Project Report On

"SGGS VEHICLE TRACKER"

By

VAIBHAV NIVRUTTI LANJEWAR

(2021BIT023)

ROHAN RAVINDRA NACHANE

(2021BIT025)

Under the guidance of

Dr. ARATI MANJRAMKAR

Department of Information Technology

Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded

May 2025

CERTIFICATE

This is to certify that this project report entitled **“SGGS VEHICLE TRACKER”** submitted to Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded is a Bonafide record of work done by Vaibhav Nivrutti Lanjewar and Rohan Ravindra Nachane under my supervision from **15st of Decemeber 2024** to 1**3rd of May 2025.**

Dr. Arati Manjramkar
Project Guide

Dr. G. K. Pakle
Head of Department
Information Technology

Place: SGGSIE&T, Nanded
Date: 13 May 2025

DECLARATION BY STUDENT

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We are aware that if any part of the report is found to be plagiarized, I/we are shall take full responsibility for it.

Vaibhav Nivrutti Lanjewar
2021BIT023
Rohan Ravindra Nachane
2021BIT025

Place: SGGSIE&T, Nanded
Date: 13 May 2025

ACKNOWLEDGMENT

It gives me immense pleasure to express my appreciation to all those who helped and guided me throughout this project.

I would like to extend my heartfelt gratitude to **Dr. Arati Manjaramkar** for her valuable guidance, continuous support, and encouragement, which played a crucial role in the successful completion of this work.

I am also thankful to **Dr. G. P. Pakle**, Head of the Department of Information Technology, for his consistent motivation, and to **Dr. Manesh Kokare**, Director of SGGSIE&T, Nanded, for providing a supportive academic environment.

I would also like to sincerely thank **Prof. Rupsingh Mathwale**, Project Coordinator, for his valuable inputs, cooperation, and timely assistance throughout the dissertation phase.

ABSTRACT

The SGGS Vehicle Tracking System is a real-time GPS-based application developed to streamline and enhance the transportation experience for students and staff at Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T). By integrating embedded hardware, cloud computing, and web-based technologies, this system offers an efficient and cost-effective solution for tracking institutional vehicles.

The core hardware components include the A9G GPS + GSM module, which captures the vehicle's geographical location (latitude and longitude), and the ESP8266 Wi-Fi module, which transmits the location data to the Firebase Realtime Database. Firebase is also utilized for secure user authentication via Google Sign-In and Email/Password login options.

The web-based frontend is built using HTML, CSS, JavaScript, Tailwind CSS, Bootstrap, and Leaflet.js. The user dashboard includes:

A real-time interactive map showing the current vehicle location.

Live updates of user and bus positions, remaining distance, and Estimated Time of Arrival (ETA).A toggle feature to enable or disable Call/SMS alerts, which are activated through the ESP module when the bus approaches within a defined radius (e.g., 2 km).

This system improves transportation reliability by reducing uncertainties around arrival times, enabling better time management, minimizing unnecessary fuel usage, and ensuring timely arrival for academic sessions.

By combining low-cost hardware with cloud-based storage and a real-time, interactive user interface, the SGGS Vehicle Tracking System provides a robust, scalable, and user-friendly solution to modern transportation challenges.

Table of Content

Table of Figures

CHAPTER 1 : INTRODUCTION

In today's fast-paced world, the need for efficient and reliable transportation systems has become increasingly important, especially in educational institutions where punctuality is crucial. At Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), students and staff often face challenges such as unpredictable bus schedules, delays, and a lack of real-time updates. These issues lead to loss of valuable time, missed lectures, and increased stress among commuters. In response to these challenges, the development of a real-time vehicle tracking system becomes not only relevant but essential.

The SGGS Vehicle Tracking System is a GPS and IoT-based project aimed at resolving daily commute problems by providing live tracking of college buses. Using an A9G GPS module, the system captures the exact latitude and longitude of the bus in real-time. This data is then transmitted via the ESP8266 Wi-Fi module to the Firebase Realtime Database, which acts as the central cloud platform for storage and retrieval.

The frontend of the application, developed using HTML, CSS, JavaScript, Tailwind CSS, Bootstrap, and Leaflet.js, offers an interactive and responsive user interface. Students and staff can log in securely using Google or Email/Password authentication and view the current location of the vehicle on a live map. The system also displays important details such as user and bus positions, distance between them, and Estimated Time of Arrival (ETA).

Furthermore, to enhance user experience and safety, a toggle button is provided in the interface to enable or disable Call/SMS alerts. These alerts are sent via the ESP module using a SIM-enabled connection when the bus is within a certain range, helping users prepare in advance and avoid missing their ride.

Unlike conventional GPS devices that often show only static coordinates, this system delivers real-time, continuous updates on the vehicle's movement and translates geographic coordinates into meaningful visual insights via the map. While the primary use case is academic transport, this architecture can be easily extended to other domains such as fleet management, logistics, school buses, and emergency services.

The SGGS Vehicle Tracking System thus provides a low-cost, scalable, and cloud-integrated solution to improve time efficiency, reduce fuel wastage, and ensure timely class attendance, making it a valuable innovation for the SGGS campus community.

1.1 Objective

The primary objective of the Real-Time Vehicle Tracking System is to design and implement a robust, user-friendly, and scalable solution that enables continuous monitoring of college buses in real-time. This system specifically targets the need for timely and accurate information about bus movements to assist students and staff in planning their daily commute more effectively.

The system aims to:

- Utilize GPS technology to accurately capture the live geographical coordinates (latitude and longitude) of each bus.
- Use the ESP8266 Wi-Fi module to efficiently transmit this location data to Firebase Realtime Database.
- Retrieve and render real-time location data on an interactive map using Leaflet.js, ensuring smooth visual tracking.
- Authenticate users through Google and Email/Password logins to maintain secure access to the dashboard.
- Calculate and display essential information such as the current location of the vehicle, the user’s location, distance between them, and estimated time of arrival (ETA).
- Provide intelligent alert mechanisms through pop-up messages and audio notifications when the bus approaches the user’s location.
- Allow users to enable or disable Call/SMS alerts triggered via a SIM-enabled ESP module to enhance convenience and safety.
- Enhance time management, reduce waiting periods, and ensure optimal class attendance while conserving fuel and transportation resources.

1.2 Problem Statement

In many colleges, students and staff often face difficulties due to the lack of a real-time system to track the movement of college buses. This leads to uncertainty about arrival times, unnecessary waiting, missed transport, and overall inconvenience in daily commuting. Traditional methods such as fixed schedules or manual calls to drivers are inefficient and unreliable in dynamic traffic conditions. The Real-Time Vehicle Tracking System addresses these issues by providing a smart solution to monitor the live location of buses using GPS technology. The system displays the bus's position on an interactive map and alerts users when the bus is approaching their location. This ensures timely information, reduces waiting time, and enhances the safety and efficiency of college transportation.

1.3 Literature Review

Real-time vehicle tracking systems have become increasingly essential in modern transportation, with various technologies being explored to improve their efficiency, accuracy, and usability. The use of GPS (Global Positioning System) lies at the core of most tracking solutions, offering continuous access to time and location data through satellite-based navigation.

