A PROJECT ON

Car Rental System

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Angadha Pawade,

Nikita Jambhulkar,

Bhushan Patil,

Omkar Potdar

UNDER THE GUIDENCE OF:

Mr. Snehal Jadhav

Faculty Member

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Angadha Pawade,

Nikita Jambhulkar,

Bhushan Patil,

Omkar Potdar

0824 PG-DAC

SIIT Pune

A PROJECT ON

"Car Rental System"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Angadha Pawade,
Nikita Jambhulkar,
Bhushan Patil,
Omkar Potdar

UNDER THE GUIDENCE OF:

Mr. Snehal Jadhav
Faculty Member
Sunbeam Institute of Information Technology, PUNE.

CERTIFICATE

This is to certify that the project work under the title 'Car Rental System is done by Angadha Pawade, Nikita Jambhulkar, Bhushan Patil, Omkar Potdar in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Snehal Jadhav
Project Guide

Date: 10-02-2025

Table of Contents:

1. INTRODUCTION……2

2. REQUIREMENTS……...……3

2.1 Functional Requirements……..3

2.1 Customer Module..……..4

2.2 Admin Module……..11

3. Non-Functional Requirements……..……..12

3.3.1 Hardware and Software Interfaces……...……12

4. DESIGN……..……...13

4.1 Database design……...13

5. CODING STANDARD IMPLEMENTED……19

6. TEST REPORT……...……...22

7. APENDIX A
1. Entity Relationship Diagram…….……27
2. Data Flow Diagram……...28
3. Class Diagram…….……...29

8. APENDIX B
UI Screenshots……...30

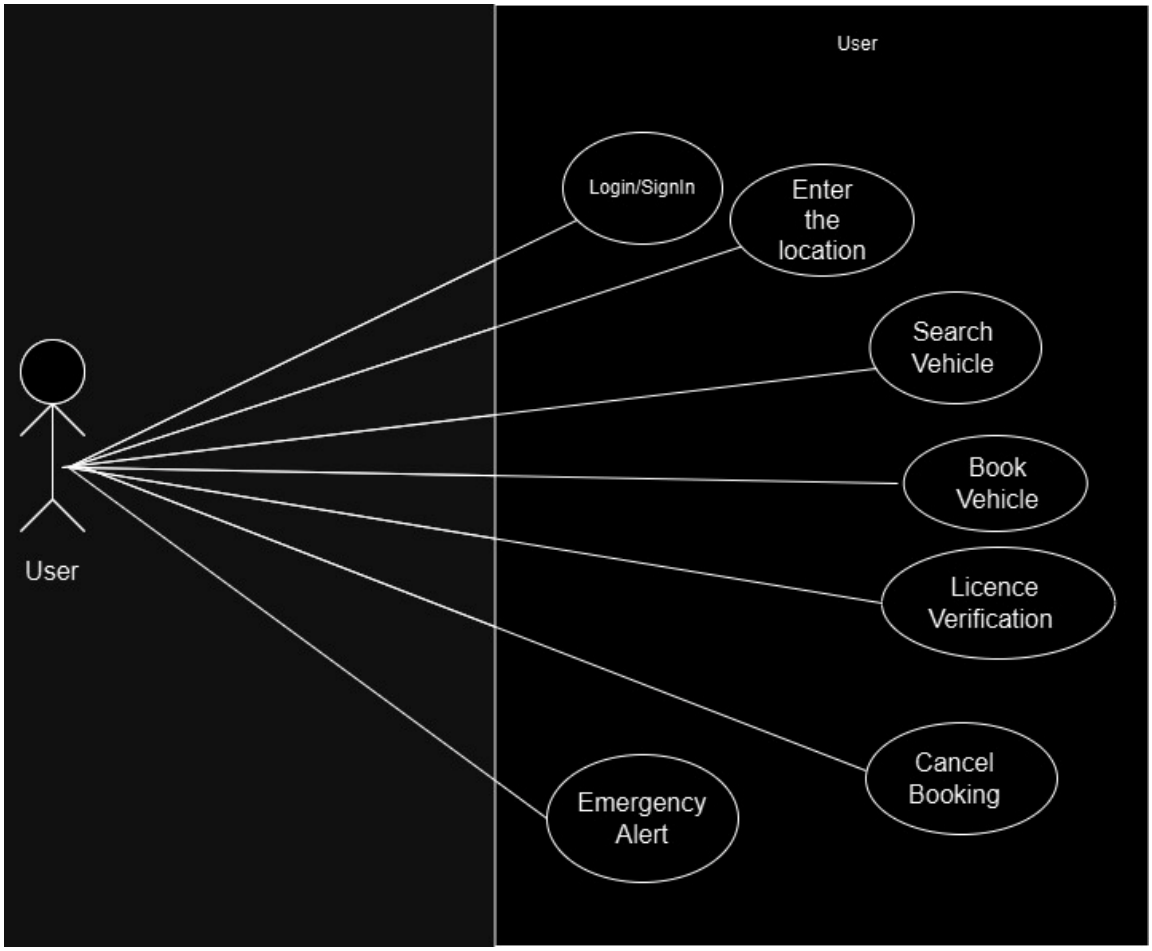
9. REFERENCES……..……..……...41

1.INTRODUCTION TO PROJECT

The Car Rental System is a comprehensive web application designed to facilitate the booking and management of car rentals. The system provides a user-friendly interface for both admins and customers to interact with the platform. Admins can manage the car fleet, booking records, and customer information, while customers can easily browse available cars, make reservations, and view their rental history. This system aims to streamline the overall rental process, offering a seamless experience for all stakeholders.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



2.1.1 Home Page (Common for Customer and Admin)

- Objective: Provide navigation options and overview information.
  Features:
  Clear navigation for customer functionalities such as vehicle search and booking.
  Admin access managed without a UI-based login (admin created using Hibernate).

