IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13308

Web Portal For Acting Driver Hiring System

S. Sankar[1], Dr. K. Santhi[2]

Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India[1]

Professor, Department of Information Technology, Dr. N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India[2]

Abstract: The increasing reliance on personal vehicles combined with heavy traffic and long travel distances has created a high demand for reliable, temporary professional drivers. This paper presents a Web Portal for an Acting Driver Hiring System, a professional platform designed to streamline the hiring of temporary drivers in the Coimbatore region. The system features role-based access for Users, Drivers, and Administrators, and utilizes a smart, distance-based driver matching algorithm. Built using React, TypeScript, and Vite, the application simplifies the hiring process by offering real-time location-based matchmaking, comprehensive driver profiles, and automated booking workflows. Experimental results demonstrate the system's effectiveness in calculating proximity using the Haversine formula and generating optimized matches using simplified KNN-like logic. Future enhancements include the integration of live GPS tracking, centralized database management, and mobile application deployment.

Keywords: Acting driver, hiring portal, distance-based matching, Haversine formula, React, TypeScript, K-Nearest Neighbours (KNN), role-based access.

I. INTRODUCTION

Hiring acting drivers traditionally involves contacting local travel agencies, negotiating rates, and relying on word-of-mouth recommendations. This process is often time-consuming, lacks transparency regarding driver experience, and provides no real-time tracking. Recent advancements in web technologies and geolocation services offer new opportunities to automate and modernize this sector.

This paper introduces the Web Portal for an Acting Driver Hiring System, a frontend web application that bridges the gap between vehicle owners and professional drivers. The platform provides a unified interface with three core functionalities tailored to different user roles: User Portal: Allows clients to select pickup locations, vehicle categories, and travel durations to find the nearest available experts. Driver Console: Enables verified drivers to toggle their availability, receive localized booking requests, and manage their network score. Admin Dashboard: Provides administrators with a control centre to oversee expert inventory, monitor operation logs, and track hub status.

The frontend is built with React and Tailwind CSS, ensuring a responsive and highly engaging user experience. The remainder of this paper is organized as follows: Section 2 reviews the literature on travel and driver allocation systems. Section 3 defines the problem statement. Section 4 describes the proposed system architecture. Section 5 details the system workflow. Section 6 presents results and discussion. Section 7 concludes the paper and outlines future research directions.

II. LITERATURE SURVEY

Several existing platforms have addressed transportation logistics, but most focus on standard cab hailing rather than acting driver services. Traditional Cab Platforms: Applications like Uber and Ola provide end-to-end transportation but require the user to hire both the car and the driver. They do not cater to users who own a vehicle but temporarily lack the ability to drive it. Local Driver Agencies: Traditional acting driver services rely on manual dispatching via phone calls. This leads to inefficiencies in driver allocation and lacks background verification transparency. Distance Optimization Models: The integration of mathematical formulas, such as the Haversine formula, to calculate the great-circle distance between two points on a sphere has become standard in modern routing applications. Existing Gaps: There is a clear need for a dedicated, localized platform that handles driver profiling, real-time availability toggling, and algorithmic distance-based matchmaking specifically for personal vehicle owners. Our work leverages React and modern web APIs to deliver this solution efficiently.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13308

III. PROBLEM STATEMENT

Vehicle owners frequently face several challenges when requiring temporary driving assistance:1.Information Asymmetry: Users lack access to a centralized database of verified, available drivers in their immediate vicinity.2.Manual Allocation: Discovering which driver is nearest and available requires manual coordination, which is highly inefficient.3.Lack of Trust: Without a transparent rating system or visible experience metrics, handing over a personal vehicle to a stranger is risky.4.Fragmented Workflow: Users must navigate separate channels for booking, driver details, and tracking. This project aims to develop a unified platform that solves these issues by computing driver proximity automatically, maintaining transparent driver profiles with experience and ratings, and managing the entire trip lifecycle from request to completion.

