CARS RENTAL MANAGEMENT SYSTEM

by

SANOJ RAJ C
(Reg. No: 24800268)
and
LOGANATHAN S
(Reg. No: 24800261)

Project report submitted in partial fulfillment of the

requirements for the degree of

MASTER OF COMPUTER APPLICATIONS

of

Pondicherry University

DEPARTMENT OF COMPUTER APPLICATIONS

POPE JOHN PAUL II COLLEGE OF EDUCATION

REDDIARPALAYAM, PUDUCHERRY-10

DECEMBER-2025

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"CARS RENTAL MANAGEMENT SYSTEM"** is a bonafide record of work done by **SANOJ RAJ C** and **LOGANATHAN S** in partial fulfilment for the degree of Master of Computer Applications of Pondicherry University.

This work has not been submitted elsewhere for the award of similar or any other degree to the best our knowledge.

INTERNAL GUIDE

Mrs. Yazhini. S,
Assistant Professor,
Department of Computer Applications,
Pope John Paul II College of Education,
Puducherry.

HEAD OF THE DEPARTMENT

Mr. Maury Marie Emmanuel,
Head of Department,
Department of Computer Applications,
Pope John Paul II College of Education,
Puducherry.

Submitted for the University Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The successful completion of any work would be incomplete without mentioning the people who made it possible. I take privilege to express a few words of gratitude and respect to all those who helped us in completion of the project. First, I thank the Almighty who had given us the strength to develop and complete this project successfully.

I express my gratitude to respected **Rev. Fr. Dr. S. Richard,** Secretary, Pope John Paul II College of Education, who has given the opportunity to take up this project work.

I express my gratitude to respected **Dr. D. Alexander,** Principal, Pope John Paul II College of Education, who has given the opportunity to take up this project work.

I am greatly indebted to **Mr. Maury Marie Immanuel,** Head of the Department of Computer Applications for her special concern for providing all the necessary facilities to pursue our project with great enthusiasm.

I would like to express my sincere heartfelt gratitude to my Internal Guide, **Mrs. Yazhini. S,** Assistant Professor, Department of Computer Applications for giving me enthusiastic encouragement and valuable suggestions for the project and for the preparation of the project report.

I have inadequate words to thank my parents and those who have contributed both directly and indirectly for the successful completion of this project.

Finally, I would like to thank my dear friends for their support and timely assistance for this project.

SYNOPSIS

The Car Rental System is a web-based application developed to automate and simplify the process of renting cars. The system enables users to view available vehicles, register, book cars online, and make payments, while allowing the administrator to manage vehicles, bookings, users, and feedback efficiently. In the existing manual system, maintaining records and tracking vehicle availability is time-consuming and error-prone.

This project overcomes those limitations by providing a centralized database-driven solution. The proposed system improves efficiency, reduces paperwork, ensures data accuracy, and provides a user-friendly interface for both customers and administrators. This project is developed as a DBMS-based mini project using PHP and MySQL. The rapid growth of information technology has transformed traditional business processes into automated and efficient systems.

The car rental industry, which was earlier dependent on manual record keeping and paperwork, now requires a computerized solution to manage increasing customer demands and operational complexity. The Car Rental Management System is developed to provide are liable, efficient, and user-friendly platform for managing car rental operations through a web-based application.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 About the Organization

The "Cars Rental Management System" is an academic project focused on creating a digital platform for browsing and renting vehicles. It provides customers with a seamless online experience to compare and book items (Hatchbacks, Sedans, SUVs, etc.) while offering administrators a robust tool for vehicle inventory and booking management. Developed using PHP, MySQL, and web technologies, this project helps students gain practical experience in full-stack development, database management, and modern UI design.

1.2 About the Project

The "Cars Rental Management System" platform is designed to bridge the gap between vehicle rental services and customers. It allows users to view available vehicles, check detailed specifications like fuel type, seating capacity, and pricing, and place bookings instantly. The system includes an administrative dashboard for monitoring all bookings, managing vehicle listings, and tracking booking statuses, ensuring a reliable and professional rental experience. Furthermore, the application optimizes business operations by automating booking confirmations and providing real-time analytics to help administrators make data-driven decisions. This comprehensive approach ensures high customer satisfaction while maintaining a streamlined workflow for the management team.

1.3 Plan of the Project

The project report is organized into the following chapters:

Chapter 1 – Introduction

Explains the project overview, purpose, organization background, and report structure.

Chapter 2 – Problem Definition & Feasibility Analysis

Identifies the need for the system, shortcomings of existing manual methods, and evaluates technical and economic feasibility.

Chapter 3 – Software Requirement Specification (SRS)

Describes functional requirements (what the system does), non-functional requirements, and the necessary hardware and software configurations.

Chapter 4 – System Design (Preliminary)

Covers fundamental design concepts such as system architecture, modularity, user interface flows, and entity-relationship diagrams (ERD).

Chapter 5 – System Design (Detailed)

Explains each module of the system in depth, including the User Booking Module, Owner Dashboard, Driver Portal, and Admin Control Panel.

Chapter 6 – Coding, Testing, and Implementation

Includes details on coding standards used, testing methodologies (Unit Testing, System Testing), and the final deployment process.

PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

CHAPTER 2

PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

2.1 INTRODUCTION

In today's digital era, the demand for convenient and reliable vehicle rental services has shifted from traditional offline booking methods to online platforms. The "Cars Rental Management System" is designed to address this transformation by digitizing the conventional car rental process into a centralized, efficient online ecosystem. This system bridges the gap between manual booking management and modern digital service needs, ensuring that both customers and administrators can interact seamlessly. By automating core business functions, the project aims to enhance operational transparency, provide real-time booking updates, and deliver a superior rental experience that meets the expectations of contemporary travelers. The platform eliminates the inefficiencies of phone-based reservations and paper-based record-keeping, offering a streamlined solution that benefits all stakeholders in the vehicle rental industry.

2.2 PROBLEM DEFINITION

The current manual system presents several significant problems that make renting vehicles inconvenient for users. Physical car rental offices suffer from limitations such as restricted parking space for displaying the full fleet, inability to operate 24/7, lack of instant price comparison across different vehicle categories, and the need for consumers to visit the office personally, which is time-consuming.

Traditional methods of managing vehicle rentals also have multiple drawbacks. Customers often face unclear pricing structures for different rental durations, have no way to verify the exact vehicle availability or specifications before visiting, and lack consolidated information about various car models. Another major issue is the absence of a unified platform where rental service providers can digitally manage their fleet and bookings. At present, many local rental services rely on manual ledgers or paper files, leading to booking discrepancies, lost reservation records, and inefficient vehicle tracking. This scattered approach creates unnecessary difficulties for both renters and service managers who prefer a smooth, effective online solution.

2.2.1 Existing System

In the traditional vehicle rental system, customers depend largely on local car rental agencies, printed brochures, or personal recommendations to make booking decisions. To verify vehicle availability or compare prices, customers are required to either visit multiple rental offices in person or contact them individually, which consumes considerable time and effort.

Such a manual approach lacks transparency, as customers are unable to clearly view detailed vehicle specifications such as fuel type, seating capacity, or model variations before booking. Additionally, the absence of a centralized online platform prevents users from accessing high-resolution vehicle images or tracking the status of their reservations digitally. From a business perspective, rental service providers face challenges in managing large fleets and multiple vehicle categories manually, often resulting in booking mismatches, delayed updates, and inefficient operational workflows.

2.2.2 Proposed System

The "Cars Rental Management System" is an integrated web platform that combines vehicle cataloging, online booking, and fleet management. It provides a digital showroom where customers can browse vehicle categories like Hatchbacks and SUVs with high-quality images and detailed specifications. This allows users to make informed decisions and place secure bookings directly from home, ensuring a 24/7 accessible rental experience.

For administrators, the system features a centralized dashboard to update vehicle listings, monitor real-time booking statuses, and manage reservation fulfillment. Users also gain access to a personal portal to track their booking history and live reservation statuses. This digital approach eliminates manual record-keeping inefficiencies, significantly improving operational speed and overall customer satisfaction.

2.3 FEASIBILITY STUDY

The Feasibility Study serves as a critical evaluation phase in the Software Development Life Cycle (SDLC) to ascertain the viability and practicality of the "Cars Rental Management System". By meticulously analyzing technical requirements, financial constraints, and operational needs, this study provides a clear roadmap for project decision-making. It ensures that the proposed solution is not only achievable within the given resources but also sustainable in the long term.

Furthermore, these evaluations play a vital role in risk management and strategic planning. By identifying potential challenges early in the development process, the feasibility study helps in optimizing resource allocation and improving implementation accuracy. Ultimately, this foundational assessment guarantees that the final system aligns with the intended business goals while remaining cost-effective and operationally sound.

2.3.1 OPERATIONAL FEASIBILITY

The Cars Rental Management System demonstrates high operational feasibility through its intuitive and user-centric design, catering to both contemporary travelers and traditional car rental service providers. By providing a streamlined interface for core features like vehicle browsing, availability verification, and booking tracking, the platform ensures ease of use and a minimal learning curve for all users. From an administrative perspective, the system simplifies complex fleet and reservation management workflows without requiring specialized technical training. Since it is entirely web-based, the system eliminates the need for proprietary hardware or complex software installations, making it a highly practical and scalable solution for real-world vehicle rental environments that enhances productivity and operational transparency.

2.3.2 TECHNICAL FEASIBILITY

The system is technically feasible because it is built using widely used and robust web technologies such as PHP, HTML, CSS, and JavaScript, backed by a MySQL database. These technologies are open-source, well-documented, and perfectly suitable for handling vehicle listings, user data, and dynamic content. The application requires only a standard web server setup (like Apache via XAMPP) and does not need expensive high-end infrastructure. The system efficiently handles the secure upload and storage of vehicle images and specifications using standard file handling protocols. Since the technologies involved are beginner-friendly and the setup does not require complex tools, the project can be successfully developed and implemented by students with basic web development knowledge.

2.3.3 ECONOMICAL FEASIBILITY

The "Cars Rental Management System" is economically viable as it is built entirely using open-source technologies such as PHP, MySQL, and Apache, which eliminates software

licensing fees and high development costs. By utilizing standard computer hardware and existing web server environments like XAMPP, the project avoids the need for significant capital investment in specialized infrastructure. Compared to traditional manual systems that incur recurring expenses for paper-based records, physical brochure printing, and intensive manual labor, this digital solution automates core business workflows to reduce operational overhead. By minimizing human errors in booking management and accelerating reservation processing, the system offers a substantial return on investment, providing a professional-grade management tool with minimal initial expenditure.

2.4 RECOMMENDED IMPLEMENTATION

Based on the feasibility results, the system should be implemented using a combination of HTML5, CSS3, and JavaScript for a responsive Front-End, and PHP for robust server-side logic. MySQL will serve as the database to store user, vehicle, and booking records. Authentication and data validation protocols should be implemented to ensure security. The system will be hosted on a local **XAMPP server** during development and testing. This approach ensures a scalable, secure, and cost-effective solution for simplifying the vehicle rental and management process.

SOFTWARE REQUIREMENT SPECIFICATION

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

Software requirements and specifications are foundational documents in the development lifecycle that provide a comprehensive blueprint of a system's functional and non-functional capabilities, operational constraints, and performance expectations. These documents serve as a detailed guide for designing, developing, and validating software by outlining the overall architecture, user interfaces, hardware dependencies, and critical security protocols necessary for successful implementation. By detailing everything from high-level system features to technical implementation details and maintenance plans, the software specification ensures that the development process remains aligned with organizational goals while maintaining technical integrity and long-term reliability.

3.1.1 Purpose

The primary objective of this Software Requirement Specification (SRS) is to establish a definitive and comprehensive framework for the development of the "Cars Rental Management System". By meticulously documenting every functional capability and operational constraint, this document serves as an indispensable reference for developers throughout the implementation cycle and provides a standardized benchmark for quality assurance teams to validate system performance. Furthermore, it acts as a strategic technical guide for project evaluators and administrators, ensuring that the system's architecture, workflow, and security measures are perfectly aligned with the intended business objectives. Ultimately, this SRS guarantees that all stakeholders possess a shared and precise understanding of the project's requirements, facilitating the delivery of a robust, secure, and highly efficient vehicle rental management solution.

3.1.2 Scope

The scope of the "Cars Rental Management System" encompasses a wide range of functionalities designed to modernize the vehicle rental experience by integrating an online vehicle catalog with a robust fleet management engine. Customers are provided with an interactive portal where they can seamlessly browse premium vehicle categories, including Hatchbacks, Sedans, SUVs, and Luxury Cars, each complemented by comprehensive specifications such as fuel type, seating capacity, and rental pricing. By incorporating secure

role-based authentication for both Customers and Administrators, the system ensures protected access to sensitive transactional and management features. Every booking processed through the platform is systematically recorded in a centralized database, allowing the administrator to exercise total control over vehicle categories, real-time availability adjustments, and booking oversight through an advanced dashboard. This unified architecture, supported by a powerful backend and an intuitive user interface, guarantees an efficient, secure, and seamless end-to-end digital rental environment.

3.1.3 Definition, Acronyms and Abbreviations

The "Cars Rental Management System" is a web-based platform designed to simplify the process of renting vehicles and managing fleet logistics. The system integrates two major modules: Vehicle Management and Booking Fulfillment. Throughout this documentation, several acronyms are used. HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) are used for structuring and designing the front end, while PHP (Hypertext Preprocessor) is used as the server-side scripting language. SQL (Structured Query Language) refers to the language used for handling database operations in MySQL. UI (User Interface) denotes the visual layout, whereas UX (User Experience) refers to the interaction quality. ID (Identification Number) is used as a unique reference for bookings, users, and vehicle items.

3.1.4 Overview

This Software Requirement Specification provides an extensive and systematic description of the "Cars Rental Management System", serving as the primary technical foundation for the entire development lifecycle. The document meticulously details the system's overall architectural perspective, its core functional capabilities, and the diverse characteristics of the user groups interacting with the platform. Beyond basic functionality, this overview encompasses a critical analysis of external interface requirements, performance benchmarks, and specific design constraints essential for maintaining system stability. By synthesizing these elements into a cohesive framework, the document ensures that all developmental efforts are precisely engineered to meet high efficiency standards while delivering a reliable and user-centric vehicle rental environment.