2.1.2 Customer Flow
Sign In, Login, and Registration (Customer)

- Objective: Provide secure login and registration for customers.
  Features:

1. Sign In:
   User-friendly interface for sign-in with validation.
2. Login:
   Enter email and password to access customer dashboard.
3. Register:
   Form to create a new account with fields for name, email, phone, and password.

2.1.3 Vehicle Selection

- Objective: Browse available vehicles for rent
  Features:
  Filter by vehicle category, brand, and pricing.
  View vehicle details including price per day and specifications.
  Add desired vehicles to the rental cart.

2.1.4 Rental and Payment

- Objective: Manage rental selections and payment.
  Features:

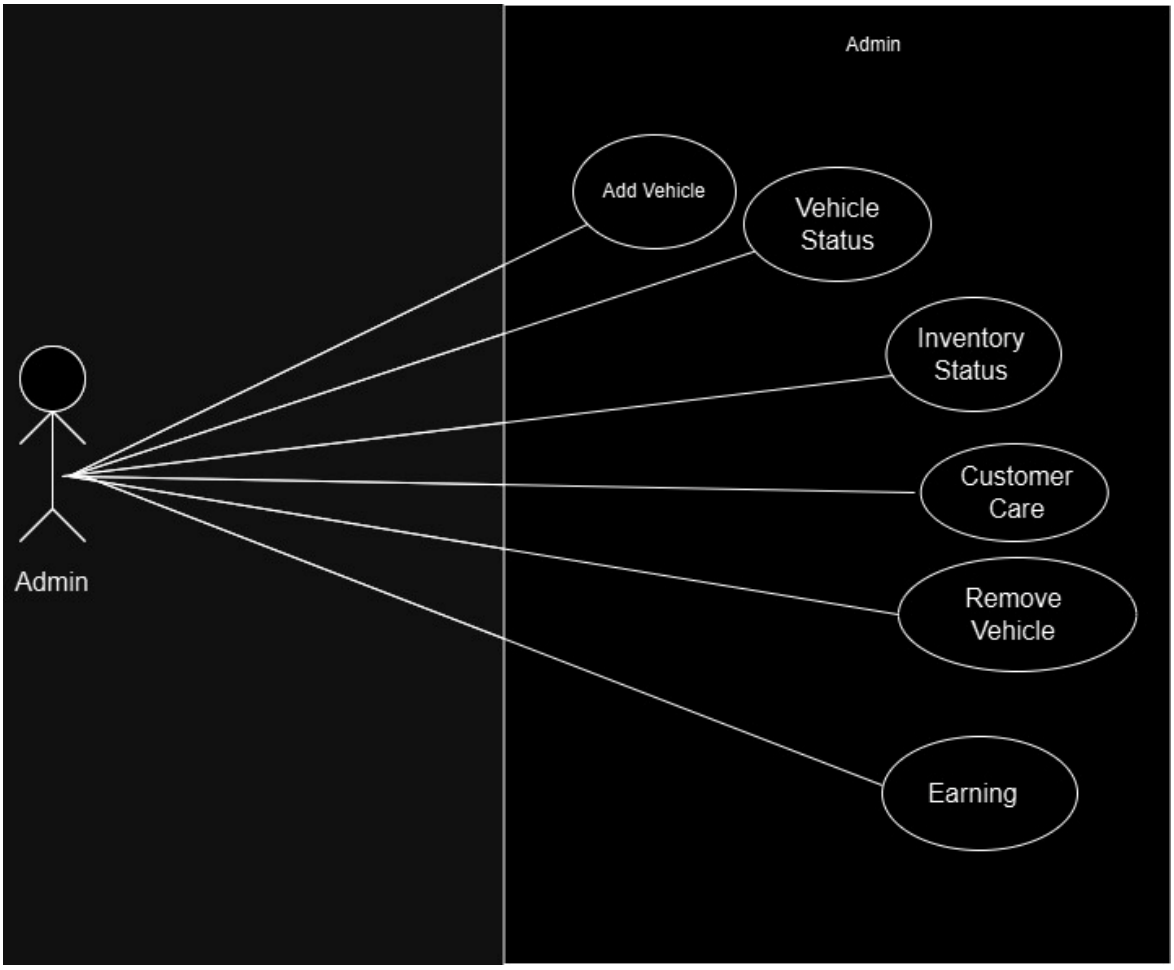
1. Rental Cart:
   View selected vehicles with rental duration and pricing.
   Choose pickup and return locations.
2. Payment:
   Multiple payment options (credit card, UPI, etc.).
   Payment confirmation and receipt generation.

2.1.5 Profile Page

- Objective: Manage user profile and view rental history.

  Features:

  Update personal details.
  View past rentals and payment receipts.
  Submit reviews for rented vehicles.
  Logout option.



2.1.6 Admin Flow

- Objective: Admin account creation through Hibernate without UI-based registration.

  Features:

  Admin details are persisted in the database using Hibernate.
  Admin entity mapped with fields like username, password, and role.
  Hibernate DAO or Repository handles the admin creation logic.

2.1.7 Category Management

- Objective: Manage car categories.

  Features:

  Add new car categories (Sedan, SUV, Hatchback, etc.) through Hibernate operations.
  Update or delete existing categories.

