Tribhuvan University

Faculty of Humanities and Social Sciences

A Project Report On

MULTIVENDOR VEHICLE RENTAL SYSTEM

Submitted to

Department of Computer Application

Itahari Namuna College

In partial fulfillment of the requirements for the degree of Bachelors in Computer Application

Submitted By

Utsav Satyal (6-2-1152-49-2022)

Nischal Bhattarai (6-2-1152-33-2022)

Under the Supervision of

Saroj Ojha

Tribhuvan University

Faculty of Humanities and Social Sciences

Itahari Namuna College

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Utsav Satyal and Nischal Bhattarai entitled “**MULTIVENDOR VEHICLE RENTAL SYSTEM”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is thoroughly researched and well executed, demonstrating the student's technical proficiency and understanding of the subject matter. I highly recommend it for the final evaluation.

Saroj Ojha

Project Supervisor

Itahari Namuna College

Itahari-8, Sunsari

Tribhuvan University

Faculty of Humanities and Social Sciences

Itahari Namuna College

LETTER OF APPROVAL

This is to certify that this project prepared by Utsav Satyal and Nischal Bhattarai entitled “**MULTIVENDOR VEHICLE RENTAL SYSTEM”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Saroj Ojha
Supervisor
Itahari Namuna College

Nabin Shrestha
HOD/Coordinator
Itahari Namuna College

Internal Examiner
Itahari Namuna College

External Examiner
Tribhuwan University

ACKNOWLEDGEMENT

The success and final outcomes of this project requires a lot of guidance and assistance from many people and are very fortunate to have this all through the completion of this project. We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this project. We are very glad to express our deepest sense of gratitude and sincere thanks to our supervisor Mr. Saroj Ojha and project teacher Mr. Ashish Shrestha whose contribution in simulating suggestions and encouragement held us in completing our project.

We are also thankful to Mr. Nabin Shrestha, HOD, Itahari Namuna College and Mr. Dharma Neupane, Principal. Itahari Namuna College for encouraging and guiding us throughout the completion of this project. All useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

Last but not the least, many thanks go to our teachers, friends, guardians who directly or Indirectly helped us in achieving the goal. We would like to appreciate all the guidance and support which helped in the completion of this project. Special thanks to their comment and advice.

ABSTRACT

A Multi-Vendor Vehicle Rental System (MVVRS) is a web-based application designed to facilitate the renting and leasing of vehicles through a centralized platform. This system allows multiple vehicle owners to list their vehicles for rent while enabling customers to browse, book, and manage their rentals conveniently. The platform provides a seamless experience for both vendors and customers by streamlining the vehicle rental process with automated bookings, real-time availability, and secure payment options.

Traditionally, renting a vehicle involves visiting multiple rental agencies, negotiating terms, and handling paperwork manually, which is both time-consuming and inefficient. Likewise, for vehicle owners, managing bookings, tracking availability, and handling transactions manually increases the risk of errors, lost revenue, and mismanagement of records. The absence of a structured system leads to difficulties in managing customer demands, pricing, and service quality.

The Multi-Vendor Vehicle Rental System offers a comprehensive solution by digitizing and automating the rental process. It ensures a user-friendly interface where customers can filter vehicles based on preferences such as type, availability, and price, while vendors can efficiently manage bookings, availability, and earnings. The system enhances operational efficiency, minimizes human errors, and saves both time and resources. Its simple, intuitive design ensures reliability, security, and ease of use, making vehicle rentals more accessible and efficient.

Keywords: *Customer Management, Multi-Vendor, Online Booking, Secure Transactions Vehicle Rental*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DBMS	Database Management System
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
JS	JavaScript
JWT	JSON Web Token
MVC	Model-View-Controller
MVVRS	Multi-Vendor Vehicle Rental System
RDBMS	Relational Database Management System
SEO	Search Engine Optimization
SQL	Structured Query Language
TCID	Test Case Identity
UI/UX	User Interface / User Experience

CHAPTER 1: INTRODUCTION

1.1. Introduction

A Multi-Vendor Vehicle Rental System is a platform that connects vehicle owners with users looking to rent cars, bikes, or other vehicles. With a user-friendly interface, owners can easily list their vehicles and set part-time hourly availability for a specific date , time or for recurring days, while users can browse, book, and pay different vehicles from different owners through the integrated Khalti e-wallet. The platform includes user authentication, document verification for owners, and secure payment processing, ensuring a seamless and trustworthy experience. Whether for a few hours or days, MVVRS makes vehicle rentals simple and efficient.

1.2. Background and Significance

In today’s fast-paced world, flexibility and convenience are key to mobility. With the increasing number of individuals being concerned about the increasing numbers of private vehicles leading to higher carbon emmissions and seeking temporary vehicle rentals for short trips, commutes, or business purposes, a reliable platform is essential to systemconnect vehicle owners with renters. MVVRS fills this gap by creating a seamless and secure environment for both parties. This multi-vendor marketplace not only offers users the ability to rent vehicles easily but also allows owners to maximize their vehicle utilization by offering part-time rentals. The platform’s integration of Khalti for payments ensures a smooth and trusted transaction process, making MVVRS an indispensable solution in the vehicle rental industry.

1.3. Problem Statement

- The vehicle rental industry in Nepal lacks a centralized platform where vehicle owners can list their vehicles and set availability based on their schedules.
- Most rental services only offer full-day rentals, limiting flexibility for users who need vehicles for short-term or part-time use.
- Manual or outdated systems create inefficiencies in booking, tracking, and managing vehicle rentals, reducing overall effectiveness.
- Traditional rental industry relies in human record keeping.

1.4. Objectives

- To develop a system that allows multiple vehicle owners to rent out their vehicles for part-time or full-time use with specific and recurring schedules.
- To enable users to book various vehicles on a pay-per-hour basis from various vendors at a centralized place.

1.5. Scope and limitation

1.5.1. Scope

- Enables vehicle owners to list, manage, and rent out their vehicles with flexible availability, including hourly rentals.
- Provides real-time vehicle booking with a pay-per-hour model, reducing idle vehicle time and maximizing earnings.
- A centralized place for the users to book different vehicles from different owners without having to worry about overpaying and security.
- Integrates document verification and authentication to ensure security and trust between renters and owners.

1.5.2. Limitation

- Success depends on vendors actively listing vehicles, especially in less populated areas.
- Managing rental regulations, insurance, and liability policies across different regions could complicate operations.
- A reliable internet connection is necessary for real-time booking and payment processing, which might be an issue.
- Gaining user trust and ensuring consistent service quality can be challenging when building a new platform.

1.6. Development Methodology

The Waterfall model was adopted as the development methodology for this project. The Waterfall model follows a linear-sequential approach, where each phase is completed before moving on to the next. This methodology was chosen because the project had clear and fixed requirements from the beginning, with a well-understood technology stack.

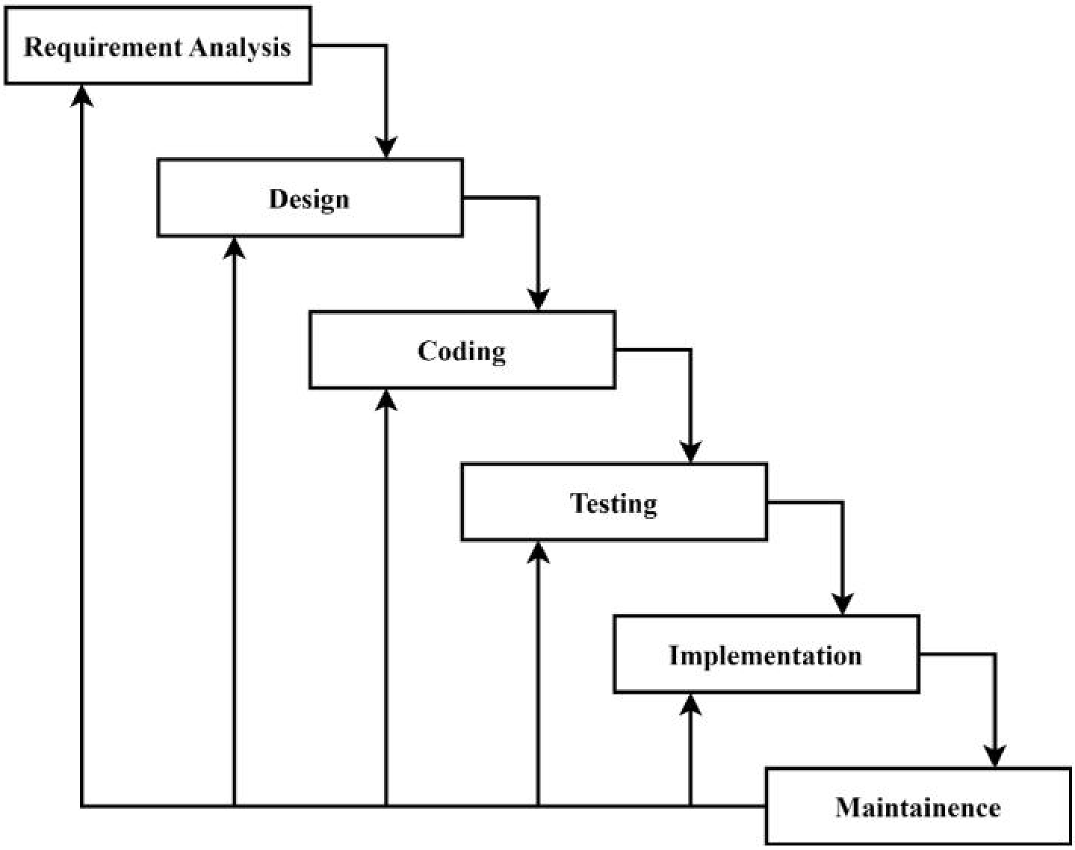


Figure 1.1 Waterfall Model

The development process started with requirement analysis, where we gathered and analyzed the system needs. After that, the system was designed, including database structure and system flow diagrams. Once the design was finalized, we moved to the coding phase and developed the system according to the finalized designs, followed by testing through unit and system testing to ensure the system's functionality. The final phase involved preparing the documentation of the entire project.

CHAPTER 2: LITERATURE REVIEW

2.1. Background Study

In Nepal, the vehicle rental industry has traditionally been dominated by single agencies offering full-day rentals, primarily targeting tourists. These services often come with higher prices and limited flexibility, as rentals are typically available only for full days or weeks. Additionally, most of these agencies operate on a single-vendor model, limiting consumer choice and competition.

While there are a few multi-vendor platforms in Nepal, they often lack integrated online payment systems and primarily serve as intermediaries connecting vehicle owners with renters. This model can lead to inefficiencies and a lack of trust between parties. Moreover, many of these platforms struggle to sustain operations due to limited user engagement and operational challenges.

MVVRS aims to address these issues by providing a centralized, multi-vendor platform that allows vehicle owners to list their vehicles with flexible availability, including hourly rentals. This approach offers consumers a wider range of options and the convenience of booking vehicles for short-term use. The integration of secure payment systems ensures trusted transactions, enhancing the overall user experience.

By offering a user-friendly interface and a variety of vehicles, MVVRS seeks to transform the vehicle rental landscape in Nepal, making it more accessible, flexible, and efficient for both vehicle owners and consumers.

2.2. Literature Review

The growing demand for flexible transportation options has led to the rise of vehicle rental platforms, especially in urban environments where individuals prefer short-term vehicle rentals for business, travel, or leisure. The global sharing economy has been a driving force behind the emergence of such platforms, transforming traditional vehicle rental methods. The evolution of online platforms for renting vehicles has become an essential tool for consumers and owners alike, allowing vehicle owners to earn revenue through underutilized vehicles, while providing customers with convenient, on-demand transportation options.

The rise of multi-vendor platforms has revolutionized various industries, and the vehicle rental sector is no exception. The Multi-Vendor Vehicle Rental System, provides a marketplace where multiple vehicle owners can list their vehicles for rent, offering users the ability to choose from a variety of vehicles based on their needs, preferences, and budget. Several studies have shown that such platforms offer significant benefits to vehicle owners, including maximizing the use of their vehicles by allowing for part-time rentals and reducing idle time. By integrating secure payment gateways such as Khalti, these platforms provide a trusted environment for both renters and owners, eliminating the need for intermediaries and offering lower operational costs than traditional rental agencies [1].

