BharatRide: Smart Bus Tracking

Akanksha Pawar†
Department of
Computer
Engineering
VES Polytechnic
Mumbai, India
co2024.akanksha.p
aw ar@ves.ac.in

Sahil Chaudhari‡
Department of
Computer
Engineering
VES Polytechnic
Mumbai, India
co2024.sahil.chaud
ha ri@ves.ac.in

Amishal Ghadi§
Department of
Computer
Engineering
VES Polytechnic
Mumbai, India
co2024.amishal.gh
adi @ves.ac.in

Aditi Thul¶
Department of
Computer
Engineering
VES Polytechnic
Mumbai, India
co2024.aditi.thul
@ves.ac.in

Vidya Lunge*
Department of
Computer
Engineering
VES Polytechnic
Mumbai, India
vidya.lunge@ves.
ac.in

***Abstract*—** BharatRide is a General Transit Feed Specification (GTFS)–based smart bus tracking and route planning system designed to enhance the usability of public transportation in Indian cities. The system integrates real-time Global Positioning System (GPS)–based bus tracking, Estimated Time of Arrival (ETA) prediction, and intelligent route planning within a scalable cloudbased architecture. The proposed platform utilizes Flutter for cross-platform mobile application development, Node.js with Express for backend services, MongoDB and Firebase for data management, and OpenStreetMap for route visualization. Machine learning modules can be incorporated to enhance ETA accuracy and support crowd estimation using historical travel data. The system aims to improve commuter experience, operational transparency, and decision-making in urban public transportation systems.

Keywords— Smart Transportation, General Transit Feed Specification (GTFS), Real-Time Bus Tracking, Estimated Time of Arrival (ETA), Route Planning, Intelligent Transportation System, Cloud-Based Architecture

I. INTRODUCTION

Public transportation plays a crucial role in urban mobility across India, where buses remain one of the most affordable and widely used travel options. However, many public bus systems still rely on static timetables and manual information systems, resulting in uncertainty, delays, and commuter dissatisfaction. The absence of realtime bus tracking, dynamic route planning, and predictive arrival systems limits the efficiency of public transport operations.

BharatRide addresses these challenges by introducing a smart bus tracking and route planning platform that combines GTFS-based transit data, real-time GPS tracking, and cloudbased processing. The system enables commuters to track buses live, estimate arrival times, search routes and stops, and receive delay notifications. By integrating modern mobile technologies, open transit standards, and scalable backend infrastructure, BharatRide aims to improve commuter convenience while supporting smart city transportation initiatives.

II. PROBLEM STATEMENT

Existing public bus systems in many Indian cities provide basic transportation services and limited digital support, which has static route information, fixed timetables, and in some cases basic GPS-based tracking. However, these systems suffer from multiple operational and passenger-related challenges:

- Lack of accurate real-time bus location visibility
- Unreliable or unavailable Estimated Time of Arrival (ETA) information
- Difficulty in identifying correct routes and nearby stops
- No alternate route suggestions during delays or service disruptions
- Limited digital engagement and personalization features

- No effective integration of historical data for predictive analysis

These limitations reduce commuter trust and lead to increased travel time, overcrowding, and poor decision-making. Therefore, a unified system capable of integrating live tracking, predictive analytics, and intelligent route planning is required to modernize urban bus transportation.

The existing bus tracking system uses a centralized cloud-based architecture. Buses equipped with GPS devices send live location data to a cloud server, where it is stored and processed. Passenger mobile applications retrieve this data to display bus locations, routes, and basic Estimated Time of Arrival (ETA) information.

