

Driver Hiring System: A Web-Based Solution Using PHP

Prajwal Angalwar¹, Minal Patle², Bhagyashree Kumbhare³, Yamini B. Laxane⁴

^{1,2}Students, Department of MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, Maharashtra, India.

³Professor&HOD, Department of MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, Maharashtra, India.

⁴Professor, Department of MCA, Smt. Radhikatai Pandav College of Engineering, Nagpur, Maharashtra, India.

To Cite this Article: Prajwal Angalwar¹, Minal Patle², Bhagyashree Kumbhare³, Yamini B. Laxane⁴, "Driver Hiring System: A Web-Based Solution Using PHP", Indian Journal of Computer Science and Technology, Volume 04, Issue 02 (May-August 2025), PP: 144-147.

Abstract: The rise of digital platforms has transformed various service industries, including transportation. One of the persistent issues in urban and semi-urban areas is the difficulty in hiring reliable and professional drivers. The traditional systems lack scalability, transparency, and operational efficiency. This paper presents the design and implementation of a dynamic, web-based Driver Hiring System developed using PHP, MySQL, and XAMPP. The system is intended to act as a middle layer between drivers and customers, offering a secure, user-friendly, and automated solution for managing driver hiring requests, registrations, and feedback. It also introduces an administrative interface for effective monitoring and management of the service. The system architecture, functionality, database design, security protocols, and testing mechanisms are comprehensively described to present a complete deployment-ready application.

Keywords: Driver Hiring System, PHP, MySQL, XAMPP, Web Application, User Management, Booking System.

I. INTRODUCTION

In today's fast-paced environment, the need for flexible, on-demand transportation services has become essential. Whether for logistics, private travel, or commercial delivery, individuals and businesses frequently face challenges in accessing trustworthy driver services. Manual driver hiring processes are plagued by inefficiencies, limited reach, lack of verification, and sometimes safety concerns. These limitations highlight the need for a modern, digitized platform that bridges the gap between customers and service providers in the driver hiring ecosystem.

The proposed Driver Hiring System provides a centralized platform that offers real-time driver availability, secure user accounts, and an integrated review mechanism to enhance service accountability. By leveraging PHP and MySQL as its core technologies and utilizing XAMPP for local server configuration, the system aims to be both efficient and cost-effective for small businesses or startups wishing to operate locally or regionally.

II. PROBLEM STATEMENT

Traditional driver hiring services often involve verbal agreements, cash payments, and little to no documentation. This results in several problems including lack of traceability, absence of legal accountability, and the potential for fraudulent behavior. Customers are often uncertain about the qualifications or credibility of drivers, while drivers struggle with inconsistent bookings and income flow. There is a need for a platform that standardizes this interaction, enabling mutual trust and operational transparency through automation and proper data handling.

III. OBJECTIVES

The main goal of this project is to build a web-based driver hiring platform that simplifies the process of booking verified drivers. It should support both end-users and service providers with relevant tools and interfaces. Specific objectives include:

- Creating a secure user authentication and registration system for both customers and drivers.
- Building a real-time booking system with availability management.
- Implementing a driver review and rating feature.
- Designing an administrator control panel for monitoring, approvals, and system management.
- Ensuring data integrity, system scalability, and ease of use across devices.

IV. LITERATURE REVIEW

In recent years, numerous systems have emerged to address the transportation needs of consumers using mobile and web technologies. Popular services like Uber and Ola have revolutionized how people book rides, but these platforms primarily focus on vehicle hire with drivers included, not standalone driver services for personal or commercial vehicles. Several research studies have analyzed ride-hailing apps, focusing on UI/UX design, security, location tracking, and dynamic pricing. However, there exists a research gap in systems specifically built for hiring drivers without vehicles, particularly in the context of small cities and towns. The proposed system differentiates itself by focusing on driver-only services and introducing localized support, affordability, and operational flexibility.

V.METHODOLOGY

The system development followed the waterfall model to ensure structured progress across different stages: requirement analysis, system design, implementation, testing, and deployment. Requirements were gathered through informal interviews and questionnaires with users who frequently hire drivers, as well as with professional drivers and small agency operators. Based on these inputs, system components and database structures were modeled. A three-tier architecture was selected to clearly separate the user interface, business logic, and data layers.

