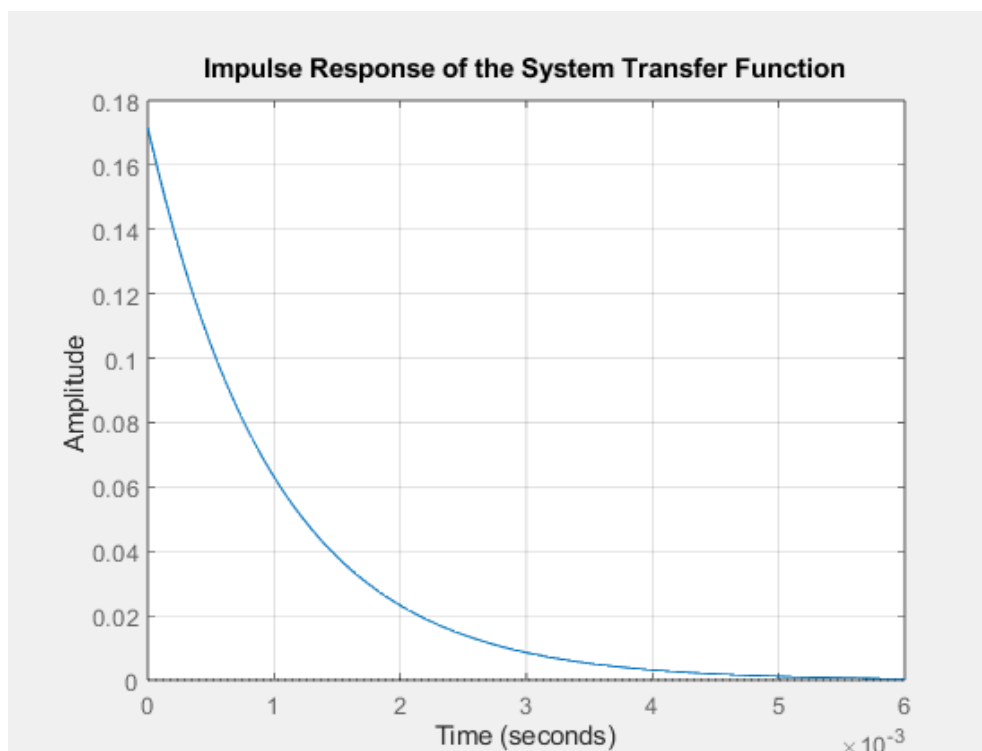


Рисунок 2.11 – Імпульсна характеристика загальної передавальної функції системи

Перехідна характеристика загальної передавальної функції системи зображена на рисунку 2.12.

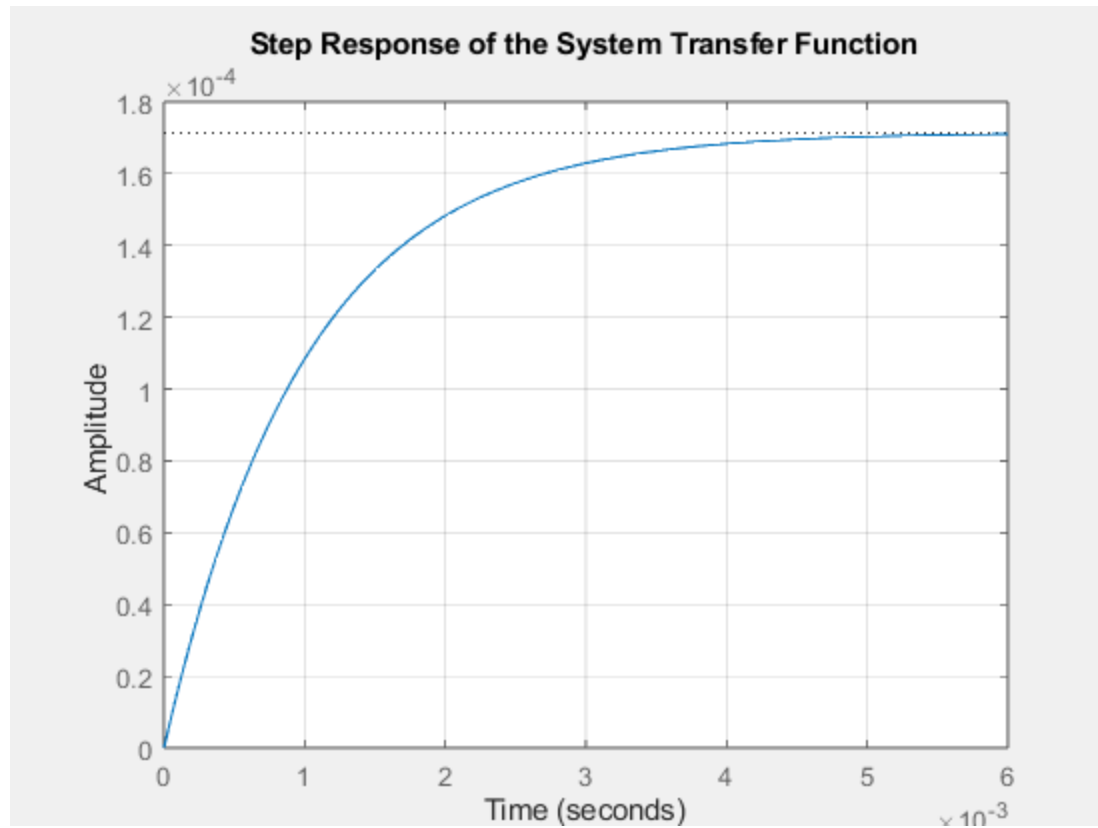


Рисунок 2.12 – Перехідна характеристика загальної передавальної функції системи

Були розглянуті передавальні функції ультразвукового датчика HC-SR04 та мікроконтролера ESP8266 з RC-фільтром. Були представлені теоретичні основи розрахунків, зображення характеристик та відповідні графіки. Отримані результати дозволяють оцінити поведінку системи в цілому, включаючи її частотні та часові характеристики.

## 2.5 Висновки до розділу

У цьому розділі було розроблено структурну схему та алгоритм функціонування системи автоматизації для оптимізації паркування

автотранспорту на базі технології IoT. Система складається з ультразвукових датчиків HC-SR04, контролерів Arduino, IoT-шлюзів ESP8266, веб-сервера та веб-сайту.

Ультразвукові датчики HC-SR04 встановлюються на кожному паркувальному місці для визначення наявності автомобіля. Дані від датчиків обробляються контролерами Arduino, які передають інформацію про стан паркувальних місць на IoT-шлюзи ESP8266 через UART інтерфейс. IoT-шлюзи ESP8266 забезпечують бездротову передачу даних на веб-сервер через Wi-Fi з'єднання. Веб-сервер зберігає отримані дані у базі даних та надає доступ до них через веб-сайт, де користувачі можуть переглядати актуальний стан паркувальних місць і здійснювати бронювання.

