VEHICLE BOOKING SYSTEM

Summer Internship Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Computer Application

By:
BHUMIK RATHOD(22MCA009)

Guided By
Prof. SONIA MITTAL

MASTER OF COMPUTER APPLICATION
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the Summer Internship Project entitled "Vehicle Booking System" submitted by Bhumik Rathod (22MCA009) towards the partial fulfilment of the requirements for the degree of Master of Computer Application of Nirma University is the record of the work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof. Sonia Mittal
Assistant Professor
Computer Science & Engg. Dept.
Institute of Technology
Nirma University
Ahmedabad

Dr. Madhuri Bhavsar
Head of Department
Computer Science & Engg. Dept.
Institute of Technology
Nirma University
Ahmedabad

INTERNSHIP CERTIFICATE

Certificate
OF INTERNSHIP

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Mr. Bhumik Rathod

We are happy to certify that **Mr. Bhumik Rathod** has successfully completed internship as **"Web Developer"** during 01 June 2023 to 15th July 2023.

During internship, He worked for Vehicle Booking System using PHP & MySql
We appreciate his work and contributions.

CLARION IT
PRIVATE LIMITED

Manager
Clarion IT Private Limited

INDEX

Sr. No.	Topic	Page No.
01	Title and Scope	01
02	System Requirements	02
03	Tools and Technology Used	04
04	Diagrams	06
05	Data Dictionary	09
06	Screenshots	10
07	Testing	19
08	Bibliography	22

TITLE AND SCOPE

TITLE:

VEHICLE BOOKING SYSTEM

DEFINITION:

A Vehicle Booking System is a software solution that enables customers to book vehicles, such as cars or buses. The Main Aim is to Develop a secure, user-friendly vehicle booking system that allows customers to easily reserve the vehicles online and manage their bookings. The system should provide an efficient and cost-effective way to track and manage vehicle reservations, as well as provide customers with a variety of payment options. Additionally, the system should provide customers with detailed information about their bookings, such as the vehicle specifications, and driver name.

SCOPE:

The scope of the Vehicle Booking System (VBS) is to provide an online booking service for customers to hire vehicles from a fleet of vehicles. The system will allow customers to book vehicles of different categories. The system will also allow the customers to pay for their bookings and track their rental history**.**

Some key modules that come under the scope of our system are listed as below:

1. User Registration: This module will allow users to register themselves on the system. It will collect the necessary details such as name, address, email, contact number, etc.
2. Vehicle Listing: This module will provide the list of vehicles available for rent. It will include information such as vehicle make and model, rate of rental, availability, etc.
3. Booking: This module will enable users to book vehicles for rent. It will provide the necessary features such as selecting a vehicle, booking dates, payment gateway, etc.
4. Payment Gateway: This module will provide a secure payment gateway for users to make payments for their bookings.
5. Feedback: This module will enable users to provide feedback regarding their experience with the system. It will include features such as rating the service, providing comments, etc.
6. Reports: This module will generate various reports such as booking statistics, customer feedback, vehicle usage, etc. This will help to get an insight into the performance of the system.

SYSTEM REQUIREMENTS

❖ Functional Requirements

- User Authentication:
  User should be able to sign up and login with valid username and password.

- Vehicle Selection:
  User should be able to select the vehicle of their choice from a list of available vehicles.

- Vehicle Booking:
  User should be able to book a vehicle for a specific time and date.

- Calculations:
  The system should calculate the total cost of the rental based on the type of vehicle and the rental period.

- Payment Processing:
  User should be able to make payment for the booked vehicle using online payment gateways.

- Cancellation:
  User should be able to cancel their bookings.

- Details:
  The system should provide customers with the ability to track their rental history and view their booking details.

- Feedback:
  User should be able to provide feedback about their experience with the service.

- Admin Panel:
  The system should provide admins with the ability to manage users, bookings, and the vehicles, including adding, editing and deleting vehicles.

- Reporting:
  Admin should be able to generate reports based on bookings and users.

❖ **Non-Functional Requirements**

- Security:
  The system should be secure from unauthorized access and data loss. Also, it should protect customer data.

- Performance:
  The system should respond quickly and efficiently to user requests.

- Usability:
  The system should be easy to use and navigate.

- Reliability:
  The system should be highly reliable and available at all times.

- Scalability:
  The system should be able to handle an increasing amount of user requests or, high volumes of traffic.

- Compatibility:
  The system should be compatible with all popular web browsers.

- Maintenance:
  The system should be easy to maintain and upgrade.

- Portability:
  The system should be able to run on a variety of platforms.

- Documentation:
  Comprehensive documentation should be provided for the system.

- Testing:
  The system should be thoroughly tested for bugs and errors.

TOOLS AND TECHNOLIGIES USED

TOOLS:

- VsCode
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.

- XAMPP
  - XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

FRONT END:

- HTML
  - The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the meaning and structure of web content. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

- CSS
  - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

- Bootstrap
  - Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

BACK END:

- PHP
    - PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

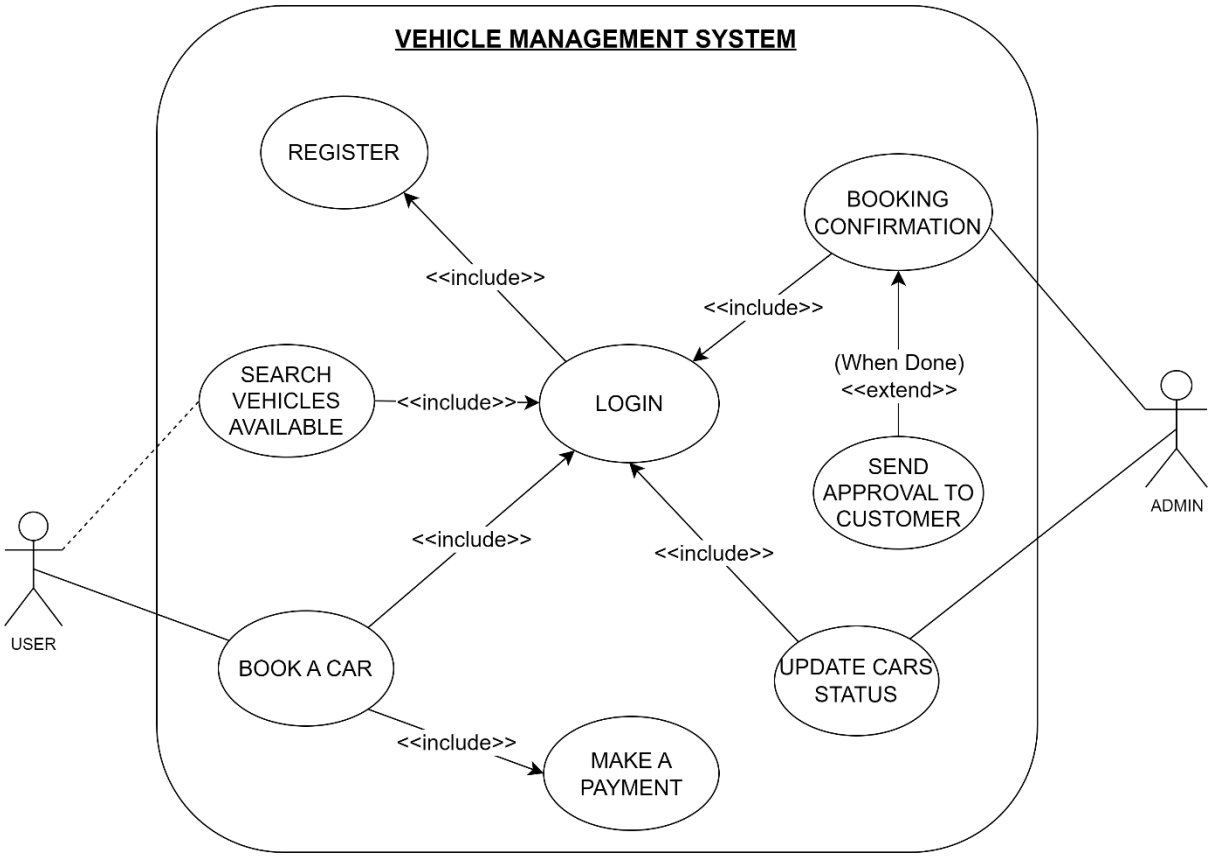
- JavaScript
    - JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2023, 98.7% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

DATABASE:

