



# BUS TRACKING AND ROUTE OPTIMIZATION

*A Phase 1 Report on Developing a Scalable Web Platform for Bus Tracking and Route Optimization using MERN Stack*

<sup>1</sup>Mr. Karthik Shenoy, <sup>2</sup>Dr. Shashidhar Kini K

<sup>1</sup>Student, <sup>2</sup>Professor & Head

<sup>1</sup>Department of Master of Computer Applications,  
<sup>1</sup>Srinivas Institute of Technology, Valachil Mangaluru, Karnataka, India

**Abstract :** This study undertakes the development of a Bus Tracking and Route Optimization system to improve urban public transport efficiency. Key features include real-time GPS tracking, route planning, and schedule management. Data such as bus locations, stops, and traffic patterns are used to dynamically suggest optimal routes. The system architecture is based on the MERN stack and integrates mapping APIs like Google Maps. The analytical framework includes data collection, system design, API integration, and implementation of routing algorithms. The objective is to provide reliable, real-time transit information for passengers and improve decision-making for transit authorities.

**IndexTerms -** Bus Tracking ,Route Optimization, Real Time GPS, Public Transport, Transport Management, Location Based Services , Urban Mobility, Web based Bus Application, Smart Transportation.

## I. INTRODUCTION

Public transportation systems are vital for urban mobility, but they often face problems such as inefficient routes, poor schedule adherence, and lack of real-time tracking. These issues lead to delays, longer travel times, and commuter dissatisfaction. Many transport services still rely on static schedules and manual oversight, which makes it difficult to respond to changing traffic patterns or service disruptions effectively.

This project focuses on building a smart Bus Tracking and Route Optimization system that uses live GPS data, real-time traffic information, and user-centric design to improve the overall efficiency of bus operations. One of its primary objectives is to streamline schedule management, allowing operators to adjust timings dynamically and provide passengers with accurate arrival information.

By integrating technologies such as the MERN stack and Google Maps API, the system offers real-time tracking, optimized routing, and an intuitive interface for users. These tools help reduce waiting times, fuel consumption, and operational inefficiencies. The benefits include:

- Passengers: Gain access to real-time bus locations and accurate schedules for better trip planning.
- Operators: Manage fleets more efficiently, reduce idle time, and adapt to route or schedule changes.
- Transport Authorities: Use data-driven insights for future planning and policy improvements.

The scope of this project is focused on city bus services, using GPS data, real-time traffic feeds, and schedule information. Built with the MERN stack and Google Maps API, the system supports live tracking, route optimization, and schedule management. This report details the Phase 1 methodology, covering system planning, data integration, route modeling, and projected outcomes.

## II. EASE OF USE

The proposed Bus Tracking and Route Optimization System is developed with a strong focus on user accessibility, real-world practicality, and operational efficiency. Designed for both commuters and transport administrators, the system offers a user-friendly interface where passengers can check live bus locations, ETAs, and optimized routes in real-time without requiring technical expertise. This makes the platform highly suitable for daily commuters, city planners, and transit authorities alike.

Integration with existing systems is a core feature. By deploying the platform as a modular API, it can be easily embedded into mobile apps, web portals, or smart city dashboards. The intuitive interface ensures that users can quickly search for routes, filter by location, and receive travel updates with minimal effort. The system is also designed for scalability—new routes, stops, and service features can be added without affecting existing operations.

To meet real-time transit demands, the system emphasizes performance and responsiveness. Built using lightweight, efficient technologies like the MERN stack (MongoDB, Express.js, React.js, Node.js), the platform ensures quick data retrieval and smooth

user interaction. Route optimization and live tracking leverage simulated or real-time GPS data, ensuring that commuters receive accurate travel information even under changing traffic conditions.

The platform balances accuracy, speed, and usability to deliver a robust solution for smart transportation. Future scalability is also considered—allowing easy integration with IoT-enabled buses, predictive analytics for delay estimation, and support for fare systems or voice-assist features. Overall, the system is built to offer reliable, adaptable, and seamless transit experiences for modern urban mobility.

### 1. Prepare Your Paper before Styling

Before finalizing the structure and formatting of the paper, substantial focus was placed on the clarity, completeness, and technical accuracy of the content. The development process began with the collection of relevant data, including simulated GPS coordinates, bus route details, stop names, and estimated arrival times (ETAs). These data points were sourced from open transportation platforms, static maps, and public transit APIs. The goal was to ensure that the sample dataset reflected realistic commuting scenarios commonly found in Indian urban transit systems.

The preprocessing phase involved cleaning, organizing, and standardizing the collected data. This included formatting inconsistent stop names, filtering out incomplete records, and encoding categorical variables such as route numbers and stop sequences. Numeric values like ETA, distance, and travel duration were normalized for consistent use across the platform. These steps were crucial to enable accurate route tracking, filtering, and display functionalities within the system.

