GPS BASED TOLL FEE PAYMENT SYSTEM

**Along with route optimization**

# A PROJECT REPORT

***submitted by***

**JAGADESHWARAN R P (230701120)**

**JANARTHANAN B (230701125)**

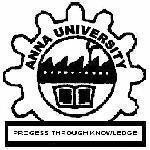
**JAGADEESAN I (230701119)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

# COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE,**

**ANNA UNIVERSITY: CHENNAI 600 025**

# MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**GPS BASED TOLL FEE PAYMENT SYSTEM”** is the bonafide work of “**JAGADESHWARAN R P (230701120), JAGADEESAN I (230701119), JANARTHANAN B (230701125)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

# SIGNATURE

Ms. S. Ponmani M.E.,MBA,

# SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on

**Internal Examiner External Examiner**

# ABSTRACT

# Manual toll collection and infrastructure-heavy RFID systems continue to pose challenges in modern transportation, contributing to traffic congestion, user inconvenience, and operational inefficiencies. To address these issues, this study proposes the development of a smartphone-based toll collection system leveraging GPS, geofencing, and Internet of Things (IoT) technologies. The methodology follows the Design Thinking framework, encompassing literature review, user requirement analysis, ideation, prototype development, testing, and iterative refinement. Core components of the system include a mobile application for user tracking and interaction, a Flask-based backend for toll validation and trip logging, and an embedded hardware module (ESP8266 with servo motor) for toll barrier control. The proposed solution enables real-time, contactless tolling through geofence-triggered events and automated backend communication, minimizing the need for physical toll booths or RFID tags. Initial evaluations demonstrate the system’s effectiveness in reducing toll queue delays and enhancing user transparency. Future work will explore improvements in GPS precision, integration with national toll networks, OCR-based enforcement for unpaid vehicles, and scalable deployment across diverse transportation infrastructures.

# ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

|  |  |  |
| --- | --- | --- |
| **CHAPTER No.** | **TITLE**  **TABLE OF CONTENTS** | **PAGE No.** |
|  | **ABSTRACT** | **iii** |
|  | **INTRODUCTION** | **1** |
|  | 1.1 Motivation | **2** |
|  | 1.2 Objectives | **2** |
|  | **LITERATURE REVIEW** | **3** |
|  | 2.1 Existing System | **4** |
|  | 2.1.1 Advantages of the existing system | **4** |
|  | 2.1.2 Drawbacks of the existing system | **4** |
|  | 2.2 Proposed system | **5** |
|  | 2.2.1 Advantages of the proposed system | **5** |
| **3.** | **SYSTEM DESIGN** |  |
|  | 3.1 Development Environment | **6** |
|  | 3.1.1 Hardware Requirements | **6** |
|  | 3.1.2 Software Requirements | **7** |
| **4.** | **PROJECT DESCRIPTION** | **8** |
|  | 4.1 System Architecture | **8** |
|  | 4.2 Methodologies | **9** |
| **5.** | **RESULTS AND DISCUSSION** | **10** |
| **6.** | **CONCLUSION AND FUTURE WORK** | **11** |
|  | 6.1 Conclusion | **11** |
|  | 6.2 Future Work | **11** |
|  | **APPENDIX** | **12** |
|  | **REFERENCES** | **15** |

**CHAPTER 1**

**INTRODUCTION**

Toll congestion is a critical issue affecting transportation systems, particularly on highways and expressways with high vehicle density. The conventional methods of toll collection, including manual payment and RFID-based systems, often result in prolonged waiting times, unnecessary fuel consumption, and increased traffic bottlenecks at toll plazas. These inefficiencies not only hinder the overall travel experience but also lead to considerable economic and environmental impacts. As digital mobility and urban transport demands continue to rise, the need for a more intelligent, seamless, and infrastructure-light tolling solution becomes increasingly vital.

To address these challenges, there has been growing interest in developing innovative toll management systems that utilize modern technologies such as Global Positioning System (GPS), geofencing, and the Internet of Things (IoT). These technologies enable real-time location tracking, automated toll validation, and remote system control without requiring physical toll booths or extensive roadside infrastructure. By integrating geolocation with mobile-based platforms, such systems aim to facilitate dynamic, contactless toll payment experiences that are efficient, transparent, and user-friendly.

