DISSERTATION

"VEHICLE RENTAL SYSTEM"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

THE DEGREE OF

BACHELOR OF COMPUTER APPLICATION (BCA)

AT

DEPARTMENT OF COMPUTER SCIENCE, APPLICATION & ANIMATION

ESTD : 1880

ST ALOYSIUS COLLEGE (AUTONOMOUS), MANGALURU

MR CLYDE CHARLES COELHO – 214724

MR REUBEN SERRAO – 214748

MR DANIEL AUSTIN PAIS – 214779

WORK CARRIED OUT AT

JEEVIKA CAR AND BIKE RENTALS

DURING THE ACADEMIC YEAR 2023 – 24

UNDER THE GUIDANCE OF

INTERNAL GUIDE

Ms Prafulla

Assistant Professor, Department of BCA
St Aloysius College (Autonomous)
Mangaluru – 575 003

EXTERNAL GUIDE

Ms Pavithra Shetty

Jeevika Car and Bike Rentals,
Saptagiri Complex, Kavoor,
Mangalore – 575 0015

MAY, 2024

ST ALOYSIUS COLLEGE (AUTONOMOUS), MANGALURU

ESTD : 1880

Department of Computer Science, Application & Animation

CERTIFICATE FOR THE APPROVAL OF THE PROJECT

This is to certify that the following students of VI Semester BCA have satisfactorily completed the project "**VEHICLE RENTAL SYSTEM**" for the **Bachelor of Computer Application (BCA)** prescribed by the College during the academic year 2023 – 24.

Mr CLYDE CHARLES COELHO – 214724

Mr REUBEN SERRAO – 214748

Mr DANIEL AUSTIN PAIS – 214779

PROJECT GUIDE SIGNATURE

Ms Prafulla

Assistant Professor, Department of BCA
St Aloysius College (Autonomous)
Mangaluru – 575 003

HOD'S SIGNATURE

Ms Shilpa Shetty

HOD, Department of BCA
St Aloysius College (Autonomous)
Mangaluru – 575 003

DEAN'S SIGNATURE

Dr Ravindra Swami K
Dean, Department of BCA &Computer Animation
St Aloysius College (Autonomous)
Mangaluru – 575 003

Examiners:

1.

2.

DECLARATION BY STUDENT

We hereby declare that this project work titled "**VEHICLE RENTAL SYSTEM**" has been prepared by us during the academic year 2023 – 24 under the guidance of **Ms Prafulla**, Assistant Professor, Department of Computer Science, Application & Animation, St Aloysius College (Autonomous), Mangaluru submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Computer Application (BCA)** as prescribed by the College.

We also declare that this project is the outcome of our efforts, that it has not been submitted to any other University for the award of any degree or diploma.

Mr CLYDE CHARLES COELHO – 214724 **SIGNATURE**

Mr REUBEN SERRAO – 214748 **SIGNATURE**

Mr DANIEL AUSTIN PAIS – 214779 **SIGNATURE**

COMPANY CERTIFICATE

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to all those who extended their co-operation and support during the course of the project. First and foremost, we thank the Almighty God for his blessings.

We express our gratitude to our Principal of St Aloysius College Rev. Dr. Praveen Martis SJ for giving us this opportunity.

We express our sincere thanks to Dr. K Ravindra Swamy (Dean) department of Computer Science, Applications and Animation.

We express our thanks to Ms Shilpa Shetty (H.O.D) Department of Computer Application and Animation.

We thank our external guide Ms Pavithra Shetty and internal guide Ms. Prafulla who took their time to provide us with necessary information and all the requirements to complete this project.

We would like to thank all the staffs of our department for their support.

We extend our heartfelt gratitude to our parents for their encouragement throughout the completion of the project. We are grateful to our friends and all those individuals who have contributed directly and indirectly towards the accomplishment of this task.

Mr CLYDE CHARLES COELHO – 214724 **SIGNATURE**

Mr REUBEN SERRAO – 214748 **SIGNATURE**

Mr DANIEL AUSTIN PAIS – 214779 **SIGNATURE**

Table of Contents

SYNOPSIS

1.1 Title of the Project

Vehicle Rental System

1.2 Abstract

The **Vehicle Rental System** proposes a comprehensive and user-friendly platform for managing the rental of various types of vehicles. This system aims to manage the entire rental process, enhancing efficiency and convenience for both the rental company and the customers. The Vehicle Rental System manages rentals of two-wheeler and four-wheeler vehicles. The two-wheeler category includes motorcycle and motor scooter, while the four-wheeler category includes different car variants like SUV, Sedan, Hatchback and Mini Van. It makes renting easier for customers and managing their fleet simpler for rental companies.

1.3 Objective of the Project

1. To Register Users.
2. To add and manage vehicle data.
3. To manage the vehicle rentals.
4. To book a vehicle.
5. To manage bookings.
6. To post Feedback.

1.4 Project Category

Web Based Application

1.5 Languages to be used

1.5.1 Front-End: HTML, CSS, JavaScript

1.5.2 Back-End: Php

1.5.3 Database: SQL

1.6 Structure of the Project

In the admin portal the admin can add unlimited list of the vehicles of different categories, manage users, manage vehicle information, manage bookings, and manage payments. The front-end will be designed using HTML, CSS and JavaScript and the back-end will be done using Php. The Database will managed using Sql.

1.6.1 Features:

1. Unlimited number of vehicle can be added.
2. Responsive Page Design.
3. Wide Range of Vehicles.
4. User Feedback Option.
5. Simple User Interface.

1.7 Module Description

1.7.1. User Login and Registration:

1.7.1.1 Registration**:** The user can create a new account by clicking on the register and provide the required details such as name, phone number, email, date of birth and a strong password.

1.7.1.2 Login: The user can login using email and password by clicking on the Login.

1.7.2. Admin Login:

The admin can login through the admin login portal by entering the admin credentials. After the admin is logged-in, he can manage user data, vehicle information and bookings and also view feedback.

1.7.3. User Profile:

The user can add information such as Aadhaar number, Aadhaar card photo, driving license number, driving license photo and also update them in user profile portal.

1.7.4. Vehicle Management:

The admin can add unlimited number of vehicles to the list and add information about the vehicle.

1.7.5. Vehicle Booking:

The user can book the vehicle and set the time period according to the

user's choice. The user can also select the category of the vehicle based on his/her interest.

1.7.6. Manage Bookings: The admin can manage the list of bookings and reserve the vehicles to the customers.

1.7.7. Feedback:

The user can provide feedback regarding their experience using the vehicle and company's service .

1.8 Any other Information

1.8.1 Hardware Requirements:

- Processor: intel CORE or AMD Ryzen
- RAM: 4GB or above
- Storage: 256GB SSD or higher

1.8.2 Software Requirements:

- Language: Php
- Web Components: HTML, CSS
- Scripting Language: JavaScript
- Operating System: Windows
- Database: SQL
- Browser: Chrome, Brave, Edge

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

2.1 Introduction

The Software Requirement Specification (SRS) provides an overview of the entire system with scope definitions, references. The document aims to gather, analyze and give an in- depth insight into the complete system. A requirements document defines what is desired from the product. It states the product's purpose and what it must achieve. An SRS minimizes the time and efforts required by the developers to achieve desired goals and also minimizes the development cost.

2.1.1 Purpose:

The main purpose of this document is to provide a detailed specification of the Vehicle Renting System. This system is designed to facilitate the rental process for vehicles, allowing users to search, book, and manage vehicle rentals. It provides customers a better user experience in renting the vehicles. It will help in managing the vehicle rentals and reduce the time and paper work.

2.1.2 Scope:

The vehicle rental system will cover the entire vehicle rental process, including user registration, vehicle selection, booking, payment processing, and administrative functions for managing the fleet and user accounts. It will provide a user friendly interface for all the users.

2.1.3 Definitions, Acronyms, Abbreviations

- **SRS:** Software Requirement Specification
- **HTML:** Hyper Text Markup Language
- **CSS:** Cascading Style Sheet
- **JS:** Java Script
- **SQL:** Structured Query Language
- **PHP:** Hypertext Pre Processor
- **DFD:** Data Flow Diagram

2.1.4 Overview:

The following subsections provides the complete overview of the Software requirements and Specifications (SRS) for Vehicle Rental System.

The SRS is documented in the view of admin and the user and the subsections are arranged to give a complete outlook of the software, its perspective, features and system requirements.

2.2 Overall Description

2.2.1 Admins Perspective:

This system aims at providing an application that enables a renter to list their vehicles that they want to rent. This system builds a platform for the renter to list their vehicles which the user can browse. The user after registering and submitting the necessary data can select a vehicle of their choice and book it for their period of choice. The user can post reviews or add complaints about a particular vehicle. It provides good and easy graphical user interface to both new, naïve as well as experienced users of the computers

2.2.1.1 Admin Functions:

1. The admin log's in to the system as the Admin by entering the valid username and password.
2. The admin then can add or manage the vehicles.
3. The admin is able to manage user details.
4. The admin has an authority to edit his profile.

2.2.2 Users Perspective:

The user can register by creating an account, browse the listed vehicles and book a vehicle of their choice after submitting the required data.

2.2.2.1 User Functions:

1. The User can browse the vehicles.
2. The user can register to the website after creating an account.
3. The user can book the vehicles after providing the required data.
4. The user can give feedback on a vehicle.

2.3 User Characteristics

The user of this system is supposed to be fairly educated about the usage of the computers and also about the legal predicaments over renting a vehicle. They should understand the process of sending their data to a third-party and must be of legal age of 18 or above. They must know about online transactions process and be responsible. A person who has no knowledge about this will find it difficult to understand the system. But with a little Knowledge it will be very easy to handle the project.

2.4 Specific Requirements

2.4.1 External Interface Requirements:

It Specifies all the interfaces of the system to the users, hardware and other systems.

2.4.1.1 User Interface:

The system will provide a graphical interface with a user friendly experience for all the users.

2.4.1.2 Hardware Interface:

The system should have these hardware requirements:

1. The processor must be at least intel CORE or AMD Ryzen.
2. The RAM should be or greater than 4GB.

2.4.1.3 Software Interface:

The browser should support HTML/HTML 5 compatible for satisfactory user experience and should have a stable internet connection.

2.4.1.4 Communication Interface:

In this system, the communication between the various modules is done through emails and notifications. The web server and web browser use several communication protocols such as HTTPS, TCP/IP, HTTP.

2.5 Functional Requirements

2.5.1 Registration:

1. Function: The Registration portal will have a form where the user has to fill the necessary details like name , email, contact and a strong password.

2. Input: Name, Email, Contact and Password.

3. Output: The user will get authorization to login using email and password.

2.5.2 Admin Login:

1. Function: In Admin login module the admin can login through admin login portal**.**

2. Input: Username and Password of Administrator.

3. Output: The admin will be logged in to the admin portal where they can manage the website.

2.5.3 User Profile:

1. Function: In this module the user will be able to manage his profile and enter the additional details.

2. Input: User Name, Date of Birth, Aadhar Number, Driving license Number, Aadhar card photo, license photo, Address.

3. Output: The details will be stored in the database and the user can modify the details whenever the user wants.

2.5.4 Booking:

1. Function: In this module the user can book the vehicle by entering the ride details and the time duration. The admin will have the access whether to confirm or deny the booking.

2. Input: Time Period, Vehicle Brand, Vehicle Model, Booking id.

3. Output: The booking request will be sent to admin and he can choose whether to confirm or deny the booking.

2.5.5 Manage Vehicle:

1. Function: In this module the admin can add, delete or edit the vehicle details.

2. Input: Vehicle Category, Vehicle Brand, Vehicle Model, Vehicle Photo, Renting Price.

3. Output: The vehicle details will be stored in the database and will be displayed in the website.

2.5.6 Feedback:

1. Function: The customer can give feedback on their experience using the vehicle.

2. Input: User Name, Email, Contact, feedback.

3. Output: The admin can view the feedbacks.

2.6 Performance Requirements

1. Response Time: The system should respond to user interactions within the specified time limit.
2. Scalability: The system should handle a scalable number of users and transactions.

2.7 Design Constraints

1. It needs all inputs to be filled and it should be valid inputs to generate accurate results.
2. Data should not become corrupted in case of a system crash or power failure.
3. Only the customers who fill in the registration details will be given the privilege to access the features of this website.

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

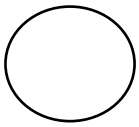
Design involves human intelligence and is a creative activity that cannot be automated. The design activity is a crucial element of a procedure that gradually changes the system requirements into the finished product through a number of intermediary steps. Software design is characterized as the division of a system into functional modules and the description of the functions and interrelationships of each module. The software architecture is this description. We can think of design as a process in which the architecture is described in steps of increasing detail, each of which implements the requirements identified in the previous step. The final step is implementation, which completes the transformation of the software architecture into programs.

The modularity principle is of utmost importance in the design of the software, which is why the components of the system identified during the design activity are referred to as modules. In other words, the software design technique emphasies breaking down the functionality of a program into separate interchangeable modules. The relationships between the subsystems are then established, and the designers agree on the expected behaviours of each subsystem. Each system is then individually analyzed, and the process is repeated until each component is sufficiently sophisticated that a single person can easily execute it.