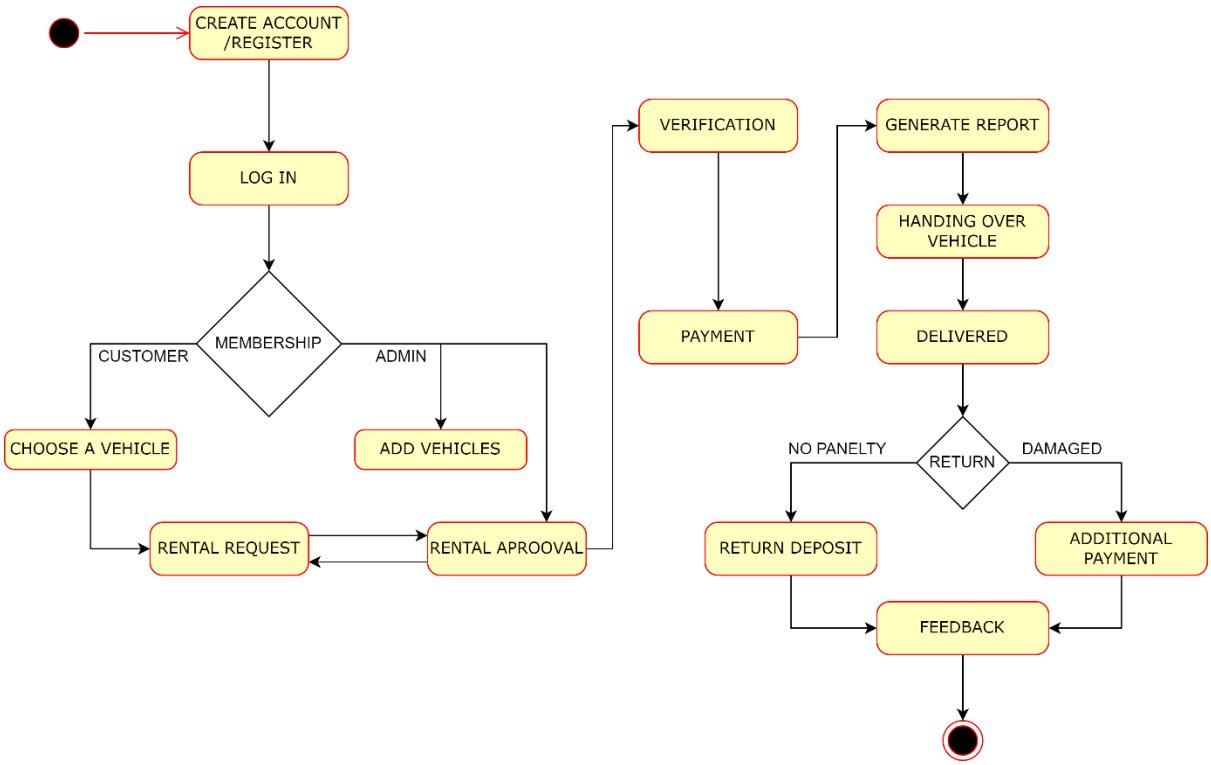
- MySql
    - MySQL is an open-source relational database management system (RDBMS). Basically, it is a Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

DIAGRAMS

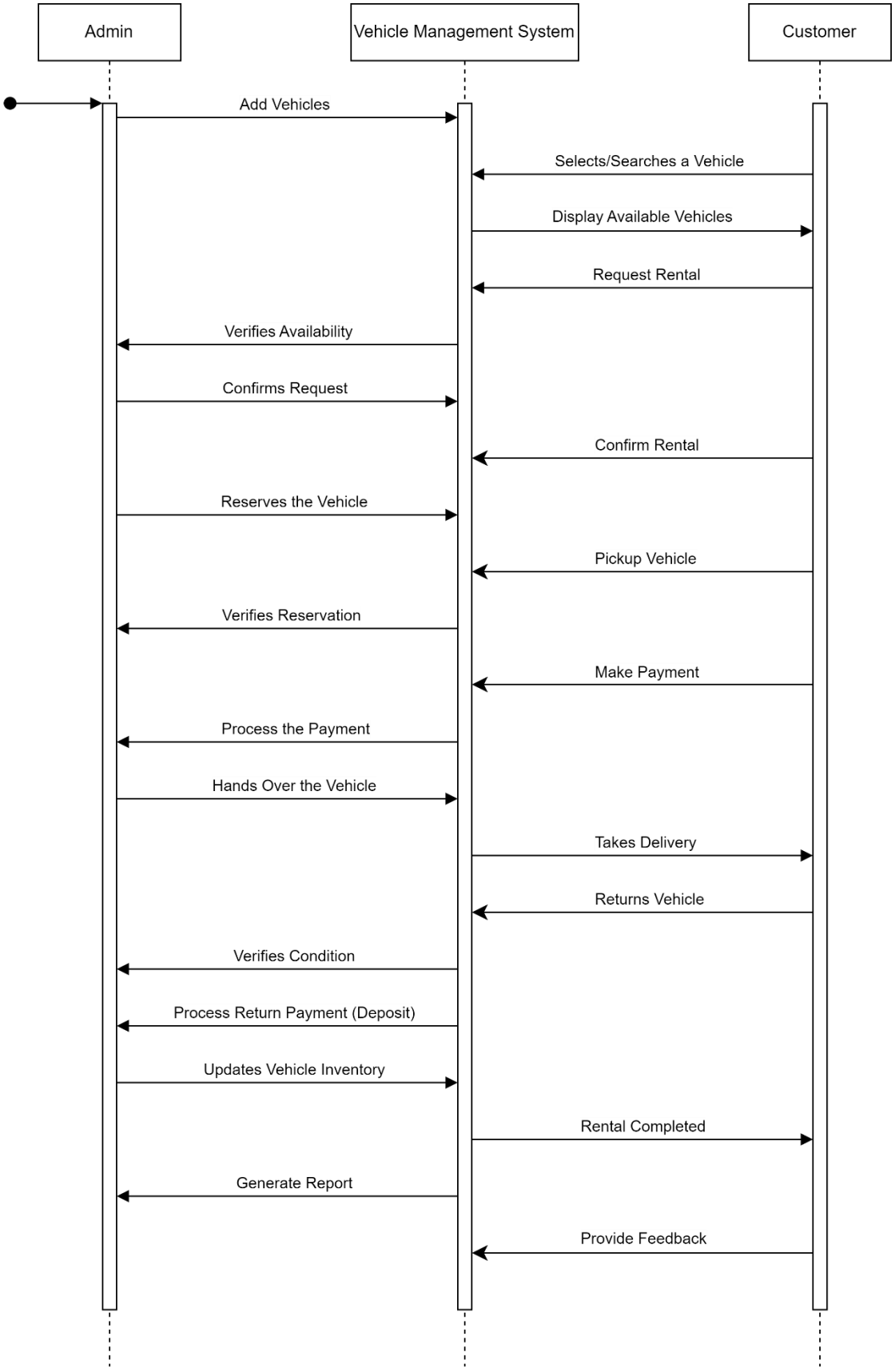
USE CASE DIAGRAM:



SYSTEM FLOW:


CREATE ACCOUNT /REGISTER
LOG IN
MEMBERSHIP
CUSTOMER
ADMIN
CHOOSE A VEHICLE
ADD VEHICLES
RENTAL REQUEST
RENTAL APROOVAL
VERIFICATION
PAYMENT
GENERATE REPORT
HANDING OVER VEHICLE
DELIVERED
RETURN
NO PANELTY
DAMAGED
RETURN DEPOSIT
ADDITIONAL PAYMENT
FEEDBACK

SEQUENCE DIAGRAM:



DATA DICTIONARY

- **Admin**

ATTRIBUTES	DATATYPES	DESCRIPTION	CONSTRAINTS
a_id	int(20)	Admin ID	PRIMARY KEY
a_name	varchar(200)	Admin Name	NOT NULL
a_email	varchar(200)	Admin Email	NOT NULL
a_pwd	varchar(200)	Admin Password	NOT NULL

- **User**

ATTRIBUTES	DATATYPES	DESCRIPTION	CONSTRAINTS
u_id	int(20)	User ID	PRIMARY KEY
u_fname	varchar(200)	User First Name	NOT NULL
u_lname	varchar(200)	User Last Name	NOT NULL
u_phone	varchar(200)	User Phone Number	NOT NULL
u_addr	varchar(200)	User Address	NOT NULL
u_category	varchar(200)	User Category (User/Driver)	NOT NULL
u_email	varchar(200)	User Email	NOT NULL
u_pwd	varchar(20)	User Password	NOT NULL
u_car_type	varchar(200)	Vehicle Type (Bus/Car)	NOT NULL
u_car_regno	varchar(200)	Car Registration Number	NOT NULL
u_car_bookdate	varchar(200)	Date of Booking	NOT NULL
u_car_book_status	varchar(200)	Status (Pending/Approved)	NOT NULL

