A

MINI PROJECT

ON

VEHICLE SERVICING SYSTEM

SRI BALAJI UNIVERSITY

SCHOOL OF COMPUTER STUDIES

Master of Computer Application.

Submitted By: -

SUMIT KUMAR [MCA2302105]

MOHIT [MCA2302042]

HRUTIK KSHATRIYA[MCA2302088]

AISHWAR BAJPAI [MCA2302159]

UNDER THE GUIDANCE OF

Prof Akash Bhattacharya

CERTIFICATE

Certified that the Project Report entitled **“VEHICLE SERVICING SYSTEM”**, submitted by **SUMIT KUMAR, MOHIT, HRUTIK KSHATRIYA, AISHWAR BAJPAI,** of MCA, is their own work and has been carried out under my supervision. It is recommended that the candidates may now be evaluated for their work by the University.

Name 1: SUMIT KUMAR	**Roll no.:** MCA2302105
Name 2: MOHIT	**Roll no.:** MCA2302042
Name 3: AISHWAR BAJPAI	**Roll no.:** MCA2302159
Name 4: HRUTIK KSHATRIYA	**Roll no.:** MCA2302088

Place –Sri Balaji University, Pune

Date – 8th November, 2024

Prof. Akash Bhattacharyya (Project Guide)

G.Y. Shitole (Principal)

ACKNOWLEDGEMENT

We feel immense pleasure to introduce “VEHICLE SERVICING SYSTEM” as our major project.

I express my sincere thanks to our instructor Prof. Akash Bhattacharyya sir who guided us to the successful completion of this project report. We take this opportunity to express our deep sense of gratitude for their individual guidance, constant encouragement and immense motivation which have sustained our efforts at all stages of this project report. We are grateful and appreciate all the staff members of the School of Computer Application for their cooperation and support.

I extend my sincere thanks to our principal Dr. G.Y. Shitole Sir for his support and for all the facilities provided for the preparation of this project report.

Also, we wish to thank our parents & friends who helped us a lot in collecting data, pictures and continuous help and support.

Finally, we would wish to thank everyone involved in this project time.

Student Name :

Mohit Pathak(MCA2302042)

Sumit Kumar(MCA2302105)

Aishwar Bajpai(MCA2302159)

Hrutik Kshatriya(MCA2302088)

ABSTRACT

The main goal of this project is to make getting your vehicle serviced easier and better. We want to use the internet to create a system that is simple for both people who own vehicles and the places that service them. This way, scheduling, keeping track of, and managing the service for vehicle becomes much simpler.

Right now, when you need to get your vehicle serviced, it can be a bit of a hassle. You might struggle to book appointments, not know how your service is going, or find it hard to locate a good service centre. This project aims to fix these problems. We want to introduce new and helpful features like predicting when your vehicle needs maintenance, keeping digital records, allowing online payments, and making the whole process more engaging for users.

Our system will have different parts to make everything work smoothly. These include signing up users, managing service requests, assigning service providers, tracking services, handling payments, getting feedback, and a control panel for administrators. Each part is designed to make the experience easy for both the people who own the vehicles and the ones who provide the service.

TABLE OF CONTENT

Chapter 1

INTRODUCTION TO THE LANGUAGES USED

INTRODUCTION TO THE LANGUAGES USED

WHY HTML & CSS?

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, whereas CSS provides the layout, for a variety of devices along with graphics and scripting. HTML and CSS are the basics of building any Web pages or Web Applications.

You can learn more in detail as follows:

1.1 HTML

In essence, HTML is used for creating the primary content of a webpage, giving it structure. You start by writing words, then apply tags or elements to these words. The web browser then reads this and can then understand the heading of a page, any paragraphs, and where the page starts and finishes, thus filling your web page with content. HTML is supported by every single browser and is established on pretty much every webpage in existence. You don't need any licenses, you don't need to pay for it, and it can be easy to learn and code. If we can compare a webpage to the human body, then HTML is the bones of the body.

Html stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, which describes the structure of a Web page. HTML consists of a series of elements that tell the browser how to display the content. These elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Developer of HTML: Tim Berners-Lee

1.2 CSS

If HTML is the bones of the body, then CSS is the skin that covers it. It's used for background colour, styling, layout, borders, shadowing – all the essential design bits and bobs that make a webpage look slick and smart.

CSS enables you to distinguish between presentation and content by modifying the design and display of HTML elements.

Presentation and ease of use are a couple of the main things that CSS has brought to web design by translating the way content looks on a webpage and what else goes on it to complement that content.

While frequently used in correlation with HTML, it is independent of it, and can be used with any XML-based markup language.

CSS stands for Cascading Style Sheets. This language describes how HTML elements are to be displayed on screen, paper, or in other media. It saves a lot of work and can also control the layout of multiple web pages all at once. External stylesheets are stored in CSS files with .CSS extension.

Developer of CSS: World Wide Web Consortium (W3C)

1.3 JAVASCRIPT:

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed. Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one.

Apart from the unlimited possibilities, there are many more reasons for web developers to use JavaScript over other programming languages. Which are as follows:

- JavaScript is the only programming language native to the web browser.
- JavaScript is the most popular language.
- There’s a low threshold to get started.
- It’s a fun language to learn.

Developer of JavaScript: Brendan Eich.

1.4 PHP:

The name PHP stands for Hypertext Preprocessor and denotes a server-side scripting language, which means that the applications written on it run on web servers and do not depend on the web browser. However, over the years, its area of use has shifted, and nowadays the PHP coding language is ranked among the best and most popular programming tools for web development due to its different virtues. It is considered as a very effective technology that offers a convenient development process with many additional tools to aid it. In fact, according to the Popularity of Programming Language Index (PYPL), PHP is the fifth most popular coding language in the world.

Another reason for its popularity is its specifications and advantages, which PHP offers such as many available specialists, a large base of reference and educational materials, better loading speed of websites, more options for database connectivity, a large collection of open source addons, inexpensive website hosting, great synergy with HTML, excellent flexibility and combinability, various benefits provided by cloud solutions, and many more.

1.5 MySQL:

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. A relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to implement, manage, and query such a database.

Chapter 2

INTRODUCTION TO “VEHICLE SERVICING SYSTEM”

INTRODUCTION TO “VEHICLE SERVICING SYSTEM

2.1 Introduction:

Imagine a world where getting your vehicle serviced is as easy as a few clicks on your phone. That's what the Vehicle Servicing System is all about. In a time where having a vehicle is essential, this online platform steps in to make the whole servicing process smoother for both people who own vehicles and the places that fix them up.

With more and more people relying on vehicles, we need a better way to keep them in good shape. The Vehicle Servicing System is here to tackle the usual headaches of scheduling appointments, figuring out costs, and finding the right service centres. It brings in cool features like predicting when your vehicle needs attention, keeping all your service records online, allowing you to pay for services online, and making the whole process more enjoyable.

Our goal is to make vehicle servicing easy, efficient, and modern. As technology advances, so should the way we take care of our vehicles. The Vehicle Servicing System is like a friendly companion in this journey, making sure both vehicle owners and service centres have a hassle-free and satisfying experience.

2.2 Purpose & Scope:

Purpose

The Vehicle Servicing System exists to make taking care of your vehicle super easy. It's like a helpful friend that uses the internet to simplify everything – from booking appointments to tracking your vehicle's health. The main goal is to save time, reduce stress, and make sure both vehicle owners and service centres have a happy and smooth experience.

Scope:

The Vehicle Servicing System is all about making life simpler for people who own vehicles and the places that fix them. It helps you easily register your vehicle, keep track of its service history, pay for services online, and get reminders for when it needs attention. The system covers everything from requesting a service to giving feedback, making the whole process smooth and user-friendly.

2.3 Advantages:

Easy Scheduling: You can book your vehicle service appointments effortlessly through the online platform, saving time and avoiding the hassle of in-person or phone scheduling.

Transparency: Know exactly what's happening with your vehicle. Get real-time updates on service status and costs, eliminating surprises and uncertainties.

Wide Service Centre Options: Easily locate authorized and reliable service centres, even in remote areas, ensuring you have options that suit your needs.

Digital Service History: Keep a digital record of all your vehicle services, reducing paperwork and making it simple to track maintenance over time.

Enhanced Service Quality: Provide feedback and ratings, promoting accountability and ensuring that service centres maintain high-quality standards.

Predictive Maintenance: Receive alerts for upcoming maintenance needs based on your vehicle's usage and condition, preventing potential issues.

Warranty Tracking: Easily keep track of your warranty information, ensuring you don't miss out on covered repairs and services.

Efficient Documentation: Streamline the documentation process, reducing inefficiencies for both customers and service centres.

User Engagement: Benefit from promotions, timely service reminders, and an engaging user experience, enhancing overall satisfaction.

Customer support:

Customer service is important as it helps in:

- Responsive Assistance
- Guidance Through Processes
- Issue Resolution
- Clear Communication
- Feedback Channel

2.4 SRS- SOFTWARE REQUIREMENT SPECIFICATION:

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all stakeholders (business, users) needs.

Use: -

The use of VEHICLE SEVICING SYSTEM website is to help the people to book the servicing for their own vehicle on specific time slot and date.

Purpose: -

The Vehicle Servicing System exists to make taking care of your vehicle super easy. It's like a helpful friend that uses the internet to simplify everything – from booking appointments to tracking your vehicle's health. The main goal is to save time, reduce stress, and make sure both vehicle owners and service centres have a happy and smooth experience.

2.5 Functional Requirements: -

Functional requirements for our website explain the features a website needs, how users can access webpages, what happens when users click on certain parts of the webpage and how the website appears in a browser. Some of the functional requirements for the website might include:

- User Registration
- Warranty Tracking
- Service History Access
- Notifications and Reminders
- Real-Time Status Updates
- Cost Estimation
- Service Request Specification
- Appointment Scheduling
- Service Centre Locator
- Admin Control Panel
- Security Measures

Non- Functional Requirements: -

A non-functional requirement defines the quality attribute of a software system. It is a function that helps software operate efficiently. These requirements are not mandatory for a system to have, though they typically increase a software's overall quality, speed and storage capacity. Non-functional requirements include: - security, reliability, usability, performance, etc.

Performance and scalability.

How fast does the system return results? --within few seconds.

How much will this performance change with higher workloads.

Reliability, maintainability, availability.

How often does the system experience critical failures? --no critical failures.

How much time does it take to fix the issue when it arises? --2 to 3 minutes.

How is user availability time compared to downtime? --depends on the internet connection.

Portability and compatibility.

Which hardware, operating systems, and browsers, along with their versions does the software run on? --All browser, any hardware.

Does it conflict with other applications and processes within these environments? – no

2.6 Software Tools

You can use any software to run this website like google chrome, Microsoft edge, Mozilla Firefox, etc. We are using google chrome.

We are working with HTML, CSS, and JavaScript for the front-end.

For back-end support we are using PHP & for data storing use MySQL.

Deployment:

- First, you need to design and develop the front-end of your system using HTML, CSS and JS. You can use frameworks like Bootstrap or Tailwind to make your web pages responsive and attractive. You can also use libraries like jQuery or React to add interactivity and functionality to your web pages. You can refer to some examples of car rental or parking management systems using HTML, CSS and JS in the web search results.
- Second, you need to create and connect to a database using SQL and PHP. You can use MySQL as your database management system and PHP as your server-side scripting language. You can use XAMPP as your local web server to run and test your system. You can refer to some examples of vehicle management systems using SQL and PHP in the web search results.
- Third, you need to deploy your system to a web hosting service or a cloud platform. You can choose from various options such as GitHub Pages, Heroku, AWS, Azure, etc. You need to follow the specific instructions and guidelines for each option to upload your files and configure your settings. You can refer to some tutorials on how to deploy a web application using HTML, CSS, JS, SQL and PHP in the web search results.

Hardware Specification

- Server:
- Client Devices:
- Networking Equipment:
- Diagnostic Tools:
- Display Screens:
- Backup and Storage Devices:
- Mobile Devices:

2.7 FEASIBILITY STUDY

We must examine whether the proposed system can be implemented by considering 5 major viewpoints of feasibility:

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility
- Legal Feasibility
- Scheduling Feasibility
- Workflow

Let's study in detail:

Economic Feasibility:

Define the scope of your vehicle service system project, including the features and functionalities it will offer. Outline the specific requirements, such as the technology stack, hardware, software, and any additional tools needed but all tools are available. Estimate the costs associated with software development, including programming, testing, and debugging. Consider the costs of any third-party tools, frameworks, or software licenses that may be required. Total cost Approx 3000 Rupees (Minimum cost approx.) lower when compared to similar systems investigated, that cost at least $100 (approx. Rs.8,500/-)The final system meets the initial goals of Vehicle service , portability and cost-effectiveness.