This document's objective is to provide a comprehensive description of the software requirements using the IEEE-compliant system design document format. It can also serve as a user manual for the software.

3.2 Data Flow Diagrams (DFD)

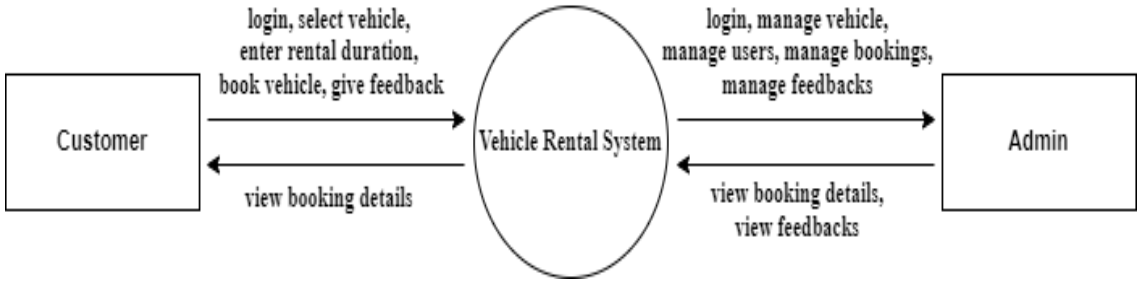
DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analyzing existing as well as proposed system.

Diagram	Name	Description
	Entity	Represents source or destination of the data
	Process	Represents a process that transforms data inputs into data outputs.
	Data Flow	Represents the flow of data into or out of a process or data store.
	Data Store	Represents the data stored.

A Data Flow Diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. Its name indicates its focus is on the flow of information, where it goes and how it gets stored. DFD provide critical insights into the systems and way the information passes through it. DFD helps structure every element of the system, keep them logically intact and interconnected. On the other hand, you have the customers who need to know what is going on in a digestible easy to follow manner.

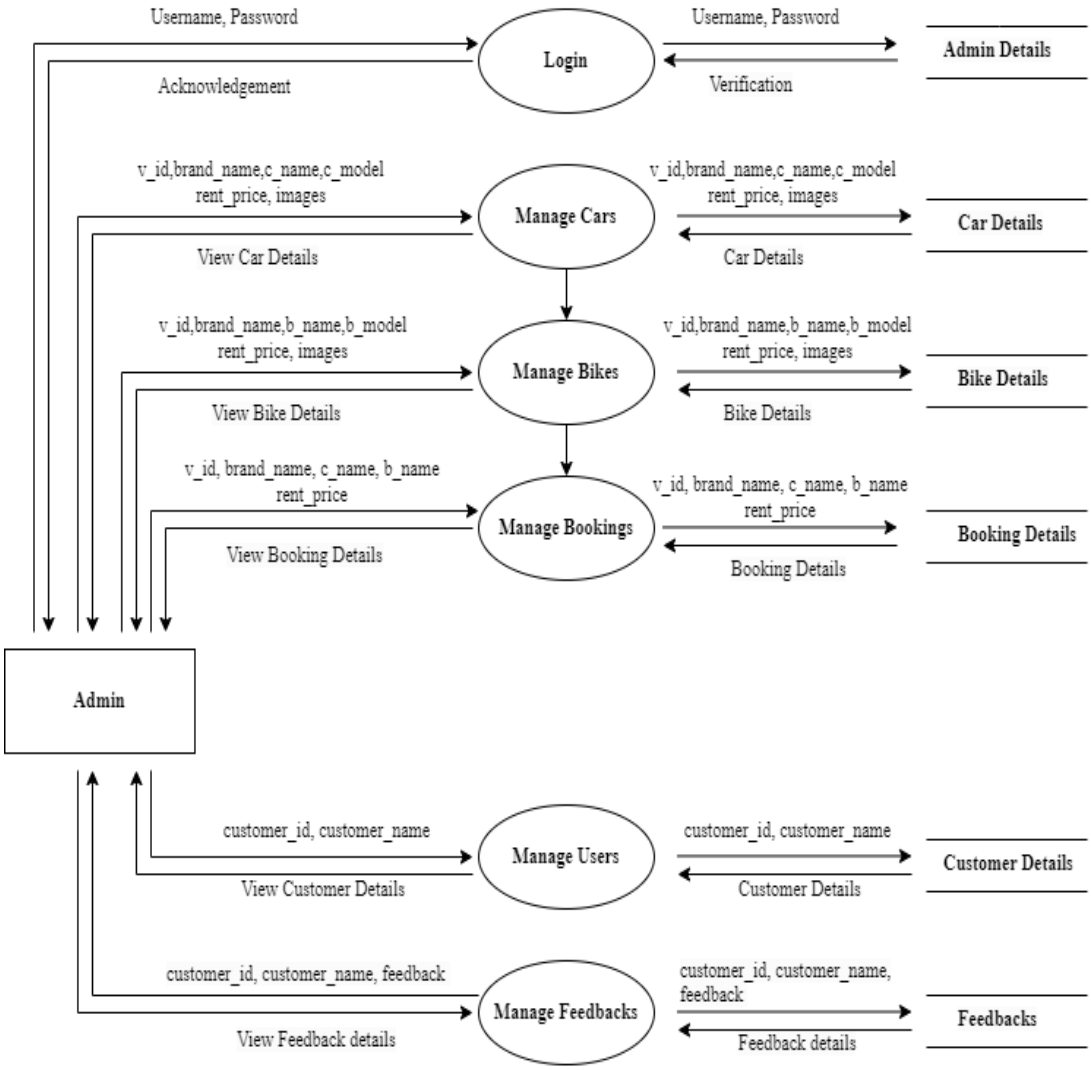
The DFD may be used to perform a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond.

3.2.1 Level 0 DFD (Context Flow Diagram):

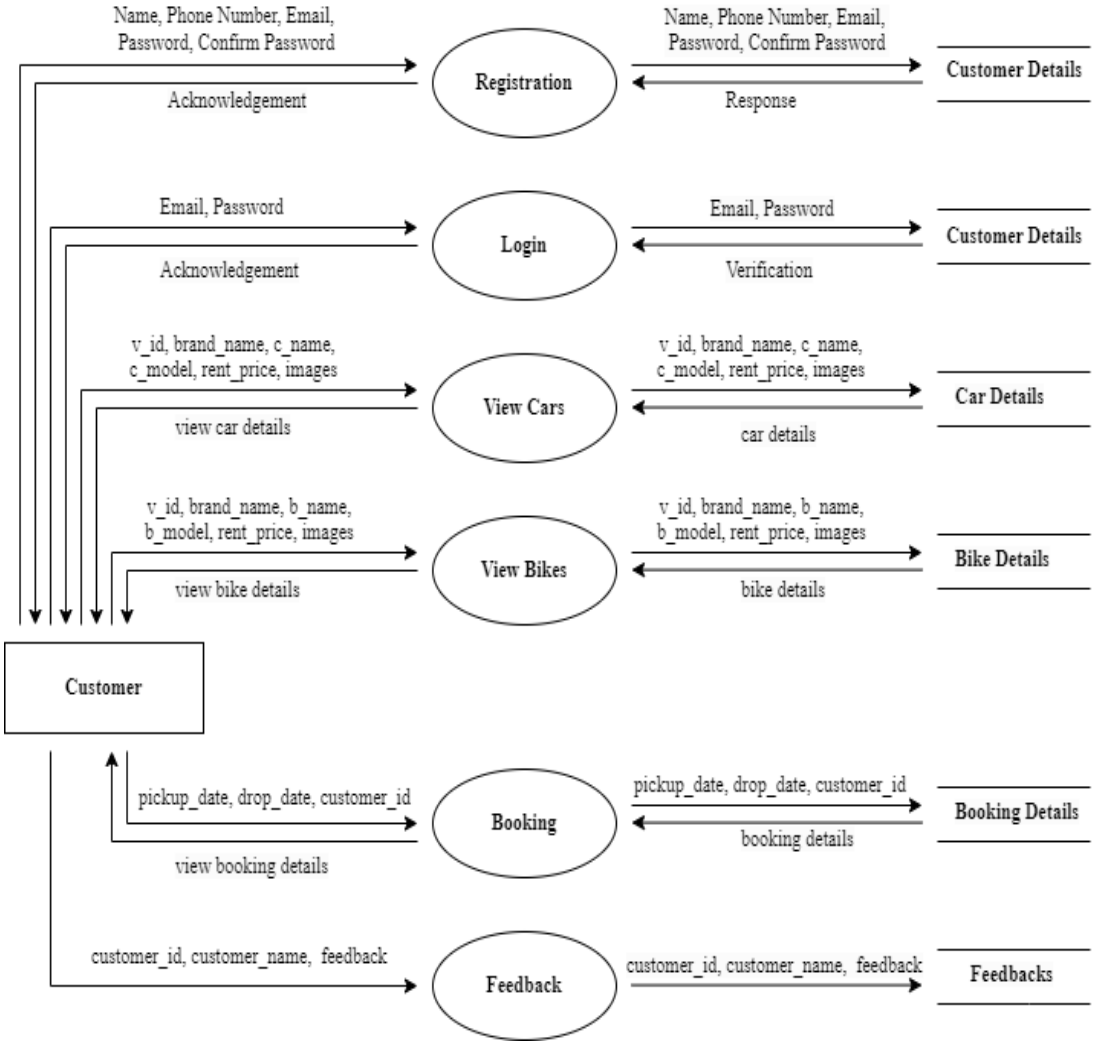


3.2.2 Level 1 DFD:

1. Admins Side:



2. Users Side:


Name, Phone Number, Email, Password, Confirm Password
Acknowledgement
Registration
Name, Phone Number, Email, Password, Confirm Password
Response
Customer Details
Email, Password
Acknowledgement
Login
Email, Password
Verification
Customer Details
v_id, brand_name, c_name, c_model, rent_price, images
view car details
View Cars
v_id, brand_name, c_name, c_model, rent_price, images
car details
Car Details
v_id, brand_name, b_name, b_model, rent_price, images
view bike details
View Bikes
v_id, brand_name, b_name, b_model, rent_price, images
bike details
Bike Details
Customer
pickup_date, drop_date, customer_id
view booking details
Booking
pickup_date, drop_date, customer_id
booking details
Booking Details
customer_id, customer_name, feedback
Feedback
customer_id, customer_name, feedback
Feedbacks

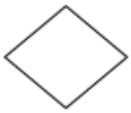
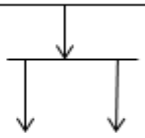
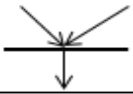
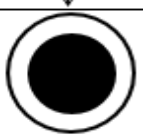
3.3 UML (Unified Modeling Language) Diagram

3.3.1 Activity Diagram:

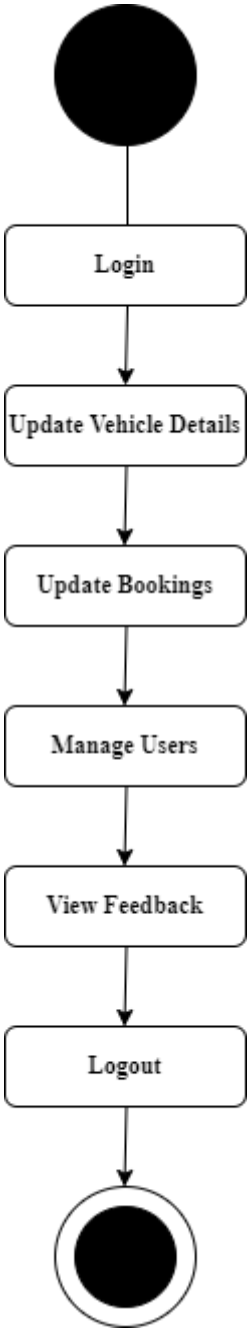
Activity Diagrams are UML Diagrams which are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram focuses on the condition of flow and the sequence in which it happens