Було детально описано алгоритм роботи системи у вигляді блок-схеми, що відображає послідовність операцій від збору даних датчиками до відображення інформації на веб-сайті та можливості бронювання.

Проведено вибір і обґрунтування компонентів системи, зокрема, ультразвукового датчика HC-SR04 та IoT-шлюза ESP8266, з наведенням їх технічних характеристик.

Виконано розрахунок передавальних функцій ультразвукового датчика HC-SR04 та мікроконтролера ESP8266 з RC-фільтром. Побудовані відповідні графіки Bode діаграм, імпульсних та перехідних характеристик для аналізу частотних і часових властивостей системи.

Розроблена система автоматизації паркування на базі IoT забезпечує ефективне управління паркувальним простором, покращує користувацький досвід за рахунок швидкого пошуку вільних місць та можливості бронювання, а також надає гнучкі можливості для масштабування та інтеграції додаткових датчиків.

## 3 РОЗРОБЛЕННЯ ВЕБ-САЙТУ СИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ ОПТИМІЗАЦІЇ ПАРКУВАННЯ АВТОТРАНСПОРТУ

### 3.1 Вибір середовища програмування та програмного забезпечення

Visual Studio Code (VS Code) [12] – це безкоштовний і відкритий вихідний код редактор коду, розроблений і підтримуваний корпорацією Microsoft. Він є одним із найпопулярніших інструментів для розробки програмного забезпечення в середовищі програмування (рисунок 3.1).

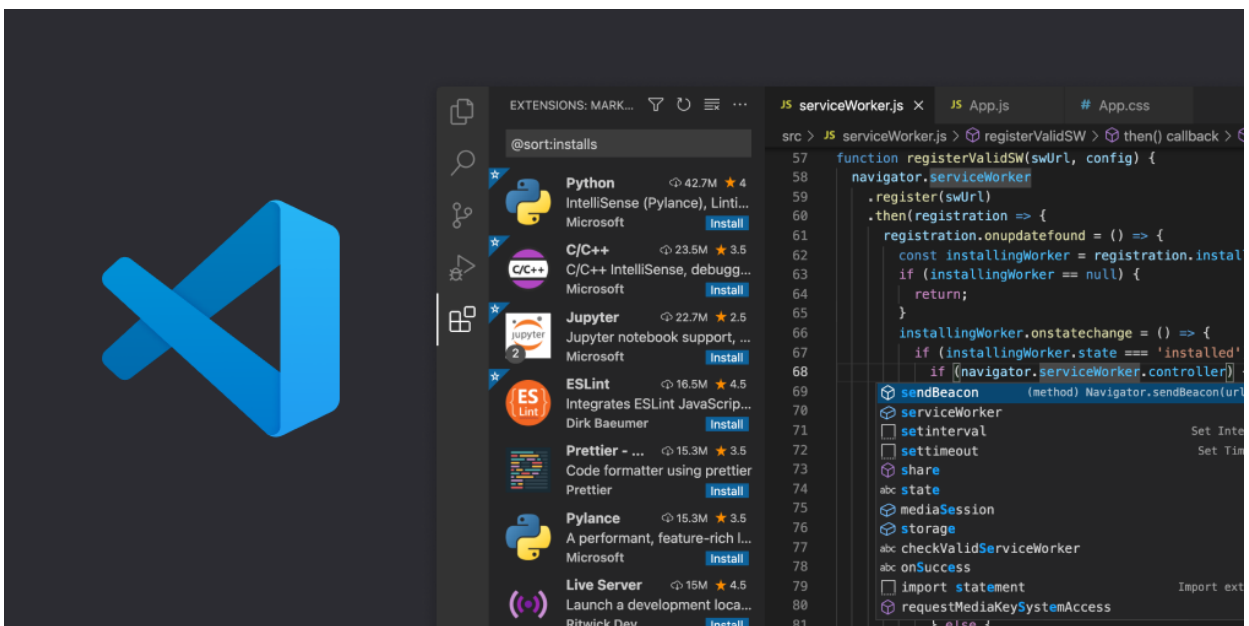


Рисунок 3.1 – Visual Studio Code [12]

VS Code пропонує багатий набір функцій, які значно прискорюють і спрощують процес розробки. Однією з ключових особливостей є вбудована підтримка інтелектуального завершення коду (IntelliSense) для багатьох мов програмування, включно з JavaScript, TypeScript, Python, C++, Java та багатьох інших. Це дає змогу розробникам швидко і точно писати код, уникаючи помилок і помилок синтаксису.

Розширюваність – ще одна сильна сторона VS Code. Через магазин розширень розробники можуть додавати нові мови, теми, налагоджувачі та інструменти для роботи з різними платформами і сервісами. Ця екосистема

розширень адаптує редактор під конкретні завдання та вподобання користувача, чи то веб-розробка, аналіз даних, чи то робота з хмарними сервісами.

Вбудовані інструменти розробника в VS Code включають налагоджувач, термінал, систему контролю версій Git, і багато іншого. Це означає, що ви можете налагоджувати код, виконувати команди і керувати змінами в коді прямо з редактора, без перемикання між різними додатками.

VS Code також вирізняється високою продуктивністю та адаптивністю. Він швидко запускається і може ефективно працювати навіть із великими проектами та файлами. Крім того, редактор має низку функцій для оптимізації робочого процесу, як-от мультикурсор, швидка навігація по коду, і настроювані гарячі клавіші.

Спільнота і підтримка навколо VS Code вражають. Microsoft регулярно оновлює редактор, додаючи нові функції та виправляючи помилки. Численні користувачі, розробники і компанії активно діляться досвідом, створюють розширення і пропонують вирішення проблем.

На закінчення, Visual Studio Code – це не просто текстовий редактор, а повноцінне середовище розробки. Його комбінація потужних функцій, розширюваності, продуктивності та активної підтримки робить його чудовим вибором для розробників усіх рівнів і спеціалізацій. Незалежно від того, чи пишете ви веб-додатки, чи займаєтеся науковими обчисленнями, чи створюєте мобільні додатки, VS Code надає інструменти та гнучкість, необхідні для ефективної та приємної роботи [12].

### 3.2 Вибір мов програмування

Код буде реалізовано трьома мовами програмування:

– HTML використовуватиметься як розмітка, що використовується для створення структури та вмісту веб-сторінок. У HTML визначатиметься структура і розмітка веб-сайту [12].



– CPP (C++) буде використовуватися для загального призначення, яка використовується для написання програм з використанням об'єктно-орієнтованого підходу. У цьому коді буде реалізовано обробку даних і логіку [12].

– JavaScript буде використано для надання взаємодії на стороні клієнта на веб-сторінках. JavaScript додає інтерактивність і динаміку до веб-сторінок, а також використовується для взаємодії з користувачем і маніпуляції веб-елементами [12].