IV. PROPOSED SYSTEM

4.1 OVERVIEW

The proposed system is a robust Single Page Application (SPA) built with React and TypeScript. It provides a seamless hiring service tailored for the Coimbatore region, covering specific zones such as Gandhipuram, RS Puram, and Peelamedu. The system dynamically calculates distances and statuses using state management and local storage persistence.

4.2 SYSTEM ARCHITECTURE

The architecture of the Web Portal for Acting Driver Hiring System is designed as a modular framework where the frontend Client Interface, crafted with React and Tailwind CSS, provides a responsive environment for User, Driver, and Admin interactions; this is seamlessly integrated with a State Management and Persistence layer that leverages React hooks and browser local storage to simulate a database-driven experience for managing driver availability and booking histories; central to the system's efficiency is the Matching Engine, a computational utility that ingests geographical coordinates to perform distance-based sorting via the Haversine formula, ensuring that the most relevant and proximal drivers are prioritized for every hiring request.

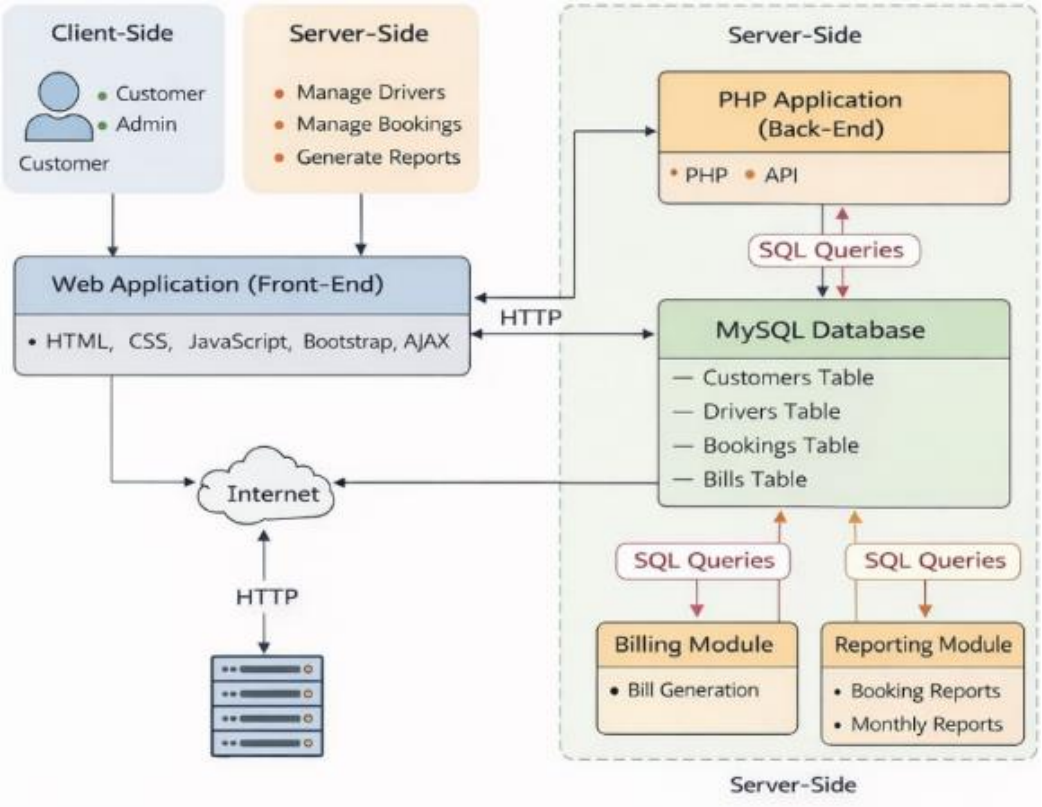


Figure 1: System Architecture Diagram showing Client (Customer/Admin), Web Application (Front-End), PHP Backend Server, MySQL Database, Billing Module, Reporting Module, and Internet Communication.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

