International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Smart Route Monitor: A Cloud-Enabled Real-Time Bus Tracking Framework

DR.V.SHANMUGA PRIYA [1], ARUNDHADHI. S[2]

Assistant Professor[1], Student of B.Sc CS[2]

Sri Krishna Arts and Science College, Kuniyamuthur - Coimbatore shanmugapriyav@skasc.ac.in[1],

arundhadhis24bcs003@skasc.ac.in[2]

ABSTRACT:

The ***Bus Tracker Application*** is a real-time web-based system designed to modernize and simplify public bus transportation. It provides live bus locations, estimated arrival times, and route details through a cloud-hosted, mobile-friendly web interface. The project leverages modern technologies such as **ReactJS**, **Node.js**, **Firebase**, and the **Google Maps API** to ensure accessibility, scalability, and reliability.

The main objective of the system is to enhance the commuter experience by reducing waiting times and improving communication between passengers and transport authorities. The application eliminates uncertainty by allowing users to track the exact position of buses in real time, view routes, and receive notifications about delays or diversions. Hosted on **Netlify** with a backend supported by **Firebase Firestore**, the system ensures seamless synchronization and real-time updates. This paper explores the design, implementation, testing, and potential future scope of the Bus Tracker Application as a sustainable urban mobility solution

Keywords: Bus Tracking System, Public transportation, ReactJS, Node.js, Firebase, Google Map API, Cloud Hosting, Netlify, Firestore Database, Web Applications, Smart Mobility, Sustainable Transport.

1. INTRODUCTION:

Public transportation plays a vital role in connecting people across cities, helping reduce traffic congestion, pollution, and travel costs. However, one of the major challenges faced by commuters is the lack of real-time information about bus locations, arrival times, and route changes. Traditional systems still depend on static printed schedules or manual updates, which often lead to uncertainty and frustration among passengers. Many commuters are forced to wait for long periods at bus stops without knowing when their bus will arrive or if it has been delayed. This lack of transparency reduces trust in public transport and encourages people to rely on private vehicles, further worsening urban traffic and environmental pollution. Addressing these issues requires an innovative and technology-driven approach that provides accurate, up-to-date information to both passengers and transport authorities.

The ***Bus Tracker Application*** is designed to overcome these limitations by providing a digital platform for real-time bus tracking and communication. The system integrates GPS technology, cloud storage, and an interactive web interface to display the live location of buses, estimated arrival times, and route information. Users can search for specific buses by number, stop, or destination, and view their current status through an intuitive map-based interface. For administrators, the system offers tools to manage bus routes, driver assignments, and scheduling updates efficiently. Developed using ReactJS, Node.js, Firebase, and Google Maps API, the application ensures scalability, reliability, and accessibility across devices. By bridging the gap between commuters and transport operators, this project aims to enhance the efficiency of public transportation and contribute to the development of smarter, more sustainable cities.

2. OBJECTIVES OF THE TRACKER SYSTEM

The main objectives of the Bus Tracker Application are as follows:

- To enable passengers to track live bus locations through GPS-based monitoring.
- To minimize waiting times by showing accurate ETAs for buses.
- To allow users to search by route, destination, or bus number.
- To provide both map-based and text-based results for better accessibility.
- To give administrators tools to manage bus routes, schedules, and driver details.
- To design a mobile-friendly, cloud-hosted platform accessible anytime, anywhere.
- To build public trust through transparency and real-time information sharing.

The overall aim is to enhance commuter convenience while making public transport smarter, sustainable, and efficient.

3. WHAT IS THE SYSTEM DESIGN AND ARCHITECTURE USED HERE

The **Bus Tracker Application** has been designed using a **modular, layered architecture** that emphasizes real-time communication, data consistency, and scalability. The system is primarily composed of three interconnected components — the **User Module**, **Admin Module**, and **Arrival Module** — each responsible for specific operations while maintaining seamless

data exchange through a cloud-based backend.

The **User Module** acts as the front-facing interface for passengers. It enables users to search for buses by route number, stop name, or destination and visualize real-time bus positions on an interactive map. This module is built with **ReactJS**, providing a highly responsive interface that adjusts smoothly to devices of all sizes, from smartphones to desktops. It incorporates search and filter functionalities, allowing users to track their chosen routes and estimate arrival times accurately. The user interface is intentionally designed to be intuitive, ensuring accessibility even for first-time or non-technical users.