In the study by Miss. A. S. Mamdyala GPS tracking system was proposed for monitoring vehicle or employee movement using GPS and satellite communication. The paper highlighted that such systems are widely used in civil, military, and commercial domains to ensure efficient supervision and operational safety. The authors pointed out that by reducing dependency on manual communication and enabling real-time visibility, GPS-based tracking systems help improve logistics, reduce fuel usage, and enhance scheduling accuracy[1].

Another related project by Mohd Hakimi Bin Zohari and Mohd Fiqri Bin Mohd Nazri focused on building a GPS-based vehicle tracking system using Arduino MEGA, Ublox NEO-6M GPS module, and SIM900A GSM module. Their system was able to send the vehicle's coordinates via SMS, allowing users to locate their vehicle through Google Maps links. While the system functioned effectively outdoors, it struggled with accuracy in obstructed environments like indoors or under a roof. The authors suggested improvements such as using higher-quality GPS modules for better satellite connectivity and more accurate data[2].

In contrast to traditional GSM-based tracking systems, our project integrates a real-time cloud database (Firebase) and web technologies (Leaflet.js) to display vehicle location live on a web interface. Unlike systems that only provide coordinates via SMS, our approach delivers continuous location updates visually, along with browser-based alerts (pop-up and audio) when the vehicle is near the user. This provides a more interactive and real-time user experience suitable for college bus tracking, and can be extended to logistics, public transport, and emergency services. The integration of cloud and frontend technologies eliminates the need for backend servers or GSM communication, making the system lightweight, scalable, and cost-effective.

CHAPTER 2 : PROJECT PLAN

2.1 Methodologies Of Problem Solving

To develop the SGGS Real-Time Vehicle Tracking System, we adopted a structured methodology inspired by the Software Development Life Cycle (SDLC). This helped ensure a systematic, efficient, and goal-driven development process. The project was carried out in four main phases: Analysis, Design, Development, and Evaluation.

Figure 1: Waterfall Model

2.1.1 Analysis Phase

- Problem Identification: Recognized the challenges faced by students and staff due to uncertain bus timings, inefficient commuting, and lack of real-time visibility.
- Requirement Gathering: Collected technical and functional requirements by surveying potential users (students, drivers, and college staff).
- Feasibility Study: Evaluated hardware availability (ESP32, GPS, SIM modules), Firebase capabilities, and technical constraints of integrating real-time systems.

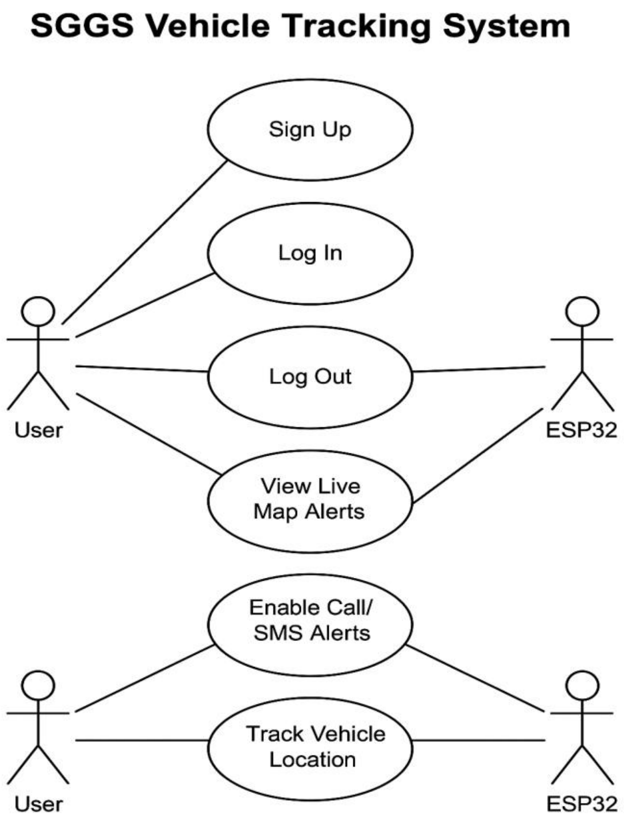

Figure 2: Use Case Diagram

2.1.2 Design Phase

- System Architecture Design: Created a layered architecture diagram showing interactions between hardware, Firebase, and the frontend dashboard.
- Database Schema: Designed data structures in Firebase to store user details, vehicle location, session status, and SMS/call preferences.
- User Interface Mockups: Designed initial wireframes of the dashboard, signup/login pages, and live tracking interface using HTML and CSS frameworks.
- Hardware Wiring Layout: Planned connections between GPS, SIM, and ESP modules for real-time data transmission.

2.1.3. Development Phase

- **Frontend Implementation**:
    - Built responsive web pages for signup, login, and user dashboard using HTML, CSS, JavaScript, Bootstrap, and Tailwind CSS.
    - Integrated Leaflet.js to display real-time vehicle positions on the map.
- **Firebase Integration**:
    - Used Firebase Authentication for secure login/signup.
    - Set up Firebase Realtime Database for storing and retrieving live GPS data, user states, and alert toggles.
- **Hardware Coding**:
    - Programmed ESP32 to fetch GPS coordinates and send them to Firebase.
    - Coded logic to enable/disable SMS/Call alerts based on toggle status and distance thresholds.

2.1.4 Evaluation Phase

- Functional Testing: Verified user authentication, GPS data updates, toggle functionality, and logout behavior.
- Hardware Testing: Conducted field tests to ensure GPS accuracy and proper message/call triggering via SIM module.
- Performance Monitoring: Monitored real-time data push and pull latency to maintain smooth user experience.
- User Feedback: Collected feedback from initial testers and refined UI/UX based on their inputs.

CHATPER 3: SYSTEM ARCHITECTURE

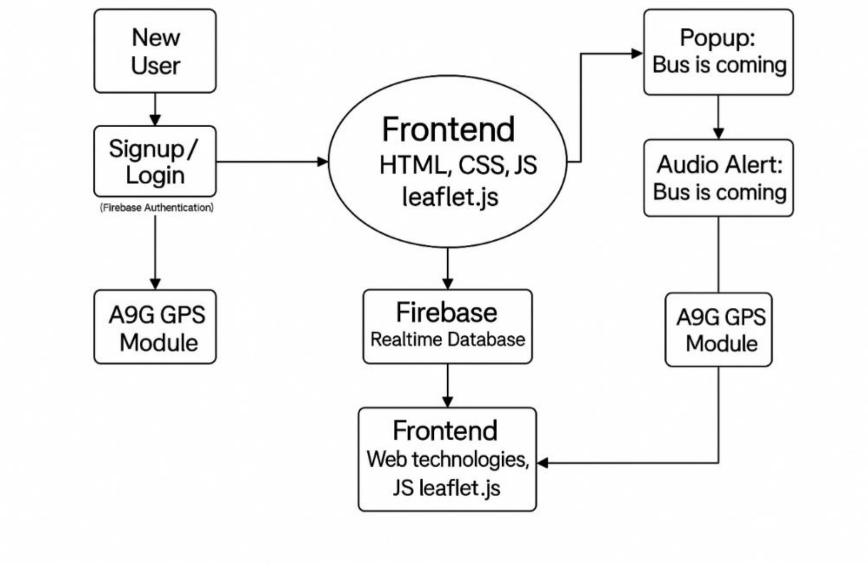

Figure 3:System Architecture

The system architecture of the Real-Time Vehicle Tracking System is designed to ensure seamless communication between hardware, cloud services, and the frontend interface. It comprises the integration of multiple components including GPS modules, Wi-Fi modules, Firebase services, and a web-based frontend to deliver accurate and real-time vehicle location information.