VI.SYSTEM ARCHITECTURE

The architecture of the Driver Hiring System is divided into three primary layers:

1. **Presentation Layer:** This is the user interface accessible through web browsers. It is designed using HTML, CSS, and Bootstrap, with JavaScript enhancing interactivity.
2. **Application Layer:** Built with PHP, this layer handles server-side operations, including session management, user inputs, data processing, and business rules enforcement.
3. **Data Layer:** The MySQL database manages the storage and retrieval of all system data, ensuring relational integrity and transaction management.

This architecture ensures modularity, simplifies debugging, and supports future scalability with minimal structural changes.

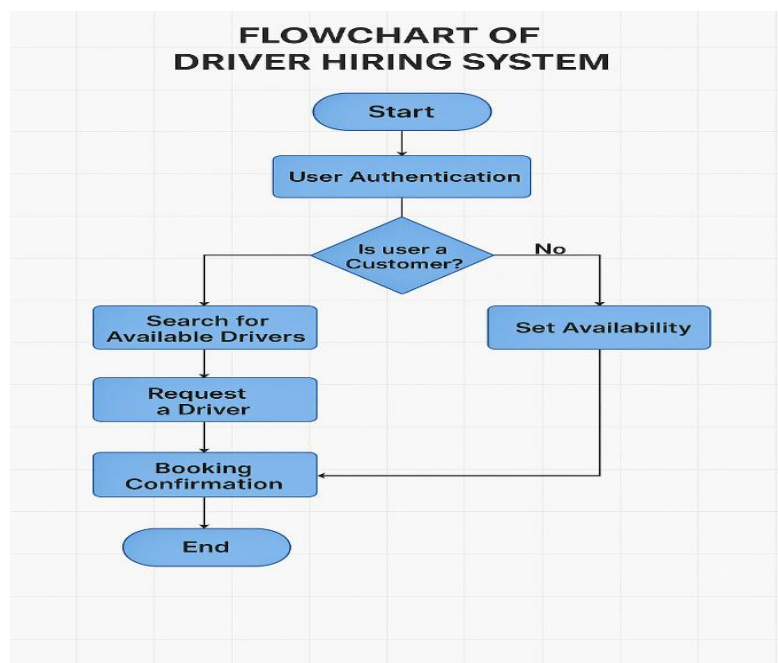


Fig. Flow chart of Driver Hiring System

VII.DATABASE SCHEMA AND DESIGN

The database is implemented using MySQL. It consists of multiple interrelated tables, including:

- **Users:** Contains fields such as user id, full name, email, hashed password, phone, and address.
- **Drivers:** Includes driver_ id, name, license number, vehicle type, availability status, rating, and documents.
- **Bookings:** Links users and drivers with fields like booking_ id, user_ id, driver_ id, booking date, pickup time, status, and feedback.
- **Reviews:** Maintains customer reviews with fields for review_ id, driver_ id, user_ id, rating, and comment.

Normalization principles are applied to reduce redundancy and maintain consistency. Primary and foreign keys are appropriately defined to ensure referential integrity.

VIII.MODULE DESCRIPTION

The system is divided into three functional modules, each tailored to its respective user type:

Customer Module enables users to register, log in, browse available drivers, make bookings, cancel or reschedule appointments, and provide feedback after service completion.

Driver Module allows drivers to register and submit verification documents. Upon approval by the administrator, drivers can log in, update their availability, view booking requests, accept or reject them, and monitor customer feedback.

Admin Module is responsible for approving driver accounts, managing user data, monitoring system activity, addressing customer complaints, and generating analytical reports regarding driver performance and customer engagement.