Technical Feasibility:

In Technical Feasibility current resources both hardware software along with required technology are analysed/assessed to develop project.

It is the availability of technical expertise to run the proposed system, the capability of proposed equipment, and the performance results of the proposed system.

The project can run on any operating system and can also be operated on any low-end systems.

System Requirements:

The system requires a web-based platform accessible through browsers for both service centres and customers.

It needs to support database management for storing customer information, vehicle details, and service history.

Technology Stack:

Web Development: HTML, CSS

Frontend Framework: JavaScript

Backend Framework: PHP

Database: My SQL

Operational feasibility:

In Operational Feasibility degree of providing service to requirements is analysed along with how much easy product will be to operate and maintenance after deployment.

The impact of operational performances particularly serving the customers. Basically, the amount of support for which both admin and users are committed too. In short, it is the measure of how well a proposed system solves the problem.

Through the project, students will guide great manner.

The project is operationally feasible.

User Acceptance:

Objective: Ensure that the system is user-friendly and meets the needs of both service centre staff and customers.

Impact on Current Processes:

Objective: Evaluate how the system will affect existing vehicle servicing processes and workflows.

Training Requirements:

Objective: Determine the level of training required for service center staff to effectively use the new system.

Resource Availability:

Objective: Ensure that the necessary resources, including hardware, software, and skilled personnel, are available.

Cost-Benefit Analysis:

Objective: Evaluate whether the benefits derived from the new system outweigh the costs associated with its implementation.

Scheduling Feasibility:

In scheduling Feasibility is to mention time taken to do each phase for designing the system.

Requirements Gathering and Analysis:

Team A (Sumit, Mohit, Hrutik, Aishwar) working on Requirements gathering and Analysis.

Time Estimation for Requirements Gathering and Analysis 1 month (30 days).

System Design:

Team B (Sumit, Mohit, Hrutik, Aishwar)

Time Estimate for System Design is about 1 month and 2 weeks (45 days).

Development:

Team C (Sumit, Mohit, Hrutik, Aishwar)

Time Estimate for Development is 3 months (90 days).

Testing:

Team C (Sumit, Mohit, Hrutik, Aishwar) is working on testing.

Time Estimate for testing is 1 month (30 days).

Deployment:

Team A (Sumit, Mohit, Hrutik, Aishwar) working on deployment.

Time Estimate for deployment is 1 month (30 days).

Post-Deployment Support and Maintenance:

Team D (Sumit, Mohit, Hrutik, Aishwar) working for this.

Time Estimate: Ongoing

Total Estimated Time:

The total estimated time for developing a vehicle servicing system, including all phases, could range from approximately 8 months, considering the provided estimates for each phase.

2.8 Workflow:

Team A:

Responsibilities:

Requirement Gathering and Analysis.

Deployment.

Oversee the entire project and ensure it aligns with the organization's goals.

Coordinate communication between team members and stakeholders.

Monitor project progress and address any roadblocks.

Final decision-maker for key project-related matters.

Team B:

Responsibilities:

System Design.

Implement the frontend and backend components of the Vehicle Servicing System.

Write code, conduct unit testing, and address any technical issues.

Work closely with Team Lead to provide progress updates.

Team C:

Responsibilities:

Development.

Testing.

Develop and execute test plans to ensure the system functions correctly.

Identify and report bugs or issues, collaborating with the Developer to resolve them.

Conduct performance testing to ensure the system meets specified requirements.

Provide feedback on system usability and overall quality.

Team D:

Responsibilities:

Post-Deployment Support and Maintenance.

Design and implement the database architecture for storing customer information, service records, etc.

Optimize database queries for efficient data retrieval.

Ensure data security and integrity.

Collaborate with the Developer to address any database-related issues.

Deployment:

First, you need to design and develop the front-end of your system using HTML, CSS and JS. You can use frameworks like Bootstrap or Tailwind to make your web pages responsive and attractive. You can also use libraries like jQuery or React to add interactivity and functionality to your web pages. You can refer to some examples of car rental or parking management systems using HTML, CSS and JS in the web search results.

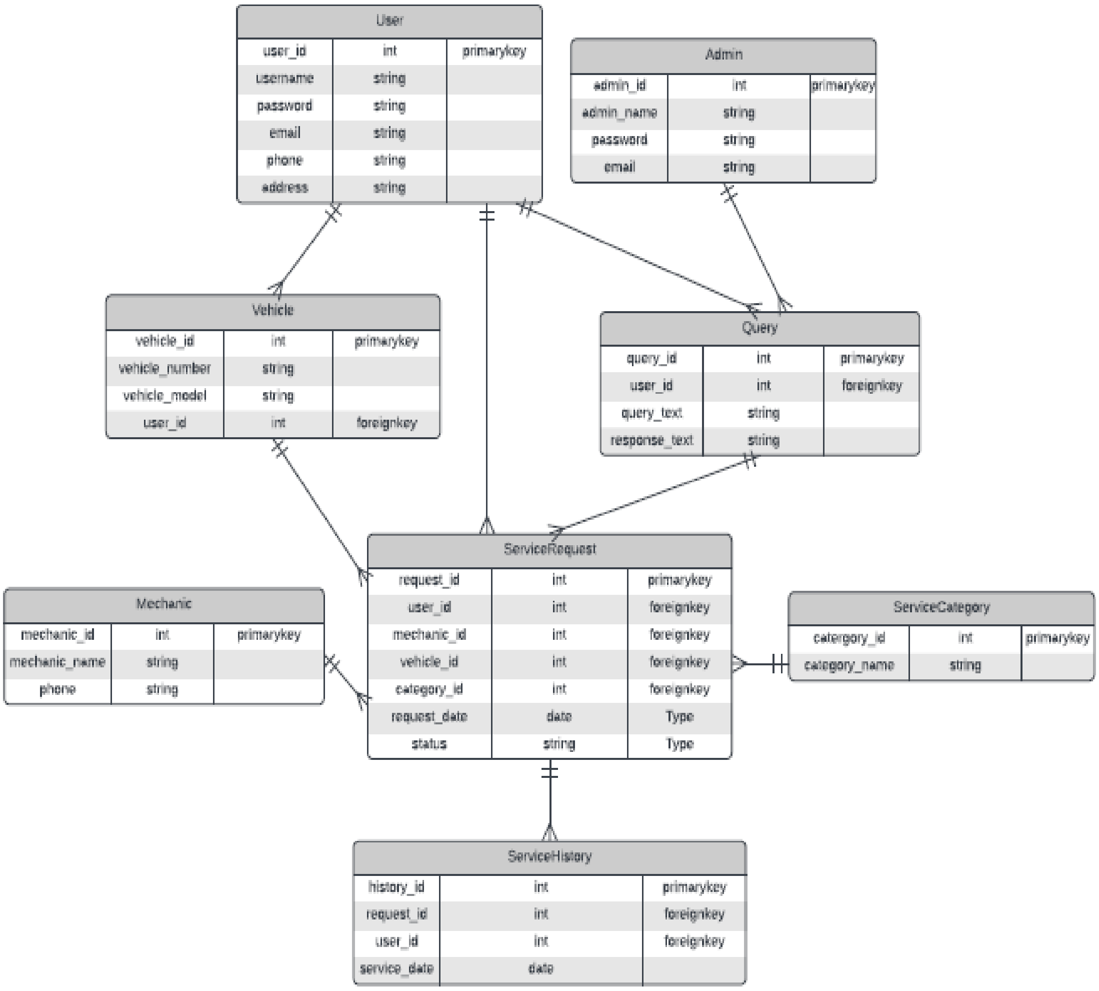
Second, you need to create and connect to a database using SQL and PHP. You can use MySQL as your database management system and PHP as your server-side scripting language. You can use XAMPP as your local web server to run and test your system. You can refer to some examples of vehicle management systems using SQL and PHP in the web search results.

Third, you need to deploy your system to a web hosting service or a cloud platform. You can choose from various options such as GitHub Pages, Heroku, AWS, Azure, etc. You need to follow the specific instructions and guidelines for each option to upload your files and configure your settings. You can refer to some tutorials on how to deploy a web application using HTML, CSS, JS, SQL and PHP in the web search results.

CHAPTER 3

VARIOUS DIAGRAMS

3.1 DATABASE ER DIAGRAM



User		
user_id	int	primarykey
username	string	
password	string	
email	string	
phone	string	
address	string	

Admin		
admin_id	int	primarykey
admin_name	string	
password	string	
email	string	

Vehicle		
vehicle_id	int	primarykey
vehicle_number	string	
vehicle_model	string	
user_id	int	foreignkey

Query		
query_id	int	primarykey
user_id	int	foreignkey
query_text	string	
response_text	string	

ServiceRequest		
request_id	int	primarykey
user_id	int	foreignkey
mechanic_id	int	foreignkey
vehicle_id	int	foreignkey
category_id	int	foreignkey
request_date	date	Type
status	string	Type

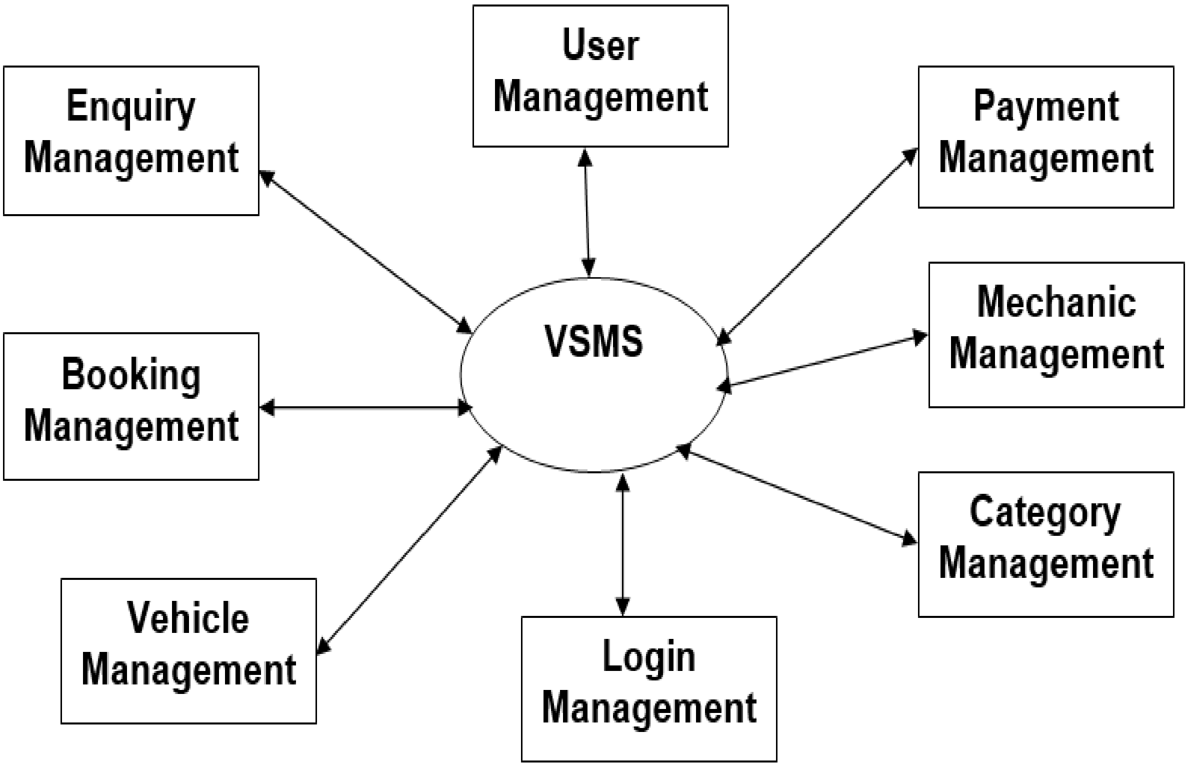
Mechanic		
mechanic_id	int	primarykey
mechanic_name	string	
phone	string	

