Ain Shams University
Faculty of Computer & Information Sciences
Computer Systems Department

GPS Based Car Tracking System

This documentation was submitted as required for the degree of bachelor's in computer and information Sciences.

By

Omar Ali Yassin

Ibrahim Mohamed Mostafa

Osama Alaa El-din Abdelsamea

Rawan Alaa El-din Hassan

Nouran Khaled Tolba

Under Supervision of

D. Heba Khaled
Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

T.A. Omar Khaled
Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

June 2024

Acknowledgements

All praise and thanks to Allah, who has granted us the ability to successfully accomplish this work. We sincerely hope that our efforts are accepted by Him.

We extend our heartfelt gratitude to our parents and family members who have consistently provided us with unwavering support and assistance throughout our years of study. We aspire to reciprocate their kindness and support in the future.

We would like to express our sincere appreciation to all those who have supported us in the completion of this project. We are especially grateful to our supervisor, Dr. Heba Khaled, for her invaluable support and guidance in coordinating our project. We would also like to extend our thanks to T.A. Omar Khaled, a Teaching Assistant at Ain Shams University, for his valuable suggestions that made our project more practical. We are grateful for his support, suggestions, and guidance during the learning phase.

Lastly, we cannot adequately express our immense gratitude and appreciation to our family and colleagues for their unwavering encouragement and support throughout this challenging and rewarding journey of completing this project. Their presence has been invaluable, and we are forever grateful for their contributions.

Abstract

Imagine a world where you can keep an eye on your cars from anywhere, anytime. Location Tracker is a revolutionary GPS-based car tracking system that enables a single user to monitor multiple vehicles through a user-friendly web application. Each car is equipped with a GPS module that transmits location and speed data every 10 seconds to an ESP32 Wi-Fi Module and subsequently to a SIM card. This constant stream of data allows users to track vehicles on a map, set geofences, and receive instant alerts if a vehicle exits its designated zone or exceeds a speed limit. The system categorizes locations by city, such as Cairo or Alexandria, etc., based on coordinates. Developed using Flask for API handling, PostgreSQL for the database, and a React front-end, Location Tracker ensures real-time monitoring and alerting, efficiently handling multiple users and vehicles. This powerful tool is perfect for applications like monitoring school buses or allowing parents to track their children's driving, providing a reliable solution for enhanced safety and security.

Table of Contents

Acknowledgements 2
Abstract 3
List of Figures 5
Chapter One: Introduction 6
1.1. Problem Definition 6
1.2. Motivation 7
1.3. Objectives 8
1.4. Methodology 9
1.5. Time Plan 12
1.6. Thesis Outline 13
Chapter Two: Literature Review 14
2.1. Introduction 14
2.2. Theoretical Background 15
Chapter Three: System Architecture and Methods 17
3.1. System's Architecture 17
3.2. Functional Requirements (Methods) 19
Chapter Four: System Implementation and Results 21
4.1. Description of Programs used 21
4.2. Software Used in Our Project 23
4.3. Setup Configuration 25
4.4. Experimental and Results 30
Chapter Five: Run the Application 34
Chapter Six: Conclusion and Future Work 42
6.1. Conclusion 42
6.2. Future Work: 44
REFERENCES 46

List of Figures

Chapter One: Introduction
Figure 1.1: Problem Definition 6
Figure 1.2: Motivation..... 7
Figure 1.3: Gantt Chart for Time Plan 12
Chapter Two: Literature Review
Chapter Three: System Architecture and Methods
Figure 3.1: System's architecture Diagram..... 18
Chapter Four System Implementation and Results
Figure 4.1: ESP32 WIFI module..... 25
Figure 4.2: NEO-6M GPS Module ……27
Figure 4.3: TP4056..... 28
Figure 4.4: SIM700C..... 29

Chapter One: Introduction

1.1. Problem Definition

This system tracks vehicles using GPS modules installed in the cars, which report their location and speed to a web application.

Figure 1.1: Problem Definition

These challenges include:

1. Data Accuracy:
    - Ensuring the GPS data is accurate and updated every 10 seconds.
2. Real-time Processing:
    - Handling real-time data transmission and processing for multiple vehicles simultaneously.
3. Alert System:
    - Developing a reliable alert system for geographical zone and speed limit violations.
4. User Interface:
    - Creating an intuitive and user-friendly interface for tracking and setting parameters.

By defining and addressing these challenges, our project seeks to provide vehicle owners with a comprehensive GPS-based car tracking system that offers enhanced security, real-time monitoring, and efficient alert management, while also integrating smoothly with existing web technologies and user interfaces.

1.2. Motivation

The motivation behind our GPS-based car tracking project stems from the growing need for enhanced vehicle security and efficient fleet management. With increasing instances of vehicle theft and misuse, it is essential to have a reliable system that allows vehicle owners to monitor their cars in real-time. Additionally, for applications such as school buses and parental control over family vehicles, it is crucial to ensure the safety of passengers and responsible vehicle usage. By providing a solution that tracks vehicle locations, sets speed limits, and establishes geographical zones, we aim to offer peace of mind to vehicle owners, improve safety, and streamline the management of multiple vehicles. Our project leverages modern web technologies to create an intuitive, scalable, and robust tracking system, addressing the critical need for security and accountability in vehicle management.

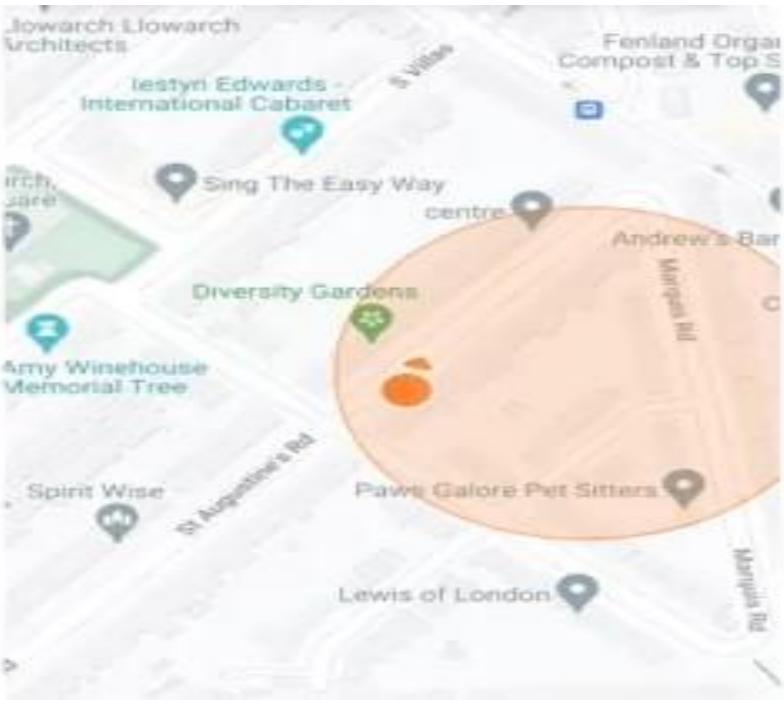


Figure 1.2: Motivation

1.3. Objectives

The primary objectives of our GPS-based car tracking project are:

1. **Real-Time Vehicle Tracking:** Develop a system that provides real-time tracking of vehicles using GPS modules, updating location data (longitude and latitude) every 10 seconds.
2. **Geofencing Capabilities:** Enable users to set geographical zones for each vehicle and receive alerts if a vehicle exits its designated zone, enhancing security and monitoring capabilities.
3. **Speed Monitoring:** Allow users to set speed limits for vehicles and generate alerts if these limits are exceeded, promoting safe driving practices.
4. **City Categorization:** Automatically categorize and display the vehicle's location by city (e.g., Cairo, Alexandria) based on its GPS coordinates.
5. **User-Friendly Interface:** Create an intuitive and responsive web application using React for front-end, ensuring ease of use for tracking and managing multiple vehicles.
6. **Robust Backend Integration:** Utilize Flask for handling API requests and PostgreSQL for database management to ensure reliable data processing and storage.
7. **Scalability and Performance:** Design the system to efficiently handle multiple users and vehicles, ensuring scalability for future growth.
8. **Enhanced Security:** Implement data security measures to protect user information and ensure the privacy of vehicle tracking data.
9. **Violation Alert System:** Develop a reliable alert system that notifies users of geographical zone and speed limit violations in real-time.

1.4. Methodology

1. **Requirements Gathering:** Define and understand the functionalities required for the GPS Tracking System, including real-time location tracking, geofencing capabilities, speed monitoring, city categorization based on GPS coordinates, and alert systems for violations. Gather specific requirements from stakeholders such as vehicle owners, fleet managers, and users of the tracking system.

2. **Hardware Selection:** Identify and select appropriate hardware components, including GPS modules, ESP32 Wi-Fi Module, SIM card for data transmission. Ensure compatibility and reliability of the selected hardware components for continuous and accurate data collection.

3. **Software Development:**
    - **Backend Development:** Implement backend functionalities using Flask framework for API development, PostgreSQL for database management, and integration with GPS data processing algorithms.
    - **Frontend Development:** Develop a responsive web application using React for real-time vehicle tracking visualization, setting geofences, defining speed limits, and displaying alerts.
    - **Integration with External Services:** Integrate mapping APIs (e.g., Google Maps) for visual representation of vehicle locations and city categorization based on GPS coordinates.

4. **Data Collection and Processing:**
    - **GPS Data Handling:** Establish mechanisms for collecting GPS data (longitude, latitude, speed) from vehicles at regular intervals (e.g., every 10 seconds) and transmitting it to the backend server.
    - **Data Storage and Management:** Design and implement a PostgreSQL database schema for storing vehicle data, user information, geofence configurations, speed limits, and historical tracking data.