Application development was carried out using the MERN stack (MongoDB, Express.js, React.js, Node.js). React.js was used for building a dynamic and responsive user interface, while MongoDB provided a flexible NoSQL structure to store route data and user queries. Backend services built with Node.js and Express.js enabled smooth API communication and handled real-time data operations. Key features like live tracking with Leaflet, route search, and ETA calculation were implemented and tested under different usage scenarios.

Only after verifying the system's performance and functionality through manual testing and simulated environments was the paper formally structured. The final documentation was organized into standard academic sections: Introduction, Literature Review, Methodology, System Architecture, Results, and Conclusion. All diagrams, tables, and references were integrated at this stage to enhance readability and technical accuracy.

### 2. Abbreviations and Acronyms

In this paper, the following abbreviations and acronyms are used:

- MERN – MongoDB, Express.js, React.js, Node.js
- UI – User Interface
- UX – User Experience
- API – Application Programming Interface
- GPS – Global Positioning System
- ODM – Object Data Modeling
- IDE – Integrated Development Environment
- DOM – Document Object Model
- CRUD – Create, Read, Update, Delete
- JSON – JavaScript Object Notation
- CI/CD – Continuous Integration / Continuous Deployment
- HTTP – Hypertext Transfer Protocol
- DBMS – Database Management System
- ETA – Estimated Time of Arrival
- JWT – JSON Web Token
- CSS – Cascading Style Sheets
- JS – JavaScript
- DB – Database

### III. RESEARCH METHODOLOGY

This section outlines the methodology adopted to develop a web-based application aimed at optimizing public transportation through live bus tracking and route management. It includes the scope of the system, sample data, data sources, application architecture, and the development tools and technologies employed.

#### 3.1 Population and Sample

The population for this study comprises urban and semi-urban public transportation systems in India, including both government-run and private bus networks. For prototyping and simulation, sample data was manually curated and also synthesized using open-source mapping tools and APIs such as Google Maps.

The sample includes a diverse set of bus routes, stops, and estimated travel times, along with simulated GPS coordinate data to replicate real-time tracking. The dataset also includes passenger-centric parameters like location-based search queries and route preference patterns.

Only routes with complete information such as start and end points, intermediate stops, estimated time of arrival (ETA), and travel durations were selected. This cross-sectional data reflects standard public transportation schedules and movement patterns as of early 2025.

### 3.2 Data and Sources of Data

This study relies on a combination of secondary and simulated primary data collected from public route information portals, government transit APIs, Google Maps, and manually created test scenarios.

Key data points used in the system include:

- Bus Route Information (e.g., Route 12A: Central Bus Stand to Tech Park)
- Stop Locations (GPS coordinates and stop names)
- Estimated Time of Arrival (ETA) and Departure Time
- Travel Distance and Average Speed
- Traffic Conditions (simulated for evaluation)
- User Preferences (source/destination queries, filter selections)

Simulated datasets were used to test system responsiveness and ensure consistent feature functionality across varied traffic and route conditions.

### 3.3 Theoretical framework

The objective of this system is to improve public transportation by allowing users to view real-time bus locations, check ETAs, and plan optimal routes. This platform is built with a strong focus on UI/UX to provide a seamless and interactive commuting experience.

While this project does not currently integrate predictive machine learning models, it is designed with a modular architecture that can support future extensions like delay prediction, crowd estimation, and AI-based route optimization.

Key variables used in the system include:

- Route ID and Stop ID (categorical)
- GPS Coordinates (numerical)
- Travel Time and ETA (numerical)
- Bus Status (Live/Idle) (categorical)
- Traffic Condition Flags (categorical)
- User Queries (source/destination inputs)

### 3.4 Development tools and Web Technologies

This section outlines the technological ecosystem, software tools, and design methodologies used to build the Bus Tracking and Route Optimization System.

#### 3.4.1 Descriptive Statistics

Initial analysis of the collected route and location data involved summarizing metrics such as average route length, estimated travel times, and stop frequencies. Data consistency was ensured by validating stop coordinates and ETA values. These insights helped in designing the interface layout, such as frequently used filters and real-time map elements.

#### 3.4.2 Exploratory Feature Mapping

Wireframing tools like Figma were used to design UI mockups, route tracking pages, and admin dashboards. Navigation flow diagrams were created to map out user journeys, including search, live tracking, and route optimization interactions. Visual tree structures represented page hierarchies, from public search pages to admin route controls.

#### 3.4.3 Backend Architecture

The backend was developed using Node.js and Express.js, supporting RESTful API endpoints for bus status updates, route data retrieval, and user interaction logging. Middleware components handled request validation, error logging, and real-time data refresh mechanisms for accurate GPS updates.