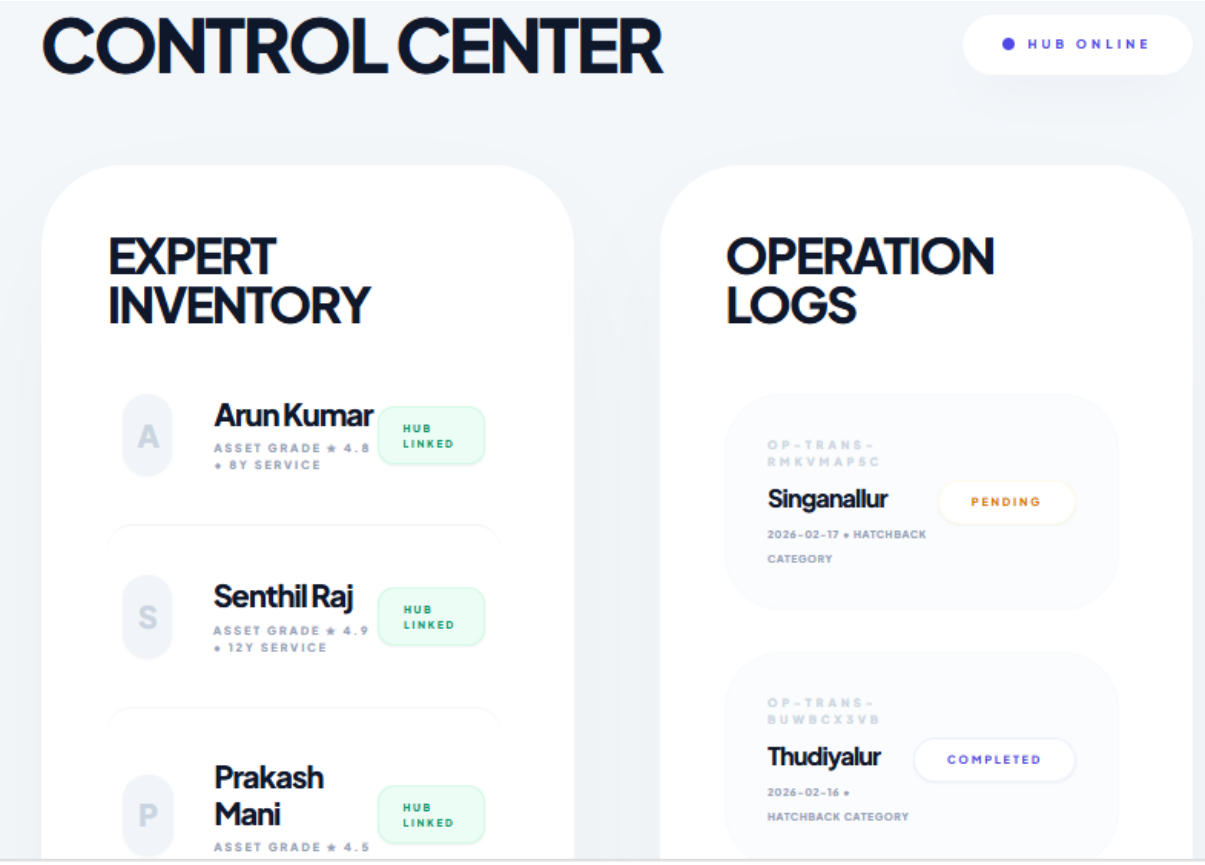
DOI: 10.17148/IARJSET.2026.13308

The frontend of the Driver Booking and Management System is developed using HTML, CSS, Bootstrap, and JavaScript to provide a responsive and user-friendly interface. It allows users to register, login, book drivers, and view booking details. JavaScript is used for form validation and dynamic interactions. The backend is implemented using PHP, which handles HTTP requests, session management, authentication, and business logic. It processes booking data, verifies driver availability, generates bills, and manages reports. All system data is stored in a MySQL database containing tables such as Customers, Drivers, Bookings, and Bills. The backend interacts with the database using SQL queries to insert, update, and retrieve records. The system follows a three-tier architecture consisting of the Presentation Layer (Frontend), Application Layer (PHP Backend), and Data Layer (MySQL Database). This structure ensures secure data handling, centralized management, and efficient driver booking operations.