This study focuses on the design and implementation of a GPS-based toll collection system that leverages smartphones, geofencing, and IoT-enabled devices to automate toll detection and barrier control. The system comprises a mobile application for vehicle tracking and user interaction, a Flask-based backend for real-time toll validation, and an ESP8266 microcontroller connected to a servo motor to physically operate toll barriers. Developed using the Design Thinking methodology, the project follows a structured workflow encompassing literature review, user research, system design, prototype development, and testing. The proposed solution aims to reduce congestion at toll gates, enhance user convenience, and serve as a foundation for next-generation intelligent transportation infrastructure.

**1.1 Motivation**

**Eliminating Toll Booth Congestion:** Toll booth congestion continues to be a major issue on highways and expressways, resulting in long wait times, unnecessary idling, increased fuel consumption, and travel delays. The primary motivation for this project is to address these inefficiencies by developing a GPS-based toll collection system that eliminates the need for physical toll booths and manual intervention, enabling vehicles to pass seamlessly without stopping.

**Enhancing Commuter Experience and Transportation Efficiency:** By streamlining toll operations and reducing travel interruptions, the project aims to enhance the overall commuting experience and improve the efficiency of transportation networks. A more efficient tolling system contributes to better traffic flow, reduced carbon emissions, and increased roadway capacity—ultimately supporting economic growth and commuter satisfaction.

**Leveraging IoT and Geofencing Technologies:** The project harnesses the power of Internet of Things (IoT) technologies, GPS tracking, and geofencing to build a smart toll management system. By using smartphones as active GPS nodes and integrating them with a cloud-based backend, the system facilitates real-time toll detection, automated payment validation, and responsive toll barrier operation. This innovative approach showcases the potential of IoT in transforming traditional infrastructure into intelligent, connected systems.

**1.2 Objectives**

**Develop a GPS-Based Virtual Toll Collection System:**  
The primary objective of this project is to design and implement a smartphone-driven, GPS-enabled toll collection system that automates toll payments without requiring physical toll booths or RFID infrastructure. The system aims to reduce congestion, enhance tolling efficiency, and provide a seamless experience for commuters by leveraging geofencing and IoT technologies.

**Integration of Mobile Sensing and Geofencing:**  
Utilize the in-built GPS modules in smartphones to detect vehicle entry and exit within geofenced toll zones. These virtual boundaries will replace physical detection mechanisms, allowing tolls to be triggered in real-time as vehicles cross designated checkpoints. The mobile application will also serve as the central interface for user interaction, trip planning, and payment validation.

**Automated Backend Communication and Barrier Control:**  
Establish a robust backend system using a Flask server to receive geolocation data from the app, match it with registered trips, and trigger toll deductions. Upon validation, the backend will communicate with an ESP8266 microcontroller over WebSocket to control a servo motor that operates the toll barrier automatically, based on pre-payment status.

**Transparent Toll Logging and User Feedback:**  
Implement a real-time trip and toll logging mechanism that records user journey details and toll deductions in a local database. This will provide transparency, enable refund processing for missed tolls, and allow users to track travel expenses. Notifications and alerts will enhance system usability and ensure users remain informed throughout their journey.

# CHAPTER 2

# LITERATURE REVIEW

**[1] Development of a GPS-based highway toll collection system.**

It outlines a framework for an intelligent traffic management system that leverages cloud computing and IoT technology. The proposed architecture includes components for information monitoring, calculation, intelligent modeling, and knowledge matching. By utilizing cloud computing for mass calculation, the system aims to achieve intelligent monitoring and management of urban traffic, ultimately improving traffic flow efficiency.

**[2] Design of electronic toll collection system based on global positioning system technique**

The proposed approach leverages smart cameras at intersections equipped with image understanding capabilities for real-time traffic monitoring and assessment. These cameras not only analyze traffic flow but also detect and track special vehicles, such as emergency vehicles, to prioritize their passage. Additionally, the system can identify traffic violations and collect traffic statistics.