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

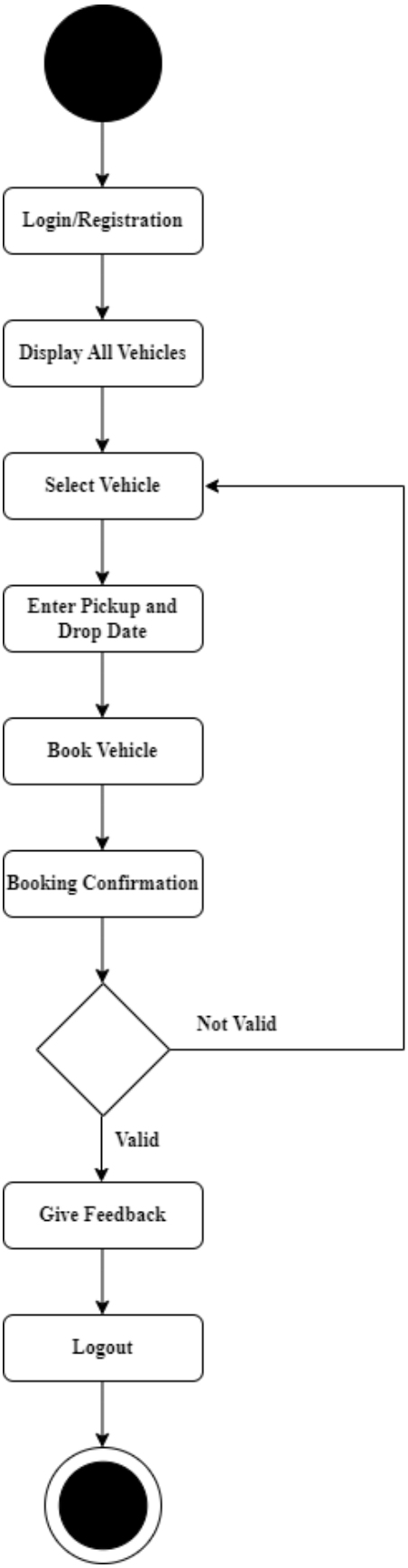
1. Identify candidate use cases, through the examination of business workflows.
2. Identify pre- and post-conditions (the context) for use cases.
3. Model in detail complex activities in a high-level activity Diagram

Sr. No	Name	Symbol
1.	Start Node	
2.	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

1. Admin


Login
Update Vehicle Details
Update Bookings
Manage Users
View Feedback
Logout

2. Customer


Login/Registration
Display All Vehicles
Select Vehicle
Enter Pickup and Drop Date
Book Vehicle
Booking Confirmation
Not Valid
Valid
Give Feedback
Logout

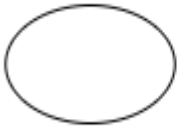
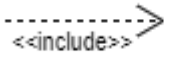
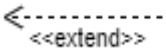
3.4 Use Case Diagram

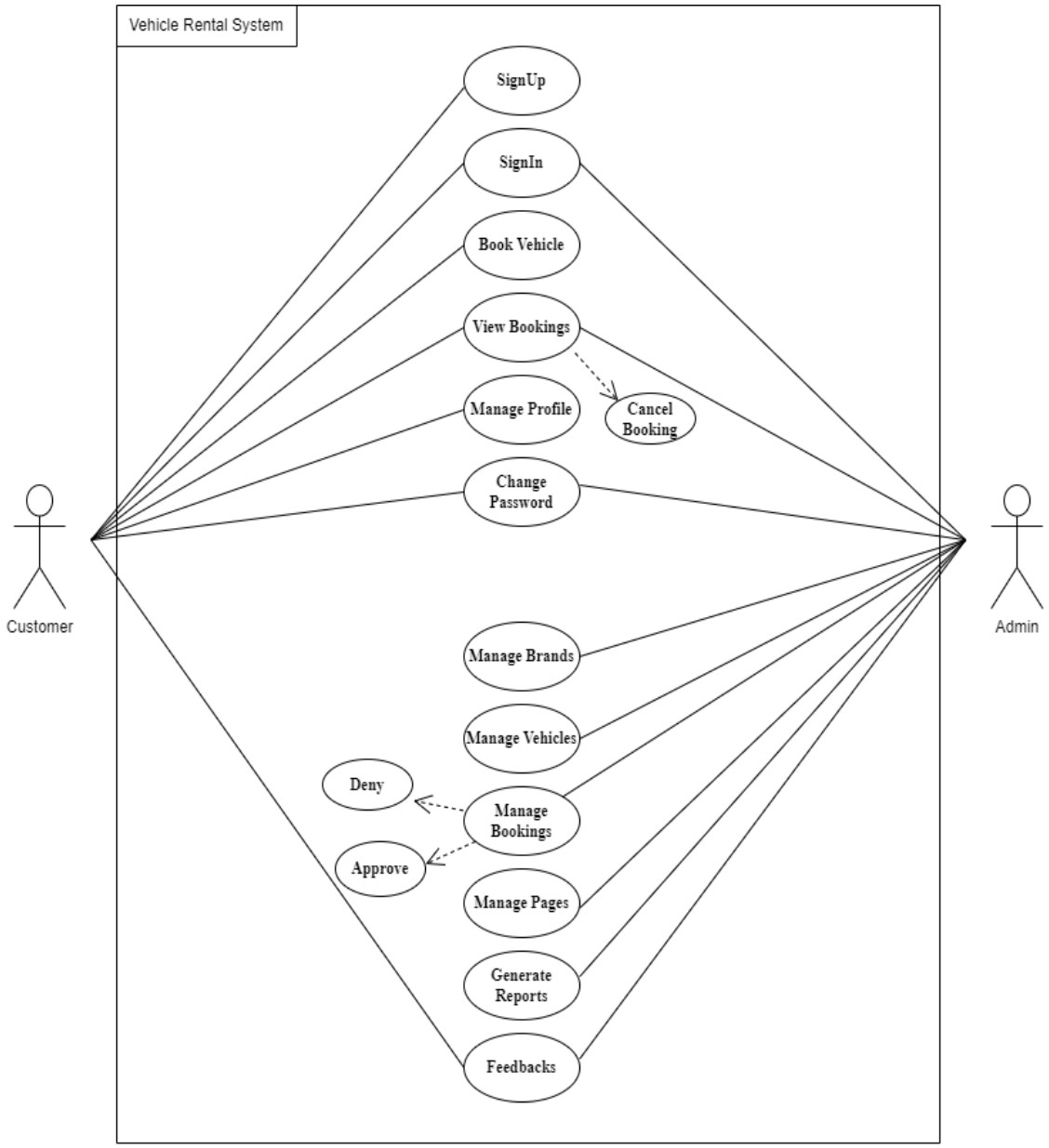
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system**.**

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

It is essential to analyze the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram .After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact. Once both the actors and use cases are enlisted, the relation between the actor and use case/ system is inspected. It identifies the no of times an actor communicates with the system. Basically, an actor can interact multiple times with a use case or system at a particular instance of time.
Following are some rules that must be followed while drawing a use case diagram:

1. A pertinent and meaningful name should be assigned to the actor or a use case of a system.
2. The communication of an actor with a use case must be defined in an understandable way.
3. Specified notations to be used as and when required.
4. The most significant interactions should be represented among the multiple no of interactions between the use case and actors

Diagram	Name	Description
	Use Case	A use case represents a user goal that can be achieved by accessing the system or software application.
	Actor	Actor and use case can be associated to indicate that the actor participates in that use case.
	Association	An actor specifies a role played by a user or any other system that interacts with the subject..
	Include	An include relationship specifies how the behaviour for the inclusion use case is inserted into the behaviour defined for the base use case.
	Extend	An extend relationship specifies how the behaviour of the extension use case can beinserted into the behaviour defined for the base use case.


Vehicle Rental System
SignUp
SignIn
Book Vehicle
View Bookings
Cancel Booking
Manage Profile
Change Password
Customer
Admin
Manage Brands
Manage Vehicles
Deny
Manage Bookings
Approve
Manage Pages
Generate Reports
Feedbacks

DATABASE MODELLING

4.1 Entity Relationship (ER) Diagram

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Here are the geometric shapes and their meaning in an E-R Diagram.

- Rectangle: Represents Entity sets
- Ellipses: Attributes
- Diamonds: Relationship Set
- Lines: They link attributes to Entity Sets and Entity sets to Relationship Set
- Double Ellipses: Multivalued Attributes
- Dashed Ellipses: Derived Attributes
- Double Rectangles: Weak Entity Sets
- Double Lines: Total participation of an entity in a relationship set

ER Diagram has three main components

1. Entity: An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

2. Weak Entity: An Entity type has a key attribute that uniquely identifies each entity in the entity set. But some entity type exists for which key attributes can't be defined**.**

3. Attribute: An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

1. Key attribute: A key attribute can uniquely identify an entity from an entity set. Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.

2. Composite attribute: An attribute that is a combination of other attributes is known as composite attribute.

3. Multivalued attribute: An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in an ER Diagram.

4. Derived attribute: A derived attribute is one whose value is dynamic

and derived from another attribute. It is represented by dashed oval in an ER Diagram.

3. Relationship: A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities.

There are four types of relationships:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

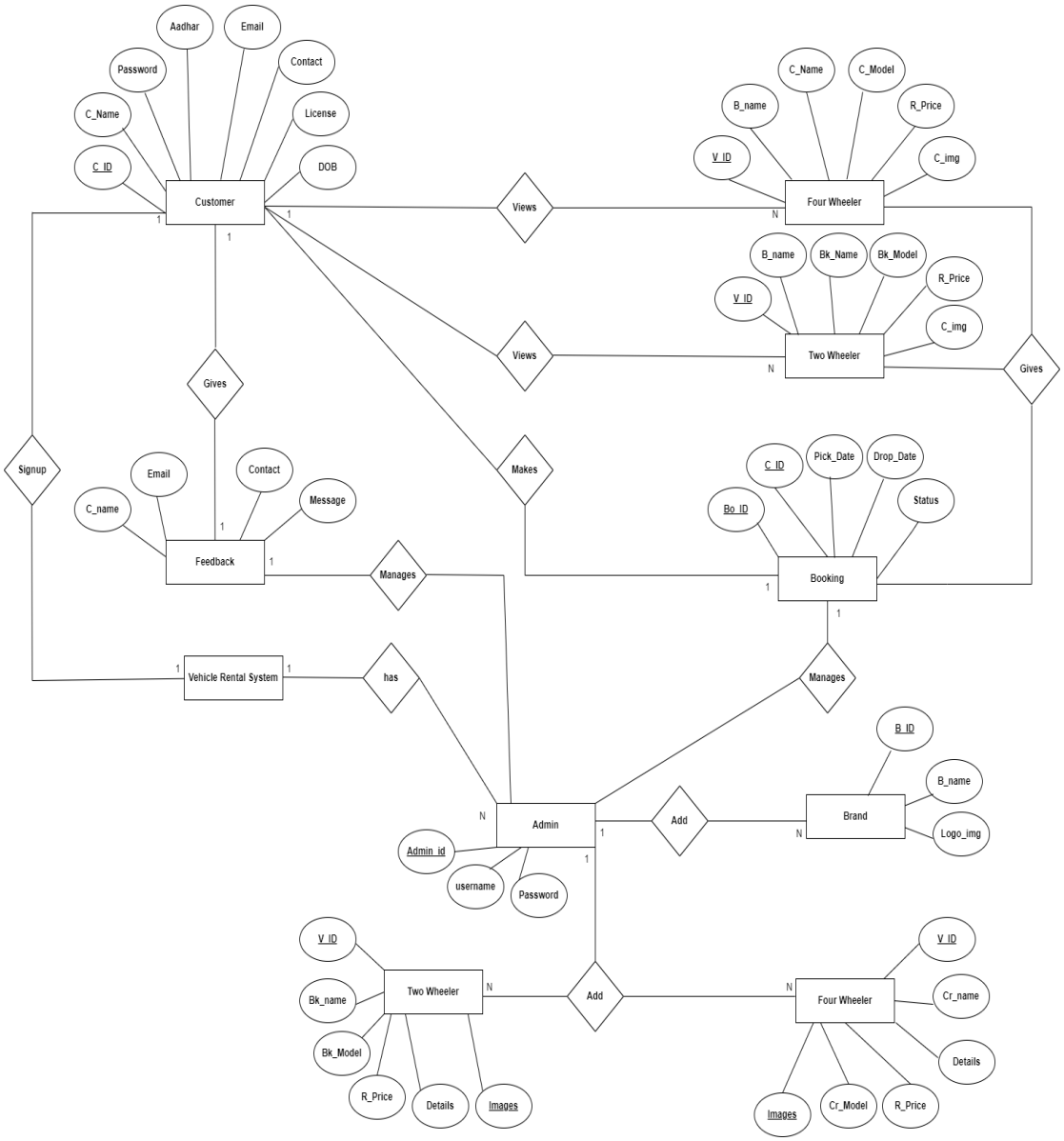
1. One to One Relationship: When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.

2. One to Many Relationship: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.

3. Many to One Relationship: When more than one instances of an entity is associated with a single instance of another entity then it is called many to one.

4. Many to Many Relationship: When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.

Total Participation of an Entity set: A Total participation of an entity set represents that each entity in entity set must have at least one relationship in a relationship set.


Aadhar
Email
Password
Contact
C_Name
License
C_ID
DOB
Customer
Views
C_Name
C_Model
B_name
R_Price
V_ID
C_img
Four Wheeler
B_name
Bk_Name
Bk_Model
R_Price
V_ID
C_img
Two Wheeler
Views
Gives
Gives
Signup
Email
Contact
C_name
Message
Feedback
Makes
C_ID
Pick_Date
Drop_Date
Bo_ID
Status
Booking
Manages
Vehicle Rental System
has
Manages
B_ID
B_name
Admin
Add
Brand
Logo_img
Admin_id
username
Password
V_ID
Bk_name
Bk_Model
R_Price
Details
Images
Two Wheeler
Add
Four Wheeler
V_ID
Cr_name
Details
Images
Cr_Model
R_Price

4.2 Table Description

A table is an arrangement of data in rows and columns, or possibly in a more complex structure. A table is a collection of related data held in a table format within a database. The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data.