5. **System Integration:**
    - **Hardware Integration:** Assemble and integrate GPS modules, ESP32 Wi-Fi Module, and SIM card into vehicles to ensure seamless data transmission and communication with the backend system.
    - **Software Integration:** Integrate backend APIs with the frontend web application to ensure real-time updates, alert notifications, and user interactions with the tracking system.

6. **Testing and Validation:**
    - **Functionality Testing:** Conduct rigorous testing of each system component, including GPS data accuracy, geofencing functionalities, speed limit alerts, and city categorization accuracy.
    - **Performance Testing:** Evaluate the system's performance under varying load conditions to ensure scalability, responsiveness, and reliability.
    - **User Acceptance Testing:** Engage stakeholders and end-users to validate the system's usability, interface intuitiveness, and adherence to functional requirements.

By following this methodology, we aim to develop a robust and efficient GPS-based car tracking system that enhances vehicle security, enables effective fleet management, and provides real-time monitoring capabilities for users across various applications.

1.5. Time Plan:

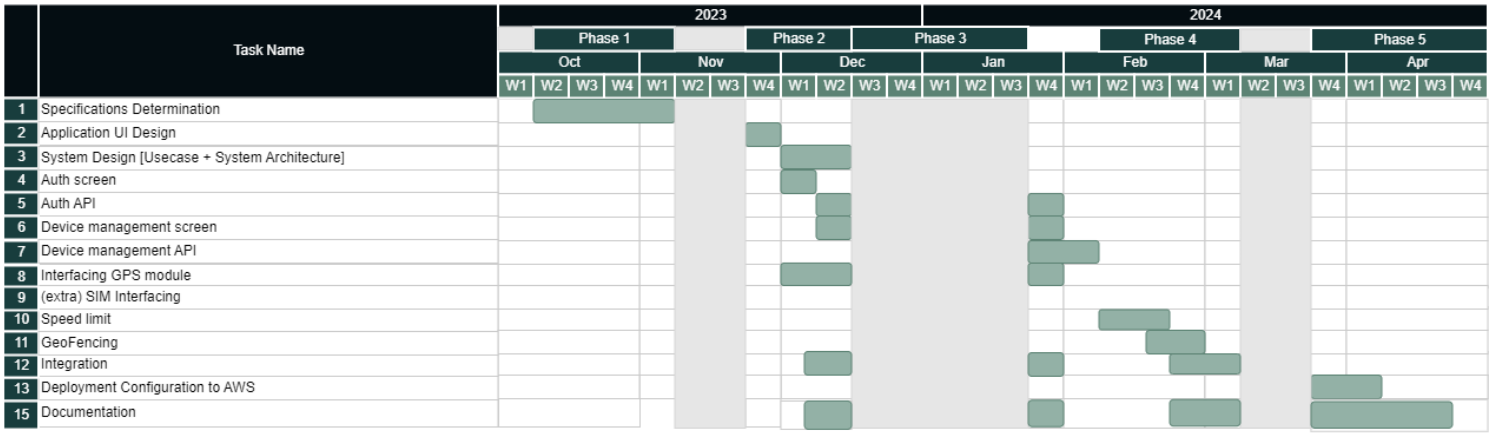


Figure 1.3: Gantt Chart for Time Plan

1.6. Thesis Outline:

The rest of this thesis is organized as follows:
Chapter 1 highlights the background and Literature Review, to situate the forthcoming research within the extant theoretical paradigms and emphasize areas where further scholarship is needed. **Chapter 2** provides System Architecture and Methods explaining how a system is working in detail with a particular focus on the scope and population parameters used. **Chapter 3** shows the implementation details and hardware explained. **Chapter 4** presents description of materials used, step-up configuration (Hardware), experimental and results, presentation of results use of tables and figures, relating results to hypotheses. Comparison with previous studies
Chapter 5 comprises a conclusion and outlines potential avenues for future work. Finally, **Chapter 6** explains how to run the application, providing users with a practical guide to effectively utilize the system.

Chapter Two: Literature Review

2.1. Introduction

The rapid evolution of technology has spurred the development of innovative solutions aimed at enhancing vehicle security and operational efficiency. Among these solutions, GPS-based tracking systems represent a pivotal advancement in fleet management and vehicle monitoring. These systems utilize a combination of hardware components such as GPS modules, communication modules like ESP32 Wi-Fi, and software solutions including web applications and databases. By integrating these elements, GPS tracking systems facilitate real-time monitoring of vehicle locations, speed, and adherence to predefined geographical boundaries.

The primary objective of this project is to review existing research and developments in GPS-based tracking systems, focusing specifically on their implementation and operational methodologies. By examining the theoretical foundations and practical applications in this domain, we aim to elucidate current practices, identify potential improvements, and propose novel approaches to enhance the functionality and reliability of GPS tracking systems.

Theoretical advancements in GPS technology, data processing, and integration methodologies are pivotal in optimizing the performance of tracking systems. By harnessing these advancements, GPS tracking systems can provide accurate and timely updates on vehicle locations, facilitate efficient route planning, and ensure compliance with safety and security protocols. Moreover, the integration of robust software frameworks like Flask for API development, PostgreSQL for data management, and React for user interfaces enhances the system's scalability, responsiveness, and user accessibility.

2.2. Theoretical Background

The Theoretical Background section aims to establish the foundational concepts and principles underpinning GPS-based tracking systems, focusing on their operational mechanisms and technological components. By delving into these fundamental aspects, we can gain a comprehensive understanding of the methodologies and technologies utilized in the development and implementation of such systems.

A. GPS Technology and Data Processing

Global Positioning System (GPS) technology forms the cornerstone of GPS-based tracking systems, enabling precise localization of vehicles through satellite-based navigation. Key theoretical concepts include:

- Satellite Communication: GPS modules communicate with satellites to determine accurate coordinates (longitude and latitude) and provide real-time location updates.
- Data Transmission: Data, including location coordinates and vehicle speed, is transmitted via communication modules like ESP32 Wi-Fi and SIM cards, ensuring continuous and reliable data flow.

B. Software Frameworks and Integration

Effective integration of software components is crucial for the seamless operation and management of GPS tracking systems. Theoretical foundations include:

- Backend Development: Utilization of Flask for API development facilitates robust communication between the frontend and backend systems, enabling real-time data processing and management.

- Database Management: PostgreSQL serves as the foundational database management system, ensuring secure storage and efficient retrieval of vehicle data, user information, and system configurations.
- Frontend Interfaces: React is employed to develop user-friendly web interfaces that enable vehicle tracking visualization, geofence management, and alert notifications for users.

By exploring these theoretical foundations, this section aims to provide a comprehensive understanding of the principles driving the development and functionality of GPS-based tracking systems. This knowledge serves as a basis for exploring current practices, identifying potential improvements, and proposing innovative approaches to enhance the effectiveness and reliability of vehicle monitoring and management.