- **Vehicle**

ATTRIBUTES	DATATYPES	DESCRIPTION	CONSTRAINTS
v_id	int(20)	Vehicle ID	PRIMARY KEY
v_name	varchar(200)	Vehicle Name	NOT NULL
v_reg_no	varchar(200)	Vehicle Registration Number	NOT NULL
v_pass_no	varchar(200)	Vehicle Passenger Capacity	NOT NULL
v_driver	varchar(200)	Vehicle Driver Name	NOT NULL
v_category	varchar(200)	Vehicle Category (Bus/Car)	NOT NULL
v_dpic	varchar(200)	Vehicle Image/Picture	NOT NULL
v_status	varchar(200)	Status (Available or Not Ava.)	NOT NULL

- **Feedback**

ATTRIBUTES	DATATYPES	DESCRIPTION	CONSTRAINTS
f_id	int(20)	Feedback ID	PRIMARY KEY
f_uname	varchar(200)	Feedback Giving Username	NOT NULL
f_content	longtext	Feedback Text/Paragraph	NOT NULL
f_status	varchar(200)	Status (Published/Pending)	NOT NULL

SCREENSHOTS

Landing Page:

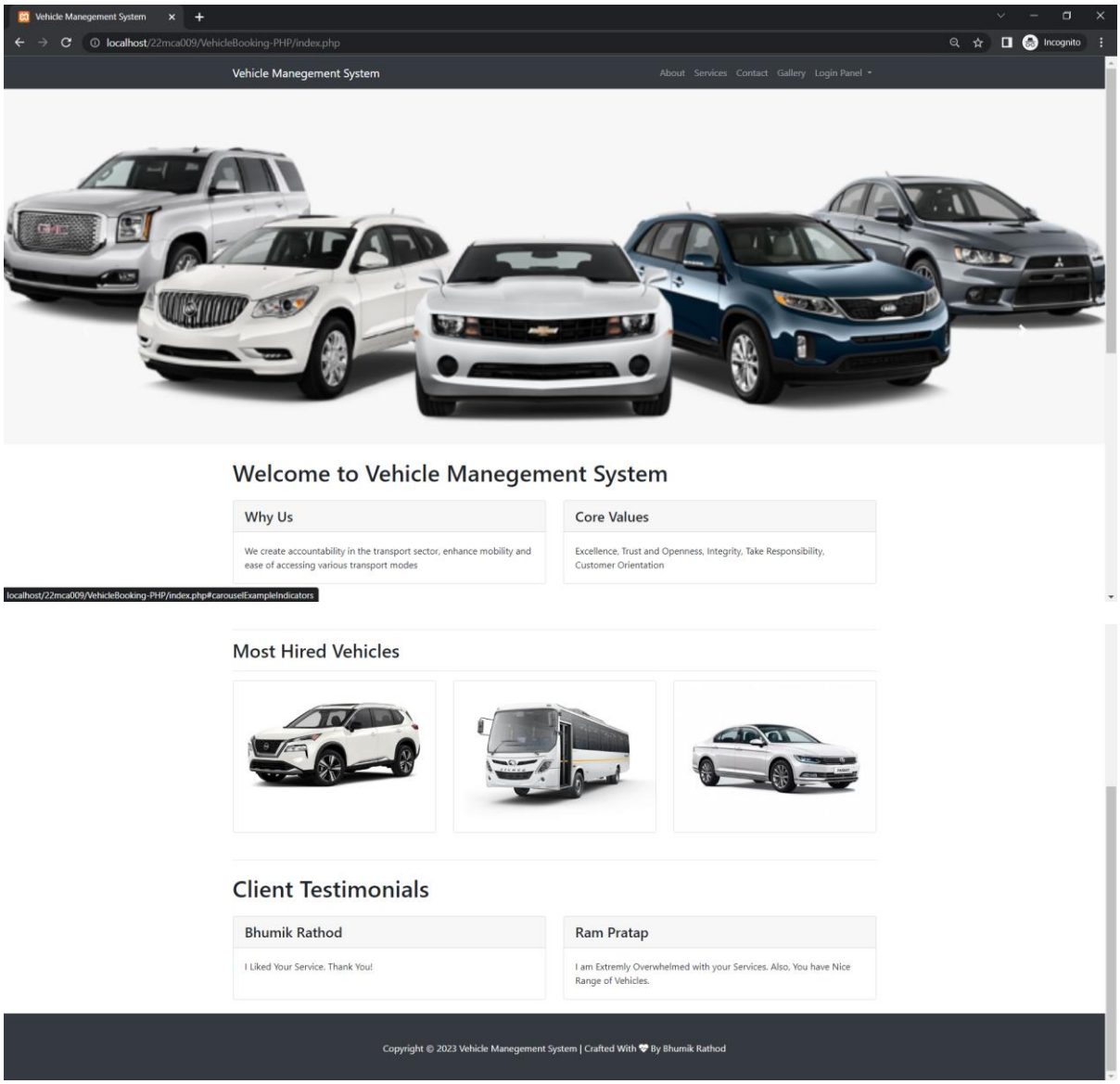


Fig.: 1.0

Description: This is the main/index page or we can say home page of our website.

About Us:

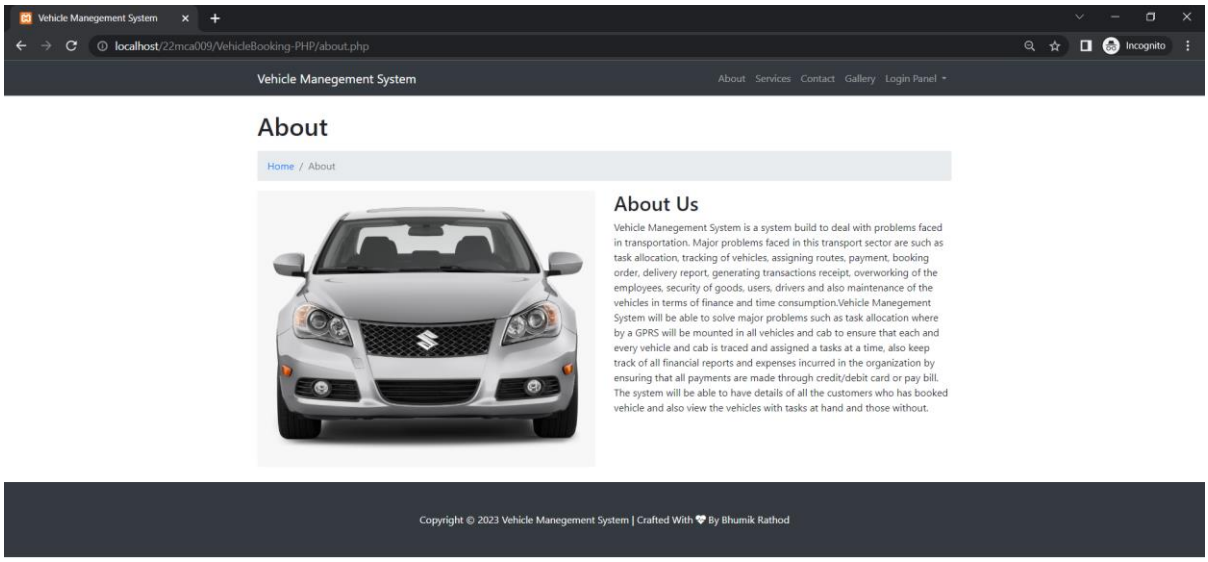


Fig.: 1.1

Services:

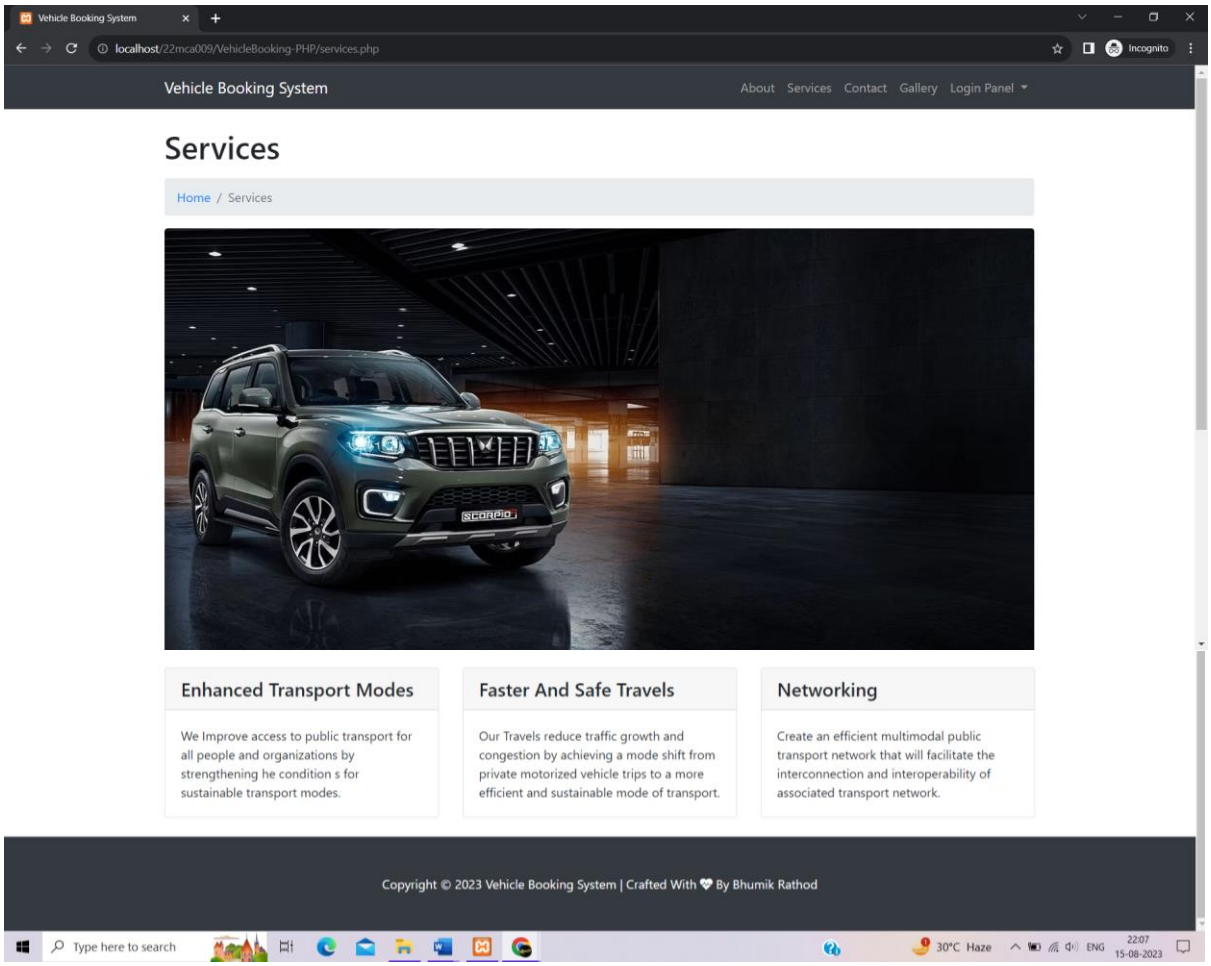


Fig. 1.2

Description: About Us and Services Page Defines, for what & why the VMS is built.

Contact Us:

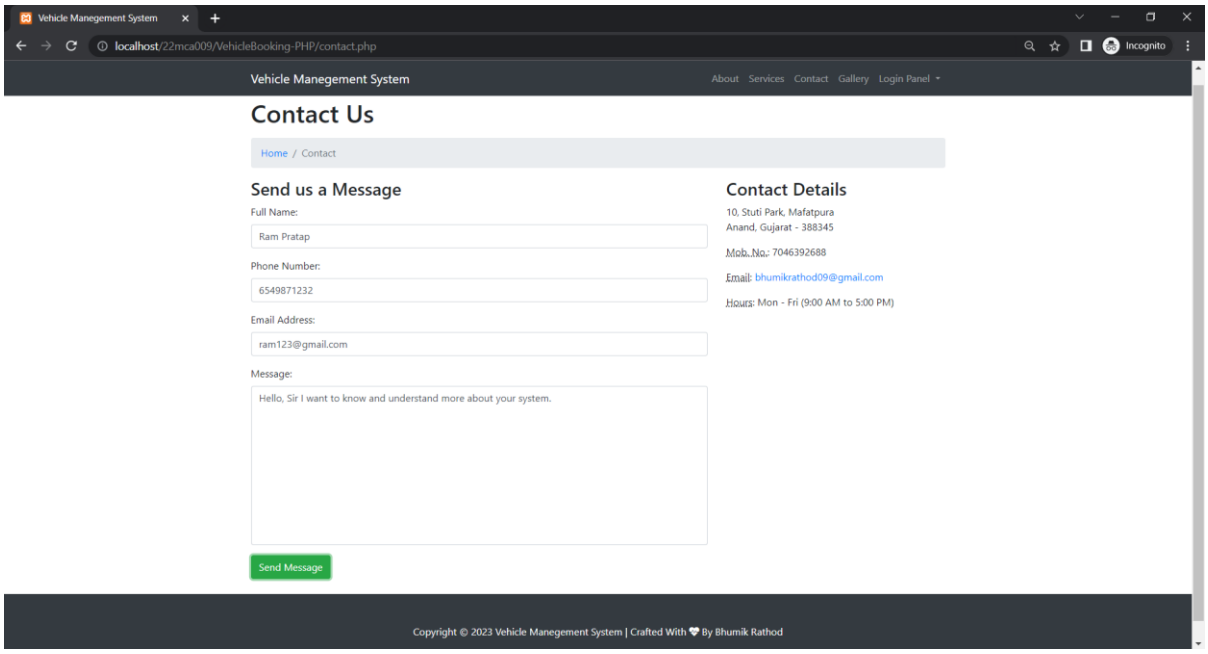


Fig.: 1.3

Description: If user have any query or he/she wants to send a message related to our services then, he/she may reach to us through the help of this page.

Gallery:

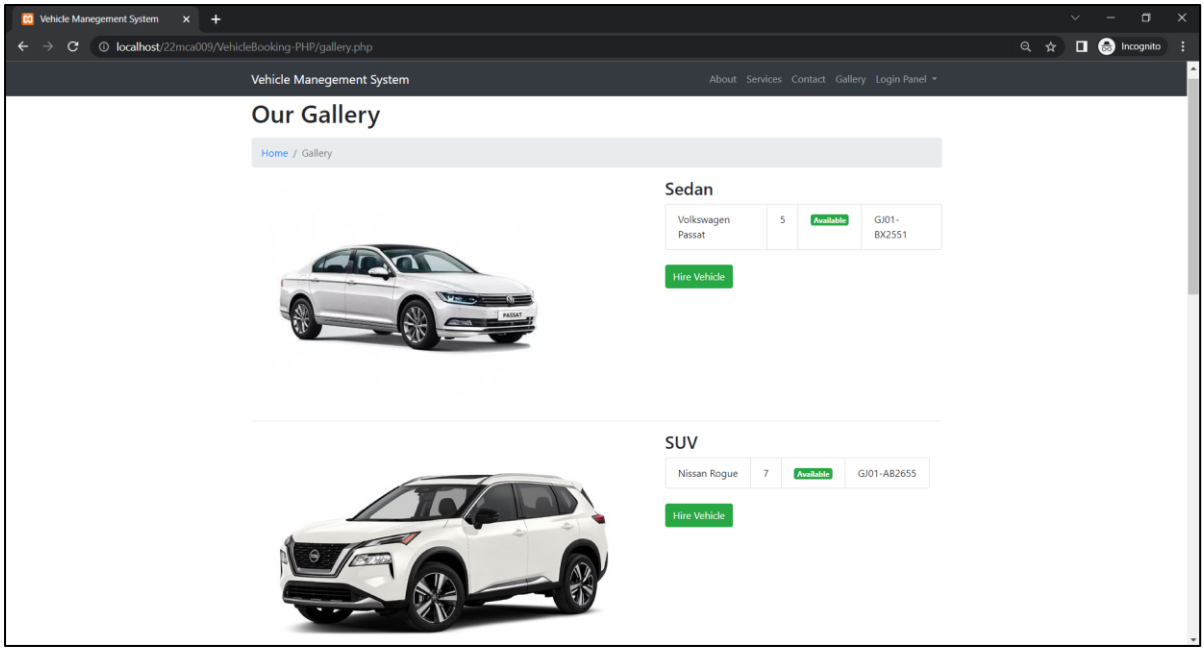


Fig.: 1.4

Description: All the vehicles registered can be seen here on the gallery page.

