On Road Vehicle Breakdown Assistance

P. Logaiyan[1], S. Sharmi[2*]

[1]Associate Professor, Department of Master Computer Application, Sri Manakula Vinayagar Engineering College, Pondicherry, India

[2]Student, Department of Master Computer Application, Sri Manakula Vinayagar Engineering College, Pondicherry, India

How to Cite this Article:

Sharmi, S. (2026). On Road Vehicle Breakdown Assistance. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(6). https://doi.org/10.55041/ijcope.v2i6.132

https://doi.org/10.55041/ijcope.v2i6.132

Abstract: Road breakdowns are one of the most stressful and potentially dangerous situations a vehicle owner can face. Whether it is a tyre going flat on a deserted stretch at night or the engine giving up in the middle of rush hour, the immediate challenge is not just mechanical — it is the complete absence of a structured, quick, and reliable way to get help. This paper presents the design and development of an On Road Vehicle Breakdown Assistance System, a technology-driven platform that uses GPS-based location tracking, real-time service provider matching, and an intuitive request workflow to connect stranded vehicle owners with the nearest available mechanic or roadside assistance professional. The system handles the entire flow — from the moment the user submits a breakdown request to the point the service provider reaches them — with a focus on speed, transparency, and ease of use. This paper describes the problem in detail, reviews existing approaches and their shortcomings, explains the proposed system architecture, and discusses the methodology, implementation, and results.

KEY WORDS: GPS Location Tracking, Roadside Assistance, Vehicle Breakdown, Real-Time Matching, Mobile Web Application, Service Provider Notification.

1. Introduction

Every day, thousands of vehicle owners across India and the world find themselves stranded on roads with no immediate recourse. A flat tyre on a deserted highway at midnight, a dead battery in the middle of peak traffic, or a sudden engine failure far from any service station — these are not rare occurrences. They are everyday realities that affect commuters, families, and professionals alike. What makes this problem particularly difficult is not the breakdown itself, but the utter lack of a fast and organised mechanism to find help.

When a vehicle breaks down, the first instinct of most people is to start calling contacts or searching the internet for a nearby mechanic. But contacts may not be available, and online searches often return outdated listings with unreliable phone numbers. Even when a mechanic is reached, communicating the exact location of the breakdown is a challenge in itself, especially in areas without prominent landmarks. The time lost in this back-and-forth can stretch from minutes into hours, turning an inconvenient situation into a genuinely dangerous one.

The existing roadside assistance ecosystem does provide some solutions, but they come with significant limitations. Most are subscription-based, requiring users to enrol in advance — something a first-time breakdown victim has not done. Some are tied to specific vehicle brands or insurance policies, making them inaccessible to the general public. And virtually all of them lack the real-time transparency that modern users have come to expect: knowing exactly when help is coming, where it currently is, and what to expect upon arrival.

This project was born out of the recognition that technology, which has already solved complex logistics problems in food delivery, cab booking, and healthcare, can and should be applied to this problem as well. The On Road Vehicle Breakdown Assistance System is a web-based platform that allows any vehicle owner to raise a breakdown request in seconds, share their GPS location automatically, and be matched with the nearest available and appropriately skilled service provider. The service provider, in turn, receives a real-time notification, can accept the request through a dedicated dashboard, and navigate to the user's location with built-in mapping support.

This paper documents the complete development of this system — the research that guided design decisions, the architecture that underpins the platform, the implementation challenges encountered, and the results achieved. It is hoped that this work contributes meaningfully to the ongoing effort to make roads safer and vehicle ownership less stressful.

2. Literature Review

A review of existing research and commercial systems in the area of roadside assistance and location-based service platforms reveals both significant progress and persistent gaps that this project seeks to address.

Sharma et al. (2020) examined the application of mobile-based platforms for emergency vehicle assistance, demonstrating that smartphone-driven service requests could reduce response times by up to 40% compared to traditional phone-based systems. Their work validated the core premise of this project — that mobile interfaces, when designed with simplicity in mind, are highly effective in emergency situations. However, their system lacked a real-time provider matching algorithm and relied on manual dispatcher assignment, which introduced delays.