3.2 GENERAL DESCRIPTION

The "Cars Rental Management System" is an integrated web-based application designed to bridge the gap between traditional vehicle rental services and modern digital commerce by combining extensive cataloging capabilities with sophisticated fleet management features. This platform provides a centralized environment where users can effortlessly browse a diverse range of vehicles for personal and business travel, complemented by detailed specifications and real-time availability updates for every vehicle. To support these user-facing features, the system incorporates a dedicated administrative portal that empowers managers to oversee critical operations, including vehicle additions, availability adjustments, and comprehensive booking tracking. Developed with a high-performance technology stack, the application utilizes HTML, CSS, and JavaScript to deliver a responsive and intuitive front-end experience, while the back-end is driven by PHP for secure authentication, complex business logic, and seamless data processing. All critical system information, ranging from user profiles to transactional records, is securely maintained within a normalized MySQL database, ensuring superior data integrity, efficient retrieval, and consistent overall system performance.

3.2.1 Product Perspective

This system is a comprehensive web application designed to provide users with a complete platform for renting vehicles. Along with user features, it includes robust management tools for Rental Service Administrators. The application is built using modern web standards on the front-end for creating an interactive interface, while contemporary server-side technologies manage operations. All data, including vehicle profiles and user accounts, is stored and managed using a MySQL database to ensure an organized system functioning.

3.2.2 Product Functions

The system allows users to browse various vehicle items, search for specific categories, view detailed vehicle information, and place booking requests. Users can also view their booking history and track the status of current reservations. On the administrative side, the system provides functions such as secure login, the ability to manage vehicle listings, and oversee customer inquiries. Administrators can add new vehicles, update pricing, and track fleet availability. Together, these features ensure smooth operations for all stakeholders.

3.2.3 User Characteristics

There are two primary types of users in this system: Customers and Administrators. Customers require only basic computer and internet knowledge to browse the catalog and place bookings, making the platform accessible to a wide range of travelers. Administrators need a basic understanding of system management to update vehicle records, manage categories, and oversee overall platform activity. Both user groups benefit from a simplified navigation structure that requires no advanced technical training.

3.2.4 General Constraints

The "Cars Rental Management System" requires a stable internet connection for both customers and administrators to fully access all integrated features, including browsing the vehicle catalog and managing real-time bookings. The application is designed to operate exclusively on modern web browsers that support HTML5 and CSS3 to ensure a responsive and interactive user experience. Since the system includes file upload functionalities for high-resolution vehicle images, the maximum upload size is determined by the hosting server's configuration, and the PHP environment must have the file_uploads setting enabled to allow administrators to update the catalog successfully. These technical conditions are essential to ensure the system functions efficiently and maintains high performance within the required operational environment.

3.2.5 Assumption and Dependencies

The system is designed with several key assumptions to ensure smooth and reliable operation. It is assumed that the Administrators have authorized access to their respective dashboards. For Customers, it is expected that they provide accurate contact details when placing bookings to ensure correct reservation confirmation. On the technical side, the hosting server must fully support PHP and MySQL. Additionally, the system assumes that the user's web browser supports JavaScript for interactive elements such as form validations and dynamic filtering.

3.3 SPECIFIC REQUIREMENTS

3.3.1 Behavioural Requirements

The "Cars Rental Management System" is engineered to facilitate flexible user interactions, allowing travelers to browse the complete vehicle catalog and verify rental pricing

without the requirement of immediate payment. The system supports a seamless registration process and provides users with persistent access to track their real-time booking status. On the administrative front, behavioral integrity is maintained through mandatory secure authentication, ensuring that only verified staff can access the management dashboard. Once authenticated, administrators operate within an isolated and protected environment where they can independently execute tasks such as availability updates, user account oversight, and fleet control without external system interference.

3.3.2 Software Requirements

Component	Tools for Application
Technology	Open-Source Technology
Front-End Technology	HTML, CSS, JavaScript
Back-End Technology	PHP
Server and Database	XAMPP (Apache) | MYSQL
Browser	Chrome, Edge, Firefox

3.3.3 Hardware Requirements

Component	Specification
System	Any Standard Laptop/PC
Hard Disk	250 GB
Monitor	15.6 LCD or Standard Display
Input Devices	Keyboard, Mouse/Touchpad
RAM	4 GB

3.3.4 External Interface Requirements

The system provides a user-friendly interface that includes a fully responsive layout, attractive vehicle cards, and easy navigation across all pages. In terms of hardware

compatibility, the application works seamlessly on Windows, Linux, and macOS systems. It can be accessed using a computer, laptop, or mobile device. The software interface supports smooth interaction between the frontend and backend components, enabling efficient system operation for rental and management tasks.

3.3.5 Performance Requirements

The system is designed to deliver a smooth and efficient user experience with fast response times for all operations such as searching vehicles or placing bookings. A major requirement is that no error or unexpected issue should cause the application to crash. Whenever an error occurs during form submission or database operations, the system provides a clear error message to guide the user. Furthermore, thorough testing ensures that interconnected features like availability deduction and booking status updates work perfectly.

3.3.6 Design Constraints

Databases must be designed in a highly efficient manner so that vehicle search and retrieval time is minimized, ensuring fast access even with a large fleet. All calculations such as total rental costs and taxes must be accurate. The overall system design is kept simple and intuitive so that non-technical users can navigate without confusion. At the same time, the interface maintains a professional and trustworthy aesthetic for all customers.

3.3.7 Attributes

The system is designed to be reliable, ensuring stable performance without frequent crashes and maintaining high availability for 24/7 access. Strong security measures protect user data such as passwords and contact info. The application focuses on usability with an intuitive and visually appealing interface. It is built with maintainability in mind, enabling developers to update categories or add new features easily. Additionally, the system is scalable to handle an increasing number of users and vehicles as the business grows.

3.3.7.1 Security

Security is enforced at multiple levels to protect sensitive data. Only authorized users with valid login details can access their accounts. Admin-level features such as updating fleet availability are accessible only to verified administrators. The system checks user identity during every login attempt through session management to prevent unauthorized access to private dashboards.

3.3.7.2 Reliability

The "Cars Rental Management System" is engineered to deliver highly reliable performance by ensuring consistent and accurate results across all core modules, from initial vehicle browsing to final booking fulfillment. The software utilizes robust error-handling mechanisms and secure transaction protocols to maintain operational integrity even during peak usage. While the application's internal logic is designed for maximum stability, overall reliability is further supported by the underlying server infrastructure and network consistency, ensuring that user actions are processed without unexpected failures or data loss.

3.3.7.3 Availability

To ensure continuous accessibility for a diverse user base, the system is architected for high availability, allowing customers to browse the vehicle catalog and place bookings at any time. The application features a fully responsive design that maintains functional parity across various devices, including desktops, tablets, and mobile smartphones. By utilizing an optimized MySQL database structure, the system ensures rapid data retrieval and minimal downtime, providing persistent access to vehicle information and account management features 24/7.

3.3.7.4 Maintainability

The architecture of the "Cars Rental Management System" prioritizes long-term maintainability through a modular design approach that clearly separates the user and administrative components. By maintaining a low degree of coupling between system modules, the platform allows developers to implement updates, security patches, or feature enhancements independently without risking the stability of the entire system. This structured codebase ensures that the application remains adaptable to future business needs while simplifying the debugging and optimization processes.

3.3.7.5 Portablility

Designed for maximum portability, the platform ensures a consistent user experience across different operating systems and modern web browsers. Since the application is built using standard web technologies like PHP and HTML, it can be seamlessly deployed on any server environment that supports a PHP-based stack. The use of lightweight, standard databases allows the system to operate efficiently on various hardware configurations without requiring specialized proprietary software, facilitating easy migration or scaling.

3.3.7.6 Reusability

The system's development adheres to high reusability standards, with core functionalities such as database connections, authenticated session handling, and UI components written as modular units. By creating generic code structures for repetitive tasks—such as vehicle card rendering and search filtering—the system allows for efficient feature expansion. This reusable architecture not only accelerates the development of future updates but also ensures code consistency and reduces the likelihood of redundancy throughout the application.

3.3.7.7 Efficiency

Operational efficiency is a core attribute of the system, achieved through optimized server-side logic and localized database processing that minimizes response times for all user interactions. By utilizing normalized database tables and efficient SQL queries, the application ensures fast data access and minimal storage consumption. This resource-efficient design contributes to a seamless and responsive user experience, allowing both customers and administrators to perform complex tasks like fleet lookups and booking processing with high speed and low latency.

3.3.8 Other Requirements

The "Cars Rental Management System" incorporates several supplementary requirements to ensure backend performance and secure data handling. This includes utilizing a lightweight MySQL database engine for structured storage and implementing secure PHP session management for user state persistence. The overall application design follows modular programming principles, ensuring that common system utilities are accessible across different modules to maintain a unified and performant operating environment.

3.3.8.1 Database

The system utilizes a MySQL database as its primary storage engine, chosen for its efficiency, reliability, and capability to handle large volumes of vehicle records and transactional data. The database schema is carefully normalized into dedicated tables for users, vehicles, categories, and bookings to prevent data redundancy and ensure referential integrity. This structured approach facilitates quick search indexing and secure data management, which are essential for maintaining a high-performance vehicle rental platform.

3.3.8.2 Code reuse

A significant emphasis is placed on code reuse within the "Cars Rental Management System" to streamline development and ensure long-term scalability. Generic functions for tasks like data sanitization, UI rendering, and session validation are implemented as shared components that can be called across multiple modules. This modularity ensures that the code for features like vehicle display lists or administrative filters can be easily adapted for future enhancements, significantly reducing the maintenance overhead and improving the overall development efficiency.

SYSTEM DESIGN

PRELIMINARY

CHAPTER 4

SYSTEM DESIGN PRELIMINARY

4.1 INTRODUCTION

The system design phase serves as a critical transition between requirement gathering and actual implementation, converting abstract software specifications into a technical blueprint for the "Cars Rental Management System". This chapter provides a comprehensive overview of the system's structural framework, encompassing its high-level architecture, core design principles, user interface aesthetics, and foundational database schema. By prioritizing modularity, security, and scalability, the design ensures that the platform remains easily maintainable and highly adaptable to future technological advancements. This proactive approach guarantees that complex enhancements, such as new vehicle categories or advanced payment modules, can be integrated seamlessly without disrupting the system's existing operational integrity.

4.2 BASIC DESIGN APPROACH

This project utilizes a modular and layered architecture to ensure that each component is separated into well-defined modules for better organization and maintainability. The system is architected into three primary layers: the presentation layer, the application logic layer, and the data layer, which allows for smooth interaction between the front-end user interface, back-end processing, and centralized database operations. This structured approach significantly improves scalability, simplifies the debugging process, and supports future technical enhancements by keeping the code highly organized.

Layer 1 – Presentation Layer

The interface includes the homepage, vehicle catalog, vehicle details, and dashboards for Customers and Administrators. Developed with HTML, CSS, and JavaScript, it ensures a clean, responsive, and user-friendly experience across all devices.

Layer 2 – Application/Business Logic Layer

This layer handles core functions like vehicle management, secure booking processing, authentication, and form validation. Powered by PHP, it ensures reliable server-side processing and smooth interaction between the UI and database.

Layer 3 – Data Layer

A MySQL database manages all essential records, including users, vehicle categories, individual vehicles, and booking logs. It facilitates efficient data storage, retrieval, and management to ensure high system performance.

4.3 DESIGN CONCEPTS

For an effective and robust system, a modular design approach was followed throughout the development of the "Cars Rental Management System". The entire application is divided into independent, well-structured modules so that components like vehicle management, booking processing, and user authentication can function separately yet integrate seamlessly within the unified platform.

4.3.1 Modularity

The project is organized into clear modules to ensure smooth functionality and easy maintenance across the platform. The Customer Module manages user accounts and booking history, providing a personalized rental experience, while the Vehicle Module handles vehicle listings and real-time availability updates for categories such as Hatchbacks, Sedans, and SUVs. The Booking Module serves as the transaction engine, processing rental requests and calculating total costs, while the Category Module allows for efficient organization of the vehicle catalog. Finally, the Admin Module provides complete oversight of the platform, including vehicle approvals, user management, and detailed booking monitoring through a centralized control center.

4.3.2 Coupling

The system modules are designed with loose coupling, meaning changes in one module—such as updating the vehicle card design—do not negatively impact others like the booking processing logic. By minimizing direct dependencies and passing data strictly through defined parameters, the system ensures high stability and allows for parallel development without functional conflicts.

4.3.3 Cohesion

Cohesion refers to how focused the responsibilities of a module are, with this system maintaining high cohesion by assigning specific functions to each module. The table below outlines the primary modules and their dedicated functions:

Module	Function
Vehicle	Add, update, delete, search, and display vehicle items
Bookings	Process vehicle rental requests and store booking details
Categories	Manage and organize various vehicle types (Hatchback, SUV, etc.)
Admin	Oversee fleet availability, manage users, and track booking performance
Customers	Manage personal accounts, view booking history, and track status
Authentication	Handle secure Login, Logout, and User Session management

4.4 USER INTERFACE DESIGN

4.4.1 User Homepage

The user interface includes a clear and visually appealing layout consisting of a professional logo and a navigation bar for effortless movement across pages. It features a prominent search bar and dynamic filters based on categories like Hatchbacks, Sedans, and SUVs, allowing users to quickly find suitable vehicles. Featured items are displayed in high-quality cards that showcase the vehicle image, model name, specification highlights, and rental pricing in a clean, modern aesthetic.

4.4.2 Vehicle Detail Page

The vehicle detail page provides a high-resolution image of the selected vehicle along with a comprehensive description covering features such as fuel type, seating capacity, and transmission type. It displays the rental price and availability status in an organized, easy-to-read format. A prominent "Book Now" button allows users to proceed directly to the reservation stage, while integrated customer reviews and ratings help ensure vehicle quality and transparency before making a decision.

4.4.3 User Profile Management

The User Profile Management section serves as a dedicated portal where customers can manage their personal authentication and contact details. It features a modern and intuitive interface that allows users to view their account information, including their registered email and join date. This section specifically focuses on enabling users to update their full name and mobile number, ensuring that their records remain accurate for booking confirmation purposes. By providing a streamlined and focused profile management space, the system ensures that user data is maintained securely and is easily accessible for administrative reference.

4.4.4 Booking Form

The booking form collects all necessary details to process a vehicle rental, including fields for the Pickup Location, Pickup Date, Return Date, Duration, Phone Number, and Destination. It also re-confirms the Customer's contact details to ensure accurate reservation communications. The form incorporates real-time field validation to prevent data entry errors and automatically calculates the total rental value, including any applicable taxes or additional fees, based on the duration and vehicle price.