3.1 User Authentication and Access Control

The system begins with a user authentication process. Both new users and registered users are directed to the Signup/Login interface. Authentication is handled via Firebase Authentication, which supports email/password login as well as third-party sign-ins such as Google. Upon successful login, the user gains access to the main frontend dashboard where real-time tracking features are available.

3.2 Frontend Interface

The frontend is developed using HTML, CSS, JavaScript, and Leaflet.js. It acts as the primary interface for end-users to interact with the system. Once logged in, users can view:

- Their current location on a map.
- The live location of the vehicle(s).
- Alerts and notifications about vehicle arrival.
- A popup message indicating that the bus is approaching.
- An audio alert to draw the user's attention.

3.3 Hardware Layer

The system uses the A9G GPS module to capture real-time GPS coordinates (latitude and longitude) of the vehicle. This data is transmitted via the ESP8266 Wi-Fi module, which sends it directly to the Firebase Realtime Database. An optional A3G GPS module is also considered in the architecture to handle future upgrades such as SIM-based SMS or call alerts.

3.4 Firebase Realtime Database

Firebase serves as the backend database for storing and synchronizing data. It receives live GPS data from the ESP8266 Wi-Fi module and updates the vehicle's position in real time. The database supports live two-way communication, which ensures that any changes in vehicle location are instantly reflected on the frontend interface without delay.

3.5 Real-Time Interaction and Alert Mechanism

The frontend continuously listens for changes in the Firebase database. It calculates the distance between the user and the vehicle.This improves user experience by minimizing waiting time and enhancing situational awareness.The overall system architecture ensures low-latency communication, secure user access, and reliable vehicle tracking. It is scalable, cost-effective, and suitable for deployment in educational institutions. The integration of hardware components with cloud infrastructure and a responsive frontend interface creates an efficient and user-friendly transportation tracking solution.

3.6. Process Flow Diagrams

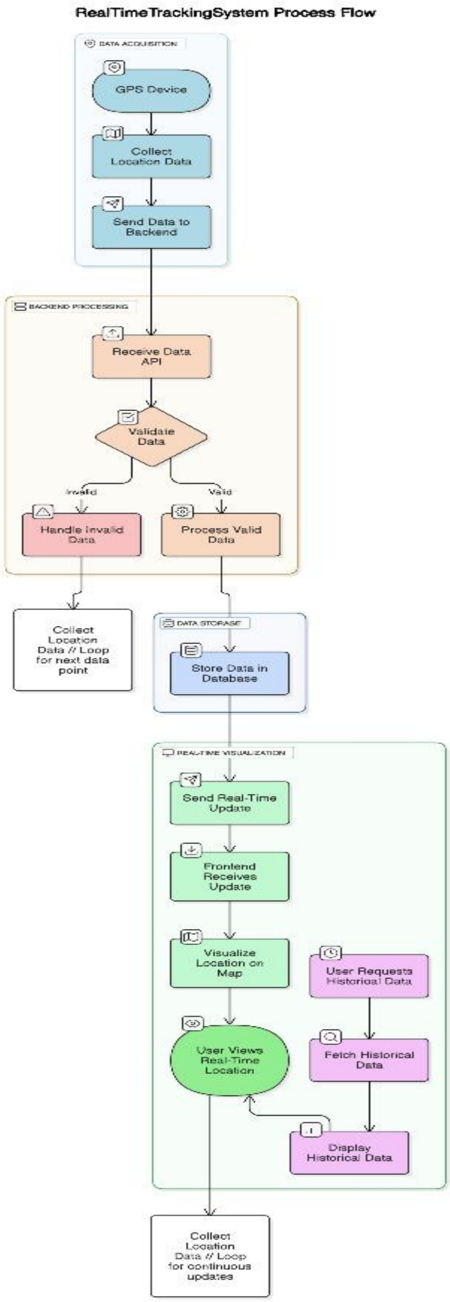

Figure 4:Process Flow Diagram

The process flow diagram represents the sequential operational flow of the *Real-Time Vehicle Tracking System*. It captures the interaction between users and system components from application initiation to alert delivery. This diagram illustrates how real-time location tracking, data fetching, and alert mechanisms work cohesively to ensure seamless vehicle monitoring for users.

1. Login/Signup : The process initiates when the user accesses the system via a web browser. This triggers the application load, presenting the user with the login/signup interface.

2. Authentication via Firebase : Users either sign up or log in using Firebase Authentication. This ensures secure access and user-specific session handling.

3.Dashboard with Map : Once authenticated, the user is directed to the dashboard where a dynamic map is displayed. This map serves as the main interface for tracking.

4.Parallel-Data-Fetching : Two processes are initiated simultaneously: Vehicle Location Fetch: The system retrieves the vehicle's real-time GPS coordinates from Firebase, which are updated by the ESP module connected to the GPS hardware.User Location Access: The system requests access to the user's current location for proximity calculations.

5. Map Update with Markers : The map interface is updated with markers representing both the user's and the vehicle's location. This provides visual real-time feedback.

6. Proximity Check Is the Bus Nearby: The system evaluates the distance between the user and the bus. If the bus is within a defined threshold, the system proceeds to notify the user.

7. Bus Nearby Trigger Actions : If proximity conditions are met, the system executes two user notificationPopup Alert: A textual popup indicates that the bus is approaching.Audio Alert: An auditory notification is played to catch the user's attention, ensuring inclusivity and real-time awareness.

8. Bus Not Nearby Wait for Updates: If the bus is not nearby, the system enters a wait state and continues polling for updated location data to reassess the condition.

9. User Views Live Location :Throughout the session, users can continuously view updated vehicle movement and their own location on the map.

10.Logout:The user may exit the system by logging out, thereby terminating their active session.

3.7 Sequence Diagram:

Sequence Diagram gives significance of the sequential execution of functionalities of system.