ServiceCategory		
catergory_id	int	primarykey
category_name	string	

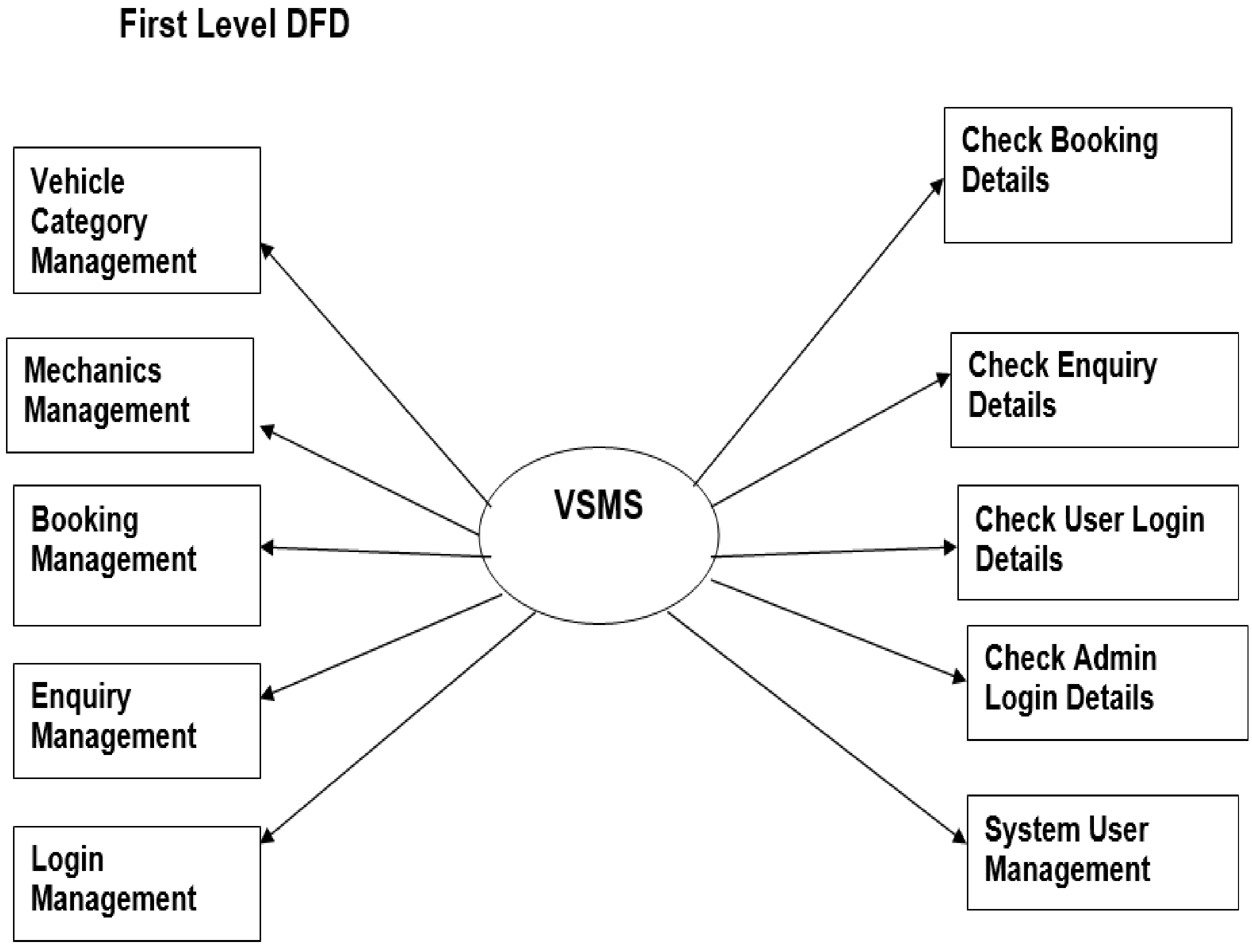
ServiceHistory		
history_id	int	primarykey
request_id	int	foreignkey
user_id	int	foreignkey
service_date	date	

