

TRAVELWHEELS: AN INTELLIGENT WEB-BASED TRAVEL VEHICLE AND DRIVER RENTAL MANAGEMENT SYSTEM

KARTHIKEYAN G

Master of Computer Applications, Nehru Institute of Information Technology and Management, Tamil Nadu, India .
Reg. No: 721524622028 | Batch: 2024 –2026 – AUTHOR –Mr. T. Mr. T. Thiyagarajan M.Sc App. Mathematics.,
[Department of computer Applications]

Abstract

The Travel Vehicle and Driver Rental System, branded as TravelWheels, is an advanced full-stack web application developed to efficiently manage and automate the complete lifecycle of vehicle rental operations. In the modern travel industry, the growing demand for on-demand transportation services has made manual booking and management systems highly challenging, time-consuming, and error-prone. Travel agencies, vehicle rental businesses, and customers alike face significant difficulties in managing vehicle availability, booking schedules, pricing, and driver assignments through traditional methods. This project addresses these challenges by providing an intelligent, automated, and user-friendly web-based solution built on a dual-portal architecture consisting of a User Portal for customers and an Admin Portal for fleet management. The User Portal allows registered users to browse the vehicle fleet with advanced search and filter capabilities, book vehicles with a live price calculator, track booking status in real time, manage their profile, and cancel bookings when needed. The Admin Portal provides a comprehensive management dashboard with real-time business analytics, vehicle and driver management with image upload support, booking approval and status lifecycle management, user account oversight, and detailed revenue reports powered by Chart.js visualizations. The system is built using PHP with MySQL on the backend and Tailwind CSS with JavaScript on the frontend, and it significantly reduces manual effort, saves

operational time, and improves overall business efficiency for travel agencies, rental businesses, administrators, and customers.

Keywords—Vehicle Rental Management, Booking Engine, PHP, MySQL, Conflict Detection, Admin Dashboard, Chart.js, Web Application, Dual-Portal Architecture

I. Introduction

In today's fast-paced digital era, the travel and transportation industry is undergoing a rapid transformation driven by technology. The increasing demand for comfortable, reliable, and on-demand travel services has created a significant need for automated, web-based vehicle rental and booking platforms. Traditional methods such as manual phone bookings, paper-based record keeping, and offline scheduling are no longer adequate to meet the expectations of modern customers and the operational needs of growing travel businesses.

Travel agencies and vehicle rental companies deal with a complex set of daily challenges, including managing vehicle availability across multiple date ranges, assigning verified drivers to bookings, tracking booking statuses from pending to completion, handling customer communications, and generating revenue reports. Without a centralized and automated system, these operations are time-consuming, inefficient, and highly prone to human errors such as double-booking, incorrect pricing, and delayed responses.

The TravelWheels system is designed as a comprehensive full-stack web application built using PHP on the backend with MySQL as the database engine, and Tailwind CSS with JavaScript on the frontend, to address all of these challenges through a unified platform serving both customers and administrators with purpose-built interfaces, powerful backend logic, and a modern, responsive design.

II. Problem Statement and Related Work

The travel and vehicle rental industry continues to rely heavily on traditional, manual booking processes managed through phone calls, spreadsheets, and paper records, leading to numerous operational inefficiencies. Existing systems suffer from manual booking processes that are inefficient and prone to miscommunication, a lack of real-time availability information that often leads to double-bookings, the absence of a centralized management dashboard that scatters information across documents and spreadsheets, no structured digital booking approval workflow, and inefficient manual revenue and performance tracking.

Basic spreadsheet-based management using tools such as Microsoft Excel or Google Sheets represents a step up from purely paper-based systems, but still lacks the dynamic features needed for real-time availability checking, automatic price calculation, status workflow management, and multi-user concurrent access, and has no built-in authentication or role separation for handling sensitive customer and financial data.

The development of TravelWheels draws on established literature in web application development, database management, interface design, and security engineering, including foundational works on PHP and MySQL web development, the OWASP Top Ten security risk framework, utility-first CSS design philosophy,

and design science methodology for information systems, all of which directly informed the system's backend architecture, security implementation, and iterative development approach.