Rao and Mehta (2021) proposed a GPS-integrated platform for fleet management that included basic breakdown reporting. Their proximity-based dispatch system showed promising results in controlled environments, but was built exclusively for fleet operators and did not extend to individual vehicle owners, leaving the majority of road users without a solution.

Kumar et al. (2022) investigated the use of the Haversine formula for distance calculation in location-aware mobile applications, establishing its accuracy and efficiency for use cases involving relatively short distances. Their findings directly influenced the provider-matching logic used in this project, where the Haversine formula serves as the backbone of the proximity-sorting algorithm.

Patel and Singh (2023) reviewed consumer-facing roadside assistance apps available in the Indian market and identified several key usability shortcomings: slow onboarding processes, poor location accuracy, and lack of status tracking after a request is submitted. Their survey of 200 users found that 73% of breakdown victims had to make more than three phone calls before reaching a provider who could assist them. This data was particularly instructive in shaping the design principles of the current system.

Zhang et al. (2022) explored the role of real-time push notifications in service booking applications, finding that provider response times dropped significantly when notifications were delivered via WebSocket connections rather than periodic polling. This insight led to the incorporation of WebSocket-based updates in the final iteration of the system developed in this project.

In summary, while existing research and commercial platforms address parts of the problem, no single solution brings together instant request submission, automatic GPS location sharing, proximity-based provider matching, and real-time status tracking in an open, accessible platform designed for all vehicle owners. This gap provides the justification for the present work.

3. Problem Statement

The core problem this project addresses can be stated as follows: when a vehicle breaks down on the road, the owner has no fast, reliable, and structured mechanism to locate and connect with a nearby service provider who can assist them. This seemingly simple statement encompasses several interconnected challenges that collectively make the breakdown experience far worse than it needs to be.

First, there is the problem of discoverability. Most vehicle owners do not maintain a list of nearby mechanics or towing services. In an unplanned emergency, finding a trustworthy provider quickly is difficult. Online directories exist, but they are often incomplete, outdated, or not organised by proximity.

Second, there is the problem of location communication. Even after a provider is reached by phone, explaining where the vehicle is located is unexpectedly difficult, particularly in areas without well-known landmarks or in the dark. This back-and-forth significantly delays the arrival of help.

Third, there is the problem of transparency. Once help is requested, the vehicle owner has no way of knowing how far the provider is, when they will arrive, or whether they are even on their way. This uncertainty compounds the stress of an already difficult situation.

Fourth, existing solutions are largely inaccessible to the average road user. Subscription services require advance enrolment. Brand-specific assistance is limited to premium vehicle owners. Insurance-linked services vary in quality and response time. There is no universal, on-demand platform that any vehicle owner can use at the moment of need.

4. Existing System

The current landscape of breakdown assistance in India and most developing markets is fragmented and inadequate. Traditional methods rely on the vehicle owner personally knowing a mechanic or using an online search to find one. Commercial roadside assistance is largely tied to vehicle manufacturer warranties or insurance policies, making it inaccessible to older or budget vehicle owners.

App-based platforms that do exist are either geographically limited to major cities, charge subscription fees that most users are unwilling to pay for a service they rarely need, or suffer from poor user interfaces that are unusable under the stress of an actual breakdown. None of them provide the combination of instant GPS-based location sharing, real-time provider matching, and live status tracking that this project delivers.

5. Proposed System

The proposed On Road Vehicle Breakdown Assistance System is a web-based platform that addresses each of the shortcomings identified above through a clean, fast, and transparent service workflow. It is designed to be used by anyone with a smartphone and an internet connection, with no prior enrolment or subscription required.

The system operates on two sides: the vehicle owner side and the service provider side. On the owner side, a registered user who experiences a breakdown opens the platform, selects the type of issue they are facing — tyre puncture, battery failure, engine trouble, towing requirement, or fuel shortage — and submits a request. The platform automatically captures their GPS coordinates through the browser's Geolocation API and embeds this information in the request. No manual entry of location is needed.

On the service provider side, registered providers have profiles that list their skills, current availability status, and GPS location. When a breakdown request is submitted, the system's matching algorithm identifies all active providers within a configurable radius and sorts them by proximity using the Haversine formula. The nearest appropriately skilled provider is notified first. If they do not accept the request within a defined timeout window, the next provider is automatically notified.