**[3] Development of a GPS-based highway toll collection system**

The paper introduces a systematic approach for monitoring road traffic congestion, aiming to enhance safety and traffic management. It proposes an improved observer that combines the benefits of a piecewise switched linear traffic (PWSL) modeling approach and a Kalman filter (KF). This observer, termed PWSL-KF, functions as a virtual sensor to simulate traffic evolution in free-flow conditions.

Top of Form

# 2.1 Existing System

The existing toll collection systems primarily rely on manual toll booths or semi-automated RFID-based methods for processing vehicle payments. These systems require vehicles to slow down or stop at toll plazas, leading to traffic bottlenecks, especially during peak hours or holiday seasons. Manual systems also involve human operators, which increases operational costs and introduces the possibility of human error or revenue leakage.

Even in partially automated systems, the lack of integration with centralized databases and real-time vehicle tracking means that enforcement is weak, and fraudulent activities (like tag swapping or toll evasion) are harder to detect. Additionally, current systems often do not dynamically adjust toll pricing based on congestion levels or time-of-day, missing out on opportunities to optimize road usage and reduce traffic loads.

Moreover, most existing systems have limited interoperability, making it inconvenient for inter-state or inter-city travel, especially when different regions use incompatible tolling technologies.

# Advantages of the existing system

* **Simplicity**: Manual and semi-automated toll systems are straightforward to operate and understand, requiring minimal technological infrastructure.
* **Proven Infrastructure**: Many of these systems are already deployed and functional, with established protocols and workforce.
* **Low Initial Tech Investment**: Compared to fully automated smart toll systems, traditional setups involve lower upfront costs.

# 2.1.2 Drawbacks of the existing system

* **Traffic Congestion**: Vehicles must stop or slow down at toll booths, causing delays and bottlenecks.
* **Limited Automation**: Minimal real-time data processing or analytics capability; not adaptable to dynamic traffic situations.
* **High Operational Costs**: Requires human operators and maintenance of physical infrastructure.
* **Inadequate Fraud Detection**: Weak verification and tracking mechanisms make it easier to evade tolls or manipulate data.
* **Lack of Interoperability**: Inconsistent systems across regions lead to inefficiencies for long-distance travelers.
* **No Dynamic Pricing**: Toll charges remain fixed regardless of traffic volume or time, missing opportunities to manage road usage more effectively.

# 2.2 Proposed System

# Our proposed SwiftToll system is a next-generation, fully automated toll collection solution that leverages IoT technology, license plate recognition (LPR), and cloud-based data management to streamline toll processing. Unlike conventional toll booths that rely on manual or semi-automated operations, SwiftToll enables seamless toll collection without requiring vehicles to stop or slow down.

# Using high-speed cameras and intelligent vehicle detection, the system captures license plate data in real time and cross-references it with a centralized database for automatic billing. This drastically reduces traffic congestion at toll points and enhances road safety by maintaining uninterrupted traffic flow. The platform is capable of dynamic toll pricing based on factors like traffic density, time of day, and vehicle class, promoting smarter traffic distribution.

# Furthermore, SwiftToll incorporates data analytics and real-time monitoring, providing authorities with valuable insights into traffic trends, toll revenue, and system performance. It’s designed with scalability and interoperability in mind, making it adaptable for city, state, and even national-level deployment.

# 2.2.1 Advantages of the proposed system

* **Seamless Toll Collection**: Vehicles are charged automatically without stopping, reducing delays and congestion at toll points.
* **Real-Time Monitoring**: Enables accurate tracking and data logging of vehicle entries and exits, improving accountability and enforcement.
* **Dynamic Pricing Support**: Can implement congestion-based or time-of-day toll rates to manage traffic loads more effectively.
* **Reduced Operational Costs**: Minimizes the need for human operators, leading to long-term cost savings.
* **Scalability & Interoperability**: Can be easily scaled and integrated across different regions and toll networks.

# CHAPTER 3

**SYSTEM DESIGN**

* 1. **Development Environment**

**3.1.1 Hardware Requirements**

1. ESP8266 WiFi Module
2. Bread Board
3. IR sensors
4. Jumper wires
5. Servo SG90 stepper motor

**ESP8266 WIFI MODULE**

The ESP8266 is the core of the hardware architecture. It functions as the microcontroller and provides built-in Wi-Fi capabilities, allowing it to send and receive data over a network without needing an external module. This feature enables real-time communication between the toll booth hardware and the backend server, making it ideal for IoT-based smart toll systems. It processes data from sensors, controls actuators like the servo motor, and ensures timely responses to user actions or system triggers.