Admin Login:

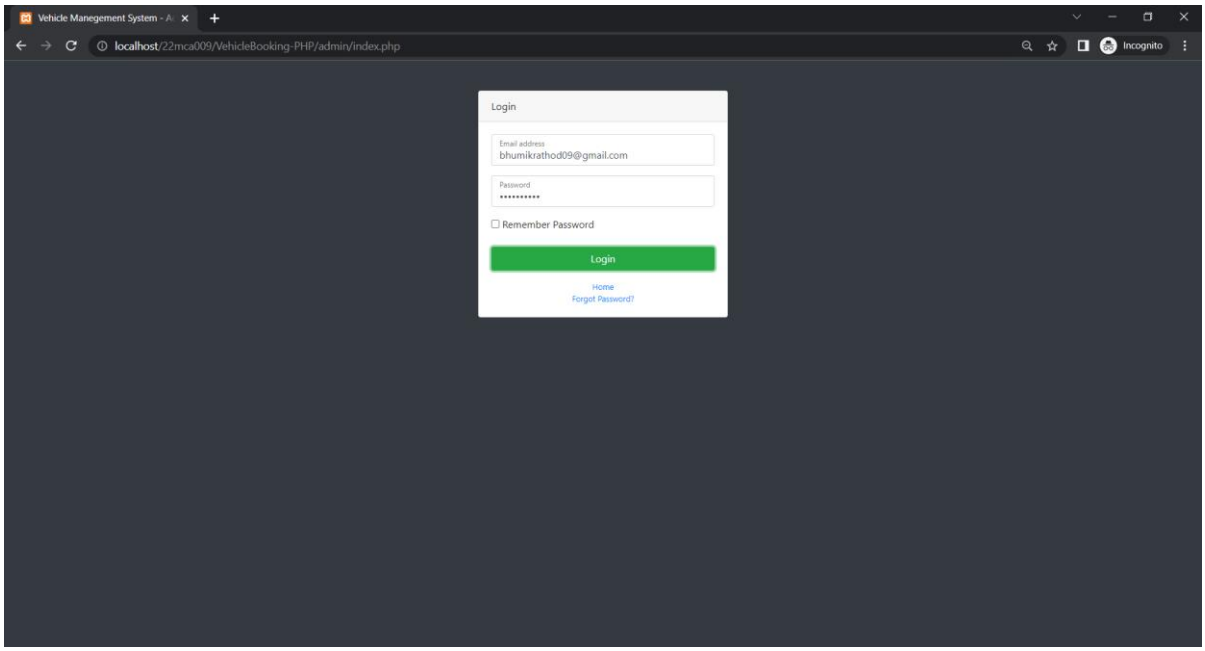


Fig.: 2.0

Description: This is the page from where the admin can login to the system.

Admin Dashboard:

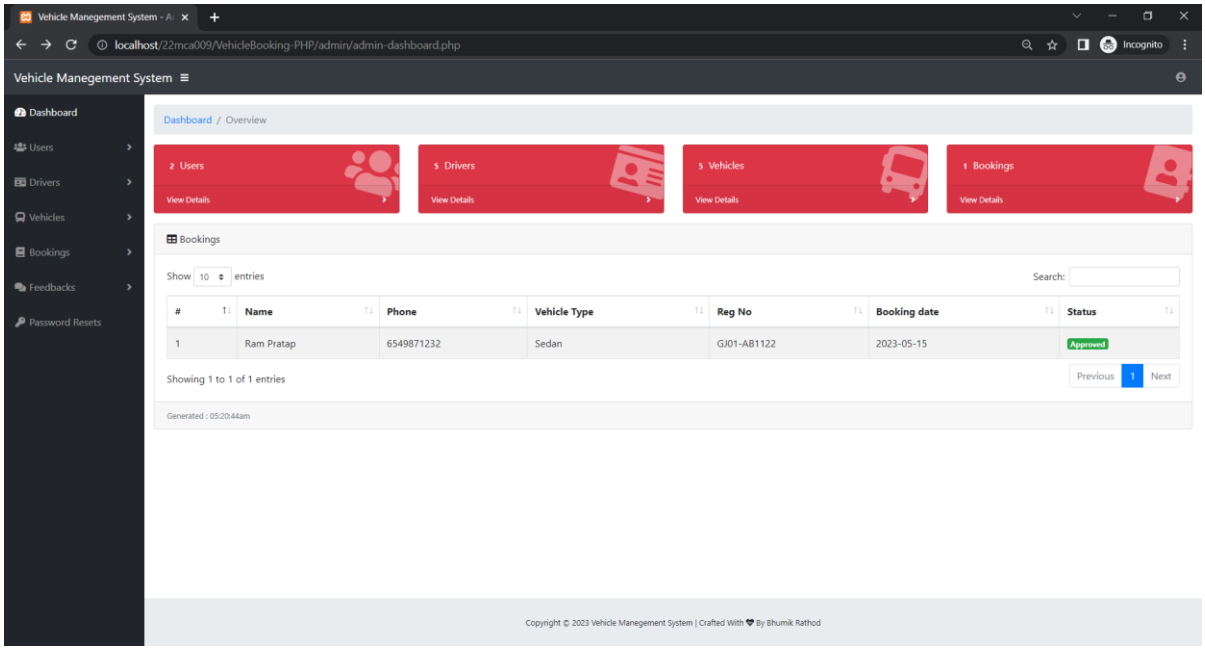


Fig.: 2.1

Description: All the admin operations can be done with the help of this dashboard.

Admin Add Vehicle:

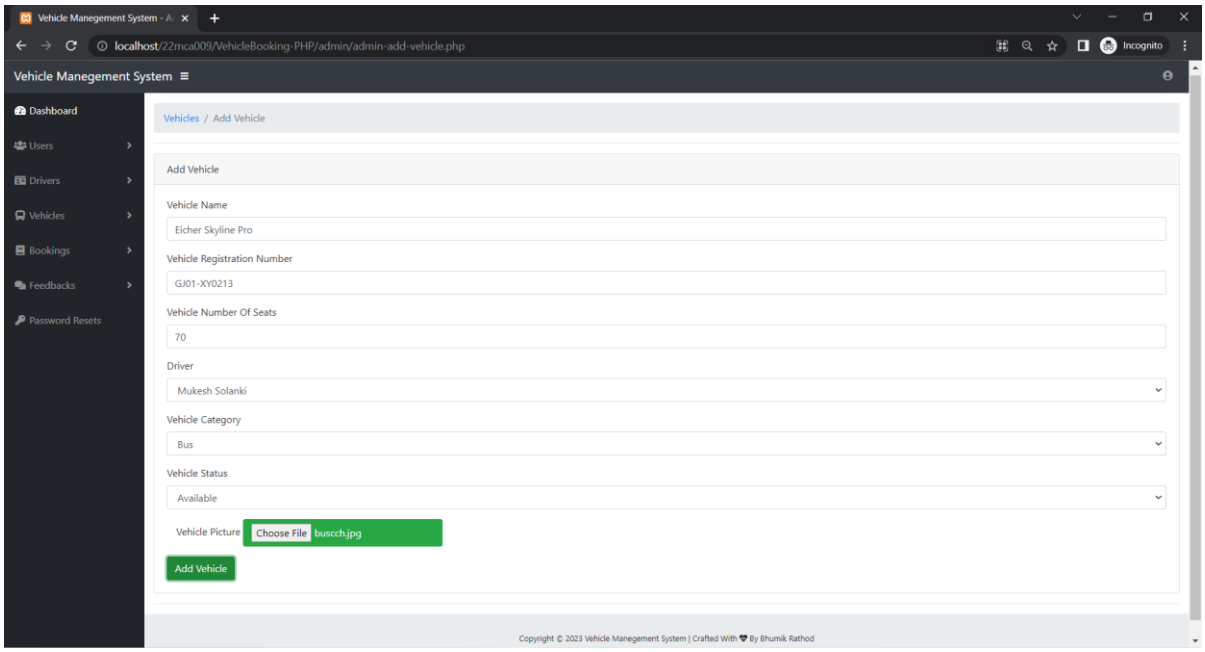


Fig.: 2.2

Description: All the new vehicles can be registered/added from here by the admin.

Admin View vehicle:

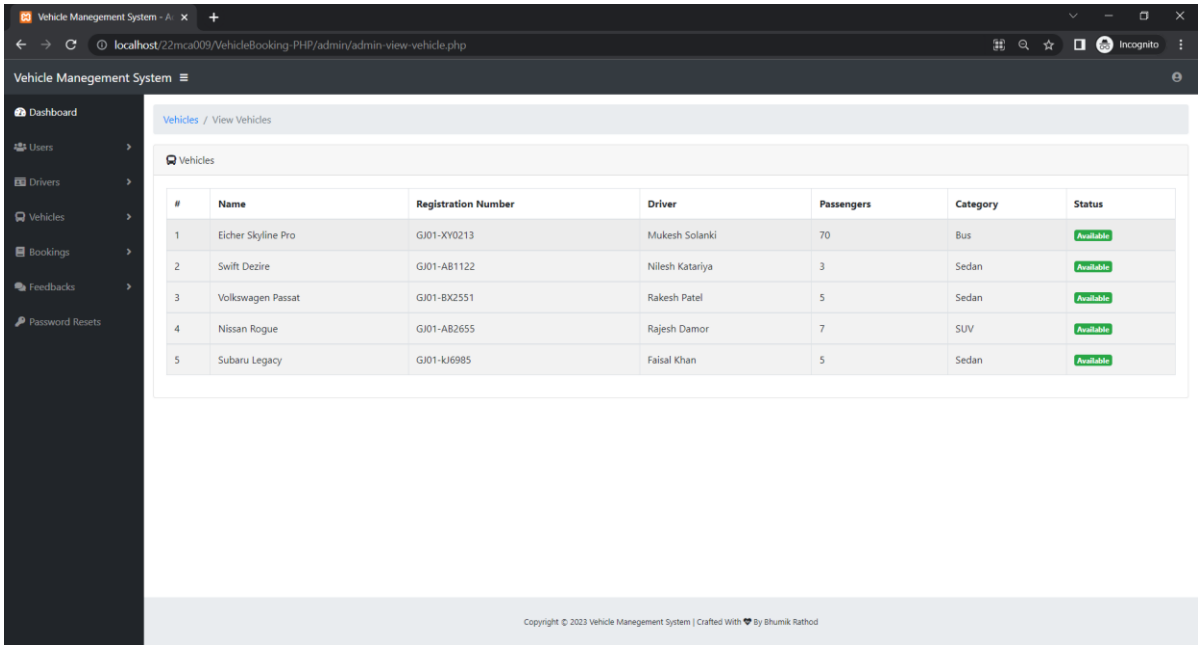


Fig.: 2.3

Description: Admin can look for the vehicle's info. and status from here.