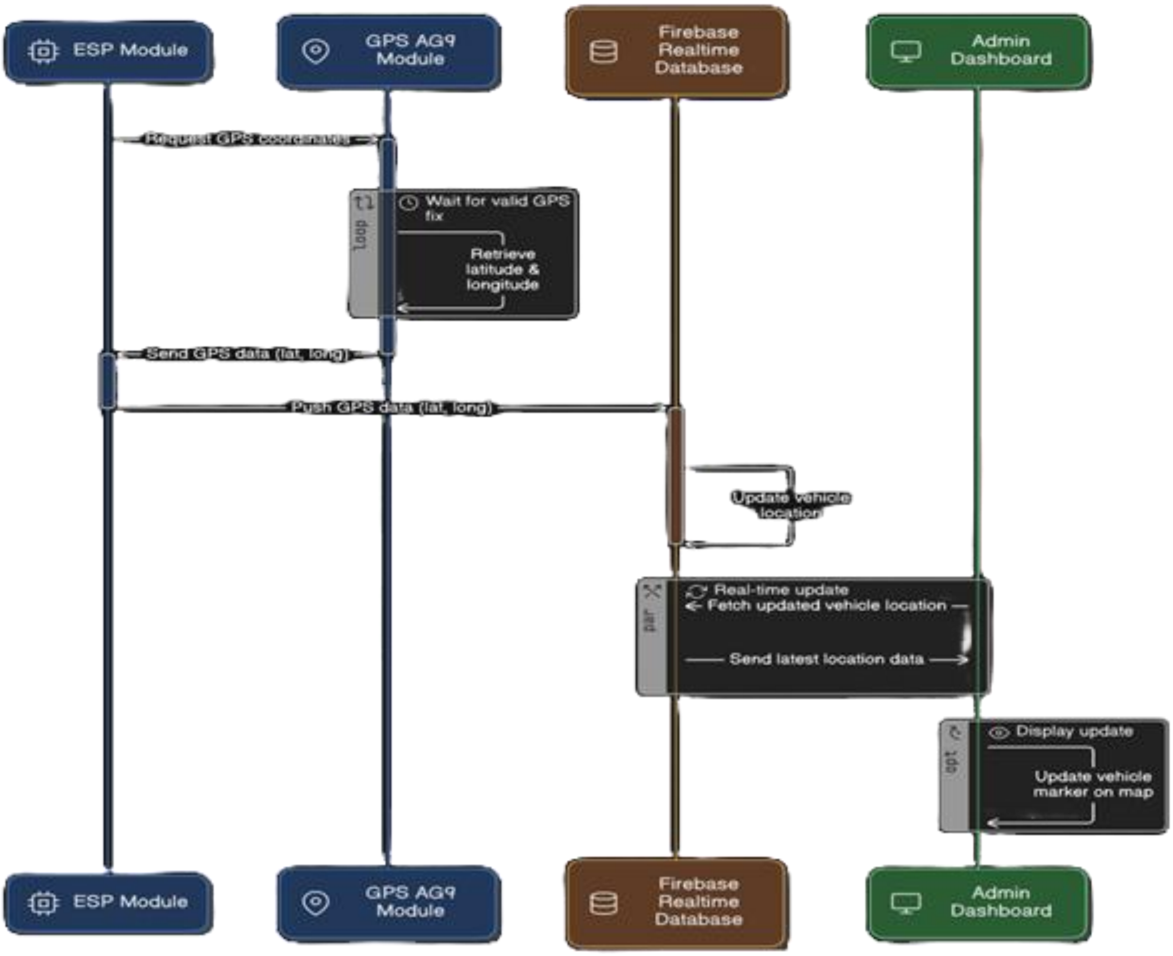

Figure 5:Sequence Diagram

CHAPTER 4: PRESENT WORK

Under this section, the implementation methods, hardware components used, and technologies integrated to build the Real-Time Vehicle Tracking System have been discussed. This system uses GPS hardware for location tracking, Firebase for cloud-based real-time data handling, and a frontend interface developed using modern web technologies. The application allows users to sign up or log in using Firebase Authentication and then view real-time bus and user locations on a map via Leaflet.js. Additionally, the system alerts users with a popup and audio when the bus is near their current location.

4.1 Hardware Components

4.1.1 A9G GPS Module

Figure 6:A9G GPS Module

The A9G is a compact GSM/GPRS and GPS module that is highly suited for mobile IoT and real-time vehicle tracking applications. It captures precise latitude and longitude coordinates from GPS satellites and provides this data via serial communication. The module supports SMS, GPRS data transfer, and voice communication. In this system, it constantly monitors the bus's location and communicates with the ESP8266 module for data transmission.

4.1.2 ESP8266 Wi-Fi Module

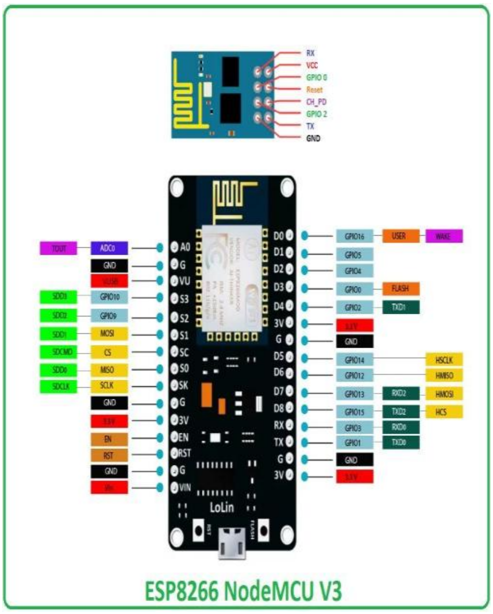

Figure 7:ESP8266 Wifi Module

The ESP8266 is a highly popular, low-cost microcontroller with built-in Wi-Fi. It acts as the communication bridge between the A9G GPS module and the Firebase Realtime Database. The ESP8266 receives serial data (coordinates) from the A9G module through its GPIO pins (typically D7 for TX and D8 for RX) and uploads this data to Firebase over Wi-Fi. It ensures uninterrupted and real-time updates of vehicle positions to the frontend interface.

4.1.3 Motor Driver

Figure 8:Motor Driver

The motor driver is used in prototype implementations to control the movement of the vehicle model. It receives control signals from the microcontroller and regulates power to the connected DC motors. This helps simulate the bus movement in a controlled testing environment. It is particularly useful when demonstrating the system using miniature vehicle models.

4.1.4 Battery / Adapter

A rechargeable battery pack or adapter is used to power the entire hardware setup. It supplies the necessary voltage and current to the A9G GPS module, ESP8266, and motor driver. Proper voltage regulation is essential to ensure stable operation and avoid hardware damage. For portable operation, lithium-ion or lithium-polymer batteries are commonly used.