4.4.5 Admin Dashboard

The Admin Dashboard serves as the command center for the entire platform, allowing administrators to oversee all business operations from a single interface. It provides tools to manage vehicle categories, update vehicle listings with new availability status, and monitor customer bookings from placement to final completion. The dashboard also facilitates the generation of booking reports and fleet analytics, helping the administration make informed decisions regarding fleet expansion and platform growth.

4.4.6 Vehicle Listing Page

The Vehicle Listing page acts as a comprehensive catalog where users can explore the entire range of vehicle offerings systematically. It features advanced categorization and filtering capabilities, allowing renters to narrow down their search by specific types such as Economy, Luxury, or SUV categories. Each vehicle in the list is displayed with a concise summary including its name, rental price, and availability status, encouraging users to click through for deeper details and facilitating a comparison-based rental experience.

4.4.7 My Bookings Page

The My Bookings page provides a secure and organized environment for customers to monitor their rental history and current booking statuses. This specialized interface displays detailed information for every booking, including the unique Booking ID, the specific vehicle rented, the total transaction amount, and real-time fulfillment status such as Pending, Approved, or Completed. By centralizing all booking data, the system ensures that users remain fully informed about their rental timelines and past transactions at all times.

4.4.8 Login and Registration Interface

The Login and Registration interface is designed to ensure a secure and streamlined entry point for all users of the "Cars Rental Management System". It features a clean, minimalist layout with dedicated forms for existing users to authenticate via their email and password, as well as for new customers to create accounts by providing essential registration details. The interface prioritizes security through real-time input validation and secure session handling, ensuring a protective barrier for personal user information from the moment of first interaction.

4.5 DATABASE DESIGN

Proper database design plays a critical role in system performance, as the database is normalized to eliminate data redundancy and ensure rapid retrieval during vehicle searches and booking processing. The system ensures data integrity through foreign key constraints, preventing inconsistencies across user, vehicle, and booking records. Furthermore, sensitive information like customer passwords is secured using modern encryption and hashing algorithms (MD5/bcrypt), maintaining high standards of data protection throughout the application lifecycle.

4.6 SYSTEM ENVIRONMENT

Hardware Environment

The system requires a basic hardware setup to run efficiently, with a laptop or desktop computer being sufficient for both development and operation. On the client side, any device such as a mobile or laptop with internet access can be used to access the platform, while for

the server side, a minimum of 4GB RAM is recommended to host the database and web server efficiently.

Software Environment

The application is designed for compatibility with standard web environments and runs smoothly on major operating systems such as Windows, Linux, and macOS. It requires a local server stack like XAMPP or WAMP supporting PHP 7.4 or higher, with data managed through MySQL 5.7+ or MariaDB. On the user's end, the system is fully compatible with all contemporary web browsers, including Google Chrome, Microsoft Edge, and Mozilla Firefox, ensuring a consistent cross-platform experience.

Development Environment

The development cycle for this project primarily utilizes Visual Studio Code as the primary Integrated Development Environment (IDE) due to its robust support for PHP and web technologies. Database administration and schema management are handled through the phpMyAdmin interface, while Git can be optionally utilized for version control. This standardized toolset ensures a reliable workflow for building, testing, and maintaining the Cars Rental Management System.

SYSTEM DESIGN

DETAILED

CHAPTER 5

SYSTEM DESIGN DETAILED

5.1 INTRODUCTION

This chapter provides a comprehensive and detailed description of the core modules within the "Cars Rental Management System". Each module is analyzed in terms of its functional purpose, processing logic, and role within the overall system architecture. This detailed breakdown serves as a precise guide for developers and evaluators to understand the internal workflows, data interactions, and modular dependencies that ensure the system operates as a unified and efficient vehicle rental platform.

5.2 MODULE-PURPOSE AND DESCRIPTION

This section outlines the functional details of the core modules that comprise the "Cars Rental Management System", describing how they facilitate user interactions and maintain database integrity through structured logic.

5.2.1 Home Page and Navigation Module

The primary purpose of this module is to serve as the portal entry point for all users, providing a visually engaging interface that highlights featured vehicles and active categories. It utilizes a centralized navigation framework that allows users to seamlessly transition between the vehicle catalog, about us section, and contact pages. By accurately fetching real-time data from the database, the module ensures that users are always presented with current offerings and can easily access their personal accounts or the booking interface.

5.2.2 Vehicle Catalog and Search Module

The Vehicle Catalog and Search Module is designed to organize the fleet of vehicles into a searchable and filterable database. Its core function is to handle complex queries that sort items by category, price, and vehicle specifications, providing customers with a streamlined method to find specific vehicles for their travel needs. The processing logic ensures that only available items are displayed, and detailed vehicle cards are dynamically generated to provide a consistent and high-quality browsing experience.

5.2.3 User Authentication and Profile Module

This module manages the security and personalization aspect of the system by handling user registration, login authentication, and session persistence. It utilizes secure hashing algorithms to protect user credentials and provides an interface for customers to maintain their profiles, including updating contact information and viewing booking history. The module operates as a gateway, ensuring that sensitive transactional features and personal dashboards are accessible only to verified individuals, thereby maintaining system-wide data privacy.

5.2.4 Booking Management and Processing Module

The Booking Management module serves as the primary transaction engine, capturing customer rental requests and translating them into structured booking records. It calculates total costs based on vehicle pricing and duration, validates delivery/pickup information, and updates the availability status in real-time to reflect successful rentals. The module tracks the lifecycle of a booking from initial placement through various fulfillment stages like Pending, Approved, and Completed, providing both the customer and the administrator with transparency regarding transaction progress.

5.2.5 Admin Fleet and Dashboard Module

The Admin Module acts as the centralized control center for the platform, empowering administrators to manage the complete lifecycle of vehicles and categories. It provides a secure environment for adding new items, updating availability status, and monitoring system-wide booking analytics. The dashboard's processing logic aggregates transactional data to provide insights into popular vehicle models and revenue trends, while the fleet controls ensure that the digital catalog accurately reflects the physical vehicles available in the fleet.

5.2.6 Contact and Inquiry Module

This module facilitates communication between users and the administration by processing messages, feedback, and support inquiries submitted through the contact interface. Each inquiry is systematically stored in the database with a unique timestamp and user reference, allowing administrators to track and respond to customer needs effectively. This ensures a high level of customer service and provides the business with valuable insights into user preferences and system performance.

5.3 EXCEPTION HANDLING

The developmental framework of the "Cars Rental Management System" incorporates a sophisticated exception handling strategy designed to encapsulate technical failures and provide a fault-tolerant user experience. To maintain structural integrity, the system implements customized redirection protocols that gracefully manage navigation anomalies and lead users back to functional system paths in the event of broken links or restricted access attempts. Database security is strictly prioritized through the abstraction of low-level SQL errors; instead of exposing raw backend schema metadata, the system generates standardized, user-centric alerts that mask underlying query failures from potential external threats. Furthermore, the system employs a hierarchical validation model where initial client-side scripts identify immediate data entry errors, followed by comprehensive server-side sanitization to neutralize complex security risks. This proactive management extends to authentication exceptions, where session expirations and unauthorized access attempts are automatically intercepted and redirected to the secure login gateway, ensuring that the platform’s administrative and personal modules remain resilient against operational disruptions.

5.4 SECURITY

The "Cars Rental Management System" implements a multi-faceted security architecture designed to ensure the confidentiality, integrity, and availability of all organizational and user-specific data. At the core of its defensive layer, the system utilizes advanced cryptographic hashing algorithms to secure user passwords, thereby neutralizing the risk of credential exposure during potential database breaches. Protection against prevalent web exploits, including SQL Injection and Cross-Site Scripting (XSS), is achieved through the systematic application of prepared statements and multi-stage input sanitization across all transactional entry points. Furthermore, administrative and customer operations are strictly delineated via a robust server-side **Role-Based Access Control (RBAC)** framework, ensuring that high-level fleet and booking modules are accessible only to verified personnel. By integrating secure, session-persistent authentication with high-standard data encryption protocols, the platform establishes a resilient and trustworthy environment that effectively safeguards the digital assets of both the enterprise and its customers.

CODING, TESTING AND IMPLEMENTATION

CHAPTER 6

CODING, TESTING AND IMPLEMENTATION

6.1 CODING

The development of the "Cars Rental Management System" adheres to rigorous coding standards to ensure that the codebase remains maintainable, readable, and highly scalable. By following a uniform set of software engineering principles, the project minimizes technical debt and ensures that the system architecture can support high-performance operations without style conflicts or structural inconsistencies. The development process utilizes a modular approach, where clean and well-documented PHP and JavaScript code ensures that individual components—such as the fleet management engine or the user booking logic—can be easily optimized or expanded. This adherence to standardized coding practices guarantees that the platform remains stable under varying loads while facilitating a more efficient long-term maintenance cycle for future system upgrades.

6.1.1 Naming Convention

Consistency in naming is the foundation of a clean codebase, and the "Cars Rental Management System" enforces strict architectural rules to ensure the identity of every system element remains intuitive and professional. All PHP variables are designated as nouns describing their data content, such as `$car_id` or `$booking_amount`, and follow a rigid `snake_case` convention to maximize readability across the backend logic. In contrast, functions and methods utilize `camelCase` and are strictly prefixed with verbs, such as `calculateBookingPrice()` or `updateAvailabilityStatus()`, to clearly signify their functional actions within the system. Advanced components, including object orientations and classes, are distinguished by `PascalCase` naming, while global configuration constants are maintained in `UPPER_SNAKE_CASE` for immediate recognition. Furthermore, to ensure seamless cross-platform compatibility across diverse operating systems, all database tables and columns strictly adhere to lowercase `snake_case`, effectively preventing potential functional conflicts in case-sensitive server environments.

6.1.2 Comments

The "Cars Rental Management System" utilizes a structured commenting strategy that serves as a technical roadmap for the codebase. Every primary source file begins with a standardized header detailing its purpose, while function definitions include DocBlocks to

specify parameters and return types. For complex logic, such as availability reconciliation or rental calculations, in-line annotations provide clarity on processing steps. This focused approach to internal documentation avoids redundancy and ensures long-term system maintainability by documenting critical architectural decisions and business rules.

6.1.3 Statement Constructions and Indentation

The architectural layout of the "Cars Rental Management System" is governed by a rigorous structural convention to optimize visual clarity and streamline the debugging process across diverse file types. By implementing a standardized 4-space indentation protocol, the system ensures precise alignment for nested control structures, effectively delineating logical hierarchy within the source code. The project adheres to a hybrid K&R bracing methodology, where terminal delimiters are isolated to maintain structural transparency, complemented by strategic whitespace management that separates distinct logical segments into manageable units. To eliminate potential functional ambiguities, the system mandates the use of explicit braces for all conditional evaluations while strictly enforcing depth limitations on nested logic to encourage modular refactoring and maintain a high standard of execution transparency.

Example:

```
function updateBookingStatus($booking_id, $new_status)
{
    if (empty($booking_id) || empty($new_status)) {
        return false;
    }
    if ($new_status === 'approved') {
        $result = executeQuery(
            "UPDATE booking SET status='approved' WHERE book_id=$booking_id"
        );
        if ($result) {
            logActivity("Booking $booking_id approved successfully.");
        }
```

```
    }
    return true;
}
```

6.2 TESTING

The testing phase of the "Cars Rental Management System" involves a systematic validation process to ensure the platform's reliability and architectural integrity. By utilizing a multi-tiered methodology, the project identifies and resolves logical anomalies across all functional modules. This structured approach ensures that every system component—from fleet management to secure transactions—operates with high-level precision, providing a resilient and seamless environment for all stakeholders.

6.2.1 Walkthroughs and Inspections

During the early stages of development, formal walkthroughs and code inspections were conducted to maintain high architectural standards. This involved systematic peer reviews of the database schema, system architecture, and core logic modules to identify potential vulnerabilities. By manually tracing the data flow from the user interface to the backend database, the development team successfully identified logical flaws and structural inconsistencies before the final coding phase. This proactive approach significantly reduced the likelihood of structural errors and ensured that the implementation remained perfectly aligned with the initial SRS objectives.

6.2.2 Unit Testing

The unit testing phase focused on validating individual functions and components of the "Cars Rental Management System" in isolation. By testing each PHP class and JavaScript utility independently, the project ensured the precision of core logic, such as rental price calculations and authentication protocols. For example, database connectivity and image processing functions were rigorously verified across multiple scenarios to ensure structural stability. This systematic isolation allowed for the early detection and remediation of logical errors, ensuring that only stable code segments were integrated into the broader system architecture.

6.2.3 Integration Testing

The integration testing phase of the "Cars Rental Management System" validated the synergetic interaction between diverse architectural modules to ensure a unified, high-performance ecosystem. For instance, testing verified that customer booking placements triggered instantaneous data propagation across the relational database, reflected as real-time updates on the administrative dashboard. By verifying these transactional interfaces, the project ensured that individual functional components formed a cohesive and reliable rental management infrastructure.

6.2.4 Validation Testing

Validation testing authenticated the system's compliance with the functional requirements and business logic established in the SRS documentation. By executing targeted test cases, the platform confirmed that critical operations—such as dynamic availability updates and role-based security—functioned precisely according to project specifications. This process successfully verified that the software fulfills the organizational objectives while providing a robust and reliable user experience for all stakeholders.

6.2.5 System Testing

System testing involved evaluating the complete, fully integrated application in an environment that simulated real-world operational usage. The platform was rigorously tested for performance, security, and responsiveness across multiple web browsers, including Google Chrome, Microsoft Edge, and Mozilla Firefox. Stress tests were conducted to verify that the platform could handle concurrent booking requests and availability updates, while comprehensive security audits ensured that authentication and data sanitization measures remained effective against common threats. This final testing stage confirmed that the entire product was stable, secure, and ready for deployment.

6.3 IMPLEMENTATION

The implementation of the "Cars Rental Management System" followed a systematic deployment strategy designed for modern web environments. The project utilized the XAMPP stack, encompassing Apache, MariaDB, and PHP, to ensure a stable and widely compatible runtime environment. During the implementation phase, the local database schema was migrated to the production server environment, and all configuration files were updated to reflect the live system credentials. A thorough smoke test was performed post-deployment to

verify that core functionalities—such as vehicle searching, booking placement, and admin authentication—were operational in the live environment. The modular architecture of the codebase facilitated a smooth transition, allowing for future scalability and ease of maintenance as the system evolves.