Furthermore, mobile commerce has become a dominant trend in the vehicle rental industry, as more customers now use their smartphones to book vehicles on the go. MVVRS capitalizes on this by providing a seamless, mobile-friendly interface that enhances user experience and convenience [2]. The system's flexibility, allowing users to rent vehicles on an hourly basis rather than by the day, provides a unique competitive edge, catering to consumers' preferences for short-term rentals, which has become increasingly popular [3].

However, like any online platform, MVVRS faces several challenges and limitations. A major issue is the dependence on vendor participation, especially in less populated regions. A wide range of available vehicles is crucial for the platform's success, and without sufficient participation from vendors, the platform may struggle to meet customer demand. Additionally, ensuring compliance with local regulations, insurance requirements, and liability policies for vehicle rentals across different regions can be a complex and time-consuming task. Legal issues, such as managing the rental contracts and protecting both owners and renters, require careful attention to detail.

Another challenge lies in the data privacy concerns associated with collecting large amounts of personal information from users and owners. Research has pointed out that the collection and handling of sensitive customer data must be managed in a secure manner to prevent breaches and maintain user trust [4]. Studies have shown that a user-friendly interface, effective search functionalities, and a smooth booking process are essential factors that influence customer satisfaction and retention in vehicle rental systems [5].

In summary, while the Multi-Vendor Vehicle Rental System presents a significant opportunity for both vehicle owners and users by offering flexibility, convenience, and security, challenges such as vendor participation, legal compliance, and data privacy need to be addressed. By improving user experience and focusing on operational efficiency, MVVRS has the potential to become a leading platform in the evolving vehicle rental market.

Existing System Overview

i. Sasto Rental

Sasto Rental is a prominent online vehicle rental service in Nepal, offering a diverse fleet that includes cars, SUVs, and buses. The platform provides services such as airport pick-up and drop-off, corporate rentals, and sightseeing tours. However, Sasto Rental primarily focuses on full-day rentals, limiting flexibility for users who need vehicles for shorter durations. Additionally, the platform's success relies on vendors joining and actively listing their vehicles. Managing rental regulations, insurance requirements, and liability policies across different locations can also complicate operations. Without robust infrastructure, the system may face performance bottlenecks during peak usage [6].

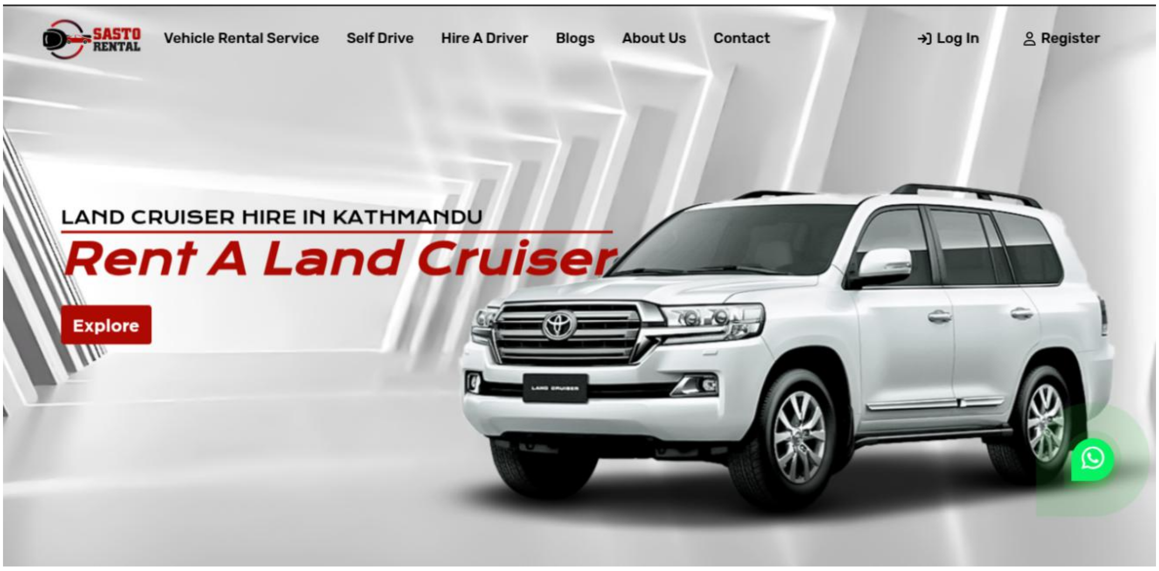
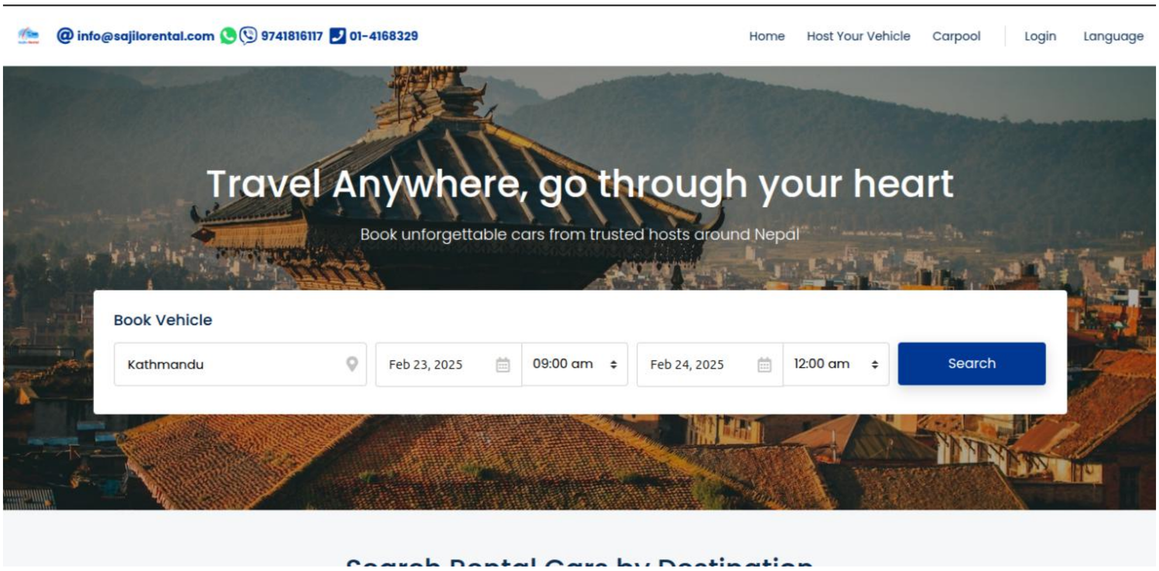


Figure 2.1 Sasto Rental

ii. Sajilo Rental

Sajilo Rental is Nepal’s first car rental marketplace, allowing users to book vehicles directly from local owners. The platform offers a variety of vehicles, including trucks, buses, and luxury cars, catering to diverse needs. Users can choose from over 1,000 vehicles listed across different cities in Nepal. However, Sajilo Rental primarily focuses on full-day rentals, which may not suit users seeking vehicles for shorter periods. The platform's success depends on the active participation of local vehicle owners, and ensuring consistent vehicle quality and availability can be challenging. Additionally, the platform's reliance on local owners may lead to inconsistencies in service quality and vehicle maintenance [7].


@ info@sajilorental.com 9741816117 01-4168329
Home
Host Your Vehicle
Carpool
Login
Language
Travel Anywhere, go through your heart
Book unforgettable cars from trusted hosts around Nepal
Book Vehicle
Search
Search Rental Cars by Destination

Kathmandu | Feb 23, 2025 | 09:00 am | Feb 24, 2025 | 12:00 am

Figure 2.2 Sajilo Rental

iii. Turo

Turo is a leading international peer-to-peer car-sharing platform that allows vehicle owners to rent out their cars to users. The platform offers a wide range of vehicles, from economy cars to luxury models, providing users with diverse options. Turo's dynamic pricing engine adjusts rental rates based on various factors. However, Turo faces challenges with vendor participation, especially in less populated regions. Managing rental regulations, insurance requirements, and liability policies across different locations can also complicate operations. Without robust infrastructure, the system may face performance bottlenecks during peak usage or as the user base grows [8].

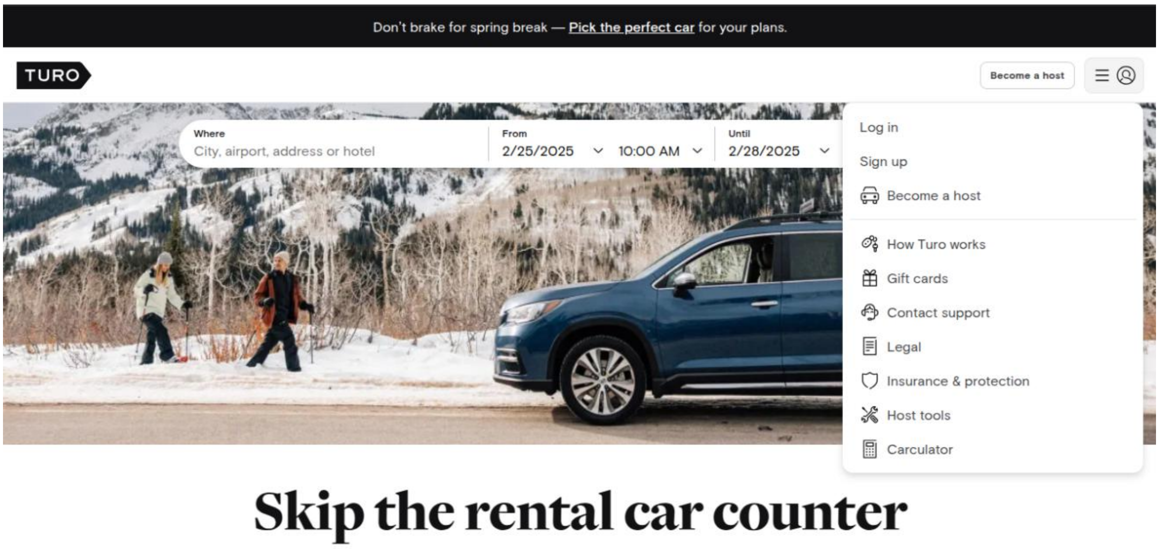
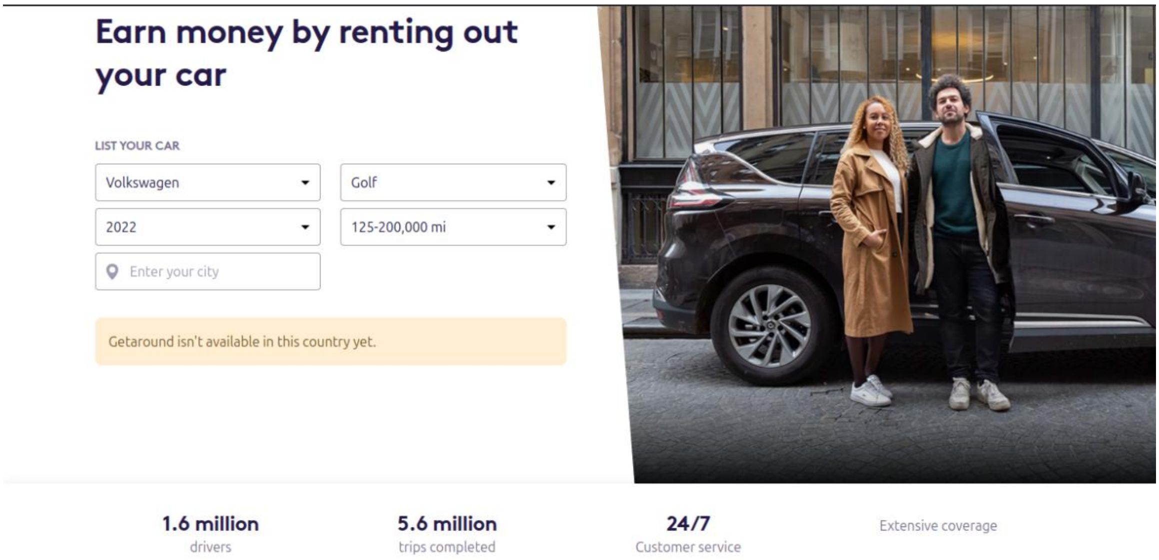


Figure 2.3 Turo

iv. Getaround

Getaround is an international platform offering instant car rentals, allowing users to rent vehicles by the hour or day through its mobile app. However, Getaround faces challenges with vehicle availability in suburban and rural areas, as the platform relies heavily on the density of cars in specific locations. Additionally, while it is a great option for urban areas, it lacks a strong presence in regions with fewer vehicles available for rent, making it less viable in certain parts of the world [9].