4.2 Hardware Connections Overview

- **ESP8266 (NodeMCU)**:
  - D7 (TX) → A9G RX
  - D8 (RX) → A9G TX
  - VIN → Battery +5V
  - GND → Battery GND
- **A9G GPS Module**:
  - TX → D8 of ESP8266
  - RX → D7 of ESP8266
  - VCC → Battery +5V
  - GND → Battery GND
- **Motor Driver**:
  - Connected to GPIO pins of ESP8266 for control signals
  - Powered by the same battery source
- **Battery/Adapter**:

- 5V output connected in parallel to ESP8266, A9G, and Motor Driver

- Ensure proper common GND connection across all components

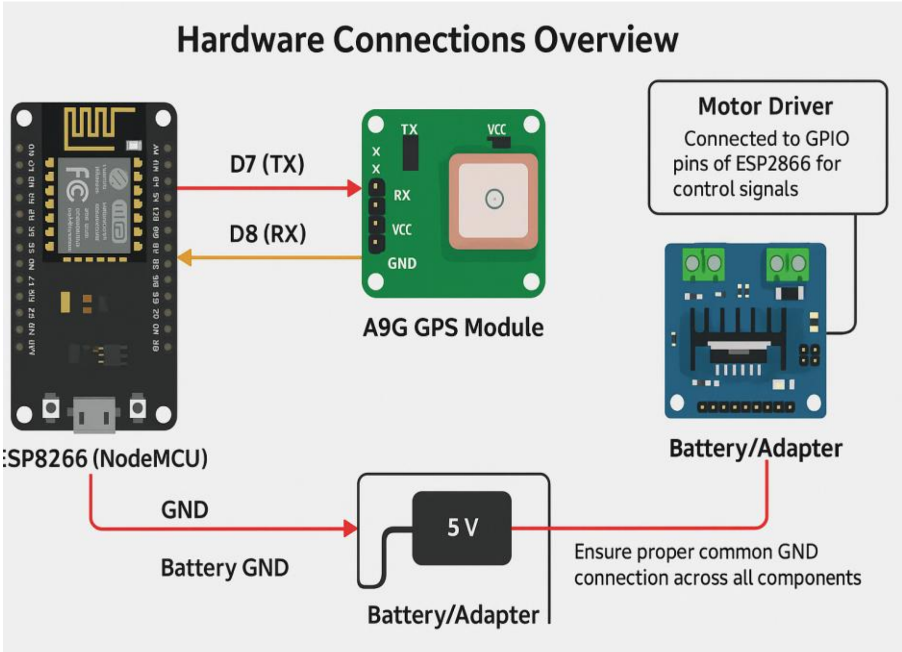

Figure 9:Hardware Connection

This setup ensures real-time location tracking, data transmission to the cloud, and physical simulation of bus movement in demo models. Proper voltage regulation and grounding are key to seamless hardware performance.

CHAPTER 5: TECHNOLOGY STACK

5.1 HTML

HTML forms the foundation of the frontend. It structures all the visual and interactive elements the user interacts with, including:

Signup/Login forms

- Dashboard layout (map container, info panels, toggle switches)
- Navbar with navigational links (About, FAQ, Contact)
- Footer section

In your system, HTML ensures that every interactive section—whether it's a form input, a real-time map display, or a toggle button—is semantically and accessibly represented. Proper use of HTML tags improves accessibility, SEO, and maintainability. Forms also support integration with Firebase Authentication by organizing user input for processing.

5.2 CSS (Cascading Style Sheets)

CSS handles the visual presentation and layout of the structured content created with HTML. It enables customization of colors, spacing, fonts, and responsiveness across devices.

5.3 JavaScript

In the SGGS Vehicle Tracking System project, JavaScript plays a crucial role in building the frontend of the web application. It is responsible for making the interface interactive and dynamic. JavaScript is used to fetch real-time location data from Firebase and update the vehicle's position on the map using the Leaflet.js library. It also handles user interactions, such as toggling the call/SMS alert feature, displaying live distance and ETA, and rendering vehicle and user markers on the map. Overall, JavaScript ensures smooth communication between the backend and frontend, providing a seamless real-time tracking experience to the users.

5.4Tailwind-CSS

Tailwind CSS is a utility-first CSS framework that enables rapid UI development.Instead of writing custom CSS for every element, Tailwind allows you to use predefined utility classes directly in HTML. This approach makes your styling modular, readable, and maintainable, especially in a complex interface like the vehicle tracking dashboard. It also simplifies responsive design, allowing you to define how elements behave at different screen sizes (e.g., mobile vs desktop).

5.5.Bootstrap

Bootstrap is a component-based frontend framework that provides a wide range of pre-designed UI elements such as navbars, forms, buttons, and modals. It complements Tailwind CSS by offering a structured grid system for building responsive layouts quickly and efficiently. With its ready-to-use styled components, Bootstrap enables fast UI prototyping, making development smoother and faster. Additionally, it ensures cross-browser compatibility for various interface elements, helping maintain a consistent look and feel across different devices and browsers.

5.6.Leaflet.js

Leaflet.js is a lightweight, open-source JavaScript library used to create interactive maps on web pages. In the SGGS Vehicle Tracking System, Leaflet is used to render a live map that displays real-time data. It shows the current location of the vehicle by fetching GPS coordinates from Firebase and also displays the user’s location using the browser’s Geolocation API. The library supports custom markers that update dynamically as new location data is received. Additionally, it enables smooth map interactions such as zooming, panning, and live marker movement, providing users with an intuitive and responsive tracking experience.

5.7 Firebase (Frontend Integration)

Firebase Authentication

- Handles user signup and login using email, password, and optionally Google Sign-In.
- Ensures secure session management, enabling each user to access the dashboard after login.
- Supports role-based access, distinguishing between students (users) and drivers (admins).

Firebase Realtime Database

- Acts as a bridge between hardware and frontend.
- ESP8266 uploads GPS data (latitude and longitude) to Firebase.
- JavaScript listens for updates and reflects them instantly on the map and vehicle info cards.
- Also stores the state of the call/SMS enable switch, which determines whether the ESP module should initiate SMS/call alerts.