3.2 DATA FLOW DIAGRAM (LEVEL-0)

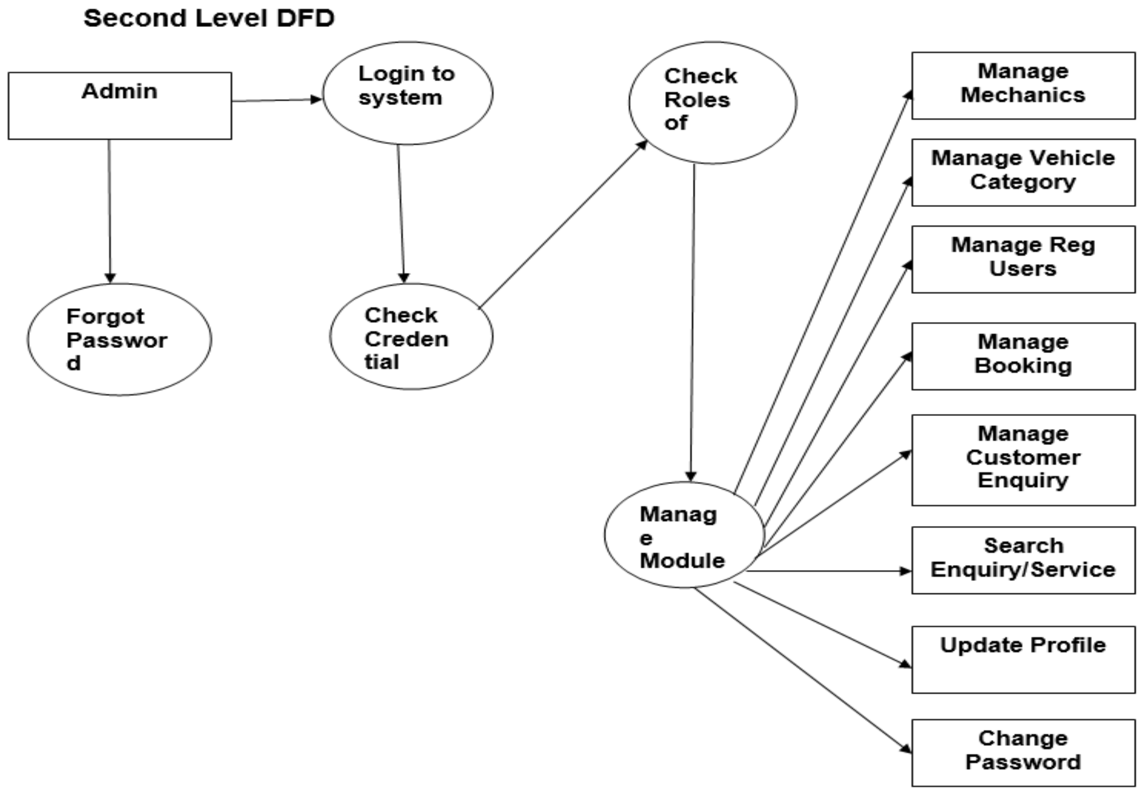
Zero Level DFD

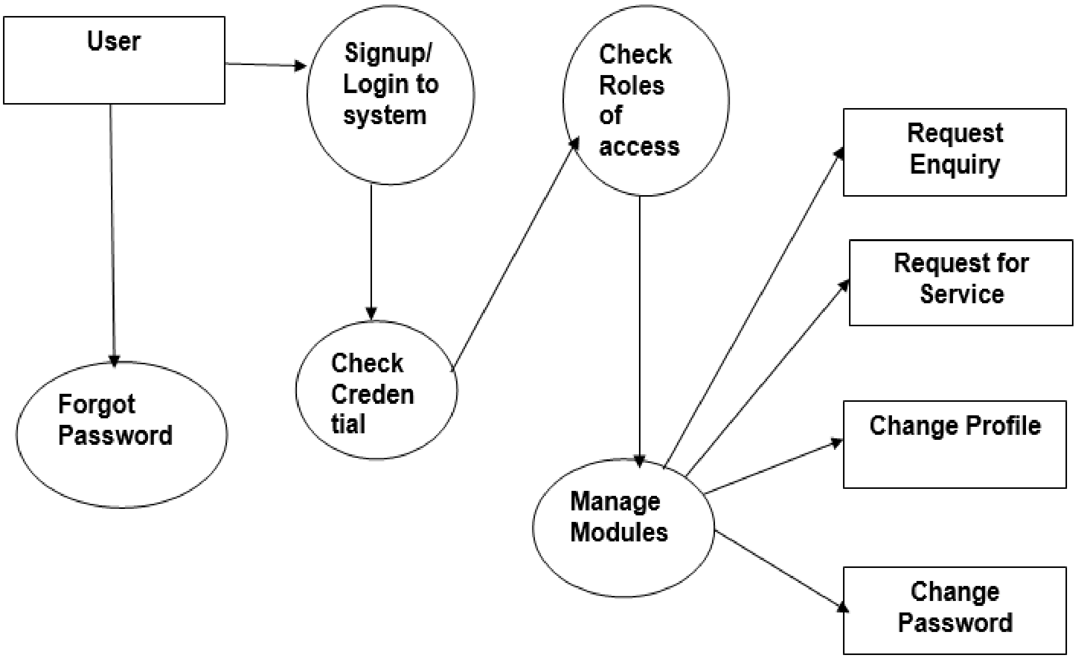


3.3 DATA FLOW DIAGRAM (LEVEL -1)

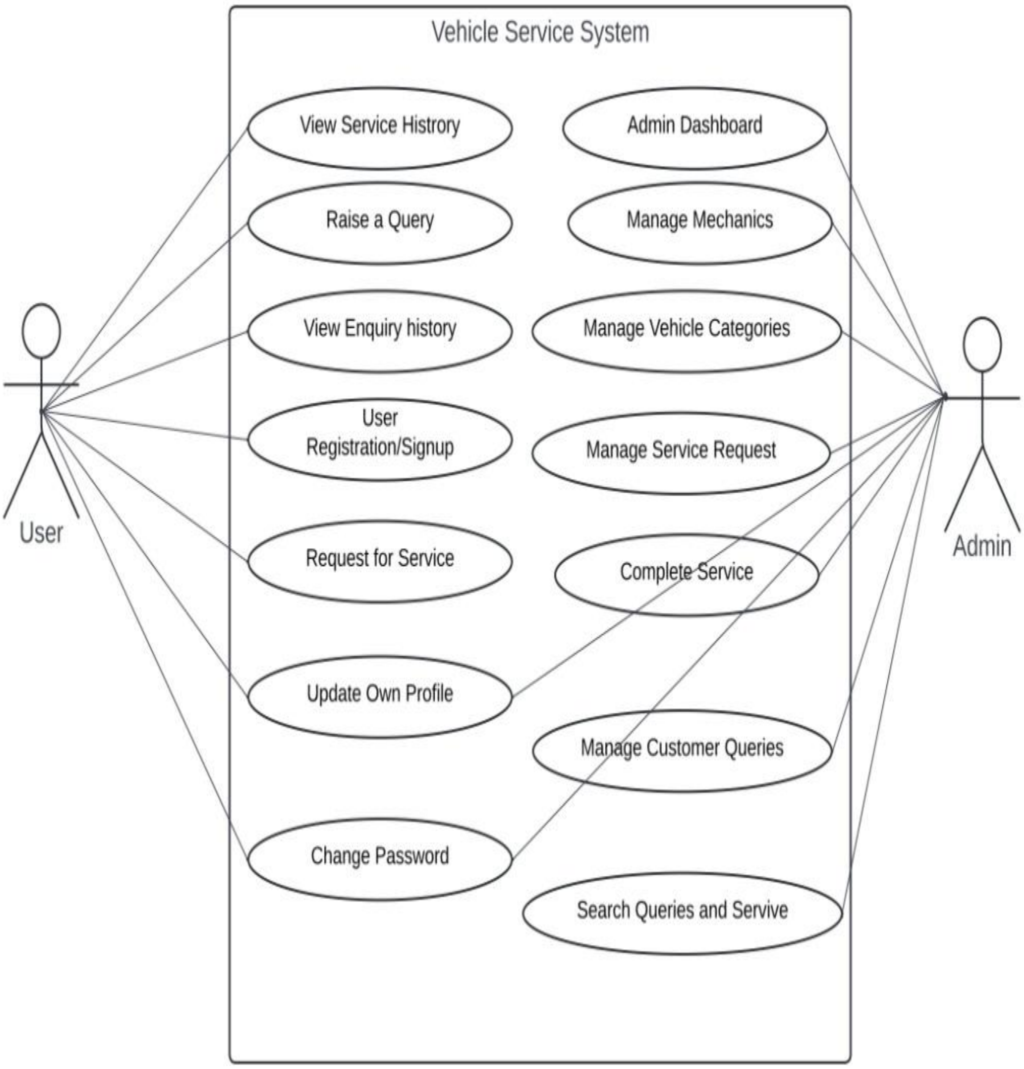


3.4 DATA FLOW DIAGRAM (LEVEL-2)

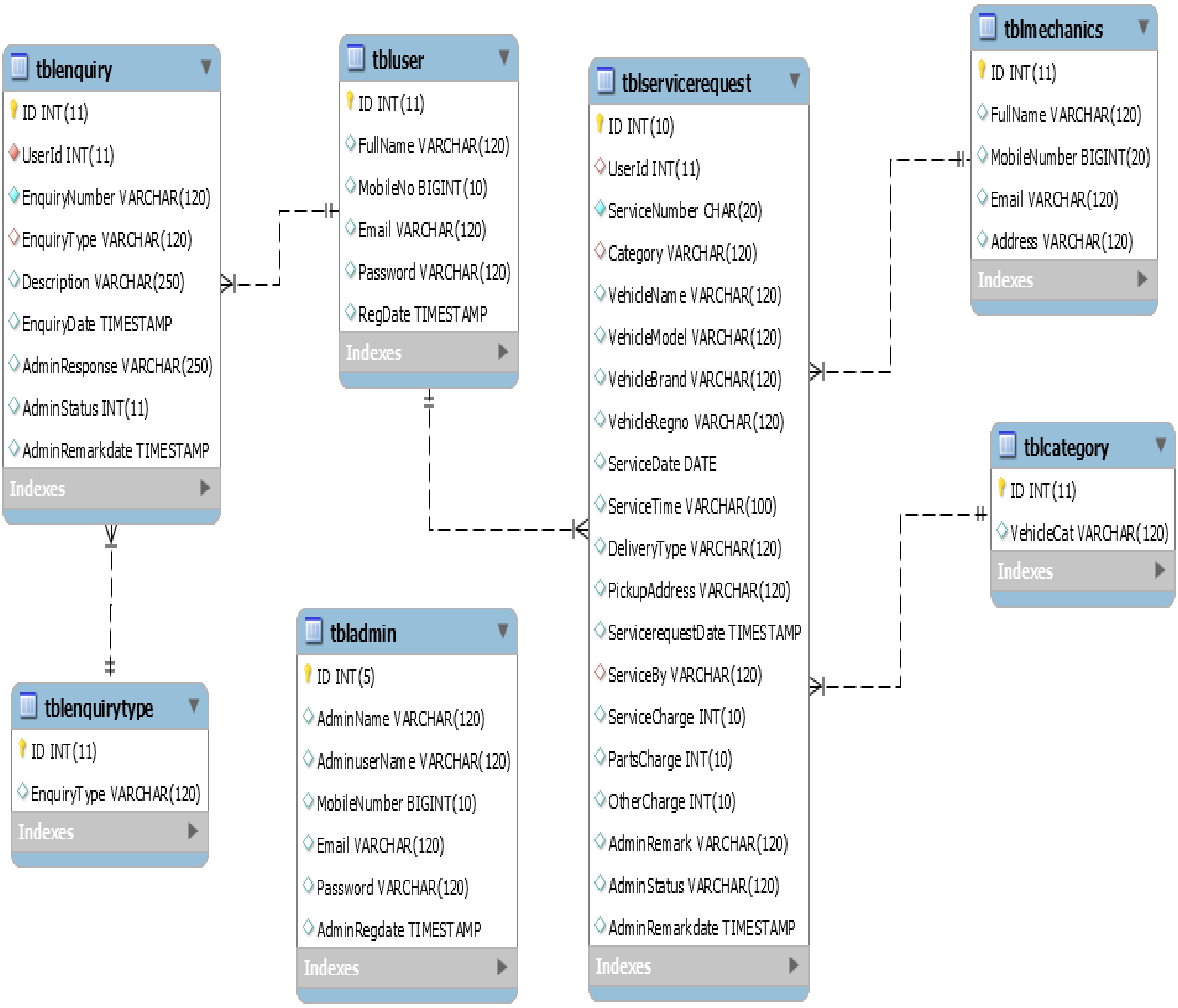




3.5 USE CASE DIAGRAM



3.6 CLASS DIAGRAM



CHAPTER -4

DATA MODEL AND USER INTERFACE

4.1 DATA MODEL

The data model for the Vehicle Servicing System consists of several entities that interact with one another to ensure a smooth and efficient process of vehicle servicing. Below is a theoretical description of the key entities and their relationships.

1. **User**: The **User** entity stores the details of individuals interacting with the system, which could be customers, administrators, or service providers. Each user has a unique identifier (UserID) and relevant information such as full name, email, phone number, and address. Depending on the role (e.g., Customer, Admin, Service Provider), different access privileges are granted.
2. **Vehicle**: Each customer (user) can own one or more vehicles. The **Vehicle** entity contains details such as the vehicle's type (e.g., car, motorcycle), model, registration number, and purchase date. It is linked to a specific user through the UserID and stores important information about the vehicle's warranty and service history.
3. **Service Request**: A **Service Request** represents a customer's request for a vehicle service. It includes the type of service (e.g., maintenance, repair), the date the service is requested, and the expected service date. The system tracks the status of each request, whether it is pending, in progress, or completed. Cost estimates for the service are also recorded, giving users a transparent view of the expenses involved.
4. **Service Provider**: The **Service Provider** entity represents the businesses or individuals offering vehicle services. Each service provider has a unique ProviderID and offers different service types. This entity stores the service provider's contact details, location, and ratings based on customer feedback, making it easy for users to select reliable service centres.
5. **Service History**: The **Service History** entity records the details of each completed service for a vehicle. This includes the service date, provider details, service costs, and feedback from the customer. Maintaining a comprehensive service history helps users track the performance of their vehicles and ensures that all necessary repairs and maintenance have been performed.

4.2 USER INTERFACE

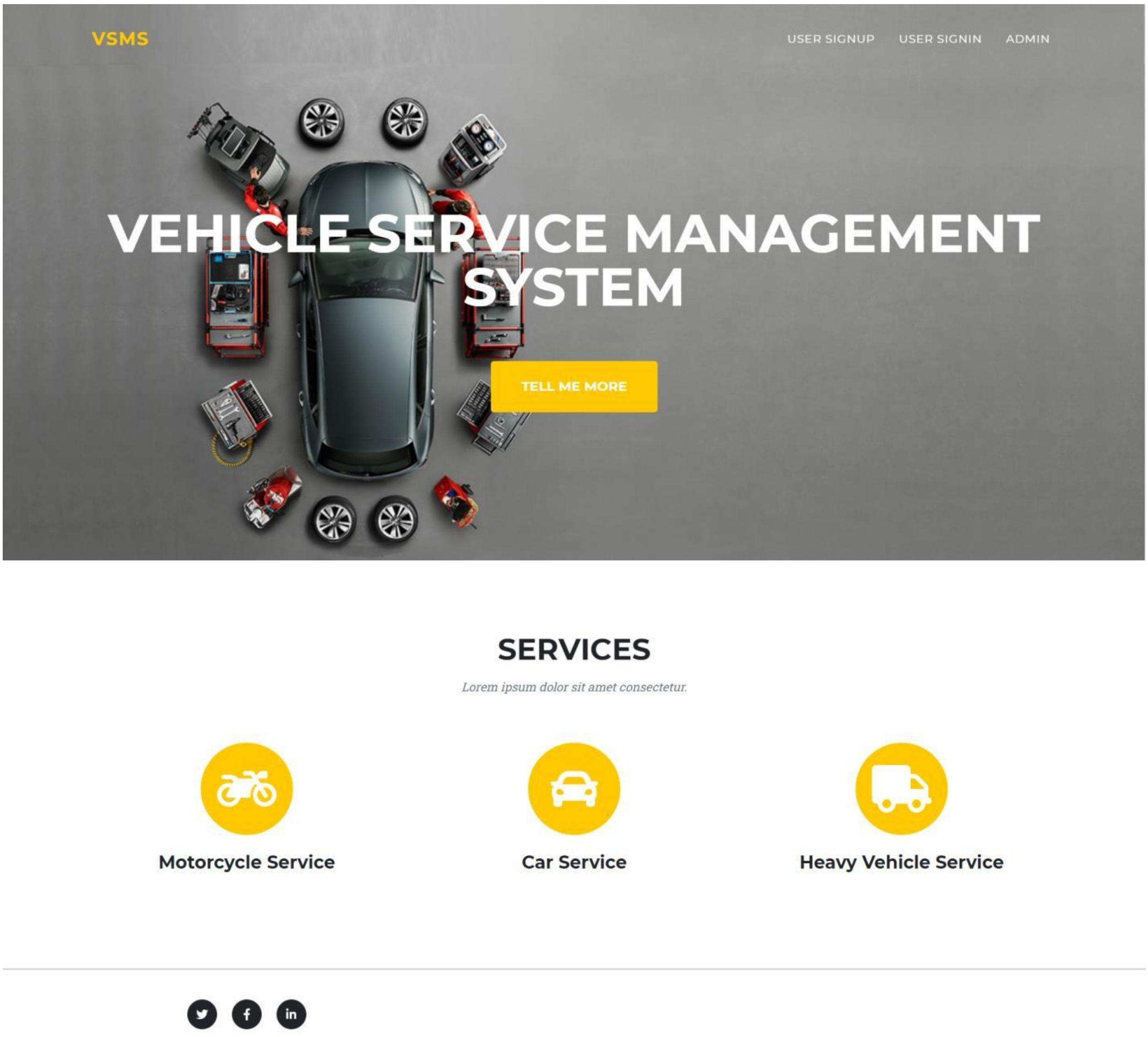
The User Interface (UI) of the Vehicle Servicing System is designed to offer an intuitive, seamless, and user-friendly experience for all stakeholders—customers, service providers, and administrators. The system's interface is structured to simplify booking services, managing vehicle data, and handling payments. Below are the main UI components.

1. **Login/Registration Screen**:
   - The first point of interaction for users is the **Login** screen, which provides fields for entering credentials such as email/username and password.
   - New users can access the **Registration** screen to sign up by providing their personal details (name, email, password, contact information) and selecting their role (Customer, Service Provider).
   - An option for password recovery is also available.
2. **User Dashboard**:
   - After logging in, users are greeted with the **Dashboard**, personalized based on their role (Customer, Service Provider, or Admin).
   - **For Customers**, the dashboard displays an overview of their vehicles, upcoming service appointments, service history, and payment statuses.
   - **Service Providers** view pending service requests, ongoing services, and feedback from completed services.
   - **Administrators** have control panels to manage users, monitor service providers, and track overall system performance.
3. **Vehicle Management**:
   - Users can view and manage details of their vehicles through the **Vehicle Management** section.
   - Features include the ability to **add new vehicles**, update existing vehicle details (model, registration number, warranty info), and view the service history.
   - Customers can upload documents related to the vehicle (e.g., insurance, warranty) for easy reference.
4. **Service Booking Interface**:
   - The **Service Booking** screen allows customers to schedule a service for their vehicle by selecting the vehicle, type of service (maintenance, repair, etc.), preferred date, and service provider.
   - A **Service Provider Locator** feature is integrated, allowing users to filter providers based on location, service type, and ratings.
   - The system provides real-time **cost estimates** for services before confirmation.
5. **Service Status and Tracking**:
   - Once a service is booked, customers can track the real-time status of their service requests through a **Service Status** interface.
   - This screen shows whether the service is pending, in progress, or completed.
   - Users receive live notifications and updates as their vehicle moves through the service stages.
6. **Service History**:
   - The **Service History** page displays all past services performed on the user's vehicles.

- Details like service dates, service providers, costs, and customer feedback are listed.
- This section allows customers to rate and review service providers based on their experiences, helping others make informed decisions.

Output Screens

Home Page



VEHICLE SERVICE MANAGEMENT SYSTEM

SERVICES

Lorem ipsum dolor sit amet consectetur.