Chapter Three: System Architecture and Methods

3.1. System's Architecture

1. Data base Tables

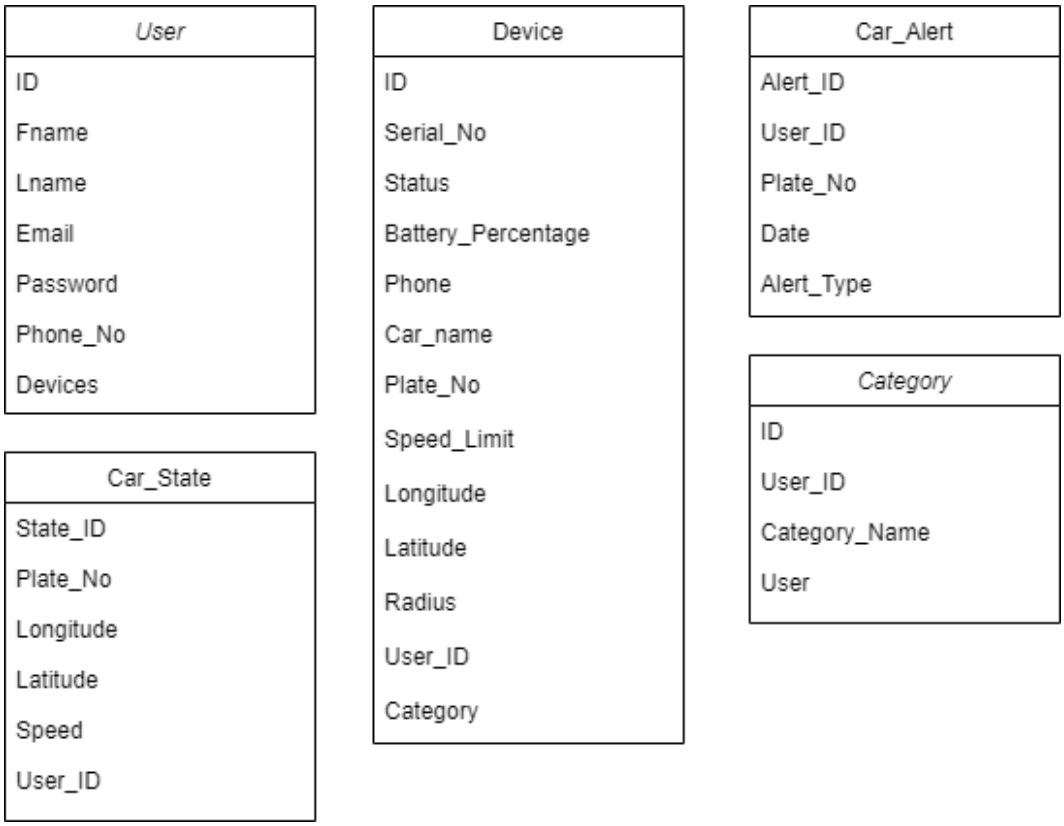


Figure 3.3: Data base Tables

2. System’s architecture

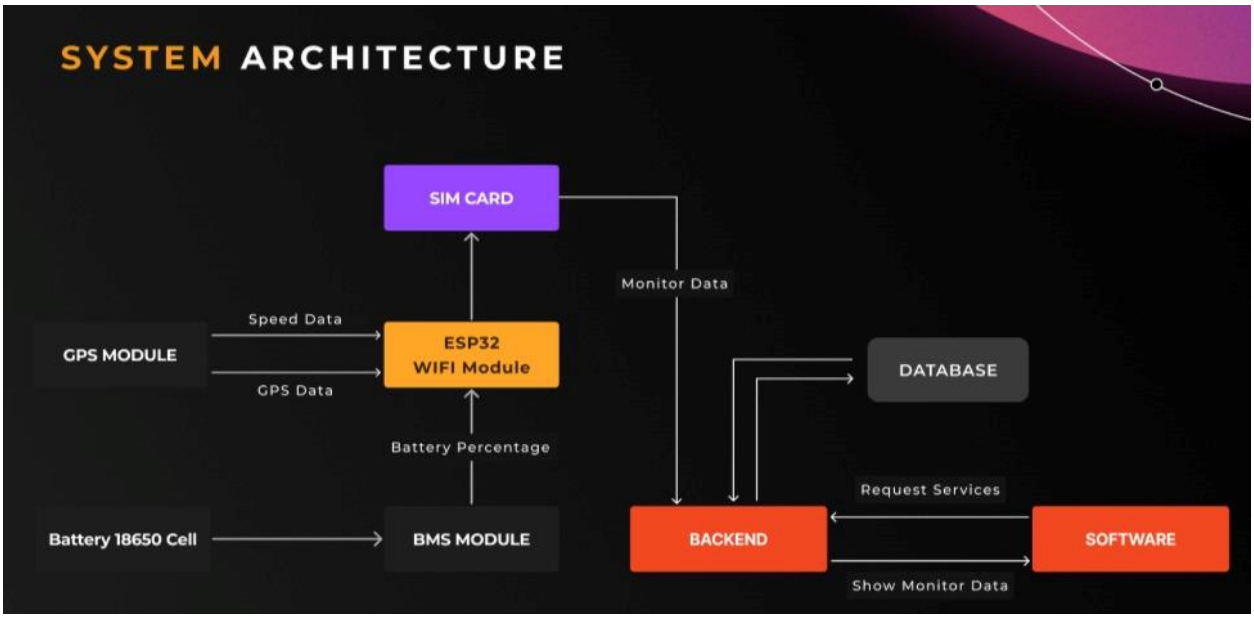


Figure 3.1: System’s architecture Diagram

Scenario 1:

A vehicle equipped with a GPS module and SIM card periodically sends an HTTP POST request through an AT command to the backend server every 10 seconds. The request includes the vehicle's current location coordinates (longitude and latitude) and speed. The backend server processes this data to update the vehicle's position on a web application map in real-time. If the vehicle crosses a predefined geographical boundary (geofence) or exceeds the set speed limit, the server triggers an alert. The alert is then sent back in the response to the web application, and got saved in the alert section.

Scenario 2:

The GPS module in a vehicle records the vehicle's speed and sends an HTTP POST request containing this data to the backend server through an AT command using the SIM card. The backend server listens for this request. Upon receiving the request, the server checks if the recorded speed exceeds the predefined speed limit set by the user. If a speed violation is detected, the backend server updates the user's web application interface to display a speed violation alert. This alert includes details such as the time of the violation, the recorded speed, and the location of the vehicle when the violation occurred. The user can then take necessary actions based on the alert received.

3.2. Functional Requirements (Methods)

1. **Speed Detection:**
   The system should accurately detect and record the speed of the vehicle using the GPS module. This functionality is essential for monitoring and ensuring compliance with speed limits.
2. **Speed Limit Enforcement:**
   The system should compare the detected speed with a predefined speed limit to determine if a violation has occurred. This comparison helps in enforcing speed regulations and identifying speeding incidents. Upon detecting a speed violation, the system should generate an alert and update the user interface with details.
3. **Geofence Detection:**
   The system should detect if the vehicle is within a predefined geographical boundary (geofence). This functionality is crucial for monitoring the vehicle's location and ensuring it stays within designated areas. Upon detecting a geofence violation, the system should generate an alert and update the user interface with details.
4. **Data Transmission:**
   The GPS module should send location coordinates and speed data to the backend server through an HTTP POST request using an AT command via the SIM card. This transmission ensures real-time data monitoring and processing.
5. **Real-Time Map Update:**
   The system should update the vehicle's position on a web application map in real-time based on the received location data. This feature provides users with a visual representation of the vehicle's current location.

6. Web Application:

The system should have a Web application that provides a user-friendly interface for accessing and interacting with the system. The application allows users to perform tasks such as logging in, viewing alerts, and managing the devices.

1. **Login (Using JWT):**
   The system should have login functionality in the application to ensure secure access. Users should be able to authenticate themselves using appropriate credentials, such as usernames and passwords.
2. **View Alerts:**
   The mobile application should provide a feature to view Alerts received from the Hardware. Users can check and read the alerts conveniently through the app's interface.
3. **Add/Remove Devices:**
   The application should allow authorized users to manage the devices. This includes adding or removing devices.
4. **Category Management:**
   The system should enable users to create and manage categories for organizing their devices. This functionality helps in categorizing and easily accessing information about different vehicles or devices.

Overall, these functional requirements ensure that the system can perform essential tasks such as speed detection, speed limit enforcement, and geofence detection. Additionally, the system supports critical functionalities like distance calculation, real-time data transmission, and alert generation for speed and geofence violations. Furthermore, the system provides comprehensive user and device management features, including user authentication, device management, category management, and user information updates. The system also supports alert retrieval, real-time map updates, and device status monitoring, ensuring users have up-to-date information and can effectively manage their vehicles and tracking devices.

Chapter Four: System Implementation and Results

4.1. Description of programs used

The GPS Tracking System project utilizes a combination of software programs for different purposes. Here are some examples:

1. **Python (Python Software Foundation):**
   Python is a versatile programming language used for various tasks in the project, including data preprocessing, machine learning model development, system integration, and database management. Its simplicity, extensive libraries, and robust ecosystem make it a popular choice for AI and software development.

2. **Flask (Pallets Projects):**
   Flask is a lightweight web framework used for developing web applications in Python. It can be employed to create the backend of the mobile app interface for the GPS Tracking System, allowing homeowners to manage and control the system's functionalities.

3. **Visual Studio Code:**
   Visual Studio Code is a versatile code editor developed by Microsoft. It provides an integrated environment for writing and managing code with features such as syntax highlighting, debugging, and extensions.
4. **PostgreSQL (PostgreSQL Global Development Group)**
   PostgreSQL is an open-source relational database management system known for its robustness and scalability. We used it for storing and managing all project data, including user information, device data, and alerts.

5. **SQLAlchemy (SQLAlchemy Project):**
   SQLAlchemy is an open-source SQL toolkit and Object-Relational Mapping (ORM) library for Python. It provides a set of high-level APIs for interacting with databases, allowing developers to work with relational databases using Python objects and SQL expressions. SQLAlchemy can be used in the project for database management, storing and retrieving data related to authorized individuals, visitor logs, and system settings of the GPS Tracking System. It offers flexibility, abstraction, and compatibility with different database systems, making it a useful tool for data persistence and management.
6. **AT Commands:**
   AT commands are instructions used to control modems and other devices that communicate over serial connections. Employed by GPS modules to send speed data to the backend via the SIM card.
7. **React (Meta Platforms, Inc.):**
   React is a JavaScript library for building user interfaces, particularly single-page applications. Used to develop the frontend of the web application, providing users with an interactive and responsive interface.
8. **JSON Web Tokens:**
   JWT is an open standard for securely transmitting information between parties as a JSON object, Utilized for user authentication and authorization, ensuring secure access to the application.

By leveraging these technologies, the GPS Tracking System project ensures efficient data processing, secure authentication, and real-time monitoring, providing users with a reliable and user-friendly solution for vehicle tracking and management.

4.2. Software Used in Our Project

Our project integrates various software components to facilitate seamless communication and functionality between hardware modules and the mobile application. Below are the key software components and their roles within the system:

1. Back End Server (Flask and PostgreSQL)

As part of our project, we have developed a backend server using Flask, a lightweight and flexible web framework in Python. PostgreSQL is utilized as the database management system to store and manage data securely.

- **Key App Routes and Functionalities:**

1. **Device Registration and Management:**
    - Allows devices (ESP32, GPS modules) to register and manage their configurations and settings through HTTP POST requests.
2. **User Authentication and Authorization:**
    - Implements user authentication and authorization using JWT (JSON Web Tokens), ensuring secure access to system functionalities.
3. **Speed Violation Monitoring:**
    - Receives speed data from GPS modules via AT commands through SIM cards, checks against predefined speed limits, and logs violations.
4. **Geofencing Functionality:**
    - Monitors device locations using GPS coordinates and determines whether they are within predefined geographical boundaries. Sends alerts for geofence violations.

5. **Database Management and Reporting:**
    - Provides endpoints for CRUD (Create, Read, Update, and Delete) operations on user accounts, devices, and alert logs stored in PostgreSQL.