CHAPTER 6 : IMPLEMENTATION

6.1 User Signup

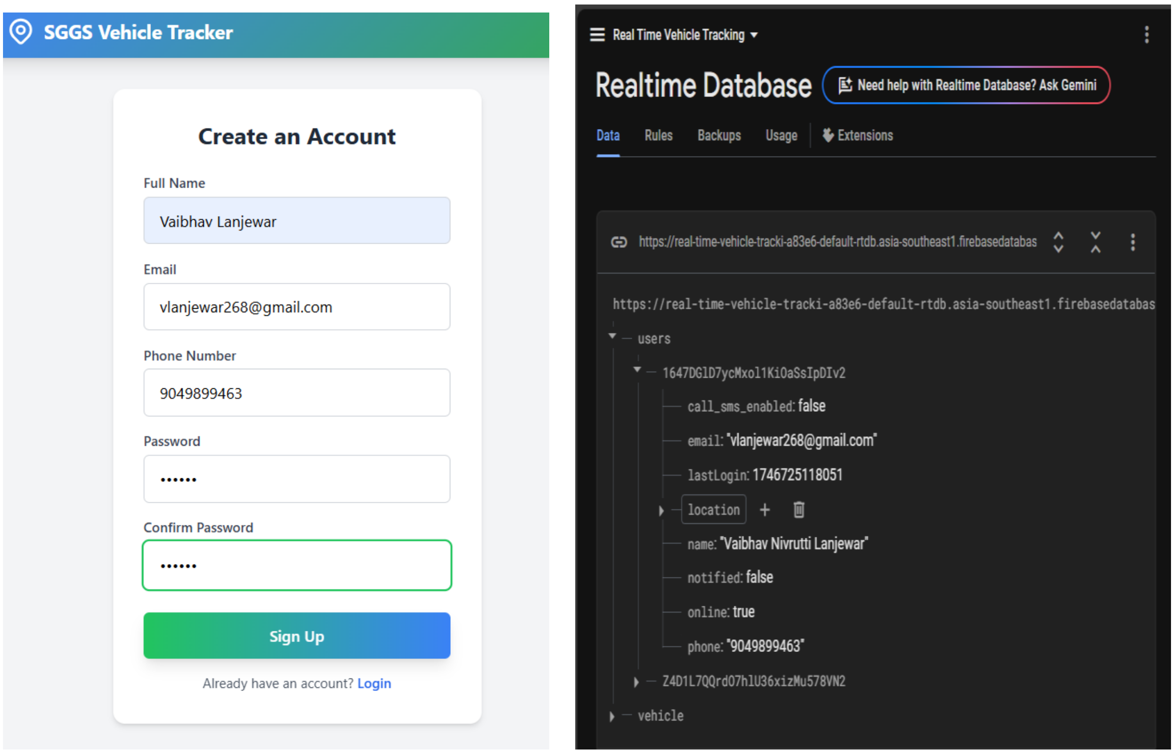

Figure 10:Signup with Firebase Realtime Database

When a new user accesses the SGGS Vehicle Tracking System for the first time, they are directed to the Signup page, which is built using HTML, Tailwind CSS, and JavaScript. The signup form prompts the user to provide the following details:

- **Email** – serves as the unique identifier for the user's account
- **Password** – stored securely and used for authentication
- **Name** – the full name of the user
- **Phone Number** – used for enabling call or SMS-based notifications from the hardware module

Firebase Authentication – Account Creation

Once the user submits the form, the provided information is sent to Firebase Authentication, which performs the following tasks:

- Creates a unique and secure user account based on the email and password

- Handles session management and maintains login states
- Ensures security using Google’s backend infrastructure

Firebase Authentication automatically verifies the validity of the email and securely stores the credentials, preventing duplicate or unauthorized account creation.

Firebase Realtime Database – Storing Additional User Information

In addition to account creation, the system also stores user-specific data in the Firebase Realtime Database. A new entry is added under the /users node using the user’s unique ID (UID) generated by Firebase. The data structure includes the following keys:

- email – the user’s email address
- name – the full name of the user
- phone – user’s mobile number
- online – a status flag indicating if the user is currently active
- lastLogin – a timestamp recording the user's most recent login time
- location – used to store live location coordinates, updated in real time
- call_sms_enabled – a boolean value indicating whether the user has enabled or disabled hardware-triggered call/SMS alerts
- notified – tracks whether the user has already been notified for a particular event

These fields enable the application to manage personalized settings for each user, control hardware communication, and track login activity in real time.

Redirect to Login Page

After the signup process is completed successfully and the user’s data is stored in the Realtime Database, the user is automatically redirected to the **Login page**. From there, they can log in using their registered email and password to access their personalized dashboard and live vehicle tracking features.

6.2. User Login

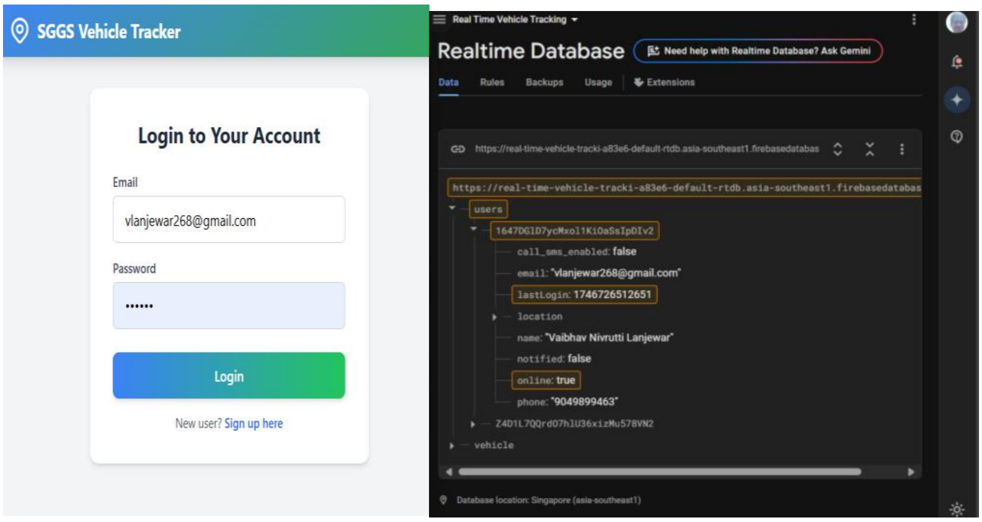

Figure 11:User Sinup

Once a user has successfully signed up, they can log in to the SGGS Vehicle Tracking System using the email and password provided during registration. The login interface is designed with a clean layout using HTML, Tailwind CSS, and JavaScript, ensuring a responsive and intuitive experience across all devices.

Firebase Authentication – Login Handling

When the user submits the login form:

- The credentials are authenticated using Firebase Authentication.
- If the credentials are valid, Firebase initiates a secure session and logs the user in.
- Firebase automatically manages the session state and persists login information until the user logs out.

Firebase Realtime Database – Updating User Status

Upon successful login, the system performs a write operation to the Firebase Realtime Database to update the user's current status. Specifically, the following fields under the /users/<UID> node are modified:

- online: true – this flag indicates that the user is currently active in the system
- lastLogin: <timestamp> – this field is updated with the exact login time (in Unix epoch format)

This real-time update allows the system to:

- Track which users are currently active

- Display live user activity on the admin panel if required
- Enable real-time personalization or alerts
- Support interaction with hardware components based on the user's presence

Access to Dashboard
After the session is authenticated and the database is updated, the user is redirected to their Dashboard. From here, they can:

- View a real-time map with their own and vehicle locations
- Toggle SMS/Call alerts via the hardware
- Access navigation components like About, FAQ, and Contact
- Log out securely, which will reset their online status to false

6.3.Vehicle Tracking Dashboard

6.3.1 User Interface Overview

The dashboard is built using HTML, CSS, JavaScript, Tailwind CSS, and Bootstrap for a responsive, mobile-friendly design. It includes the following key components:

Navbar:

- Web Title**:** Displays the project name "SGGS Vehicle Tracker".

- Dropdown Menu (Menu) Contains :
    - About Us
    - Why Sggs Vehicle Tracker
    - FAQs
    - Contact Page

- Call/SMS Toggle**:** A functional switch to enable or disable alerts via SMS or Call.When toggled, it updates the value of call_sms_enabled in Firebase (true/false).This value is read in real time by the ESP32 microcontroller to decide if alerts should be triggered when the bus is near a user (e.g., within 300 meters).

- Logout Button: Signs the user out and updates their online status in Firebase to false.