**Arduino UNO**

The servo motor is used to automate the physical movement of the toll gate barrier. When a vehicle is successfully verified and toll payment is confirmed, the motor receives a signal from the ESP8266 to rotate, lifting the barrier. Once the vehicle passes, it automatically lowers the gate. Its precision, reliability, and ease of control make it a perfect fit for this type of application.

**Breadboard**

A breadboard is used as a temporary circuit board to assemble and test the hardware components without soldering. It allows for easy changes, additions, and debugging during the prototyping phase. This flexibility helps in experimenting with different circuit designs and refining the hardware logic.

**IR Sensor**

Infrared (IR) sensors play a crucial role in vehicle detection. Positioned near the toll gate, these sensors detect the presence of a vehicle by identifying the change in infrared light caused by the object. Once a vehicle is detected, the system is triggered to begin toll verification and gate control processes. These sensors are essential for initiating real-time responses without manual input.

**Jumper wires**

Jumper wires are used to create electrical connections between the ESP8266, sensors, servo motor, LEDs, and the breadboard. They are essential for ensuring seamless communication and power flow between components. Male-to-male, female-to-female, and male-to-female jumper wires are used depending on the pin types of the modules.

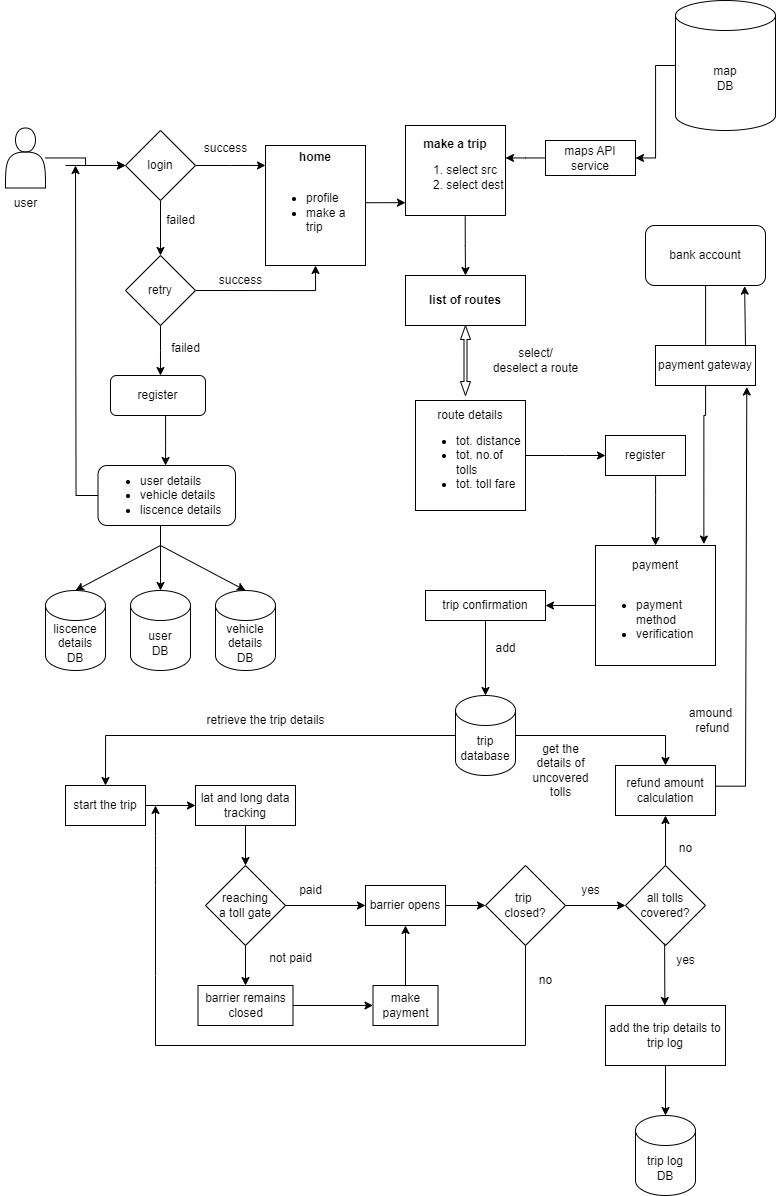
**3.1.1 Software Requirements**

* + - 1. Arduino IDE
      2. Flask (python)
      3. React Native
      4. PostgreSQL