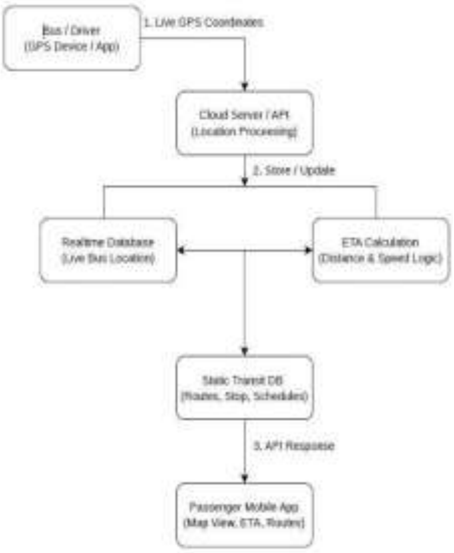

Fig. 1. Existing System Architecture

III. OBJECTIVES

The main objectives of the BharatRide system are:

- To provide real-time bus tracking using GPS-enabled devices

- To deliver accurate ETA predictions using live and historical data
- To visualize routes and stops using interactive digital maps
- To enable source–destination journey planning with multiple route suggestions
- To provide user personalization features such as saved routes and alerts
- To design a scalable, secure, and cloudbased system architecture
- To support offline access using locally cached GTFS data

IV. PROPOSED SYSTEM

A. System Architecture

BharatRide follows a modular client–server architecture designed to ensure scalability, reliability, and realtime performance. The system is divided into four primary layers, each responsible for a specific set of tasks.

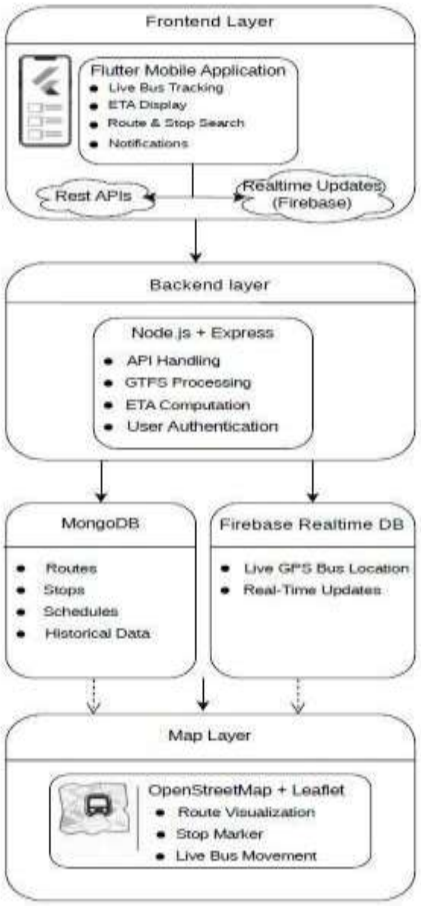

Fig. 2. Block diagram of the proposed BharatRide system

1) Frontend Layer (Flutter Mobile Application)

The frontend layer is developed using Flutter, enabling cross-platform deployment on Android and iOS devices. This layer provides the user interface through which commuters interact with the system. Key functionalities include:

- Viewing real-time bus locations on the map
- Checking Estimated Time of Arrival (ETA) for selected stops
- Searching buses by route number
- Searching nearby stops using GPS location
- Receiving notifications and alerts

The application communicates with backend services through REST APIs and listens to realtime updates from Firebase for live tracking.

2) Backend Layer (Node.js + Express)

The backend layer is built using Node.js with the Express framework, which handles server-side logic and API management. Its responsibilities include:

- Processing user requests from the mobile application
- Parsing and storing GTFS datasets
- Managing route and stop queries
- Computing ETAs based on live and historical data
- Handling authentication and user data
- Acting as a middleware between databases and the frontend

This layer ensures secure communication and efficient data processing.

3) Data Layer

The data layer consists of two major components:

- MongoDB:

Stores static transit data derived from GTFS feeds, including routes, trips, stops, stop times, and calendars. It also stores historical trip and travel time data used for predictive analysis.

- Firebase Realtime Database:

Handles live GPS location updates from buses. It provides low-latency data synchronization between the backend and the Flutter app, enabling real-time tracking.

Together, these databases ensure both persistent storage and real-time responsiveness.

4) Map Layer (OpenStreetMap + Leaflet)

The map layer is responsible for geographical visualization. OpenStreetMap provides opensource map tiles, while Leaflet is used for rendering routes and markers. This layer:

- Displays bus routes using polylines
- Shows stops as interactive markers
- Updates live bus positions dynamically
- Allows zooming, panning, and route interaction

This separation of layers improves maintainability and allows independent scaling of system components.