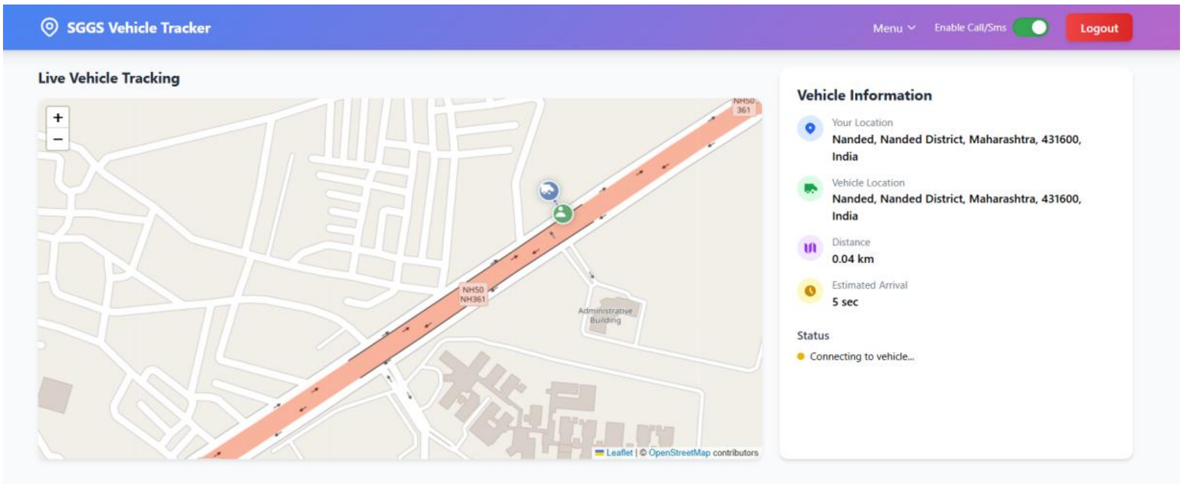

Figure 12:Sggs vehicle tracker Dashboard

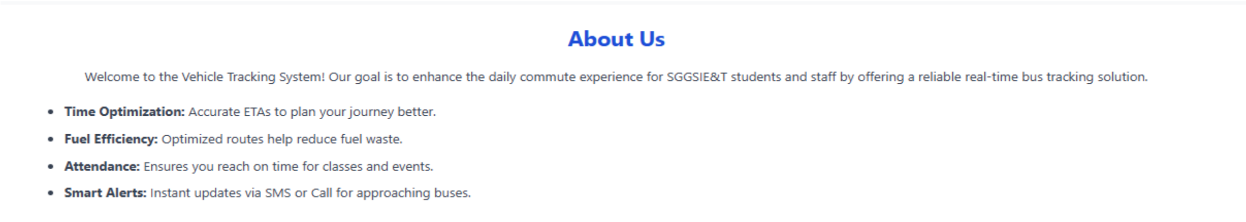

Figure 13:About us

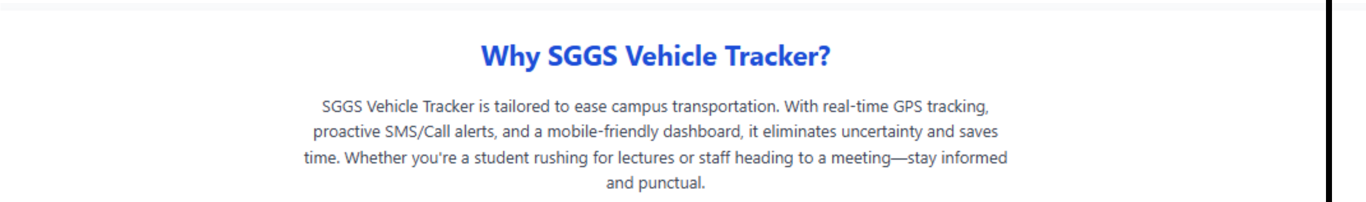

Figure 14:why sggs vehicle tracker

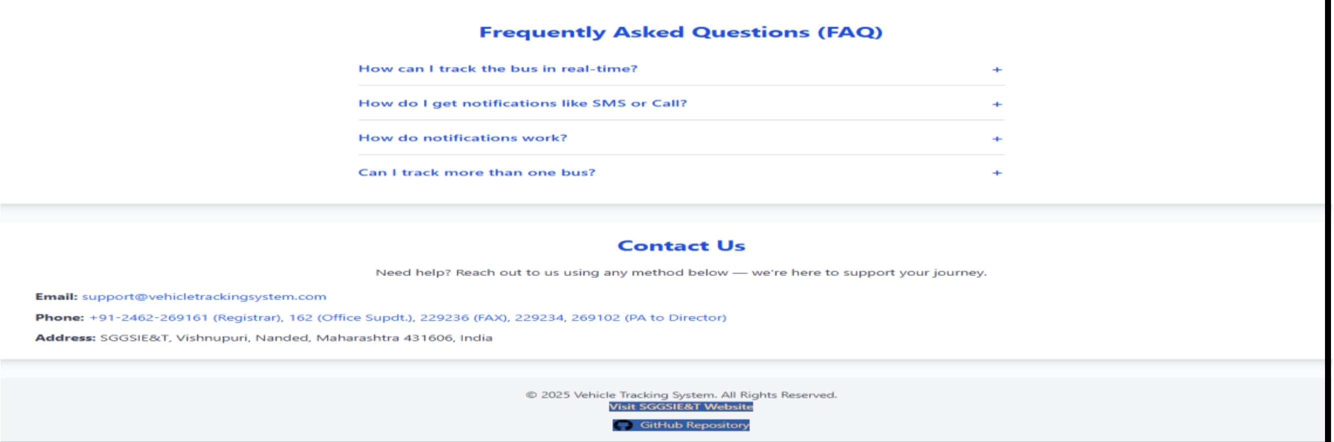

Figure 15:FAQs , Contact us and Footer

6.3.2 Live Vehicle Tracking Map

- Leaflet.js Library**:** A powerful JavaScript library is used to render the real-time interactive map.
- Map Display:
  - Shows User Location and Vehicle Location with distinct markers.
  - Pulls coordinates (latitude, longitude) for both from the Firebase Realtime Database.
- Distance and ETA Calculation**:**Using the Haversine formula or similar geolocation functions in JS, it calculates:

6.3.3. Vehicle Information Panel

This section on the right-hand side provides:

- **Your Location:** Shows the user's current address converted using reverse geocoding APIs (if used).
- **Vehicle Location:** Displays the current bus location fetched from Firebase (updated from ESP/GPS module).
- **Distance:** Real-time distance between the vehicle and user location.
- **Estimated Arrival:** Time estimation based on proximity and movement.
- **Status:** Shows real-time vehicle connectivity status like "Connecting to vehicle…" or "Vehicle is nearby".

6.3.4 Backend Integration with Firebase

The frontend dashboard continuously communicates with the Firebase Realtime Database to fetch and update data:

- **Reading Live Data:**Fetches user-specific data (email, name, location, call_sms_enabled.Reads GPS data pushed by the ESP32 module (vehicle latitude, longitude).

- **Updating Data:**
    - On logout: Updates online: false.
    - On toggling call/SMS: Updates call_sms_enabled: true/false.

6.3.5 Hardware Interaction (ESP + GPS Module)

- The ESP32 microcontroller is connected to a GPS module (like AG9) to constantly fetch the vehicle’s real-time location.
- This data is sent to Firebase under the vehicle’s node (e.g., /vehicle/latitude, /vehicle/longitude).
- It also listens for the call_sms_enabled value: If enabled and vehicle is near a user’s location (within 300m), it triggers a predefined SMS or Call via a SIM module connected to ESP.