# CHAPTER 4

# PROJECT DESCRIPTION

**4.1 SYSTEM ARCHITECTURE**

****

**Fig 4.1 System Architecture**

**4.2 METHODOLOGY**

**Problem Definition:**  
The objective of the SwiftToll project is to develop an IoT-based toll collection system that automates toll payments, detects vehicles, and manages access control. This system aims to enhance the overall toll booth experience by eliminating manual processes, reducing wait times, and improving traffic flow at toll gates. The system will use real-time vehicle detection, Wi-Fi communication, and secure payment verification to provide seamless access at toll points.

**Literature Review:**  
A comprehensive review of existing toll collection systems, such as RFID-based and ANPR-based technologies, was carried out to identify current limitations in manual toll collection, including traffic congestion, long wait times, and errors in toll calculation. The review also focused on IoT integration for vehicle detection and smart payment systems. Insights were drawn from various traffic management, payment gateway, and IoT security protocols to inform the design of SwiftToll.

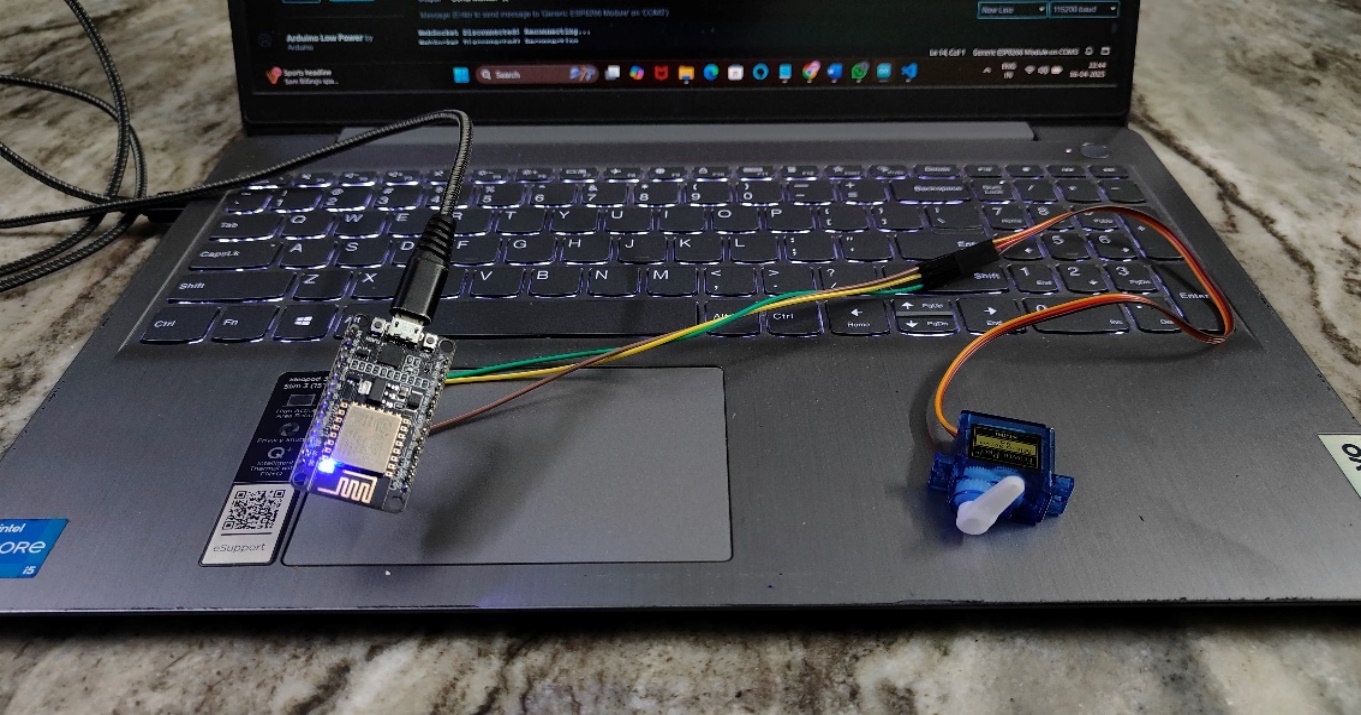
**Requirements Analysis:**  
The system’s functional requirements include automatic vehicle detection, toll payment verification, access control (via servo motor), real-time communication with a mobile app, and secure transaction processing. Non-functional requirements considered include scalability (for large toll networks), high system reliability, minimal latency in decision-making (for gate operations), and strong security measures to protect user data and financial transactions. Stakeholder feedback was integrated into the system design to ensure it meets user and regulatory needs.