The **Admin Module** serves as the control center for transportation authorities. Through a secure login system, administrators can add or modify bus details, assign drivers, edit schedules, and manage multiple routes simultaneously. The module also allows performance tracking by displaying metrics such as route delays, bus availability, and GPS synchronization frequency. This centralization reduces manual workload and ensures that updates are reflected instantly across the entire system. Administrators can also issue notifications regarding diversions, breakdowns, or maintenance schedules, which are automatically pushed to passengers' interfaces in real time.

The **Arrival Module** operates in the background, integrating **GPS tracking devices** installed on buses with the central database. Each bus transmits its location coordinates periodically to the backend server, which then updates the live map view for users. This module plays a critical role in estimating the **Estimated Time of Arrival (ETA)** by comparing the bus's current speed and route progress. The real-time data stream is stored and managed using **Firebase Firestore**, a NoSQL cloud database known for its automatic synchronization and scalability. Firestore ensures that all clients—passenger devices and admin dashboards—receive live updates without manual refreshes, maintaining high consistency even when multiple users access the system simultaneously.

The application architecture follows a **client-server model**, where the frontend communicates with the backend through **RESTful APIs** built with **Node.js and Express**. The backend handles API requests, authentication, data validation, and communication with the Firestore database. This setup allows data to flow efficiently between modules while maintaining data integrity and reducing latency. The **Google Maps API** is integrated for route visualization, mapping bus movements, and rendering geolocation-based features such as zoom, route lines, and stop markers.

Deployment and hosting are achieved using **Netlify** for the frontend and **Firebase Hosting** for the backend and database services. This cloud-based deployment ensures 24/7 uptime, global accessibility, and the ability to scale dynamically with user traffic. Furthermore, by leveraging **asynchronous data updates** and **API-based architecture**, the system supports multiple concurrent users without performance degradation.

Overall, this modular design promotes reliability, reusability, and adaptability. Each module can be enhanced or modified independently without affecting the overall system, allowing easy integration of future features such as AI-based ETA prediction, multilingual support, and mobile app versions. The combination of **ReactJS**, **Node.js**, **Firebase**, and **Google Maps API** results in a robust, scalable, and efficient architecture that provides real-time tracking, efficient data management, and an enhanced commuter experience.

4. METHODOLOGY

The Bus Tracker Application was developed using the **Agile Software Development Life Cycle (SDLC)**. Each phase was implemented iteratively to ensure consistent testing and improvements.

4.1 Requirement Analysis

Research on Coimbatore's public transport system revealed the need for real-time data and digital convenience. Existing systems relied on printed schedules, lacking flexibility and accuracy. The goal was to replace this outdated setup with a digital, interactive platform.

4.2 System Development and Its Working

The **frontend** of the Bus Tracker Application was developed using **ReactJS** and **TailwindCSS**, which together create a dynamic, fast, and responsive user interface. **ReactJS** enables the system to update data in real time without reloading the page, ensuring a smooth experience for users tracking live bus locations. Each part of the interface—such as the map, route search, and schedule display—is built as an independent **React component**, making the code modular and easy to maintain. React's **virtual DOM** improves efficiency by updating only the sections that change, which keeps the application responsive even when handling large data sets.

TailwindCSS was used to enhance the design and responsiveness of the frontend. Its utility-based approach allows for quick styling and ensures consistent layouts across all screen sizes. This makes the interface clean, mobile-friendly, and easy to navigate. The combination of ReactJS and TailwindCSS ensures that users enjoy a visually appealing and functional experience across devices without the need for a separate mobile app.

The **backend** was developed using **Node.js** and **Express.js**, which together provide a strong and scalable server environment. **Node.js** handles multiple client requests simultaneously through its event-driven, non-blocking structure, ensuring fast data processing. **Express.js** was used to create RESTful APIs that connect the frontend with the database. These APIs handle core functions such as retrieving live bus positions, managing schedules, and updating route data. Express also ensures secure data handling through validation and error management.

For storing and synchronizing data, the project uses **Firebase Firestore**, a real-time NoSQL cloud database. Firestore stores key information such as bus details, GPS coordinates, and schedules, and updates the data instantly across all connected clients. Its **real-time listener** feature ensures that every change in bus position is immediately reflected on the user's screen without refreshing. Firestore organizes data into collections and documents, making it simple to scale and manage.

It also includes **Firebase Authentication** for security and **cloud hosting** for easy deployment.