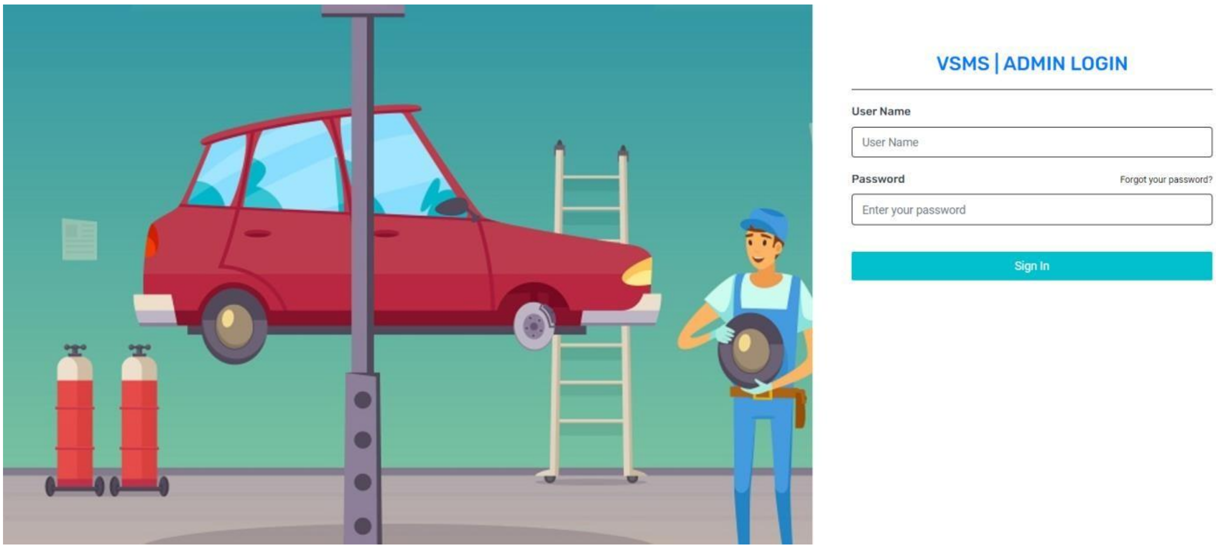
Motorcycle Service

Car Service

Heavy Vehicle Service

Admin Panel

Login Page



VSMS | ADMIN LOGIN

User Name

User Name

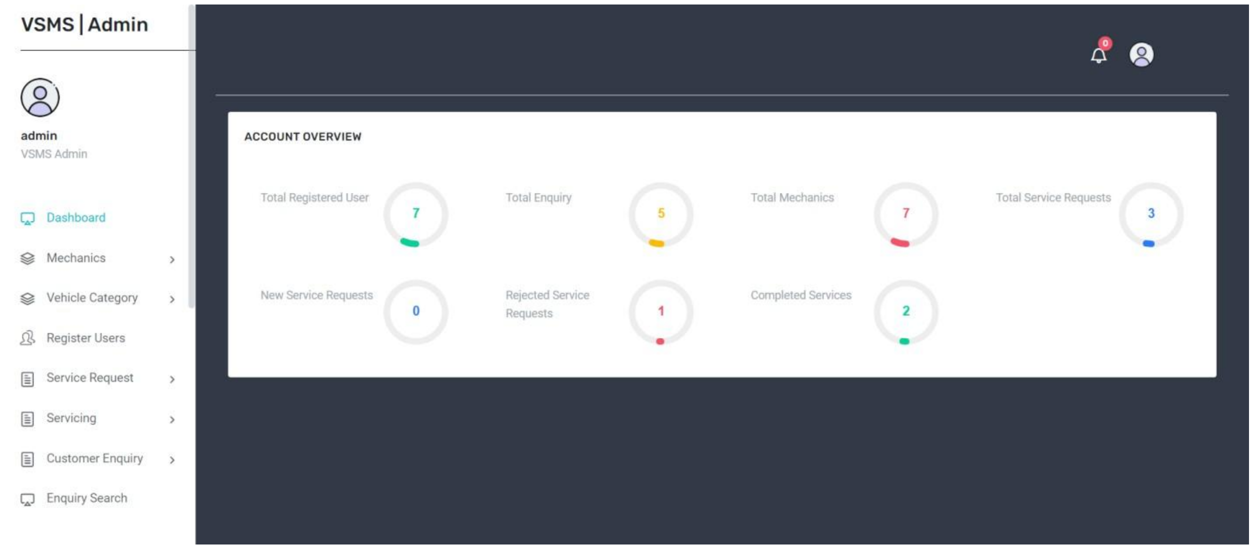
Password

Forgot your password?

Enter your password

Sign In

Dashboard



VSMS | Admin

admin

VSMS Admin

ACCOUNT OVERVIEW

Total Registered User 7

Total Enquiry 5

Total Mechanics 7

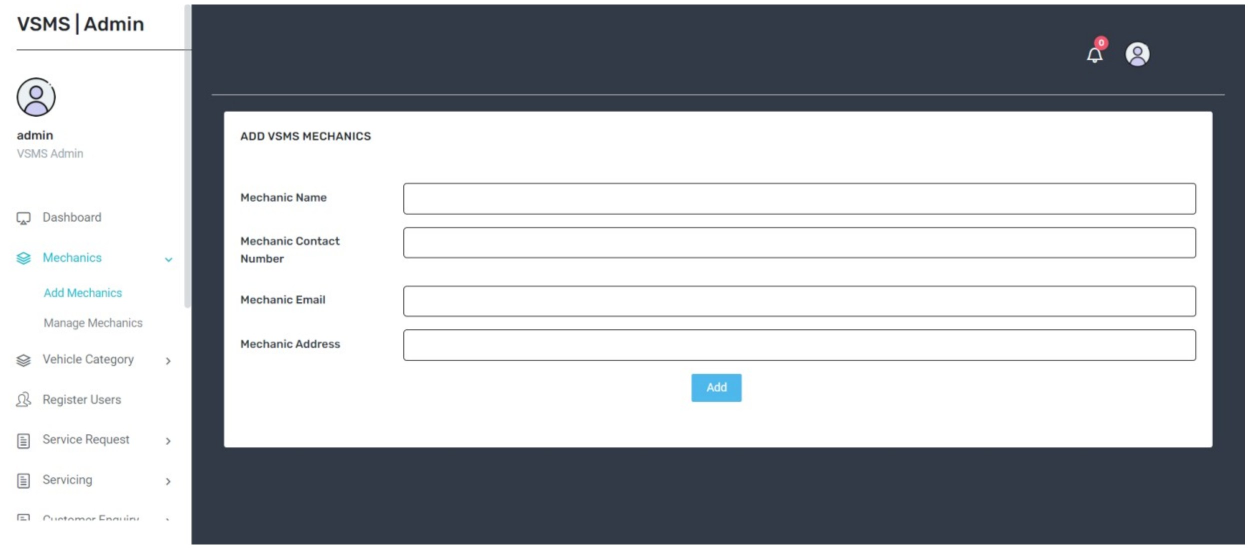
Total Service Requests 3

New Service Requests 0

Rejected Service Requests 1

Completed Services 2

Add Mechanics

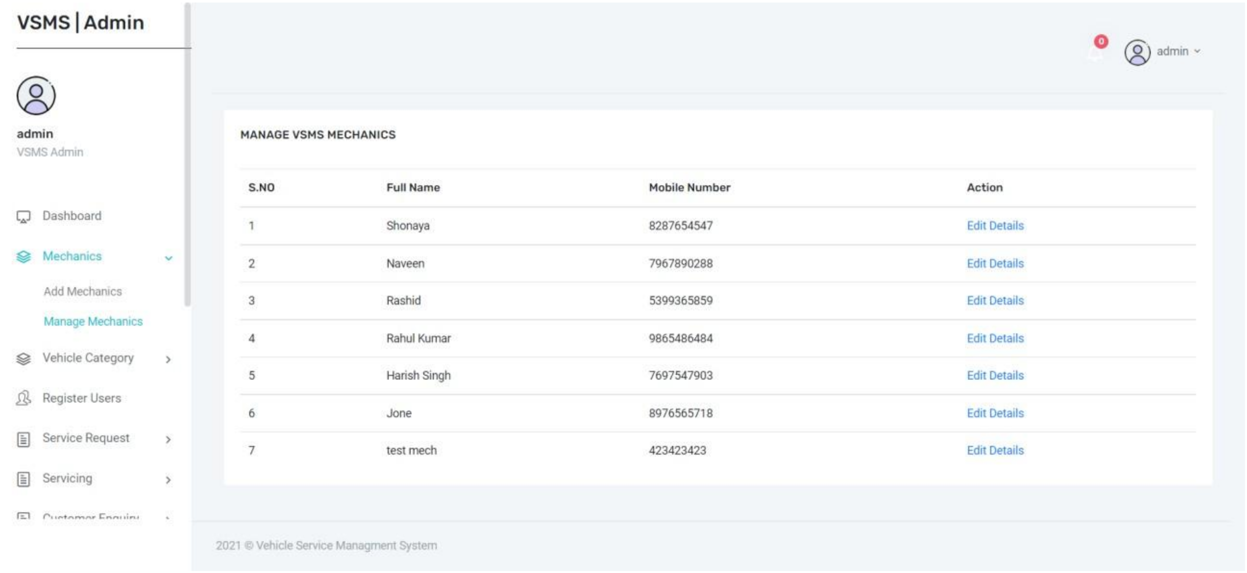


ADD VSMS MECHANICS

Mechanic Name	
Mechanic Contact Number	
Mechanic Email	
Mechanic Address	

Add

Manage Mechanics

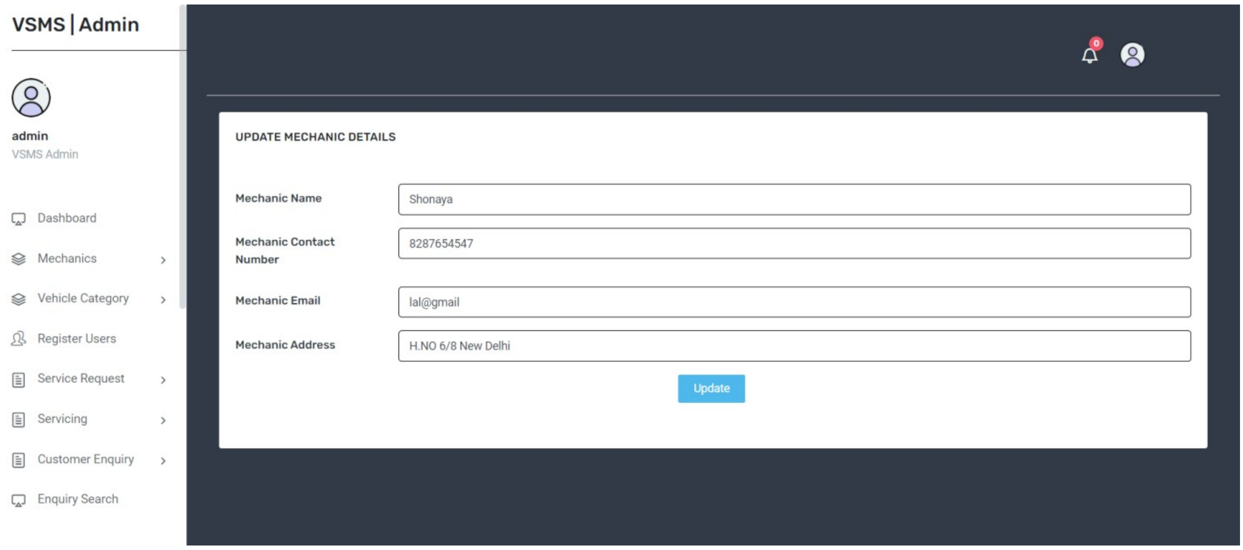


MANAGE VSMS MECHANICS

S.NO	Full Name	Mobile Number	Action
1	Shonaya	8287654547	Edit Details
2	Naveen	7967890288	Edit Details
3	Rashid	5399365859	Edit Details
4	Rahul Kumar	9865486484	Edit Details
5	Harish Singh	7697547903	Edit Details
6	Jone	8976565718	Edit Details
7	test mech	423423423	Edit Details

2021 © Vehicle Service Managment System

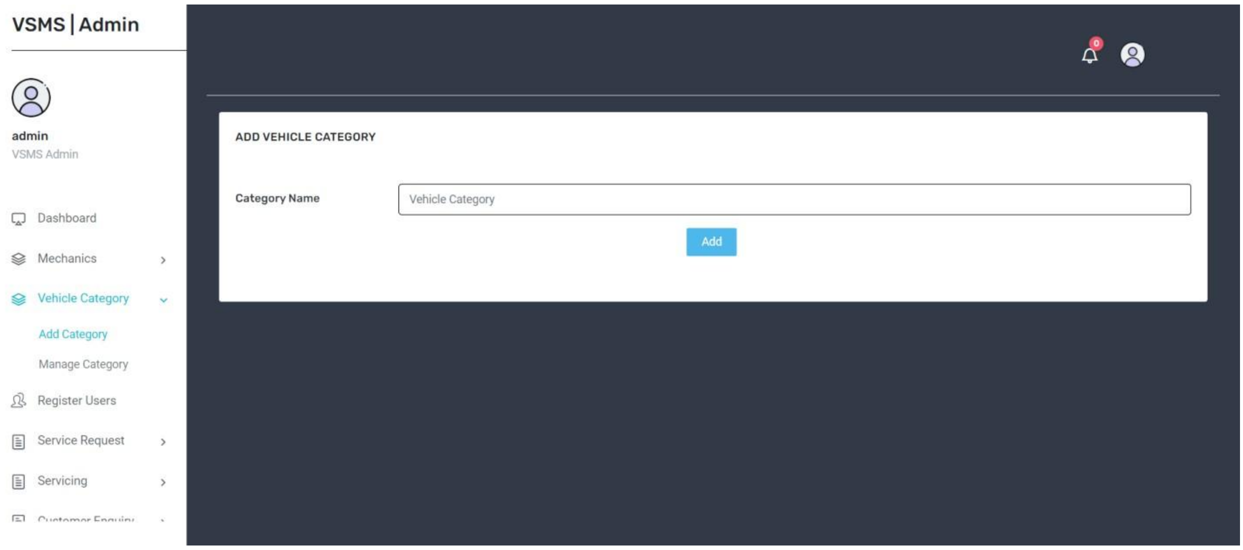
Update Mechanics



UPDATE MECHANIC DETAILS

Mechanic Name	Shonaya
Mechanic Contact Number	8287654547
Mechanic Email	lal@gmail
Mechanic Address	H.NO 6/8 New Delhi