4.3 MODULES DESCRIPTION

The user authentication module allows customers and administrators to register and log in securely. During registration, passwords are encrypted using hashing techniques before storing them in the MySQL database, and duplicate usernames or email addresses are rejected to maintain data integrity. Upon successful login, user credentials are verified against stored records, and a session is created to maintain authentication throughout the interaction. Role-based access control ensures that administrators and customers access only their respective dashboards.

The driver management module enables the administrator to add, update, delete, and manage driver records. Driver details such as name, contact number, license number, availability status, and assigned bookings are stored in the database. The system automatically updates driver availability when a booking is confirmed. Figure 2 shows the admin dashboard with driver management options.



[Figure 2: Admin dashboard displaying driver management controls and driver list.]

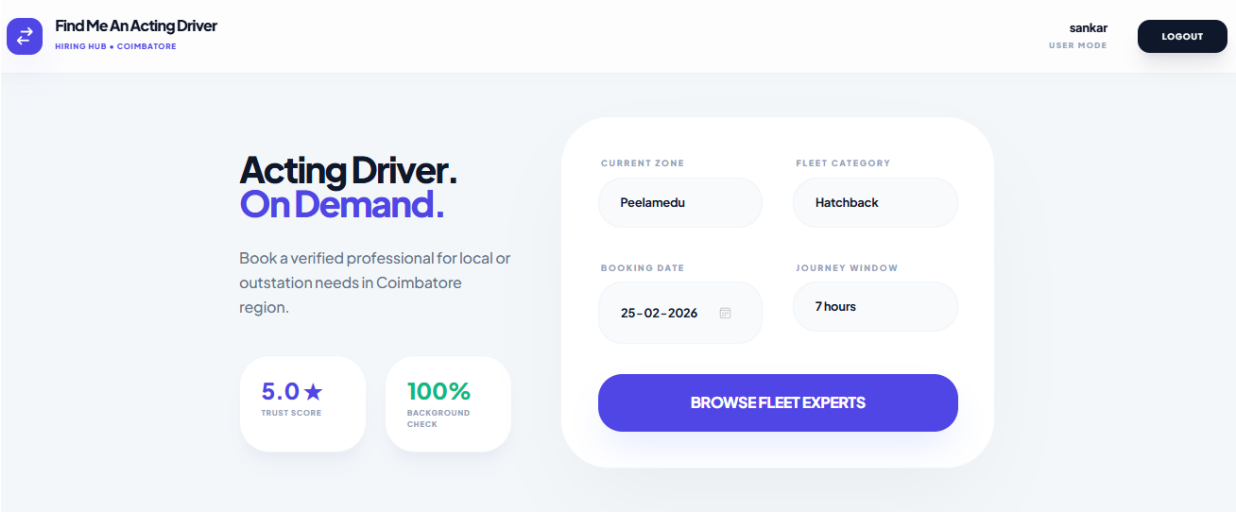
The booking module allows customers to book a driver by entering pickup location, date, duration, and other trip details through an online form. Upon submission, the backend validates the input and checks driver availability in the database. If a driver is available, the booking is confirmed and stored in the Bookings table; otherwise, an appropriate message is displayed. Figure 3 presents the booking form.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