2.1.8 Add Vehicle

- Objective: Add vehicles to the system.

  Features:

  Admin uses Hibernate-based service to enter vehicle details (model, brand, price per day, category, availability).
  Save vehicle information to the database without requiring UI-based forms.

2.1.9 View Vehicle

- Objective: Manage and view the list of available vehicles.

  Features:

  Retrieve vehicle details using Hibernate queries.
  Update or delete vehicle information through service layer operations.

2.1.10 Rental Management

- Objective: Manage active and past rental orders.

  Features:

  Hibernate-based retrieval and update of rental records.
  View customer and vehicle details for each order.
  Update rental statuses directly from the database.

2.1.11 Payment Management

- Objective: Monitor and track rental payments.

  Features:

  View payment history through Hibernate database operations.
  Generate payment reports for rentals.

3. Non-Functional Requirements

3.1 Interface

- User interfaces must be intuitive and user-friendly. Detailed designs are provided in Appendix B.

3.2 Performance

- **Number of Concurrent Users**: The system should handle at least 1000 transactions/inquiries per second.
- **System Resilience**: The application should be resilient to temporary server failures.

3.3 Constraints

- The system should maintain performance standards of handling 1000 transactions/inquiries per second.

3.4 Other Requirements

3.4.1 Hardware Interfaces

Requirements: Intel Core i5 or higher (or AMD equivalent), 8 GB RAM, 512 GB SSD or larger.

3.4.2 Software Interfaces

- **Operating Systems**: MS Windows 11
- **Database**: MySQL.
- **Server**: Embedded Tomcat.
- **Browsers**: Compatible with modern web browsers.

4.System Design

4.1 Architecture

- **Front-End**: Developed using React.js and Redux for state management.
- **Back-End**: Built with Spring Boot for server-side logic.
- **Database**: MySQL for storing user data, orders, and other system information.
- **Server**: Embedded Tomcat for hosting the application.

4.1 Database Design

The following table structures depict the database design.

Table 1 . **CATEGORY**

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
image	longblob	YES		NULL	
name	varchar(255)	YES		NULL	

Table 2 . **FEEDBACKS**

Field	Type	Null	Key	Default	Extra
feedback_id	bigint	NO	PRI	NULL	auto_increment
comment	varchar(255)	YES		NULL	
user_id	bigint	NO	MUL	NULL	

Table 3 . **PAYMENTS**

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
amount	double	YES		NULL	
payment_date	date	YES		NULL	
rental_id	bigint	NO	MUL	NULL	

Table 4 . **RENTAL**

Field	Type	Null	Key	Default	Extra
rental_id	bigint	NO	PRI	NULL	auto_increment
end_date	date	YES		NULL	
start_date	date	YES		NULL	
user_id	bigint	NO	MUL	NULL	
vehicle_id	bigint	NO	MUL	NULL	

Table 5. **USER**

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
contact	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
pincode	varchar(255)	YES		NULL	
role	varchar(255)	YES		NULL	
user_name	varchar(255)	YES		NULL	

Table 6. **VEHICLE**

Field	Type	Null	Key	Default	Extra
vehicle_id	bigint	NO	PRI	NULL	auto_increment
description	varchar(255)	YES		NULL	
price	double	NO		NULL	
product_image	longblob	YES		NULL	
quantity	double	NO		NULL	
vehicle_name	varchar(255)	YES		NULL	
category_id	bigint	YES	MUL	NULL	

5. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	User, Order, UserController	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	SignUp, SignIn, addReviews	Methods should use **verbs** or verb phrases.
Parameter	Camel	firstName, lastName, email, password	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	UserRepository, OrderRepository, MenuRepository	Do not use the '_' sign
Annotation	Pascal	SpringBootApplication	Use @ at start of annotation
DTOs	Camel	ApiResponseDTO, SignUpReqDTO, OrderDetailsResDTO	Use to transfer data between the processes

Exception Class	Pascal with "Exception" suffix	ResourceNotFoundException	

Comments

- Comment each type, each non-public type member, and each region declaration.
- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

6. TEST REPORT

GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	SignUp Page	Signup successfully message	OK	Nothing
2	SignIn Page	Pop-up will come	Ok	Please enter username and password again .
3	HomePage	Car List fetched from db	Ok	Failed to fetched car list
4	Car List page	Gives all car types for selected category	Ok	Nothing
5	Customer Order history	Order history list render successfully	Ok	Failed to fetched orders
7	Add Car Category	Car category added successfully	Ok	Nothing
8	See placed orders by customer	Placed order list view	Ok	No orders is placed
9	Generate Token after first login	Token generated successfully	Ok	Failed to generate token
10	View all users list by admin	Seeing the list of all users	Ok	Nothing
11	Placing Order by customer	Order placed successfully and redirect to home page	Ok	Nothing
12	Logout	It will logout from user profile.	Ok	Nothing

	STATIC TESTING			
SR-NO	**Deviation**	**Program**		
1	Commenting not followed	All Web Application		