#### 3.4.4 Database Design

MongoDB was selected for its schema-less structure and ability to handle geospatial queries. Collections were designed for routes, buses, stops, users, and admin configurations. Mongoose was used to define schemas for validating fields such as coordinates, ETAs, and route identifiers.

#### 3.4.5 Frontend Framework

The frontend was built using React.js, offering a responsive, component-based UI. Key components included route selection dropdowns, live map visualization (via Leaflet), ETA displays, and filterable search forms. React Hooks and Context API were used for state management across the platform.

#### 3.4.6 API Integration

Axios was used to connect frontend and backend through asynchronous REST API calls. Google Maps API and Leaflet.js were integrated for dynamic map rendering and plotting live bus locations. API keys and sensitive credentials were managed securely using environment variables.

#### 3.4.7 Testing Strategies

Backend routes and API endpoints were tested using Postman and automated scripts. Manual testing simulated user actions like route filtering, bus tracking, and ETA lookups. Integration testing ensured proper synchronization between the database, server, and UI components.

#### 3.4.8 Evaluation Criteria

System performance was evaluated based on API response time, map rendering speed, and database query latency. User experience was measured through UI responsiveness, accuracy of live bus data, and overall navigation flow. Resource usage and memory efficiency were monitored under simulated load.

#### 3.4.9 Feedback and Iteration

User feedback was collected via test users and informal surveys. Suggestions like improving mobile view responsiveness, adding delay alerts, and refining search filters were incorporated in an agile development cycle. Continuous deployment tools ensured that updates could be pushed efficiently after each sprint.

## IV. RESULTS AND DISCUSSION

## 4.1 Results of Descriptive Statics of Study Variables

Table 4.1 Descriptive Statics

User ID	Route ID	Query	Time Stamp	Result Count	Action Taken
101	12A	Central → IT Hub	2025-04-28 09:22	3	Viewed Live Status
102	7B	Station → Mall	2025-04-14 09:10	2	Set ETA Notification

Table 4.1 These simulated logs represent user-generated interaction data, showcasing how commuters engage with the platform to check bus availability, filter by routes, and track estimated arrival times in real-time.

## V. ACKNOWLEDGMENT

The author wishes to express sincere gratitude to the Project Guide and Head of the Department of MCA, Dr. Shashidhar Kini K, for his invaluable guidance, constant encouragement, and kind support throughout this research work on *Bus Tracking and Route Optimization*. Appreciation is also extended to the Principal, Dr. Shrinivasa Mayya D, for providing the institutional support and an environment conducive to the successful execution of this project. The author also thanks the management of Srinivas Institute of Technology for their direct and indirect support. Gratitude is due to all the faculty members and non-teaching staff of the MCA department for their consistent help and technical input throughout the development process. Finally, the author is deeply thankful to parents and friends for their unwavering encouragement, patience, and belief throughout this academic endeavour.

## REFERENCES

- [1] Lin, Y., Zhang, Z., & Zhao, H. (2019). *Real-Time Bus Tracking Using GPS and Cloud Computing*. *Journal of Intelligent Transportation Systems*, 23(2), 145–157. <https://doi.org/10.1080/15472450.2019.1587554>
- [2] Sharma, R., & Rao, S. (2021). *Public Transportation System Using MERN Stack*. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12(5), 78–84.
- [3] Ahmed, N., & Khan, F. (2020). *Application of NoSQL Databases in Transport Systems*. *Journal of Computer and Communications*, 8(12), 25–33.
- [4] Taylor, J., Patel, S., & Natarajan, P. (2019). *Frontend Optimization in Real-Time Dashboards Using React.js*. *Proceedings of the Web Engineering Conference (WEBCON)*, 102–110.
- [5] Gupta, A., & Verma, R. (2022). *Node.js and Express for Scalable Web Services in Urban Mobility Solutions*. *IEEE International Conference on Smart Technologies for Transportation and Urban Mobility*, 45–51. <https://doi.org/10.1109/ICSTUM.2022.10023456>
- [6] Parker, L., & Adams, K. (2020). *Designing Accessible Public Transportation Applications*. *ACM Conference on Human Factors in Computing Systems (CHI)*, 502–509.
- [7] Wilson, M., Fernandez, T., & Roy, A. (2021). *Contactless Fare Payment Systems in Smart Transit*. *International Journal of Transportation Management*, 18(3), 112–118.
- [8] Kumar, S., & Mehta, D. (2022). *AI and Predictive Analytics in Route Optimization*. *Journal of Artificial Intelligence Research*, 59(4), 305–320. <https://doi.org/10.1613/jair.12035>