A PROJECT REPORT ON

"SMART PARKING SYSTEM USING ARDUINO WITH WEB PORTAL"

Submitted To

BANGALORE NORTH UNIVERSITY

Tamaka, Kolar-563103

In partial fulfilment of the requirement for the award of degree

MASTER OF COMPUTER APPLICATION

Submitted by

Kalimisetti Badrinath (P19CJ22S126033)

Anudeep Thatapudi (P19CJ22S126039)

3rd Sem MCA

Under the guidance of

Mr. Gopi Krishna T L,

Asst. Professor Dept. Of Computer Application

NAGARJUNA
COLLEGE OF MANAGEMENT STUDIES

NAGARJUNA COLLEGE OF MANAGEMENT STUDIES

DEPARTMENT OF COMPUTER APPLICATION

2023 - 2024

NAGARJUNA COLLEGE OF MANAGEMENT STUDIES

Chikkaballapur- 562 101

DEPARTMENT OF COMPUTER APPLICATION

CERTIFICATE

This is to certify that, the project work carried out by **Kalimisetti Badrinath (**P19CJ22S126033**)** and **Anudeep Thatapudi (**P19CJ22S126039**)** has completed 3rd Semester project work entitled **"Smart Parking System Using Arduino with Web Portal"** as partial fulfillment for the award of Master in Computer Application Degree, during the academic year 2023-2024 under my supervision. It is certified that all corrections suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements concerning the project work prescribed for the said degree.

Signature of Guide

Mr. Gopi Krishna T L

Asst. Professor, Dept. of MCA

NCMS, Chikkaballapur

Signature of HOD

Mr. Ajith S

HOD Dept. of MCA,

NCMS, Chikkaballapur.

EXTERNAL VIVA

Name of the Examiner **Signature with Date**

DECLARATION

We, **Kalimisetti Badrinath** and **Anudeep Thatapudi** Students of 3rd Semester MCA, NCMS, Chikkaballapur affiliated to Bangalore North University, bearing registration numbers **P19CJ22S126033** and **P19CJ22S126039** respectively. Hereby declare that the project entitled **"Smart Parking System Using Arduino with Web Portal"** has been carried out by us under the supervision of internal guide, **Ajith S**, Asst. Professor, Dept. of MCA, NCMS, Chikkaballapur, and submitted in partial fulfillment of the requirements for the award of the degree Master of Computer Applications by Bangalore North University during the academic year 2023-24. This report has not been submitted to any other organization or university for the award of any degree or certificate.

Kalimisetti Badrinath (P19CJ22S126033)

Anudeep Thatapudi (P19CJ22S126033)

ACKNOWLEDGEMENT

The completion of this project work would not have been possible without the help of the following people and departments to whom we would like to express our deepest appreciation and gratitude.

We wish to express our sincere gratitude to **Dr. Anandamma N**, Principal, Nagarjuna College of Management Studies for her valuable support and encouragement during the course of this project work.

We also wish to express our deep gratitude to **Mr. Ajith S,** Head, Department of MCA, NCMS for her valuable support and encouragement during the course of this project work.

We also take this opportunity to express our profound and wholehearted thanks to our internal guide **Mr**. **Ajith S**, Asst. Professor, Dept. Of MCA, NCMS for his valuable guidance and support during the course of this project work.

We would like to express our deep gratitude to **all the staff members of Nagarjuna** College of Management Studies for their valuable support and encouragement during the course of this project work.

Last but not least, we also like to thank my parents and friends for their valuable support and encouragement throughout my project work.

Kalimisetti Badrinath (P19CJ22S126033)

Anudeep Thatapudi (P19CJ22S126033)

Abstract

This project focuses on developing a smart parking system using Arduino, infrared (IR) sensors, servo motors, and a power supply board to address the growing challenges of urban parking management. With urbanization leading to increased vehicle numbers and parking difficulties, traditional systems often fail to provide accurate slot availability information, resulting in prolonged search times and congestion. The proposed solution integrates Arduino microcontrollers with IR sensors to detect vehicle presence in parking slots. This data is transmitted to a local web application via serial communication, which is developed using Python Flask. The web application provides users with real-time updates on parking slot availability and allows for seamless reservation and monitoring. Additionally, servo motors are used to control physical barriers, enabling automated slot reservation. By leveraging these technologies, the system aims to optimize parking space utilization, enhance user convenience, and contribute to efficient urban traffic management. The project demonstrates a holistic approach to modernizing parking systems through the integration of hardware and software, ensuring effective management of urban parking challenges.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The growing urban population and vehicle ownership have led to significant challenges in managing parking spaces. Urbanization has led to a significant increase in vehicle ownership, posing challenges for effective parking management in urban areas. Traditional parking systems often struggle with providing accurate and real-time slot availability information, leading to congestion and frustration among drivers. The core objective of this project is to create a solution that enhances the efficiency of parking space utilization while improving user experience through real-time updates and seamless reservation capabilities. By integrating Arduino microcontrollers with IR sensors, the system can accurately detect the presence of vehicles in parking slots. This data is then transmitted to a local web application developed using Python Flask, a lightweight web framework known for its simplicity and flexibility. The web application serves as the interface through which users can monitor real-time parking slot availability, make reservations, and receive immediate updates on parking status changes. Through the integration of hardware (Arduino, IR sensors, servo motors, power supply board) and software (Python Flask, HTML/CSS/JavaScript), the project aims to provide a comprehensive and efficient solution to urban parking management challenges. By optimizing parking space utilization and reducing search times for available slots.

1.1 Background

The demand for effective parking management systems has escalated significantly. To address this challenge, our project proposes a robust solution combining Arduino and Python Flask technologies for real-time parking slot detection and reservation. Utilizing infrared (IR) sensors interfaced with Arduino, the system detects slot availability and communicates this data to a local web application via serial communication. This web application, developed using Python Flask, empowers users to seamlessly book and monitor parking slots in real-time. By integrating these components, our solution aims to optimize parking operations, alleviate congestion.

1.2 Problem statement

Traditional parking management systems typically rely on manual checks or basic sensors that do not offer real-time information on slot availability. This outdated approach results in suboptimal utilization of parking spaces, leading to user frustration and exacerbating traffic congestion issues. To address these shortcomings, there is a critical need for an integrated

system capable of detecting parking slot availability in real-time and facilitating online reservations. Such a system would not only streamline parking operations but also enhance user convenience by providing accurate, up-to-date information and enabling efficient planning of parking needs.