6.3.1 Deployment Environment

The "Cars Rental Management System" is deployed on a standardized technical environment designed for optimal performance and security. The platform utilizes an Apache web server with `mod_rewrite` for secure URL routing, supported by a PHP 7.4+ backend for advanced data processing and secure authentication. Data persistence is managed through a MySQL or MariaDB database using the InnoDB engine to ensure transactional integrity and ACID compliance. This unified configuration provides a stable and reliable runtime environment, ensuring seamless operations for both travelers and system administrators.

6.3.2 Installation Steps

1. Install **XAMPP**
2. Start **Apache** and **MySQL**
3. Place project folder inside:
   C:\xampp\htdocs\rental.pdf\
4. Import the SQL database
5. Config: Update `connection.php` with local DB credentials.
6. Run the system:
   http://localhost/rental.pdf/

CONCLUSION AND FORESEEABLE ENHANCEMENTS

CHAPTER 7

CONCLUSION AND FORSEEABLE ENHANSEMENTS

7.1 Conclusion

The development of the "Cars Rental Management System" has successfully established a robust, professional, and technologically advanced digital solution for the modern vehicle rental industry. By integrating a modular PHP-based architecture with a secure MySQL database, the platform effectively bridges the gap between traditional vehicle rental services and contemporary e-commerce expectations. The system provides a seamless and visually premium experience for customers to browse, select, and book vehicles, while simultaneously offering administrators a comprehensive toolset for fleet management, booking tracking, and customer communication. Ultimately, the project achieves its core objective of enhancing organizational efficiency, ensuring data integrity, and providing a scalable foundation that can adapt to the evolving demands of the digital marketplace.

7.2 Foreseeable Enhancements

While the current iteration of the "Cars Rental Management System" serves as a comprehensive rental platform, several future enhancements have been identified to further elevate its functional depth and user engagement. One primary direction involves the integration of Real-time GPS Tracking, allowing customers to track their rented vehicles and providing administrators with precise fleet location data. Additionally, the implementation of AI-driven recommendation engines can provide personalized travel suggestions and vehicle choices based on user rental history and preferences. Future updates could also encompass the development of dedicated mobile applications for iOS and Android, full integration with diverse global payment gateways, and the inclusion of advanced analytics dashboards for administrative personnel to monitor rental trends and fleet utilization through machine learning algorithms.

BIBLIOGRAPHY

REFERENCE

1. PHP Manual: https://www.php.net/manual

2. MySQL Reference: https://dev.mysql.com/doc/refman/8.0/en/

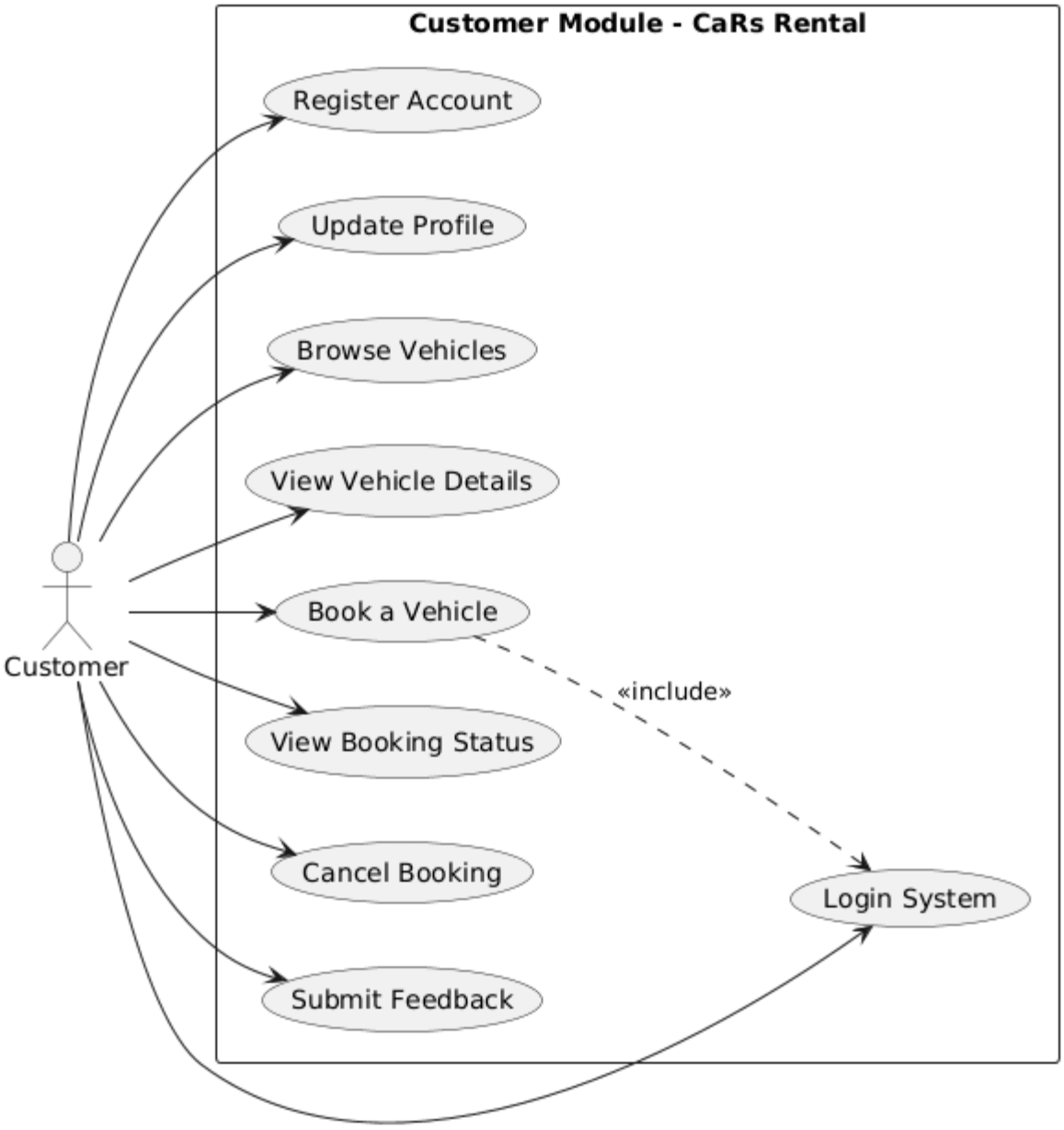
3. W3Schools: (HTML/CSS/JS Tutorials) https://www.w3schools.com

APPENDIX A

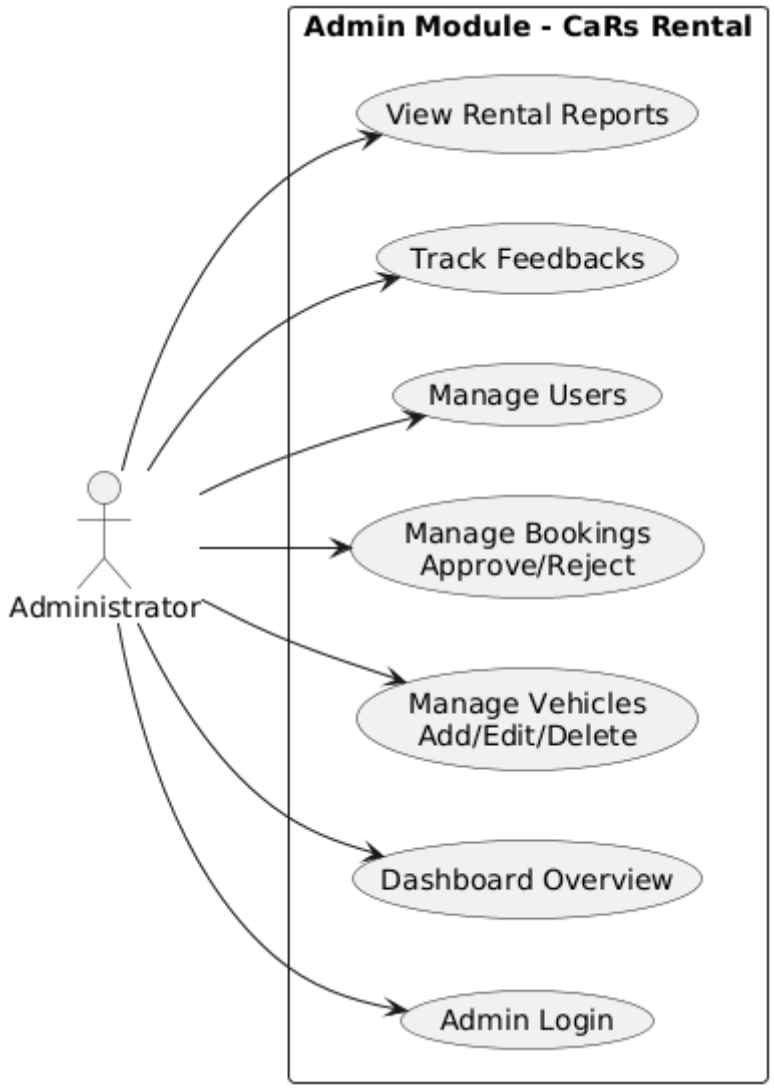
USE CASE DIAGRAMS

USE CASE DIAGRAMS

1. USER USE CASE DIAGRAM



2. ADMIN USE CASE DIAGRAM



APPENDIX B

DATA DICTIONARY

DATA DICTIONARY

1. ADMIN TABLE

Field Name	Data Type	Description
ADMIN_ID	VARCHAR	Unique ID for Administrator Login
ADMIN_PASSWORD	VARCHAR	Secure password for Admin access

2. USERS TABLE

Field Name	Data Type	Description
FNAME	VARCHAR	Customer's First Name
LNAME	VARCHAR	Customer's Last Name
EMAIL	VARCHAR	Unique Email Address (used for Login)
LIC_NUM	VARCHAR	Driver's License Number
PHONE_NUMBER	BIGINT	Contact Number
PASSWORD	VARCHAR	Encrypted Login Password
GENDER	VARCHAR	Gender (Male/Female)

3. FEEDBACK TABLE

Field Name	Data Type	Description
FED_ID	INT	Unique Feedback Reference ID
EMAIL	VARCHAR	Reference to Users Table
COMMENT	TEXT	Descriptive Feedback from Customer

4. CARS TABLE

Field Name	Data Type	Description
CAR_ID	INT	Unique ID for each Car (Auto Increment)
CAR_NAME	VARCHAR	Car Brand and Model Name
FUEL_TYPE	VARCHAR	Type of Fuel (Petrol/Diesel/Gas)
CAPACITY	INT	Number of Seating Capacity
PRICE	INT	Rental Price per Day
CAR_IMG	VARCHAR	Image Filename of the Vehicle
AVAILABLE	VARCHAR	Current Availability (Y/N)

5. PAYMENT TABLE

Field Name	Data Type	Description
PAY_ID	INT	Unique Payment Transaction ID
BOOK_ID	INT	Reference to Booking Table
CARD_NO	VARCHAR	Card Number used for Payment
EXP_DATE	VARCHAR	Card Expiry Date
CVV	INT	Card Security Code
PRICE	INT	Total Amount Successfully Paid

6. BOOKING TABLE

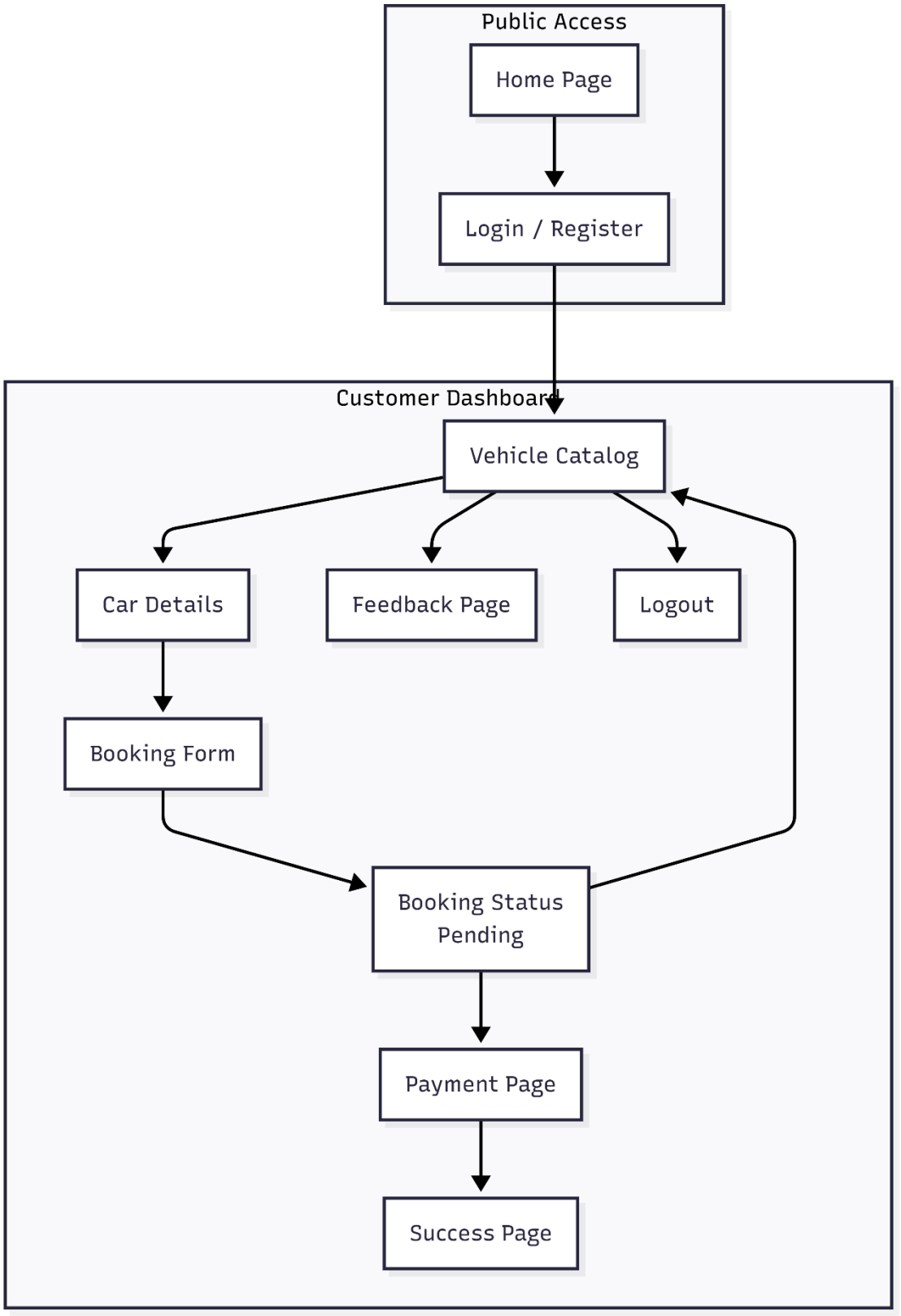
Field Name	Data Type	Description
BOOK_ID	INT	Unique Booking Reference ID
CAR_ID	INT	Reference to Cars Table
EMAIL	VARCHAR	Reference to Users Table
BOOK_PLACE	VARCHAR	Pickup Location Name
BOOK_DATE	DATE	Starting Date of Rental
DURATION	INT	Number of Days Rented
PHONE_NUMBER	BIGINT	Contact Number for Booking
DESTINATION	VARCHAR	Trip Destination
RETURN_DATE	DATE	Expected Vehicle Return Date
PRICE	INT	Total Rental Cost Calculated
BOOK_STATUS	VARCHAR	Status (Pending/Approved/Returned)