Once a provider accepts a request, the vehicle owner's status screen updates in real time to reflect the acceptance, showing the provider's name, contact number, and estimated time of arrival. The provider receives the owner's exact coordinates and can navigate to the location using an integrated map view. Throughout the service, both parties can communicate through a built-in messaging module.

The system is designed to be scalable, modular, and easy to extend. Future versions can incorporate payment processing, user ratings and reviews, and a mobile native application. The current prototype demonstrates the complete core workflow end to end, from request submission to service completion.

6. Methodology

The methodology for developing the On Road Vehicle Breakdown Assistance System was structured across five well-defined phases, each building upon the previous one to produce a functioning prototype. The process followed an iterative development approach, where each phase involved planning, execution, and review before moving forward.

Requirement Analysis: The first phase involved understanding the problem in depth. This included reviewing existing systems, conducting informal interviews with vehicle owners about their past breakdown experiences, and mapping out the end-to-end user journey for both vehicle owners and service providers. The output of this phase was a detailed set of functional and non-functional requirements that guided all subsequent design decisions.

System Design: Based on the requirements, the system architecture was designed using a three-tier model: a React.js front-end for the user interface, a Node.js and Express.js back-end for API handling and business logic, and a MongoDB database for data storage. The database schema was designed around three primary collections — Users, ServiceProviders, and BreakdownRequests — each with clearly defined fields and relationships. The API routes were mapped out in advance, covering authentication, request creation, provider matching, status updates, and messaging.

Implementation: Development followed a feature-by-feature approach. The authentication module was built first, using JSON Web Tokens for secure, stateless session management. This was followed by the provider matching engine, which implemented the Haversine formula to compute distances and sort providers by proximity. The request lifecycle management system was then implemented, handling the full sequence of status transitions from request creation to completion. The front-end was developed in parallel, with separate views for vehicle owners and service providers, and a shared map component for location visualisation.

Testing: The system was tested at multiple levels. Unit tests were written for all core API endpoints and helper functions using the Jest framework. Integration testing verified that the request-response cycle worked correctly end to end. User acceptance testing was conducted with a small group who role-played as vehicle owners and service providers, providing practical feedback that led to several UI improvements.

Deployment and Evaluation: The system was deployed on a local development server and evaluated against the original objectives. The evaluation confirmed that all core features functioned as intended, with the matching algorithm correctly identifying the nearest provider and the status tracking interface updating in real time.

7. Working Process

The working process of the system follows a clear and logical sequence. The vehicle owner opens the platform and logs in. Upon experiencing a breakdown, they select the type of issue and submit a request. The system captures their GPS coordinates automatically. The matching algorithm queries all active service providers, computes distances using the Haversine formula, and sends a notification to the nearest appropriately skilled provider. If the first provider accepts, both parties are connected and the status tracking screen activates. If the provider declines or does not respond within the timeout, the next provider in the sorted list is notified. Once the provider arrives and completes the service, the request is marked complete and both parties can rate the experience.

8. Implementation

The implementation of the system was carried out in a structured manner, beginning with environment setup and progressing through back-end development, front-end development, and integration testing. The development environment consisted of Node.js version 18.x, npm for package management, a locally running MongoDB instance, and the React development server. Visual Studio Code was used as the primary editor, with Git and GitHub managing version control.

The back-end authentication module used bcrypt for password hashing and JSON Web Tokens for session management. Protected API routes validated the JWT on every request, ensuring that only authenticated users could access sensitive endpoints. The provider matching algorithm was implemented as an asynchronous function that queried the ServiceProviders collection, applied the Haversine formula to each active provider, sorted the results by distance, and returned the ordered list to the request handler.

The front-end was built as a multi-page React application. The request submission flow was designed to require no more than three user interactions from opening the app to submitting a request. The status tracking screen used WebSocket connections to receive real-time updates without requiring the user to manually refresh the page. The service provider dashboard displayed incoming requests with accept and decline buttons, and showed the requestor's location on an embedded map.