### 3.3 Розроблення програмної частини

#### 3.3.1 Функціонал сайту, який потрібно реалізувати

Основна задача розробки сайту полягає у створенні модульної архітектури, яка дозволить постійно вдосконалювати, покращувати та додавати новий функціонал без значних витрат часу. Це досягається завдяки використанню сучасних веб-технологій і принципів розробки, таких як компонентний підхід, відокремлення логіки від представлення даних, і використання API для взаємодії з сервером. Повний програмний код відображено в додатку А.

Для зручності користувачів і виконання поставлених завдань нам потрібно зробити наступне:

а) зробити сторінку авторизації та реєстрації для нових користувачів, також зробити кнопку деавторизації на сайті:

1) створити сторінку авторизації, де зареєстровані користувачі можуть увійти за допомогою логіна (email) і пароля;

2) розробити форму реєстрації для нових користувачів, де вони зможуть ввести необхідні дані (ім'я, email, пароль тощо);

3) реалізувати кнопку деавторизації на всіх сторінках сайту для забезпечення безпеки користувача;

б) зробити всі вкладки інтеграцію з картами Google на головній вкладці:

- 1) додати всі вкладки для подальшого поліпшення;
- 2) додати розділ з Google Maps API, що дозволить користувачам візуально орієнтуватися та знаходити найближчі паркувальні майданчики;
- 3) на карті відмічати доступні паркінги з можливістю переходу на їх детальні сторінки;

в) зробити інформаційний розділ:

- 1) створити окремий розділ, де буде розміщена інформація про систему та використані технології;
- 2) списати переваги системи, принцип її роботи та користь для користувачів і міської інфраструктури;

г) зробити сторінки окремих паркінгів:

- 1) розробити індивідуальні сторінки для кожного паркінгу, де користувачі зможуть в реальному часі бачити схему паркінгу з відміченими вільними та зайнятими місцями;
- 2) використовувати кольорове кодування або символи для швидкого сприйняття статусу місць (зелений – вільне, червоний – зайняте);

г) зробити систему бронювання:

- 1) впровадити функціонал, який дозволить користувачам бронювати вільні місця на певний час;
- 2) після підтвердження бронювання, місце має відображатися як "зарезервоване" для інших користувачів;

д) передача інформації з датчиків і відображення на сайті:

- 1) розробити код для Arduino IDE, який збиратиме дані завдяки ультразвуковим датчикам, перевірятиме зайнятість місця та надсилатиме цю інформацію через IoT шлюз для подальшого використання;
- 2) розробити окремий блок для прийому інформації і відображення цієї інформації на веб-сайті.

### 3.3.2 Розробка сторінки авторизації та реєстрації.

Для зручності користувачів і виконання поставлених завдань необхідно реалізувати систему авторизації та реєстрації. Це передбачає створення сторінки авторизації, де зареєстровані користувачі зможуть увійти за допомогою логіна і пароля. Також потрібна форма реєстрації для нових користувачів, де вони зможуть ввести необхідні дані. Важливим елементом безпеки є кнопка деавторизації, яка має бути присутня на всіх сторінках сайту.

Програмний код для ініціалізації задумки (рисунок 3.2).

```

37 // Ініціалізація
38 document.addEventListener('DOMContentLoaded', function() {
39     showPage('login'); // Показати сторінку входу за замовчуванням
40
41     // Обробник форми входу
42     document.getElementById('loginForm').addEventListener('submit', function(event) {
43         event.preventDefault();
44         const username = document.getElementById('username').value;
45         const password = document.getElementById('password').value;
46         const role = checkLogin(username, password);
47         if (role) {
48             showNavBar();
49             showPage('home');
50             const expirationTime = new Date().getTime() + 10 * 60 * 1000; // Поточний час + 10 хвилин
51             localStorage.setItem('currentUser', username);
52             localStorage.setItem('currentRole', role);
53             localStorage.setItem('expirationTime', expirationTime);
54             showLoggedInUser(username);
55         } else {
56             alert('Неправильний логін або пароль.');
```

Рисунок 3.2 – Код javascript для реєстрації та авторизації

Відображення коду на веб-сайті зображено на рисунках 3.3-3.6.

## Kharkiv Smart Parking

### Авторизація

Логін:

Пароль:

[Увійти](#)

[Реєстрація](#) [Увійти як гість](#)

Рисунок 3.3 – Сторінка авторизації

## Kharkiv Smart Parking

### Реєстрація

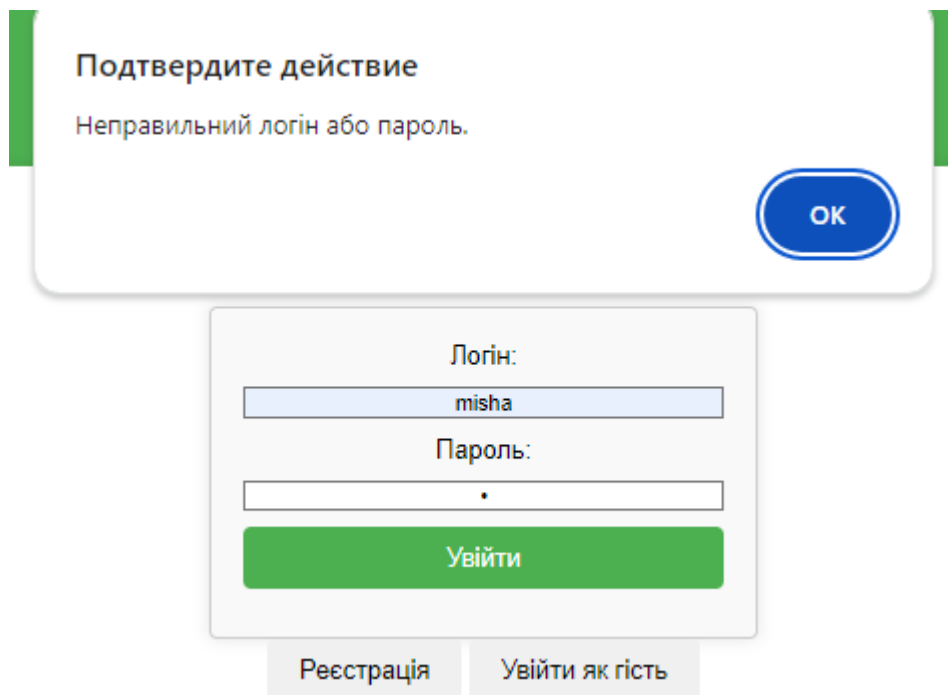
Логін:

Пароль:

[Зареєструватися](#)

[Вже зареєстровані? Увійти](#)

Рисунок 3.4 – Сторінка реєстрації



Подтвердите действие  
Неправильный логин або пароль.

OK

Логін:  
misha

Пароль:  
•

Увійти

Реєстрація Увійти як гість

Рисунок 3.5 – Результат вводу неправильного логіну або пароля

Ви увійшли як: misha | [Вихід](#)

Рисунок 3.6 – Кнопка виходу на всіх сторінках для деавторизації.