Overall, the integration of **ReactJS, TailwindCSS, Node.js, Express,** and **Firebase Firestore** creates a stable and high-performing ecosystem. This architecture allows the Bus Tracker Application to deliver real-time tracking, smooth user experience, and reliable data management—making it a practical model for modern, technology-driven public transportation systems.

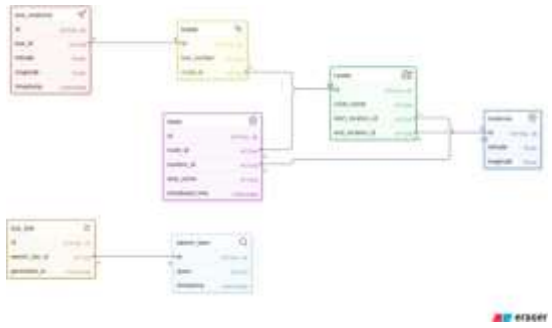

Fig: System Flow Diagram[1]

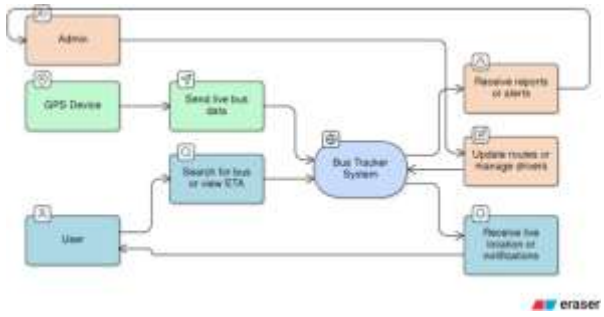

Fig: System Flow Diagram[2]

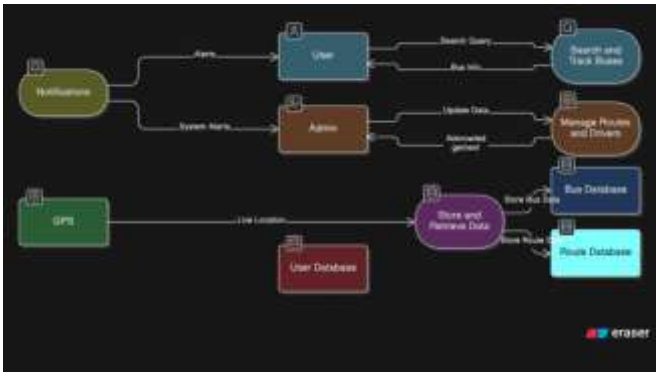

Fig: System Flow Diagram[3]

4.3 GPS and Cloud Integration

GPS modules installed in buses send coordinates to the database every few seconds. The application processes these updates to calculate ETAs and display live routes. The integration of **Google Maps API** ensures precise visualization of routes and stops.

4.4 Testing and Validation

Testing included **functional**, **system**, and **acceptance testing**. Real users tested the app on various devices and browsers to ensure reliability, usability, and real-time performance. The testing confirmed the app's ability to display live bus data accurately while handling concurrent users effectively.

5. RESULTS AND DISCUSSION

The development and deployment of the **Bus Tracker Application** led to substantial improvements in the overall efficiency and reliability of the public transport experience when compared to the traditional, manually operated system. The introduction of real-time GPS tracking and cloud integration has transformed how passengers interact with bus services. By providing continuous live updates on bus positions, routes, and arrival times, the system has significantly reduced passenger waiting periods at bus stops. Commuters can now plan their journeys more accurately, knowing exactly when their bus will arrive, which has increased convenience and reduced frustration. The ability to track buses in motion not only eliminates guesswork but also builds trust in public transport schedules, encouraging more people to rely on buses instead of personal vehicles.

From an administrative standpoint, the system has improved operational transparency and management efficiency. Transport authorities can remotely monitor the performance of each bus, identify delays, and make quick updates to schedules or routes whenever necessary. This instant data synchronization ensures that any modification made by an administrator is reflected immediately on the user interface, maintaining consistent information for all users. The platform's **cross-platform compatibility**—working equally well on smartphones, tablets, and desktops—has also made it highly accessible to a broad range of users without requiring any installation or software updates.

Another important achievement is the scalability of the architecture. Since the application is hosted entirely on cloud platforms such as Firebase and Netlify, it can easily expand to support additional routes, buses, and even other cities with minimal changes to its configuration. The use of asynchronous communication between the client and server helps maintain fast performance even under heavy usage. During user testing, students and daily commuters particularly appreciated the real-time tracking feature, noting that it helped them save time and better plan their travel routines. Administrators, on the other hand, highlighted the convenience of centralized control and automated data management, which reduced manual errors and administrative workload.