III. Proposed System Overview

The proposed system, TravelWheels, introduces a modern web-based vehicle rental management platform that transforms traditional manual operations into an intelligent, automated, and scalable digital system. It operates as an integrated two-portal application combining customer-facing booking features with comprehensive administrative management tools. The application allows users to analyze available vehicles, apply multi-criteria filters, receive automatic noise-free date-conflict checking, and place bookings through a real-time price calculator, while administrators gain full visibility into fleet operations, booking workflows, and revenue performance.

Key Features of the Proposed System

- **Dual-Portal Architecture:** Separate, purpose-built interfaces for customers (User Portal) and administrators (Admin Portal) with independent role-based authentication.
- **Smart Vehicle Discovery:** Multi-criteria vehicle search and filtering by type, passenger capacity, price range, and date-based availability with automatic conflict exclusion.
- **Real-Time Booking Engine:** Server-side booking processor with date validation, conflict detection, price verification, and flash confirmation messaging.
- **Admin Fleet Management:** Complete vehicle CRUD operations with image upload, driver assignment, availability toggling, and deletion.
- **Booking Lifecycle Management:** Full status workflow from pending through

approved, rejected, completed, or cancelled, with admin action controls.

- **Business Analytics:** Interactive Chart.js dashboards showing monthly revenue trends, booking status distribution, vehicle type performance, and top vehicle rankings.
- **Security-First Design:** bcrypt password hashing, PDO prepared statements, session isolation, file upload validation, and server-side sanitization throughout.

This tool has applications across domains such as customer-facing rental booking platforms, fleet and driver management for travel agencies, and revenue and performance analytics for rental business owners. The final output is a lightweight, fully responsive, and self-contained web application deployable on any standard PHP and MySQL hosting environment, making it an ideal solution for travel agencies, administrators, and customers.

IV. Module Description

A. Authentication Module

The Authentication Module manages the complete identity and access control lifecycle for both users and administrators. It handles user registration with email uniqueness checks and password confirmation, secure login using bcrypt hash verification, separate admin login with dedicated session variables, and reusable authentication guard functions that protect restricted pages by redirecting unauthenticated visitors to the appropriate login screen.

B. Vehicle Management Module

This module handles all operations related to the vehicle fleet, accessible through both the admin portal and the user-facing vehicle listing. It supports dynamic vehicle listing with filter parameters such as type, capacity, price range, and date availability using overlap-exclusion subqueries, an admin vehicle grid with availability toggling and cascading deletion, and

vehicle addition and editing forms that include validated image uploads with file-extension whitelisting and size limits.

C. Booking Engine Module

The Booking Engine is the most critical module, responsible for creating, validating, and managing booking records throughout their complete lifecycle. The booking form provides a vehicle selector, location inputs, date pickers, traveler count, and special requests, with a JavaScript price calculator that updates the summary panel instantly. The server-side booking processor enforces authentication, validates all required fields and date logic, calculates the number of rental days, fetches vehicle pricing directly from the database, runs a conflict-detection query, and inserts the booking record with a pending status. Administrators can transition bookings from pending to approved, rejected, or completed, while users can cancel pending bookings.

D. User Dashboard Module

The User Dashboard Module provides customers with a personalized view of their rental history and profile, including summary statistics such as total bookings, upcoming trips, and amount spent, a detailed booking history page with status filter tabs, and profile management for updating contact details and optionally changing the account password.

E. Admin Control Panel Module

The Admin Control Panel provides comprehensive business management through a consistent sidebar layout. It includes a real-time dashboard with statistic cards and a combined bar and line chart for monthly bookings and revenue, a booking management page with search and status filters, a user management page listing registered users with booking counts and total spending, and a reports page with revenue trend charts, a booking status distribution chart, a

vehicle type comparison chart, and a top-performing vehicles table.

F. Search and Availability Engine

The Search and Availability Engine powers the intelligent vehicle discovery experience on the vehicle listing page. It accepts multiple filter parameters and dynamically constructs a parameterized SQL query combining conditions for each active filter. The date-availability component uses a subquery to identify vehicles with overlapping approved or pending bookings and excludes them from the results, ensuring customers only see vehicles that are genuinely available for their requested travel dates.