### 3.3.3 Розробка всіх вкладок та інтеграції з картами Google

Інтеграція з картами Google значно підвищить зручність користування сайтом. Додавання розділу з Google Maps API дозволить користувачам візуально орієнтуватися та знаходити найближчі паркувальні майданчики. На карті мають бути відмічені доступні паркінги з можливістю переходу на їх детальні сторінки.

Програмний код для ініціалізації (рисунок 3.8).

```

<nav id="nav-bar" style="display:none;">
  <ul>
    <li><a href="#" onclick="showPage('home')">Головна сторінка</a></li>
    <li><a href="#" onclick="showPage('about')">Про нас</a></li>
    <li><a href="#" onclick="showPage('parking20')">Парковка для мотоциклів</a></li>
    <li><a href="#" onclick="showPage('parking40')">Парковка для автомобілів</a></li>
    <li><a href="#" onclick="showPage('parkingBike')">Парковка для велосипедів</a></li>
  </ul>
</nav>

```

Рисунок 3.8 – Створення всіх вкладок сайту (html)

Вид вкладок на сайті (рисунок 3.9).

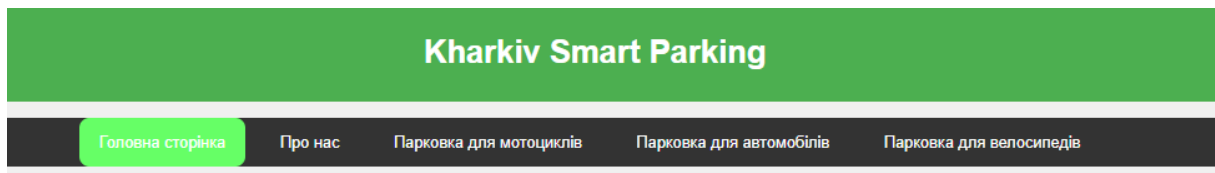


Рисунок 3.9 – Створені вкладки

Код для ініціалізація карти (рисунок 3.10).

```

// Ініціалізація карти
function initMap() {
  new google.maps.Map(document.getElementById("map"), {
    zoom: 13,
    center: { lat: 49.9935, lng: 36.2304 } // Координати Харкова
  });
}

```

Рисунок 3.10 – Код для ініціалізації карти (javascript)

Підключення API ключа GoogleMaps для відображення карти на веб-сайті (рисунок 3.11).

```

<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA1MN3C8CAPGS0JGT8dZAWFK1fAqs-h2vY&callback=initMap" async defer></script>

```

Рисунок 3.11 – Підключення API ключа GoogleMaps (html)

Відображення Google карти на веб-сайті (рисунок 3.12).

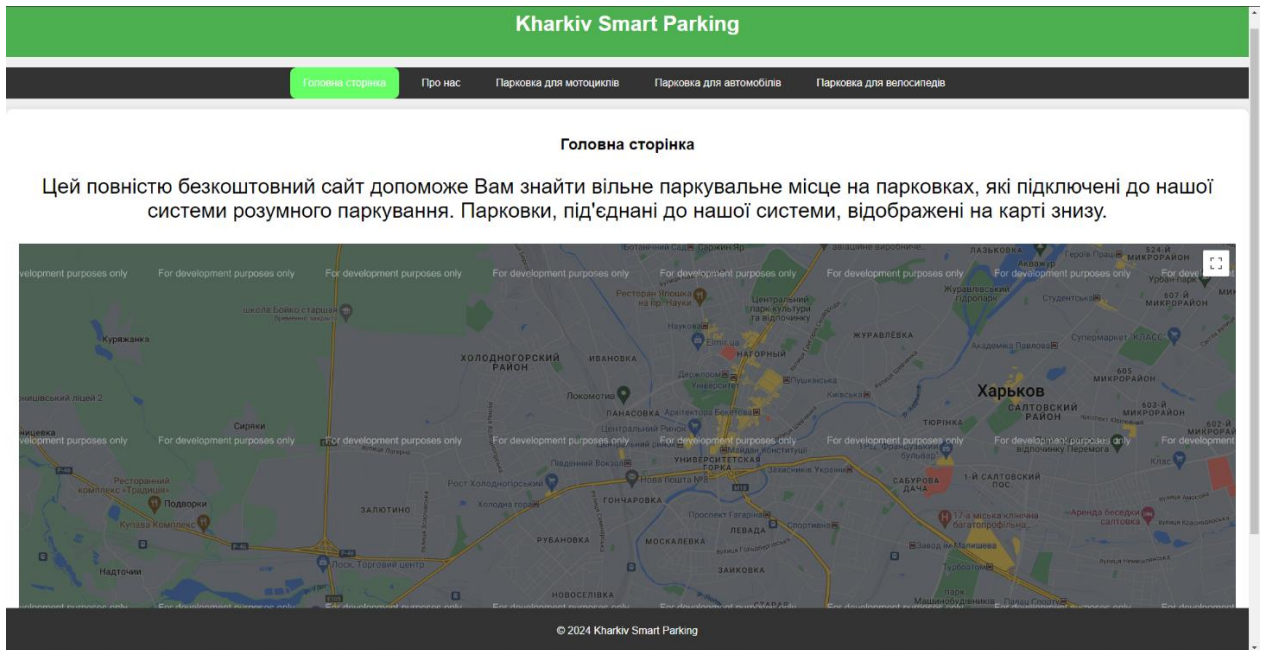


Рисунок 3.12 – Інтегрована карта Google

### 3.3.4 Розробка інформаційного розділу для загальної інформації

Для підвищення довіри користувачів і розуміння принципів роботи системи, необхідно створити інформаційний розділ. Тут буде розміщена інформація про використані технології, і будь-яка інша інформація, яка може бути потрібною користувачам.

Програмний код для ініціалізації задумки (рисунок 3.13).

```
<section id="about" class="content-section" style="display:none;">
  <h2>Про нас</h2>
  <p class="large-text">Ми використовуємо технологію ультразвукових датчиків для перевірки зайнятості стоянкових місць, щоб зекономити ваш час під час пошуку місць для паркування та надати інформацію з мінімальною затримкою, щоб уникнути ситуацій, коли ви приїжджаєте на зайняте місце ви можете його забронювати для себе. Ми додаємо до системи всі бажані компанії, які хочуть приєднати свої парковки до загальної мережі.</p>
  
</section>
```

Рисунок 3.13 – Додання інформації на вкладку “Про нас” (html)

Відображення інформації на вкладці “Про нас” на веб-сайті (рисунок 3.14).