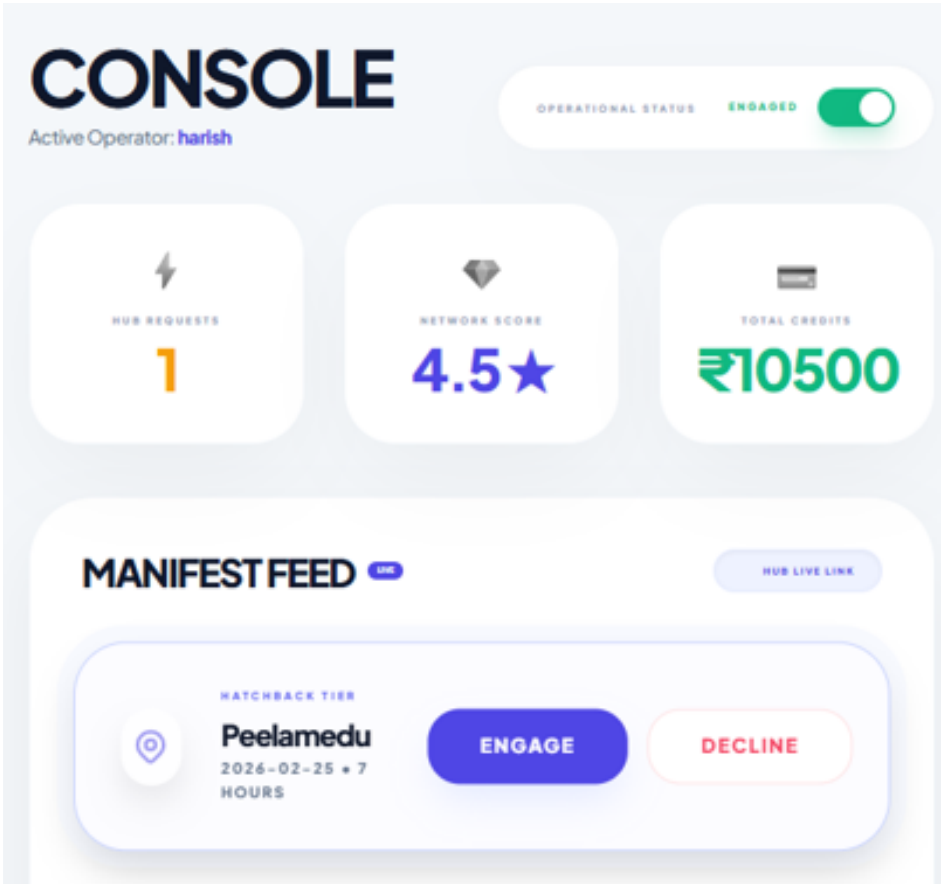
Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13308



[Figure 3: Driver booking form.]

When a customer submits a booking request, the system checks the availability status of drivers stored in the database. If a driver is marked as "Available," the system assigns that driver to the booking and automatically updates the availability status to "Not Available" to prevent double booking. Once the trip is completed, the administrator can update the driver's status back to "Available." Figure 5 shows the drivers login.



[Figure 4: Driver's console.]

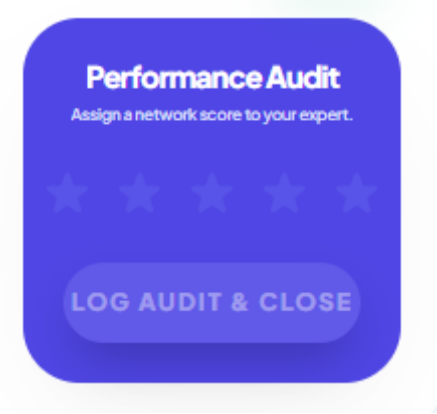
After the completion of a trip, customers are given the option to rate the assigned driver. The rating may be provided in the form of a numerical scale (for example, 1 to 5 stars). The submitted rating details are stored in a dedicated Ratings table in the MySQL database, which includes fields such as Booking ID, Driver ID. Figure 5 shows the rating page for the ride done by the user after the ride.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