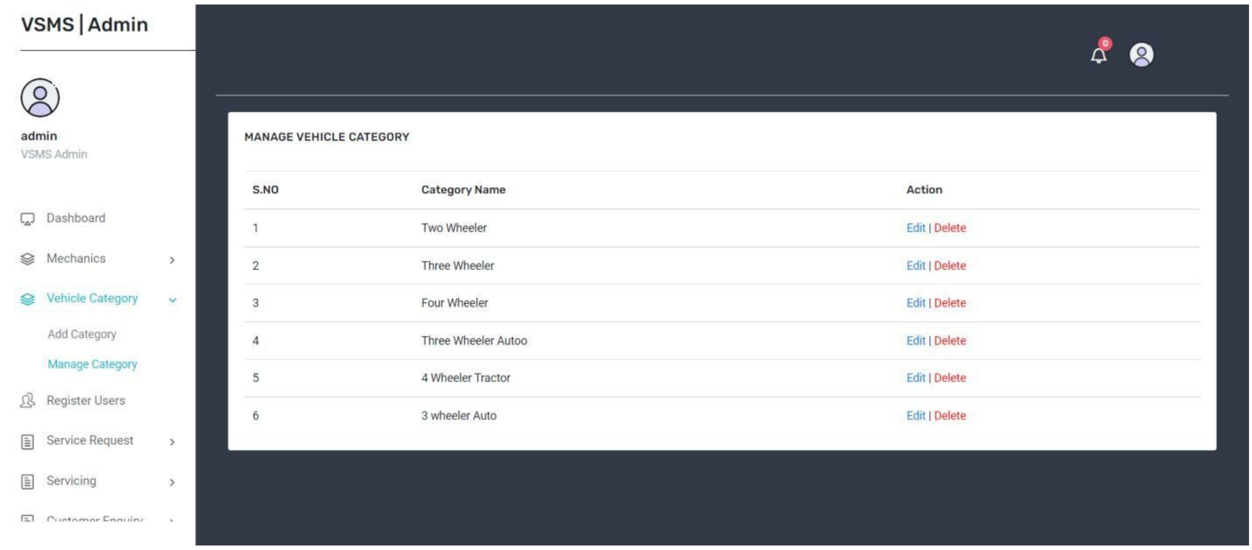
Add Vehicle Category



ADD VEHICLE CATEGORY

Category Name	Vehicle Category

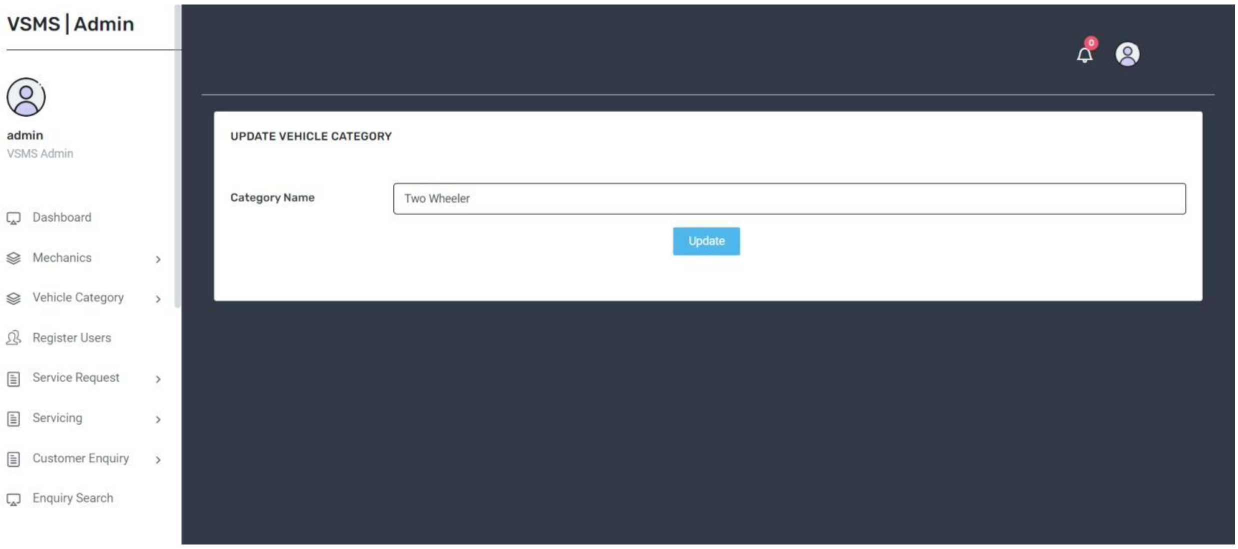
Manage Vehicle Category



MANAGE VEHICLE CATEGORY

S.NO	Category Name	Action
1	Two Wheeler	Edit | Delete
2	Three Wheeler	Edit | Delete
3	Four Wheeler	Edit | Delete
4	Three Wheeler Autoo	Edit | Delete
5	4 Wheeler Tractor	Edit | Delete
6	3 wheeler Auto	Edit | Delete

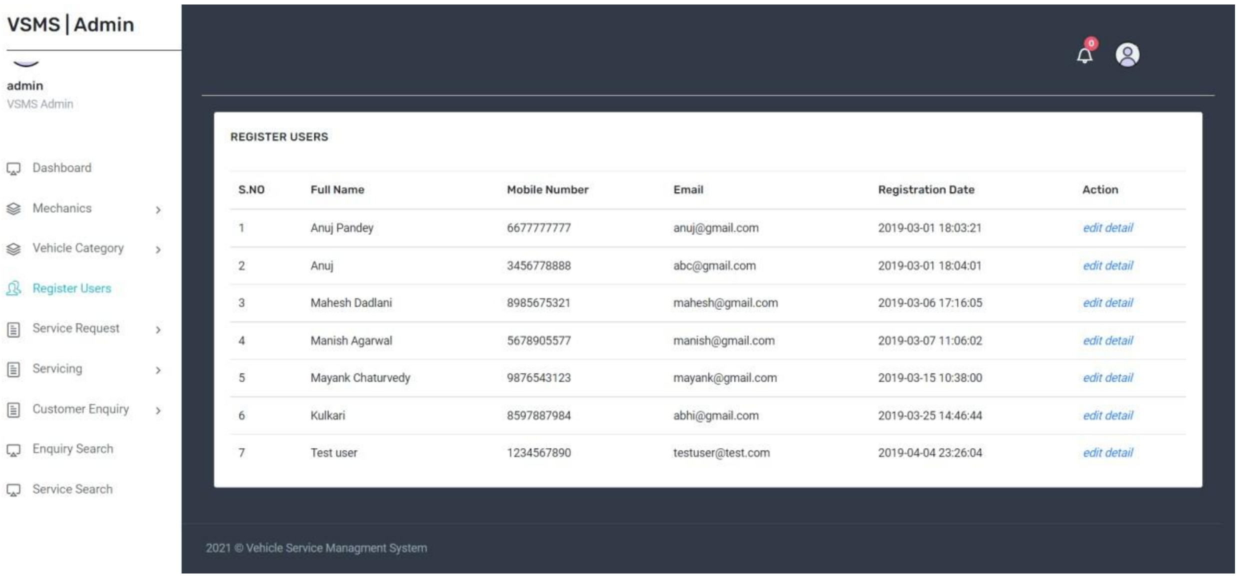
Update Vehicle Category



UPDATE VEHICLE CATEGORY

Category Name: Two Wheeler

Registered Users

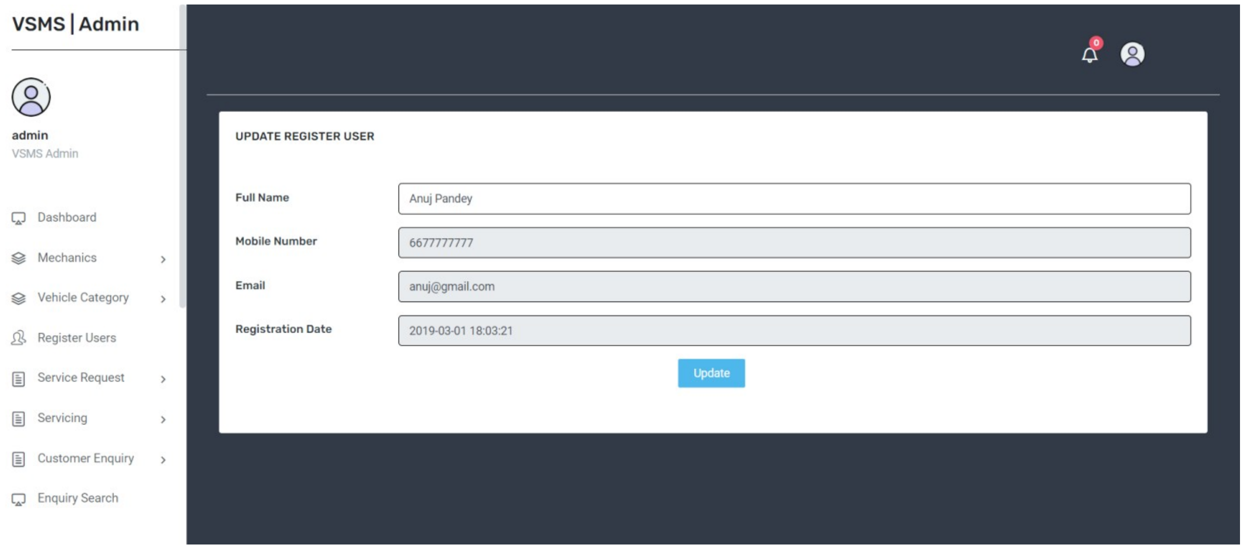


REGISTER USERS

S.NO	Full Name	Mobile Number	Email	Registration Date	Action
1	Anuj Pandey	6677777777	anuj@gmail.com	2019-03-01 18:03:21	*edit detail*
2	Anuj	3456778888	abc@gmail.com	2019-03-01 18:04:01	*edit detail*
3	Mahesh Dadlani	8985675321	mahesh@gmail.com	2019-03-06 17:16:05	*edit detail*
4	Manish Agarwal	5678905577	manish@gmail.com	2019-03-07 11:06:02	*edit detail*
5	Mayank Chaturvedy	9876543123	mayank@gmail.com	2019-03-15 10:38:00	*edit detail*
6	Kulkari	8597887984	abhi@gmail.com	2019-03-25 14:46:44	*edit detail*
7	Test user	1234567890	testuser@test.com	2019-04-04 23:26:04	*edit detail*

2021 © Vehicle Service Managment System

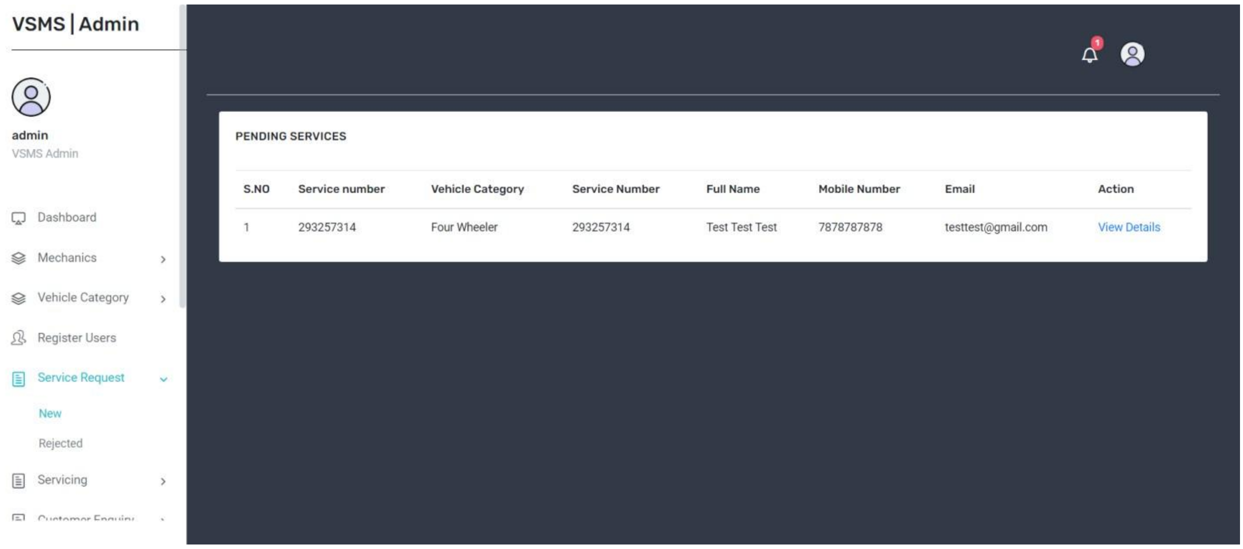
Update Registered Users



UPDATE REGISTER USER

Full Name	Anuj Pandey
Mobile Number	6677777777
Email	anuj@gmail.com
Registration Date	2019-03-01 18:03:21