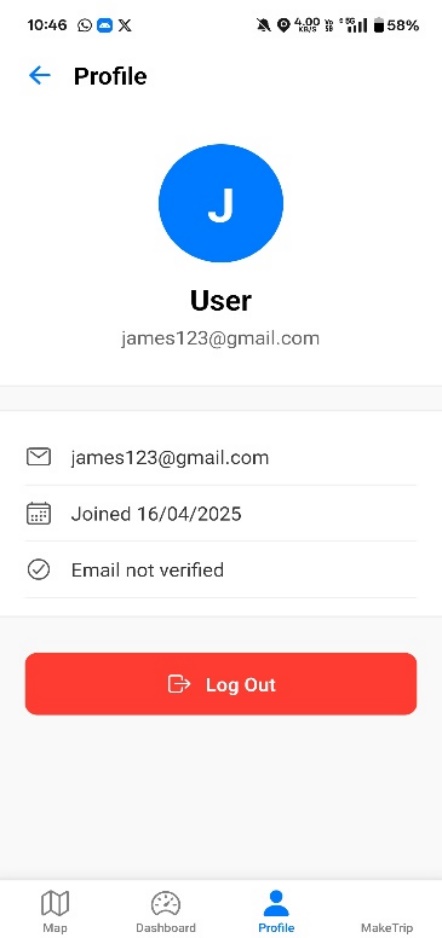
**System Design:**  
Based on the requirements analysis, the system architecture was designed to incorporate IoT components such as the ESP8266 microcontroller for communication, IR sensors for vehicle detection, a servo motor for barrier control, and LEDs for visual user feedback. The backend was developed using Python, integrated with PostgreSQL for managing user and toll data, while React Native was chosen for building a mobile app that provides a user-friendly interface for payment tracking and toll status updates. Communication protocols like HTTP over Wi-Fi were established to ensure seamless data transfer between the mobile app, backend, and hardware components.

