International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

BharatRide: AI-Powered Multi-City Road Trip and Mobility Platform

Prof. Chetana R Shivanagi[1], Aishwarya Sangam[2], Rohit Lokare[3], Basavaraj S Yaragatti[4], Aishwarya Bhandarakavate[5]

[1]*Assistant Professor, Dept. of Information Science and Engineering, Basaveshwar Engineering College, Bagalkote, Karnataka, India.*

[2345]*Student, Dept. of Information Science and Engineering, Basaveshwar Engineering College, Bagalkote, Karnataka, India,*

Abstract - *Planning a road trip across several cities can require considerable effort. Travellers need to manage destinations, stay durations, travel schedules, transportation, and payments, and unexpected events such as driver cancellations, vehicle breakdowns, and major delays can disturb the journey. BharatRide is a proposed AI-powered road trip and mobility platform that lets users create a complete journey by selecting a starting location, adding several destinations, choosing the number of days to stay at each place, and selecting a final return destination. Customers can view estimated fares, book drivers, track their journeys, and pay online. Drivers manage their profiles, vehicles, trip requests, and earnings, and one Admin manages users, drivers, vehicles, bookings, payments, and reported problems within the same mobile application. The Trip Rescue feature helps customers manage interruptions by identifying possible replacement drivers or vehicles and updating the remaining itinerary. The customer application also provides weather prediction for the planned destinations and suggestions for nearby places. The implementation uses React Native and TypeScript for the mobile application, Java 21 and Spring Boot for the backend, MongoDB Atlas for data storage, and Python with FastAPI for AI-related functions, supported by Google Maps, Google OR-Tools, the OpenAI API, Firebase Cloud Messaging, and Razorpay. Trip Rescue, weather prediction, and nearby place suggestions have been implemented, and this paper presents the system design, implementation approach, and a testing plan. The benefits of the platform must be confirmed through implementation and testing.*

Key Words: Multi-city travel, road trip planning, mobility platform, route optimization, Trip Rescue, weather prediction, nearby place suggestions, artificial intelligence.

1. INTRODUCTION

Planning a road trip across multiple cities can require considerable effort. Travellers need to manage destinations, stay durations, travel schedules, transportation, and payments. Unexpected situations, such as driver cancellations, vehicle breakdowns, and major delays, can further disturb the journey. Finding a suitable replacement and updating the remaining travel plan may become difficult.

BharatRide is an AI-powered road trip and mobility platform designed to make multi-city and multi-day travel easier to plan and manage. It allows users to create a complete journey by selecting a starting location, adding multiple destinations, choosing the number of days to stay at each place, and selecting a final return destination. Instead of managing every part of the journey separately, users can organize their travel through a single application.

The platform includes separate functions for customers, drivers, and one Admin. Its Trip Rescue feature focuses on helping customers manage unexpected interruptions by identifying possible replacement drivers or vehicles and updating the remaining itinerary. To help customers prepare for each stop, the application also shows weather predictions for the destinations and suggests nearby places to visit. Trip Rescue, weather prediction, and nearby place suggestions have been implemented. This paper describes the design and implementation approach; it does not claim that the platform has already been tested in real-world travel.

2. LITERATURE SURVEY

Research on tour itinerary recommendation shows that planning a trip with several stops is complicated by real-life limits such as available touring time, uncertain traffic conditions, and weather [1]. Travel planning has also been used to test AI language agents. The TravelPlanner benchmark found that even a strong language model completed very few complex multi-constraint travel plans, because the agents struggled to keep track of several constraints at once [2]. This supports the BharatRide design, in which the language model only explains suggestions, while route order and fares come from route planning and backend calculation.

Route planning for a vehicle that must visit many places has been studied extensively as the vehicle routing problem. Surveys of multi-attribute routing problems review heuristics that handle the extra constraints found in real applications [3], and surveys of dynamic routing describe how routes are revised when new information arrives during execution [4]. These works provide the basis for tools such as Google OR-Tools, which BharatRide uses to evaluate destination orders.