Client Registration:

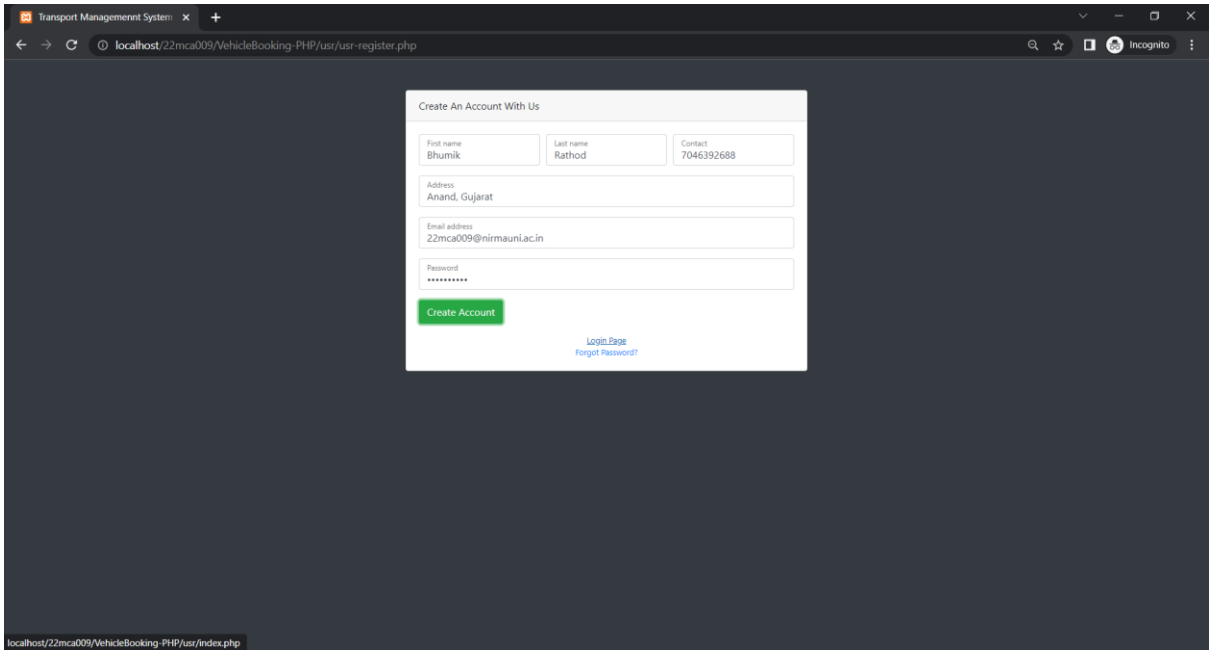


Fig.: 3.0

Description: A new user can register from here by providing the necessary details.

Client Login:

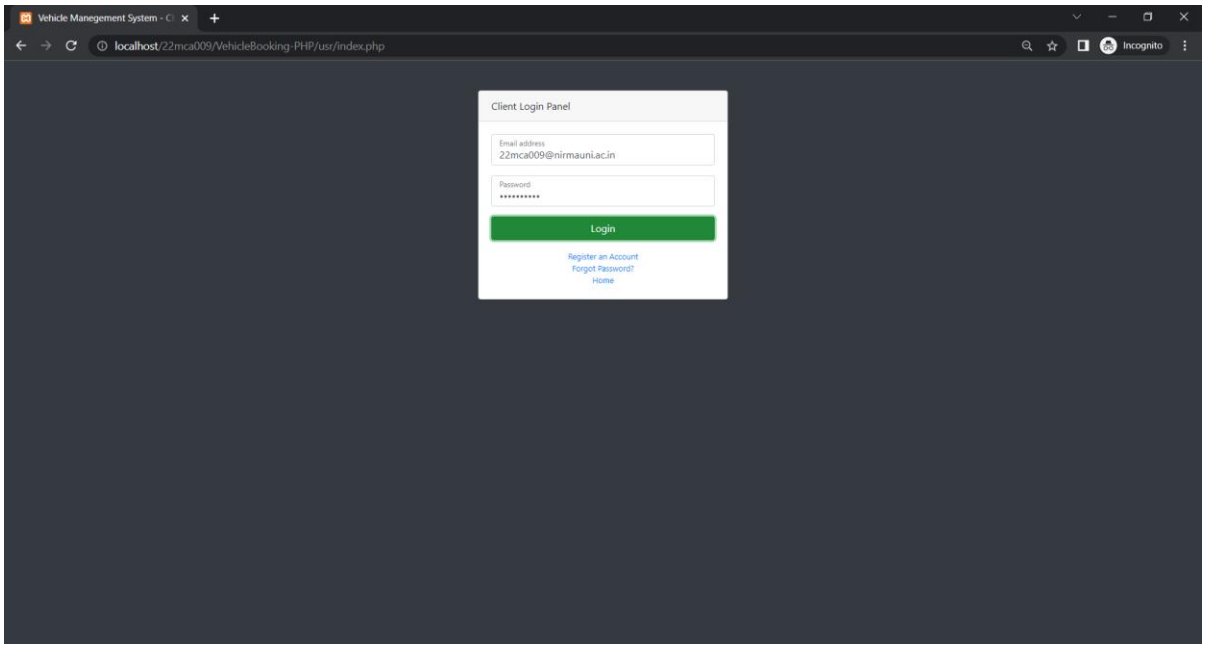


Fig.: 3.1

Description: This is the page from where the user can login to the system.

Client Dashboard:

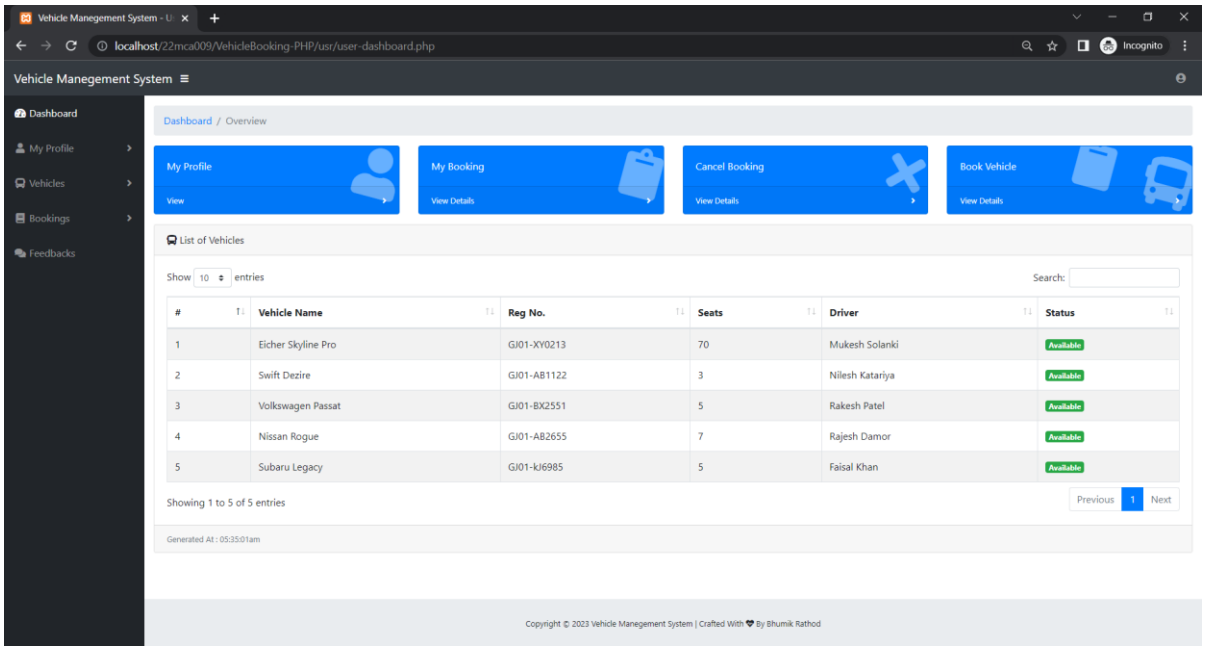


Fig.: 3.2

Description: All the user facilities can be availed through this dashboard's help.