APPENDIX C

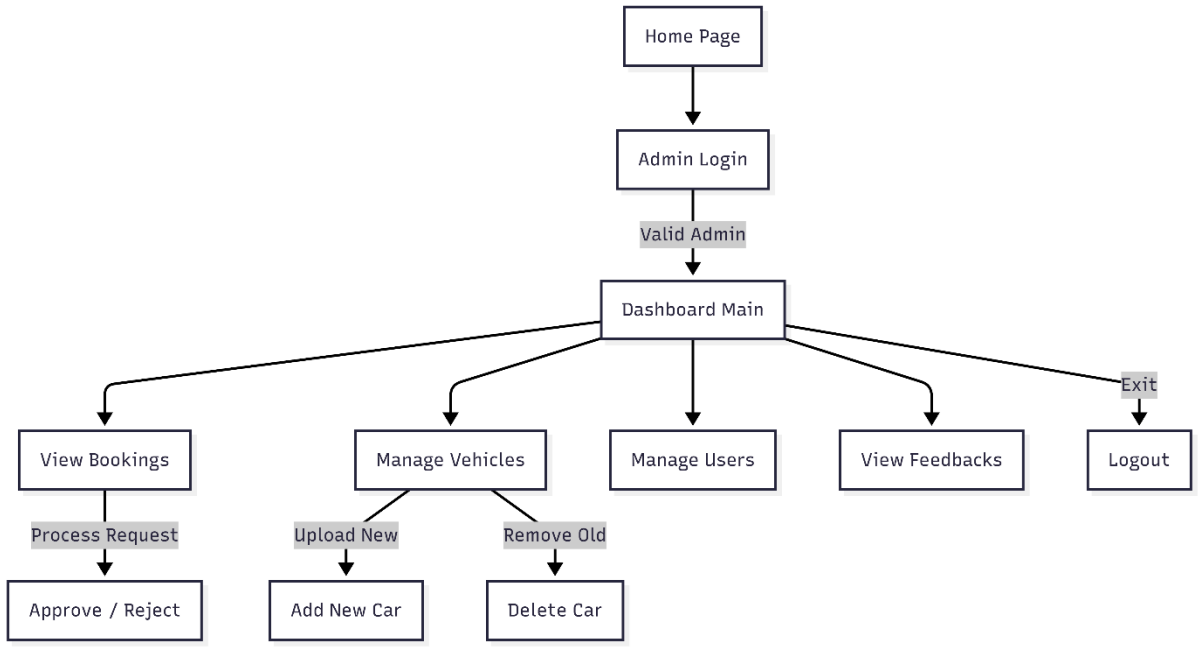
PAGE FLOW DIAGRAMS

PAGE FLOW DIAGRAMS

1. USER PAGE FLOW DIAGRAM



2. ADMIN PAGE FLOW DIAGRAM

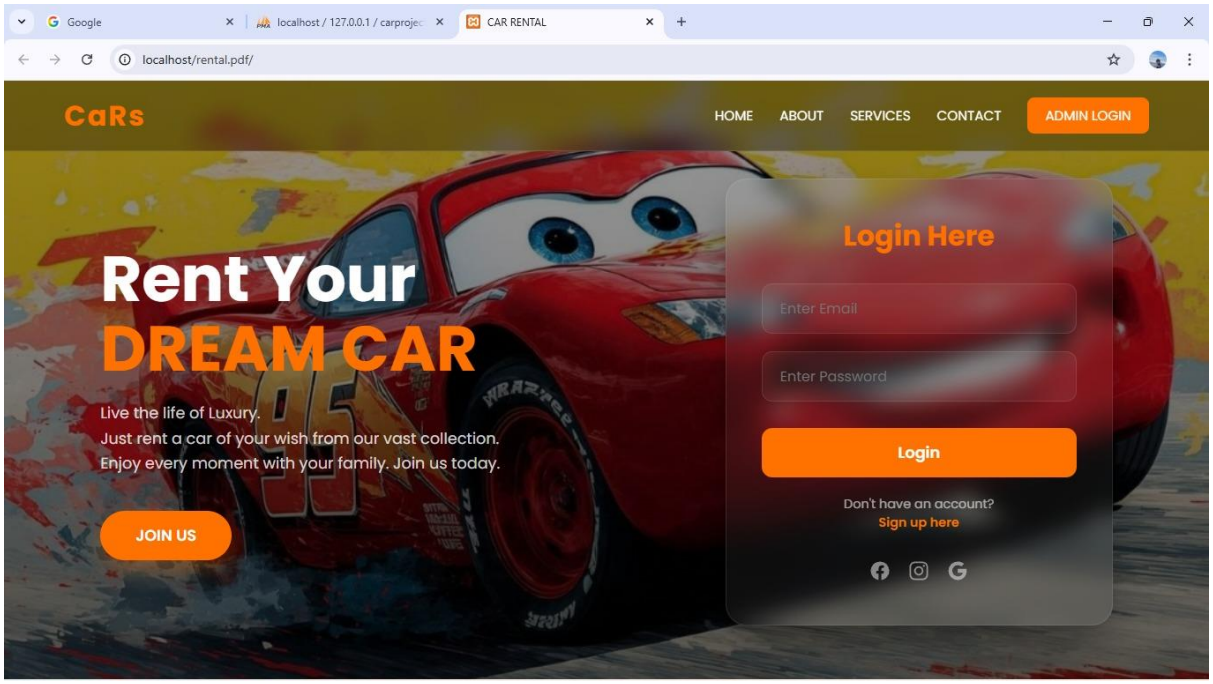


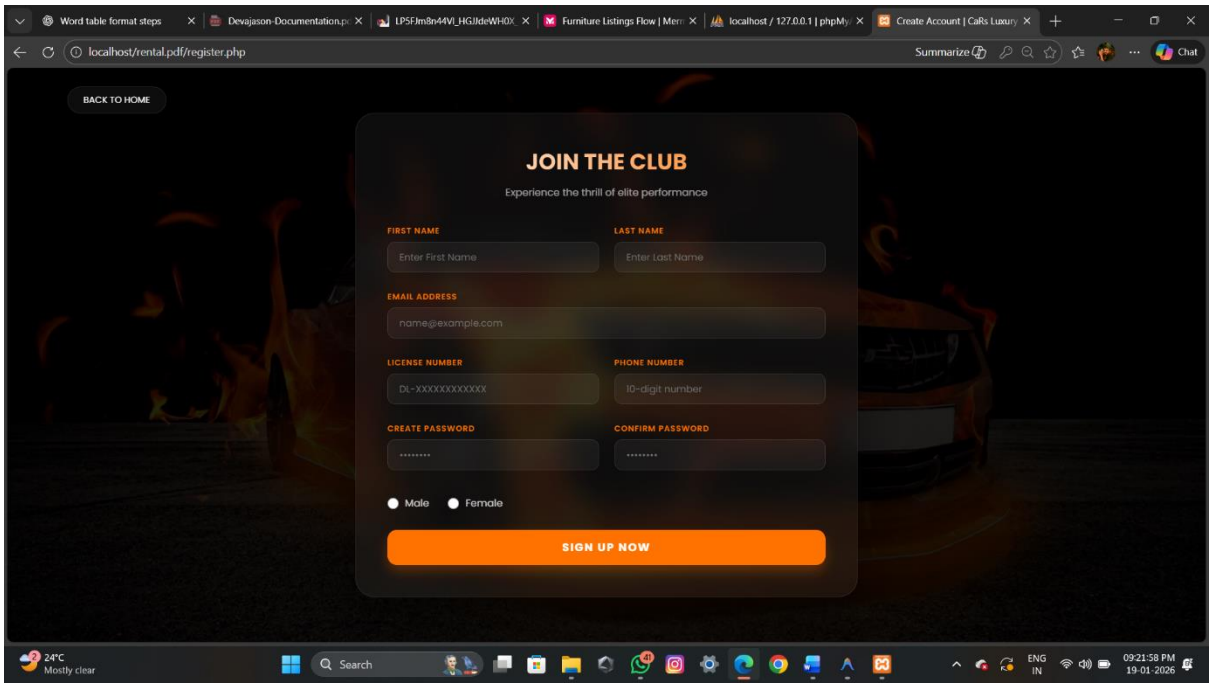
APPENDIX D

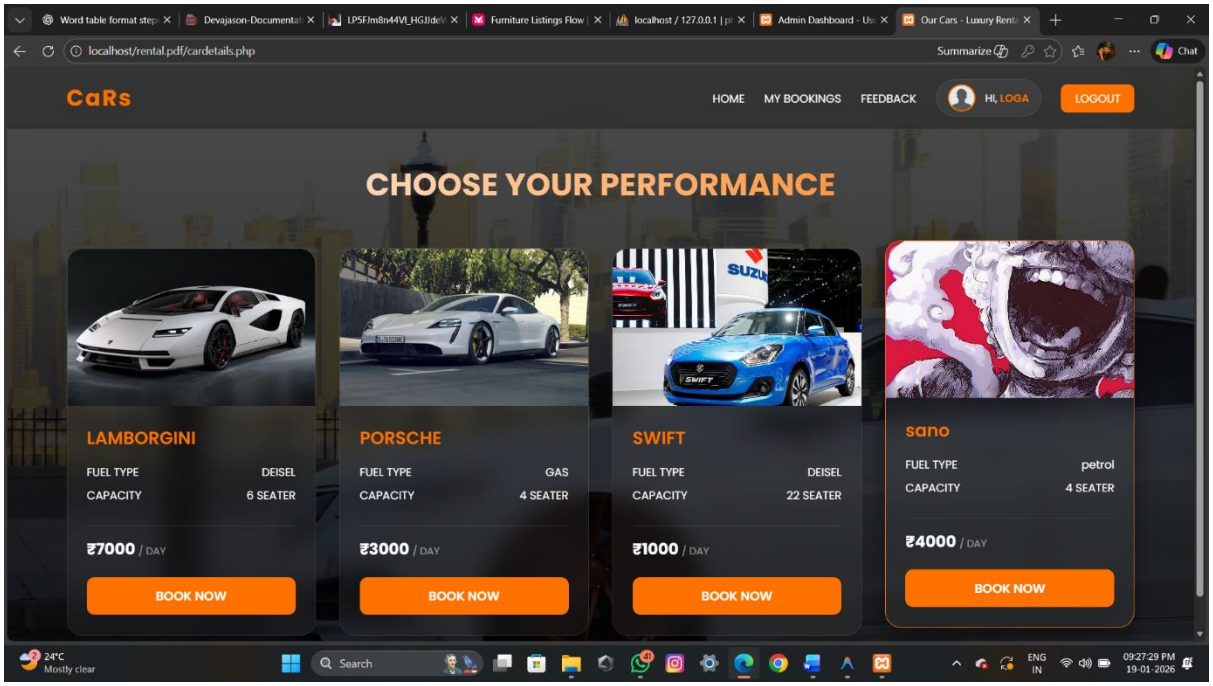
SAMPLE SCREENS

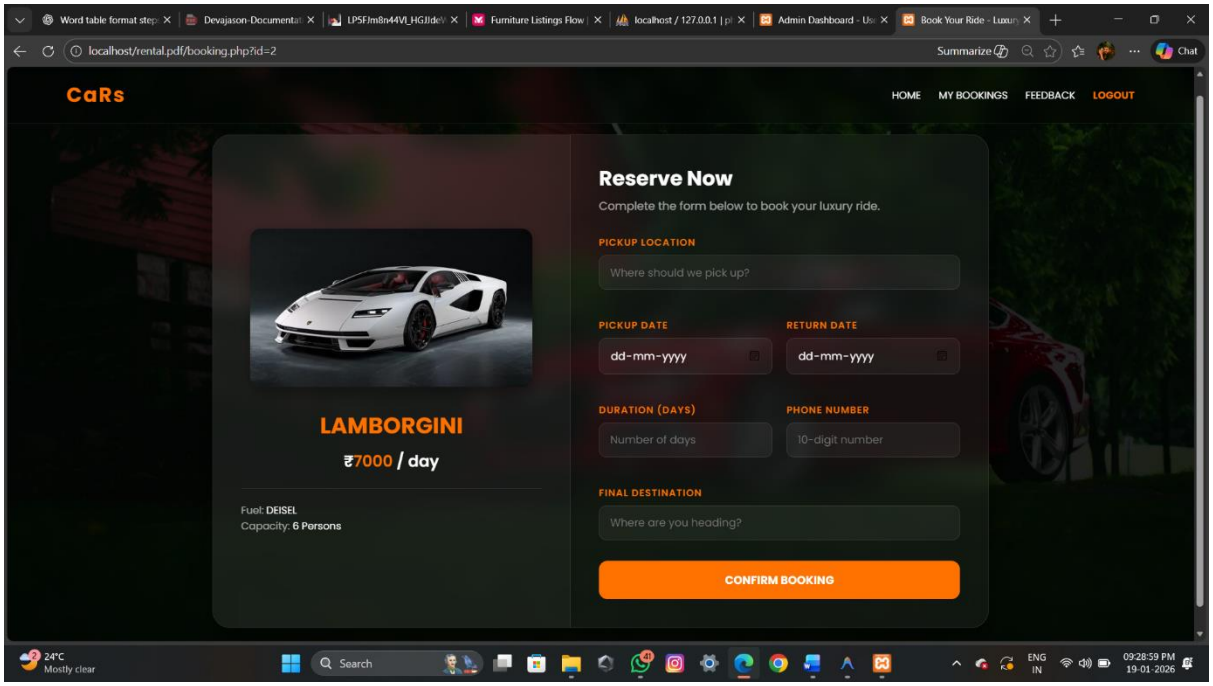
SAMPLE SCREENS

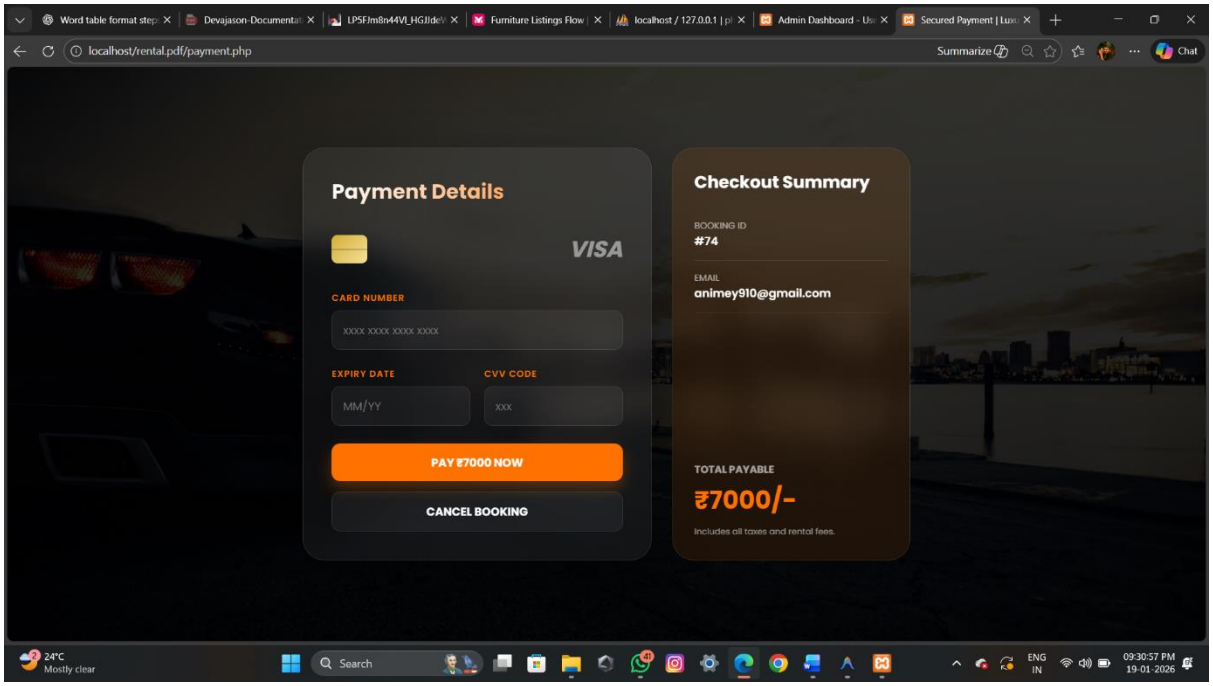
USER PAGE:

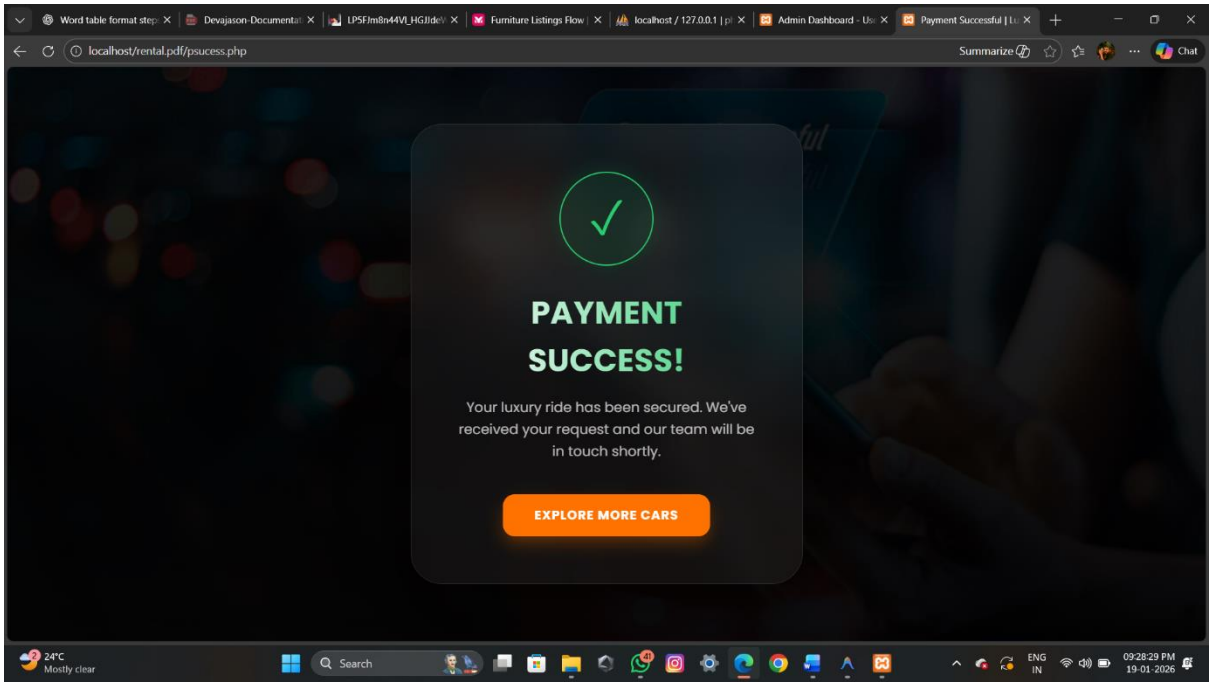


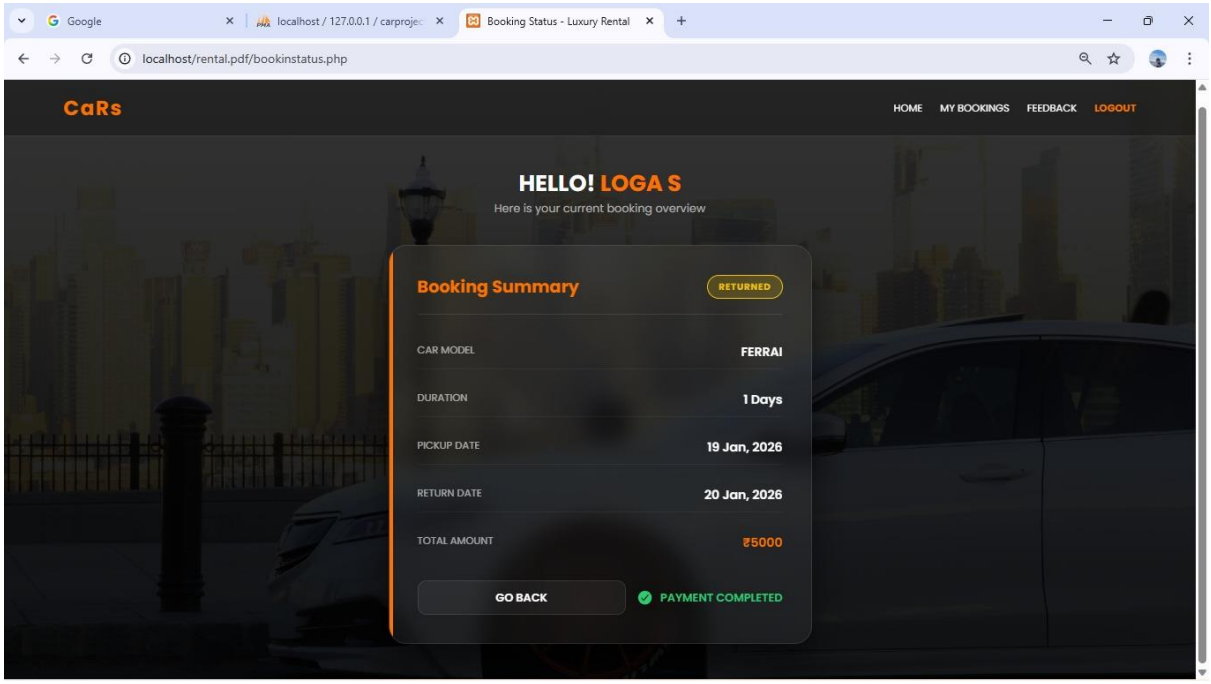


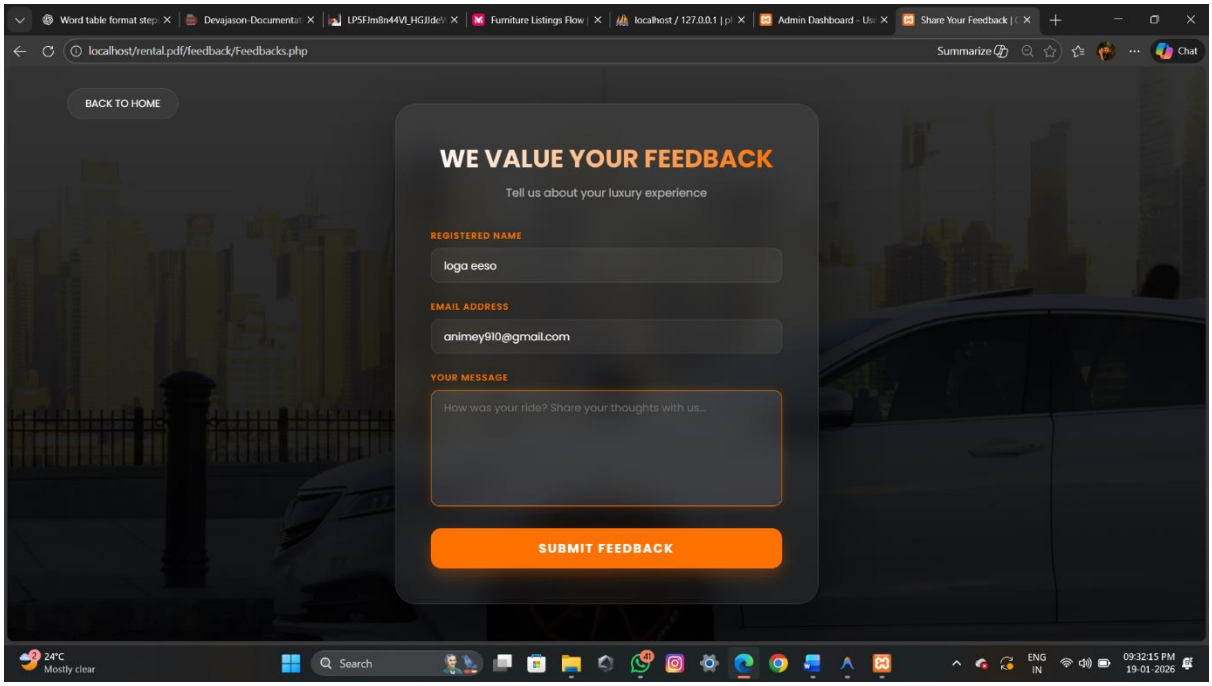

CaRs
HOME
MY BOOKINGS
FEEDBACK
HI, LOGA
LOGOUT
CHOOSE YOUR PERFORMANCE
LAMBORGINI
FUEL TYPE DEISEL