Driver cancellation is a known problem for on-demand mobility. Work on order dispatching in ride-sourcing markets notes that inefficient dispatch often leads to cancellation by the customer or the driver, and models this behaviour to improve dispatch and pricing decisions [5]. For the case where a vehicle breaks down after a plan has started, disruption-management research develops methods to produce a new routing solution quickly and keep costs low [6]. Commercial services such as Uber Intercity and Savaari already offer outstation rides, vehicle choices, fare estimates, and driver support [10], [11], so multi-stop booking and intercity cabs are established.

Weather and nearby places also matter to travellers. Weather affects tourists' decisions, and some places or activities should not be suggested in dangerous weather. A context-aware recommender that used the weather at the suggested places was rated higher for recommendation quality and choice satisfaction in a live user study [7], and the gTravel framework predicts the weather at points of interest for groups of tourists [8]. Suggesting places near a traveller is a point-of-interest recommendation task. A survey of this field reviews the main techniques and information sources, including the geographical signal that links places to location [9]. Table 1 summarises these areas and the gap relevant to BharatRide.

Table -1: Summary of literature survey

Area	Existing Focus	Gap Relevant to BharatRide
Multi-stop trip planning [1]	Tour itineraries under time and traffic limits	Connect the itinerary to booking and drivers
AI travel planning [2]	Language agents planning under many constraints	Use AI for explanation; use routing and backend for values
Route optimization [3], [4]	Vehicle routing heuristics and dynamic re-routing	Apply to a customer's destination order and remaining journey
Cancellation and dispatch [5]	Driver and customer cancellation in ride-sourcing	Customer-facing replacement review for long trips
Disruption management [6]	Re-planning after a vehicle breakdown	Update the same trip record and fare for the customer
Weather-aware suggestions [7], [8]	Weather used to suggest or predict at places of interest	Show weather for each stop of a multi-city trip
Nearby place suggestion [9]	Point-of-interest recommendation from location data	Suggest places near each destination in the customer's trip
Intercity cab services [10], [11]	Outstation rides, vehicle options, fare estimates	Limited public detail on recovery after a disruption

2.1 Identified Research Gap

The reviewed services and research cover trip planning, route optimization, dispatch, vehicle re-planning, weather-aware suggestions, and place recommendation. However, the public information reviewed does not clearly describe one flow that starts when a driver cancels or a vehicle fails and then checks replacements, updates the remaining itinerary, estimates schedule and fare changes, and lets the customer review the choices under the same trip record, alongside weather and nearby-place information for the trip. This is a possible area for improvement, not proof that no existing service offers similar support.

3. PROPOSED SYSTEM

3.1 System Architecture

BharatRide follows a client-server architecture. The React Native mobile application communicates with the Java Spring Boot backend through REST APIs. The backend manages authentication, user information, trips, bookings, driver assignments, payments, and other business operations. MongoDB Atlas stores the application's data.

For AI-related tasks, the backend communicates with a separate Python FastAPI service. Google Maps provides road-distance and travel-time information, while Google OR-Tools helps evaluate route options. The OpenAI API can generate travel suggestions and explain recommendations in simple language. Firebase Cloud Messaging sends notifications, and Razorpay handles payment processing through the backend. Weather information for the destinations is obtained from an external weather data service through the backend, and nearby places are suggested using map and location data. The Admin interface is part of the same mobile application. The system maintains three roles: CUSTOMER, DRIVER, and ADMIN, and the backend enforces the permissions of the single Admin.