Getaround isn't available in this country yet.

Figure 2.4 Getaround

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1. System Analysis

The Multi-Vendor Vehicle Rental System will be developed with a strong focus on user needs and seamless functionality. The platform will be built using modern technologies such as HTML, CSS, JavaScript, React, Node.js, and MySQL. The system will be designed to offer a user-friendly interface with key features including vehicle listing management, vendor and customer profiles, real-time vehicle availability updates, secure payment gateway integration, and an intuitive search and filter system. MVVRS will allow users to rent vehicles on both hourly and daily bases, offering flexibility for a variety of transportation needs while ensuring security and efficiency.

3.1.1. Requirement Analysis

For any system, there are functional and non-functional requirements to be considered. While determining the requirements of the system. Functional requirements are product features or functions those developers must implement to enable users to accomplish their tasks. And Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability.

i. Functional Requirements

Only authenticated users, including admins, Vehicle Owners and Regular Users should have access to their dashboard but anybody can view the listed cars and their details.

Admin:

Admin must be authenticated before accessing system. They'll be able to access admin dashboard.

- Admins can approve, update, or deactivate vendor and user accounts.
- Admins need to update the listed vehicles before they appear in search results or list pages.
- Admins track bookings, payments, and see performance reports for system insights.
- A admin can also create login credentials for the other admins and can also change their passwords.

Vehicle Owner:

Vehicle owners will be able to add their vehicles and also upload the documents of that vehicle so that the admin can verify the vehicle and allow it to be listed in the platform.

- Vehicle owners can add, update, or remove vehicles available for rent.
- Owners can set rental rates and update vehicle availability in real time.
- Owners can view booking requests with driver requests and track ongoing rentals.

User:

Users will be able to search/view the listed vehicles and book the vehicles for more than 3 hours time from the available time slots of the vehicle set by the respective owner.

- Users can search for vehicles using filters (e.g. name, type, price) and make reservations.
- Users can securely pay for bookings and access invoices for completed rentals.
- Users can rate vehicles and leave feedback about their rental experience.

ii. Non-Functional Requirements

- **Performance:** The system should respond quickly to user requests, ensuring minimal latency during vehicle search, booking, and payment processes**.**
- **Scalability:** The platform must handle an increasing number of users, vendors, and transactions without performance degradation.
- **Security:** Ensure secure user authentication, data encryption, and safe payment transactions to protect sensitive information.
- **Availability:** The system should have a 99.9% uptime, ensuring access at all times with minimal disruptions.
- **Usability:** The user interface should be intuitive and easy to navigate for both vehicle owners and renters.

3.1.2. Feasibility Analysis

We conducted a detailed analysis by considering different aspects of a proposed project in order to determine the likelihood of it succeeding. It helped us in determining the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable.

Four major analysis is done to predict the system to be success and they are as follows:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Schedule Feasibility

i. Technical Feasibility:

This study is carried out to check the technical requirements of the system. We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for development and maintaining the system as well. All the resources needed for the development of the software are easily collected.

ii. Operational Feasibility:

The system is easy to operate with the basic knowledge of computation and internet. Well trained manpower is not necessary. Users can easily access the system as it is user friendly in many aspects with good UI.

iii. Economic Feasibility:

Development of this system is highly economically feasible. The organization need not must spend too much money for the development of the system.

iv. Schedule Feasibility:

The schedule feasibility shows the estimated time to complete the project. This includes the schedules of each process. This system we developed is schedule feasible as it doesn't require more time for the development phase. A Gantt chart is used for planning projects of all sizes, is a useful way of showing that work is scheduled to be done on a specific day.

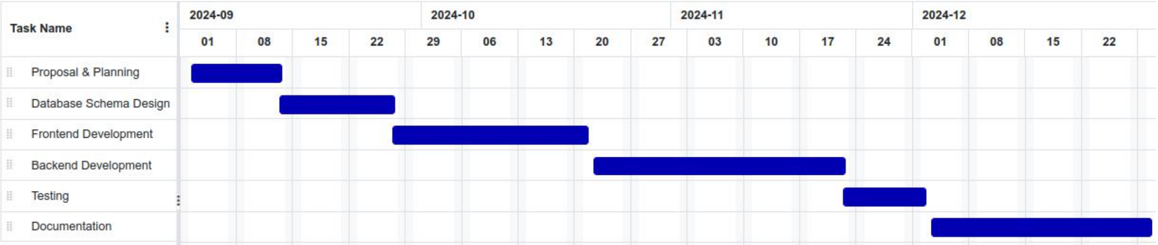


Figure 3.2 Gantt Chart

3.1.3. Conceptual ER Diagram

A conceptual ER diagram (Entity Relationship Diagram) is a high-level representation of a system's data, focusing on the key business entities and their relationships without going into detailed attribute specifications, providing a broad overview of the data structure from a business perspective, often used in the early stages of database design to capture the overall concept of the system before moving into more technical implementations.

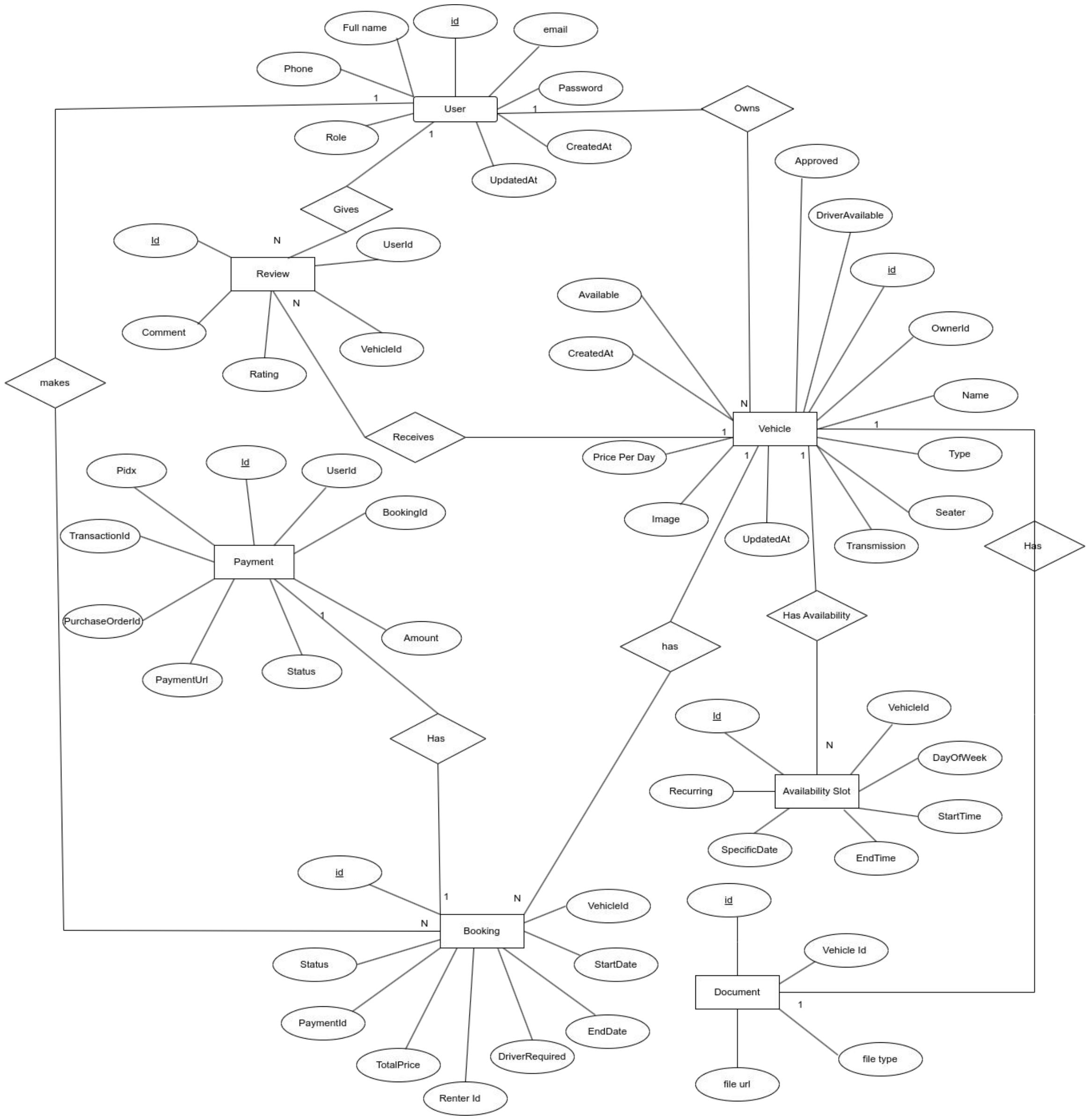
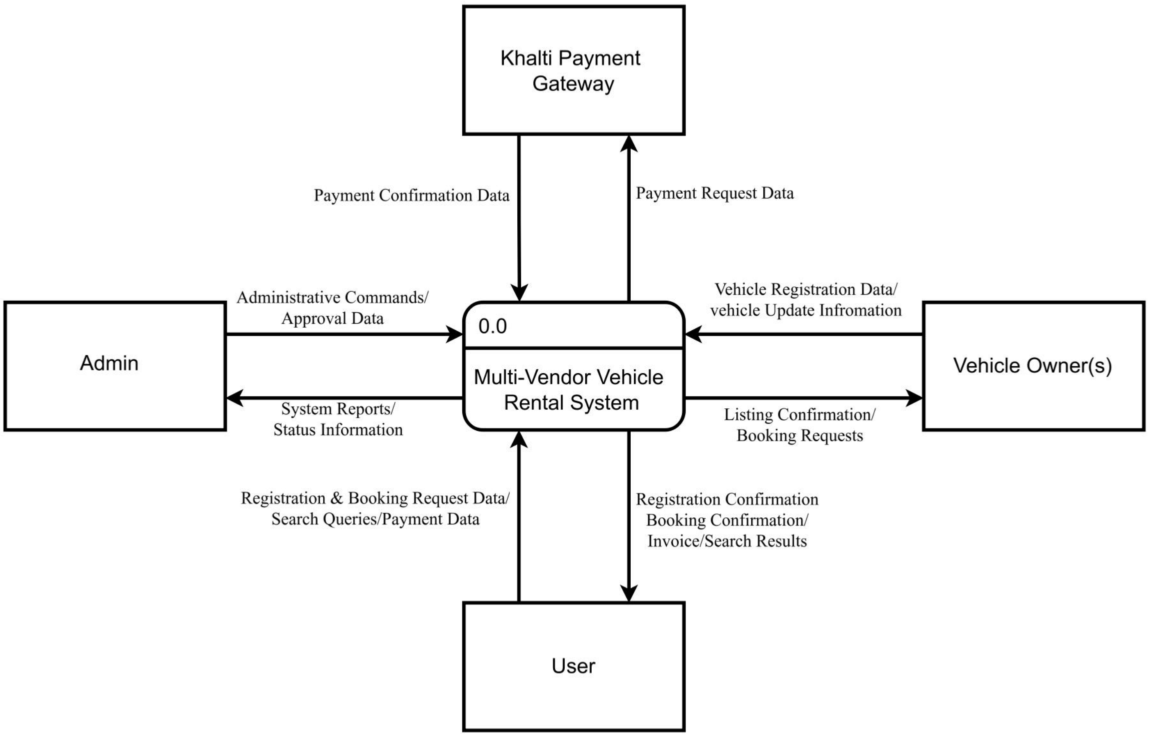


Figure 3.3 Conceptual ER diagram of MVVRS

3.1.4. Process Modelling (DFD)

Process Modelling involves graphically representing the processes, or actions, that capture, manipulate, store, and distribute data between a system and its environment and among components within a system. A common form of a process model is a data flow diagram (DFD). A data flow diagram is a graphic that illustrates the movement of a data between external entities and the processes and data stores within a system.