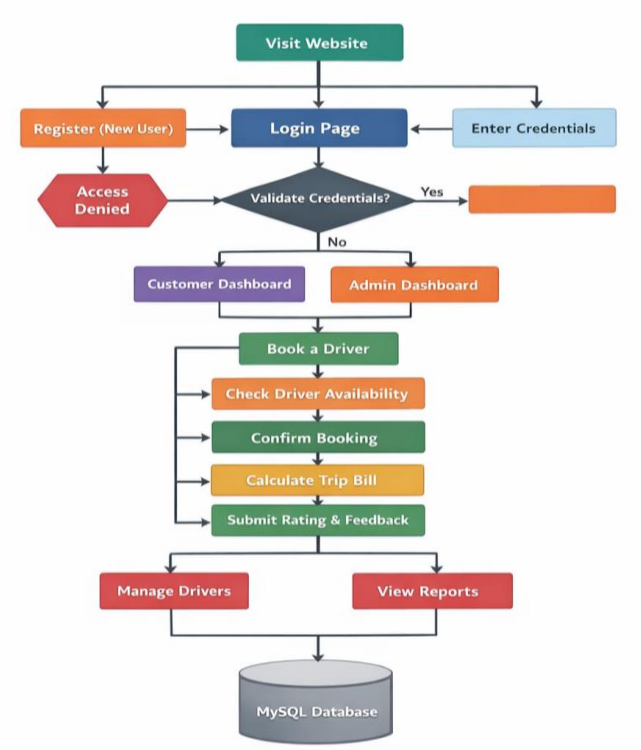
DOI: 10.17148/IARJSET.2026.13308



[Figure 5: shows the rating page for the ride done by the user after the ride.]

V. SYSTEM WORKFLOW

The operational flow of the system begins when a user accesses the application through the web browser and is presented with the login page. New users can register by filling out the registration form with required details such as username and password. Upon successful registration, the user is redirected to the login page. Existing users enter their credentials, which are validated against the MySQL database. If the credentials are valid, a session is created, and the user is redirected to their respective dashboard based on their role (Admin or Customer). After authentication, customers can access the booking module from the dashboard. The user fills in booking details such as pickup location, date, and duration. When the form is submitted, the data is sent via HTTP POST to the backend PHP script. The backend validates the input, checks driver availability in the database, and assigns an available driver if found. The booking details are then stored in the Bookings table, and the driver's availability status is updated accordingly. The administrator can manage drivers, view bookings, monitor ratings, and generate monthly reports through the admin dashboard. All interactions between the frontend and backend occur through HTTP requests, where form data is processed, SQL queries are executed, and results are rendered dynamically on web pages. The data flow within the system follows a structured sequence: user input is sent to the PHP backend, processed according to business logic, stored or retrieved from the MySQL database, and finally displayed on the browser interface.



[Figure 6: System Workflow.]

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13308

VI. IMPLEMENTATION AND RESULTS

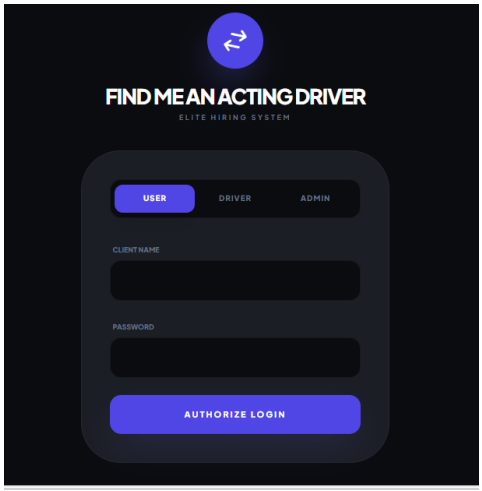
6.1 TECHNOLOGY STACK

The system is implemented using PHP as the server-side scripting language for backend development. The database used is MySQL, which is accessed through PHP using SQL queries for storing and retrieving user, driver, booking, billing, and rating information. The application runs on an Apache Web Server within the XAMPP environment for local development and deployment.