V. Methodology

The development of TravelWheels followed a structured Iterative Development Methodology combined with a modular architecture design approach, chosen because the system involves multiple interconnected components — user authentication, vehicle management, booking engine, admin control panel, and reporting — each requiring independent development, testing, and integration.

Phase 1: Requirements Analysis and Planning

This phase involved identifying the two primary user types, defining the complete feature set for each portal, designing the database schema, planning the technology stack, and mapping all user interactions through use-case analysis.

Phase 2: Database Design and Backend Architecture

The database schema was designed using MySQL with five core tables — admins, users, vehicles, bookings, and blocked_dates — with foreign key constraints and cascade deletes to maintain data integrity. The PHP backend architecture was established with a central configuration file providing PDO database

connectivity, session management, and authentication guards.

Phase 3: User Portal Development

This phase focused on developing all user-facing pages, including the landing page, vehicle listing with filtering, booking form with live price calculation, user dashboard, booking history, and profile management, with JavaScript powering client-side interactions such as the price calculator and filter tab switching.

Phase 4: Admin Portal Development

The admin portal was developed with a distinct visual identity and a shared sidebar layout, covering the admin dashboard with analytics, vehicle CRUD operations with image upload, booking management with approval actions, user management, and comprehensive reports, each protected by an administrator authentication guard.

Phase 5: Testing, Validation, and Refinement

The final phase involved comprehensive unit, integration, and system-level testing of the booking workflow, along with security testing to validate SQL injection prevention, XSS protection, and authentication mechanisms, and performance evaluation for response time and query efficiency.

VI. Requirements Specification

The hardware requirements for running TravelWheels effectively include an Intel Core i3 processor or higher, 8 GB of RAM, 512 GB of SSD storage, a 14-inch monitor with a minimum resolution of 1366×768, and standard input devices.

The software requirements include Windows, Linux, or macOS as the operating system, Apache 2.4 or higher (via XAMPP) as the web server, PHP 8.0 as the backend language, MySQL 5.7 as the database, Tailwind CSS v3 for frontend

styling, Chart.js v4.4.1 for data visualization, Font Awesome 6.5 for icons, and VS Code, Notepad++, or PhpStorm as the development environment.

VII. System Implementation

System implementation translated the booking-engine and fleet-management design into a fully functional, user-friendly PHP and MySQL-based web application. The environment setup involved installing core libraries including PDO for database access, PHP's password hashing functions, and the Chart.js and Tailwind CSS front-end dependencies via CDN, all developed using PHP 8.x in VS Code.

Module development proceeded functionally: a Data Ingestion and Vehicle Listing module for browsing the fleet, a Booking Engine handling validation and conflict detection, an Admin Control Panel for fleet and booking oversight, and a Security Layer enforcing bcrypt hashing, prepared statements, and file-upload validation throughout. The Streamlit-style real-time feel was instead achieved through a fully server-rendered PHP interface enhanced with JavaScript widgets, including file uploaders with image previews, sidebar navigation panels, and a live feedback dashboard for booking statistics.

Testing and debugging covered unit and integration testing of the filtering and booking modules, verification of edge cases such as corrupted uploads and overlapping date ranges, and responsiveness testing across multiple screen sizes. The application can be deployed locally through a standard PHP development server or hosted on any shared PHP and MySQL web hosting environment for production access via a web browser.

VIII. Results and Testing

System testing followed the V-Model, in which each development phase has a corresponding