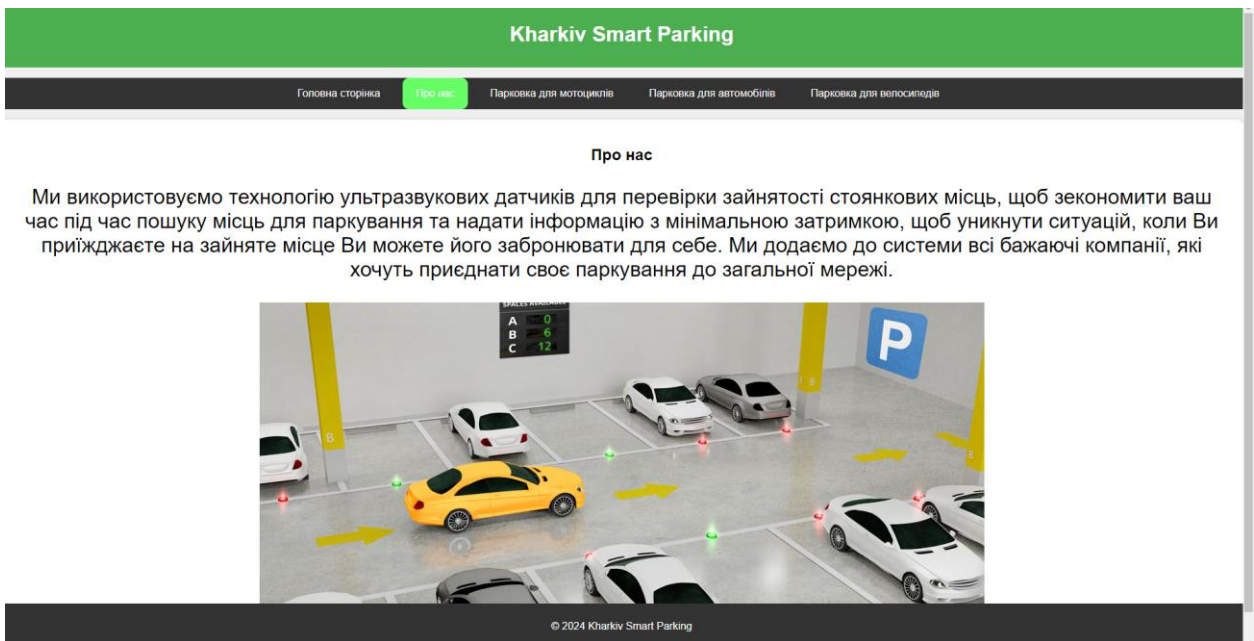


Рисунок 3.14 – Вигляд вкладки “Про нас”

### 3.3.5 Розробка сторінок окремих паркінгів

Ключовим елементом сайту мають стати індивідуальні сторінки для кожного паркінгу. На цих сторінках користувачі в реальному часі зможуть бачити схему паркінгу з відміченими вільними та зайнятими місцями. Використання кольорового кодування або символів (зелений – вільне, червоний – зайняте) забезпечить швидке сприйняття статусу місць.

У майбутньому сюди буде надходити інформація з датчиків, що дозволить автоматично оновлювати стан паркомісць на сторінці. Це забезпечить актуальність даних та зручність для користувачів, які зможуть оперативно отримувати інформацію про наявність вільних місць на паркінгу.

Програмний код для ініціалізації задумки зображений на рисунках 3.15-3.16.



```

64 .parking-grid {
65     display: grid;
66     grid-template-columns: repeat(10, 50px);
67     gap: 10px;
68     justify-content: center;
69 }
70
71 .parking-grid.large-gap {
72     gap: 20px;
73 }
74
75 .parking-space {
76     width: 50px;
77     height: 50px;
78     background: #ccc;
79     display: flex;
80     align-items: center;
81     justify-content: center;
82     cursor: pointer;
83     border: 1px solid #999;
84 }
85
86 .parking-space.occupied {
87     background: #ff6666;
88 }
89
90 .parking-space.free {
91     background: #66ff66;
92 }
93
94 .parking-image {
95     display: block;
96     margin: 20px auto;
97     max-width: 100%;
98     height: auto;
99 }

```

Рисунок 3.15 – Стили вкладок із сітками паркінгу (css)

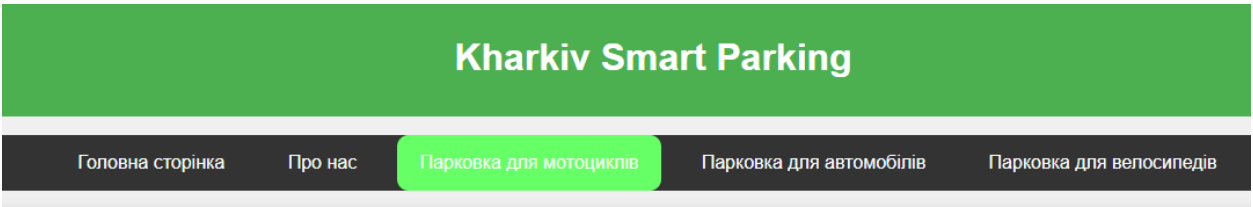
```

24 // Функція для створення сітки парковки
25 function createParkingGrid(gridId, places, sensorData, updateSensorData) {
26     const grid = document.getElementById(gridId);
27     grid.innerHTML = ''; // Очищення попередніх даних
28     for (let i = 1; i <= places; i++) {
29         const space = document.createElement('div');
30         space.className = `parking-space ${sensorData[i - 1] ? 'free' : 'occupied'}`;
31         space.dataset.space = i; // Додаємо атрибут data-space з номером місця
32         space.innerText = i;
33         grid.appendChild(space);
34     }
35 }

```

Рисунок 3.16 – Функція для створення сітки паркінгу

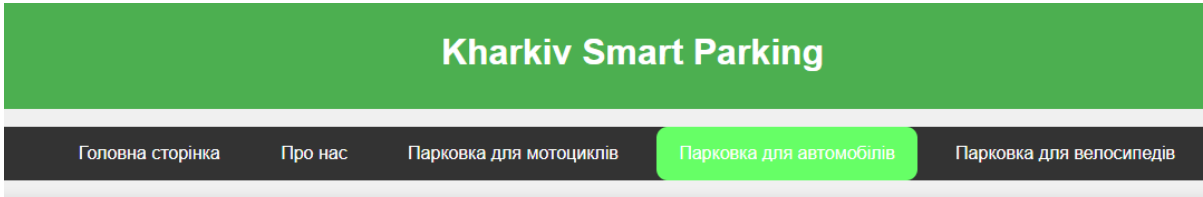
Відображення коду на веб-сайті зображено на рисунках 3.17-3.19.



**Парковка для мотоциклів**

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

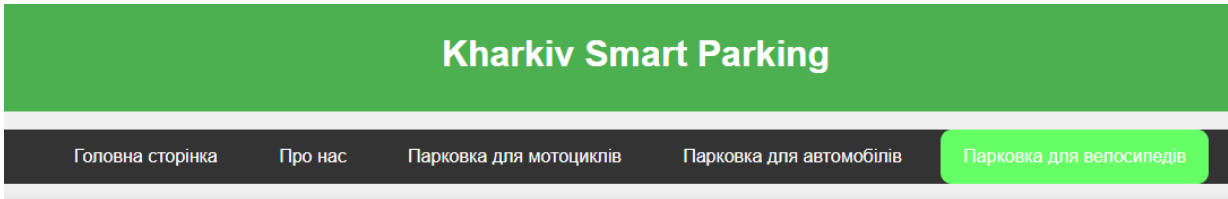
Рисунок 3.17 – Вкладка парковки для мотоциклів



**Парковка для автомобілів**

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Рисунок 3.18 – Вкладка парковки для автомобілів



**Парковка для велосипедів (40 місць)**

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Рисунок 3.19 – Вкладка парковки для велосипедів

### 3.3.6 Розробка системи бронювання

Система бронювання додасть зручності для користувачів, які планують своє прибуття заздалегідь. Реалізація функціоналу, який дозволить бронювати вільні місця на певний час, включає форму бронювання з вказанням часу прибуття і тривалості паркування. Після підтвердження бронювання, місце має відображатися як "зарезервоване" для інших користувачів.