Context Diagram



Khalti Payment Gateway
Payment Confirmation Data
Payment Request Data
Administrative Commands/ Approval Data
Vehicle Registration Data/ vehicle Update Infromation
Admin
0.0
Multi-Vendor Vehicle Rental System
Vehicle Owner(s)
System Reports/ Status Information
Listing Confirmation/ Booking Requests
Registration & Booking Request Data/ Search Queries/Payment Data
Registration Confirmation Booking Confirmation/ Invoice/Search Results
User

Figure 3.4 Context Diagram of MVVRS

This diagram provides a high-level overview of the Multi-Vendor Vehicle Rental System by representing it as a single process. It shows how external entities—namely, the Admin, Owner, Customer, and Khalti Payment Gateway—interact with the system through major data flows (e.g., registration data, vehicle details, booking requests, and payment confirmations). This high-level view establishes the system boundaries and external interactions without showing internal process details.

Level-0 Diagram

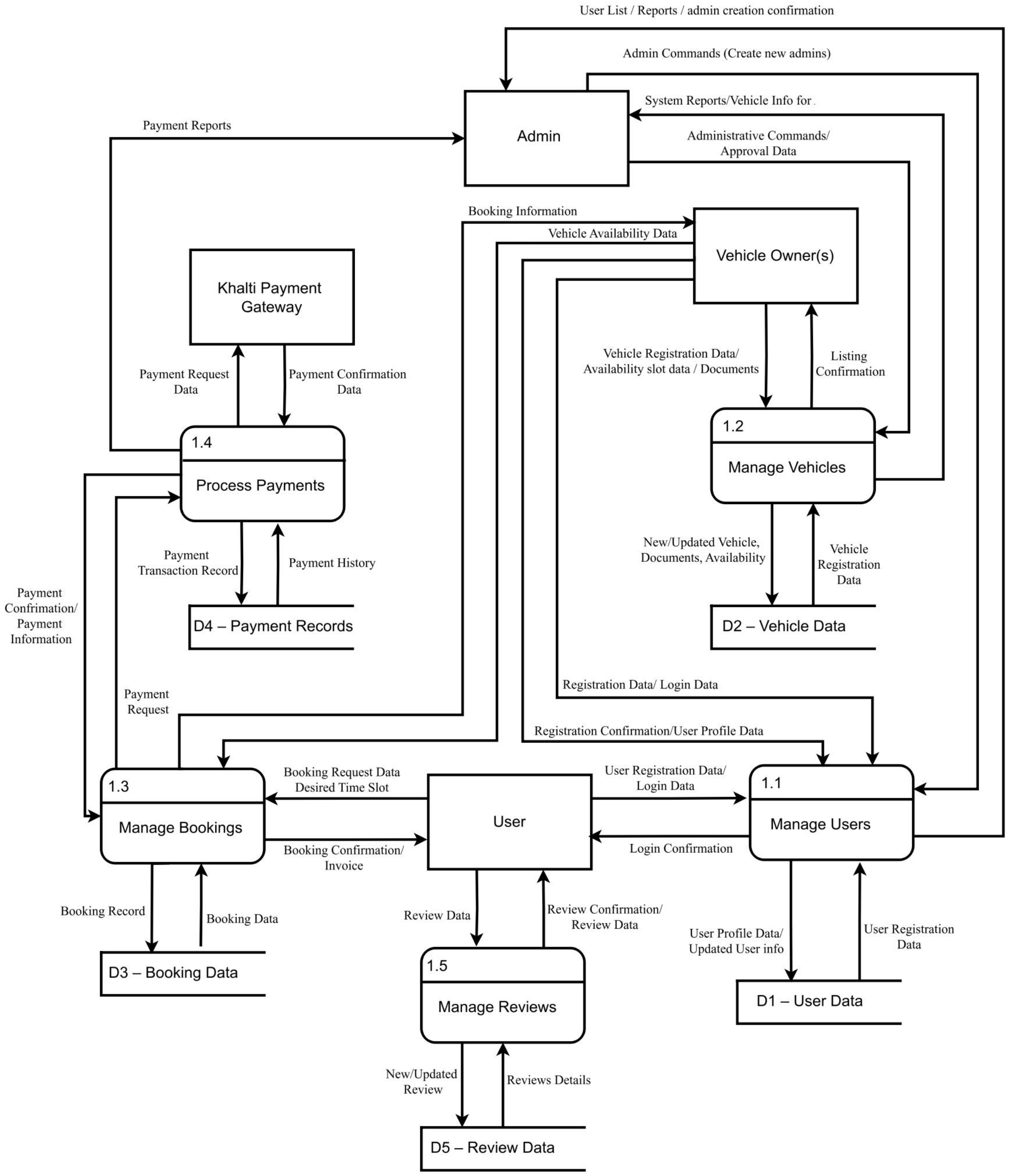

User List / Reports / admin creation confirmation
Admin Commands (Create new admins)
System Reports/Vehicle Info for
Administrative Commands/ Approval Data
Payment Reports
Admin
Booking Information
Vehicle Availability Data
Vehicle Owner(s)
Khalti Payment Gateway
Vehicle Registration Data/ Availability slot data / Documents
Listing Confirmation
Payment Request Data
Payment Confirmation Data
1.4
Process Payments
1.2
Manage Vehicles
New/Updated Vehicle, Documents, Availability
Vehicle Registration Data
Payment Transaction Record
Payment History
Payment Confrimation/ Payment Information
D4 – Payment Records
D2 – Vehicle Data
Payment Request
Registration Data/ Login Data
Registration Confirmation/User Profile Data
1.3
Manage Bookings
Booking Request Data Desired Time Slot
Booking Confirmation/ Invoice
User
User Registration Data/ Login Data
Login Confirmation
1.1
Manage Users
Booking Record
Booking Data
Review Data
Review Confirmation/ Review Data
User Profile Data/ Updated User info
User Registration Data
D3 – Booking Data
1.5
Manage Reviews
D1 – User Data
New/Updated Review
Reviews Details
D5 – Review Data

Figure 3.5 Level 1 DFD of MVVRS

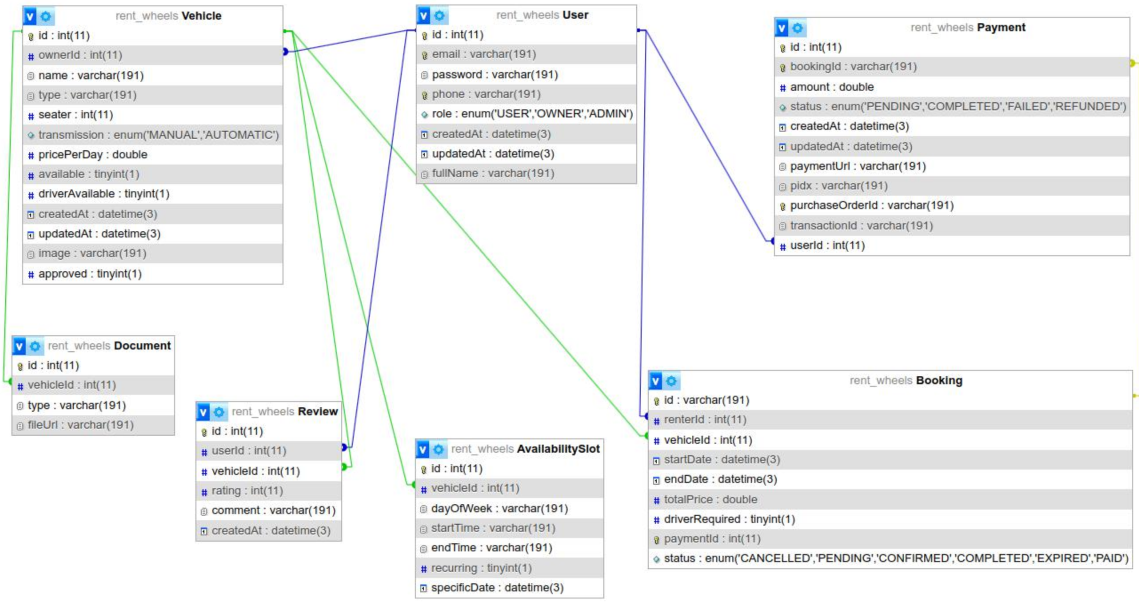
The Level-0 Diagram decomposes the RentWheels system into its major internal processes, such as managing users, vehicles, bookings, reviews, and payments. It also depicts the primary data stores (User Data, Vehicle Data, Booking Data, Payment Data, and Review Data) that support these processes. This diagram offers a more detailed view of the system's internal workings while maintaining clear boundaries with external entities.

3.2. System Design

System Design is the process of defining the elements of a system such as architectures, Modules and components, the different interfaces of the component and the data that goes through the system.

3.2.1. Database Schema Diagram

The design of the database is called schema. This tells us about the structural view of the database. A database schema represents the logical configuration of all or part of a relation database.



rent_wheels **Vehicle**
id : int(11)
ownerId : int(11)
name : varchar(191)
type : varchar(191)
seater : int(11)
transmission : enum('MANUAL','AUTOMATIC')
pricePerDay : double
available : tinyint(1)
driverAvailable : tinyint(1)
createdAt : datetime(3)
updatedAt : datetime(3)
image : varchar(191)
approved : tinyint(1)

rent_wheels **User**
id : int(11)
email : varchar(191)
password : varchar(191)
phone : varchar(191)
role : enum('USER','OWNER','ADMIN')
createdAt : datetime(3)
updatedAt : datetime(3)
fullName : varchar(191)

rent_wheels **Payment**
id : int(11)
bookingId : varchar(191)
amount : double
status : enum('PENDING','COMPLETED','FAILED','REFUNDED')
createdAt : datetime(3)
updatedAt : datetime(3)
paymentUrl : varchar(191)
pidx : varchar(191)
purchaseOrderId : varchar(191)
transactionId : varchar(191)
userId : int(11)

rent_wheels **Document**
id : int(11)
vehicleId : int(11)
type : varchar(191)
fileUrl : varchar(191)

rent_wheels **Review**
id : int(11)
userId : int(11)
vehicleId : int(11)
rating : int(11)
comment : varchar(191)
createdAt : datetime(3)

rent_wheels **AvailabilitySlot**
id : int(11)
vehicleId : int(11)
dayOfWeek : varchar(191)
startTime : varchar(191)
endTime : varchar(191)
recurring : tinyint(1)
specificDate : datetime(3)

rent_wheels **Booking**
id : varchar(191)
renterId : int(11)
vehicleId : int(11)
startDate : datetime(3)
endDate : datetime(3)
totalPrice : double
driverRequired : tinyint(1)
paymentId : int(11)
status : enum('CANCELLED','PENDING','CONFIRMED','COMPLETED','EXPIRED','PAID')

Figure 3.6 Database Schema Diagram of MVVRS

3.2.2. Architectural Design

An architectural diagram is a visual representation that maps out the physical Implementation for components of a software system. The architectural design of the multi-vendor vehicle rental system follows a three-tier architecture, ensuring scalability, maintainability, and efficiency. The system consists of the frontend (presentation layer), backend (application layer), and database (data layer).