CAPACITY 6 SEATER
₹7000 / DAY
BOOK NOW
PORSCHE
FUEL TYPE GAS
CAPACITY 4 SEATER
₹3000 / DAY
BOOK NOW
SWIFT
FUEL TYPE DEISEL
CAPACITY 22 SEATER
₹1000 / DAY
BOOK NOW
sano
FUEL TYPE petrol
CAPACITY 4 SEATER
₹4000 / DAY
BOOK NOW

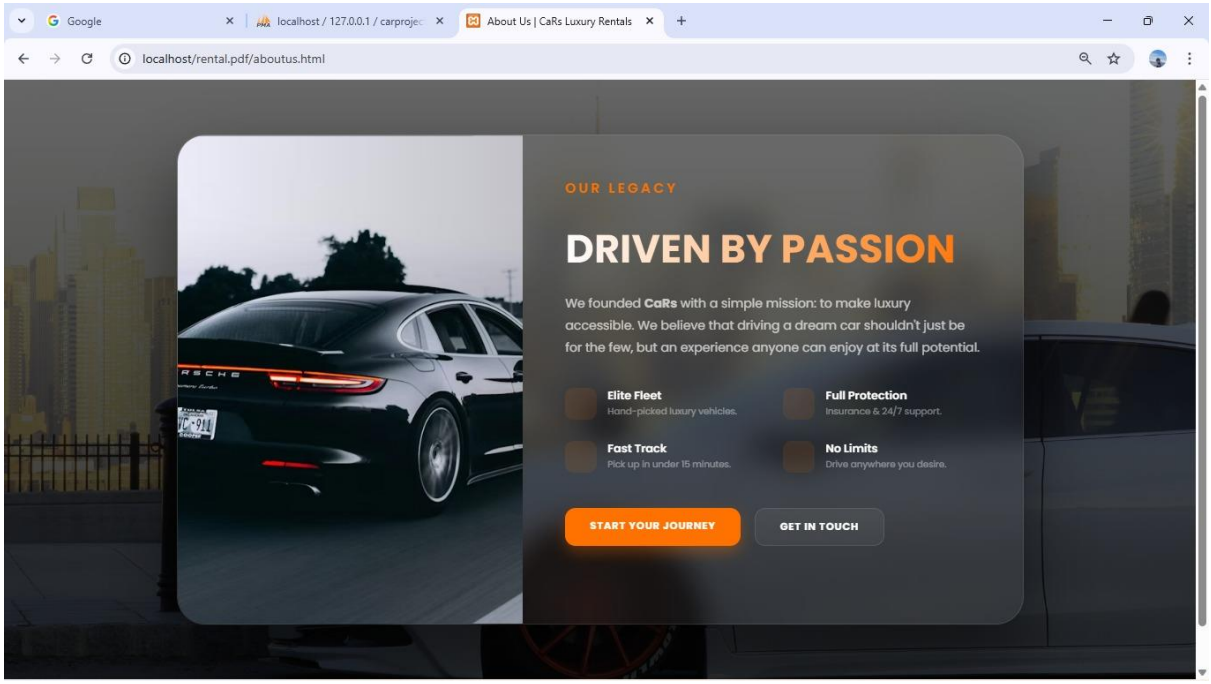

CaRs
HOME
MY BOOKINGS
FEEDBACK
LOGOUT
Reserve Now
Complete the form below to book your luxury ride.
PICKUP LOCATION
Where should we pick up?
PICKUP DATE
dd-mm-yyyy
RETURN DATE
dd-mm-yyyy
DURATION (DAYS)
Number of days
PHONE NUMBER
10-digit number
FINAL DESTINATION
Where are you heading?
CONFIRM BOOKING
LAMBORGINI
₹7000 / day
Fuel: DEISEL
Capacity: 6 Persons