A database consists of one or more tables. Each table is made up of rows and columns. Each row in a relational table is uniquely identified by a primary key. This can be by one or more sets of column values. In most scenarios it is a single column, such as student ID.

Every relational table has one primary key. Its purpose is to uniquely identify each row in the database. No two rows can have the same primary key value. The practical result of this is that you can select every single row by just knowing its primary key.

Admin

Name	Data Type	Constraints	Description
a_id	int(10)	primary key	unique id for admin
username	varchar(120)	not null	admin username
password	varchar(120)	not null	admin password
mobilenumber	bigint(10)	not null	admin phone number
regdate	timestamp	current timestamp	registration date

User

Name	Data Type	Constraints	Description
c_id	int(10)	primary key	unique id for user
firstname	varchar(120)	not null	user first name
lastname	varchar(120)	not null	user last name
email	varchar(120)	not null	user email
mobilenumber	bigint(10)	not null	user contact no
password	varchar(120)	not null	user password
dob	timestamp	not null	date of birth
dl_no	varchar(120)	not null	license number
dl_img	varchar(120)	not null	license image
aad_no	int(20)	not null	aadhar number
regdate	timestamp	current timestamp	registration date

Brand

Name	Data Type	Constraints	Description
bid	int(10)	primary key	unique id for brand
bname	varchar(120)	not null	brand name
blogo	varchar(120)	not null	brand logo image
creationdate	timestamp	current timestamp	creation date of brand

Vehicles (bike)

Name	Data Type	Constraints	Description
v_id	int(10)	primary key	unique id for vehicle
categoryname	varchar(120)	not null	vehicle category
brandname	varchar(120)	not null	vehicle brand
vehiclename	varchar(120)	not null	vehicle name
regnumber	varchar(120)	not null	registration number
rentalprice	varchar(120)	not null	renting price
vehiclemodel	varchar(120)	not null	vehicle model
vehicledescription	varchar(120)	not null	vehicle description
seatingcap	int(10)	not null	seating capacity
image1	varchar(120)	not null	vehicle image
image2	varchar(120)	not null	vehicle image
image3	varchar(120)	not null	vehicle image
image4	varchar(120)	not null	vehicle image
image5	varchar(120)	not null	vehicle image
creationdate	timestamp	current timestamp	creation date of brand

Vehicles (car)

Name	Data Type	Constraints	Description
v_id	int(10)	primary key	unique id for vehicle
categoryname	varchar(120)	not null	vehicle category
brandname	varchar(120)	not null	vehicle brand
vehiclename	varchar(120)	not null	vehicle name
regnumber	varchar(120)	not null	registration number
rentalprice	varchar(120)	not null	renting price
vehiclemodel	varchar(120)	not null	vehicle model
vehicledescription	varchar(120)	not null	vehicle description

seatingcap	int(10)	not null	seating capacity
image1	varchar(120)	not null	vehicle image
image2	varchar(120)	not null	vehicle image
image3	varchar(120)	not null	vehicle image
image4	varchar(120)	not null	vehicle image
image5	varchar(120)	not null	vehicle image
creationdate	timestamp	current timestamp	creation date of brand

Bookings

Name	Data Type	Constraints	Description
b_id	int(10)	primary key	unique id for booking
v_id	int(10)	foreign key	foreign key
c_id	int(10)	foreign key	foreign key
vehiclename	varchar(120)	not null	vehicle name
regno	varchar(120)	not null	registration number
location	varchar(120)	not null	renting location
rentalprice	double(100)	not null	renting price
pickupdate	timestamp	not null	pickupdate
dropdate	timestamp	not null	dropdate
totalprice	double(100)	not null	total renting price
rentingdate	timestamp	current timestamp	rentingdate

Feedback

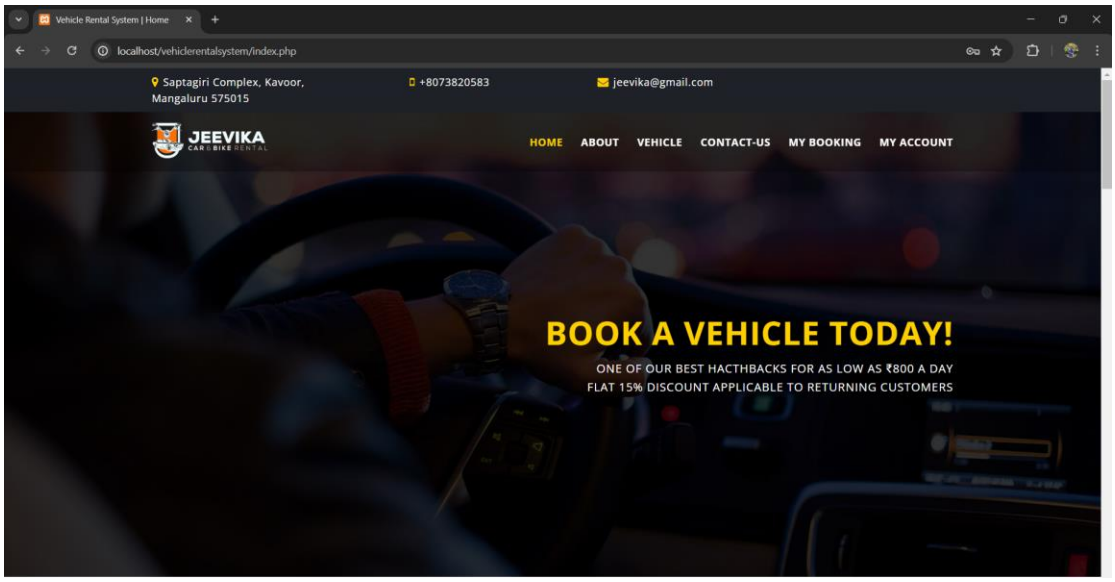
Name	Data Type	Constraints	Description
fullname	varchar(120)	notnull	user full name
email	varchar(120)	not null	user email
mobilenumber	bigint(10)	not null	user contact no
message	longvarchar	not null	feedback message

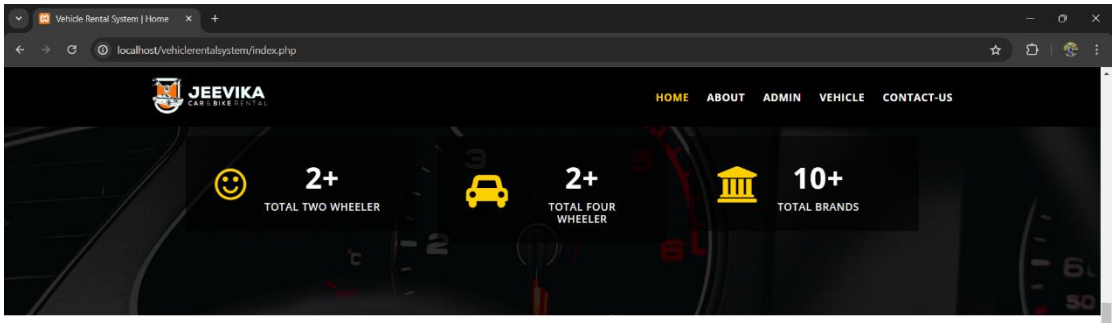
Pages

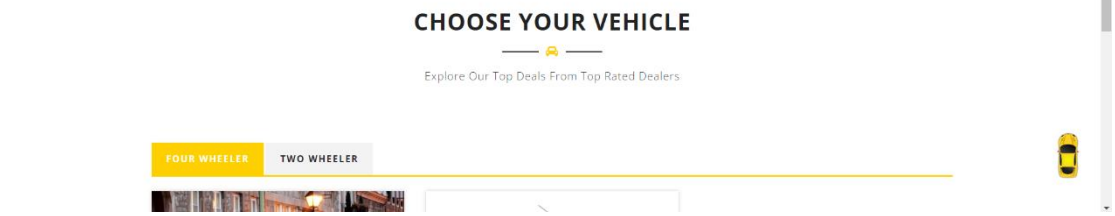
Name	Data Type	Constraints	Description
pid	int(10)	primary key	unique key for pages
pagetype	varchar(120)	not null	page type
pagetitle	varchar(120)	not null	title of page
pagedescription	mediumtext	not null	descriptions for page
email	varchar(120)	not null	company email
mobilenumber	bigint(10)	not null	company contact number
updationdate	timestamp	currenttimestamp	updationdate