Система також надаватиме можливість моніторингу використання паркінгу адміністраторам. Це включає перегляд статистики зайнятості місць, аналіз пікових годин і виявлення тенденцій використання. Адміністратори зможуть оптимізувати розподіл ресурсів, підвищуючи ефективність управління паркінгом. Використання таких даних допоможе приймати обґрунтовані рішення щодо розширення паркувальних зон або впровадження додаткових сервісів для користувачів.

Програмний код для ініціалізації зображений на рисунках 3.20-3.21.

```

121 // Функція для бронювання місця парковки
122 function bookParkingSpace(gridId) {
123     const spaceNumber = document.getElementById(`parkingSpace${gridId}`).value;
124     const bookingTime = document.getElementById(`bookingTime${gridId}`).value;
125     const grid = document.getElementById(`grid${gridId}`);
126     const parkingSpace = grid.querySelector(`.parking-space[data-space="${spaceNumber}"]`);
127     if (parkingSpace.classList.contains('free')) {
128         parkingSpace.classList.remove('free');
129         parkingSpace.classList.add('occupied');
130         console.log(`Місце ${spaceNumber} заброньовано на ${bookingTime} хвилин`);
131     } else {
132         console.log(`Місце ${spaceNumber} вже зайнято`);
133     }
134 }

```

Рисунок 3.20 – Функція для бронювання місця парковки (javascript)

```

63 </section>
64 <section id="parking20" class="content-section" style="display:none;">
65 <h2>Парковка для мотоциклів</h2>
66 <div class="parking-grid" id="grid20"></div>
67 <form id="bookingForm20">
68 <label for="parkingSpace20">Номер місця:</label>
69 <input type="number" id="parkingSpace20" name="parkingSpace20" min="1" max="20" required>
70 <label for="bookingTime20">Час бронювання (хв):</label>
71 <input type="number" id="bookingTime20" name="bookingTime20" min="1" required>
72 <button type="button" onclick="bookParkingSpace(20)">Забронювати</button>
73 </form>
74 
75 </section>
76 <section id="parking40" class="content-section" style="display:none;">
77 <h2>Парковка для автомобілів</h2>
78 <div class="parking-grid" id="grid40"></div>
79 <form id="bookingForm40">
80 <label for="parkingSpace40">Номер місця:</label>
81 <input type="number" id="parkingSpace40" name="parkingSpace40" min="1" max="40" required>
82 <label for="bookingTime40">Час бронювання (хв):</label>
83 <input type="number" id="bookingTime40" name="bookingTime40" min="1" required>
84 <button type="button" onclick="bookParkingSpace(40)">Забронювати</button>
85 </form>
86 
87 </section>
88 <section id="parkingBike" class="content-section" style="display:none;">
89 <h2>Парковка для велосипедів (40 місць)</h2>
90 <div class="parking-grid" id="gridBike"></div>
91 <form id="bookingFormBike">
92 <label for="parkingSpaceBike">Номер місця:</label>
93 <input type="number" id="parkingSpaceBike" name="parkingSpaceBike" min="1" max="40" required>
94 <label for="bookingTimeBike">Час бронювання (хв):</label>
95 <input type="number" id="bookingTimeBike" name="bookingTimeBike" min="1" required>
96 <button type="button" onclick="bookParkingSpace('Bike')">Забронювати</button>
97 </form>
98 
99 </section>

```

Рисунок 3.21 – Створення функції бронювання для трьох вкладок паркувальних місць (html)

Ця функція допоможе користувачам уникнути зайвих кіл під час пошуку вільного місця, особливо в годину пік. За допомогою інтерактивної карти вони зможуть бачити актуальну інформацію про наявність місць, а також швидко оцінити ситуацію на паркінгу. Це зменшить кількість автомобілів, що крутяться в пошуках паркування, що своєю чергою знизить навантаження на дорожню інфраструктуру і покращить загальну екологічну ситуацію в місті.

Відображення коду на веб-сайті зображено на рисунках 3.22-3.25.

## Kharkiv Smart Parking

Головна сторінка
Про нас
Парковка для мотоциклів
Парковка для автомобілів
Парковка для велосипедів

### Парковка для мотоциклів

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Номер місця:

Час бронювання (хв):

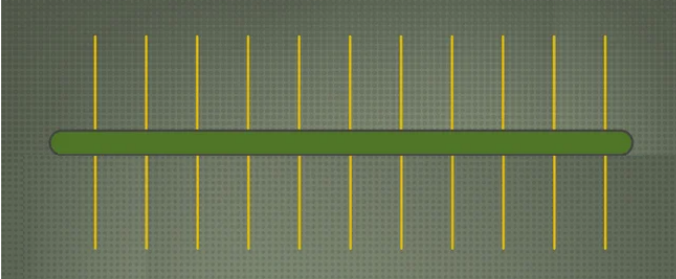


Рисунок 3.22 – Функція бронювання місця для мотоциклів

## Kharkiv Smart Parking

Головна сторінка
Про нас
Парковка для мотоциклів
Парковка для автомобілів
Парковка для велосипедів

### Парковка для автомобілів

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Номер місця:

Час бронювання (хв):

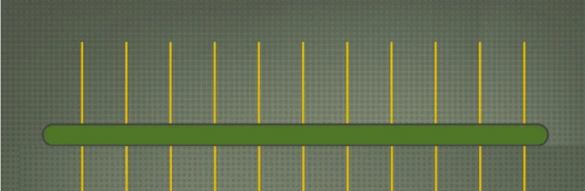


Рисунок 3.23 – Функція бронювання місця для автомобілів

## Kharkiv Smart Parking

Головна сторінка
Про нас
Парковка для мотоциклів
Парковка для автомобілів
Парковка для велосипедів

### Парковка для велосипедів (40 місць)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Номер місця:

Час бронювання (хв):

Забронювати

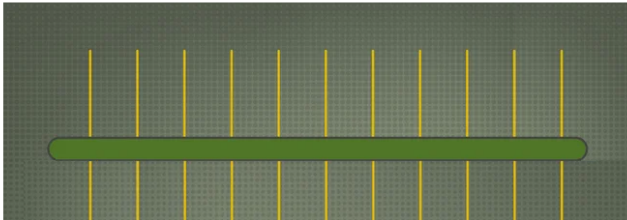


Рисунок 3.24 – Функція бронювання місця для велосипедів

## Kharkiv Smart Parking

Головна сторінка
Про нас
Парковка для мотоциклів
Парковка для автомобілів
Парковка для велосипедів

### Парковка для автомобілів

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Номер місця:

Час бронювання (хв):

Забронювати

Рисунок 3.25 – Приклад роботи функції бронювання місця для автомобілів

### 3.3.7 Передача даних з датчиків і відображення на сайті

Для підключення ESP8266 до Arduino та передачі даних по Wi-Fi виконаємо наступні кроки, спершу налаштуємо апаратну частину, підключимо датчик HC-SR04 та модуль ESP8266 до Arduino. Потім створимо скетч для Arduino, який буде збирати дані з датчика, перевіряти стан паркувального місця та передавати ці дані по Wi-Fi.