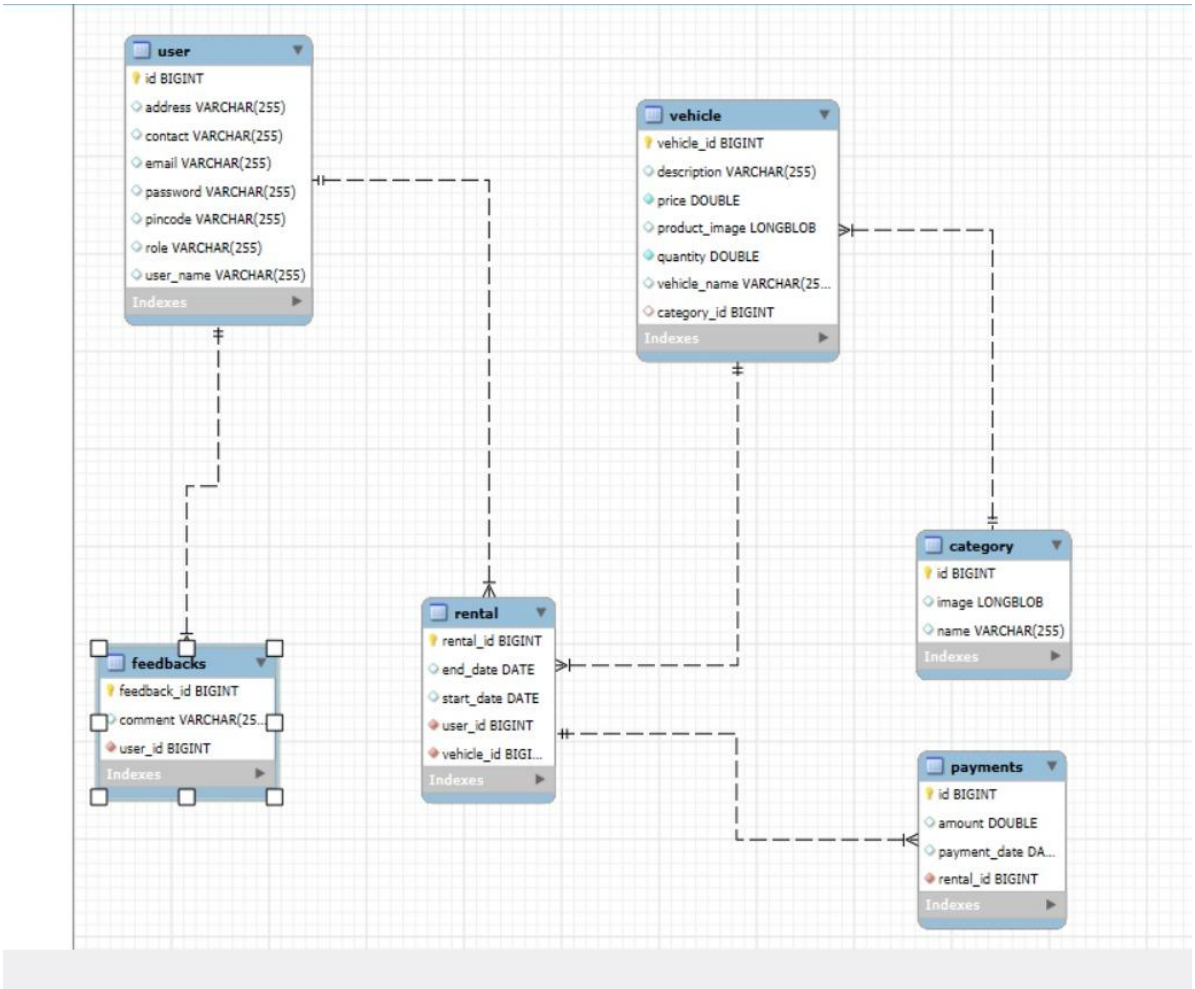
7. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	SLC PHASE	Additional Notes
October 02, 2025	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements.
October 12, 2025	Initial SRS Document Validation and Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better.
October 30, 2025	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram, and User Interfaces	Requirement Analysis & Design Phase	Database Design completed.
Nov 10, 2025	Business Logic Component Design Started	Design Phase	----------
Nov 20, 2025	Coding Phase Started	Coding Phase	70% of Class Library implemented.
Nov 27, 2025	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
Dec 15, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
Dec 25, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	----------

Jan 05, 2025	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
Jan 12, 2025	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	---------- -
Jan 26, 2025	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
Feb 05, 2025	The Errors Found were Removed	Debugging	The Project was complete for submission
Feb 12, 2025	Final Submission of Project	---------- --	---------- -