2. React Front End (Mobile Application)

The project includes a React-based front end for the mobile application, providing a user-friendly interface to interact with the system's functionalities.

- **Key Features:**

    1. **User Interface Design:** Implements intuitive UI components for device management, speed monitoring, geofencing, and alert notifications.
    2. **Real-time Data Display:** Integrates with backend APIs to fetch and display real-time device statuses, location data, and alert logs.
    3. **User Authentication:** Allows users to securely log in, manage their profiles, and access system features based on their roles and permissions.
    4. **Interactive Mapping:** Utilizes mapping libraries to visualize device locations, geofence boundaries, and alerts for enhanced situational awareness.

4.3. Setup Configuration

1. Hardware:

1. Hardware Component:

i. ESP32
ii. NEO-6M GPS Module
iii. TP4056
iv. Boost Converter
v. SIM7000C

2. Hardware Details:

1. ESP32

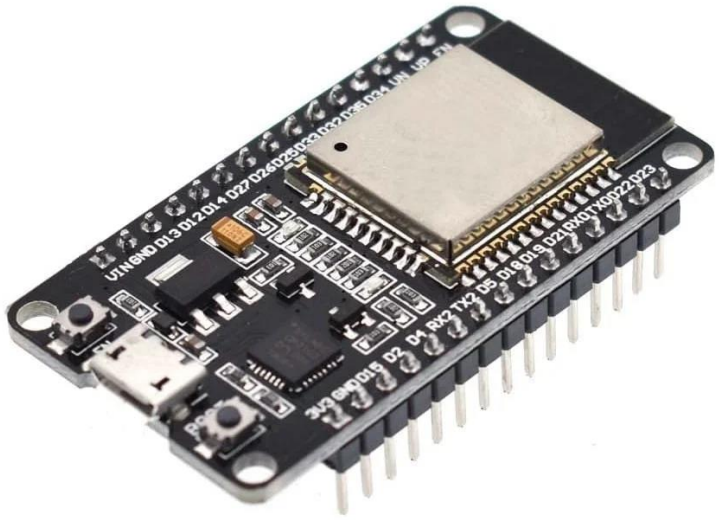


Figure 4.1: ESP32 WIFI module

The ESP32 is a low-cost, low-power system on a chip (SoC) developed by Espressif Systems. It features integrated Wi-Fi and Bluetooth capabilities, making it a versatile solution for various IoT applications. Here are detailed specifications and features of the ESP32 Wi-Fi model:

Key Features:

1. **Integrated Wi-Fi and Bluetooth**
    - The ESP32 supports IEEE 802.11 b/g/n Wi-Fi standards.
    - It includes Bluetooth v4.2 BR/EDR and BLE (Bluetooth Low Energy).
2. **Powerful Processing Capabilities**
    - It features a dual-core Tensilica LX6 microprocessor.
    - It operates at a clock frequency of up to 240 MHz.
    - It has 520 KB of SRAM and 448 KB of ROM.
3. **Low Power Consumption**
    - The ESP32 is designed for ultra-low power consumption applications.
    - It includes multiple power modes and power-saving features.
    - It supports fine-grained clock gating, dynamic voltage and frequency scaling, and power modes including deep sleep and light sleep.
4. **Rich Peripheral Interfaces**
    - It offers multiple GPIO pins, PWM, ADC (Analog-to-Digital Converter), DAC (Digital-to-Analog Converter), I2C, SPI, UART, and SDIO interfaces.
    - It includes touch sensors, temperature sensors, and Hall sensors.

Specifications:

1. **Wi-Fi Specifications**
    - Frequency Range: 2.4 GHz ~ 2.5 GHz
    - Protocols: 802.11 b/g/n (802.11n up to 150 Mbps)
    - Security: WPA/WPA2, WAPI
2. **Bluetooth Specifications**
    - Protocols: Bluetooth v4.2 BR/EDR and BLE
3. **Processing and Memory**
    - CPU: Dual-core Tensilica LX6 up to 240 MHz
    - RAM: 520 KB SRAM
    - ROM: 448 KB ROM
    - Flash: Typically 4 MB (can vary depending on the specific module)

4. **Power Consumption**
    - Operating Voltage: 2.2 V to 3.6 V
    - Deep Sleep Current: ~10 µA
    - Light Sleep Current: ~0.8 mA
    - Active Mode Current: ~80 mA
5. **Peripheral Interfaces**
    - GPIO: 34 programmable GPIOs
    - ADC: 18 channels of 12-bit ADC
    - DAC: 2 channels of 8-bit DAC
    - PWM: Up to 16 channels
    - Communication Interfaces: 4 SPI, 2 I2S, 2 I2C, 3 UART, SDIO, Ethernet MAC, CAN 2.0

2. NEO-6M GPS Module:

Specifications:

- High Sensitivity and Accuracy: It supports a position accuracy of up to 2.5 meters.

- Designed for low power consumption, the module is ideal for battery-operated applications.

- Power Consumption:

Operating Voltage: 2.7V to 3.6V

Power Consumption: 45 mA (typical)

Figure 4.2: NEO-6M GPS Module

3. TP4056

The TP4056 is a linear battery charger IC specifically designed for single-cell lithium-ion batteries. It features a complete constant-current/constant-voltage linear charger suitable for applications such as charging circuits in portable devices. Here are the key features and specifications of the TP4056:

Figure 4.3: TP4056

Key Features:

1. **High Accuracy Charging**
   - Provides accurate constant current/constant voltage charging.
2. **Automatic Recharge**
   - Automatically recharges the battery when the voltage drops below a certain threshold.
3. **Built-in Protection**
   - Includes overvoltage, overcurrent, and thermal protection.
4. **Status Indicators**
   - Features LED indicators for charge status.

Specifications:

1. **Input Voltage:**
   - 4.5V to 5.5V
2. **Charge Voltage:**
   - 4.2V ± 1%
3. **Charge Current:**
   - Programmable up to 1A

4. SIM700C:

The SIM7000C is a cellular module designed for IoT (Internet of Things) applications, providing reliable communication over cellular networks. Here are the key features and specifications of the SIM7000C module:

Figure 4.4: SIM700C

Key Features:

2. **Cellular Connectivity**: Operates on multiple frequency bands, ensuring compatibility with various carriers worldwide.
3. **Integrated GNSS:** Includes GPS/GLONASS/BeiDou/Galileo/QZSS support for accurate positioning and location-based services.
4. **Low Power Consumption:** Optimized for low-power applications, extending battery life in IoT devices.
5. **SIM Card Interface:** Supports standard SIM cards for network authentication and data connectivity.
6. **AT Command Interface:** Communicates with the host microcontroller via AT commands, simplifying integration and control.

4.4. Experimental and Results

During the project, we tested a lot of cases to make our project get the optimal results.

Experimental Setup:

1. **Location Detection:** We were able to accurately locating the device using longitude and latitude.

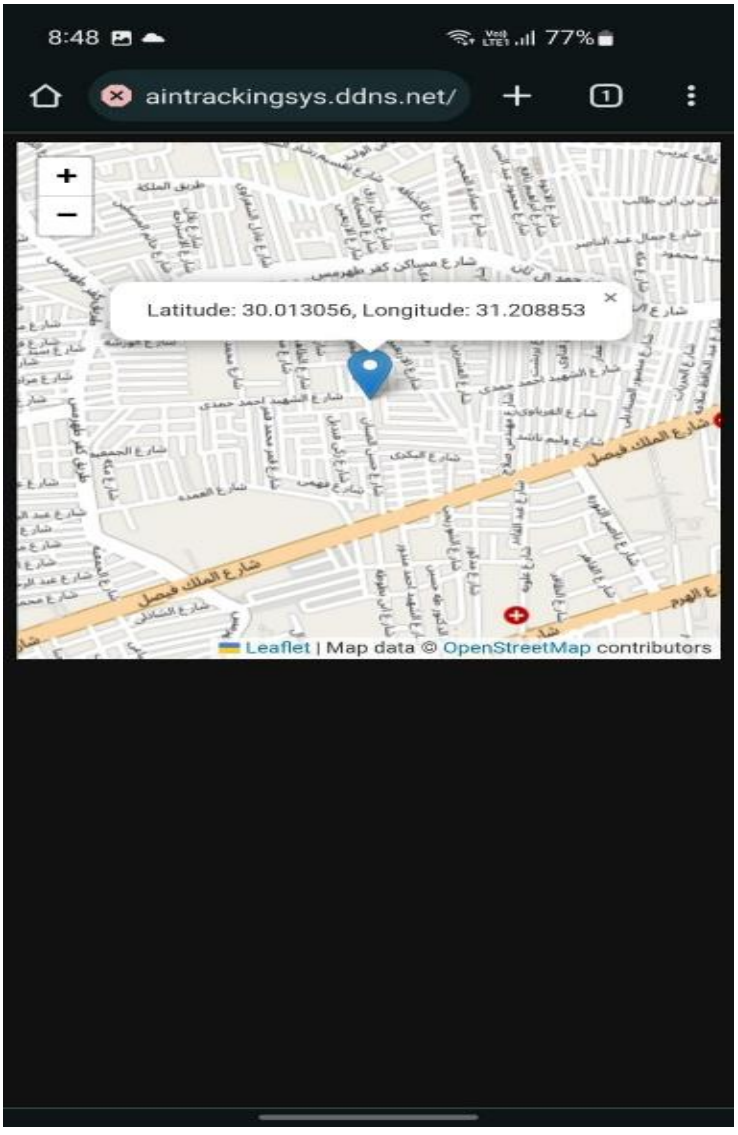


Figure 4.5: Accurate Locating

2. **Device Registration:** We were able to add devices to specific user and add it to a category.