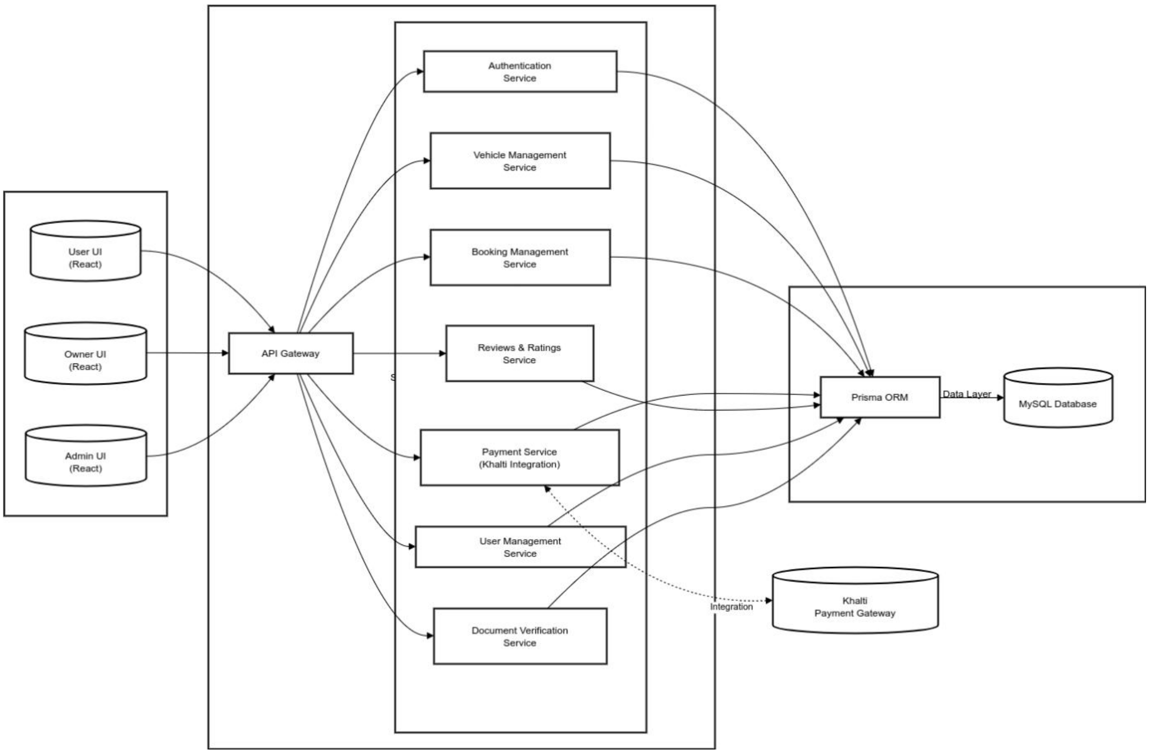


Figure 3.7 Architectural Design of MVVRS

3.2.3. Interface Design

Note: Screenshots are available at Appendix 1.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1. Implementation

4.1.1. Tools used

In the development of the Multivendor Vehicle Rental System, a combination of frontend and backend technologies has been utilized to ensure a robust and efficient application.

i. Frontend

The frontend of the system is implemented using React, Tailwind CSS, and JavaScript to ensure a seamless and interactive user experience.

React: React [10] is a JavaScript library for building user interfaces, primarily used for developing single-page applications. It allows developers to create reusable UI components, ensuring fast rendering and an efficient development process.

Tailwind CSS: Tailwind CSS [11] is a utility-first CSS framework that provides pre-defined classes to design responsive and modern interfaces. It helps developers build custom designs quickly without writing extensive custom CSS.

JavaScript: JavaScript (JS) [12] is a versatile, lightweight, and object-oriented programming language used to add interactivity to web pages. It is widely adopted in both frontend and backend development and supports frameworks like React for building dynamic web applications.

ii. Backend and Database

The backend is developed using Express.js, and the database is managed using MySQL with Prisma ORM.

Express.js: Express.js [13] is a minimal and flexible Node.js web application framework that simplifies backend development. It provides features for handling routes, middleware, and API requests, making it ideal for building scalable web applications.

Prisma: Prisma [14] is a modern database toolkit that acts as an ORM (Object-Relational Mapper) for MySQL and other databases. It simplifies database interactions by providing type-safe queries and an intuitive schema-based approach to managing data.

MySQL: MySQL [15] is an open-source relational database management system (RDBMS) widely used for handling structured data. It ensures data integrity, scalability, and efficiency in managing the records of the vehicle rental system.

Various software tools have been utilized throughout development process, including:

- **Microsoft Word:** For documentation purposes.
- **Visual Studio Code:** As an integrated development environment (IDE) for coding.
- **Google Meet:** For communication and collaboration among team members.
- **App Diagram:** For creating diagrams and visual representations of system architecture and workflow.

4.1.2. Implementation Details of Modules

The implementation involved the creation of specific modules to achieve different aspects of Multi Vendor Vehicle Rental System:

- **User Registration and Login:** This module ensures a seamless and secure registration and login process for users. React is used to build an intuitive UI, while JavaScript and Tailwind CSS handle client-side validation. Authentication and authorization are managed using Express.js and Prisma, ensuring secure user access.
- **Vehicle Management:** In this module, the vehicle owners can manage vehicle listings, including adding new vehicles, updating details such as availability, pricing, descriptions, and categories. This ensures that customers can browse and rent vehicles conveniently
- **Vehicle Booking:** Customers can browse available vehicles, select their preferred option, and book it online. They can also request for a driver by selecting the option.
- **Ratings and Reviews:** Regular users/customers can provide feedback, rate their rental experience, and leave reviews for vehicles they have rented. This helps maintain service quality and allows future customers to make informed decisions.

4.2. Testing

Software testing is an essential aspect of ensuring the functionality, correctness, and quality of the MVVRS. It involves a systematic technical investigation aimed at identifying and rectifying errors and anomalies within the software. Testing encompasses various methodologies and techniques to validate the system's behavior and performance within its operational context, ultimately ensuring a seamless and robust user experience.

4.2.1. Test Cases for Unit Testing

We tested the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Table 4.1 Test Cases For Unit Testing

TC ID	Test Case	Test Data	Description	Expected Result	Actual Result	Remarks
1	Testing Admin Login	Email: admin@rentwheels.com Password: AdminPass3#	While entering correct credentials	Successfully logged in and redirect to admin home page.	As expected result	pass
2	Testing Admin Login	Email: admin08@rentwheels.com Password: Admin12345#	While entering wrong credentials	Show error and please try again.	As expected result	Pass
3	Click logout button	Click Logout button	Clicking logout button	Redirect to login page	As expected result	Pass
4	Testing Owner Login	Email: owner@rentwheels.com Password: OwnerPass1!	While entering correct credentials	Successfully logged in and redirect to owner home page.	As expected result	Pass

TC ID	Test Case	Test Data	Description	Expected Result	Actual Result	Remarks
5	Testing Owner Login	Email: Utsav77@rentwheels.com Password: Hello12345#	While entering wrong credentials	Show error and please try again.	As expected result	Pass
6	Click logout button	Click Logout button	Clicking logout button	Redirect to login page	As expected result	Pass
7	Testing User Login	Email: user@rentwheels.com Password: Userpass2@	While entering correct credentials	Successfully logged in and redirect to user home page.	As expected result	pass
8	Testing User Login	Email: Nischal44@rentwheels.com Password: React4567	While entering wrong credentials	Show error and please try again.	As expected result	Pass
9	Click logout button	Click Logout button	Clicking logout button	Redirect to login page	As expected result	Pass

4.2.2. Test Cases for System Testing

System testing evaluates the complete and integrated MVVRS platform to ensure all modules function correctly together. It involves testing the overall workflow, including authentication, vehicle listing, booking management, rental agreements, user interactions, and payment processing. This testing phase ensures that the system behaves as expected under real-world conditions, including different user roles, load handling, and security aspects.

Table 4.2 Test Cases for System Testing

Test Scenario	Test Steps	Expected Result	Actual Result	Status
User Registration & Login	User registers with valid details and logs in.	User is successfully registered and logged in.	User is able to register and log in successfully.	Pass
Vehicle Listing by Vendor	Vendor adds a new vehicle to the system.	Vehicle is successfully listed and visible to users.	Vehicle appears in the listing.	Pass
Vehicle Search & Filtering	User searches for a vehicle using filters like type, and name.	Search results display relevant vehicle listings.	Search results display correctly.	Pass
Booking Process (Customer)	User selects a vehicle and proceeds to book.	Booking is confirmed, and order appears in dashboard.	Booking appears in user dashboard.	Pass

Test Scenario	Test Steps	Expected Result	Actual Result	Status
Payment Processing	User makes payment via Khalti	Payment is processed, and invoice is generated.	Payment is processed, and invoice received.	Pass
Vehicle Availability Update	Vendor updates vehicle availability status.	Vehicle status reflects as "Available" or "Booked".	Status updates correctly.	Pass
Review & Rating System	User submits a review and rating after vehicle use.	Review is displayed on the vehicle's page.	Review appears correctly.	Pass
Admin Dashboard Analytics	Admin checks dashboard for reports and statistics.	Reports display real-time data on revenue & bookings.	Data is displayed accurately.	Pass
Security - Unauthorized Access	A non-logged-in user tries to access vendor or admin panel.	System restricts access and redirects to login page.	Access is denied, and user redirected.	Pass

CHAPTER 5: CONCLUSION AND FURTHER WORK

5.1. Conclusion

With the increasing demand for convenient and efficient transportation solutions, online vehicle rental services have gained significant popularity. This Vehicle Rental System provides customers with an easy-to-use platform to browse, book, and manage vehicle rentals from the comfort of their homes. It ensures a seamless and hassle-free booking process while allowing vehicle owners to manage their fleet efficiently.

In addition to enhancing customer convenience, this system streamlines record-keeping, improves vehicle management, and optimizes rental operations for business owners. It enables vehicle rental businesses to reach a wider audience, maximizing their profitability and service efficiency. For customers, it means access to a diverse range of vehicles anytime, anywhere, eliminating the need for traditional rental hassles. With smart features like booking tracking and reviews, the system enhances user experience while ensuring transparency and reliability in vehicle rentals.

5.2. Lesson learnt/Outcome

As a team involved in this project, we have gained valuable insights and hands-on experience throughout the development process. We realized the importance of proper planning, teamwork, and effective communication in building a functional and user-friendly system. This project enhanced our ability to understand client requirements, translate them into technical solutions, and implement them efficiently.

This Vehicle Rental System focuses on modernizing traditional vehicle rental services by bringing them to an online platform. It provides convenience to customers by allowing easy vehicle booking while streamlining operations for rental businesses. By integrating secure payment gateways, real-time availability tracking, and a user-friendly interface, the system enhances the overall rental experience. Throughout this project, we learned the significance of full-stack development, database management, and system design while reinforcing the need for user-centric solutions. Additionally, we gained hands-on experience with backend optimization, API integration, and cloud deployment, which are essential for building scalable applications. This experience has not only strengthened our technical skills but also improved our problem-solving and project management abilities, preparing us for real-world software development challenges.

5.3. Future Plans

We intend to develop this web app further and enhance it to make it more productive and useful. Possible future enhancement of this MVVRS would be:

- Integrate more digital wallets and payment options for seamless transactions.
- Develop a dedicated mobile app for better accessibility and convenience.
- Introduce advanced analytics for vendors to monitor trends and performance.
- Enhance system security with multi-factor authentication for data protection.
- Introduce reward points, discounts, and membership benefits for frequent users.

REFERENCES

[1] A. White and B. Green, “Utility-first design with Tailwind CSS,” *International Journal of Frontend Development*, vol. 9, no. 3, pp. 212-227, 2022.

[2] Getaround, “Getaround: Instant Car Rentals Near You,” *Getaround Website*, [Online]. Available: https://www.getaround.com.

[3] J. Clark, “React: A JavaScript library for building user interfaces,” *Journal of Web Development*, vol. 14, no. 2, pp. 101-118, 2023.

[4] J. Smith, “E-commerce and its impact on the vehicle rental industry,” *Journal of Business Innovation*, vol. 12, no. 3, pp. 45-56, 2023.

[5] K. Patel, “Building scalable web applications with Express.js,” *Journal of Backend Technologies*, vol. 8, no. 4, pp. 145-163, 2023.

[6] L. Harris, “Data privacy in the sharing economy: A case study of vehicle rental platforms,” *Cybersecurity and Privacy*, vol. 6, no. 2, pp. 14-28, 2020.

[7] M. Johnson and A. Lee, “The role of mobile commerce in the vehicle rental market,” *Transportation Studies*, vol. 8, no. 1, pp. 78-92, 2022.

[8] R. Adams and M. Zhao, “The evolution of Object-Relational Mapping: A study on Prisma,” *International Journal of Database Management*, vol. 12, no. 1, pp. 98-112, 2022.

[9] R. Kumar and P. Singh, “Flexibility in rental services: Consumer preference for hourly vehicle rentals,” *International Journal of Business & Management*, vol. 17, no. 4, pp. 23-38, 2021.

[10] S. Thompson, “Relational database systems: MySQL in modern applications,” *Computing Advances Journal*, vol. 15, no. 2, pp. 189-204, 2021.