1.3 Objective of the study

- Implement IR sensors connected to an Arduino to detect the availability of parking slots.
- Establish reliable serial communication between the Arduino and a laptop.
- Develop a Python Flask-based web application for users to monitor and book parking slots.
- Design the system to be easily scalable for different parking lot sizes and configurations.
- Ensure the system provides accurate and consistent real-time updates.

1.4 Scope of the study

The scope of this study encompasses the design, development, and implementation of a parking management system using Arduino and Python Flask technologies. It focuses on integrating infrared (IR) sensors with Arduino to detect real-time parking slot availability and relay this information to a local web application via serial communication. The web application, constructed with Python Flask, enables users to efficiently reserve and monitor parking slots online. The study aims to evaluate the effectiveness of this integrated approach in optimizing parking operations, mitigating congestion, and improving user experience within urban environments. Key aspects include sensor accuracy, system reliability, user interface design, and overall system performance under varying conditions of usage and scale.

1.5 Limitations of the study

While our proposed parking management system offers promising advancements, several limitations should be noted. Firstly, the accuracy of slot detection using IR sensors may be influenced by environmental factors such as lighting conditions and sensor calibration. Additionally, the scalability of the system to handle a large number of parking slots and users simultaneously needs careful consideration to ensure seamless performance. Moreover, the reliance on local web application hosting limits accessibility beyond local network coverage. Furthermore, the effectiveness of the system in diverse urban settings and its integration with existing infrastructure pose potential challenges that warrant further exploration. Despite these limitations, our study provides a foundational framework for enhancing urban parking.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Existing Literature

The existing literature on parking management systems highlights a range of methodologies and technologies aimed at addressing urban parking challenges. Numerous studies have explored the use of sensor networks, including ultrasonic, infrared, and magnetic sensors, for real-time parking detection. Research has shown that these sensors can significantly improve the accuracy of slot availability information. Moreover, the integration of Internet of Things (Arduino with Web Portal) technologies with cloud computing has been extensively investigated, demonstrating the potential for scalable and robust parking management solutions. Several papers have also examined the role of machine learning algorithms in predicting parking space occupancy and enhancing reservation systems. Additionally, mobile applications and web-based platforms have been evaluated for their effectiveness in providing user-friendly interfaces for real-time parking information and reservations. Collectively, these studies underscore the need for a comprehensive approach that combines sensor technology, data processing, and user interface design to create efficient and user-centric parking management systems.

2.2 Comparison with Existing System

When comparing this Arduino and Python Flask-based parking management system with existing traditional systems, the advantages are striking. Traditional methods often rely on manual checks or outdated sensors that fail to provide real-time updates, resulting in inefficiencies and user frustration due to inaccurate information and extended search times for available slots. Furthermore, these systems typically lack a user-friendly interface for real-time monitoring and reservations.

In contrast, our proposed solution harnesses the precision of IR sensors interfaced with Arduino for accurate slot availability detection. Real-time data transmission to a Python Flask-powered web application ensures users have immediate access to current parking information and can effortlessly reserve slots. This not only enhances user convenience but also significantly reduces congestion and optimizes parking space utilization. By offering a reliable, efficient, and user-centric alternative, our system represents a substantial advancement in parking management technology, addressing critical pain points and improving the overall urban parking experience.

Title	Author(s)	Year	Methodology	Result
Carnatic Smart Parking System Using Internet of Things (Arduino with Web Portal) (S.R, 2015).[1]	Mr. Basavaraju S R	2015	The proposed system utilizes the Internet of Things (Arduino with Web Portal) to create a Smart Parking System (SPS) that aids users in finding available parking slots efficiently.	Develop a system with less cost with more performance.
Smart Parking System Based on the Internet of Things.[2]	Deepthi S., Anil A. R.	2016	smart parking systems that leverage the Internet of Things (Arduino with Web Portal) to address urban parking challenges	Reduced Traffic Congestion and User Interfaces
Carmate Car Parking System Commanded by Android Application.[3]	D. J. Bonde	2012	Automating car parking using an Android application.	A miniature model of a carmate car parking system
Arduino with Web Portal-Based Smart Parking System.[4]	Abhirup Khanna	2016	Technology to develop a cloud-integrated smart parking system.	Remote Booking System and Improving Quality of Life
Intelligent Parking Space Detection System Based on Image Processing.[5]	R. Yusnita, Fariza Norbaya, Norazwinawati Basharuddin	2012	Image processing techniques to detect parking space availability	Developed in software and hardware platform.

Carnatic Parking Management System.[6]	M. M. Rashid, A. Musa, M. Ataur Rahman, N. Farahana, A. Farhana	2012	Carnatic parking system and electronic fee collection	Security in public parking lots,
Integrated Approach in the Design of Car Park Occupancy Information System (COINS).[7]	D.B.L. Bong, K.C. Ting, K.C. Lai	2007	Improving traffic flow within the parking areas.	COINS system utilizing an integrated approach of image processing
"Smart Parking" System Based on Optimal Resource Allocation and Reservations.[8]	Vishwanath Y	2009	smart parking system is designed for urban areas and aims to assign	Minimizes reservation conflicts and maintains cost-effective parking assignments.
Android Based Smart Car Parking System.[9]	B.D. Deore1, S. R. Kurkute2, Pooja Bhalerao3, Kajal Barve4, Mokshada Deore5	2016	The system introduces a miniature model of a car park that directs and manages the number of cars that can be parked based on space availability	Android-based smart car parking system effectively automates the reservation and management of parking spaces.

Table 2.2 Literature

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Required component

3.1.1 Arduino UNO

Arduino Uno Of the many Arduinos, it is the one that is most frequently used. It is newcomers' top preference. Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, A USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC to-DC adapter or battery to get started. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.