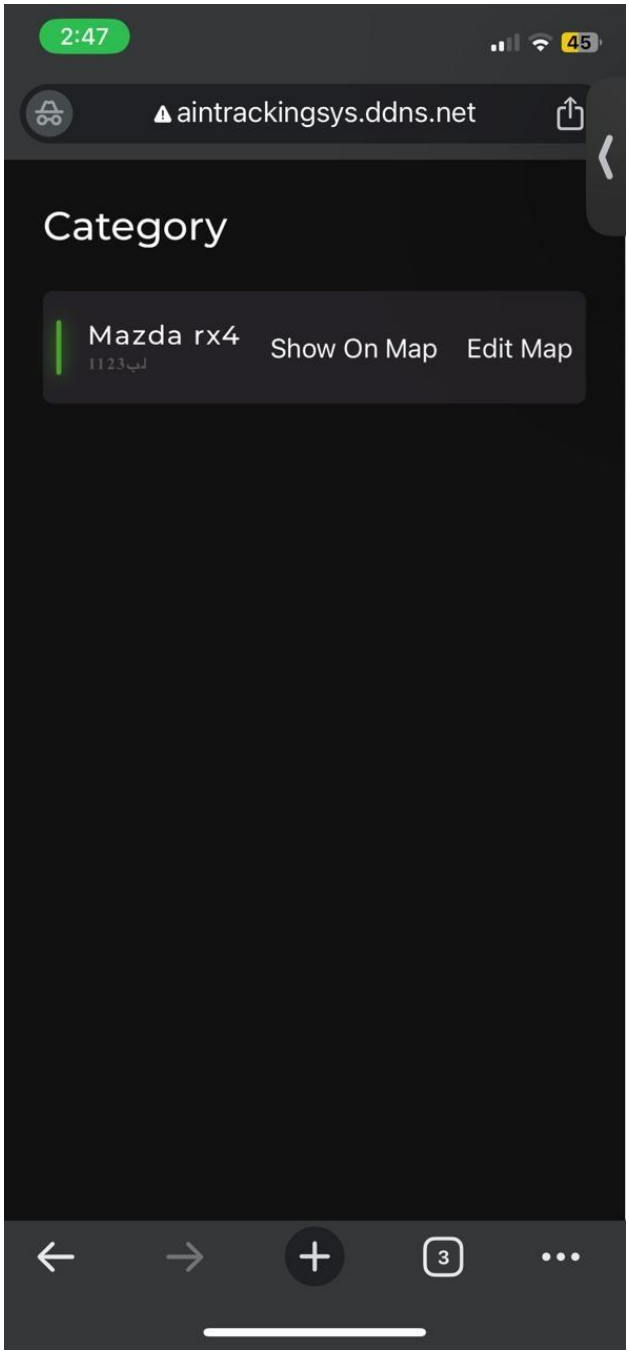


Figure 4.6: Adding Devices

3. **Alerts:** Also adding alerts to a user regarding whether the driver accused of Geographical Violation or Speed Violation.