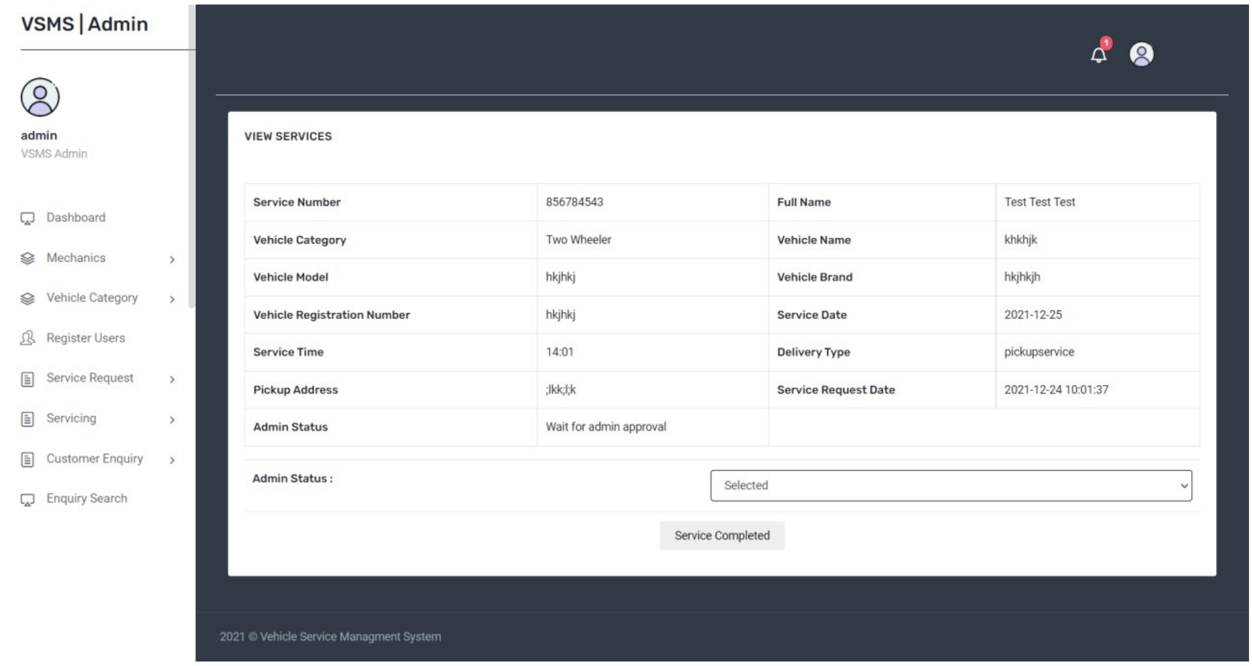
New Service Request



PENDING SERVICES

S.NO	Service number	Vehicle Category	Service Number	Full Name	Mobile Number	Email	Action
1	293257314	Four Wheeler	293257314	Test Test Test	7878787878	testtest@gmail.com	View Details

View Pending Services

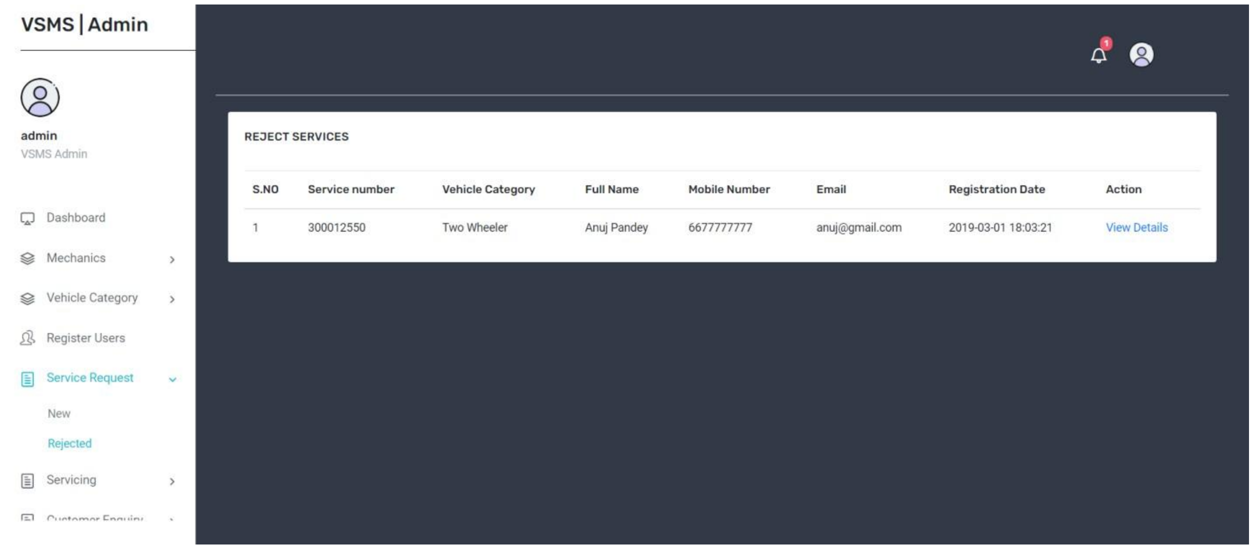


VIEW SERVICES

Service Number	856784543	Full Name	Test Test Test
Vehicle Category	Two Wheeler	Vehicle Name	khkhjk
Vehicle Model	hkjhkj	Vehicle Brand	hkjhkjh
Vehicle Registration Number	hkjhkj	Service Date	2021-12-25
Service Time	14:01	Delivery Type	pickupservice
Pickup Address	;lkk;l;k	Service Request Date	2021-12-24 10:01:37
Admin Status	Wait for admin approval		

Admin Status : Selected

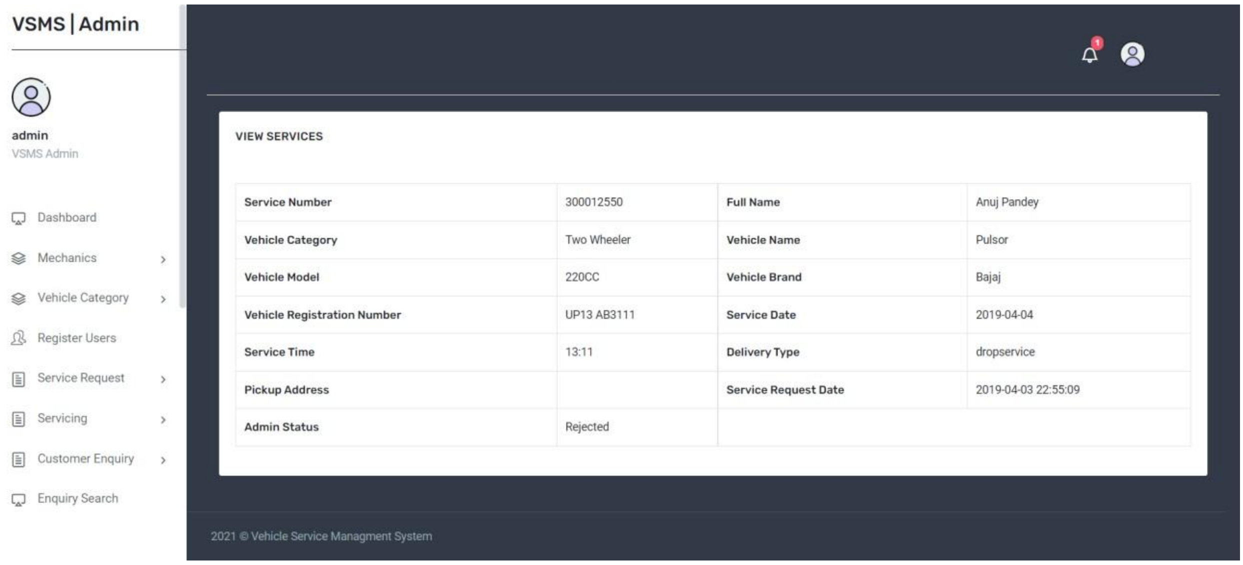
Rejected Service



REJECT SERVICES

S.NO	Service number	Vehicle Category	Full Name	Mobile Number	Email	Registration Date	Action
1	300012550	Two Wheeler	Anuj Pandey	6677777777	anuj@gmail.com	2019-03-01 18:03:21	View Details

View Rejected Service

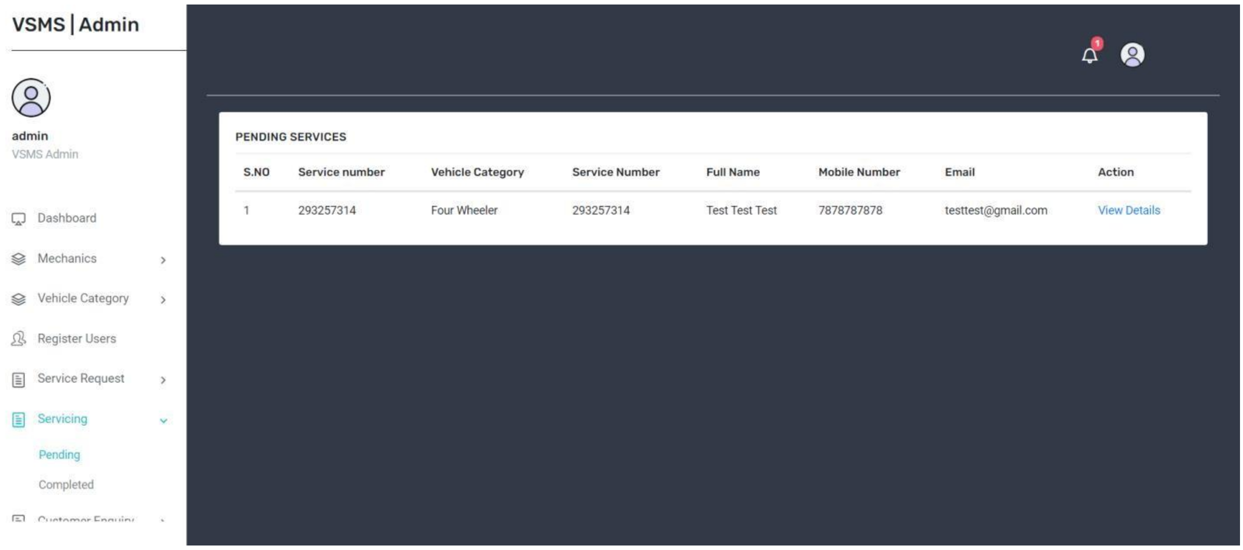


VIEW SERVICES

Service Number	300012550	Full Name	Anuj Pandey
Vehicle Category	Two Wheeler	Vehicle Name	Pulsor
Vehicle Model	220CC	Vehicle Brand	Bajaj
Vehicle Registration Number	UP13 AB3111	Service Date	2019-04-04
Service Time	13:11	Delivery Type	dropservice
Pickup Address		Service Request Date	2019-04-03 22:55:09
Admin Status	Rejected		