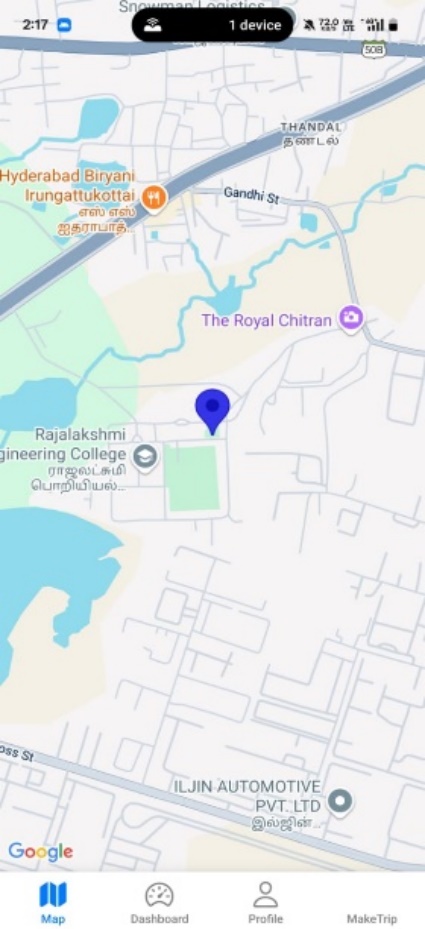
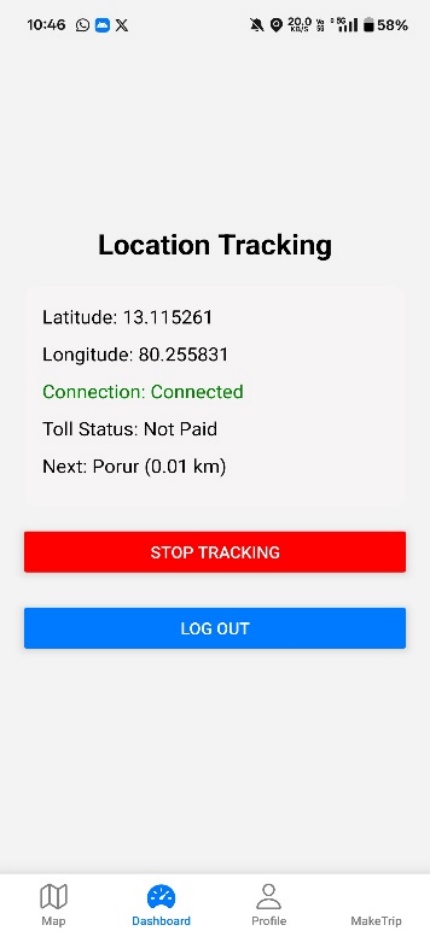
**Prototype Development:**  
A prototype of the SwiftToll system was created to integrate all components, including the ESP8266, IR sensors, servo motor, and LED indicators. The mobile app was developed in React Native to allow users to check toll payment status, view toll history, and receive notifications about their toll trips. The backend, written in Python, handles vehicle verification, toll calculation, and the command-response system for opening and closing toll gates. The prototype was tested in a controlled environment to verify all components function as expected.

**Evaluation and Testing:**  
The prototype system was evaluated based on performance metrics such as vehicle detection accuracy, response time for toll verification, reliability of the servo motor control, and system security. Testing included simulation in real-world traffic scenarios and feedback collection from users who interacted with the mobile app. The results confirmed the system's effectiveness in automating toll collection and providing real-time updates, while also identifying potential areas for optimization, such as scaling the backend to support more toll booths and improving the app’s user interface.

**CHAPTER 5**

 **RESULTS AND DISCUSSION**



****

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

* 1. **Conclusion**

The development of the SwiftToll IoT-based toll collection system marks a significant advancement in streamlining the tolling process and improving traffic management at toll booths. By integrating IoT technologies, such as vehicle detection sensors, Wi-Fi communication, and real-time payment verification, the system automates toll collection, reduces wait times, and enhances traffic flow. This not only improves the overall user experience but also contributes to reducing congestion and ensuring a smoother and more efficient tolling process for commuters. SwiftToll has the potential to be scaled and integrated into larger toll networks, offering an innovative solution for modernizing toll operations.

# Future Work

Future enhancements for the SwiftToll system focus on scalability, performance optimization, and further integration with emerging technologies. This includes exploring advanced sensor technologies that offer greater accuracy in vehicle detection, improving the communication infrastructure for faster, more reliable interactions between the toll gate hardware, backend systems, and user mobile apps. We also plan to integrate machine learning algorithms to analyze traffic patterns and predict peak times, further optimizing toll collection and gate management. Additionally, the integration of more secure and efficient payment protocols will be explored to ensure the safety and privacy of user transactions. By continually evolving the system, SwiftToll aims to provide a robust, scalable solution for modern toll management that can be deployed across diverse urban environments.

# APPENDIX

**SOFTWARE INSTALLATION**

**Arduino IDE**

To run and mount code on the Arduino NANO, we need to first install the Arduino IDE. After running the code successfully, mount it.