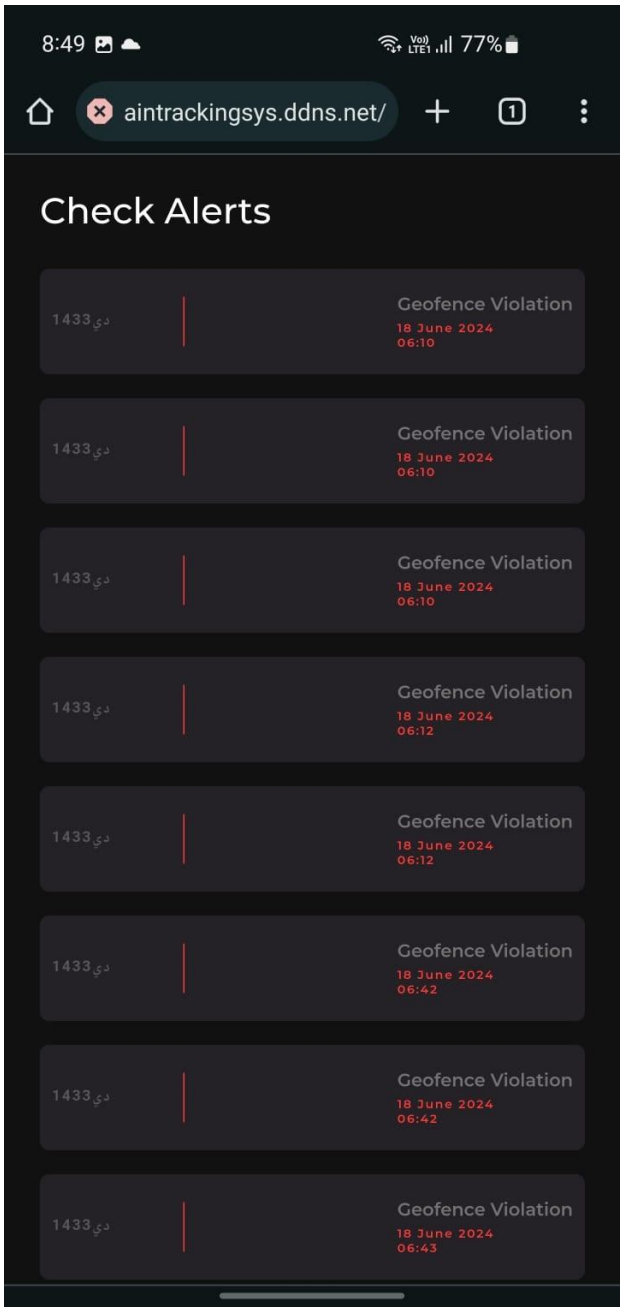


Figure 4.7: Alerts on Car

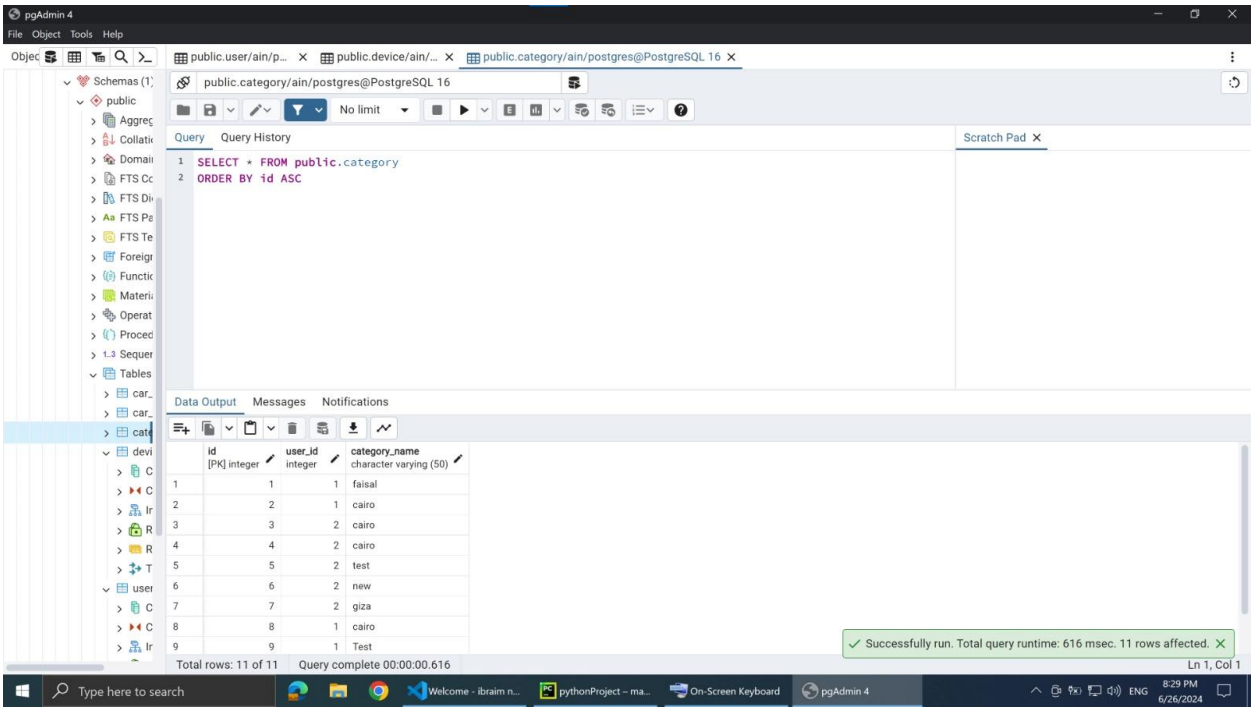


Figure 4.7: Categories added by User

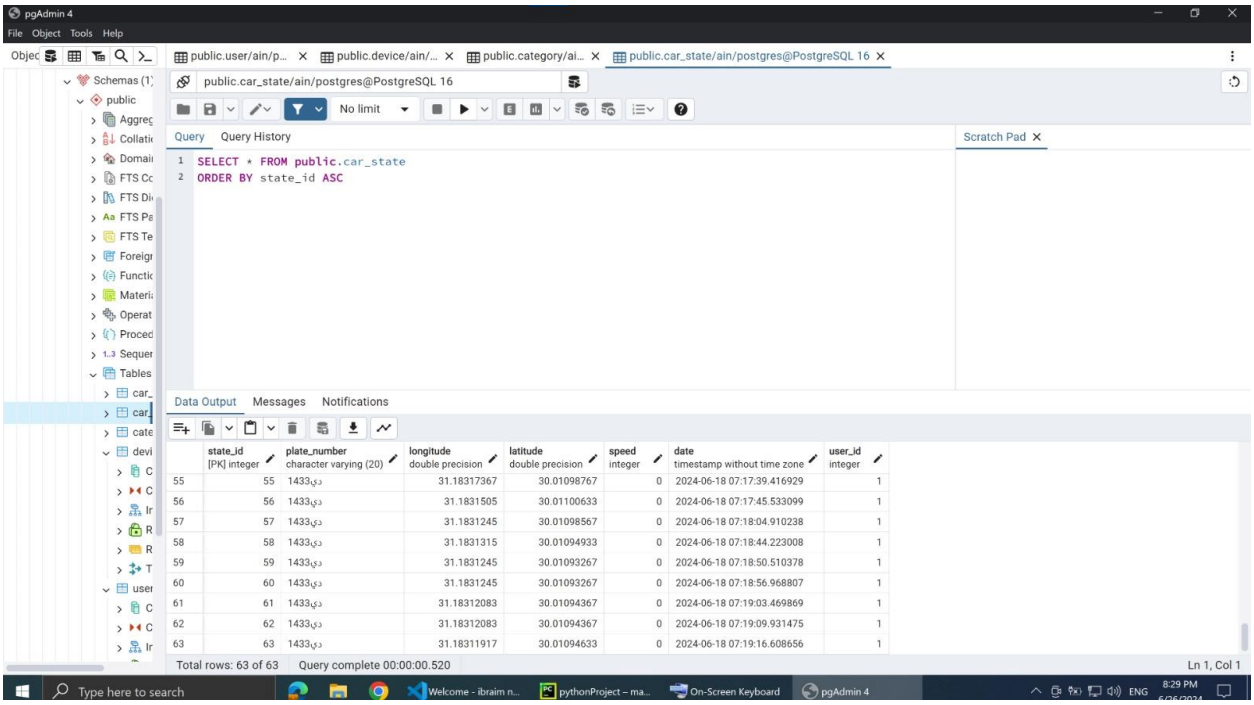


Figure 4.7: State of Car

Chapter Five: Run the Application

Welcome to the visual journey through our application! In this chapter, we will take you on a captivating exploration of the user interface and functionality of our application, accompanied by a series of screenshots and detailed explanations.

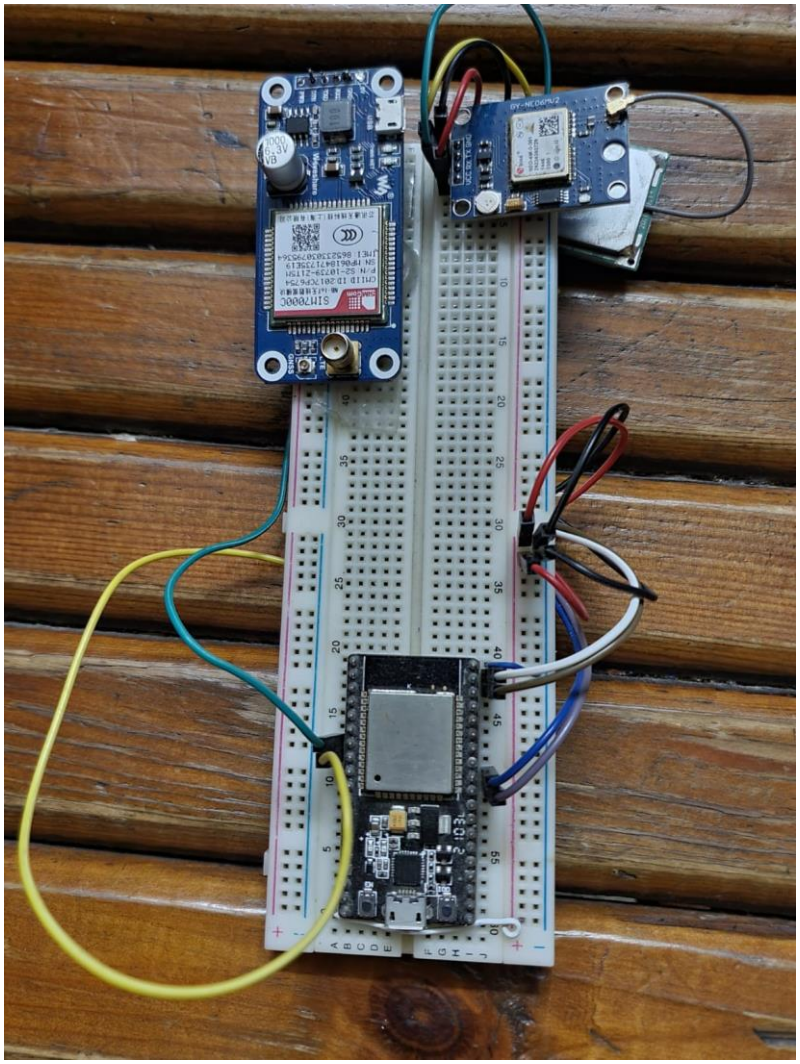
These screenshots serve as a window into the world of our application, providing a firsthand visual representation of its features, design, and user experience. By immersing yourself in these visuals, you will gain a deeper understanding of the unique aspects and capabilities that set our application apart.

The purpose of presenting these screenshots is to engage readers, such as project stakeholders, investors, or evaluators, in a compelling and informative manner. By combining visuals with explanations, we aim to create a rich and immersive experience that will enable you to grasp the core components, interactions, and user journey within the application.

Organized in a logical sequence, these screenshots will guide you through the various sections and features of the application, allowing you to explore its structure and layout. We have carefully curated this visual narrative to ensure a seamless and intuitive flow, making it easier for you to navigate through the different aspects of our application.

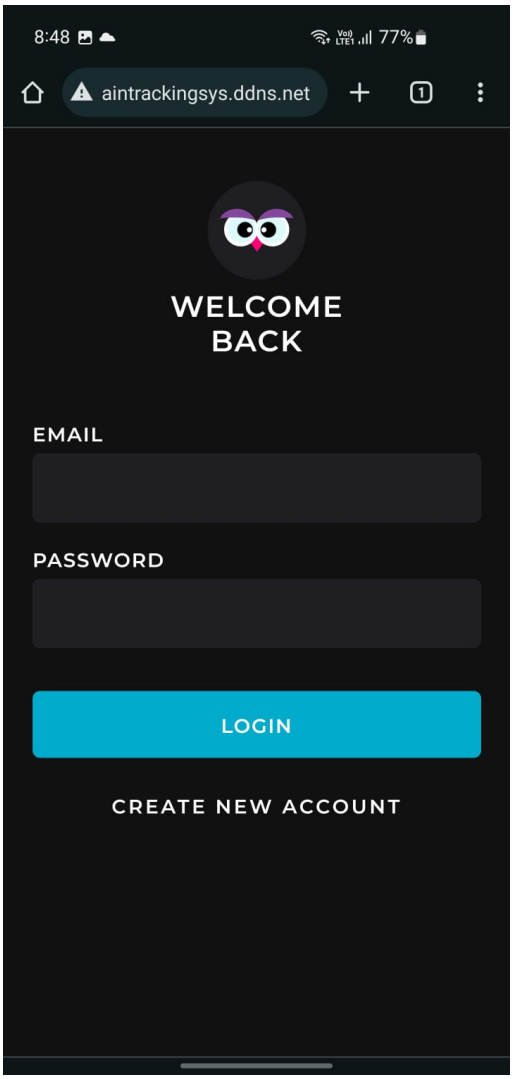
While these screenshots accurately represent the state of the application at the time of document creation, it's important to note that subsequent iterations may have introduced minor variations or enhancements. However, the essence and core functionality depicted in these visuals remain true to the overall concept and design of our application.

We encourage you to fully immerse yourself in the screenshots and their accompanying explanations. They provide a unique opportunity to experience the aesthetics, features, and user-centered design that define our application. Should you have any questions or require further clarification, please do not hesitate to reach out to us.



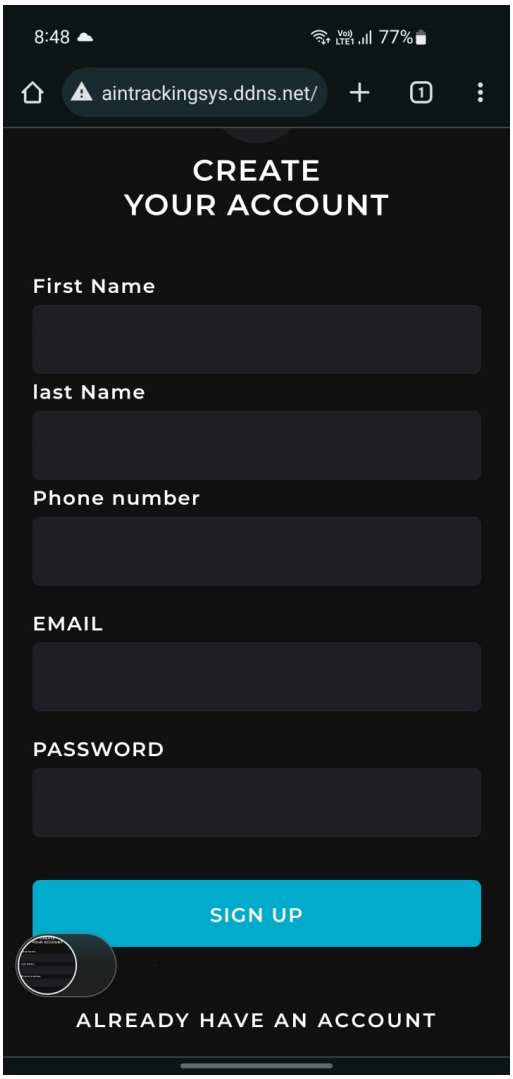
In this system, when a vehicle is detected moving beyond a predefined area, the ESP32 module sends a signal to capture the GPS location using the NEO-6M GPS module. The location data is then transmitted via the SIM7000C module to a remote server using AT commands. This server processes the data and updates the vehicle's location on a real-time tracking interface accessible through the mobile application.