Client Book Vehicle:

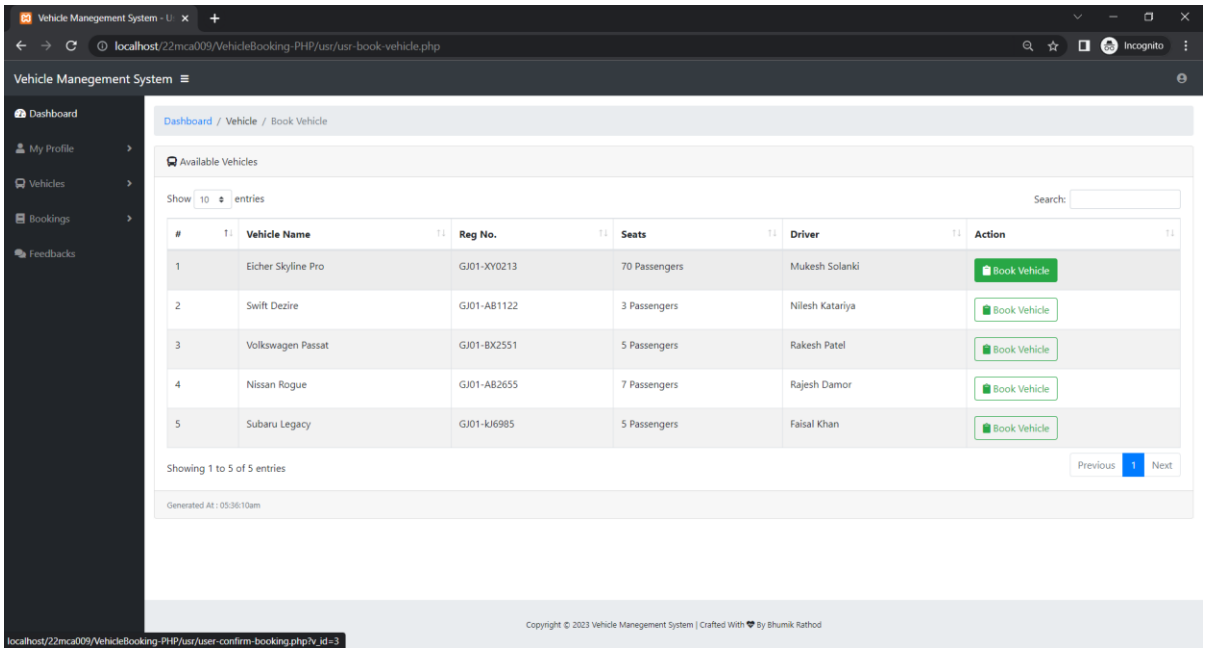


Fig.: 3.3

Description: Any vehicles can be booked from here by the user according to needs.

Admin Approve Booking:

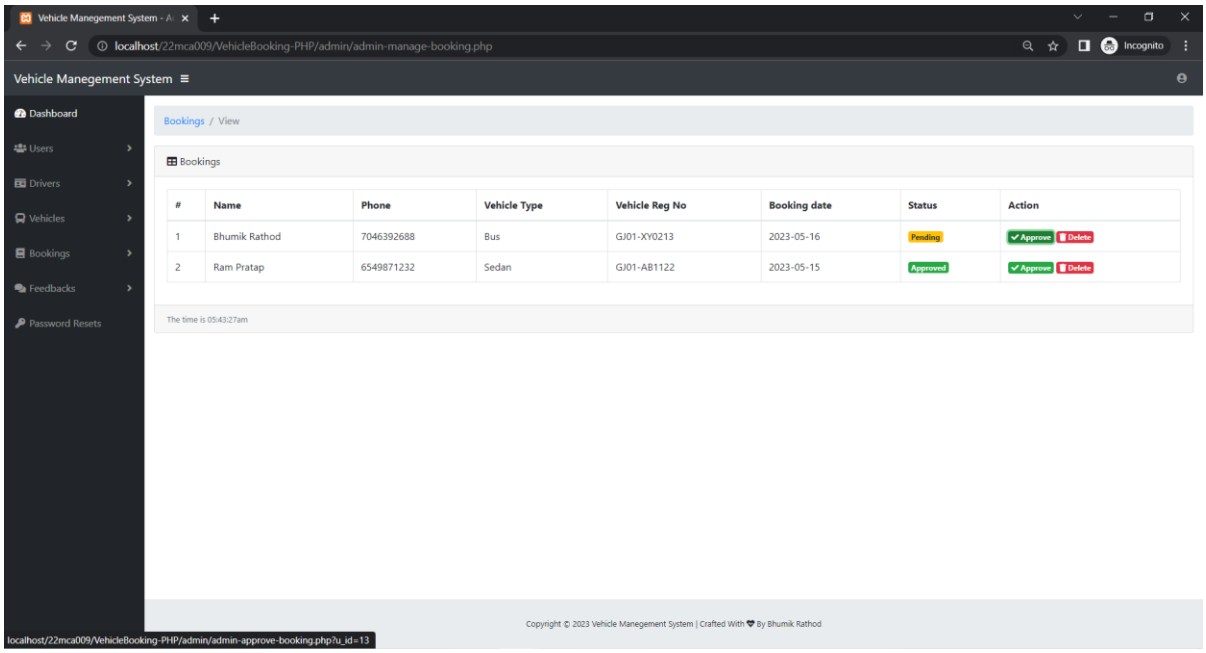


Fig.: 4.0

Description: Admin can manage/take actions on all the bookings from this page.

Booking Approved Message:

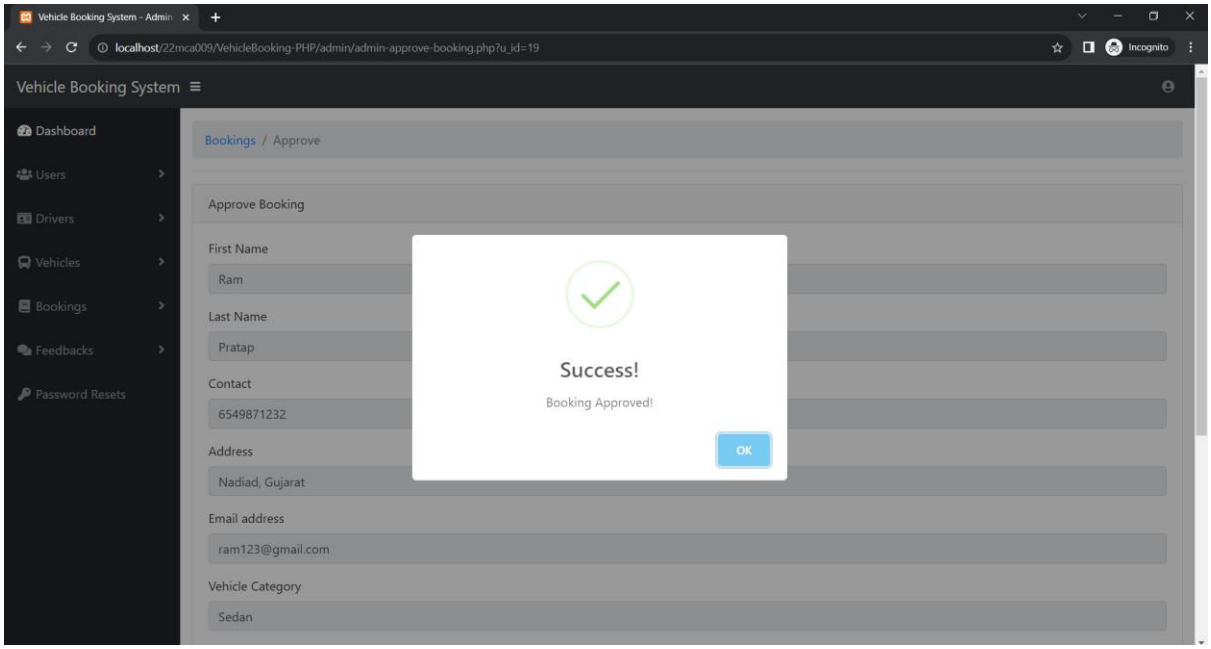


Fig.: 4.1

Description: When admin approves a booking this message box will be displayed.