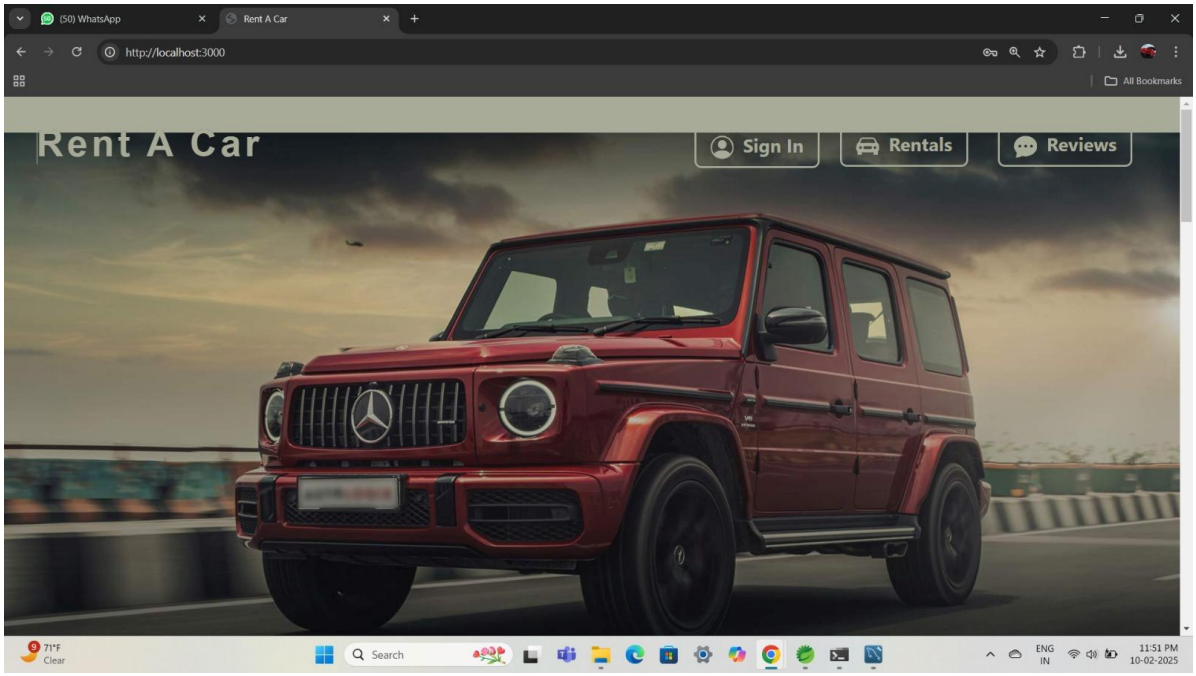
Appendix A

Entity Relationship Diagram

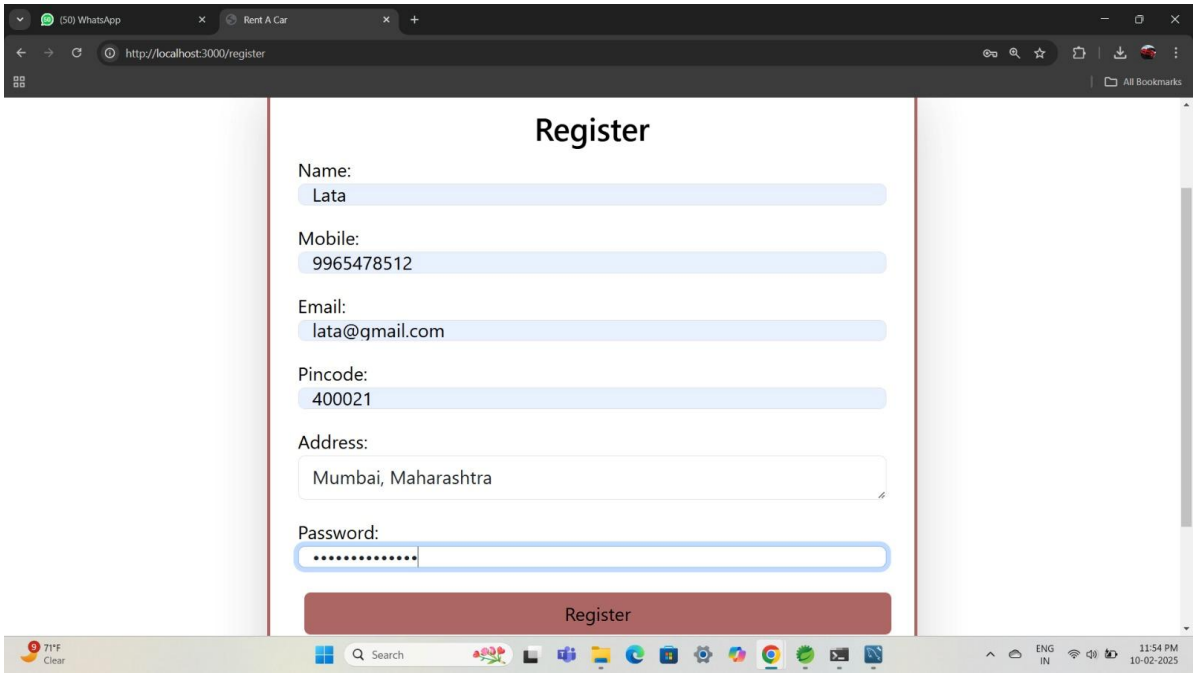


Appendix B

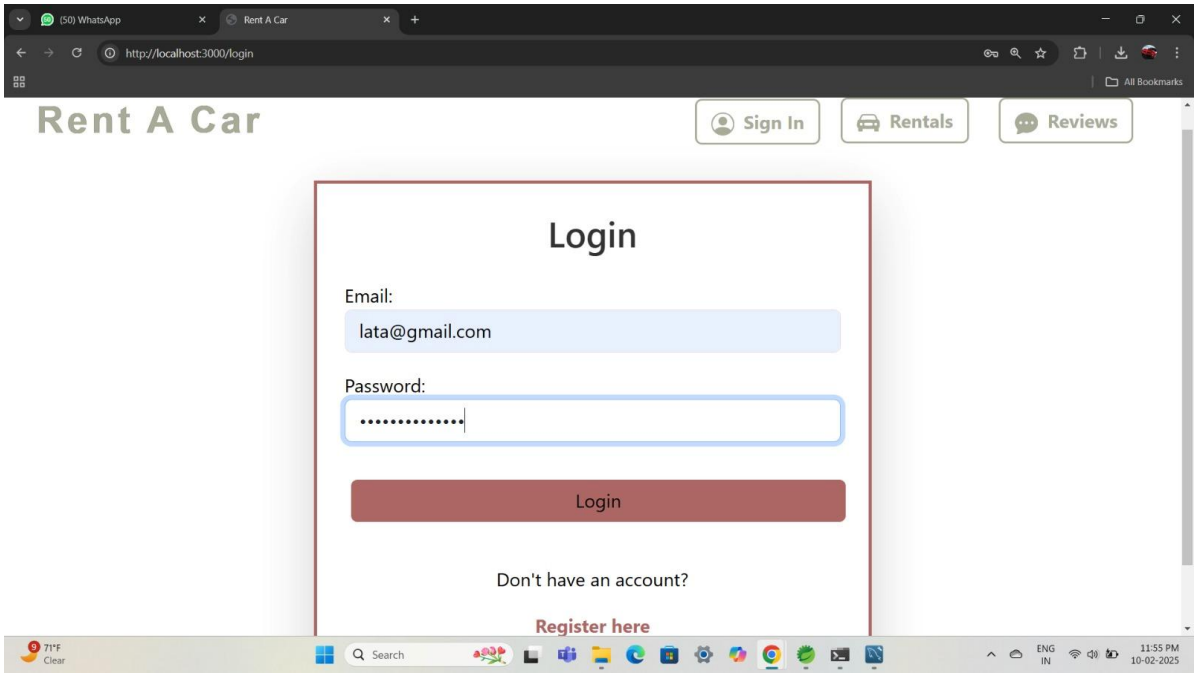
Homepage :