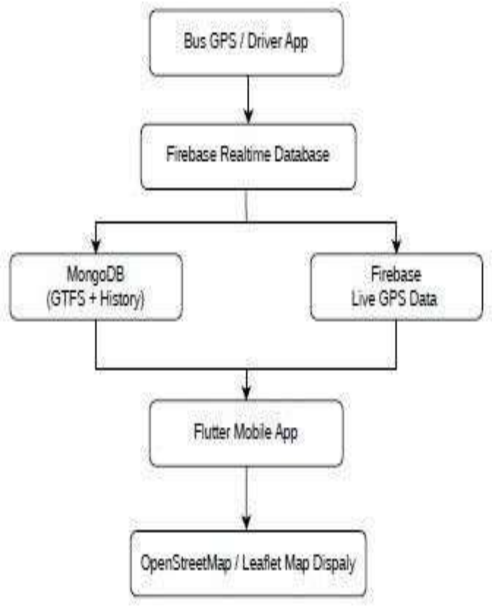

Fig. 3. Overall BharatRide System Flow

V. FUNCTIONAL MODULES

A. *Live Bus Tracking*

Each bus is equipped with a GPS-enabled device or driver mobile application that periodically transmits latitude and longitude coordinates. These updates are sent to Firebase Realtime Database, ensuring lowlatency communication. The Flutter app listens for location updates and dynamically displays the moving bus on the map.

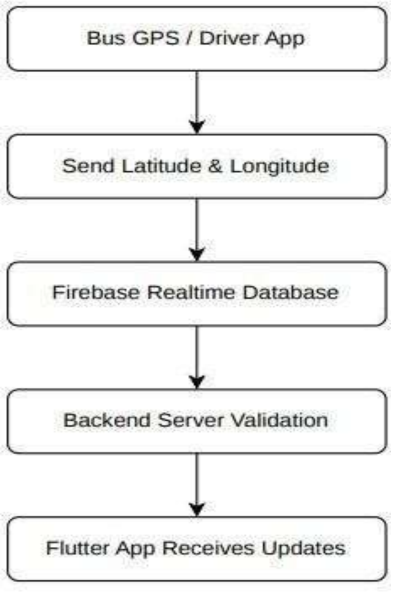
Bus GPS / Driver App
Send Latitude & Longitude
Firebase Realtime Database
Backend Server Validation
Flutter App Receives Updates

Fig. 4. Live Bus Tracking Data Flow

B. *ETA Prediction*

The ETA module computes the expected arrival time of a bus by integrating real-time operational data with static transit information. The current bus speed and geographic position are obtained from continuous GPS updates, while the remaining distance to upcoming stops is calculated using route geometry and map data. Predefined route schedules derived from GTFS feeds are used to provide baseline timing information and stop sequences. In addition, historical travel time data collected from past trips is utilized to capture recurring traffic patterns, peak-hour variations, and route-specific delays. Based on these inputs, the backend system performs ETA estimation using distance–time calculations and statistical averaging techniques. An optional machine learning–based regression model can be incorporated to further enhance prediction accuracy by learning complex relationships between historical trip durations, time of day, and traffic conditions, enabling more reliable ETA estimates under dynamic urban traffic scenarios.

VI. TECHNOLOGY STACK

Component	**Technology**
Mobile Application	Flutter [5]
Backend	Node.js [6], Express
Database	MongoDB [4], Firebase [3]
Maps	OpenStreetMap [2], Leaflet
Realtime Communication	Firebase [3]
Machine Learning	Python, Scikit-learn

Table I. Technology stack used in BharatRide Technology

Stack Description

Flutter:
Used for developing a cross-platform mobile application with a consistent user interface.

Node.js and Express: Used the handle backend logic, REST API communication, and data processing.

MongoDB:
Stores static transit data and historical travel records.

Firebase Realtime Database: Manages live GPS location updates with low latency.