User Feedback:

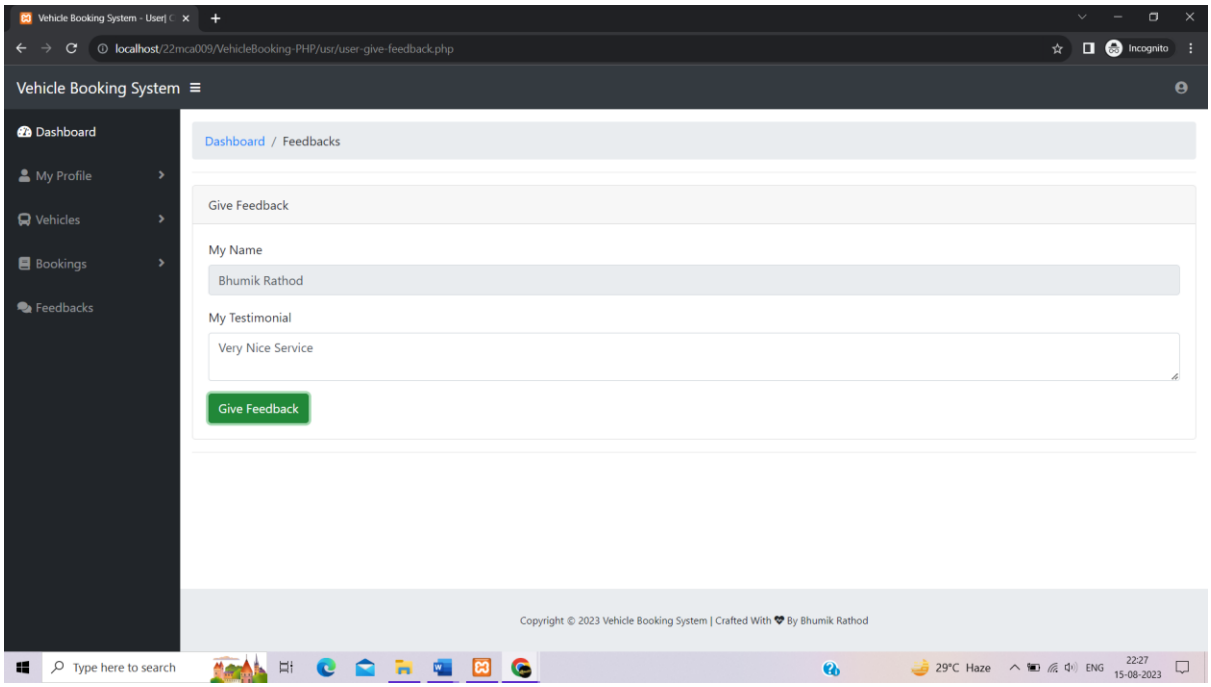


Fig.: 4.2

Description: A user can give is valuable feedback from here, which will help in improving the current system.

Admin Feedback Management:

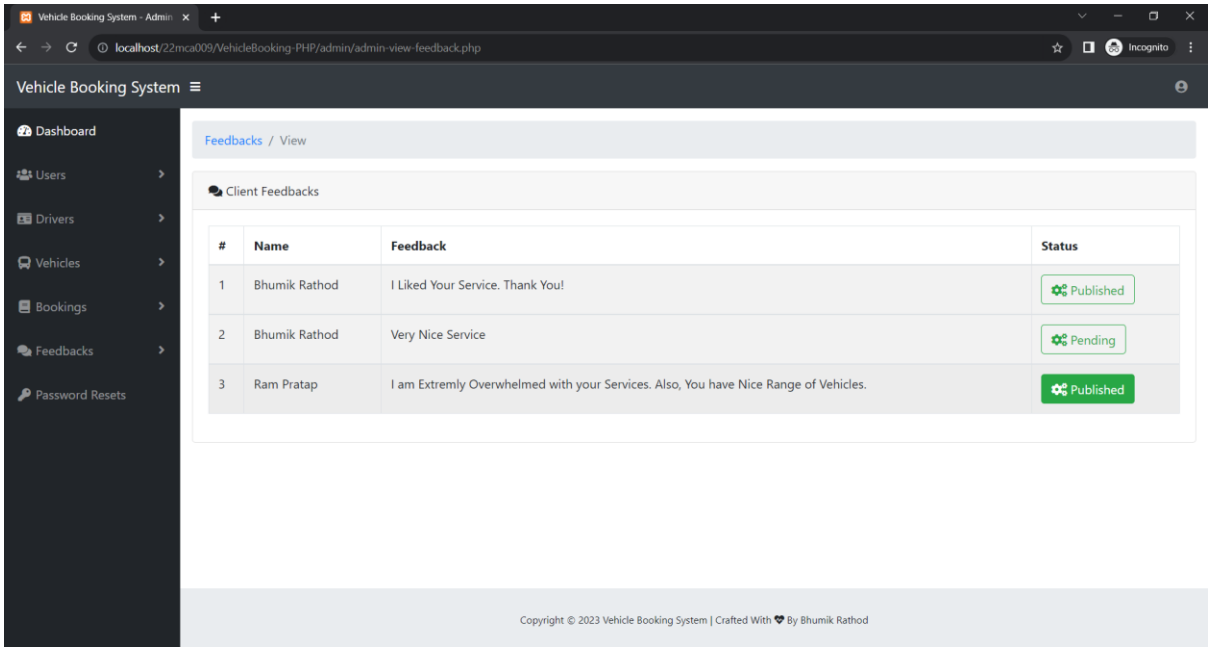


Fig.: 4.3

Description: Admin can manage user's feedback from here according to needs.

TESTING

- **Test Strategy Document:**

1. **Test Objectives:**
   To ensure the proper functioning of the Vehicle Registration module.
   To verify the association of vehicles with specific drivers.

2. **Test Scope:**
   Functional testing of the Vehicle Registration module.
   Integration testing with other related modules (if applicable).

3. **Test Approach:**
   Test cases will be designed based on functional requirements and design specifications.
   Automated tests will be developed to execute the test cases efficiently.
   Both positive and negative test scenarios will be covered.
   Integration testing will be performed with dependent modules.

4. **Test Data:**
   Test data will be prepared to cover various scenarios, including valid and invalid vehicle details, unique vehicle numbers, and driver associations.

5. **Test Execution:**
   5.1. Test Case Execution:
   Execute the prepared test cases using the chosen testing tools.
   Record the test results, including pass/fail status and any defects identified.

   5.2. Defect Tracking:
   Track and manage defects using the chosen defect tracking tool.
   Assign and prioritize defects for resolution.

   5.3. Test Reporting:
   Generate test reports summarizing the test results, including test coverage, pass/fail status, and any open defects.
   Provide relevant metrics and statistics to assess the quality of the Vehicle Registration module.

6. Test Deliverables:

6.1. Test Plan:

Document detailing the overall test strategy, objectives, scope, and approach for the Vehicle Registration module.

6.2. Test Cases:

Detailed test cases covering various scenarios for the Vehicle Registration module.

6.3. Test Reports:

Summary reports providing an overview of test results, including pass/fail status, defect status, and test coverage.

6.4. Defect Reports:

Reports documenting identified defects, including their severity, and priority.

7. Risks and Mitigation:

7.1. Risk: Incomplete or inaccurate requirements may lead to incorrect test coverage.

Mitigation: Review and clarify requirements with stakeholders, ensuring a common understanding before test case design.

7.2. Risk: Insufficient test data coverage may result in incomplete testing.

Mitigation: Prepare a comprehensive test data set covering various scenarios, including boundary cases and invalid inputs.

7.3. Risk: Dependency failures or unavailability of external services may impact test execution.

Mitigation: Create mock services or use stubs to simulate dependencies and ensure test execution is not hindered by external factors.

8. Conclusion:

The test strategy outlined above aims to ensure thorough testing of the Vehicle Registration module, covering its functionalities, integration, and performance. By following this strategy, the team can identify and rectify any issues or defects, ultimately improving the quality and reliability of the Vehicle Management System.

- **Test Cases:**

1.1 Test Case: User Registration

- Test the successful registration of a new user.
    - Input: Registered and Logged in to Add a New User.
    - Output: Successfully Registered, Logged In.

1.2 Test Case: Vehicle Registration

- Test the successful registration of a new vehicle.
    - Input: Registration and Login for a Successful Booking.
    - Output: Successfully Registered, Login and Booked a Vehicle.

1.3 Test Case: Associate Vehicle with Driver

- Test the association of a vehicle with its specific Driver.
    - Input: Vehicle ID and Driver Name.
    - Output: Successful association of the vehicle with its Driver.

BIBLIOGRAPHY

- https://en.wikipedia.org/wiki/Visual_Studio_Code
- https://en.wikipedia.org/wiki/XAMPP
- https://en.wikipedia.org/wiki/HTML
- https://en.wikipedia.org/wiki/CSS
- https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
- https://en.wikipedia.org/wiki/PHP
- https://en.wikipedia.org/wiki/JavaScript
- https://en.wikipedia.org/wiki/MySQL
- https://getbootstrap.com/
- https://www.ibm.com/topics/software-testing