Overall, the implementation results demonstrate that the Bus Tracker Application successfully meets its objectives of improving efficiency, transparency, and user satisfaction. The system has proven capable of bridging the communication gap between commuters and transport authorities, modernizing the public transport experience in Coimbatore, and laying the groundwork for a smarter and more sustainable urban mobility solution.

6. ADVANTAGES AND LIMITATIONS

6.1 ADVANTAGES

Real-Time Tracking: The most important strength of the Bus Tracker Application is its ability to show live bus locations and provide accurate estimated arrival times using GPS integration. This real-time feature helps commuters make better travel decisions and reduces unnecessary waiting time at bus stops.

1. **User-Friendly Interface:** The system's interface is simple, clean, and easy to understand. Passengers of all age groups can use it comfortably without technical knowledge, which improves the overall user experience.
2. **Centralized Management:** Administrators can manage all transport- related data—such as bus routes, driver details, and schedules—from one unified platform. This reduces manual errors and improves coordination across different routes.
3. **Cloud-Based Accessibility:** Since the entire system is hosted on cloud platforms, users do not need to install any software. It can be accessed easily from smartphones, tablets, or desktop browsers anytime and anywhere, ensuring continuous service.
4. **Environmentally Friendly:** By improving the reliability and convenience of public transport, the application encourages more people to use buses instead of private vehicles. This helps reduce traffic congestion, fuel usage, and air pollution, supporting sustainable urban living.

6.2 LIMITATIONS:

1. **Internet Dependency:** The system requires a stable internet connection for real-time updates. Users in areas with weak connectivity may experience delayed or incomplete information.
2. **GPS Synchronization Delay:** In certain situations, there can be a small delay between the actual bus position and its display on the map due to network latency or slow data refresh rates.
3. **Browser Compatibility Issues:** Some older or less frequently updated browsers may not display certain interface elements correctly, leading to minor performance inconsistencies.
4. **Limited Offline Support:** The current system depends on live data fetching and lacks offline capabilities, which can affect usability when network access is temporarily unavailable.

Even though these limitations exist, they do not significantly impact the overall performance or reliability of the application. Future enhancements such as offline data storage, API optimization, and stronger synchronization algorithms can help overcome these challenges. Overall, the advantages of the Bus Tracker Application clearly outweigh its limitations, making it a dependable and modern solution for improving public transportation systems.

8. RESULTS AND DISCUSSIONS

The development and deployment of the **Bus Tracker Application** led to substantial improvements in the overall efficiency and reliability of the public transport experience when compared to the traditional, manually operated system. The introduction of real-time GPS tracking and cloud integration has transformed how passengers interact with bus services. By providing continuous live updates on bus positions, routes, and arrival times, the system has significantly reduced passenger waiting periods at bus stops. Commuters can now plan their journeys more accurately, knowing exactly when their bus will arrive, which has increased convenience and reduced frustration. The ability to track buses in motion not only eliminates guesswork but also builds trust in public transport schedules, encouraging more people to rely on buses instead of personal vehicles. From an administrative standpoint, the system has improved operational transparency and management efficiency. Transport authorities can remotely monitor the performance of each bus, identify delays, and make quick updates to schedules or routes whenever necessary. This instant data synchronization ensures that any modification made by an administrator is reflected immediately on the user interface, maintaining consistent information for all users. The platform's **cross-platform compatibility**—working equally well on smartphones, tablets, and desktops—has also made it highly accessible to a broad range of users without requiring any installation or software updates.

Another important achievement is the scalability of the architecture. Since the application is hosted entirely on cloud platforms such as Firebase and Netlify, it can easily expand to support additional routes, buses, and even other cities with minimal changes to its configuration. The use of asynchronous communication between the client and server helps maintain fast performance even under heavy usage. During user testing, students and daily commuters particularly appreciated the real-time tracking feature, noting that it helped them save time and better plan their travel routines. Administrators, on the other hand, highlighted the convenience of centralized control and automated data management, which reduced manual errors and administrative workload.

Overall, the implementation results demonstrate that the Bus Tracker Application successfully meets its objectives of improving efficiency, transparency, and user satisfaction. The system has proven capable of bridging the communication gap between commuters and transport authorities, modernizing the public transport experience in Coimbatore, and laying the groundwork for a smarter and more sustainable urban mobility solution.