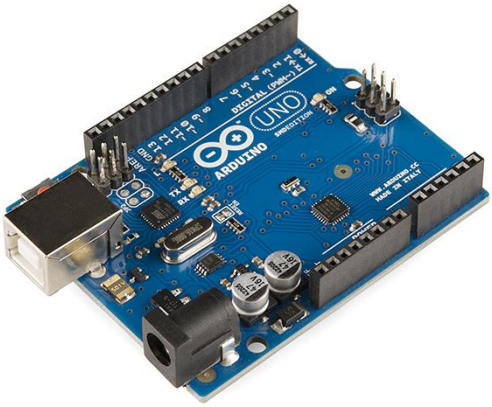

Figure 3.1 Arduino UNO

3.1.2 IR sensor

An IR sensor is a piece of technology that is designed to detect smells in the immediate area. An IR senso senses motion while simultaneously measuring an object's heat. These detectors are referred to as unresistant IR detectors since they do not emit any light; instead, they solely measure infrared radiation. In the infrared spectrum, most items emit some sort of heat radiation. These radiations can be detected by an infrared detector even if they are undetectable to human vision.

These sensors consist of two main parts:

IR Emitter:

Although IR sensors used in parking systems are often passive and do not emit their own light, some types may use an IR emitter to illuminate the area with infrared light.

IR Detector:

The IR detector senses the infrared radiation emitted by nearby objects. In smart parking applications, this is typically the heat radiated by vehicles.

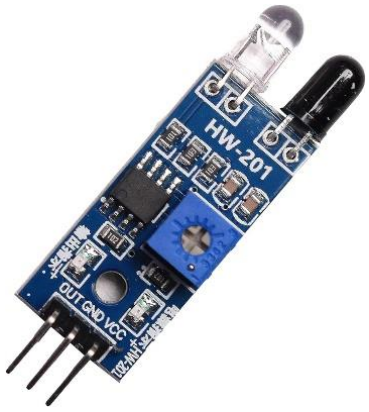

Figure 3.1 IR Sensor

3.1.3 Servo Motor

The servo motor is mostly employed in robotization technology and other high-tech artificial processes. It is a tone-contained electrical gadget that perfectly and effectively turns a machine's corridor. Additionally, this motor's affair shaft may be adjusted to a specific angle. Buses, airplanes, toys, home electronics, and a host of other applications heavily utilize servo motors. As a result, this blog describes a servo machine's description, types, medium, principle, operating, controlling, and beginning operations. A servo motor is a type of motor that enables complete control over acceleration, haste, and angular position.

Figure 3.1 Servo Motor

3.1.4 Power Supply Board

The Internet of Things has been around almost as long as the internet itself. Some of the key

components are small Arduino with Web Portal power supplies, usually in the 1-20W category, that provide power at the required voltage for the electronic system that handles sensor input and data transfer to the internet.

Figure 3.1 Power Supply Board

3.1.5 Jumper Wires

Jumper wires, both male-to-male and male-to-female, are essential for connecting components on the breadboard and to the Arduino board. Here's how you can use these Jumper wires in the setup of a smart parking system:

Types of Jumper Wires and Their Uses

Male-to-Male Jumper Wires: Used to connect points on the breadboard to each other. Connect components on the breadboard to the Arduino's pins.

Male-to-Female Jumper Wires: Used to connect the Arduino's pins to components that have male headers, such as sensors or modules.

Figure 3.1 Jumper Wires

3.2 Software Specification

3.2.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Writing a Sketch: Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. Ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right and corner of the window display the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