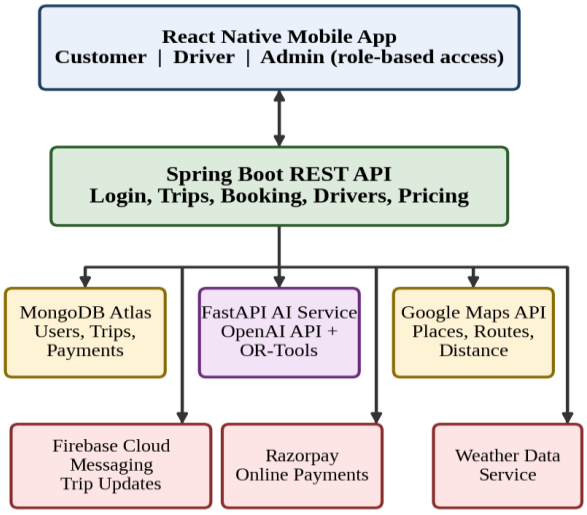


Fig -1: System architecture of the proposed BharatRide platform

3.2 Main Features

1. Multi-City Trip Planning: Users can add several destinations to one itinerary.

2. Multi-Day Scheduling: Users can specify their travel dates and stay duration at each destination.

3. Flexible Return Destination: Users can return to their starting location or select another destination.

4. AI-Based Route Suggestions: The system evaluates possible routes and suggests a suitable destination order.

5. Fare Estimation: The platform estimates trip costs based on distance, duration, and applicable charges.

6. Driver and Vehicle Management: Drivers can register, provide vehicle details, manage requests, and update trip status.

7. Trip Rescue: The system helps identify replacement options when a driver cancels or a vehicle breaks down. This feature has been implemented.

8. Return-Trip Matching: The platform can suggest suitable bookings for drivers travelling back toward another city.

9. Live Trip Tracking: Customers can view driver location and trip progress when tracking is available.

10. Weather Prediction: Customers can view the predicted weather for the destinations in their itinerary.

11. Nearby Place Suggestions: Customers receive suggestions for places to visit near their destinations.

12. Online Payments: Razorpay supports payment processing and verification.

13. Admin Management: One Admin manages users, drivers, bookings, vehicles, payments, and reported problems.

3.3 Trip Rescue Workflow

Fig. 2 shows the implemented response when a trip is interrupted. The system checks the remaining itinerary and searches for possible replacement options. It can present updated travel times and estimated costs when sufficient information is available. The system should not assume a replacement is available: driver availability must be checked, the customer reviews the options, and the Admin can assist when automatic matching is unsuccessful.

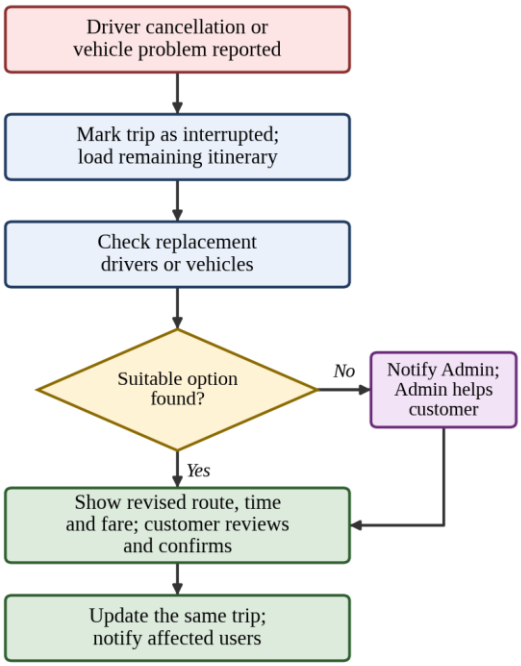


Fig -2: Proposed Trip Rescue workflow

3.4 Main Contribution

The main proposed contribution of BharatRide is the integration of multi-city trip planning with a structured trip-recovery process. Multi-stop booking, route optimization, and intercity cab services already exist in various forms, so BharatRide does not claim that these individual features are entirely new. Instead, the project connects a customer's complete itinerary with the management of unexpected travel problems. Trip Rescue aims to help customers review replacement options,

understand changes to their schedule and estimated fare, and continue managing their journey under the original trip record. Weather prediction and nearby place suggestions support the customer at each stop of the journey, and return-trip suggestions are a supporting feature that may help drivers find compatible bookings and reduce empty journeys when suitable demand is available.