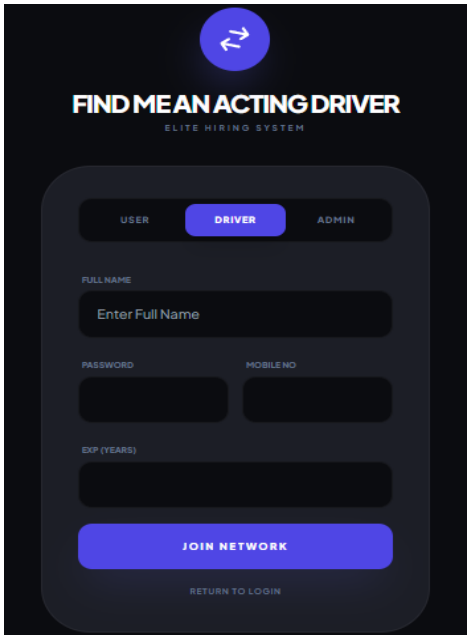
The frontend utilizes HTML5 and CSS3 for designing web pages, along with Bootstrap to ensure a responsive and user-friendly interface. JavaScript is used for client-side validation and dynamic form interactions such as booking confirmation and status updates. Password security is maintained using hashing techniques before storing credentials in the database, ensuring secure authentication and data protection within the system.

6.2 USER INTERFACE

The user interface is designed to be intuitive and visually appealing. The login page (Figure 7) features a gradient background and a cantered form with icons. The registration page (Figure 8) includes a password strength meter to guide users in creating strong passwords. The homepage (Figure 9) home page for user, each with an icon and a brief description, linking to the respective modules.



[Figure 7: Login page.]



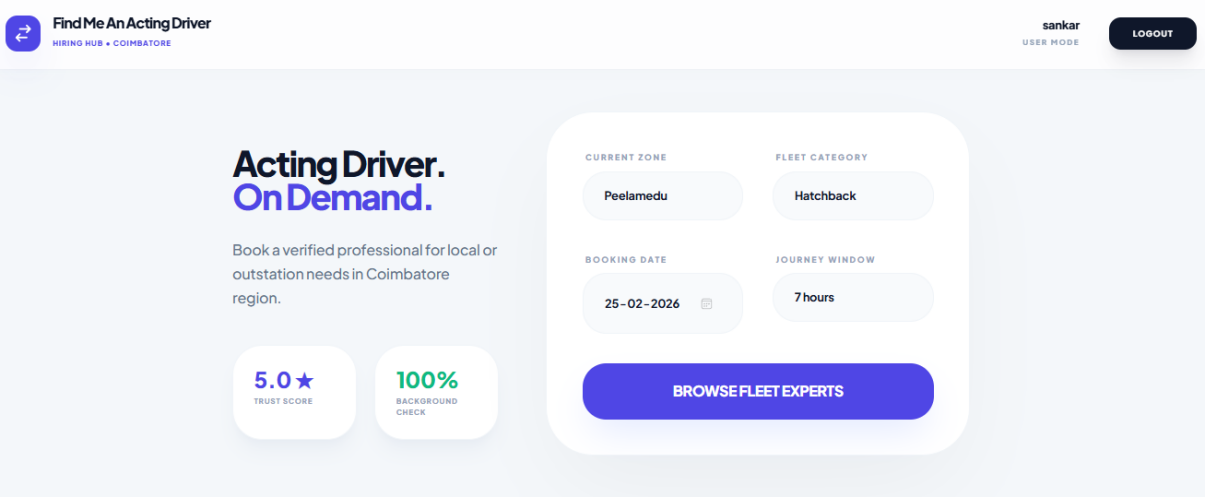
[Figure 8: Registration page with password strength meter]

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.13308



[Figure 9: Homepage for user.]

6.3 SYSTEM TESTING AND SAMPLE OUTPUTS

The system was tested with various inputs to ensure proper functionality of all modules. During user registration, valid details were successfully stored in the database, while duplicate usernames or emails were rejected with appropriate error messages. Login testing confirmed that only valid credentials allowed access to the dashboard, and invalid credentials displayed an “Incorrect Username or Password” message.In the booking module, when a customer entered pickup location, drop location, date, and duration, the system successfully checked driver availability and assigned an available driver. If no driver was available for the selected date, the system displayed a “Driver Not Available” notification. Upon trip completion, the billing module correctly calculated the total fare based on predefined rates and generated a bill summary. The ratings module was tested by submitting sample feedback with a 4-star and 5-star rating. The system stored the ratings in the database and updated the driver’s average rating correctly. The admin module successfully displayed driver details, booking records, revenue summaries, and rating information in structured format.