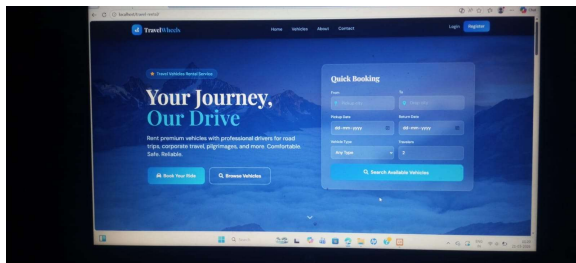
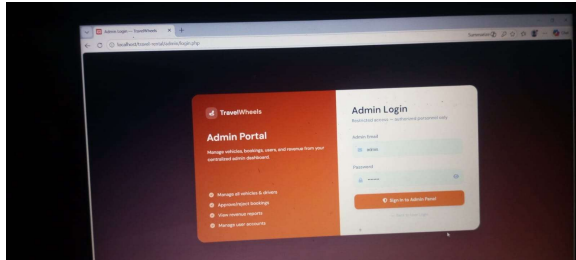
testing phase to ensure systematic validation and early bug detection. A multi-layered testing approach covering unit, integration, system, security, and user-acceptance testing was adopted.

Unit testing validated the correct behaviour of individual functions such as input sanitization, currency formatting, and image-upload validation, all of which behaved correctly under both valid and invalid input scenarios. Integration testing confirmed that registration and login correctly created sessions, that date-based vehicle filtering excluded already-booked vehicles, that booking submissions created accurate database records, and that admin approval actions correctly propagated status changes to the user view.

System testing evaluated complete end-to-end workflows, confirming that bookings were generated with a pending status, that overlapping date conflicts were reliably blocked, and that the application remained responsive across mobile and desktop screen sizes. Security testing confirmed that SQL injection and cross-site scripting attempts were neutralized through prepared statements and output sanitization, that unauthorized admin access attempts were blocked by session guards, and that client-side price manipulation attempts were mitigated through server-side price verification.

User acceptance testing conducted with students and travel agency staff found the vehicle discovery interface clear and easy to navigate, the real-time price calculator a strongly appreciated feature, the colour-coded booking status tracking informative, the admin sidebar navigation logical and efficient, and the responsive design consistent across all tested screen sizes.

SAMPLE OUTPUT



IX. Conclusion

The Travel Vehicle and Driver Rental System, TravelWheels, has been successfully designed, developed, and tested as a comprehensive full-stack web application that addresses the real-world challenges of modern vehicle rental management. The system effectively transforms traditional, manual booking processes into an intelligent, automated, and scalable digital platform that benefits both customers and business administrators.

The system demonstrates complete booking lifecycle automation through its intelligent date-conflict detection engine, a dual-portal architecture that provides each user type with a purpose-built interface, robust security implementation including bcrypt hashing and prepared statements, a fully responsive design across all tested device sizes, and a business

intelligence layer that generates accurate, interactive analytics for administrators.

Overall, TravelWheels stands as a scalable, cost-effective, and efficient tool that addresses the modern challenges of vehicle rental businesses, travel agencies, and independent operators. The project successfully meets all its initial objectives, delivering a high-performance application that is ready for real-world deployment. Future enhancements identified for the system include online payment gateway integration, automated email and SMS notifications, a dedicated mobile application, and cloud-based deployment for improved scalability.

References

- [1] L. Welling and L. Thomson, PHP and MySQL Web Development, 5th ed. Boston, MA: Addison-Wesley Professional, 2017.
- [2] R. Nixon, Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, 6th ed. Sebastopol, CA: O'Reilly Media, 2021.
- [3] OWASP Foundation, "OWASP Top Ten: The Ten Most Critical Web Application Security Risks," 2021. [Online]. Available: <https://owasp.org/Top10/>
- [4] PHP Documentation, "PHP Manual — PDO, password_hash, Sessions," 2024. [Online]. Available: <https://www.php.net/manual>
- [5] MySQL Documentation, "MySQL 8.0 Reference Manual," 2024. [Online]. Available: <https://dev.mysql.com/doc>
- [6] Tailwind CSS Documentation, "Tailwind CSS Official Guide," 2024. [Online]. Available: <https://tailwindcss.com/docs>
- [7] Chart.js Contributors, "Chart.js Documentation v4," 2023. [Online]. Available: <https://www.chartjs.org/docs/latest/>
- [8] Mozilla Developer Network, "JavaScript Documentation — MDN Web Docs," 2024. [Online]. Available: <https://developer.mozilla.org>
- [9] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, vol. 13, no. 6, pp. 377–387, 1970.