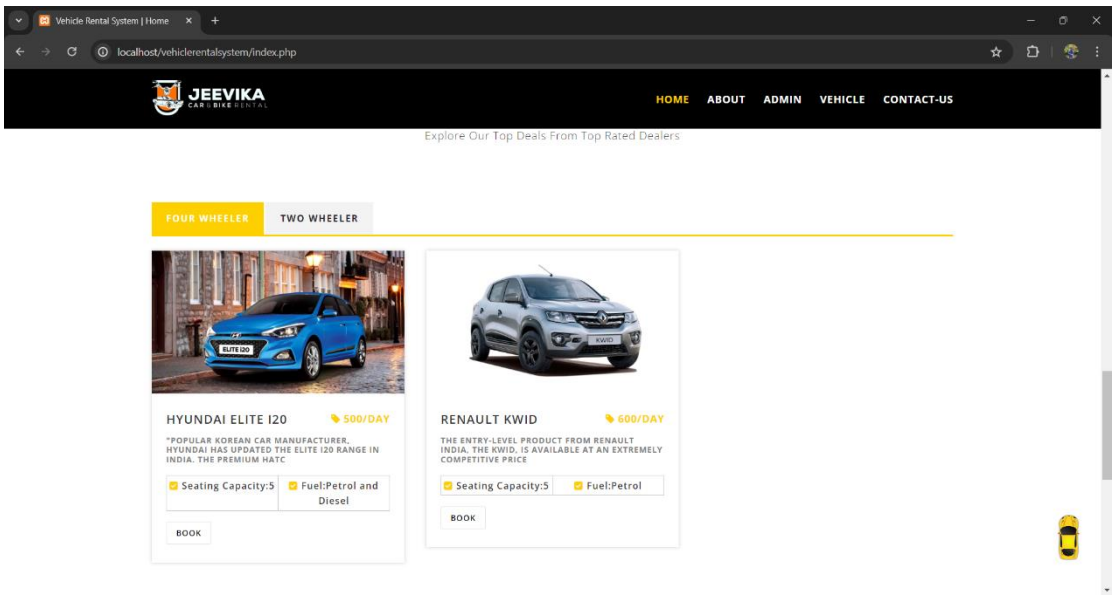
TEST CASES

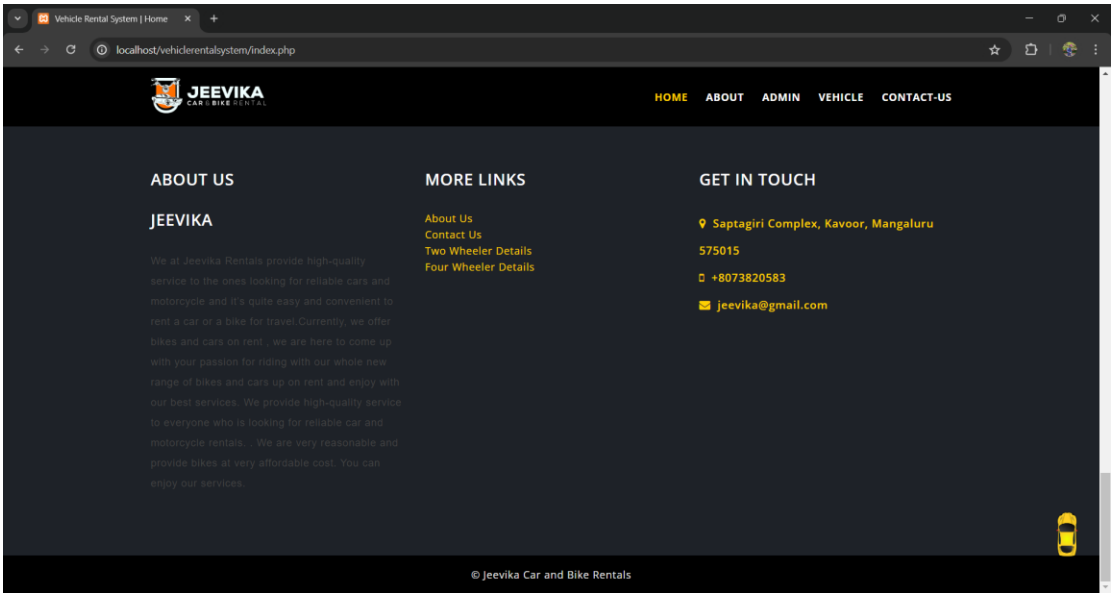
5.1.1 Home Page



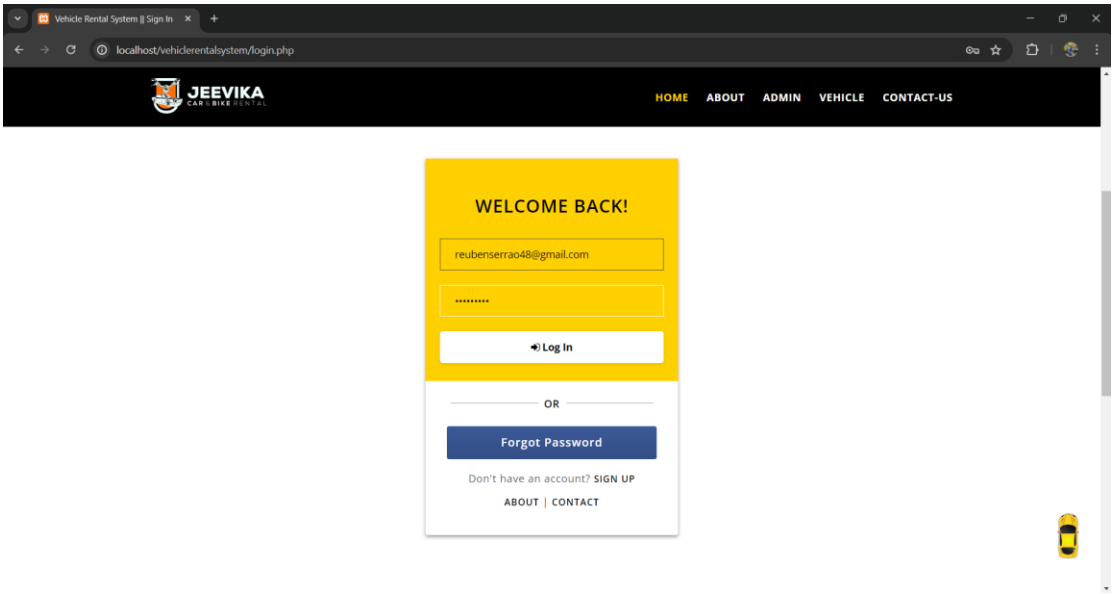




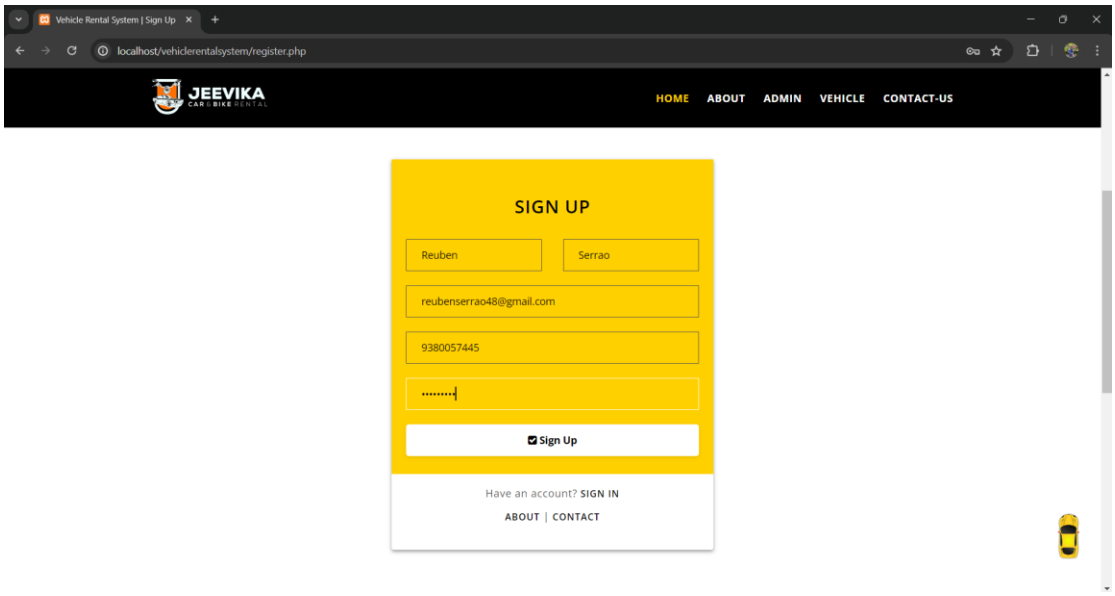




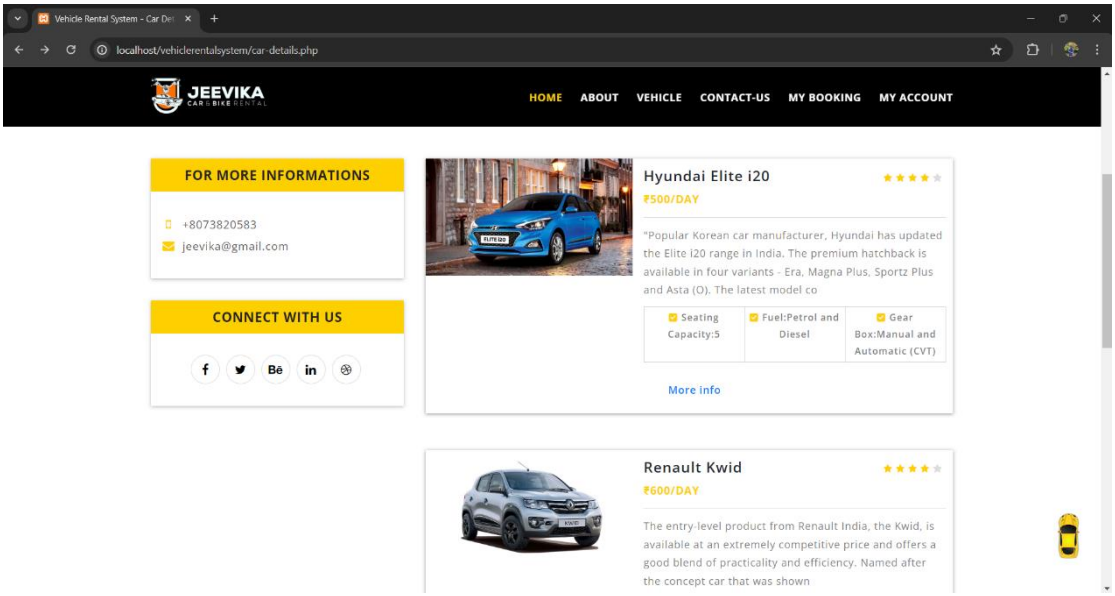
5.1.2 Login Page

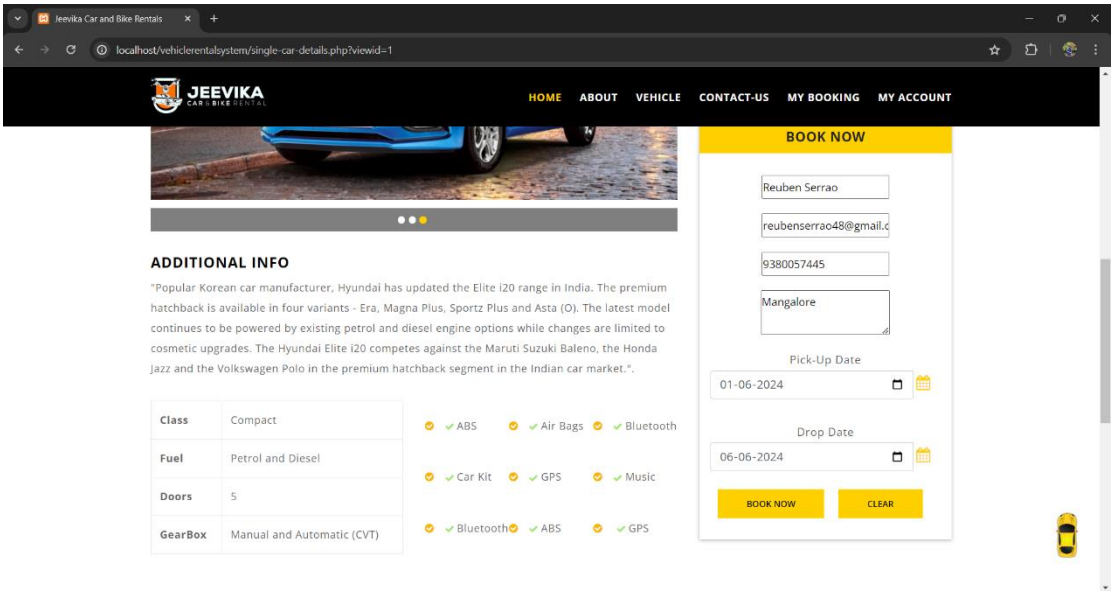


5.1.3 User Registration Page

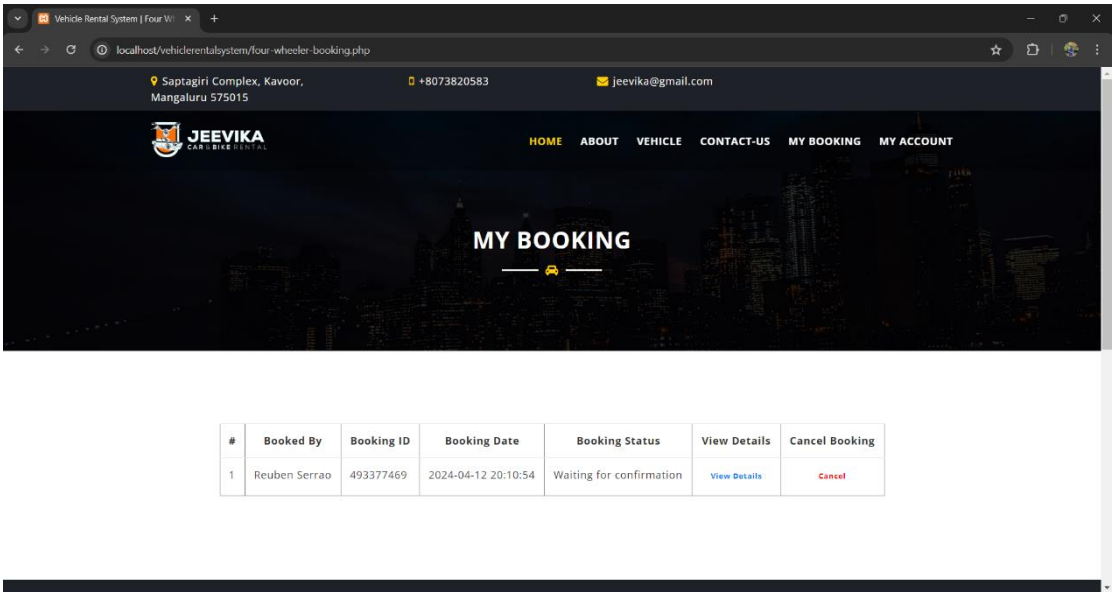


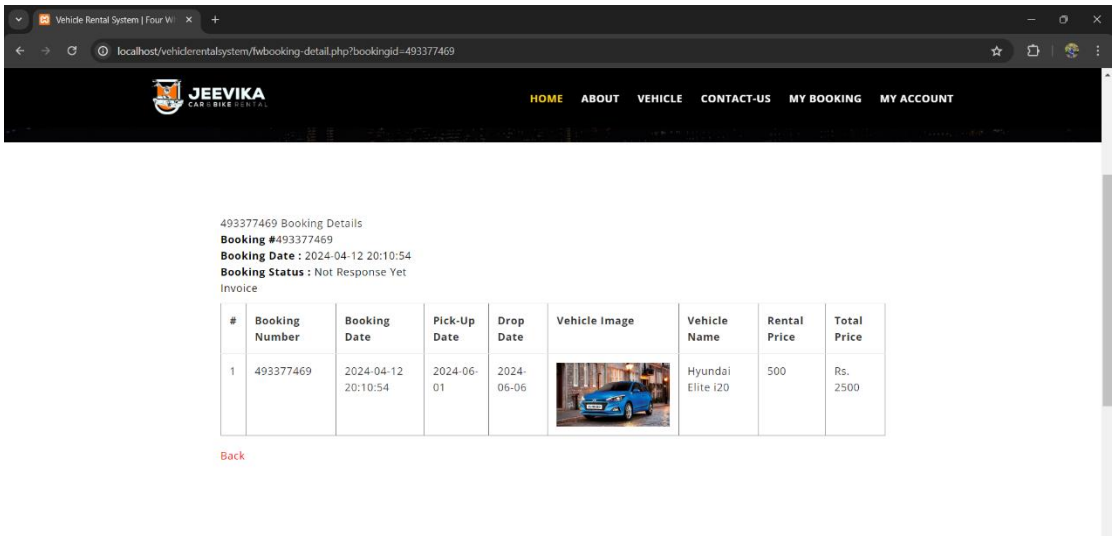
5.1.4 Booking Vehicle Page



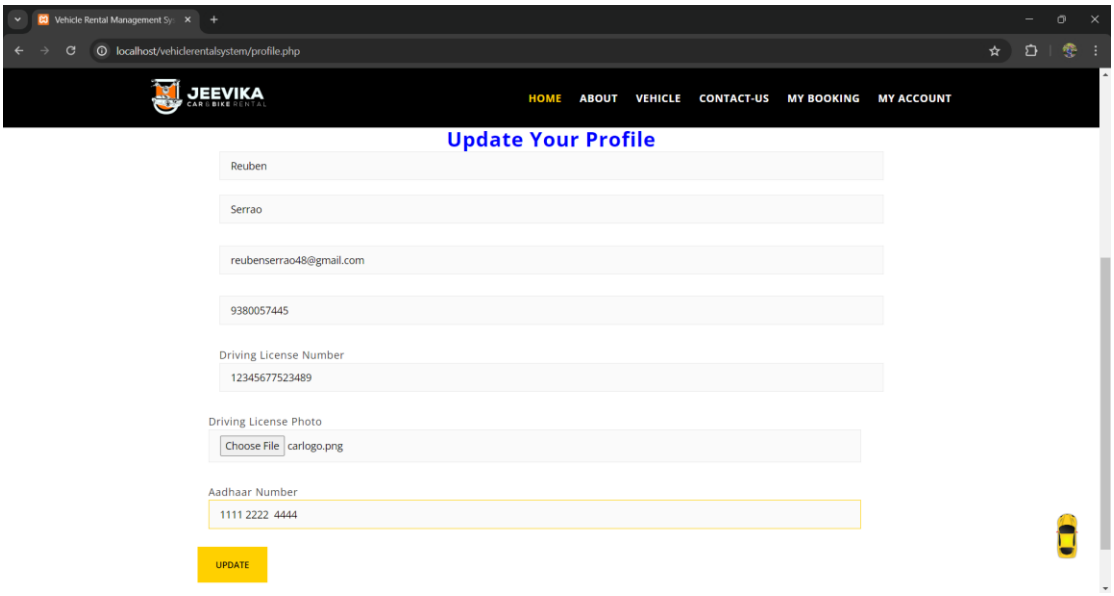


5.1.5 View Booking Page

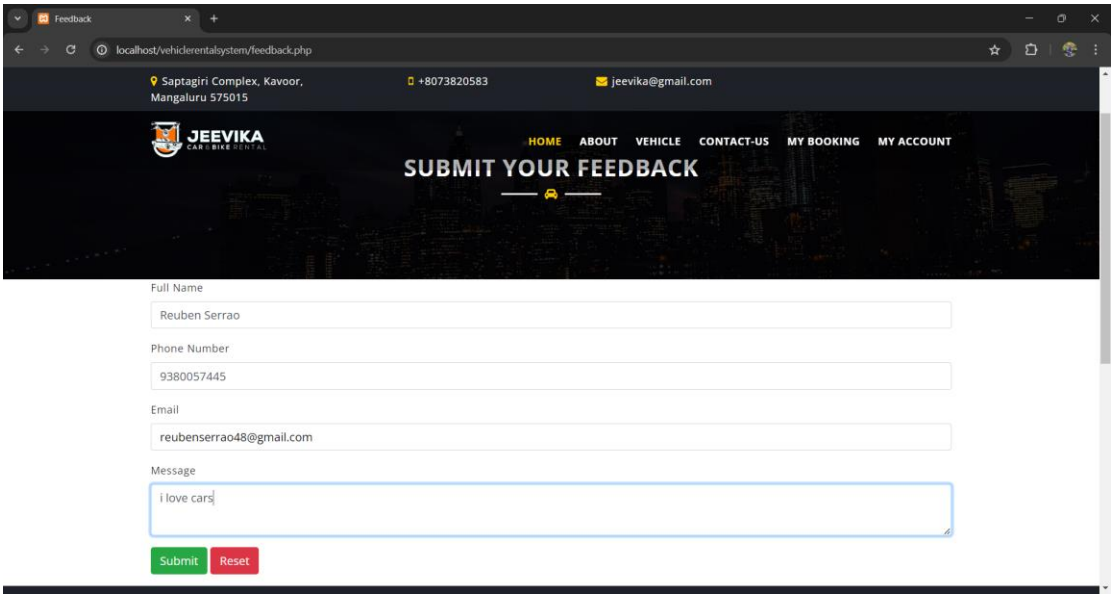




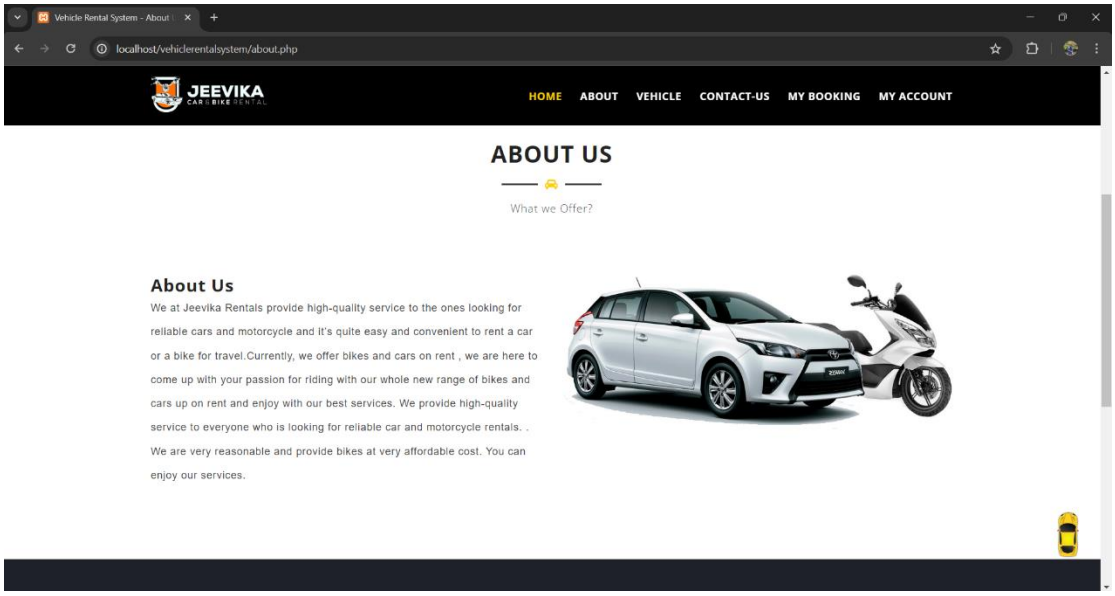
5.1.6 User Profile Page



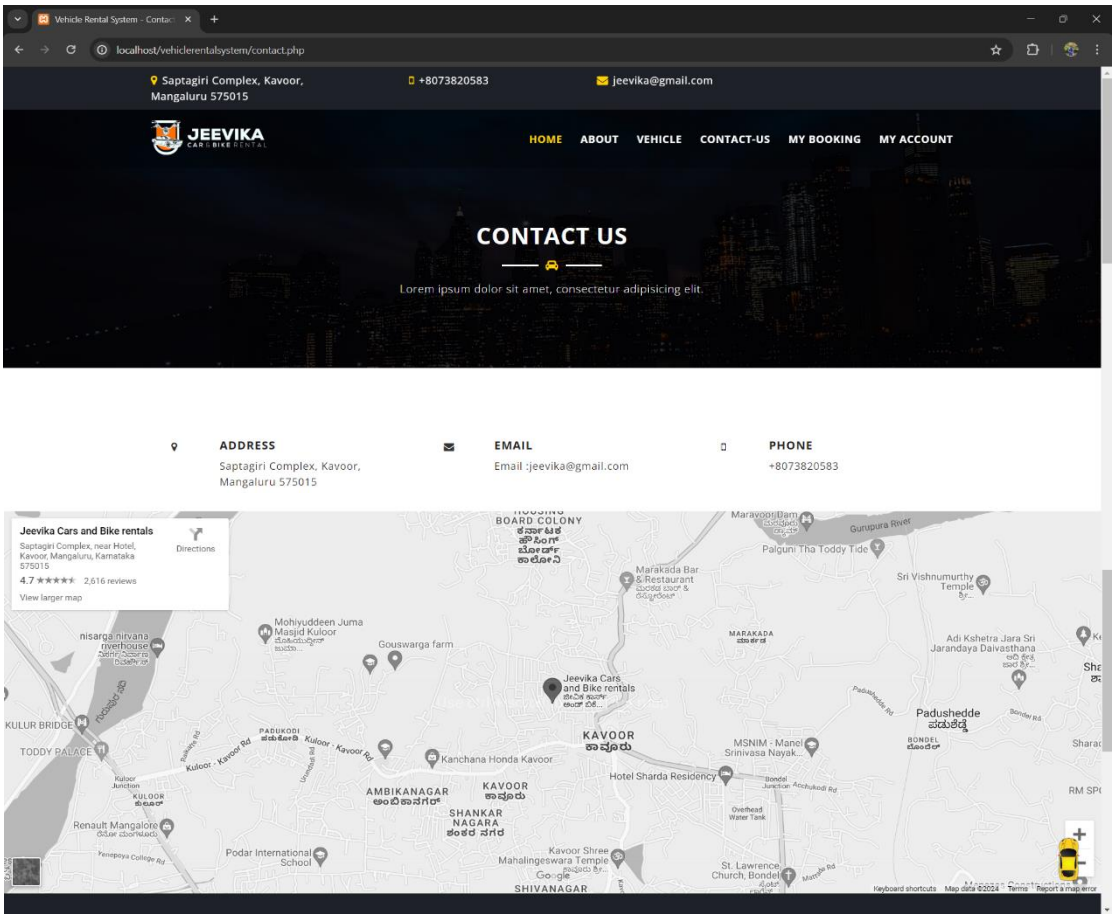
5.1.7 Feedback Page



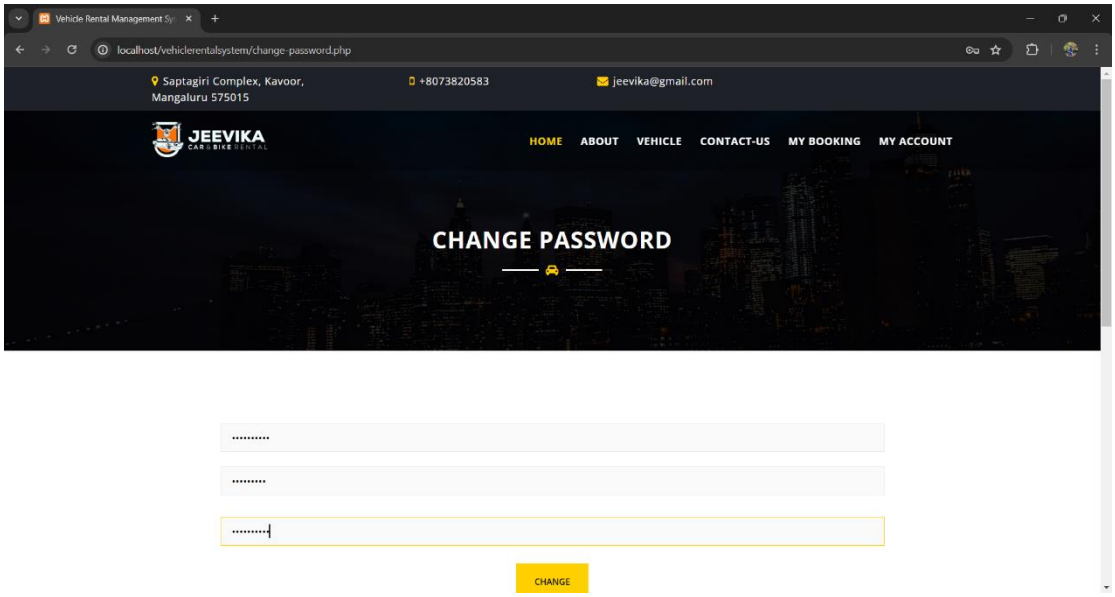