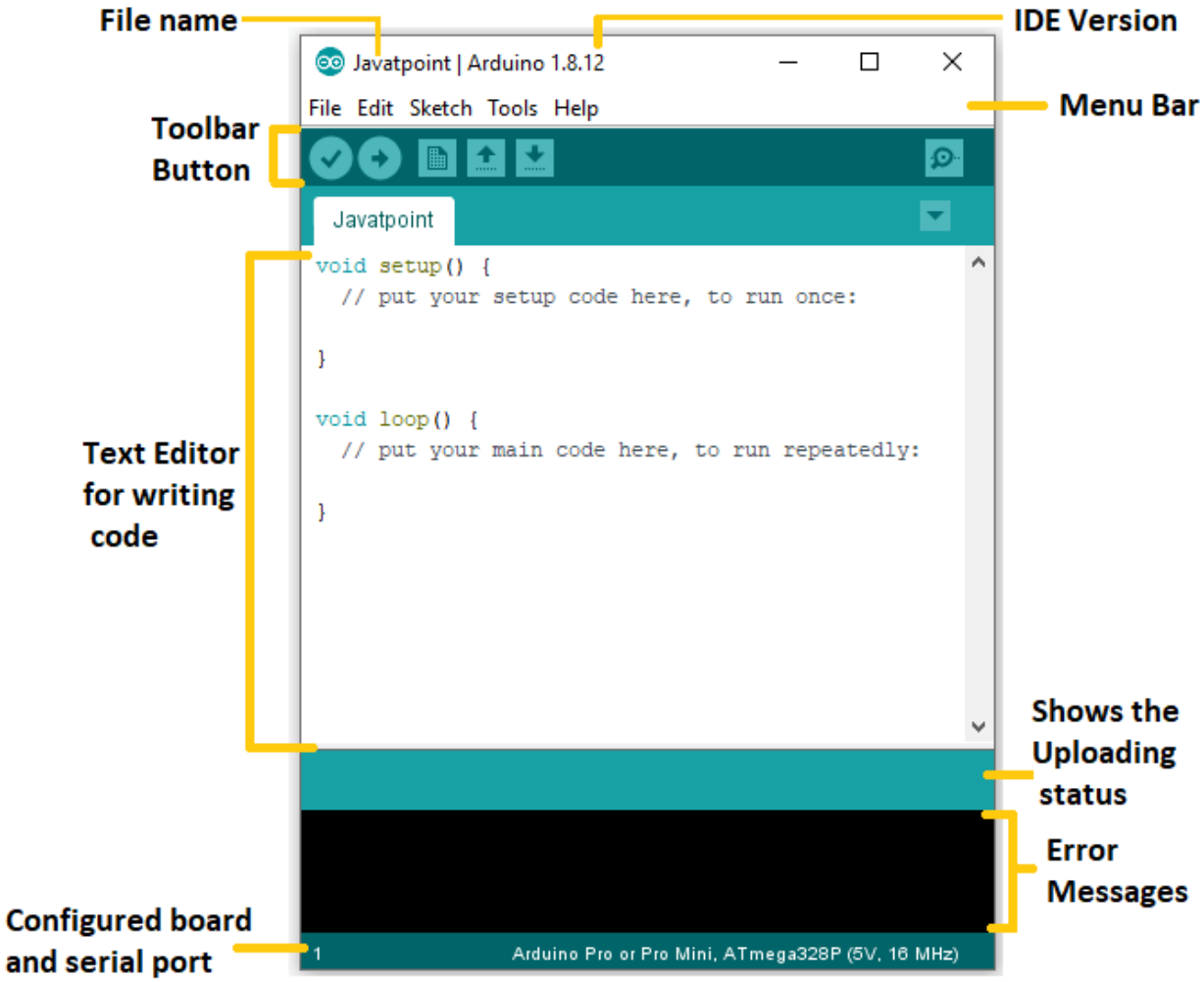

Figure 3.2 Arduino IDE

3.2.2 EMBEDDED C

Use of embedded processors in passenger cars, mobile phones, medical equipment, aerospace systems and defense systems is widespread, and even everyday domestic appliances such as dish washers, televisions, washing machines and video recorders now include at least one such device. Because most embedded projects have severe cost constraints, they tend to use low-cost processors like the 8051 family of devices considered in this book. As a result, developing embedded software presents significant new challenges, even for experienced desktop programmers. If you have some programming experience - in C, C++ or Java - then this book and its accompanying CD will help make your move to the embedded world as quick and painless as possible.

3.2.3 BACK-END TECHNOLOGIES

1.FLASK

Flask is a lightweight web framework for Python that prioritizes simplicity and flexibility. Unlike full-stack frameworks that come bundled with features like ORM (Object Relational Mapper), Flask takes a minimalist approach. Its core functionality focuses on providing essential tools and libraries necessary for building web applications without imposing constraints on developers. This microframework philosophy means Flask gives developers the freedom to choose and integrate various libraries and extensions as needed, rather than including them by default. For database interactions requiring an ORM, Flask integrates seamlessly with SQL Alchemy, a powerful Python SQL toolkit and Object Relational Mapper.

2.SQLITE3

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQ Lite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly. SQLite3 is a lightweight and self-contained relational database management system (RDBMS) that is widely used in applications where simplicity, portability, minimal setup are priorities. Unlike client-server database systems like MySQL, SQLite operates directly with the application, storing the entire database as a single file on the disk. This architecture makes SQLite highly suitable for embedded systems, mobile applications, and small to medium-scale web applications.

3.3 Data Flow Diagram

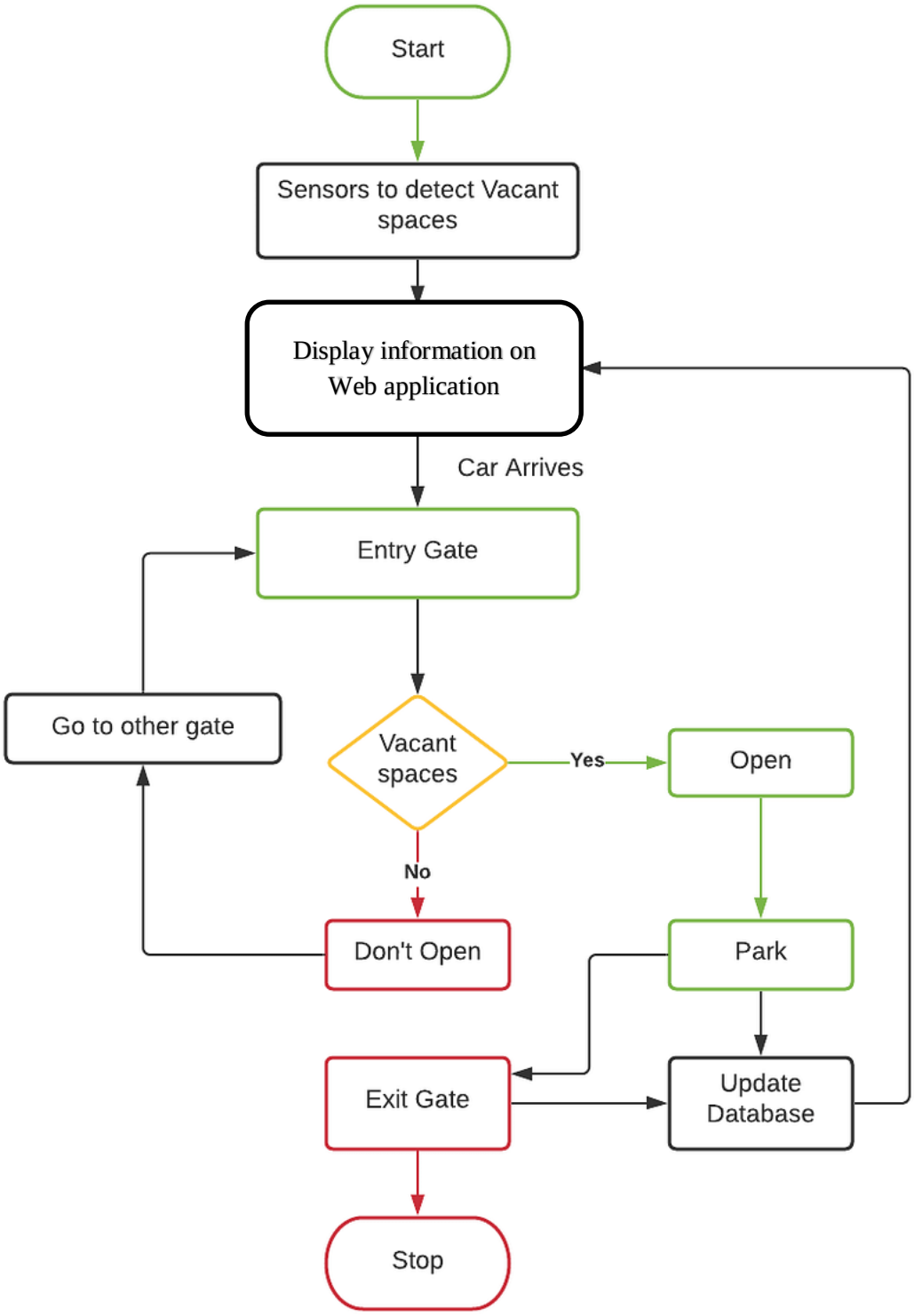

Figure 3.3 Flow Diagram

CHAPTER 4

METHODOLOGY

4.1 Tools and Technologies Used:

4.1.1Arduino Microcontrollers: Arduino boards are fundamental to our smart parking system, serving as the core hardware platform. These programmable microcontrollers enable seamless integration with a variety of sensors and actuators essential for monitoring and controlling parking slots. By leveraging Arduino's versatility and ease of programming, we ensure robust functionality and reliable operation of our system. The Arduino boards play a pivotal role in facilitating real-time data acquisition from sensors such as infrared (IR) sensors, thereby enabling accurate detection of parking slot availability. Their flexibility allows us to implement sophisticated control logic and efficient communication protocols, contributing to the overall effectiveness and efficiency of our smart parking solution.