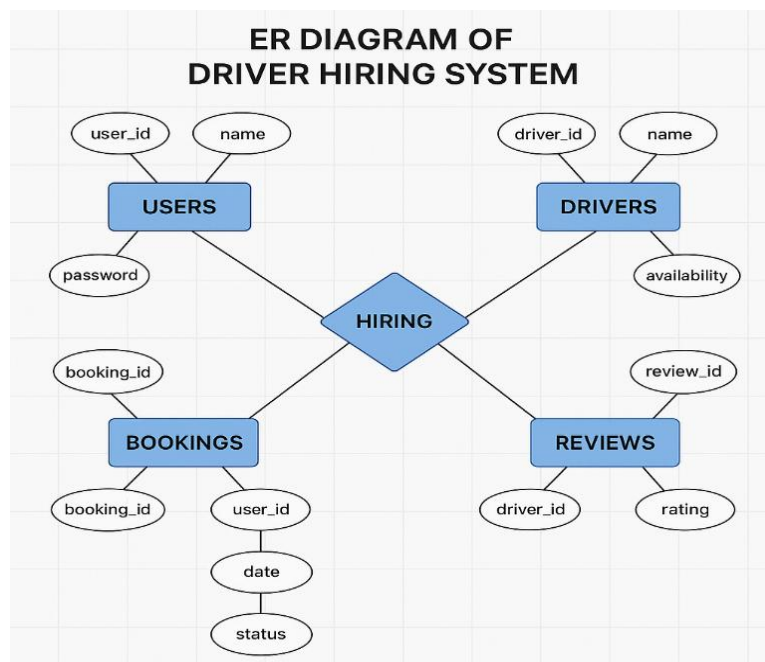


Fig. ER Diagram of Driver Hiring System

IX.SECURITY IMPLEMENTATION

Security is a central aspect of the system design. User passwords are hashed using the bcrypt algorithm to ensure they are not stored in plaintext. PHP's prepared statements are used throughout the application to prevent SQL injection attacks. Additionally, form inputs are validated both on the client and server sides to avoid malicious injections and broken workflows. Access control mechanisms ensure that only authorized users can access or modify certain parts of the system.

X.SYSTEM TESTING

The system underwent both functional and non-functional testing. Functional testing included test cases for user login, booking, driver acceptance, and admin approval flows. Each module was tested independently and as part of an integrated system. Non-functional testing focused on performance under simulated load, database response times, and system usability. The application passed cross-browser compatibility tests and was validated for responsiveness on different devices using browser emulators.

XI.COMPARATIVE ANALYSIS

Unlike commercial platforms that combine vehicle and driver booking, this system is exclusively driver-focused, making it suitable for users who own vehicles but need professional drivers. Compared to traditional offline hiring services, the system provides transparency, accountability, and historical record-keeping. When compared with larger platforms, it is more localized and cost-effective, and can be customized easily for specific business models, such as hourly hires, corporate engagements, or event-based services.

XII.DEPLOYMENT STRATEGY

Although developed and tested using XAMPP locally, the system can be deployed to a cloud server or hosting platform supporting PHP and MySQL, such as Hostinger, Go Daddy, or AWS with a LAMP stack. Basic configurations like .htaccess rewrite rules and SSL integration can be applied during production deployment. A domain name and mail server integration would enable notifications and account confirmations.

XIII.REAL-WORLD APPLICATIONS

This system can be deployed in urban centers, especially in developing countries where ride-hailing apps are less prevalent. It can also be used by corporate entities that regularly require drivers for employees or logistics. Event planners and tourism agencies could benefit by offering chauffeur services through this platform. Additionally, it can serve elderly individuals or persons with disabilities who need regular driving assistance.

XIV.FUTURE ENHANCEMENTS

Planned enhancements include GPS integration using Google Maps API to allow real-time tracking and proximity filtering of drivers. A native mobile application is proposed to improve accessibility for both users and drivers. Machine learning features could be introduced to suggest drivers based on past user preferences and service ratings. Finally, a wallet system and third-party payment gateways will streamline financial transactions.

XV.CONCLUSION

The Driver Hiring System offers a much-needed digital solution to a traditionally inefficient process. By leveraging commonly used web technologies such as PHP and MySQL, and utilizing XAMPP for development, the system provides a secure, reliable, and scalable approach to driver booking. It addresses key concerns such as authentication, data management, and system transparency. With further enhancements, it has the potential to scale into a fully-fledged commercial platform capable of serving a broad demographic.

References

1. *PHP Documentation* – <https://www.php.net>
2. *MySQL Official Documentation* – <https://dev.mysql.com>
3. *XAMPP Official Site* – <https://www.apachefriends.org>
4. *Mozilla Developer Network* – <https://developer.mozilla.org>
5. *W3Schools Web Development Tutorials* – <https://www.w3schools.com>
6. *Stripe Developer Documentation*. [Online]. Available: <https://stripe.com/docs>