Апаратне підключення.

Підключення датчика HC-SR04:

VCC -> 5V на Arduino

GND -> GND на Arduino

Trig -> цифровий пін 9 на Arduino

Echo -> цифровий пін 10 на Arduino

Підключення ESP8266:

VCC -> 3.3V на Arduino

GND -> GND на Arduino

TX -> RX на Arduino (через дільник напруги)

RX -> TX на Arduino

CH\_PD -> 3.3V на Arduino

Скетч Arduino зображено на рисунку 3.26.

```

1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3
4 const char* ssid = "VIKI"; // SSID Wi-Fi мережі
5 const char* password = "88888888"; // Пароль Wi-Fi мережі
6
7 WiFiServer server(80);
8
9 const int trigPin = 9;
10 const int echoPin = 10;
11
12 void setup() {
13   Serial.begin(115200);
14   pinMode(trigPin, OUTPUT);
15   pinMode(echoPin, INPUT);
16
17   // Підключення до Wi-Fi
18   Serial.println();
19   Serial.println();
20   Serial.print("Підключення до ");
21   Serial.println(ssid);
22
23   WiFi.begin(ssid, password);
24
25   while (WiFi.status() != WL_CONNECTED) {
26     delay(500);
27     Serial.print(".");
28   }
29
30   Serial.println("");
31   Serial.println("WiFi підключено");
32   Serial.println("IP адреса: ");
33   Serial.println(WiFi.localIP());
34
35   server.begin();
36 }
37
38 void loop() {
39   long duration;
40   int distance;
41
42   // Вимірювання відстані з датчика HC-SR04
43   digitalWrite(trigPin, LOW);
44   delayMicroseconds(2);
45   digitalWrite(trigPin, HIGH);
46   delayMicroseconds(10);
47   digitalWrite(trigPin, LOW);
48
49   duration = pulseIn(echoPin, HIGH);
50   distance = duration * 0.034 / 2; // Розрахунок відстані в см
51
52   // Перевірка зайнятості місця
53   int status = (distance > 400) ? 1 : 0;
54
55   // Надсилання даних по Wi-Fi
56   WiFiClient client = server.available();
57   if (client) {
58     Serial.println("Новий клієнт.");
59     String request = client.readStringUntil('\r');
60     Serial.println(request);
61     client.flush();
62
63     String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
64     s += "Parking status: ";
65     s += status;
66     client.print(s);
67     delay(1);
68     Serial.println("Клієнт відключився.");
69   }
70   delay(2000); // Очікування перед наступним вимірюванням
71 }

```

Рисунок 3.27 – Скетч Arduino

Розробимо функцію для прийому даних і оновлення їх на веб сайті, код для реалізації зображений на рисунку 3.28.



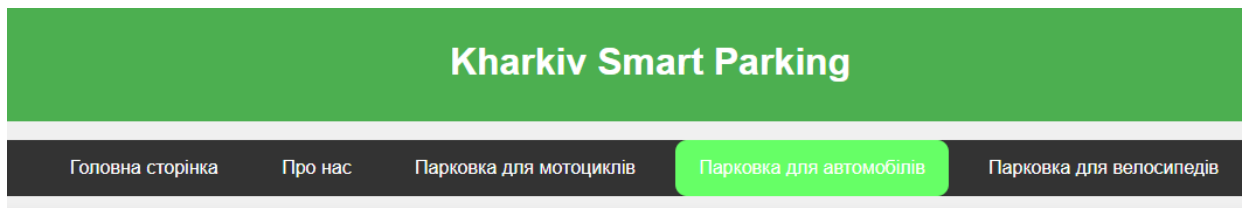
```

1  async function fetchParkingStatus() {
2      try {
3          const response = await fetch('http://<ESP8266_IP_ADDRESS>');
4          const data = await response.text();
5          const status = parseInt(data.match(/Parking status: (\d)/)[1], 10);
6          updateSensorData(status);
7      } catch (error) {
8          console.error('Помилка при отриманні даних з ESP8266:', error);
9      }
10 }
11
12 function updateSensorData(status) {
13     // Оновлення стану паркувального місця
14     const space = document.querySelector('.parking-space');
15     space.className = `parking-space ${status ? 'free' : 'occupied'}`;
16 }
17
18 setInterval(fetchParkingStatus, 5000); // Оновлення кожні 5 секунд

```

Рисунок 3.28 – Функція для прийому даних і оновлення їх на веб сайті

Відображення зайнятості місць (рисунок 3.29).



#### Парковка для автомобілів

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Рисунок 3.29 – 1-4 датчики HC-SR04 показують, що місця зайняті

### 3.4 Висновки до розділу

У цьому розділі було розроблено модульну архітектуру сайту, яка забезпечує можливість постійного вдосконалення, покращення та додавання

нового функціоналу без значних витрат часу. Це досягнуто завдяки використанню сучасних веб-технологій і принципів розробки. Створено систему авторизації та реєстрації користувачів, що підвищує безпеку та персоналізацію досвіду. Інтеграція Google Maps API дозволяє користувачам візуально орієнтуватися та знаходити найближчі паркінги, а інформаційний розділ "Про нас" підвищує довіру та розуміння роботи системи.

Ключовим елементом стали індивідуальні сторінки для різних типів паркінгів, де користувачі в реальному часі бачать схему з відміченими вільними та зайнятими місцями. Система бронювання додає зручності, дозволяючи резервувати місця на певний час. Також продемонстровано можливість інтеграції фізичних датчиків через мікроконтролери, що є основою для створення повноцінної системи розумного паркування.

Загалом, розроблена програмна частина забезпечує зручний, інтерактивний та інформативний інтерфейс, а також дозволяє легко інтегрувати фізичні компоненти в загальну архітектуру. Це створює потужну основу для системи, яка може значно покращити досвід користувачів та ефективність міської інфраструктури паркування.

## 4 ОХОРОНА ПРАЦІ

### 4.1 Здоров'я та безпека при роботі за комп'ютером

Робота за комп'ютером передбачає специфічні ризики, що пов'язані з тривалим часом, проведеним за комп'ютером, статичним навантаженням та інтелектуальним напруженням. Розглянемо детально основні правила безпеки для програмістів [14].