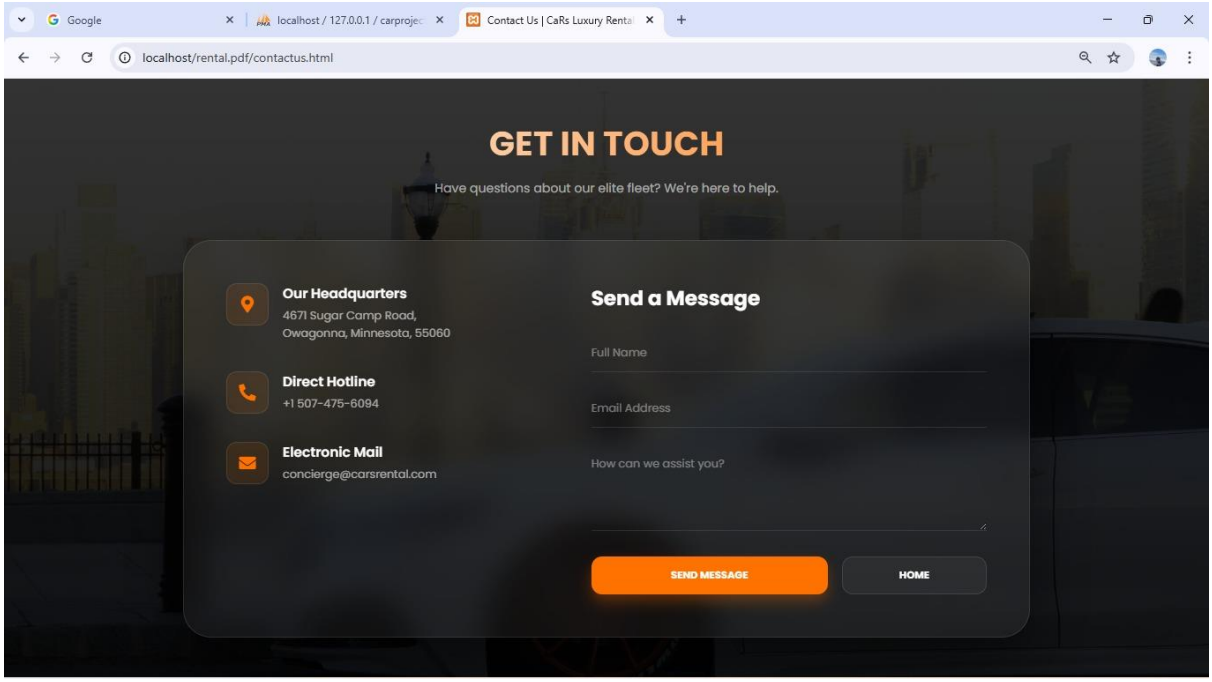

Payment Details
VISA
CARD NUMBER
xxxx xxxx xxxx xxxx
EXPIRY DATE
MM/YY
CVV CODE
xxx
PAY ₹7000 NOW
CANCEL BOOKING
Checkout Summary
BOOKING ID
#74
EMAIL
animey910@gmail.com
TOTAL PAYABLE
₹7000/-
Includes all taxes and rental fees.


PAYMENT SUCCESS!
Your luxury ride has been secured. We've received your request and our team will be in touch shortly.
EXPLORE MORE CARS

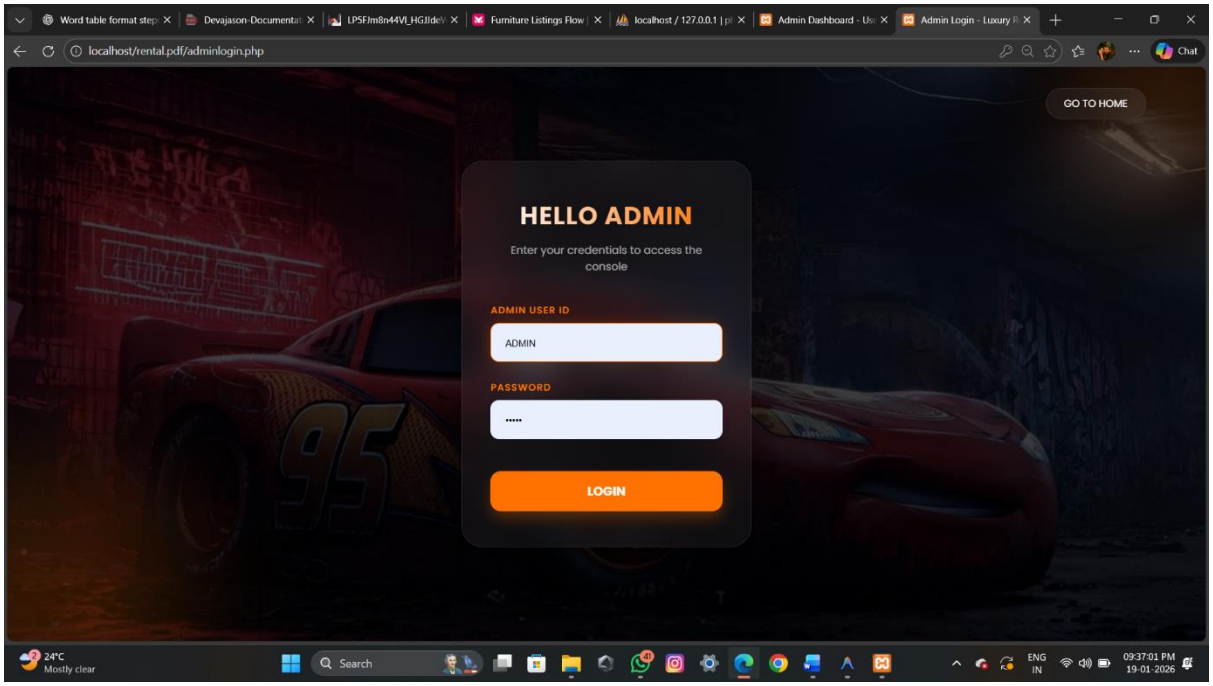

Google
localhost / 127.0.0.1 / carproje
Booking Status - Luxury Rental
localhost/rental.pdf/bookinstatus.php
CaRs
HOME
MY BOOKINGS
FEEDBACK
LOGOUT
HELLO! LOGA S
Here is your current booking overview
Booking Summary
RETURNED
CAR MODEL
FERRAI
DURATION
1 Days
PICKUP DATE
19 Jan, 2026
RETURN DATE
20 Jan, 2026
TOTAL AMOUNT
₹5000
GO BACK
PAYMENT COMPLETED

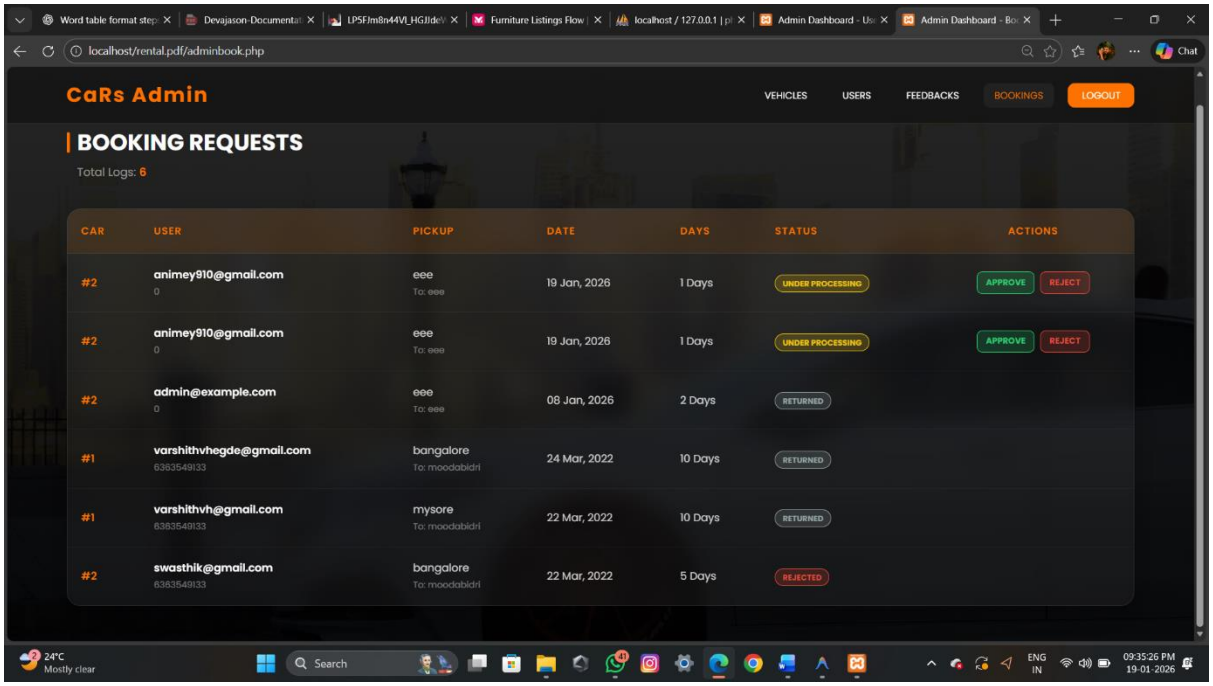

localhost/rental.pdf/feedback/Feedbacks.php
Summarize
Chat
BACK TO HOME
WE VALUE YOUR FEEDBACK
Tell us about your luxury experience
REGISTERED NAME
loga eeso
EMAIL ADDRESS
animey910@gmail.com
YOUR MESSAGE
How was your ride? Share your thoughts with us...
SUBMIT FEEDBACK
24°C
Mostly clear
Search
ENG
IN
09:32:15 PM
19-01-2026

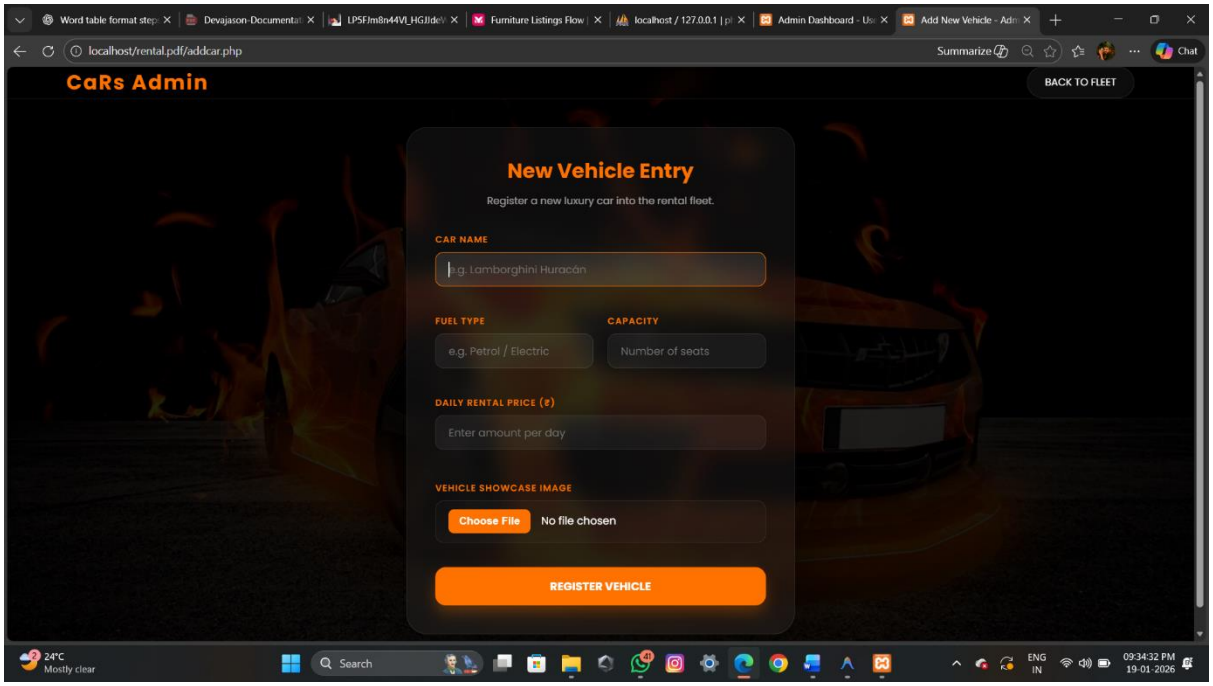

Google
localhost / 127.0.0.1 / carproje
About Us | CaRs Luxury Rentals
localhost/rental.pdf/aboutus.html
OUR LEGACY
DRIVEN BY PASSION
We founded CaRs with a simple mission: to make luxury accessible. We believe that driving a dream car shouldn't just be for the few, but an experience anyone can enjoy at its full potential.
Elite Fleet
Hand-picked luxury vehicles.
Full Protection
Insurance & 24/7 support.
Fast Track
Pick up in under 15 minutes.
No Limits
Drive anywhere you desire.
START YOUR JOURNEY
GET IN TOUCH