6.4 PERFORMANCE EVALUATION

System performance was evaluated under normal usage conditions. Database operations such as booking insertion, driver status updates, and bill generation were executed instantly due to optimized SQL queries and small-scale data storage. Page loading times remained responsive on both desktop and mobile browsers. Session management functioned reliably without unauthorized access. Driver assignment logic prevented double booking by automatically updating availability status. Overall, the application demonstrated stable performance, efficient data processing, and smooth user interaction across all modules.

6.5 DISCUSSION

The system successfully demonstrates a complete web-based Driver Booking and Management solution integrating authentication, booking, billing, rating, and reporting functionalities. The modular design ensures that each component operates independently while maintaining centralized database control. However, certain limitations exist. The system currently operates on a local server environment and does not include online payment gateway integration. Real-time GPS tracking is not implemented, and driver availability updates depend on manual status changes after trip completion. Despite these limitations, the developed prototype effectively fulfils the core objective of managing driver bookings efficiently and provides a strong foundation for future enhancements such as online payments, live tracking, and automated notifications.

VII. CONCLUSION AND FUTURE SCOPE

This project presented a web-based Driver Booking and Management System that integrates user authentication, driver management, booking allocation, billing, ratings, and reporting functionalities within a centralized platform. The system successfully demonstrates automated driver assignment based on availability, secure data handling using password hashing, structured database management through MySQL, and an intuitive user interface built with HTML, CSS, Bootstrap, and JavaScript. The results indicate that the system effectively reduces manual effort in managing driver bookings while improving accuracy, transparency, and operational efficiency. The implemented modules ensure smooth interaction between customers and administrators, providing real-time booking confirmation, automated bill calculation, and structured performance monitoring through ratings and reports. The application proves to be responsive, reliable,

DOI: 10.17148/IARJSET.2026.13308

and suitable for small- to medium-scale transport service providers seeking digital transformation. However, the prototype has certain limitations that open opportunities for future enhancement. Integration of an online payment gateway would enable secure digital transactions. Implementing real-time GPS tracking would allow customers to monitor driver location during trips. Automated SMS or email notifications could improve communication between drivers and customers. Deploying the system on a cloud-based server would enhance scalability and remote accessibility. Additionally, developing a dedicated Android and iOS mobile application would provide greater convenience for users. Incorporating analytics dashboards and performance-based driver ranking systems could further optimize service quality. With these improvements, the system can evolve into a fully scalable commercial solution capable of supporting large transport agencies and ride-service platforms, thereby contributing to efficient and technology-driven transport management.

REFERENCES

[1]. M. A. Jabbar, P. Chandra, and B. L. Deeksha Tulu, “Real-Time Cab Booking System Using GPS and GSM Technology,” International Journal of Computer Applications, vol. 95, no. 25, pp. 1–5, 2014.
[2]. S. A. Shinde and P. B. Mane, “Online Cab Booking System,” International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 5, no. 3, pp. 452–456, 2016.
[3]. R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner’s Approach, 8th ed., New York, NY, USA: McGraw-Hill, 2015.
[4]. I. Sommerville, Software Engineering, 10th ed., Pearson Education, 2016.
[5]. P. J. Deitel and H. M. Deitel, Internet & World Wide Web: How to Program, 5th ed., Pearson, 2012.
[6]. L. Welling and L. Thomson, PHP and MySQL Web Development, 5th ed., Addison-Wesley, 2017.
[7]. R. Nixon, Learning PHP, MySQL & JavaScript, 5th ed., O’Reilly Media, 2018.