If the system detects any unauthorized movement or deviation from the allowed route, it immediately sends an alert to the registered mobile application. The alert includes the vehicle's current location and the time of the detected activity.

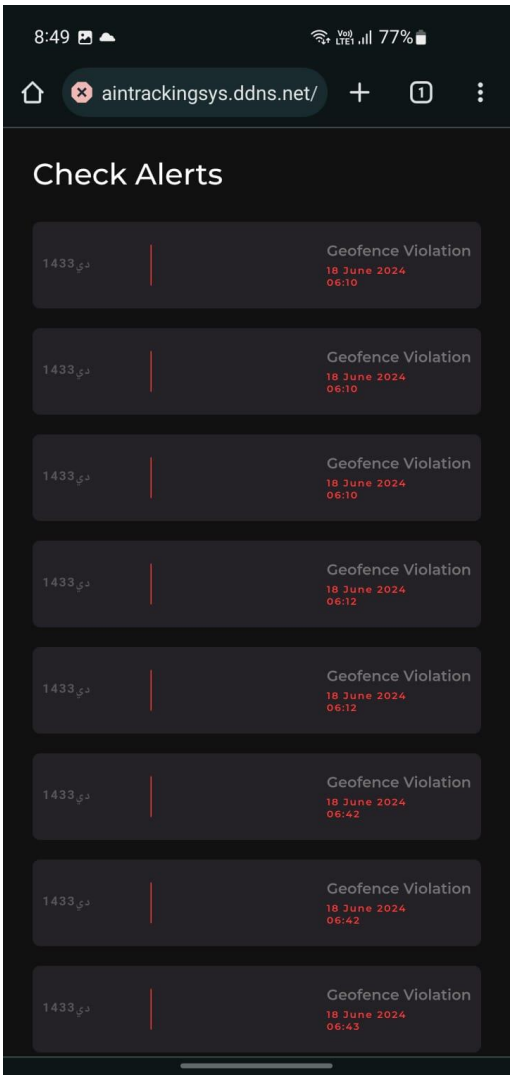


The login page features a customized user interface designed for ease of use and intuitive navigation. Users are prompted to enter their credentials to access the system. If a user does not have an account, they can easily redirect to the account creation page by clicking a clearly labeled link.

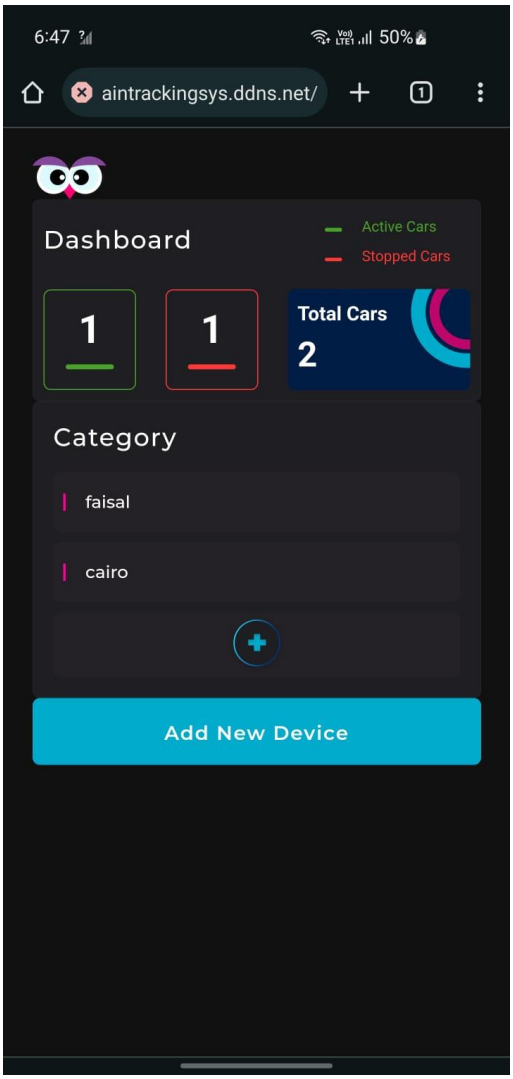
The login page ensures secure authentication using JSON Web Tokens (JWT) to protect user data and maintain session integrity. Upon successful login, users can access various functionalities of the system.

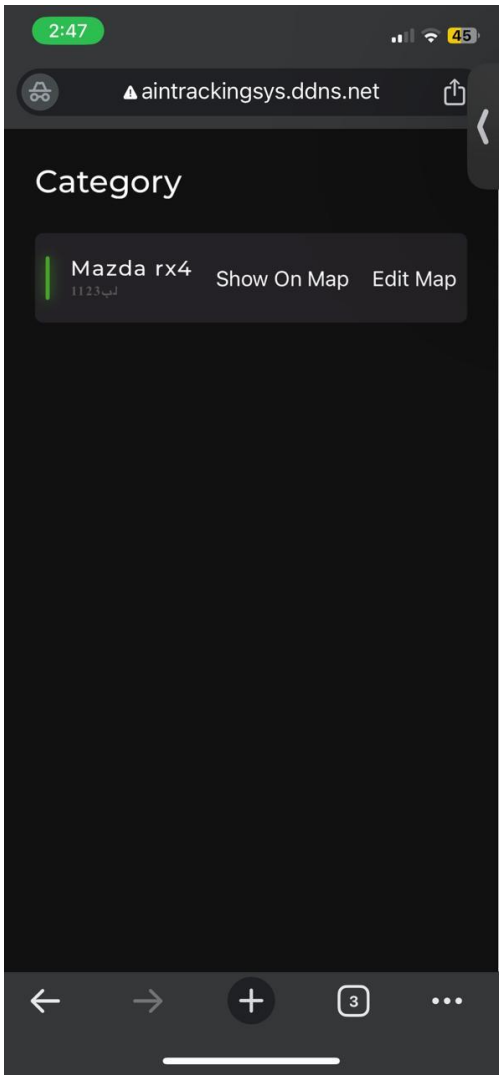


Users are required to fill in all the fields to create an account. The form validates the entered data, ensuring that all required information is provided and meets the necessary format and security standards. Once the form is completed and submitted, the user's information is securely processed and stored, allowing them to proceed with logging into the system.

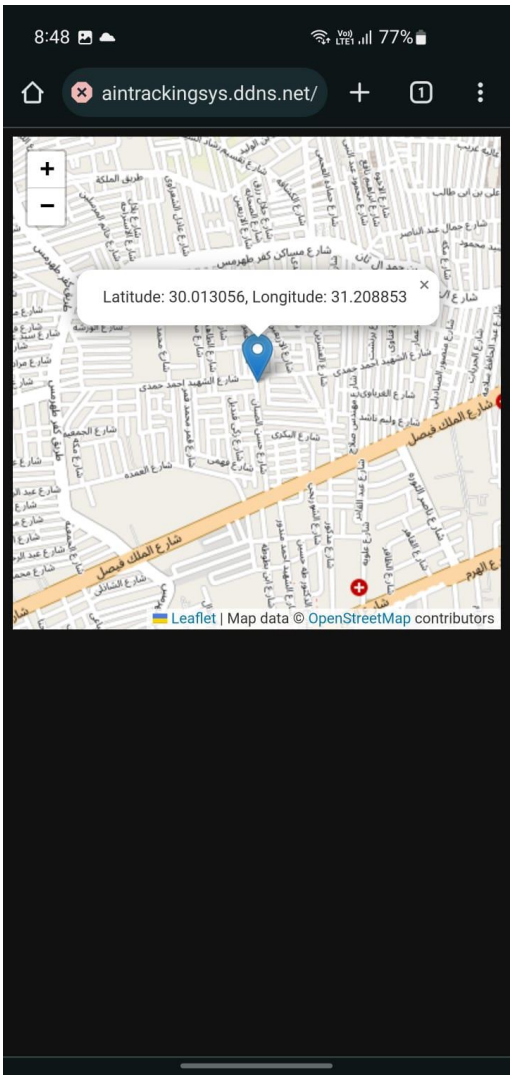


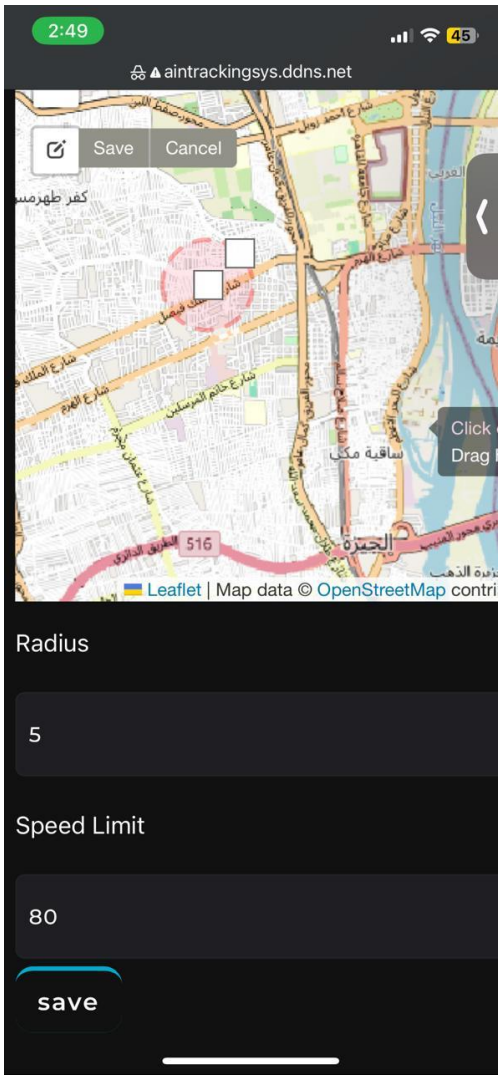
Here are the alerts generated by the car equipped with our device, which are divided into two categories: Geofence Violation and Speed Violation. When the vehicle exits the predefined zone set by the user, a geofence violation alert is triggered to notify the user of any unauthorized movement beyond the allowed geographic area. Similarly, when the vehicle exceeds the speed limit specified by the user, a speed violation alert is raised, helping the user monitor and control the vehicle's speed to promote safe driving practices and prevent potential accidents. These alerts ensure that stakeholders are promptly informed of any violations, allowing them to take necessary actions to maintain security and safety.