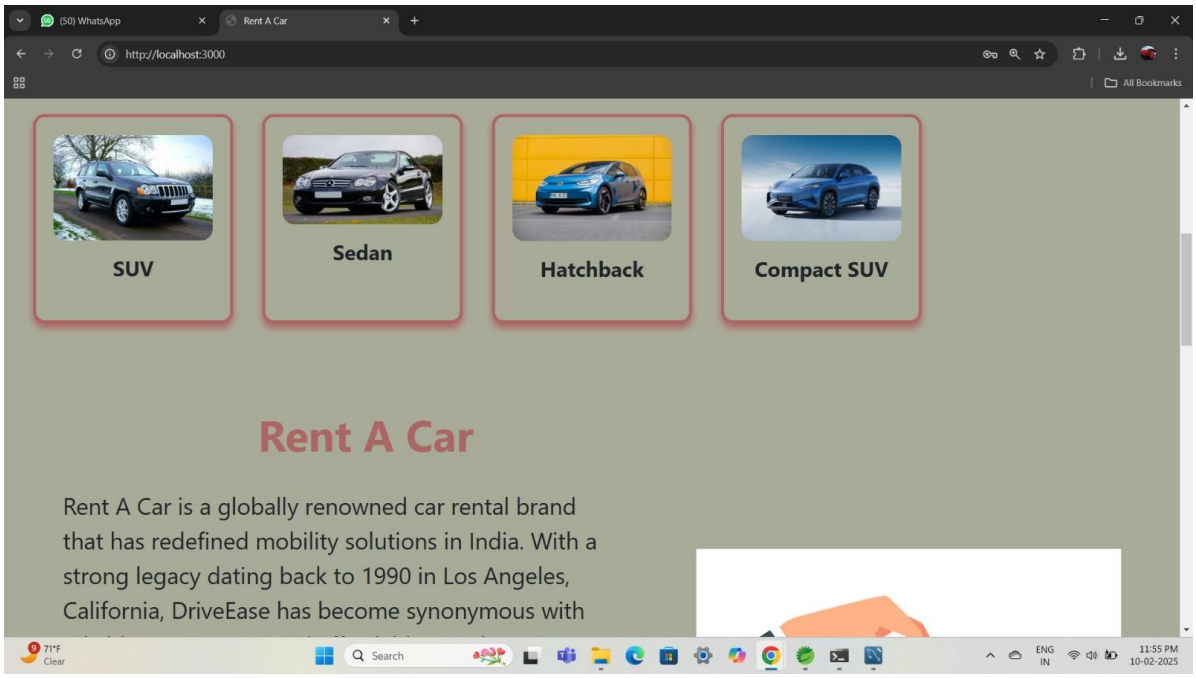
Register:



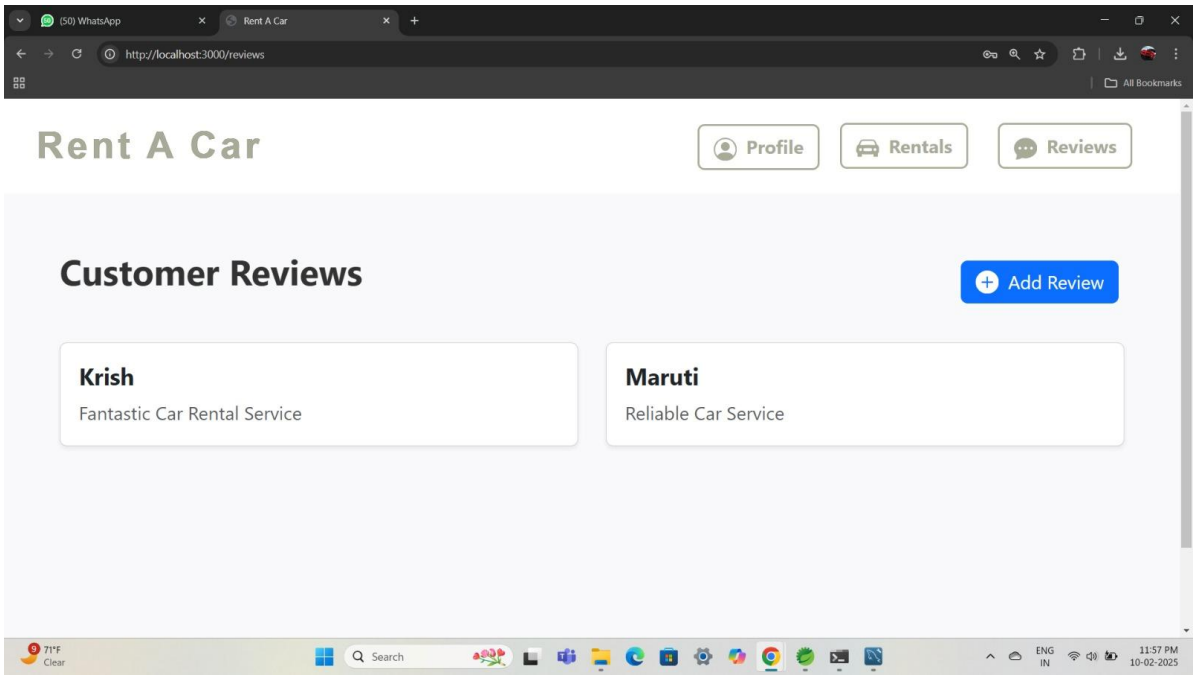
Login :