[11] Sajilo Rental, “Nepal’s First Car Rental Marketplace,” *Sajilo Rental Website*, [Online]. Available: https://apppage.net/preview/com.vedaItPvtLtd.sajiloRental.

[12] Sasto Rental, “Best Vehicle Rental Services Provider in Nepal,” *Sasto Rental Website*, [Online]. Available: https://www.sastorental.com/.

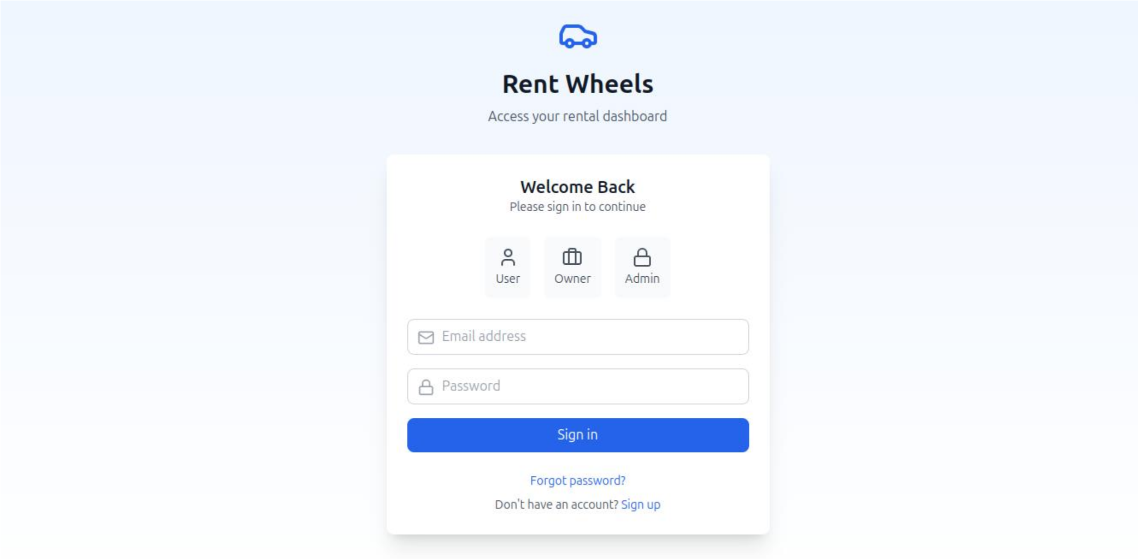
[13] T. Brown, “Modern JavaScript: Enhancing web interactivity,” *Software Engineering Review*, vol. 11, no. 1, pp. 67-81, 2021.

[14] T. Zhang and J. Wang, “User experience in online vehicle rental platforms,” *Journal of User-Centered Design*, vol. 7, no. 3, pp. 121-135, 2019.

[15] Turo, “Turo: The World’s Largest Car Sharing Marketplace,” *Turo Website*, [Online]. Available: https://www.turo.com

APPENDICES

Appendix 1: Screenshots

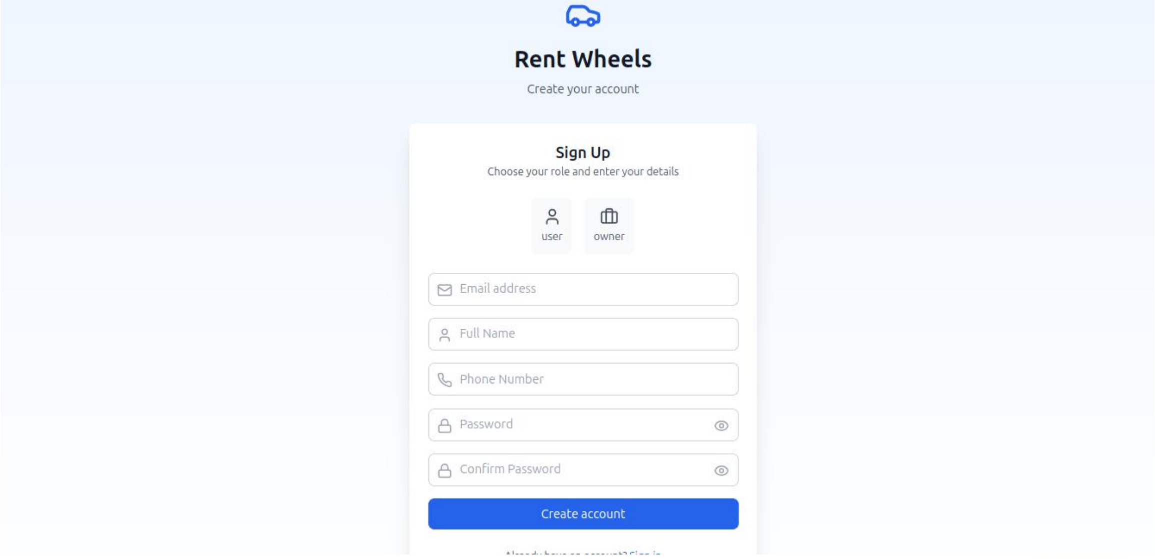


Rent Wheels

Access your rental dashboard

Welcome Back

Please sign in to continue

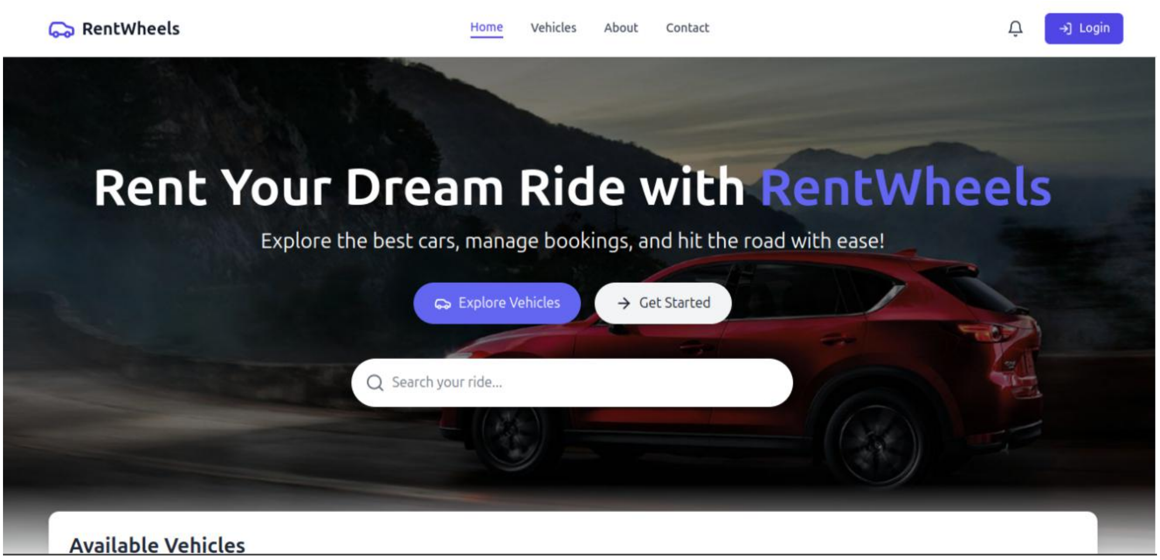


Rent Wheels

Create your account

Sign Up

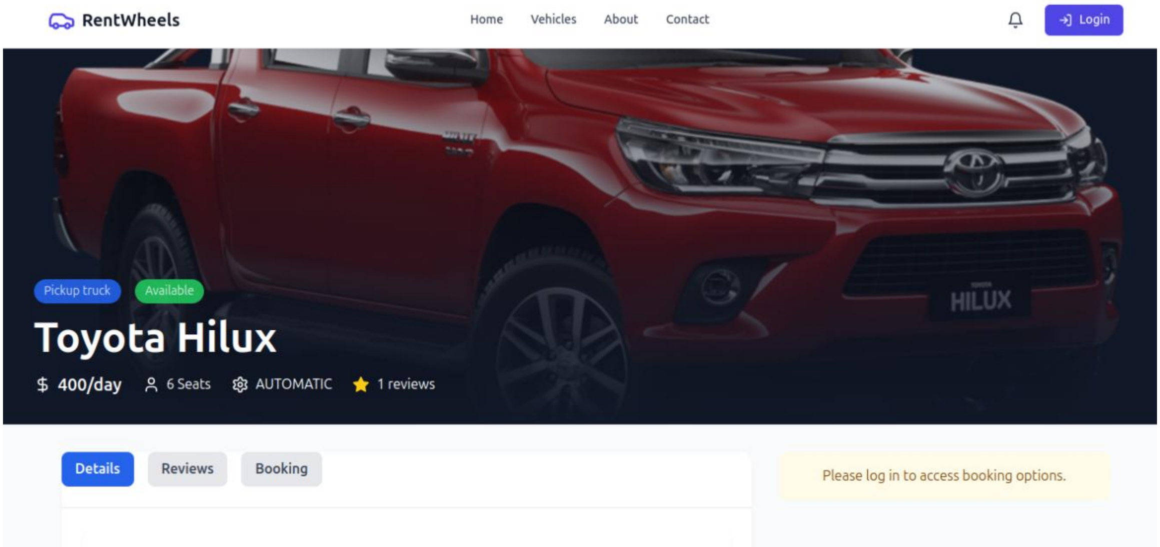
Choose your role and enter your details



Rent Your Dream Ride with RentWheels

Explore the best cars, manage bookings, and hit the road with ease!

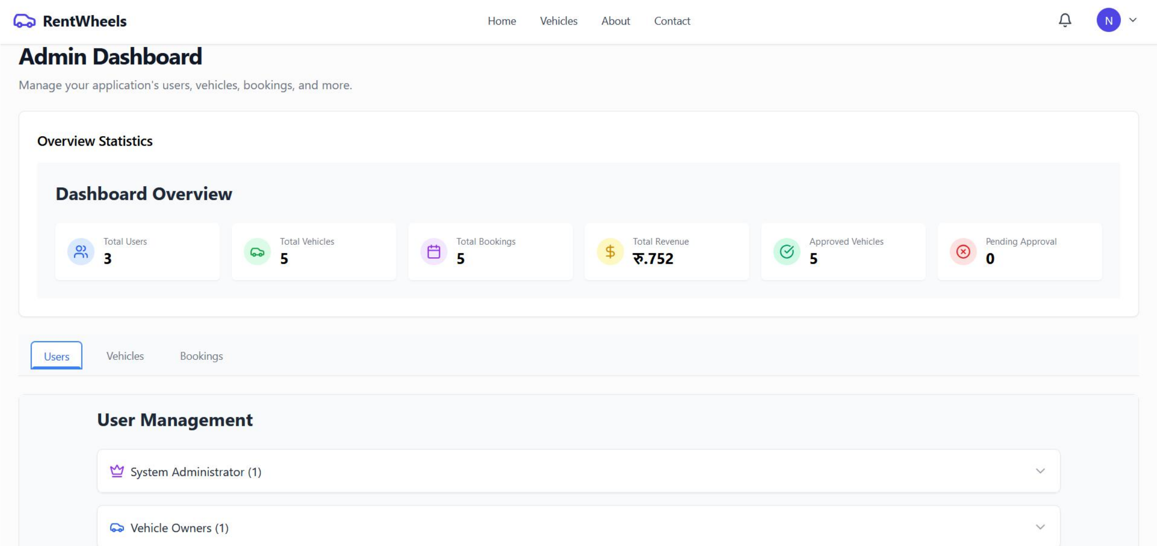
Available Vehicles



Toyota Hilux

$ 400/day · 6 Seats · AUTOMATIC · ★ 1 reviews

Please log in to access booking options.



Admin Dashboard

Manage your application's users, vehicles, bookings, and more.