5.1.8 About Us Page



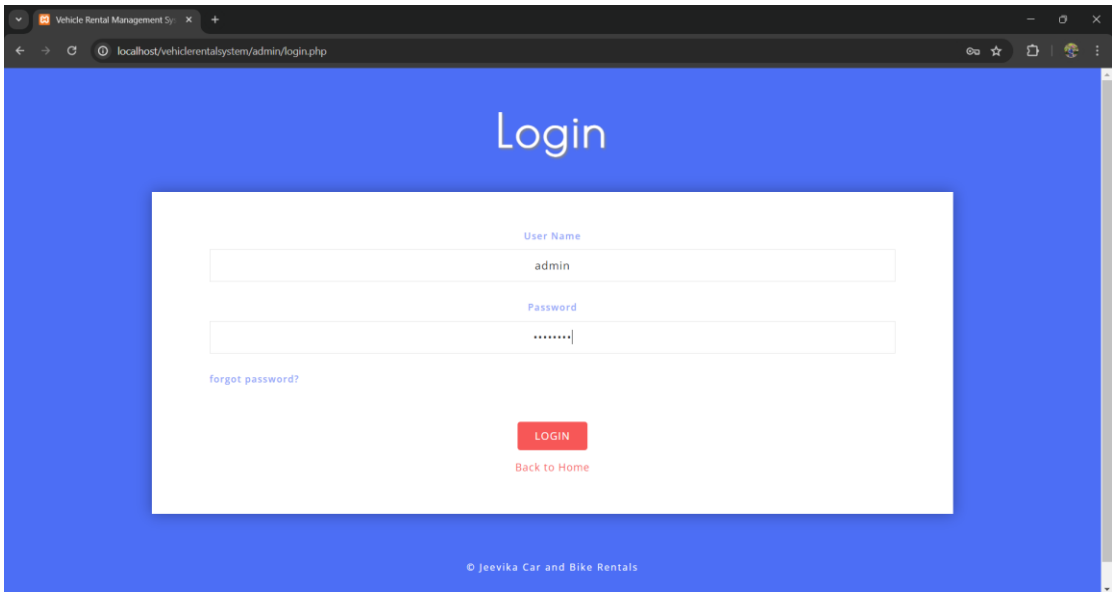
5.1.9 Contact us Page



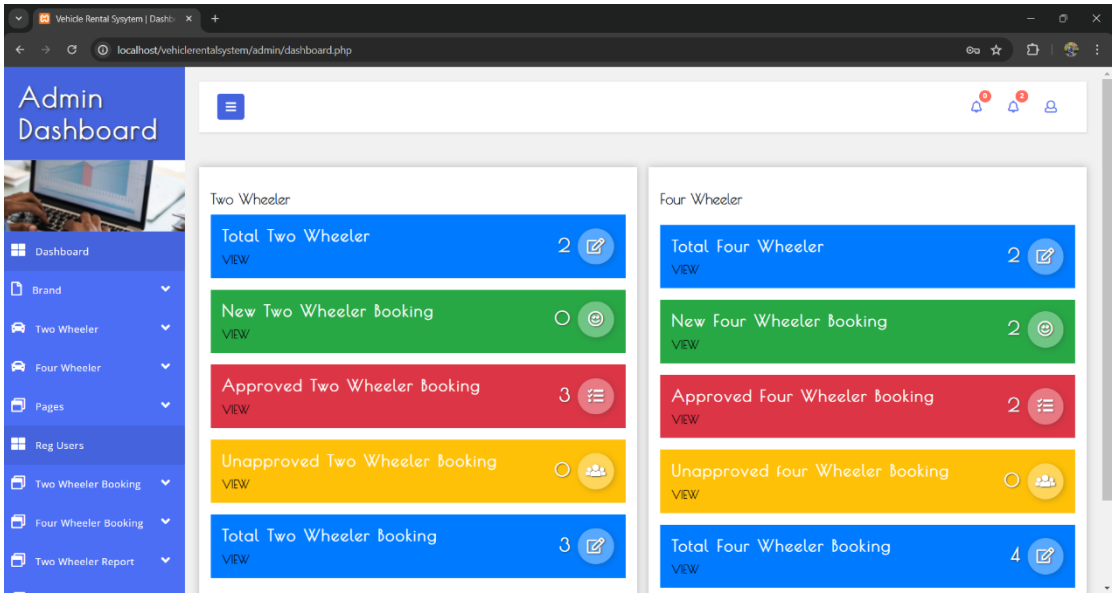
5.1.10 Change Password



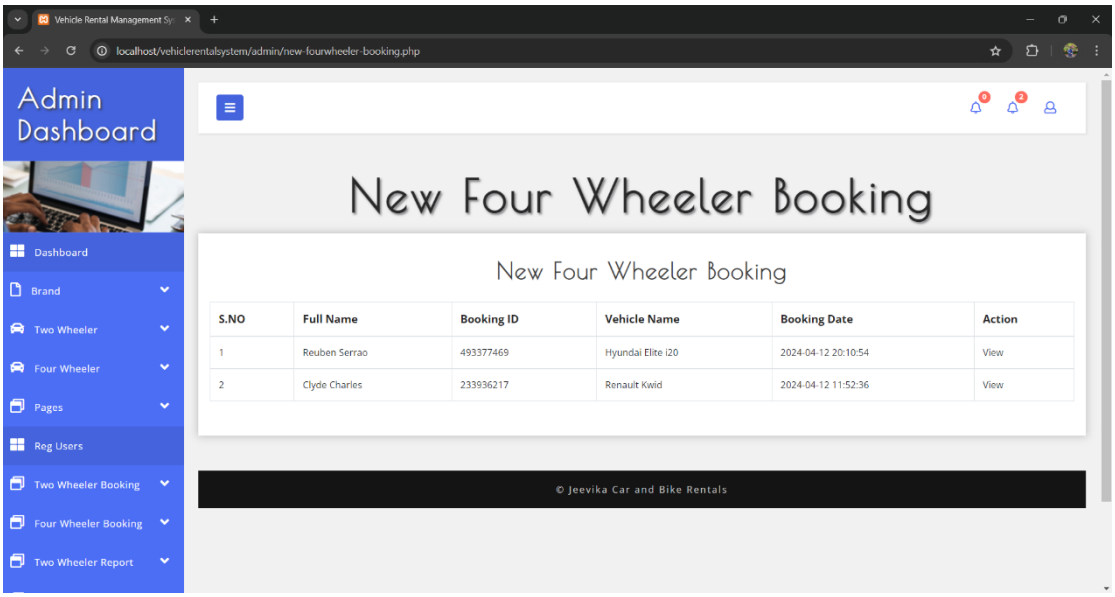
5.1.11 Admin Login

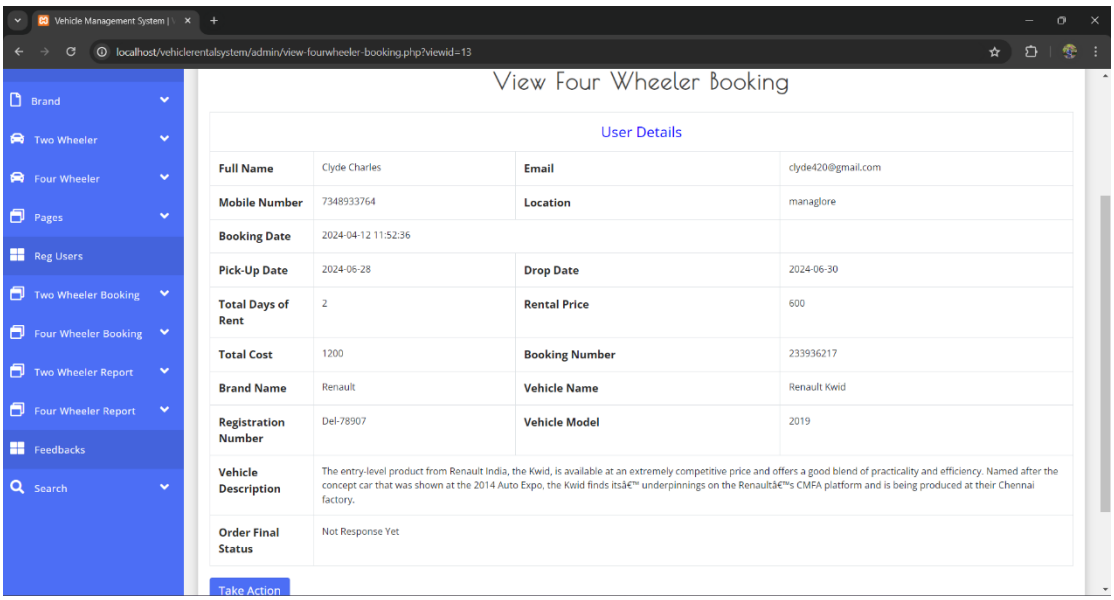


5.1.12 Admin Dashboard

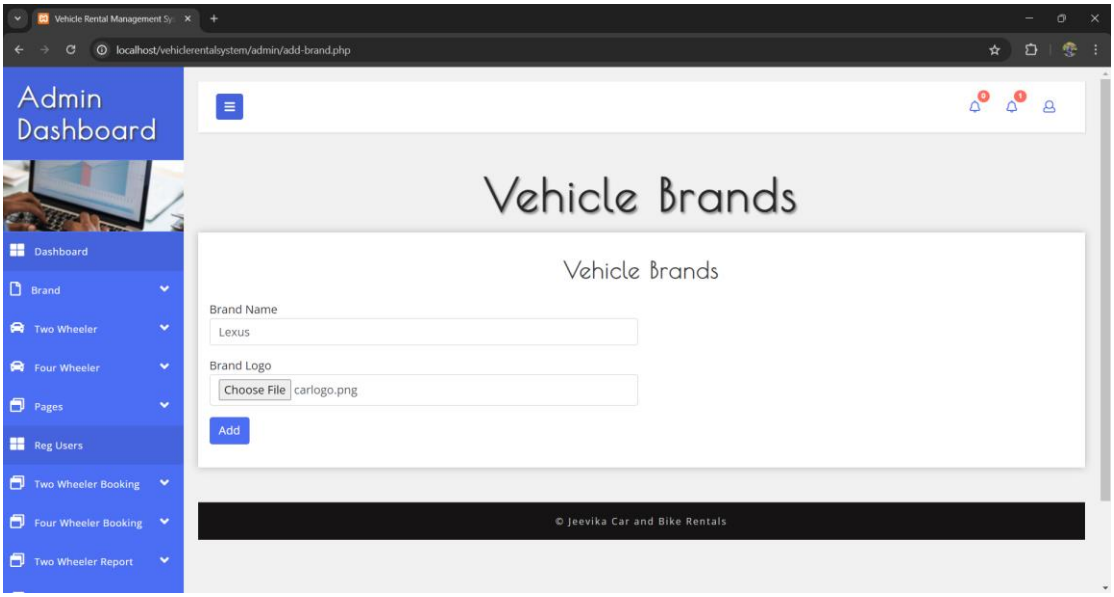


5.1.13 View Bookings

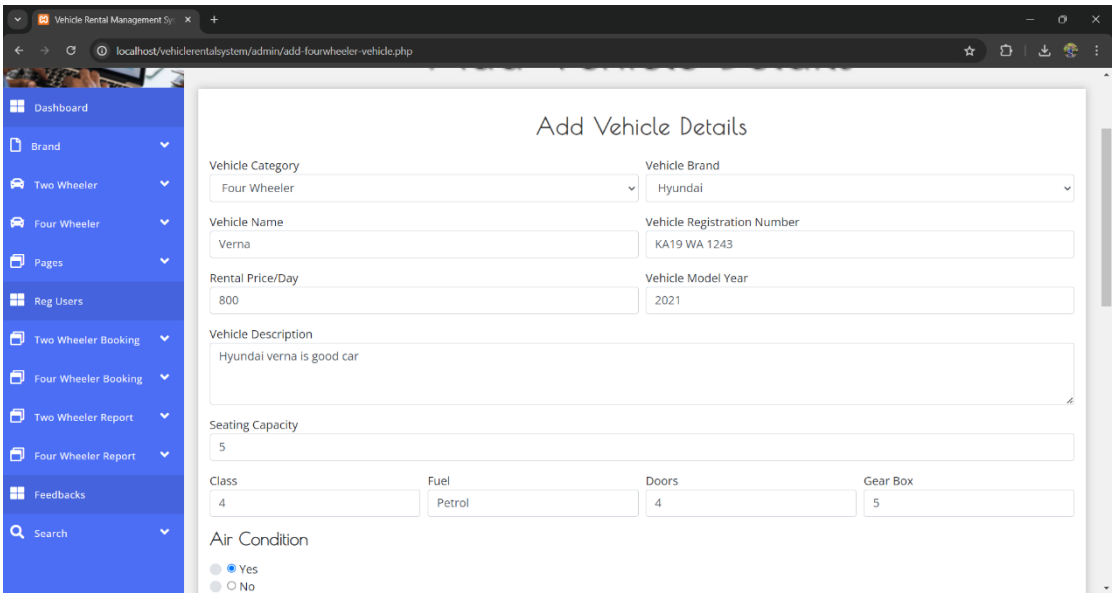


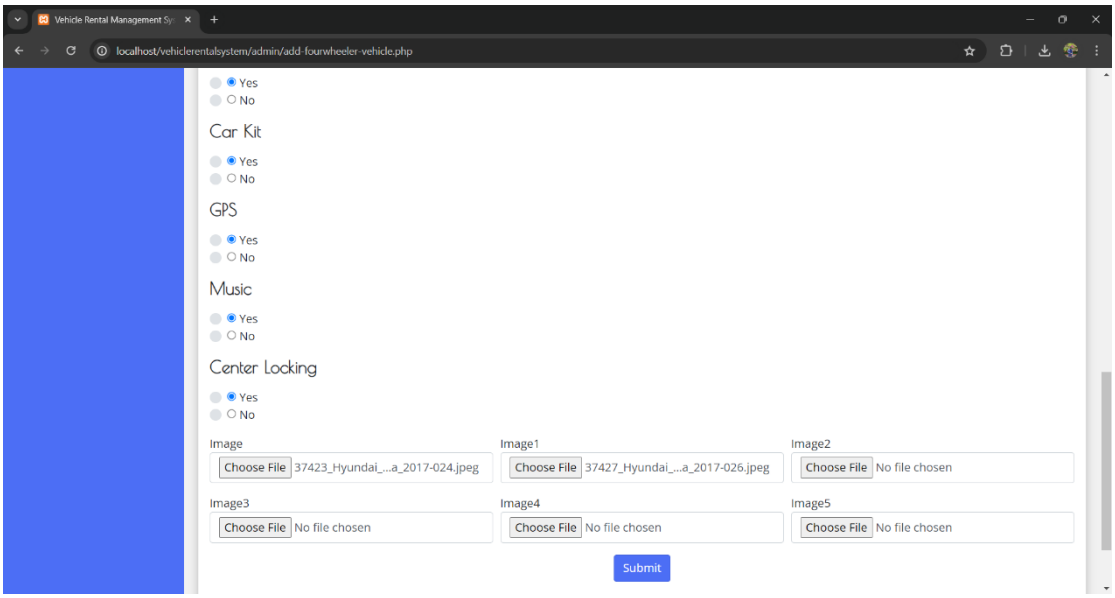


5.1.14 Add Brand

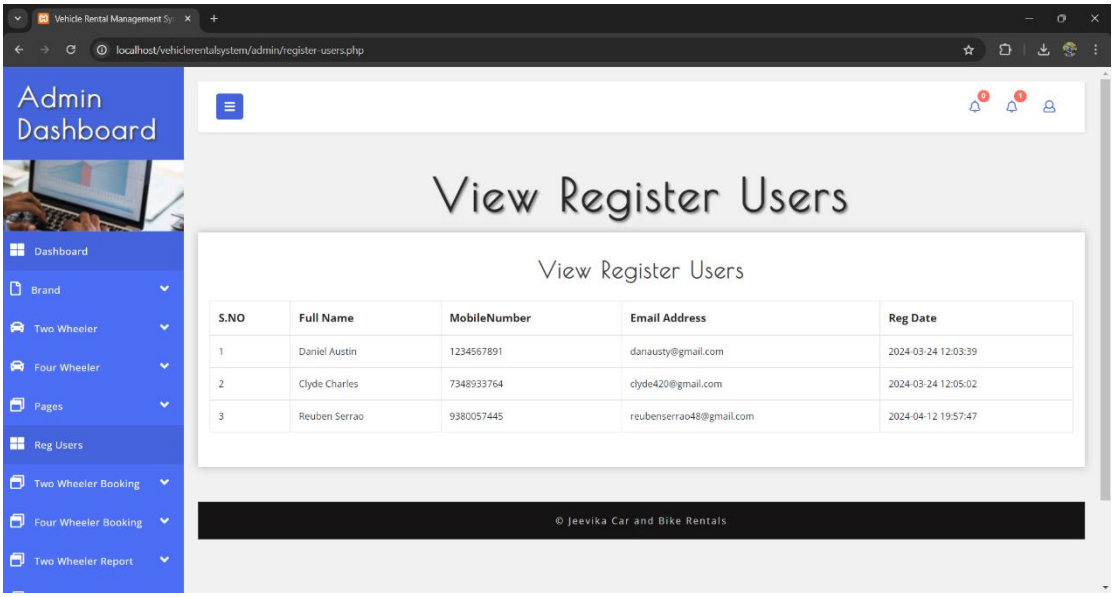


5.1.15 Add Vehicle page

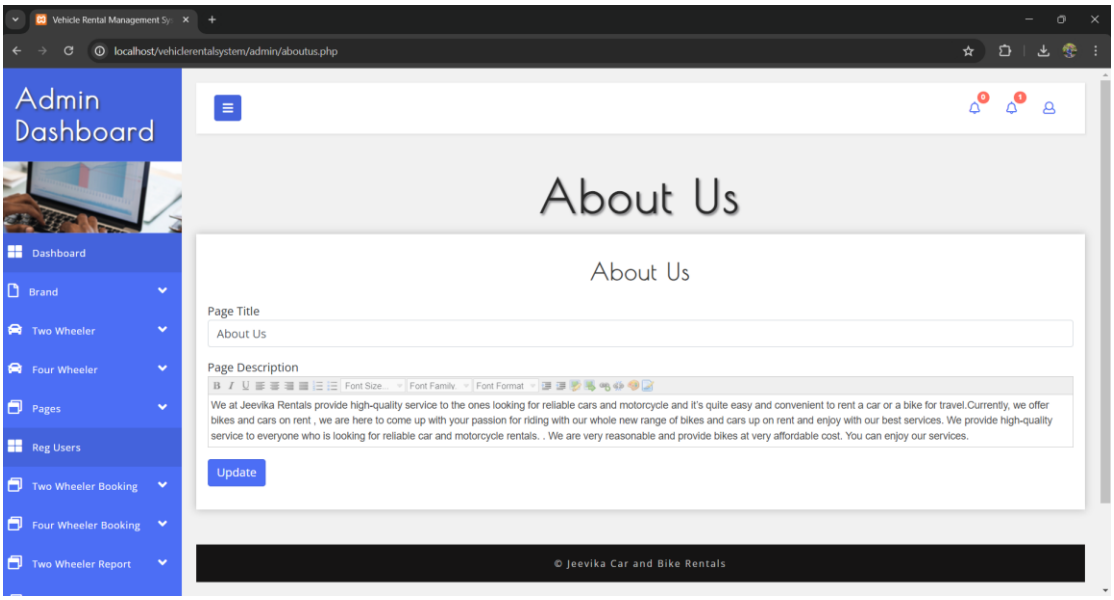




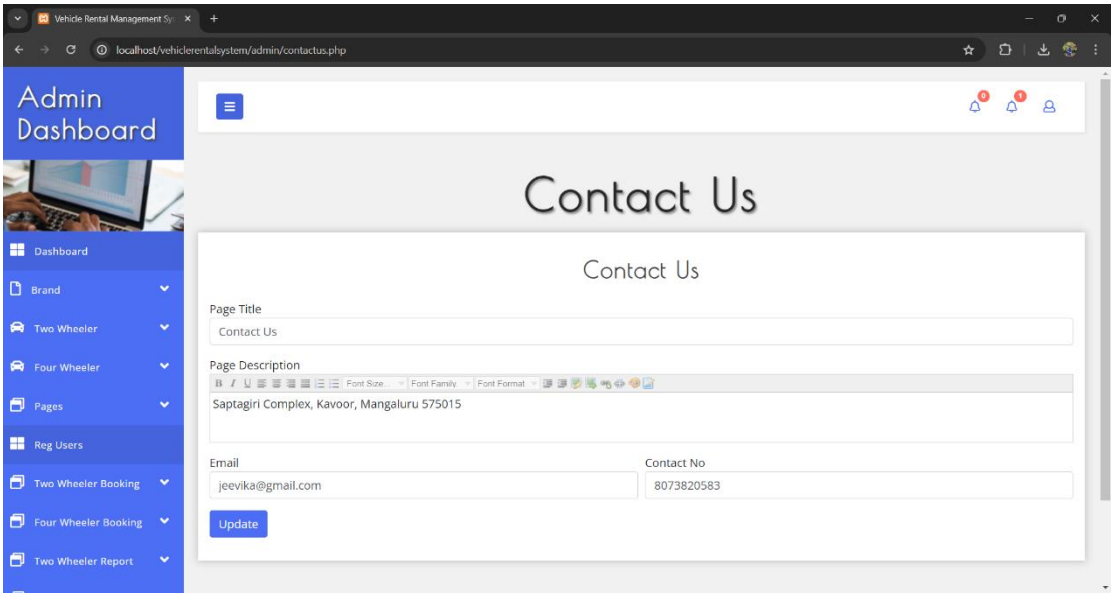
5.1.16 View Registered Users page



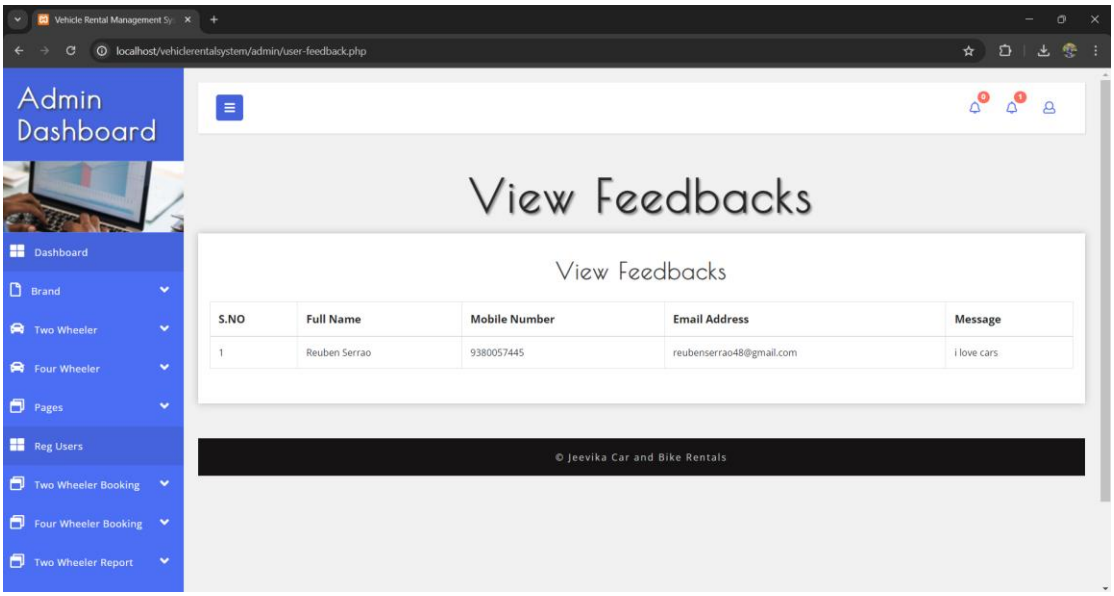
5.1.17 Manage About Us Page



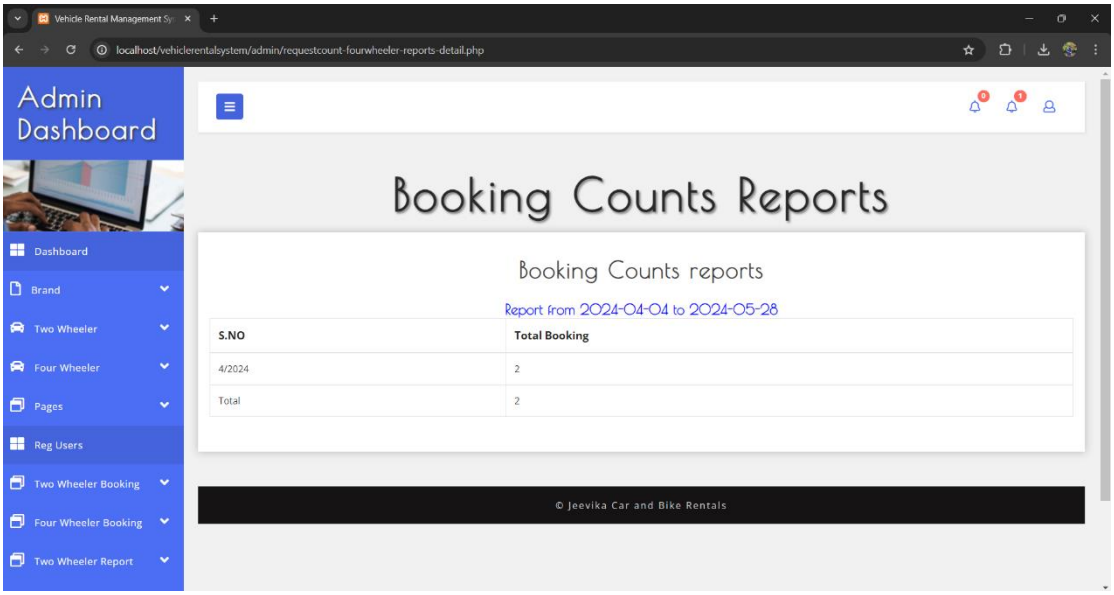