Ергономіка робочого місця:

- робочий стіл і крісло: висота робочого столу повинна бути такою, щоб лікті знаходилися під кутом 90 градусів або трохи більше при роботі з клавіатурою. Крісло повинно мати регулювання висоти, нахилу спинки та підтримку попереку для запобігання розвитку захворювань хребта;

- розташування монітора: монітор слід розташовувати на відстані 50-70 см від очей, причому верхня частина екрану повинна бути на рівні або трохи нижче рівня очей. Це допоможе уникнути перенапруження очей та м'язів шиї;

- освітлення: робоче місце повинно бути добре освітлене, причому світло не повинно створювати відблиски на моніторі. Рекомендується використовувати додаткові джерела освітлення, такі як настільні лампи з регульованим рівнем світла.

Охорона здоров'я:

- профілактика зорового перенапруження: рекомендується робити перерви кожні 20-30 хвилин, виконуючи вправи для очей, такі як фокусування на далеких і ближніх об'єктах по черзі. Також важливо підтримувати достатній рівень вологості повітря в приміщенні;

- фізична активність: для уникнення проблем зі спиною та суглобами слід регулярно виконувати прості фізичні вправи та розтяжки. Програмістам рекомендується проводити "розминки" кожні 1-2 години роботи.

### Психологічне здоров'я:

- стрес-менеджмент: важливо навчитися управляти стресом, що виникає через високе інтелектуальне навантаження. Це можна досягти за допомогою регулярних перерв, фізичних вправ, медитацій та інших технік релаксації;

- організація робочого часу: ефективний тайм-менеджмент допомагає уникнути перенапруження та вигорання. Рекомендується використовувати методики планування, такі як Pomodoro, для чіткого розподілу робочого часу та перерв.

### Інформаційна безпека:

- захист даних: програмісти повинні дотримуватись правил збереження конфіденційності та безпеки даних, використовувати складні паролі, двофакторну аутентифікацію та регулярно оновлювати програмне забезпечення для захисту від шкідливих програм;

- резервне копіювання: регулярне створення резервних копій даних є критично важливим для запобігання втраті інформації через збої в системі або кібератаки.

### Пожежна безпека:

- електробезпека: програмістам необхідно стежити за станом електропроводки та обладнання. Важливо не перевантажувати електромережу, використовувати справні розетки та подовжувачі;

- пожежогасіння: на робочому місці повинні бути в наявності засоби пожежогасіння, а програмісти повинні бути навчені їх використовувати. Важливо знати план евакуації та місцезнаходження найближчих виходів.

Дотримання цих правил сприяє створенню безпечних умов праці за комп'ютером та знижує ризик виникнення професійних захворювань і нещасних випадків.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз існуючих систем оптимізації паркування автотранспорту, зокрема їх технічних характеристик та способів реалізації. Розглянуто класифікацію систем паркінгу, а також аналіз проблем, з якими стикаються користувачі автостоянок.

Розроблено загальну структурну схему системи автоматизації для оптимізації паркування, де відображені основні компоненти, необхідні для її функціонування. Проведено аналіз та вибір технічних засобів, включаючи ультразвукові датчики, контролери Arduino, Wi-Fi модулі ESP8266, а також компоненти для реалізації програмного забезпечення.

Створено програмну частину, яка включає розробку веб-додатка для моніторингу та бронювання паркувальних місць у реальному часі. Веб-додаток дозволяє користувачам бачити доступні місця на парковці, забронювати місце заздалегідь та отримувати інформацію про зміну стану заброньованих місць. Розробка програмного забезпечення здійснювалася в середовищі Visual Studio Code із використанням мов HTML, C++, JavaScript.

Проведено тестування програмного забезпечення для забезпечення його надійної роботи та визначення можливих проблем.

Робота охопила питання охорони праці та безпеки.

Таким чином, в результаті виконання кваліфікаційної роботи, підвищено ефективність управління паркувальними місцями на підземній парковці.

Також, отримані результати роботи можна віднести до Цілі сталого розвитку 9 "Промисловість, інновації та інфраструктура", а саме п.9.3 "Забезпечити доступність дорожньотранспортної інфраструктури, яка базується на використанні інноваційних технологій, зокрема через розширення форм участі держави у різних інфраструктурних проектах"

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с.

2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. - 29 с.

3. Методичні вказівки до Підготовки атестаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми: «Автоматизація та комп'ютерно-інтегровані технології» / упоряд.: І. Ш. Невлюдов, О. В. Токарева, Г. В. Пономарьова. – Харків: ХНУРЕ, – 2019. – 36 с.

4. Автоматична парковка Park System. URL: <https://overlook.expert/avtomaticheskaiia-parkovka-s-dvumia-stoikami-i-terminalom-oplaty-vezdnaia-i-vyezdnaia/> (дата звернення 08.05.2024).

5. Автоматичні парковочні системи Park System. URL: <https://parksystem.com.ua/ua/> (дата звернення 08.05.2024).

6. AFAPARK інноваційні рішення з управління парковками URL: <http://www.afapark.com/EN/index.html> (дата звернення 08.05.2024).

7. Система контролю та оплати стоянок FIDPARK URL: <https://fidpark.com/content/en/295/License-Plate-Recognition-LPR-ANPR-.html> (дата звернення 08.05.2024).

8. Makerobot. Як підключити датчик HC-SR04 до Arduino UNO. URL: <https://makerobot.com.ua/How-to-connect-HC-SR04-to-Arduino> (дата звернення 25.05.2024).

9. Arfuino. Ультразвуковий датчик відстані HC-SR04. URL: <https://arduino.ua/prod182-ultrazvukovoi-datchik-rasstoyaniya-hc-sr04> (дата звернення 25.05.2024).

10. Miniboard. Ультразвуковий датчик відстані HC-SR04. URL: <https://miniboard.com.ua/sensors/68-ultrazvukovoj-datchik-rasstoyaniya-hc-sr04.html> (дата звернення 25.05.2024).

11. HobbyTech IoT шлюз ESP8266. URL: <https://hobbytech.com.ua/%D0%B7%D0%BD%D0%B0%D0%BA%D0%BE%D0%BC%D0%B8%D0%BC%D1%81%D1%8F-%D1%81-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%BC-esp8266-%D0%BF%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D0%B5%D0%B5/> (дата звернення 20.05.2024).

12. Visual Studio Code - Code Editing. URL: <https://code.visualstudio.com/> (дата звернення 25.05.2024).

13. GitHub: Let's build from here URL: <https://github.com/> (дата звернення 31.05.2024).

14. FoxmindEd. Робоче місце програміста. URL: <https://foxminded.ua/roboche-mistse-prohramista/> (дата звернення 08.06.2024).

15. Технології Інтернету речей в управлінні пристроями на мікроконтролерах [Електронний ресурс] : навч. посібник / І. Ш. Невлюдов, В. А Андрусевич, С. П. Новоселов, О. Г. Резніченко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – електрон. вид. – Харків: ХНУРЕ, 2023. – 214 с.