9. FUTURE SCOPE FOR ENHANCEMENT

The **Bus Tracker Application** has strong potential for future enhancement and expansion to meet the evolving needs of modern transportation systems. One of the most promising directions is the development of a **dedicated mobile application** for Android and iOS platforms. Although the current version works efficiently on web browsers, a mobile app would provide smoother performance, faster notifications, and offline support, making it even more convenient for daily commuters. Another key improvement lies in the integration of

7.

Artificial Intelligence (AI) and **Machine Learning (ML)** to predict bus arrival times more accurately. By analyzing factors such as traffic conditions, weather, and peak-hour congestion, AI- based systems can generate smarter and more reliable ETA forecasts. Additionally, the application can be enhanced with **multilingual support**, allowing users to access the interface in Tamil, Hindi, and English. This would make the system more inclusive and user-friendly for people from different linguistic backgrounds.

Further advancements can include **IoT-based integration**, where smart sensors installed in buses can monitor vehicle health, engine performance, and fuel usage in real time. This would help transport authorities maintain their fleets more efficiently and reduce maintenance costs. The system can also be **expanded city-wide and across multiple regions**, making it suitable for deployment in other cities beyond Coimbatore. With minimal modifications, the existing architecture can handle larger datasets and support more routes and buses. In the long run, these technological upgrades would help the Bus Tracker Application evolve into a complete **smart transport ecosystem**, aligning with the goals of smart city initiatives and promoting sustainable, data-driven urban mobility.

10. CONCLUSION

The **Bus Tracker Application** provides an effective and innovative solution to many long- standing challenges in the public transportation system of Coimbatore. By integrating modern web technologies such as **ReactJS**, **Node.js**, **Firebase**, and the **Google Maps API**, the system bridges the communication gap between commuters and transport authorities through real-time data sharing. It allows passengers to track live bus locations, view estimated arrival times, and plan their journeys with greater confidence and convenience. The system's cloud-based structure ensures accessibility across all devices without the need for installation, making it both practical and scalable. For administrators, the platform serves as a centralized management tool that simplifies route monitoring, driver assignments, and performance evaluation.

The implementation of this application has proven that technology can play a key role in transforming traditional public transport into a smarter, more reliable, and environmentally friendly system. It not only helps reduce passenger waiting times and improve trust in public transport services but also promotes sustainability by encouraging people to use buses instead of private vehicles. Although there are some minor technical limitations, such as dependency on internet connectivity and occasional GPS delays, these issues can be resolved through continuous updates and the inclusion of predictive algorithms. Overall, the Bus Tracker Application successfully meets its objectives of efficiency, transparency, and user satisfaction. It sets a strong foundation for future developments like AI-driven ETA prediction, multilingual support, and IoT integration, marking an important step toward building a **smart, connected, and sustainable public transport system** for the future.

11. REFERENCES

1. Firebase. (2024). *Cloud Firestore Documentation.* Google Firebase. Retrieved from https://firebase.google.com/docs/firestore
2. Google Developers. (2024). *Google Maps JavaScript API Documentation.* Retrieved from https://developers.google.com/maps/docu mentation/javascript
3. Netlify. (2024). *Netlify Deployment and Hosting Guide.* Netlify Docs. Retrieved from https://docs.netlify.com
4. React Team. (2024). *React – A JavaScript Library for Building User Interfaces.* Meta Open Source. Retrieved from https://react.dev
5. Express.js Foundation. (2024). *Express: Fast, unopinionated, minimalist web framework for Node.js.* Retrieved from https://expressjs.com
6. Noback, M. (2021). *Advanced Web Application Architecture.* Leanpub.
7. Mikowski, M., & Powell, J. (2013). *Single Page Web Applications: JavaScript End- to-End.* Manning Publications.
8. Singh, H., & Bhatt, M. (2022). *Serverless Web Applications with React and Firebase.* Apress Publishing.
9. Sharma, R., & Gupta, S. (2023). *Real- Time Vehicle Tracking Systems Using GPS and Cloud Integration. International Journal of Computer Science Trends and Technology, 11*(2), 45–52.
10. Kumar, A., & Babu, V. (2022). *Enhancing Urban Mobility Through Smart Transport Applications. International Journal of Emerging Research in Engineering and Technology, 9*(4), 210–217.