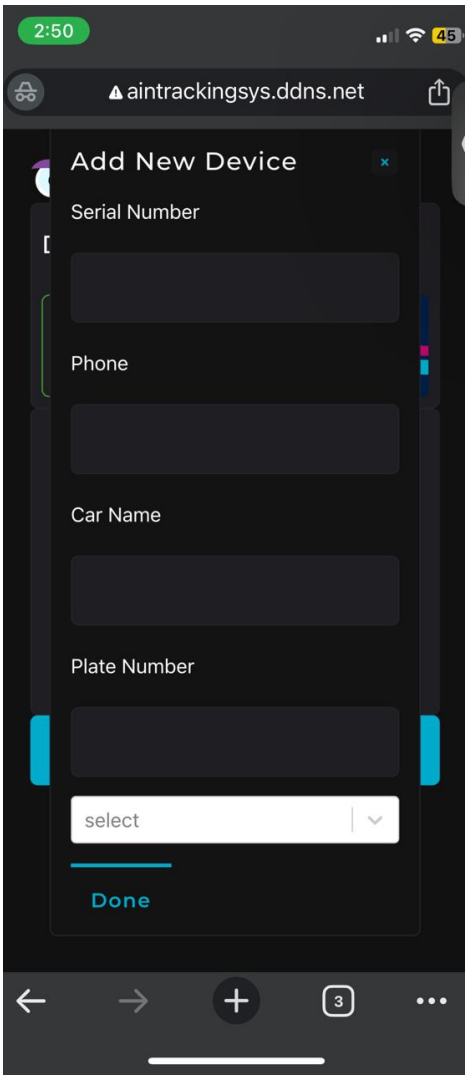
In this screen, the user can add a new category. Additionally, there is a list of existing categories, and when a category is selected, it displays the cars on which devices are installed and have been added to this category by the user.





On the left screen, it shows a specific car with its geographical coordinates on the map, which we use to detect if a geographical violation has occurred or not.

On the right screen, we set the geographical zone using a circle radius on a specific location and also set the speed limit



In this screen, the user can add a new device by entering the device serial number, phone number, car name, and plate number. This information links the device to a specific car, enabling tracking and monitoring functionalities

Chapter Six: Conclusion and Future Work

6.1. Conclusion

In our project, we have developed an advanced vehicle monitoring and security system that integrates multiple hardware components with sophisticated software functionalities. This system is designed to enhance vehicle tracking, speed monitoring, and geo-fencing capabilities using IoT technologies. Below are the key achievements and outcomes of our project:

Vehicle Monitoring and Security:

Our system effectively monitors vehicle activities, including real-time GPS tracking, speed detection, and geo-fencing. It ensures comprehensive vehicle security and management through continuous data collection and analysis.

Integration of Hardware and Software:

By integrating ESP32 microcontroller, NEO-6M GPS module, TP4056 battery charger, boost converter, and SIM7000C cellular module, our system achieves robust performance and seamless communication across various components.

Real-time Alerts and Notifications:

The system promptly alerts users about speed violations and geofence breaches through cellular communication. Notifications are sent to designated users via a web application, ensuring immediate awareness and action.

Scalability and Adaptability:

Designed with scalability in mind, our system can accommodate additional functionalities and scale to support a larger fleet or diverse vehicle types. It provides a flexible framework for future enhancements and upgrades.

Conclusion:

Our vehicle monitoring and security system represents a significant advancement in IoT-based solutions for fleet management and vehicle security. By leveraging advanced hardware components and intelligent software algorithms, we have successfully developed a robust, scalable, and effective solution for monitoring vehicle activities, ensuring compliance with speed limits, and enhancing overall fleet security. This project sets a foundation for further innovations in vehicle tracking systems and contributes to the evolution of smart transportation technologies.

6.2. Future Work:

1. **Microservices Architecture with Docker and Kubernetes:** Transition the monolithic application architecture to microservices using Docker containers orchestrated with Kubernetes (K8s). This approach enhances scalability, fault isolation, and deployment flexibility.
2. **Cloud Hosting on AWS:** Host the application infrastructure on Amazon Web Services (AWS) cloud platform for improved reliability, scalability, and global accessibility. Utilize AWS services such as EC2 for compute, S3 for storage, and RDS for database management.
3. **Integration of Battery Management System (BMS):** Incorporate a sophisticated Battery Management System (BMS) to monitor and manage the health, performance, and charging of vehicle batteries. This enhances battery efficiency, longevity, and overall system reliability.
4. **Enhanced Component Selection:** Upgrade to more advanced and reliable hardware components that offer better accuracy, durability, and performance under varying environmental conditions. This ensures consistent and dependable operation of the monitoring system.
5. **Advanced Analytics and Predictive Maintenance:** Implement machine learning algorithms to analyze vehicle data trends, predict maintenance needs, and optimize fleet management strategies. Use historical data to forecast potential issues and proactively schedule maintenance tasks.
6. **User Experience Improvements:** Enhance the mobile application interface with intuitive dashboards, real-time data visualization, and customizable alerts. Improve user interaction with the system through user-friendly features and proactive notifications.

7. **Security and Compliance Enhancements:** Strengthen data encryption, access controls, and compliance measures to ensure data privacy and regulatory compliance (e.g., GDPR, CCPA). Implement secure communication protocols and periodic security audits.

REFERENCES

[1] GPS tracking GSM based vehicle theft alert and engine locking system using Arduino by Dr. D. Ramanaidu, P. Haripriya, K. L. Saranya, P. Pavan Kumar, M. Sandeep, M. Manohar (link)

[2] Vehicle tracking system and theft detection by Mr. Saif Iqlas, Mr. Owais Ahmed, Mr. Mohammed Ahmed Khan, Dr. Shilpa Mangshetty (link)

[3] Cloud-Based Vehicle Tracking System by Ali Mustafa, Mohammed I. A al-Nouman, Osama A. Awad, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq. Contact: alimastafa43@gmail.com, {m.aalnouman, usamaawad}@coie-nahrain.edu.iq. Received: 19/10/2019, Accepted: 17/11/2019 (link)

[4] A Smart Real-Time Tracking System Using GSM/GPRS Technologies by Ali Mustafa, Mohammed I. A al-Nouman, and Osama A. Awad from College of Information Engineering, Al-Nahrain University, Baghdad, Iraq. (link)

Useful Links for the components:

[1] https://www.waveshare.com/wiki/SIM7000E_NB-IoT_HAT

[2] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[3] https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

[4] https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf

[5] https://www.ti.com/lit/ds/symlink/tps61045.pdf?ts=1719368537504&ref_url=https%253A%252F%252Fwww.google.com%252F

الملخص

يستهدف هذا المشروع تلبية الحاجة إلى نظام فعال لتتبع السيارات باستخدام نظام تحديد المواقع العالمي (**GPS**)، مما يسمح لمستخدم واحد بمراقبة عدة سيارات عبر تطبيق ويب. يستخدم النظام وحدات **GPS** مثبتة في السيارات، والتي ترسل بيانات مثل الموقع (خط الطول وخط العرض) والسرعة كل 10 ثوانٍ إلى وحدة **ESP32 WiFi** ثم إلى بطاقة **SIM**. يمكن للمستخدمين تتبع السيارات على الخريطة، وتحديد مناطق جغرافية (أسوار جغرافية)، وتلقي تنبيهات في حالة خروج السيارة من المنطقة المحددة أو تجاوزها للحد الأقصى للسرعة المحدد. بالإضافة إلى ذلك، يصنف التطبيق موقع السيارة حسب المدينة، مثل القاهرة أو الإسكندرية، بناءً على إحداثياتها. تم تطوير النظام باستخدام **Flask** لمعالجة حالات **API**، **PostgreSQL** لقاعدة البيانات، ودمجها مع واجهة أمامية **React**، مما يضمن المراقبة والتنبيه في الوقت الفعلي. تشير النتائج إلى أن هذا النظام فعال للغاية للتطبيقات مثل مراقبة حافلات المدارس أو السماح للآباء بتتبع استخدام أبنائهم للسيارات. في الختام، يوفر هذا المشروع حلاً موثوقًا لتتبع السيارات في الوقت الفعلي، وتحديد المناطق الجغرافية، ومراقبة السرعة، مما يعزز السلامة والأمان.