Overview Statistics

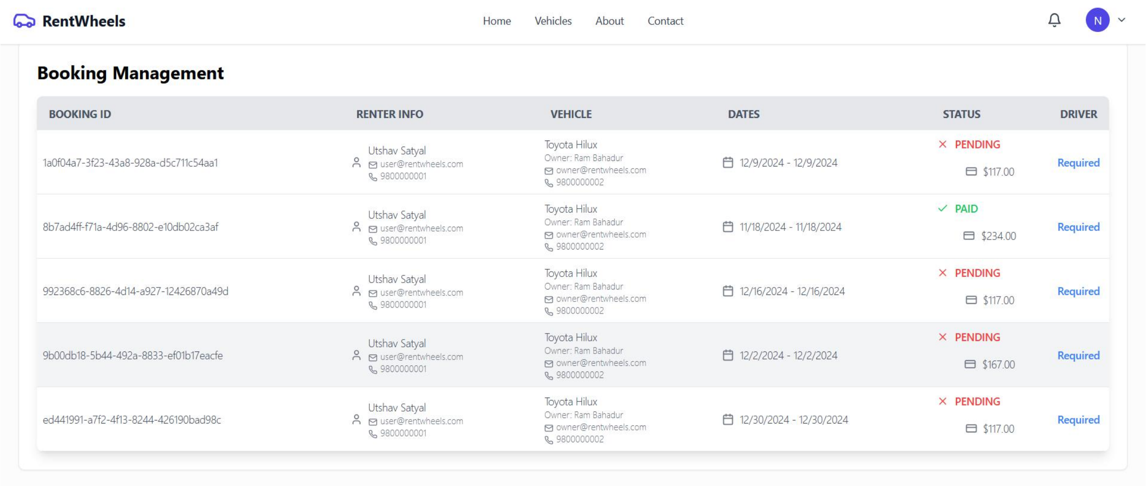
Dashboard Overview

Total Users	Total Vehicles	Total Bookings	Total Revenue	Approved Vehicles	Pending Approval
3	5	5	रु.752	5	0

User Management

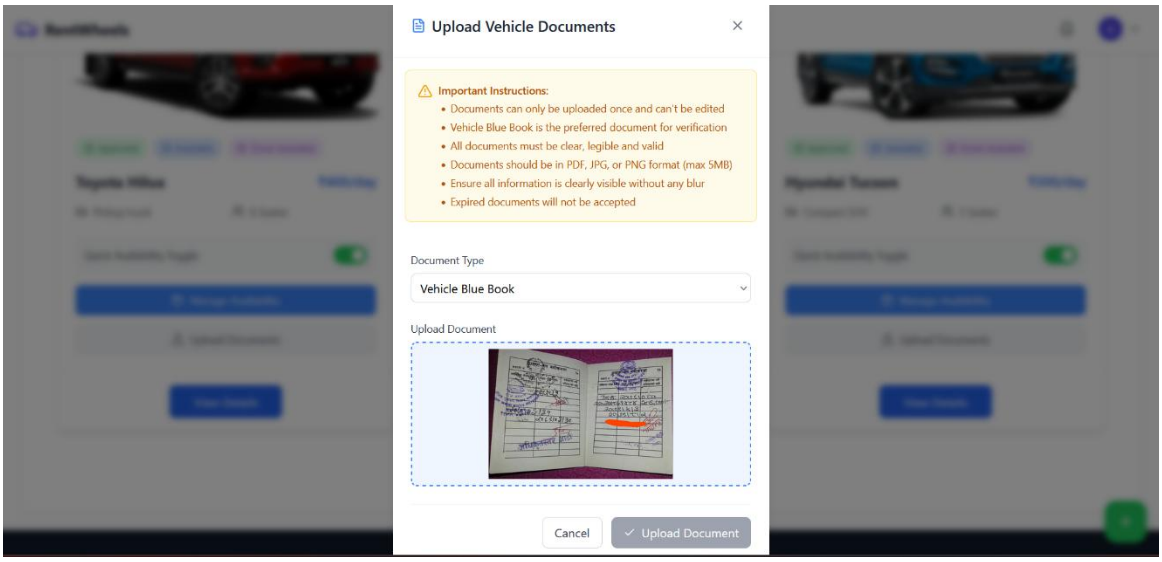
System Administrator (1)

Vehicle Owners (1)



Booking Management

BOOKING ID	RENTER INFO	VEHICLE	DATES	STATUS	DRIVER
1a0f04a7-3f23-43a8-928a-d5c711c54aa1	Utshav Satyal user@rentwheels.com 9800000001	Toyota Hilux Owner: Ram Bahadur owner@rentwheels.com 9800000002	12/9/2024 - 12/9/2024	✕ PENDING $117.00	Required
8b7ad4ff-f71a-4d96-8802-e10db02ca3af	Utshav Satyal user@rentwheels.com 9800000001	Toyota Hilux Owner: Ram Bahadur owner@rentwheels.com 9800000002	11/18/2024 - 11/18/2024	✓ PAID $234.00	Required
992368c6-8826-4d14-a927-12426870a49d	Utshav Satyal user@rentwheels.com 9800000001	Toyota Hilux Owner: Ram Bahadur owner@rentwheels.com 9800000002	12/16/2024 - 12/16/2024	✕ PENDING $117.00	Required
9b00db18-5b44-492a-8833-ef01b17eacfe	Utshav Satyal user@rentwheels.com 9800000001	Toyota Hilux Owner: Ram Bahadur owner@rentwheels.com 9800000002	12/2/2024 - 12/2/2024	✕ PENDING $167.00	Required
ed441991-a7f2-4f13-8244-426190bad98c	Utshav Satyal user@rentwheels.com 9800000001	Toyota Hilux Owner: Ram Bahadur owner@rentwheels.com 9800000002	12/30/2024 - 12/30/2024	✕ PENDING $117.00	Required

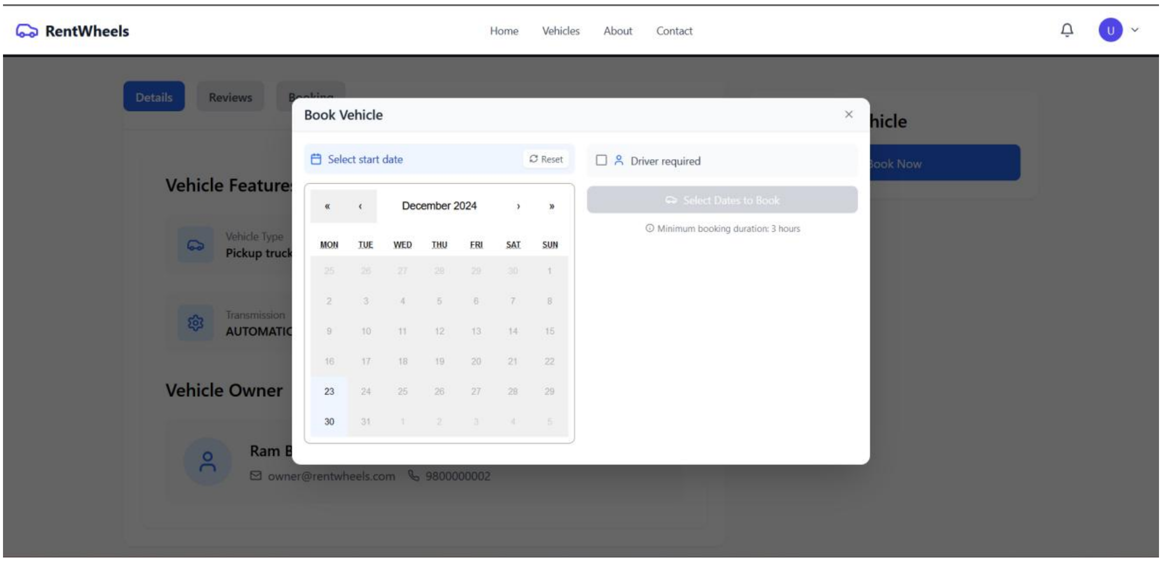


Upload Vehicle Documents

Important Instructions:
- Documents can only be uploaded once and can't be edited
- Vehicle Blue Book is the preferred document for verification
- All documents must be clear, legible and valid
- Documents should be in PDF, JPG, or PNG format (max 5MB)
- Ensure all information is clearly visible without any blur
- Expired documents will not be accepted

Document Type: Vehicle Blue Book

Upload Document



Vehicle Features

Vehicle Type: Pickup truck

Transmission: AUTOMATIC

Vehicle Owner

Ram B

owner@rentwheels.com 9800000002

Book Vehicle

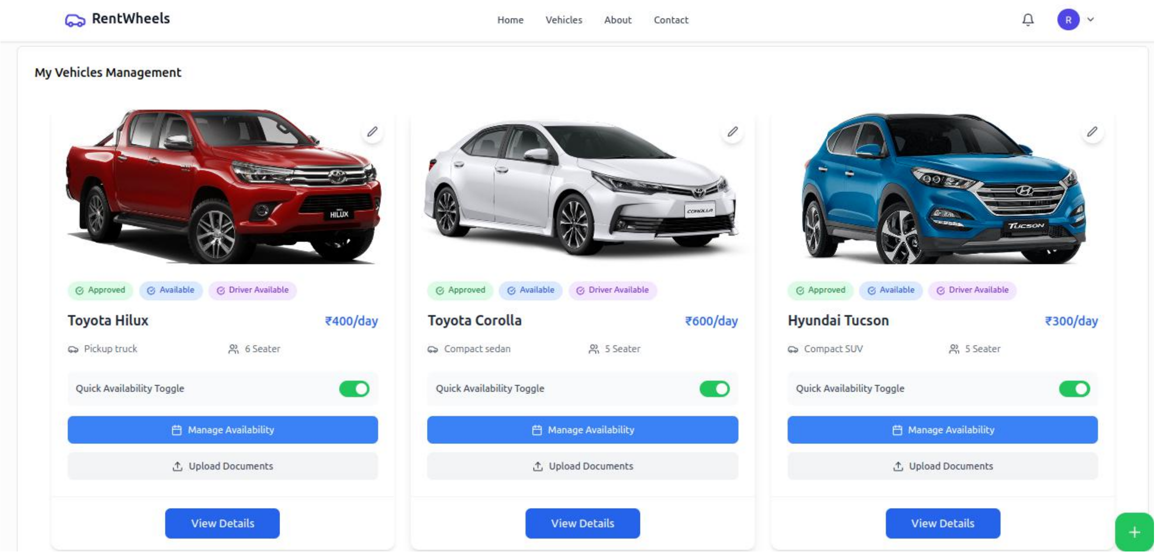
Select start date

□ Driver required

Minimum booking duration: 3 hours

« ‹ December 2024 › »

MON	TUE	WED	THU	FRI	SAT	SUN
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5