9. Tools and Technology

The system was developed using a modern and well-supported technology stack. React.js was used for the front-end due to its component-based architecture and strong ecosystem. Node.js with Express.js formed the back-end API layer, chosen for its non-blocking I/O model which handles concurrent requests efficiently. MongoDB served as the database, with Mongoose providing schema validation and query support. The Google Maps JavaScript API was integrated for map display and location visualisation. Socket.IO was used for WebSocket-based real-time communication. The entire project was managed using Git, with the repository hosted on GitHub.

10 . Challenges

GPS Accuracy in Indoor Environments: The browser Geolocation API returned significantly inaccurate coordinates when tested indoors. This was partially addressed by prompting users to enable high-accuracy mode and to move outdoors for better signal, and partially through a minimum accuracy threshold that flagged low-quality location readings.

Asynchronous Logic Complexity: The cascading notification system — where providers are notified sequentially with timeouts between attempts — required careful management of asynchronous operations. Early implementations resulted in unhandled promise rejections and race conditions. The solution involved refactoring the notification logic using a proper async queue pattern.

Simultaneous Request Acceptance: A race condition was identified where two providers could simultaneously attempt to accept the same request, leading to inconsistent state. This was resolved by implementing an atomic update operation in MongoDB that only succeeded for the first accepting provider, rejecting subsequent attempts.

Responsive Design for Emergency Use: Since the platform is most likely to be used on a mobile browser in a stressful situation, the interface needed to be functional on small screens without any user confusion. Achieving this required multiple rounds of layout refinement and user feedback.

11.Future Work

Mobile Application Development: A dedicated Android and iOS application would significantly improve accessibility and allow for more reliable push notifications. Native apps can also access device features like background location updates, which would improve the accuracy of provider tracking.

Payment Gateway Integration: Adding an in-app payment option would complete the service transaction cycle within the platform, removing the need for cash exchanges and providing a record of payments for both parties.

User Rating and Review System: A post-service rating system would help maintain quality standards among service providers and give vehicle owners confidence in the providers they are matched with.

Machine Learning for Predictive Assistance: Future versions could incorporate machine learning models that analyse vehicle usage patterns and alert owners to potential breakdown risks before they occur, shifting the system from reactive to proactive assistance.

Multilingual Support: Given the linguistic diversity of India, adding support for regional languages would greatly expand the platform's reach and usability, particularly in rural and semi-urban areas.

12 . Conclusion

The On Road Vehicle Breakdown Assistance System developed in this project demonstrates that the problem of roadside emergencies can be meaningfully addressed through thoughtful application of existing technologies. By combining GPS-based location sharing, proximity-sorted provider matching, and real-time status tracking in a single, accessible platform, the system delivers a solution that is faster, more transparent, and more reliable than anything currently available to the average vehicle owner.

The prototype successfully completes the full service cycle — from breakdown request submission to provider acceptance and status tracking — and has been validated through functional testing and user feedback. The challenges encountered during development, particularly around GPS accuracy, asynchronous logic, and responsive design, were resolved through systematic debugging and iterative refinement, each contributing to a stronger and more robust final product.

This work also serves as a demonstration of the broader potential of location-aware, real-time web applications in emergency assistance contexts. The architecture and methodology developed here are applicable beyond vehicle breakdowns to any scenario where someone in distress needs to be quickly connected with a nearby service provider. With the extensions outlined in the future work section, this platform has the potential to grow into a comprehensive, commercially viable product that makes Indian roads genuinely safer for everyone.

13.Reference

- Sharma, R., et al. (2020). Mobile-Based Emergency Vehicle Assistance Platforms: A Study of Response Time Improvements. Journal of Transportation Technology

- Rao, K., & Mehta, S. (2021). GPS-Integrated Fleet Management and Breakdown Reporting Systems. International Journal of Logistics and Supply Chain Management

- Kumar, A., et al. (2022). Haversine Formula Applications in Location-Aware Mobile Systems.

- Patel, N., & Singh, D. (2023). User Experience Study of Roadside Assistance Applications in the Indian Market.

- Zhang, W., et al. (2022). Real-Time Push Notification Strategies in Service Booking Applications.

- Google Developers. (2023). Maps JavaScript API Documentation.

- Mozilla Developer Network. (2023). Geolocation API.