OpenStreetMap:
Provides open-source map data for route and stop visualization.

Firebase:
Supports real-time communication and notification delivery.

Python and Scikit-learn :
Can be used to improve ETA prediction using historical data.

Component	Specification
Operating System	Microsoft Windows 11 Home (64-bit)
Processor	12th Gen Intel® Core™ i5-12450H
RAM	8 GB
System Model	Victus by HP Gaming Laptop 15-fa1xxx

Table II. Hardware Used for Development

VII. RESULTS

Parameter	Existing Bus System	BharatRide
Real-time Tracking	Limited or unreliable	Accurate real-time GPS-based tracking
ETA Prediction	Static or inaccurate	Dynamic ETA using live and historical data
Route Planning	Fixed routes only	Intelligent source–destination routing
Alternate Routes	Not available	Multi-route suggestions supported
User Interaction	Minimal digital features	Notifications and interactive map view

Table III. Comparison of Existing System and BharatRide

The comparison shows that BharatRide overcomes key limitations of existing systems by providing reliable real-time tracking, improved ETA accuracy, and intelligent route planning, leading to better commuter experience.

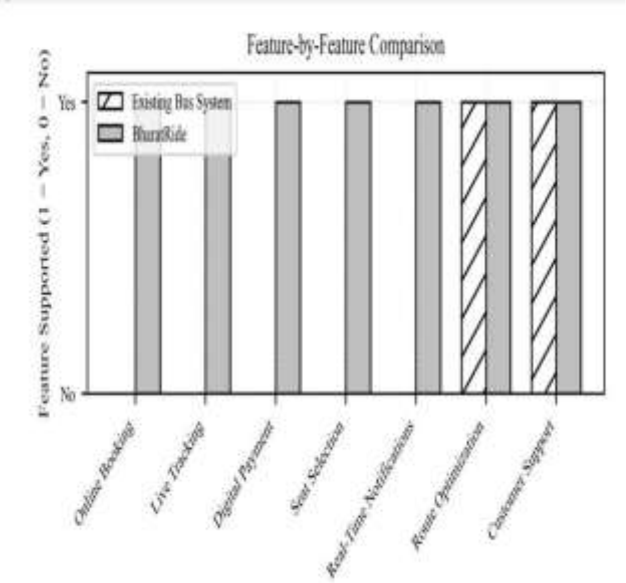

Fig. 5. Feature support comparison between existing bus systems and BharatRide

VIII. CONCLUSION

This paper presented BharatRide, a GTFS-based smart bus tracking and route planning system designed to address key limitations observed in existing public bus information systems. Based on the comparative analysis and results, BharatRide demonstrates significant improvements in realtime bus tracking, ETA prediction accuracy, and overall feature support when compared to traditional bus systems.

The results show that, unlike existing systems that rely on static schedules and limited tracking, BharatRide integrates live GPS data with historical travel patterns to provide more reliable and dynamic ETA estimates. The feature comparison further highlights the system’s ability to support intelligent route planning, alternate route suggestions, and real-time user notifications, thereby enhancing commuter experience and decision- making.

The modular client–server architecture ensures scalability, reliability, and real-time performance, making the system suitable for deployment across urban transportation networks. While the current implementation focuses on core tracking and planning functionalities, optional machine learning modules can be incorporated in the future to further improve ETA accuracy and predictive capabilities.

Overall, BharatRide provides a practical and extensible solution for modernizing urban bus transportation systems and has the potential to improve operational transparency and commuter satisfaction in Indian cities.

IX. REFERENCES

[1] Google, "General Transit Feed Specification (GTFS)," Available: https://gtfs.org/

[2] OpenStreetMap Contributors, "OpenStreetMap," Available: https://www.openstreetmap.org/

[3] Firebase, "Firebase Realtime Database," Google, Available: https://firebase.google.com/

[4] MongoDB Inc., "MongoDB Documentation," Available: https://www.mongodb.com/docs/

[5] Flutter, "Flutter Documentation," Available: https://docs.flutter.dev/

[6] Node.js Foundation, "Node.js

Documentation," Available: https://nodejs.org/