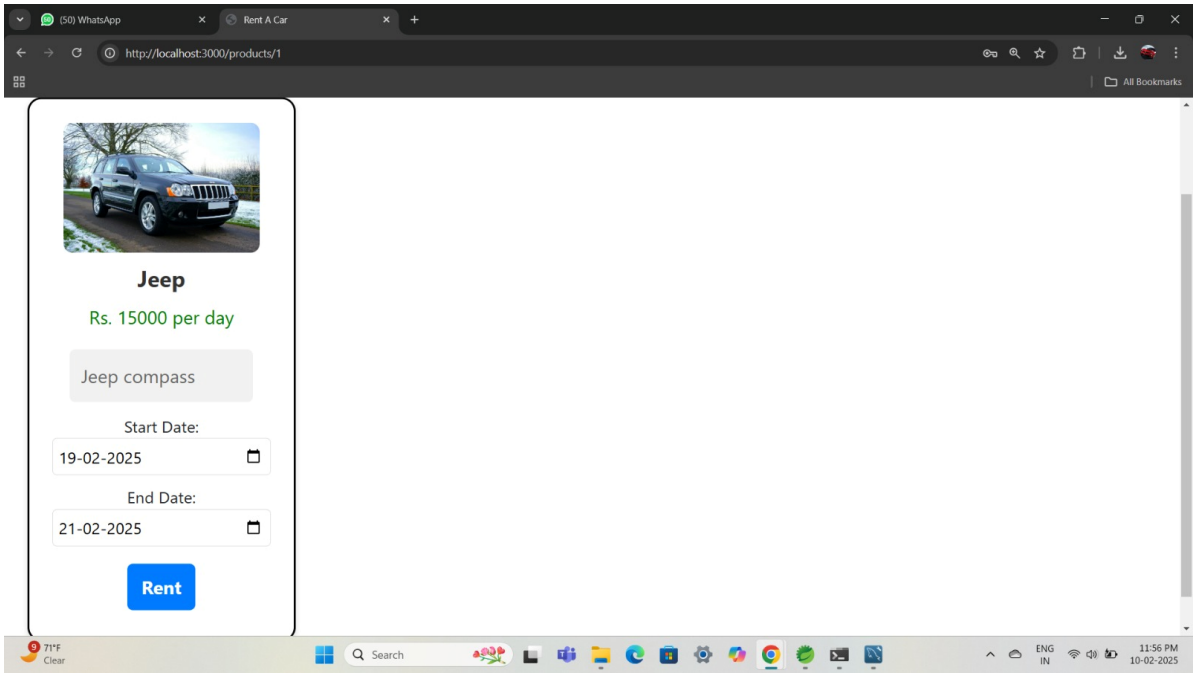
Category

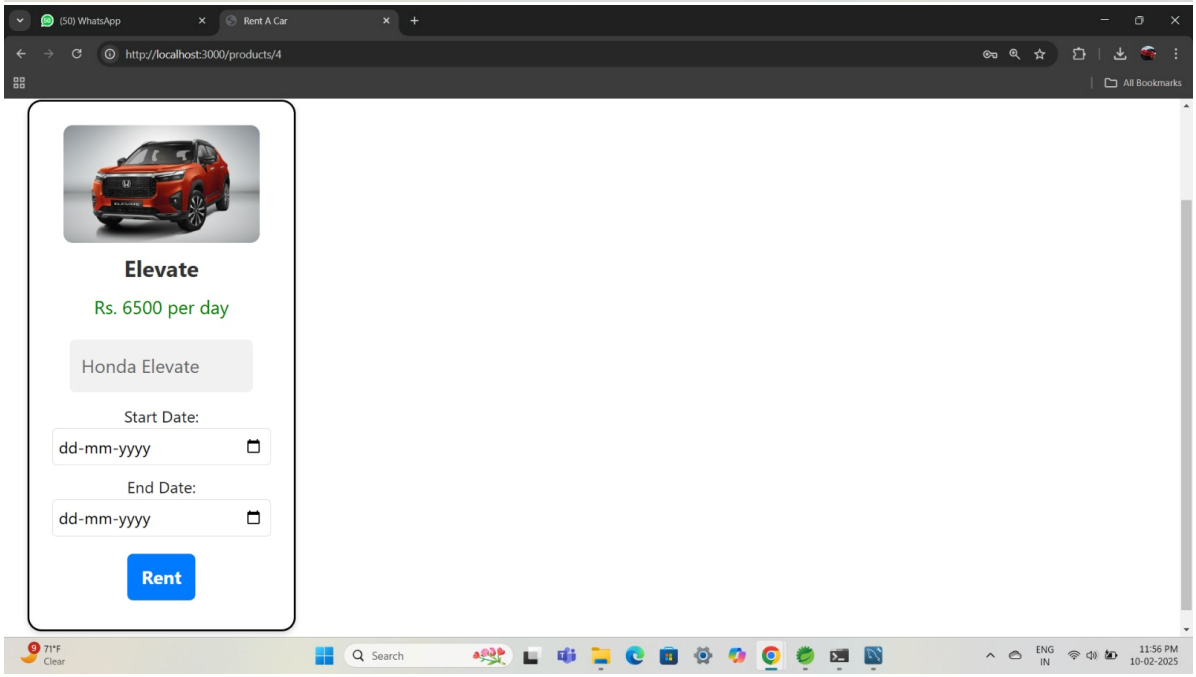


Reviews

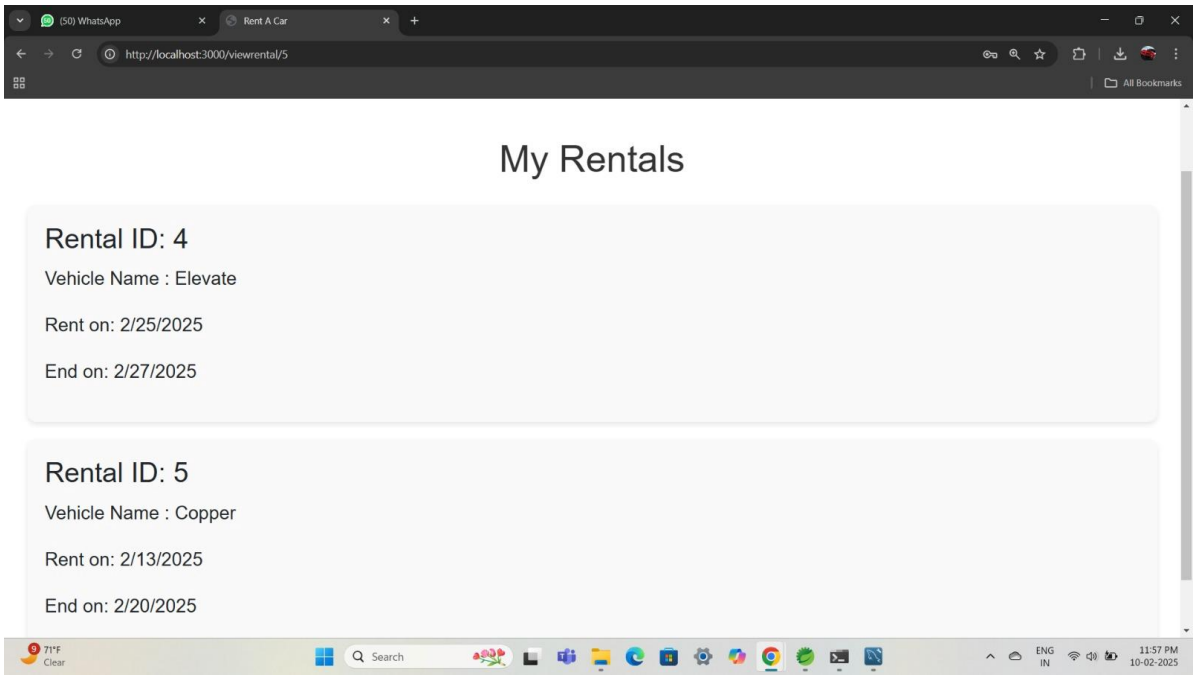

(50) WhatsApp
Rent A Car
http://localhost:3000/reviews
All Bookmarks
Rent A Car
Profile
Rentals
Reviews
Customer Reviews
Add Review
Krish
Fantastic Car Rental Service
Maruti
Reliable Car Service
71°F
Clear
Search
ENG
IN
11:57 PM
10-02-2025

Products


(50) WhatsApp
Rent A Car
http://localhost:3000/products/1
All Bookmarks
Jeep
Rs. 15000 per day
Jeep compass
Start Date:
19-02-2025
End Date:
21-02-2025
Rent
71°F
Clear
Search
ENG
IN
11:56 PM
10-02-2025


(50) WhatsApp
Rent A Car
http://localhost:3000/products/4
All Bookmarks
Elevate
Rs. 6500 per day
Honda Elevate
Start Date:
dd-mm-yyyy
End Date:
dd-mm-yyyy
Rent
71°F
Clear
Search
ENG
IN
11:56 PM
10-02-2025

ViewRentals



7. REFERENCES

1. **Spring Boot Documentation**
   URL: https://spring.io/projects/spring-boot

2. **React.js Documentation**
   URL: https://reactjs.org/docs/getting-started.html

3. **Redux Documentation**
   URL: https://redux.js.org

4. **Java Programming Language**
   URL: https://www.oracle.com/java/

5. **MySQL Workbench Documentation**
   URL: https://dev.mysql.com/doc/workbench/en/

6. **Spring Boot with React and Redux**
   URL: https://www.baeldung.com/spring-boot-react-and-redux

7. **Java Persistence API (JPA) Documentation**
   URL: https://www.eclipse.org/eclipselink/documentation/2.7/

8. **Swagger Documentation for Spring Boot** URL:
   https://springdoc.org/

9. **MDN Web Docs**
   URL: https://developer.mozilla.org/

10. **React Redux Integration Guide**
    URL: https://react-redux.js.org/