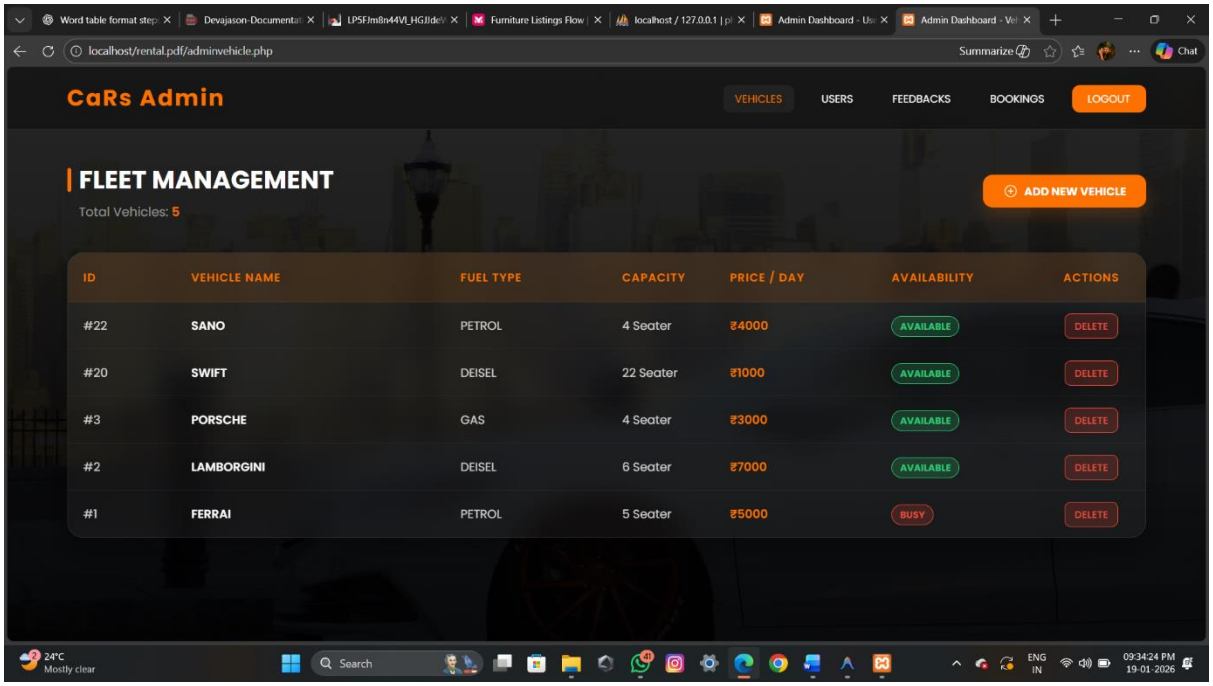

Google
localhost / 127.0.0.1 / carproje
Contact Us | CaRs Luxury Rental
localhost/rental.pdf/contactus.html
GET IN TOUCH
Have questions about our elite fleet? We're here to help.
Our Headquarters
4671 Sugar Camp Road,
Owagonna, Minnesota, 55060
Direct Hotline
+1 507-475-6094
Electronic Mail
concierge@carsrental.com
Send a Message
Full Name
Email Address
How can we assist you?
SEND MESSAGE
HOME

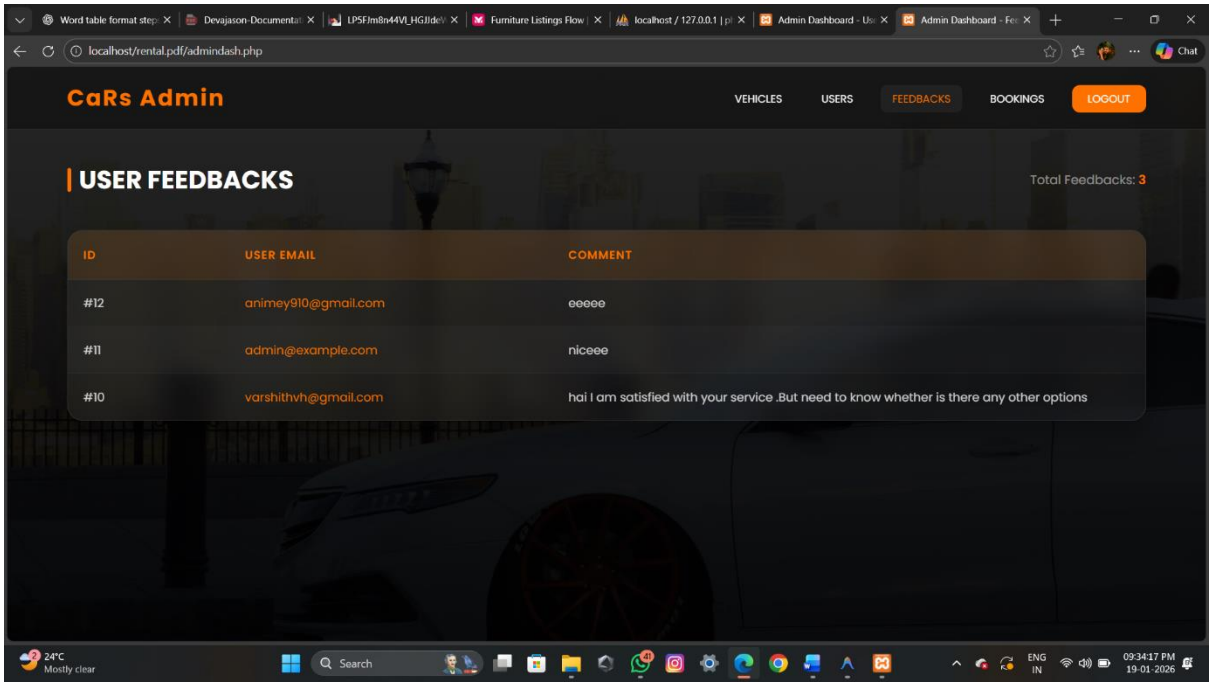
ADMIN PAGE

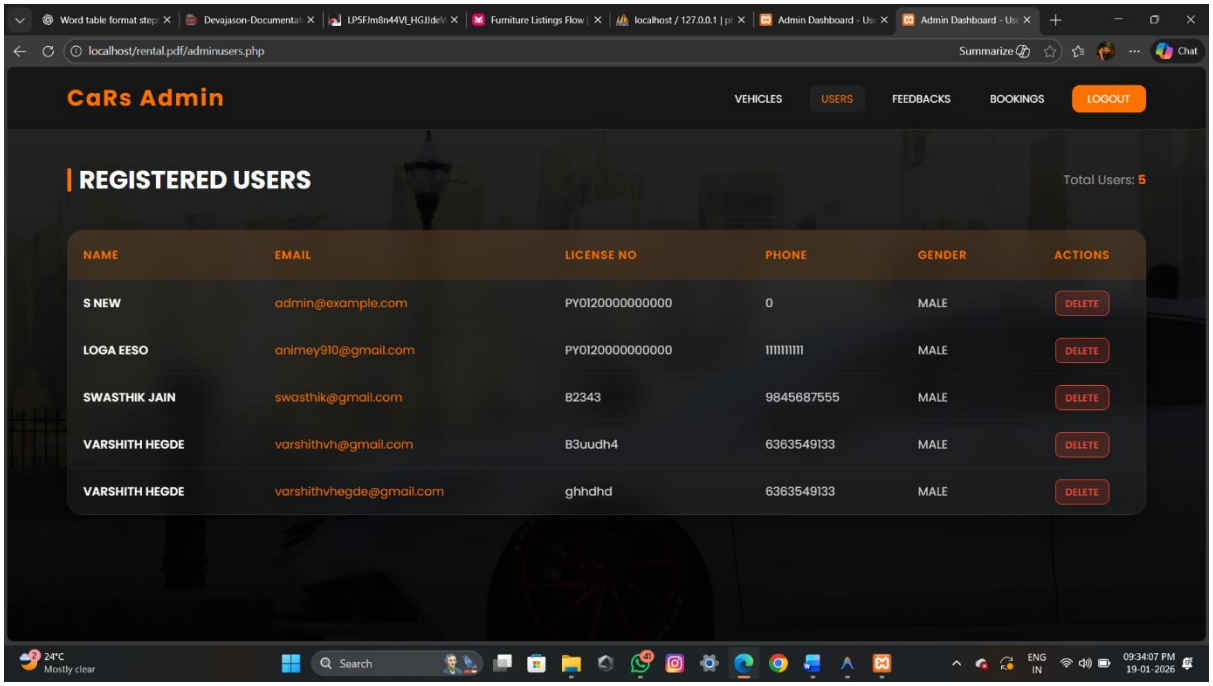

localhost/rental.pdf/adminlogin.php
GO TO HOME
HELLO ADMIN
Enter your credentials to access the console
ADMIN USER ID
ADMIN
PASSWORD
LOGIN


localhost/rental.pdf/adminbook.php
CaRs Admin
VEHICLES
USERS
FEEDBACKS
BOOKINGS
LOGOUT
BOOKING REQUESTS
Total Logs: 6
CAR
USER
PICKUP
DATE
DAYS
STATUS
ACTIONS
#2 animey910@gmail.com eee 19 Jan, 2026 1 Days UNDER PROCESSING APPROVE REJECT
#2 animey910@gmail.com eee 19 Jan, 2026 1 Days UNDER PROCESSING APPROVE REJECT
#2 admin@example.com eee 08 Jan, 2026 2 Days RETURNED
#1 varshithvhegde@gmail.com bangalore 24 Mar, 2022 10 Days RETURNED
#1 varshithvh@gmail.com mysore 22 Mar, 2022 10 Days RETURNED
#2 swasthik@gmail.com bangalore 22 Mar, 2022 5 Days REJECTED


localhost/rental.pdf/addcar.php
CaRs Admin
BACK TO FLEET
New Vehicle Entry
Register a new luxury car into the rental fleet.
CAR NAME
e.g. Lamborghini Huracán
FUEL TYPE
e.g. Petrol / Electric
CAPACITY
Number of seats
DAILY RENTAL PRICE (₹)
Enter amount per day
VEHICLE SHOWCASE IMAGE
Choose File
No file chosen
REGISTER VEHICLE


localhost/rental.pdf/adminvehicle.php
CaRs Admin
VEHICLES
USERS
FEEDBACKS
BOOKINGS
LOGOUT
FLEET MANAGEMENT
Total Vehicles: 5
ADD NEW VEHICLE
ID
VEHICLE NAME
FUEL TYPE
CAPACITY
PRICE / DAY
AVAILABILITY
ACTIONS
#22 SANO PETROL 4 Seater ₹4000 AVAILABLE DELETE
#20 SWIFT DEISEL 22 Seater ₹1000 AVAILABLE DELETE
#3 PORSCHE GAS 4 Seater ₹3000 AVAILABLE DELETE
#2 LAMBORGINI DEISEL 6 Seater ₹7000 AVAILABLE DELETE
#1 FERRAI PETROL 5 Seater ₹5000 BUSY DELETE


localhost/rental.pdf/admindash.php
CaRs Admin
VEHICLES
USERS
FEEDBACKS
BOOKINGS
LOGOUT
USER FEEDBACKS
Total Feedbacks: 3
ID
USER EMAIL
COMMENT
#12
animey910@gmail.com
eeeee
#11
admin@example.com
niceee
#10
varshithvh@gmail.com
hai I am satisfied with your service .But need to know whether is there any other options


localhost/rental.pdf/adminusers.php
CaRs Admin
VEHICLES
USERS
FEEDBACKS
BOOKINGS
LOGOUT
REGISTERED USERS
Total Users: 5
NAME
EMAIL
LICENSE NO
PHONE
GENDER
ACTIONS
S NEW
admin@example.com
PY0120000000000
0
MALE
DELETE
LOGA EESO
animey910@gmail.com
PY0120000000000
11111111
MALE
DELETE
SWASTHIK JAIN
swasthik@gmail.com
B2343
9845687555
MALE
DELETE
VARSHITH HEGDE
varshithvh@gmail.com
B3uudh4
6363549133
MALE
DELETE
VARSHITH HEGDE
varshithvhegde@gmail.com
ghhdhd
6363549133
MALE
DELETE

DATABASE SCREENSHOT

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
booking	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	48.0 KiB	-
cars	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
feedback	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
payment	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
users	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
6 tables	Sum	24	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

Admin

	ADMIN_ID	ADMIN_PASSWORD
Edit Copy Delete	ADMIN	ADMIN

Booking

	BOOK_ID	CAR_ID	EMAIL	BOOK_PLACE	BOOK_DATE	DURATION	PHONE_NUMBER	DESTINATION	RETURN_DATE	PRICE	BOOK_STATUS
Edit Copy Delete	66	2	swasthik@gmail.com	bangalore	2022-03-22	5	6363549133	moodabidri	2022-04-09	35000	REJECTED
Edit Copy Delete	68	1	varshithvh@gmail.com	mysore	2022-03-22	10	6363549133	moodabidri	2022-04-02	50000	RETURNED
Edit Copy Delete	69	1	varshithvhegde@gmail.com	bangalore	2022-03-24	10	6363549133	moodabidri	2022-03-31	50000	RETURNED
Edit Copy Delete	71	2	admin@example.com	eee	2026-01-08	2	0	eee	2026-01-22	14000	RETURNED
Edit Copy Delete	73	2	animey910@gmail.com	eee	2026-01-19	1	0	eee	2026-01-20	7000	UNDER PROCESSING
Edit Copy Delete	74	2	animey910@gmail.com	eee	2026-01-19	1	0	eee	2026-01-20	7000	APPROVED

Cars

	CAR_ID	CAR_NAME	FUEL_TYPE	CAPACITY	PRICE	CAR_IMG	AVAILABLE
Edit Copy Delete	1	FERRAI	PETROL	5	5000	ferrari.jpg	N
Edit Copy Delete	2	LAMBORGINI	DEISEL	6	7000	lamborghini.webp	N
Edit Copy Delete	3	PORSCHE	GAS	4	3000	porsche.jpg	Y
Edit Copy Delete	20	SWIFT	DEISEL	22	1000	IMG-6239c94ea8a4a0.51789849.jpg	Y
Edit Copy Delete	23	Loga	Petrol	4	2500	IMG-696e5635bca3b7.25891596.jpg	Y

Feedback

	FED_ID	EMAIL	COMMENT
Edit Copy Delete	10	varshithvh@gmail.com	hai I am satisfied with your service .But need to ...
Edit Copy Delete	11	admin@example.com	niceee
Edit Copy Delete	12	animey910@gmail.com	eeeee

Payment

	PAY_ID	BOOK_ID	CARD_NO	EXP_DATE	CVV	PRICE
Edit Copy Delete	24	68	4444444444444444	11/22	333	50000
Edit Copy Delete	26	71	1111111111111111	11111	111	14000
Edit Copy Delete	29	73	1111111111111111	11111	111	7000
Edit Copy Delete	30	74	1111111111111111	11111	111	7000

Users

	FNAME	LNAME	EMAIL	LIC_NUM	PHONE_NUMBER	PASSWORD	GENDER
Edit Copy Delete	s	new	admin@example.com	PY0120000000000	0	4cf8b37a5426ed5e2a6010331460bb24	male
Edit Copy Delete	loga	eeso	animey910@gmail.com	PY0120000000000	1111111111	07cf07b4e3c242f06da89de86f85de61	male
Edit Copy Delete	Swasthik	Jain	swasthik@gmail.com	B2343	9845687555	c788b480e4a3c807a14b6f3f4b1a1ae6	male
Edit Copy Delete	Varshith	Hegde	varshithvh@gmail.com	B3uudh4	6363549133	e6235c884414e320c8781c22b0c38c9b	male
Edit Copy Delete	Varshith	hegde	varshithvhegde@gmail.com	ghhdhd	6363549133	e6235c884414e320c8781c22b0c38c9b	male