4.1.2 Infrared (IR) Sensors: IR sensors are deployed within each parking slot to detect the presence of vehicles. These sensors emit infrared light and detect its reflection to accurately determine whether a parking slot is occupied or vacant. IR sensors are chosen for their reliability and effectiveness in various lighting conditions IR sensors can cover a wide detection range, making them suitable for various parking slot sizes and configurations.

4.1.3 Servo Motors: Servo motors play a crucial role in the smart parking system by controlling physical barriers at the entrance and exit of parking lots. These motors enable automated opening and closing of barriers based on real-time data from IR sensors or user reservations, ensuring efficient management of parking access, contributing to the overall effectiveness and efficiency of our smart parking solution.

4.1.4 Power Supply Board: A dedicated power supply board is employed to provide stable and reliable power to Arduino boards, sensors, and servo motors throughout the parking system. This board ensures uninterrupted operation and minimizes the risk of system failures due to power fluctuations Equipped with multiple output channels to power various components simultaneously, Allows for organized and efficient distribution of power.

4.1.5 Python Flask: Python Flask serves as the web framework for developing the local web application that interfaces with the smart parking system. Flask facilitates communication between the Arduino-based hardware components and the user interface. It enables real-time updates on parking slot availability to users and allows them to make reservations seamlessly through the web application.

4.2 Modules Description:

4.2.1 Arduino Microcontroller Integration:

Purpose: The Arduino microcontroller serves as the central hardware component of the smart parking system.

Functionality: It interfaces with various sensors, including IR sensors, to detect the presence or absence of vehicles in parking slots.

Implementation: Arduino collects real-time data from IR sensors and processes it to determine slot occupancy status.

4.2.2 Infrared (IR) Sensors:

Purpose: IR sensors are deployed within each parking slot to detect the presence of vehicles.

Functionality: They emit infrared light and measure the reflection to ascertain whether a slot.

Implementation: The accuracy and reliability of IR sensors ensure precise detection, crucial for real-time updates on parking slot availability.

4.2.3 Python Flask Web Application:

Purpose: The Python Flask framework is used to develop the local web application.

Functionality: It provides a user-friendly interface for monitoring parking slot availability.

Implementation: Flask processes data received from Arduino, presents it dynamically to users, and supports interactive features such as real-time updates and reservation functionalities.

4.2.4 Servo Motors:

Purpose: Servo motors are employed to control physical barriers at parking entrances/exits.

Functionality: They automate the process of slot reservation and release based on user interactions through the web application.

Implementation: Upon reservation confirmation, servo motors actuate barriers to allow entry, ensuring efficient management of parking slots.

4.2.5 Power Supply Board:

Purpose: The power supply board provides regulated power to Arduino, IR sensors, and servo.

Functionality: It ensures continuous and reliable operation of all system components.

Implementation: Proper power management prevents system failures and ensures consistent performance of the smart parking system, allows for organized and efficient distribution of power.

Backend Processing: Executes business logic related to reservation management, authentication, and data validation, Allows users to reserve a parking slot through the web interface, with immediate updates to reflect the reservation status, Handles user requests and interactions through Flask routes.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Code Snippets

Serial test.py code

```
import serial
import time

data = serial.Serial(
        'COM4',
        baudrate = 9600,
        parity=serial.PARITY_NONE,
        stopbits=serial.STOPBITS_ONE,
        bytesize=serial.EIGHTBITS,
        timeout=1
        )

def read_data():
    print('reading.....')
    Data = []
    while True:
        data.write(str.encode('O'))
        time.sleep(1)
        d = data.readline()
        d = d.decode('UTF-8', 'ignore')
        if d:
            d = d.split(',')
            print(d)
            if len(d) == 3:
                Data.append(int(d[0]))
                Data.append(int(d[1]))
                Data.append(int(d[2].replace('\r\n', '')))
                break
    return Data
```

app.py code

```
from flask import Flask, render_template, request, jsonify, session
import sqlite3
import secrets
connection = sqlite3.connect('user_data.db')
cursor = connection.cursor()
command = """CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT,
mobile TEXT, email TEXT, address TEXT)"""
cursor.execute(command)
app = Flask(__name__)
app.secret_key = secrets.token_hex(16)
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/home')
def home():
    return render_template('home.html')
@app.route('/signin', methods=['GET', 'POST'])
def signin():
    if request.method == 'POST':
        connection = sqlite3.connect('user_data.db')
        cursor = connection.cursor()
        name = request.form['name']
        password = request.form['password']
       query = "SELECT * FROM user WHERE name = '"+name+"' AND password=
'"+password+"'"
        cursor.execute(query)
        result = cursor.fetchone()
        if result:
            session['name'] = result[0]
            return render_template('home.html')
        else:
            return render_template('signin.html', msg='Sorry, Incorrect Credentials Provided,  Try
Again')
```

```
    return render_template('signin.html')
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        connection = sqlite3.connect('user_data.db')
        cursor = connection.cursor()

        name = request.form['name']
        password = request.form['password']
        mobile = request.form['phone']
        email = request.form['email']
        address = request.form['address']

        print(name, mobile, email, password)
        cursor.execute("INSERT INTO user VALUES ('"+name+"', '"+password+"',
'"+mobile+"', '"+email+"', '"+address+"')")
        connection.commit()

        return render_template('signin.html', msg='Successfully Registered')
    return render_template('signup.html')
@app.route('/get_data')
def get_data():
   from serial_test import read_data
   Data1 = read_data()
   print("recieved data {}".format(Data1))
   return jsonify(Data1)
@app.route('/signout')
def signout():
    return render_template('index.html')
if __name__ == "__main__":
    app.run(debug=True)
```