CHAPTER 7: RESULT AND DISCUSSION

The Real-Time Vehicle Tracking System was successfully designed, developed, and tested to fulfill its core objective: providing live tracking of campus vehicles for students and staff. The system efficiently integrates hardware and software components, including A9G GPS modules, ESP8266 Wi-Fi modules, Firebase Realtime Database, and a web-based frontend using HTML, CSS, JavaScript, and Leaflet.js for dynamic map rendering.

7.1. Results

- **Live Location Tracking**: The system accurately displayed real-time vehicle positions on a map interface with updates reflected within 1–2 seconds of GPS data push to Firebase.
- **Authentication**: Secure login and registration were implemented using Firebase Authentication with support for both email/password and Google sign-in.
- **User Interaction**: Users could view their current location, track the bus in real-time, and receive popup or audio alerts when the bus was nearby.
- **Accuracy**: The GPS module provided location accuracy within 5–10 meters during field testing on the SGGS campus.
- **Responsiveness**: The frontend interface loaded quickly and responded well on both desktops and mobile browsers.

7.2. Discussion

The system meets its goal of providing timely, reliable information to users, thereby reducing waiting time and improving transport planning. It was observed that users found the alert system particularly useful in determining when to leave for their bus stop, which aligns with the project's intent to reduce time wastage and improve daily commutes.

Some challenges encountered during implementation included:

- Intermittent network issues affecting real-time data upload from the ESP8266.
- Initial synchronization delays between the GPS module and Firebase during power restarts.

CONCLUSION

The SGGS Vehicle Tracking System represents a successful integration of hardware and software technologies to solve a specific, real-world problem—streamlining the campus transportation system for Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), Nanded. This project was conceptualized and executed with a strong focus on improving daily commutes for students and faculty members by providing real-time visibility into bus movement and location.

At its core, the system combines GPS hardware (AG9 GPS module) and ESP32 microcontroller to collect accurate location data from moving campus vehicles. This data is transmitted over the internet to Firebase Realtime Database, ensuring a consistent and scalable cloud backend. On the frontend, a responsive web dashboard built using HTML, CSS, JavaScript, Tailwind CSS, Bootstrap, and Leaflet.js provides users with an intuitive map interface, allowing them to track live vehicle movement, calculate distance and ETA, and receive status updates dynamically.

The implementation of Firebase Authentication ensures secure login for users, while real-time session tracking updates the online status of users and facilitates personalized access to vehicle tracking features. Additionally, the call/SMS toggle feature, synced directly with Firebase, allows users to opt-in for proximity-based alerts. These alerts are processed by the ESP32 using distance calculations and triggered through the GSM/SIM interface, ensuring that users are notified when their bus is near—minimizing wait times and preventing missed rides.

This system has shown to enhance operational efficiency and user satisfaction by:

- Providing real-time tracking and estimated arrival times
- Reducing wait time and fuel consumption
- Improving punctuality for academic schedules
- Allowing administrative oversight of vehicle usage and movement

In conclusion, the SGGS Vehicle Tracking System not only fulfills its intended goal of improving the commuting experience within the campus but also exemplifies the application of IoT, cloud, and web technologies in solving practical transportation issues. It lays a strong foundation for future enhancements and extensions, potentially contributing to smart campus infrastructure and broader smart mobility initiatives.

FUTURE SCOPE

The SGGS Vehicle Tracking System, although currently tailored for college bus monitoring, is built on a scalable and flexible architecture that can be extended well beyond the educational domain. Its modular hardware setup, cloud-based backend, and interactive frontend make it highly adaptable and relevant for a broad range of sectors where real-time vehicle tracking and monitoring are critical.

Potential Areas of Expansion:

- **School-Transportation**
  The system can be seamlessly integrated into school transport services to enhance student safety. Real-time location access and automated alerts can help parents and administrators ensure children's safe and timely commutes.
- **Public-Transport**
  Adapting the system for municipal transport services can improve schedule accuracy, reduce passenger wait times, and provide commuters with live tracking and ETA—contributing to a smarter public transport ecosystem.
- **Fleets-Management**
  For organizations managing employee transport or industrial logistics, the system can optimize routing, reduce idle time, and improve fuel usage. It supports effective fleet management by offering real-time status and location data of all company vehicles.
- **Emergency-Response-Vehicles**
  In critical services such as ambulances, police vehicles, and fire trucks, real-time tracking can significantly enhance emergency response efficiency. The system can be tailored to prioritize routing, reduce delays, and coordinate multi-vehicle responses in real time.
- **Logistics-Delivery-Services**
  Delivery-based industries can benefit from constant monitoring of delivery vans and trucks, ensuring timely arrivals and improving customer satisfaction through live tracking and estimated delivery times.

Environmental and Institutional Benefits:

In academic settings, the system continues to prove valuable by:

- Saving time for students and faculty
- Improving punctuality and reducing class disruption
- Promoting fuel efficiency and eco-friendly transport

- Reducing carbon emissions by minimizing idle time
- Data-driven resource management
- Fuel usage optimization
- Cost-effective transport planning

Future Enhancements:

1. **AI-Powered-ETA-Prediction**
   By integrating machine learning models, the system could predict arrival times more accurately based on traffic, historical data, and road conditions.
2. **Offline-GPS-Data-Sync**
   In areas with poor connectivity, the GPS module can temporarily store location data and sync with Firebase once back online, ensuring data continuity.
3. **Mobile-App-Integration**
   A mobile application can be developed for Android/iOS to offer push notifications, real-time alerts, and more accessible dashboard access for students and drivers.
4. **Multi-Vehicle-Tracking-Dashboard**
   Enhancing the dashboard to support multiple vehicles simultaneously will make it ideal for institutions with a larger fleet.
5. **Advanced-Analytics**
   Integration of dashboards that provide trip statistics, usage patterns, and heatmaps for route optimization and long-term planning.

In conclusion, the SGGS Vehicle Tracking System lays a solid foundation for smart, sustainable, and intelligent mobility solutions. With its potential to serve broader transportation needs, the system can be a stepping stone toward smart campus environments and even contribute to Smart City initiatives with minimal upgrade

REFERENCE

References:

[1] Mamdyal, Miss & Sandupatla, Miss & Saka, I & Kothawale, J & Shirashayad, V & Kazi, Kutubuddin. (2022). GPS Tracking System. 2. 2581-9429.

[2] Zohari, Mohd Hakimi & Zohari, Bin & Bin, Mohd & Nazri, Mohd. (2021). GPS Based Vehicle Tracking System. International Journal of Scientific & Technology Research. 10. 278-282.

[3] Arduino Official Documentation,
A9G GPS/GPRS Library.
https://docs.arduino.cc/libraries/a9gmod/

[4] Arduino Official Documentation,
ESP8266 AT Command Library.
https://docs.arduino.cc/libraries/arduinoespat/

[5] Firebase Documentation,
Firebase Authentication and Realtime Database.
https://firebase.google.com/docs/