My Vehicles Management

Approved · Available · Driver Available

Toyota Hilux ₹400/day

Pickup truck · 6 Seater

Quick Availability Toggle

Approved · Available · Driver Available

Toyota Corolla ₹600/day

Compact sedan · 5 Seater

Quick Availability Toggle

Approved · Available · Driver Available

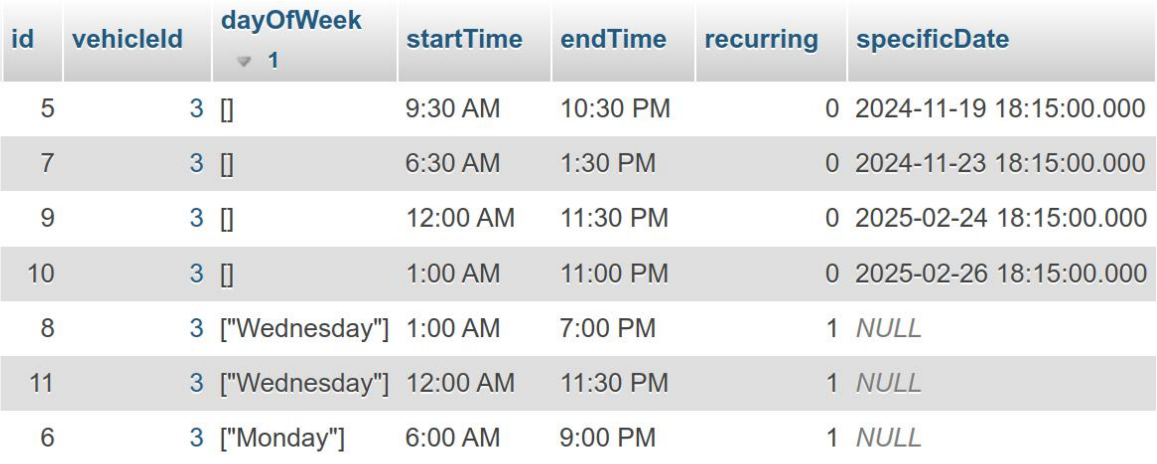
Hyundai Tucson ₹300/day

Compact SUV · 5 Seater

Quick Availability Toggle

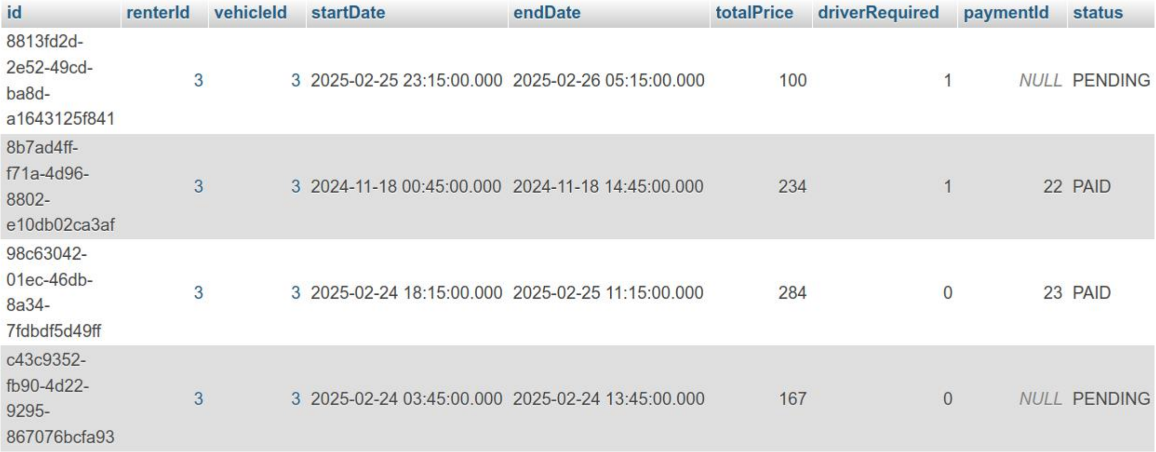
Appendix 2: Database Table

Availability slot Table



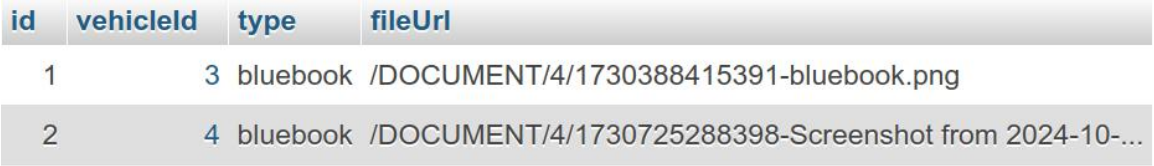
id	vehicleId	dayOfWeek ▼ 1	startTime	endTime	recurring	specificDate
5	3	[]	9:30 AM	10:30 PM	0	2024-11-19 18:15:00.000
7	3	[]	6:30 AM	1:30 PM	0	2024-11-23 18:15:00.000
9	3	[]	12:00 AM	11:30 PM	0	2025-02-24 18:15:00.000
10	3	[]	1:00 AM	11:00 PM	0	2025-02-26 18:15:00.000
8	3	["Wednesday"]	1:00 AM	7:00 PM	1	NULL
11	3	["Wednesday"]	12:00 AM	11:30 PM	1	NULL
6	3	["Monday"]	6:00 AM	9:00 PM	1	NULL

Booking Table



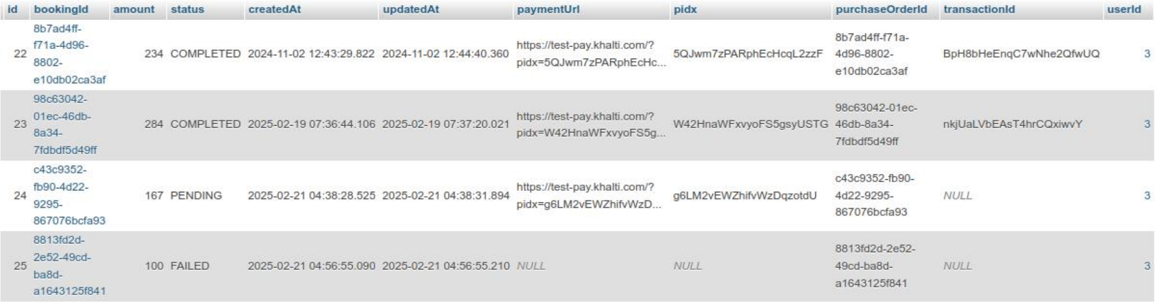
id	renterId	vehicleId	startDate	endDate	totalPrice	driverRequired	paymentId	status
8813fd2d-2e52-49cd-ba8d-a1643125f841	3	3	2025-02-25 23:15:00.000	2025-02-26 05:15:00.000	100	1	NULL	PENDING
8b7ad4ff-f71a-4d96-8802-e10db02ca3af	3	3	2024-11-18 00:45:00.000	2024-11-18 14:45:00.000	234	1	22	PAID
98c63042-01ec-46db-8a34-7fdbdf5d49ff	3	3	2025-02-24 18:15:00.000	2025-02-25 11:15:00.000	284	0	23	PAID
c43c9352-fb90-4d22-9295-867076bcfa93	3	3	2025-02-24 03:45:00.000	2025-02-24 13:45:00.000	167	0	NULL	PENDING

Document Table



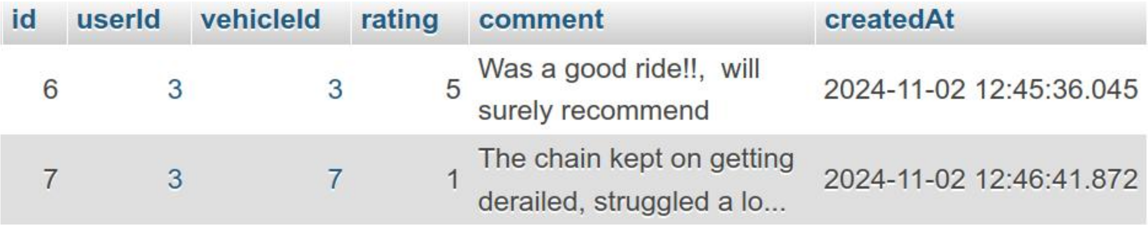
id	vehicleId	type	fileUrl
1	3	bluebook	/DOCUMENT/4/1730388415391-bluebook.png
2	4	bluebook	/DOCUMENT/4/1730725288398-Screenshot from 2024-10-...

Payment Table



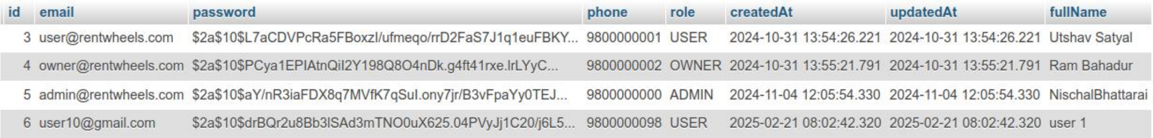
id	bookingId	amount	status	createdAt	updatedAt	paymentUrl	pidx	purchaseOrderId	transactionId	userId
22	8b7ad4ff-f71a-4d96-8802-e10db02ca3af	234	COMPLETED	2024-11-02 12:43:29.822	2024-11-02 12:44:40.360	https://test-pay.khalti.com/?pidx=5QJwm7zPARphEcHc...	5QJwm7zPARphEcHcqL2zzF	8b7ad4ff-f71a-4d96-8802-e10db02ca3af	BpH8bHeEnqC7wNhe2QfwUQ	3
23	98c63042-01ec-46db-8a34-7fdbdf5d49ff	284	COMPLETED	2025-02-19 07:36:44.106	2025-02-19 07:37:20.021	https://test-pay.khalti.com/?pidx=W42HnaWFxvyoFS5g...	W42HnaWFxvyoFS5gsyUSTG	98c63042-01ec-46db-8a34-7fdbdf5d49ff	nkjUaLVbEAsT4hrCQxiwvY	3
24	c43c9352-fb90-4d22-9295-867076bcfa93	167	PENDING	2025-02-21 04:38:28.525	2025-02-21 04:38:31.894	https://test-pay.khalti.com/?pidx=g6LM2vEWZhifvWzD...	g6LM2vEWZhifvWzDqzotdU	c43c9352-fb90-4d22-9295-867076bcfa93	NULL	3
25	8813fd2d-2e52-49cd-ba8d-a1643125f841	100	FAILED	2025-02-21 04:56:55.090	2025-02-21 04:56:55.210	NULL	NULL	8813fd2d-2e52-49cd-ba8d-a1643125f841	NULL	3

Review Table



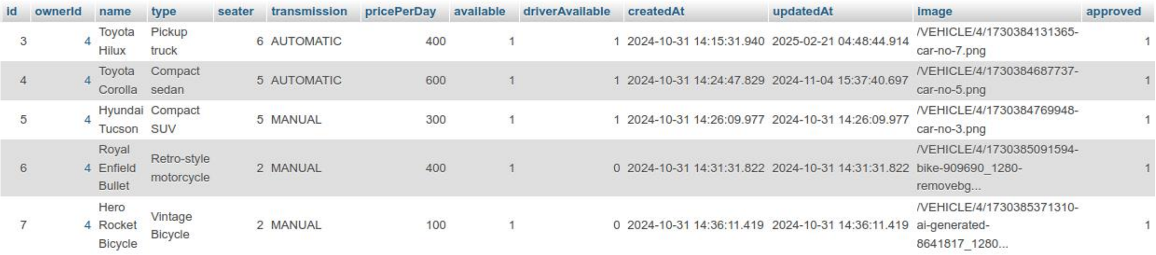
id	userId	vehicleId	rating	comment	createdAt
6	3	3	5	Was a good ride!!, will surely recommend	2024-11-02 12:45:36.045
7	3	7	1	The chain kept on getting derailed, struggled a lo...	2024-11-02 12:46:41.872

User Table



id	email	password	phone	role	createdAt	updatedAt	fullName
3	user@rentwheels.com	$2a$10$L7aCDVPcRa5FBoxzl/ufmeqo/rrD2FaS7J1q1euFBKY...	9800000001	USER	2024-10-31 13:54:26.221	2024-10-31 13:54:26.221	Utshav Satyal
4	owner@rentwheels.com	$2a$10$PCya1EPIAtnQiI2Y198Q8O4nDk.g4ft41rxe.lrLYyC...	9800000002	OWNER	2024-10-31 13:55:21.791	2024-10-31 13:55:21.791	Ram Bahadur
5	admin@rentwheels.com	$2a$10$aY/nR3iaFDX8q7MVfK7qSuI.ony7jr/B3vFpaYy0TEJ...	9800000000	ADMIN	2024-11-04 12:05:54.330	2024-11-04 12:05:54.330	NischalBhattarai
6	user10@gmail.com	$2a$10$drBQr2u8Bb3lSAd3mTNO0uX625.04PVyJj1C20/j6L5...	9800000098	USER	2025-02-21 08:02:42.320	2025-02-21 08:02:42.320	user 1

Vehicle Table



id	ownerId	name	type	seater	transmission	pricePerDay	available	driverAvailable	createdAt	updatedAt	image	approved
3	4	Toyota Hilux	Pickup truck	6	AUTOMATIC	400	1	1	2024-10-31 14:15:31.940	2025-02-21 04:48:44.914	/VEHICLE/4/1730384131365-car-no-7.png	1
4	4	Toyota Corolla	Compact sedan	5	AUTOMATIC	600	1	1	2024-10-31 14:24:47.829	2024-11-04 15:37:40.697	/VEHICLE/4/1730384687737-car-no-5.png	1
5	4	Hyundai Tucson	Compact SUV	5	MANUAL	300	1	1	2024-10-31 14:26:09.977	2024-10-31 14:26:09.977	/VEHICLE/4/1730384769948-car-no-3.png	1
6	4	Royal Enfield Bullet	Retro-style motorcycle	2	MANUAL	400	1	0	2024-10-31 14:31:31.822	2024-10-31 14:31:31.822	/VEHICLE/4/1730385091594-bike-909690_1280-removebg...	1
7	4	Hero Rocket Bicycle	Vintage Bicycle	2	MANUAL	100	1	0	2024-10-31 14:36:11.419	2024-10-31 14:36:11.419	/VEHICLE/4/1730385371310-ai-generated-8641817_1280...	1