4. IMPLEMENTATION

4.1 Implementation Methodology

The implementation is divided into several modules. First, authentication and role management are developed to provide secure access for customers, drivers, and the Admin. The trip-planning module then allows customers to create itineraries, add destinations, set stay durations, and select return locations. The route-planning module obtains mapping information and evaluates suitable destination orders, and the fare-estimation module calculates the estimated cost using the configured pricing rules. The driver and booking modules manage driver availability, vehicle details, booking requests, and trip status.

The Trip Rescue module handles disruptions. When a cancellation or vehicle breakdown is reported, it checks the remaining itinerary and searches for possible replacement options. The weather prediction and nearby place suggestion modules provide customers with destination information during planning and travel. Trip Rescue, weather prediction, and nearby place suggestions have been implemented. Finally, payment processing, live tracking, notifications, trip history, and administrative functions are integrated into the application.

4.2 Mobile Application and User Interface

The mobile application is developed with React Native and TypeScript and is used by customers, drivers, and the Admin. The customer interface provides trip creation, destination and stay-duration entry, route and fare details, driver booking, live tracking, weather prediction, nearby place suggestions, payment, and trip history. The driver interface provides profile and vehicle management, trip requests, status updates, and earnings. The Admin interface, in the same application, provides customer, driver, vehicle, booking, payment, and problem management.

4.3 Backend and Database

The backend is developed using Java 21 and Spring Boot and exposes REST APIs. It manages authentication, trip and booking rules, pricing, driver assignment, payment verification, and communication with the database and the AI service. Spring Security provides authentication and role-based access. MongoDB Atlas stores users, vehicles, trips, bookings, and payment records, and Spring Data MongoDB gives the backend access to the stored data.

4.4 AI and Route-Planning Module

The AI service is a separate Python FastAPI application. It uses Google OR-Tools to evaluate suitable destination orders, while the OpenAI API generates travel suggestions and itinerary explanations in simple language. Google Maps supplies location search, distance, routes, and travel time. The language model should not invent road distances, live driver availability, or fare values, which must come from map results and backend calculations. If the AI service is unavailable, basic trip viewing and booking should remain available where possible.

4.5 Weather Prediction and Nearby Place Suggestions

For each destination in the itinerary, the backend requests weather information from an external weather data service and shows the prediction to the customer, so that the traveller can prepare for the stay. Nearby place suggestions use map and location data to list places close to the destination. Both results are shown as information for the customer. They do not change the route or the fare, and they should be labelled with their data source. If a service is unavailable, the rest of the trip details remain available.

4.6 Security

Role-based access separates the CUSTOMER, DRIVER, and ADMIN functions, with permissions enforced by the backend. Passwords must be stored as secure hashes, sensitive keys must remain on the server, and payment status must be verified by the backend. Only the customer, assigned driver, and authorised Admin should see the trip details needed for their role, and location data should be used only for trip needs.

Table -2: Technology stack

Component	Technology	Purpose
Frontend	React Native, TypeScript	Mobile application for customers, drivers, and Admin
Backend	Java 21, Spring Boot	Application logic and REST APIs
Security	Spring Security	Authentication and role-based access
Database Integration	Spring Data MongoDB	Backend access to stored data
Database	MongoDB Atlas	Users, vehicles, trips, bookings, and payment records
AI Service	Python, FastAPI	Route-planning and AI-related functions