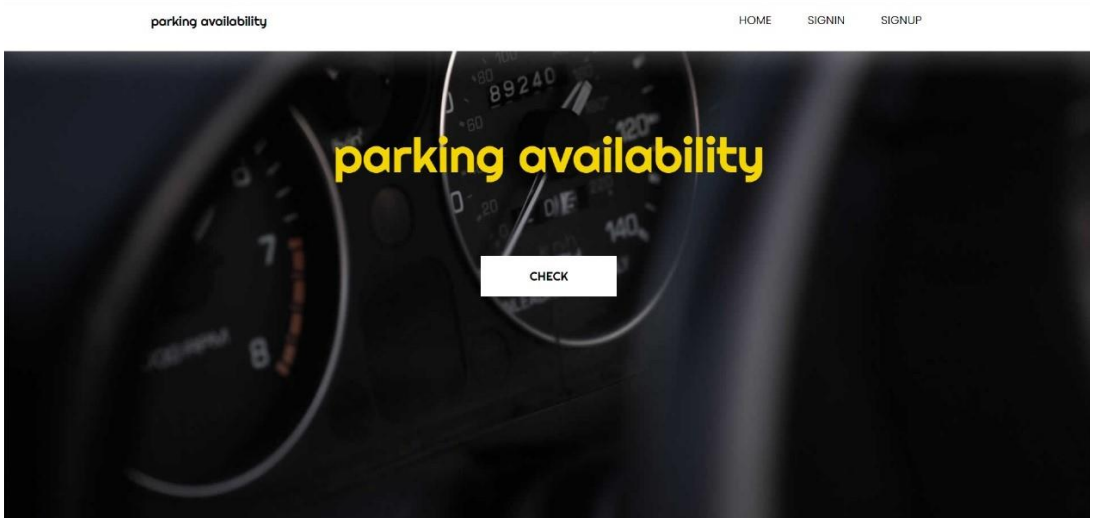

Figure 5.1 Home Page

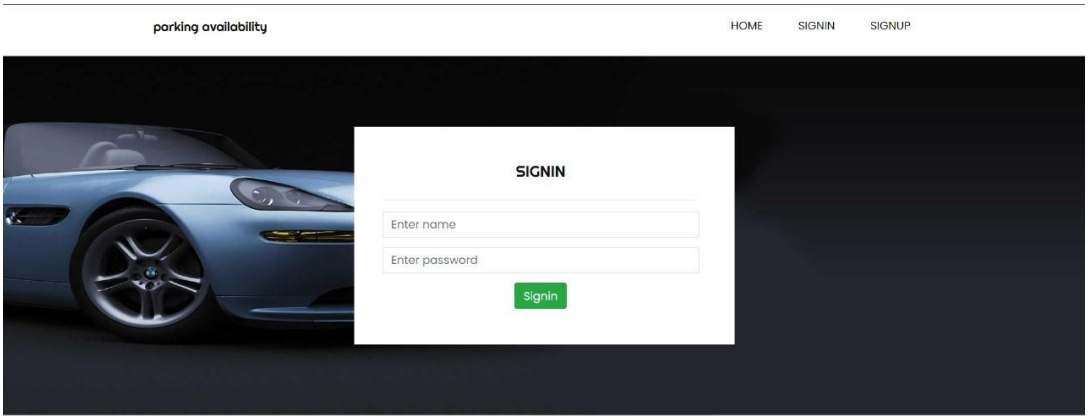

Figure 5.1 Sign In

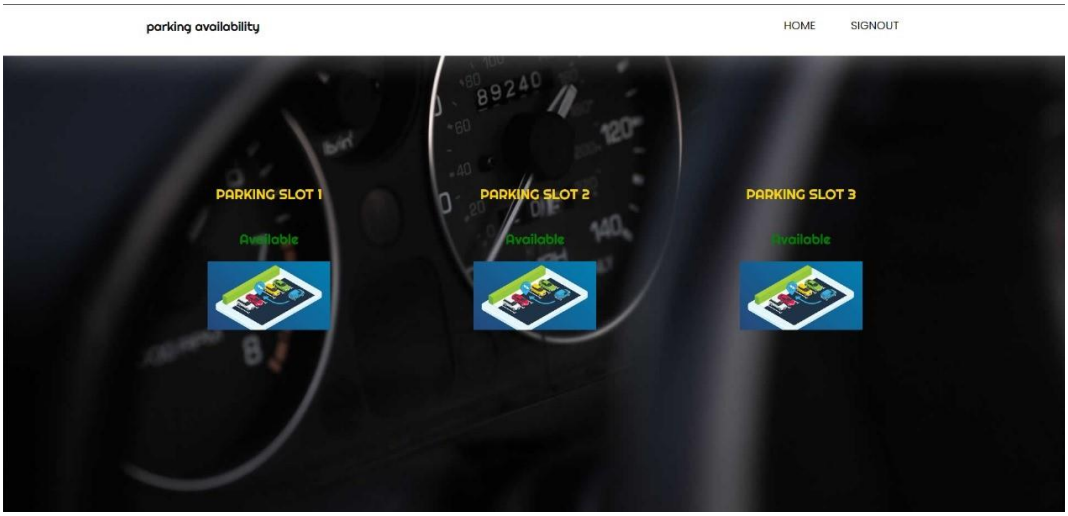

Figure 5.1 Parking Slot

5.2 Circuit Diagram

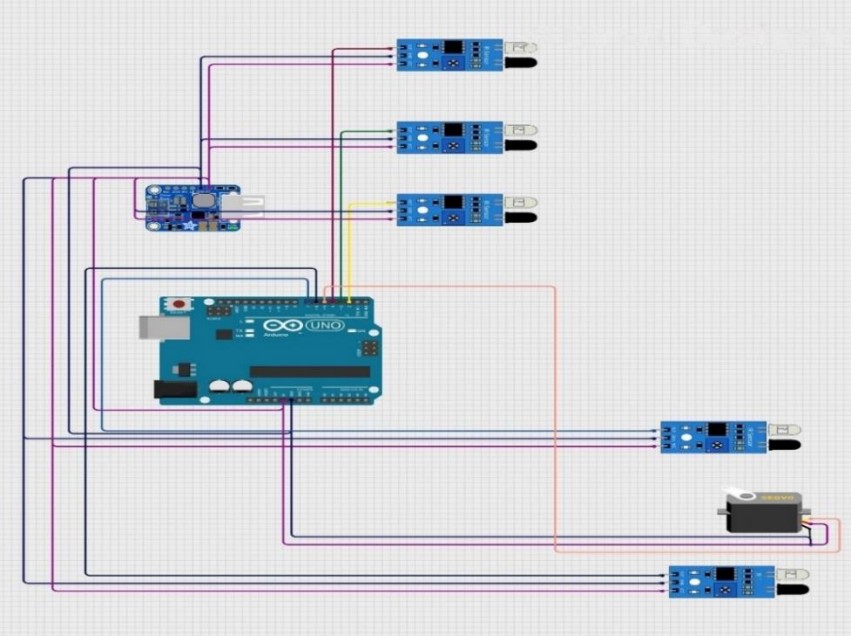

Figure 5.2 Circuit Diagram

5.3 Test Case

5.3.1 Sensor Functionality:

Unit Test Case 1: IR Sensor Detection

Objective: Verify that the IR sensor accurately detects vehicle presence.

Setup: Place a vehicle within the detection range of the IR sensor.

Steps: Observe the sensor output.

Expected Result: The IR sensor detects the vehicle and triggers the appropriate response.

Test case	Unit Test Case 1
Name	Vehicle detection
Item	IR Sensor
Input	Blocking IR unit
Expected output	Detection of Vehicle
Actual output	Successfully vehicle is detected
Remark	Pass

Table 5.3 Unit test case 1

5.3.2 User Interface:

Test Case 5: web application Integration

Objective: Test the integration of the smart parking system with a web app.

Setup: Use the web application to reserve a parking space.

Steps: Check if the reservation reflects accurately on the app.

Expected Result: The app displays the correct reservation status and updates in real-time.

Test Case	Unit Test Case 2
Name	web application
Item	Power Supply Board To web app
Input	Send data from Power supply
Expected output	Data from power supply to web app
Actual output	Successfully communication achieved
Remark	Pass

Table 5.3 Unit test case 2

5.3.3 System Integration:

Test Case 7: Arduino Uno and power supply board Communication

Objective: Verify communication between Arduino Uno and Power Supply Board.

Setup: Connect Arduino Uno and Power Supply Board are secure and correctly configured.

Steps: Initiate communication signals/messages between Arduino Uno and Power Supply Board. Send test data or signals that simulate typical operational conditions.

Expected Result: Arduino Uno and Power Supply successfully communicate without errors.

Test Case	Unit Test case 3
Name	Updation
Item	Arduino Uno
Input	Network Connectivity
Expected output	Update to user
Actual output	Successfully updated
Remark	Pass

Table 5.3 Unit test case 3

5.4 Result

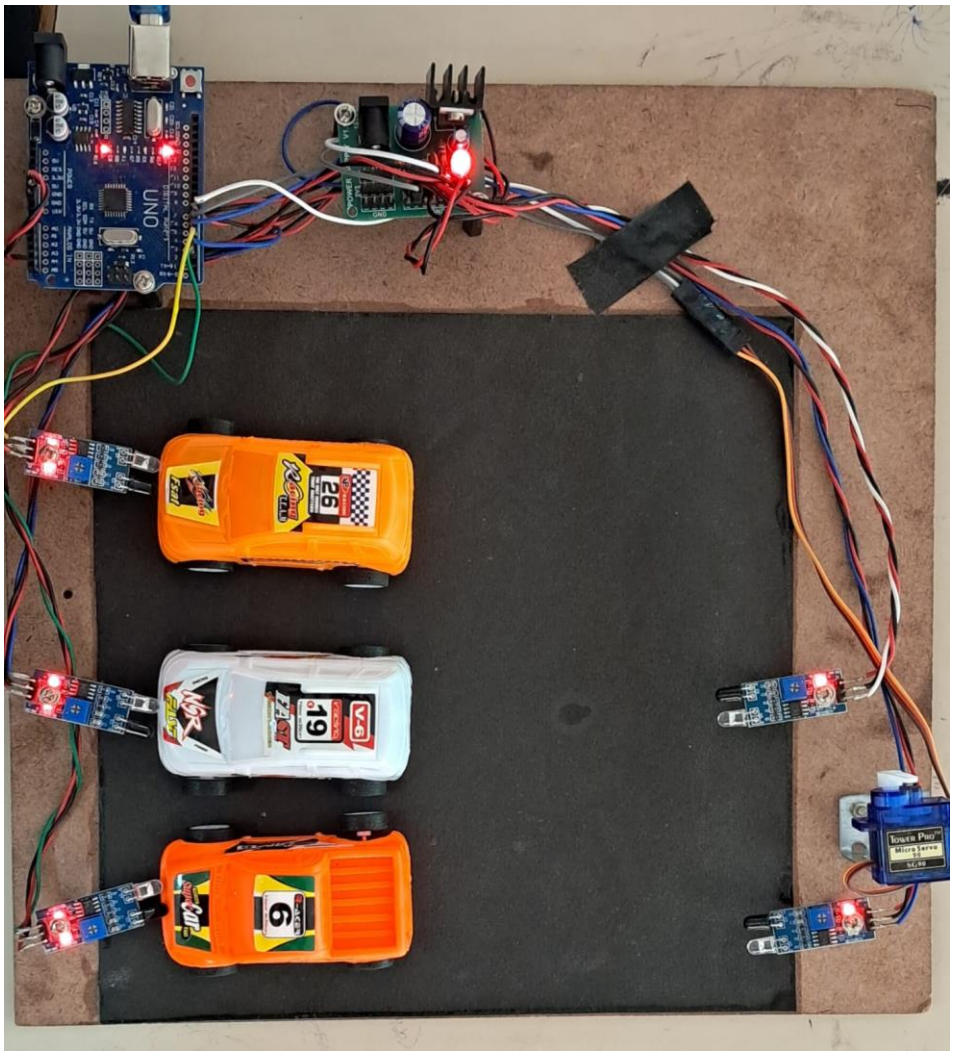

Figure 5.4 Result

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Analysis of the results

The implementation of the smart parking system using Arduino, IR sensors, servo motors, and a power supply board has shown promising results in addressing urban parking challenges. By integrating Arduino microcontrollers with IR sensors, the system accurately detects vehicle presence in parking slots and transmits this data to a Python Flask-based web application via serial communication. The web application successfully provides real-time updates on parking slot availability, enabling users to monitor and reserve slots seamlessly. The use of servo motors for automated slot reservation further enhances user convenience and operational efficiency.