5.1.18 Manage Contact Us Page



5.1.19 View Feedbacks Page



5.1.20 Generate Reports page



CONCLUSION

The **Vehicle Rental System** aims to enhance business processes, provide online vehicle reservations, manage customer registrations, and facilitate group bookings for events. Leveraging internet technology, this system expands the rental company's reach at ease and improves return on investment (ROI)

In conclusion, the project successfully achieves its objectives by streamlining car rental processes, enhancing customer experience, and enabling efficient management of the rental fleet. The project focuses on customer-centricity, growth, innovation, and efficiency. By digitizing client details and improving search processes, it enhances overall performance.

FUTURE SCOPE

1. IoT and Telematics: Utilizing Internet of Things (IoT) devices for real-time monitoring, predictive maintenance, and gathering data on vehicle usage patterns.

2. Smart Contracts: Implementing smart contracts to automate rental agreements and transactions, reducing paperwork and increasing efficiency.

BIBILOGRAPHY

WEBSITES REFERRED

1. Car Rental System Project - ProjectsGeek

2. Car Rental System (researchgate.net)

3. GeeksforGeeks | A computer science portal for geeks

4. Tutorials List - Javatpoint

5. W3Schools Online Web Tutorials

TEXT BOOKS REFERRED

1. PHP - A Beginners Guide by Vikram Vaswani

2. Aggarwal K.K and Singh Yogesh (2008). Software Engineering Textbook.