Component	Technology	Purpose
AI Assistance	OpenAI API	Travel suggestions and itinerary explanations
Route Optimization	Google OR-Tools	Evaluating suitable destination orders
Maps	Google Maps API	Location search, distance, routes, travel time, and nearby places
Weather	External weather data service	Weather prediction for destinations
Notifications	Firebase Cloud Messaging	Booking and trip notifications
Payments	Razorpay	Online payment processing

5. RESULTS AND DISCUSSION

The system should be evaluated through functional, integration, security, and performance testing. Functional tests can check login, trip creation, destination management, fare estimation, driver assignment, and payment status. Integration tests can verify communication between the mobile application, backend, database, maps service, AI service, and payment gateway. As the complete system has not been confirmed as tested, no measured improvement in travel time, cost, or recovery success is claimed here. Table 3 lists the planned functional test cases, whose status will be updated only after the tests are run.

5.1 Results

Login and Register:

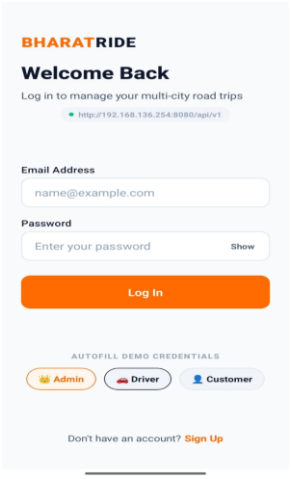


Fig -3: Login screen

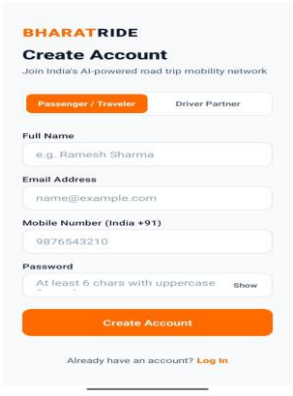


Fig -4: Register screen

Admin Panel:

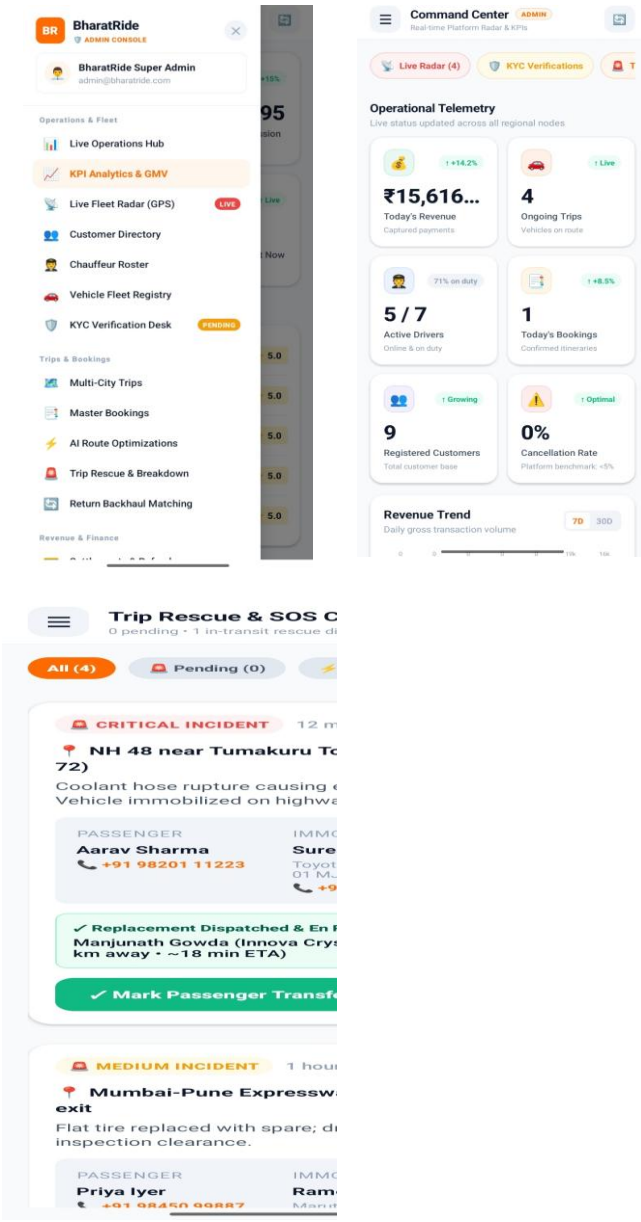


Fig -5: Admin panel screens

Driver Panel:

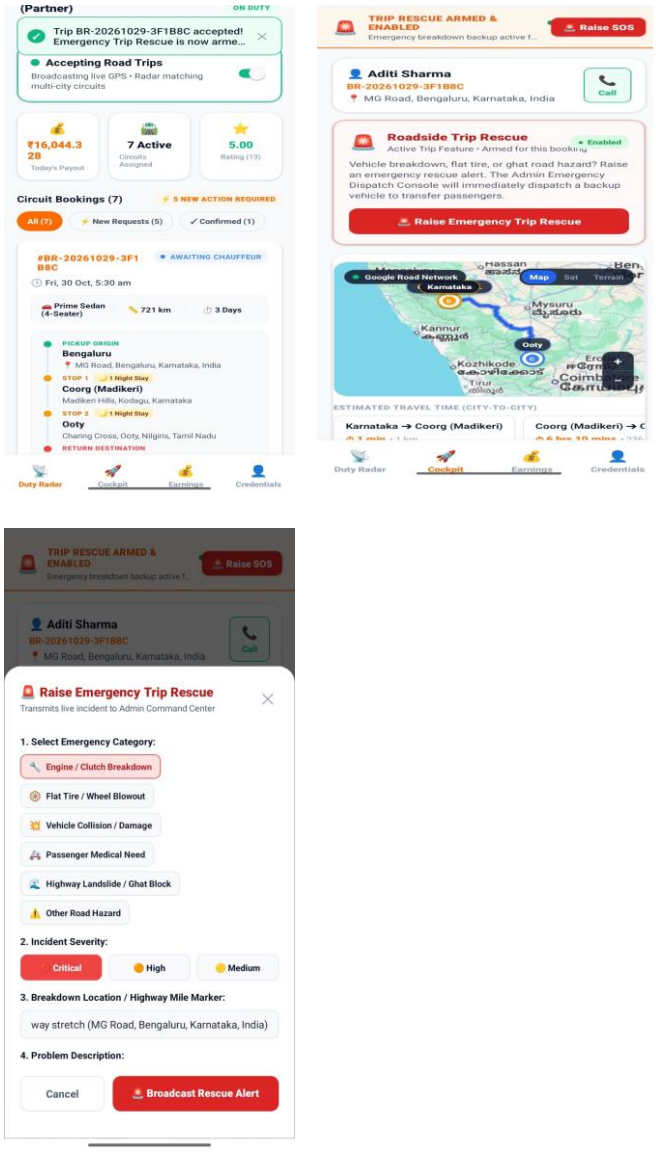


Fig -6: Driver panel screens

Customer Panel:

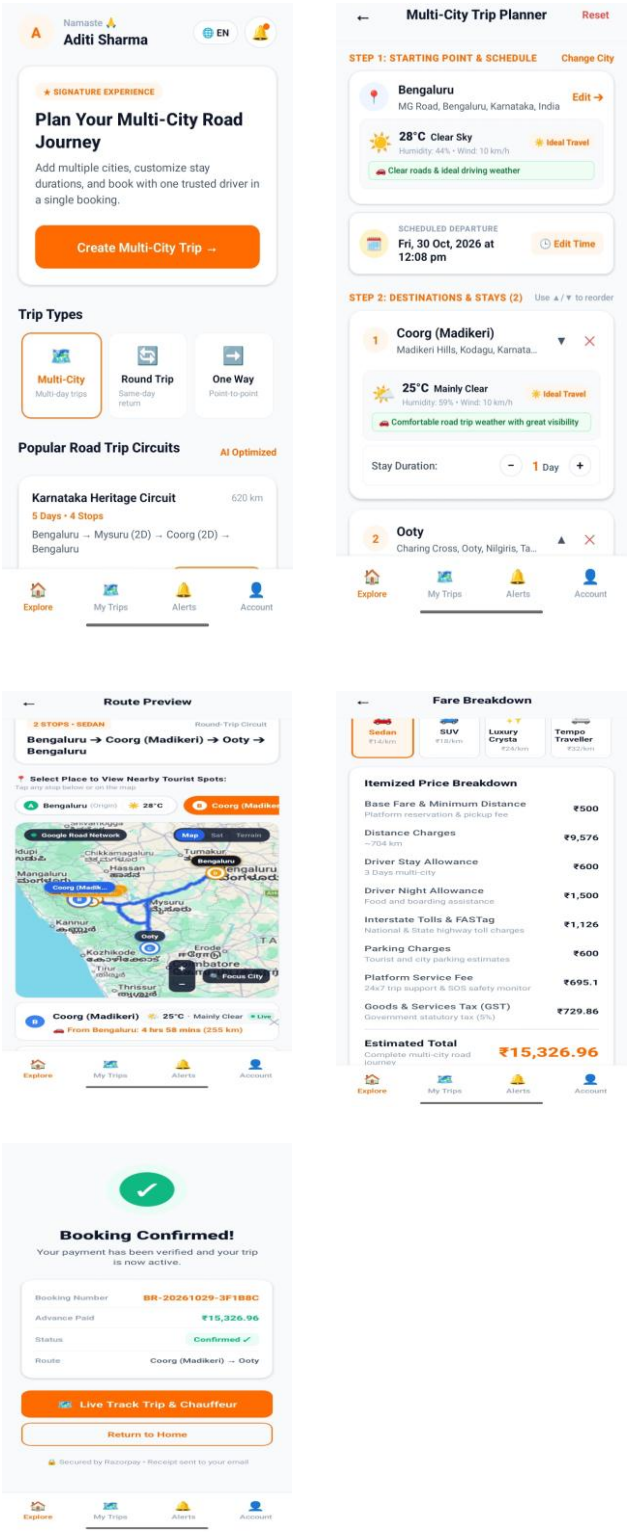


Fig -7: Customer panel screens

Table -3: Functional test cases

Test Case	Expected Result	Status
Role-based login	Customer, driver, and Admin see their own features	pass
Create multi-city trip	Trip saves with stops and stay dates	pass
Invalid travel dates	Clear validation message appears	pass
Route suggestion	Route and map details are displayed	pass
Fare estimate	Itemised estimated fare is displayed	pass
Driver assignment	Driver receives the booking request	pass
Driver cancellation	Trip Rescue search starts	pass
No replacement found	Admin help options are shown	pass
Payment failure	Failed status is recorded safely	pass
Weather prediction	Weather for each destination is displayed	pass
Nearby places	Places near the destination are suggested	pass
Live location lost	App reports unavailable tracking	pass

5.2 Trip Rescue Testing

Trip Rescue should be tested using scenarios involving driver cancellation, vehicle breakdown, replacement-driver availability, and the absence of suitable replacements. For example, consider a customer travelling from Bagalkote to Gokarna, Udupi, and Chikkamagaluru before returning to Bagalkote. If the assigned driver reports a vehicle breakdown in Gokarna, BharatRide marks the trip as interrupted and checks the remaining itinerary. It then looks for verified drivers or vehicles that may be available and able to cover the remaining route, uses map data to estimate the revised travel time, and recalculates the expected fare. The customer reviews the options and confirms a choice. If no suitable replacement is found, the Admin is asked to help. This workflow has been implemented, but it must be tested with real availability data before it can be treated as a reliable service.