# Sample code

#include <ESP8266WiFi.h>

#include <WebSocketsClient.h>

#include <Servo.h>

#include <ArduinoJson.h>

const char\* ssid = "OnePlus 11R 5G";

const char\* password = \*\*\*\*\*\*\*\*;

const char\* websocket\_server = "192.168.115.165";

const int websocket\_port = 8765;

WebSocketsClient webSocket;

Servo barrierServo;

#define SERVO\_PIN 2

bool isWebSocketConnected = false;

void handleWebSocketMessage(char\* message, size\_t length);

void sendConnectionMessage();

void setup() {

    Serial.begin(115200);

    WiFi.begin(ssid, password);

    Serial.print("Connecting to WiFi...");

    unsigned long startAttemptTime = millis();

    while (WiFi.status() != WL\_CONNECTED) {

        delay(500);

        Serial.print(".");

        if (millis() - startAttemptTime > 15000) {

            Serial.println("\nWiFi Connection Failed! Restarting...");

            ESP.restart();

        }

    }

    Serial.println("\nConnected to WiFi, IP: " + WiFi.localIP().toString());

    webSocket.begin(websocket\_server, websocket\_port, "/");

    webSocket.onEvent(webSocketEvent);

    barrierServo.attach(SERVO\_PIN);

    barrierServo.write(0);

    Serial.println("Barrier initialized in closed position");

    Serial.println("Connecting to WebSocket server...");

}

void loop() {

    if (WiFi.status() != WL\_CONNECTED) {

        Serial.println("WiFi disconnected! Reconnecting...");

        WiFi.disconnect();

        WiFi.begin(ssid, password);

        unsigned long startReconnectTime = millis();

        while (WiFi.status() != WL\_CONNECTED) {

            delay(500);

            Serial.print(".");

            if (millis() - startReconnectTime > 15000) {

                Serial.println("\nWiFi Reconnection Failed! Restarting...");

                ESP.restart();

            }

        }

        Serial.println("\nWiFi Reconnected!");

    }

    webSocket.loop();

}

void sendConnectionMessage() {

    DynamicJsonDocument doc(256);

    doc["action"] = "connect";

    doc["device\_type"] = "esp8266";

    String message;

    serializeJson(doc, message);

    webSocket.sendTXT(message);

}

void webSocketEvent(WStype\_t type, uint8\_t \*payload, size\_t length) {

    switch (type) {

        case WStype\_CONNECTED:

            Serial.println("WebSocket Connected!");

            isWebSocketConnected = true;

            sendConnectionMessage();

            break;

        case WStype\_DISCONNECTED:

            Serial.println("WebSocket Disconnected! Reconnecting...");

            isWebSocketConnected = false;

            break;

        case WStype\_TEXT:

            handleWebSocketMessage((char\*)payload, length);

            break;

        case WStype\_ERROR:

            Serial.print("WebSocket Error: ");

            Serial.println((char\*)payload);

            break;

    }

}

void handleWebSocketMessage(char\* message, size\_t length) {

    if (message == nullptr || length == 0) return;

    Serial.print("Received: ");

    Serial.println(message);

    DynamicJsonDocument doc(1024);

    DeserializationError error = deserializeJson(doc, message);

    if (error) {

        Serial.print("deserializeJson() failed: ");

        Serial.println(error.c\_str());

        return;

    }

    if (doc.containsKey("barrier\_control")) {

        String command = doc["barrier\_control"].as<String>();

        Serial.print("Barrier control command: ");

        Serial.println(command);

        if (command == "Paid") {

            Serial.println("Payment Verified! Opening Barrier...");

            barrierServo.write(90);

            delay(3000);

            barrierServo.write(0);

            Serial.println("Barrier Closed");

        } else if (command == "Not Paid") {

            Serial.println("Payment Not Verified - Barrier Remains Closed");

            barrierServo.write(0);

        }

    }

}

# REFERENCES

[1] J. Y. Tan, P. J. Ker, D. Mani and P. Arumugam, "Development of a GPS-based highway toll collection system," *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, 2016, pp. 125-128, doi: 10.1109/ICCSCE.2016.7893557.

[2] S. Lu, T. He and Z. Gao, "Design of electronic toll collection system based on global positioning system technique," *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, Sanya, China, 2009, pp. 350-353, doi: 10.1109/CCCM.2009.5268110.

[3] M. Randriamasy, A. Cabani, H. Chafouk and G. Fremont, "Geolocation Process to Perform the Electronic Toll Collection Using the ITS-G5 Technology," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8570-8582, Sept. 2019, doi: 10.1109/TVT.2019.2931883. keywords: {Europe;Sensors;Geology;Automobiles;Global Positioning System;Road safety;ETC system;V2X Communication;Mapping and Localization;Enhanced Dynamic Model;Cooperative messages},