Accuracy: IR sensors reliably detect occupancy status, minimizing errors in slot availability information.

Real-time Updates: The Flask web application delivers timely updates, reducing search times and congestion by guiding users to available parking slots efficiently.

User Interaction: The interface allows intuitive reservation processes, enhancing user convenience and satisfaction.

6.2 Discussion on findings

The development of the smart parking system using Arduino, IR sensors, servo motors, and a power supply board has provided significant insights into addressing urban parking challenges. By integrating these technologies, the project successfully demonstrated real-time detection of parking slot occupancy and automated reservation capabilities. Findings indicate that accurate detection of vehicle presence through IR sensors coupled with Arduino microcontrollers is reliable and responsive. The implementation of Python Flask for the web application facilitated seamless communication of parking availability data to users, enhancing user convenience and reducing search times. Moreover, the use of servo motors for automated barrier control streamlined the reservation process, contributing to efficient utilization of parking spaces. Overall, the project underscores the effectiveness of combining hardware and software solutions in modernizing parking management, paving the way for more efficient urban traffic flow and improved user experiences in congested areas. Future iterations could explore scalability, integration with larger networks, and additional features to further optimize urban parking operations.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The development of the smart parking system using Arduino, infrared sensors, servo motors, and a power supply board represents a significant advancement in addressing the complexities of urban parking management. Traditional parking systems often struggle to provide accurate and real-time information on slot availability, leading to inefficiencies such as congestion and prolonged search times. By integrating hardware components with a Python Flask-based web application, this project has successfully demonstrated a solution that not only detects vehicle presence accurately but also provides users with immediate updates and seamless reservation capabilities.

The key achievements of this project include:

- Accurate Detection: Utilizing infrared sensors interfaced with Arduino, the system reliably detects vehicle presence in parking slots.
- Real-Time Updates: Through serial communication, data on parking slot occupancy is transmitted to a local web application developed using Python Flask, ensuring users receive timely updates on available slots.
- Enhanced User Experience: The web application offers an intuitive interface for monitoring, reserving, and managing parking slots in real time, thereby reducing search times and congestion.
- Automated Control: Integration of servo motors enables automated physical barriers for efficient slot reservation and release based on user interactions.

7.2 Summary

The smart parking system project addresses the pressing challenges of urban parking management by integrating Arduino microcontrollers, infrared (IR) sensors, servo motors, and a power supply board. As urbanization increases vehicle numbers and exacerbates parking difficulties, traditional systems often struggle to provide accurate slot availability information, leading to congestion and frustration. In response, our solution employs Arduino boards interfaced with IR sensors to detect vehicle presence in parking slots reliably. This real-time data is transmitted via serial communication to a local web application built with Python Flask, allowing users to access up-to-date parking slot availability and enabling seamless reservation and monitoring. Additionally, servo motors control physical barriers, automating slot

reservation processes. By leveraging these technologies, our system optimizes parking space utilization, enhances user convenience, and contributes to efficient urban traffic management. This project showcases a comprehensive approach to modernizing parking systems through innovative hardware and software integration, ensuring effective management of urban parking challenges.

7.3 Future Enhancements:

Looking ahead, several avenues for enhancing the smart parking system could be explored:

- Scalability: Extend the system to manage larger parking lots or multiple parking locations within a city.
- Mobile Integration: Develop mobile applications for iOS and Android platforms to provide users with on-the-go access to parking availability and reservations.
- Predictive Analytics: Implement machine learning algorithms to predict parking demand based on historical data, weather conditions, or events.
- Arduino with Web Portal Integration: Explore Internet of Things (Arduino with Web Portal) technologies to enhance sensor connectivity and data processing capabilities.
- Green Parking Initiatives: Incorporate features to promote eco-friendly practices such as electric vehicle (EV) charging station management and incentivizing use of sustainable transport options.
- Integration with Smart City Infrastructure: Collaborate with urban planners to integrate parking data into broader smart city initiatives, such as traffic flow optimization and urban mobility management.

BIBLIOGRAPHY

1. Carnatic Smart Parking System Using Internet of Things (ARDUINO WITH WEB PORTAL) https://www.ijsrp.org/research-paper-1215.php?rp=P484954
2. Smart Parking System Based on Internet of Things https://www.researchgate.net/publication/303842610_Arduino with Web Portal_based_Smart_Parking_System
3. Carmate Car Parking System Commanded by Android Application ++https://fardapaper.ir/mohavaha/uploads/2018/02/Fardapapr-Automated-car-parking-system-commanded-by-Android-application.pdf
4. ARDUINO WITH WEB PORTAL Based Smart Parking System https://www.researchgate.net/profile/Abhirup-Khanna/publication/303842610_Arduino with Web Portal_based_Smart_Parking_System/links/59fc5dc3458515d070644fb2/Arduino with Web Portal-based-Smart-Parking-System.pdf
5. Intelligent Parking Space Detection System Based on Image Processing https://www.ijimt.org/papers/228-G0038.pdf
6. Carnatic Parking Management System https://ijmlc.org/papers/95-A009.pdf
7. Integrated Approach in the Design of Car Park Occupancy Information System (COINS) https://www.iaeng.org/IJCS/issues_v35/issue_1/IJCS_35_1_02.pdf
8. Smart Parking" System Based on Optimal Resource Allocation and Reservations https://3ciencias.com/wp-content/uploads/2021/11/Parte-2_3C-Tecnologi%CC%81a_Special_Issue_39-2_nov.pdf
9. Android Based Smart Car Parking System https://www.ijarcce.com/upload/2018/march-18/IJARCCE%2032.pdf
10. Android Based Smart Car Parking System https://www.ijarcce.com/upload/2018/march-18/IJARCCE%2032.pdf