5.3 Performance Evaluation

Performance evaluation can measure API response times, route-optimization time, database operations, tracking delays, and the time taken to load weather and nearby-place information. Notifications should also be checked so that customers and drivers receive the correct updates. Actual test results and measurements should be reported only after executing the tests, and features that are still planned or partially implemented should be identified clearly.

5.4 Limitations

The initial version focuses on road travel within India. The system depends on internet connectivity, mapping services, available drivers, and accurate trip information. A replacement driver may not always be available, and route suggestions depend on the quality of the available travel data. Weather predictions depend on the accuracy of the weather data, and nearby place suggestions depend on the quality of the map and location data.

6. CONCLUSION AND FUTURE ENHANCEMENTS

BharatRide aims to provide a more organized way to plan and manage multi-city road trips. By combining mobile trip planning, AI-supported route suggestions, driver management, fare estimation, payments, and the implemented Trip Rescue, weather prediction, and nearby place suggestion features, the platform seeks to improve convenience for customers and simplify operations for drivers. Its effectiveness and practical benefits must be established through implementation, testing, and comparison with existing travel services.

Future improvements may include hotel booking, more detailed travel recommendations, better driver-demand prediction, improved return-trip matching, and integration with additional travel services.

REFERENCES

1) K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, “Tour recommendation and trip planning using location-based social media: A survey,” Knowl. Inf. Syst., vol. 60, no. 3, pp. 1247–1275, 2019, doi:10.1007/s10115-018-1297-4.

2) J. Xie et al., “TravelPlanner: A benchmark for real-world planning with language agents,” in Proc. 41st Int. Conf. Mach. Learn. (ICML), PMLR vol. 235, 2024, pp. 54590–54613.

3) T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, “Heuristics for multi-attribute vehicle routing problems: A survey and synthesis,” Eur. J. Oper. Res., vol. 231, no. 1, pp. 1–21, 2013, doi:10.1016/j.ejor.2013.02.053.

4) V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” Eur. J. Oper. Res., vol. 225, no. 1, pp. 1–11, 2013, doi:10.1016/j.ejor.2012.08.015.

5) J. P. Wang, H. Wang, P. Liu, and H. J. Huang, “Order dispatching strategy and pricing scheme in ride-

sourcing markets with consideration of service cancellation," Transp. Res. B, Methodol., vol. 199, Art. no. 103266, 2025, doi:10.1016/j.trb.2025.103266.

6) Q. Mu, Z. Fu, J. Lysgaard, and R. Eglese, "Disruption management of the vehicle routing problem with vehicle breakdown," J. Oper. Res. Soc., vol. 62, no. 4, pp. 742–749, 2011, doi:10.1057/jors.2010.19.

7) M. Braunhofer, M. Elahi, F. Ricci, and T. Schievenin, "Context-aware points of interest suggestion with dynamic weather data management," in Information and Communication Technologies in Tourism 2014. Cham, Switzerland: Springer, 2013, pp. 87–100, doi:10.1007/978-3-319-03973-2_7.

8) R. Trivedi, B. Pati, and S. K. Rath, "gTravel: Weather-aware POI recommendation engine for a group of tourists," Comput. Sist., vol. 27, no. 3, pp. 667–674, 2023, doi:10.13053/cys-27-3-4550.

9) P. Sánchez and A. Bellogín, "Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective," ACM Comput. Surv., vol. 54, no. 11s, Art. no. 223, 2022, doi:10.1145/3510409.

10) Uber, "Book Outstation Cabs for Intercity Travel," [Online]. Available: https://www.uber.com/in/en/ride/intercity/. Accessed: Oct. 16, 2026.

11) Savaari Car Rentals, "Cab Booking App: Outstation, Local and Airport Taxi Booking," [Online]. Available: https://www.savaari.com/mobile-app. Accessed: Oct. 16, 2026.