2021 © Vehicle Service Managment System

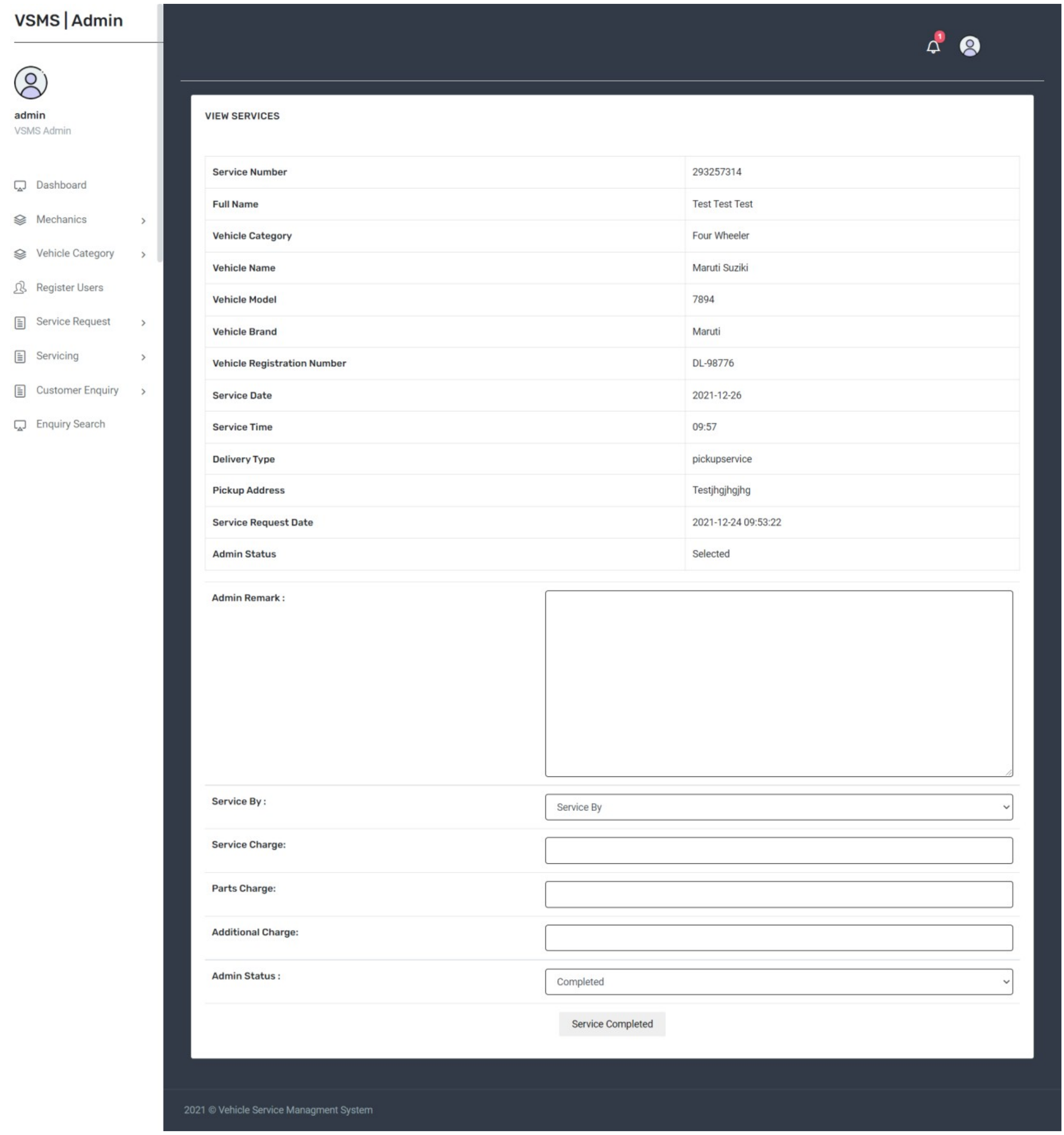
Service Pending



PENDING SERVICES

S.NO	Service number	Vehicle Category	Service Number	Full Name	Mobile Number	Email	Action
1	293257314	Four Wheeler	293257314	Test Test Test	7878787878	testtest@gmail.com	View Details

View Service Pending



VIEW SERVICES

Service Number	293257314
Full Name	Test Test Test
Vehicle Category	Four Wheeler
Vehicle Name	Maruti Suziki
Vehicle Model	7894
Vehicle Brand	Maruti
Vehicle Registration Number	DL-98776
Service Date	2021-12-26
Service Time	09:57
Delivery Type	pickupservice
Pickup Address	Testjhgjhgjhg
Service Request Date	2021-12-24 09:53:22
Admin Status	Selected

Admin Remark :

Service By : Service By

Service Charge:

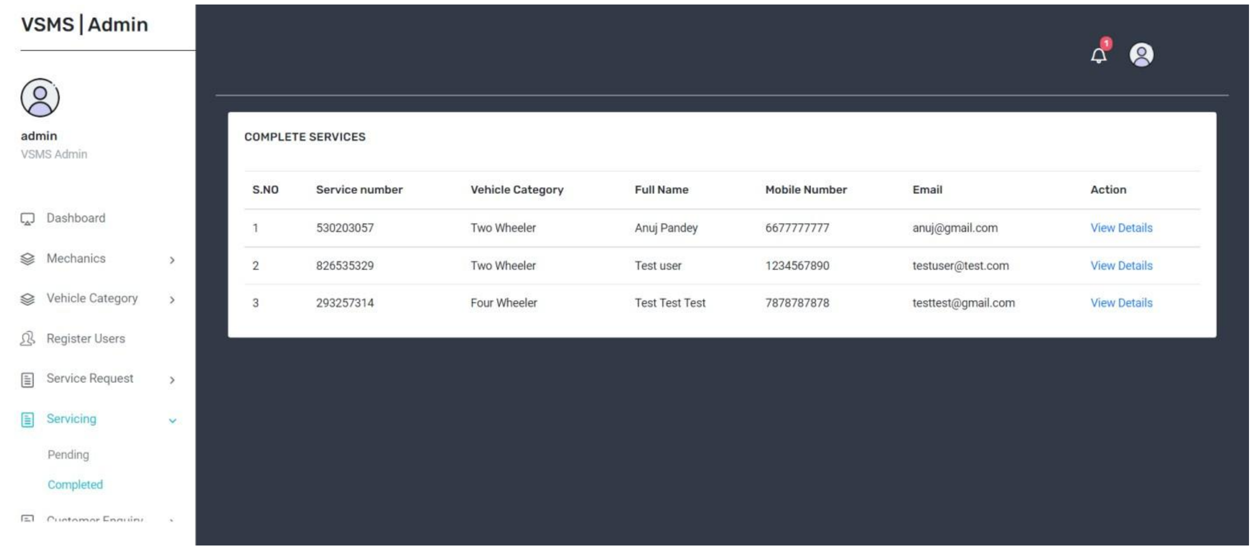
Parts Charge:

Additional Charge:

Admin Status : Completed

2021 © Vehicle Service Managment System

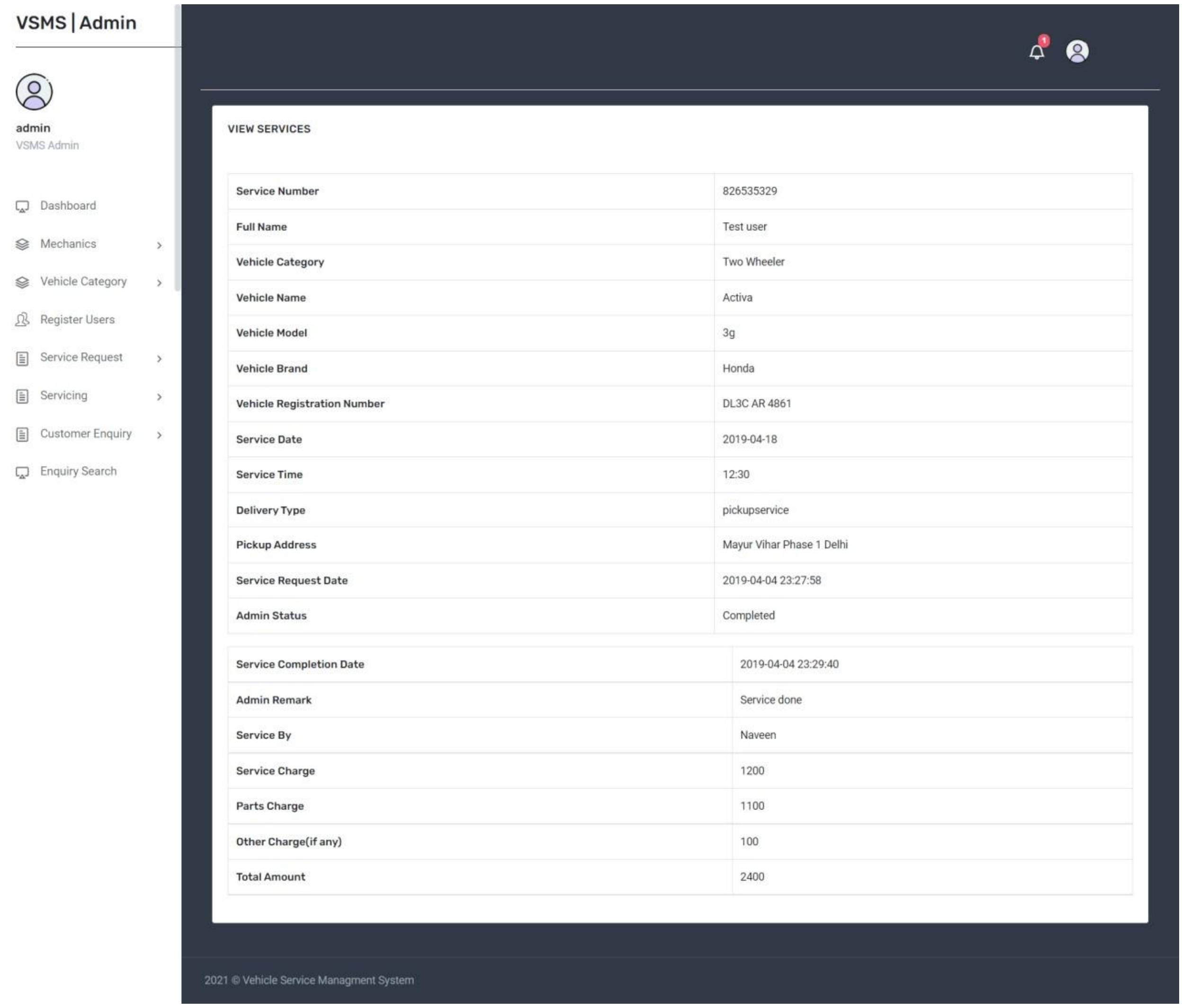
Completed Service



COMPLETE SERVICES

S.NO	Service number	Vehicle Category	Full Name	Mobile Number	Email	Action
1	530203057	Two Wheeler	Anuj Pandey	6677777777	anuj@gmail.com	View Details
2	826535329	Two Wheeler	Test user	1234567890	testuser@test.com	View Details
3	293257314	Four Wheeler	Test Test Test	7878787878	testtest@gmail.com	View Details

View Completed Service



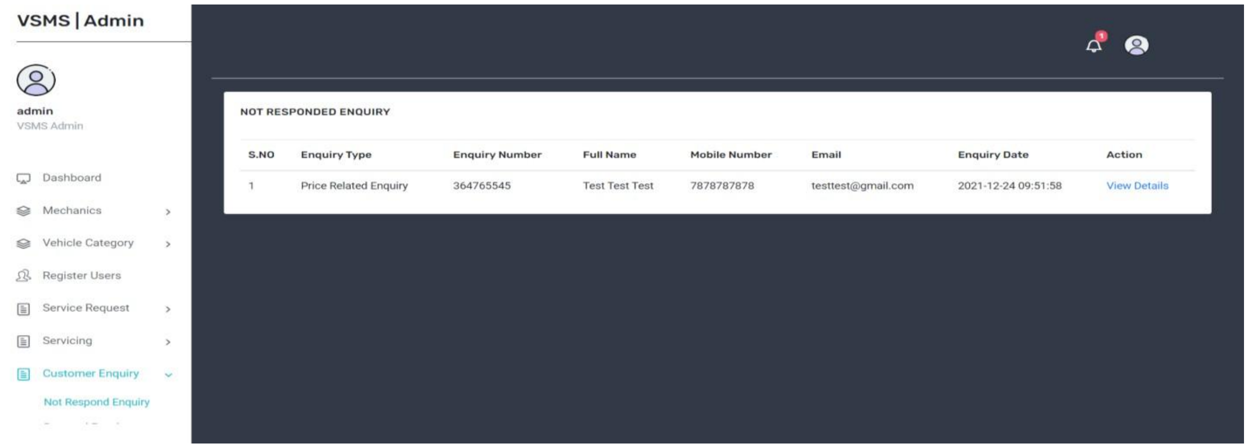
VIEW SERVICES

Service Number	826535329
Full Name	Test user
Vehicle Category	Two Wheeler
Vehicle Name	Activa
Vehicle Model	3g
Vehicle Brand	Honda
Vehicle Registration Number	DL3C AR 4861
Service Date	2019-04-18
Service Time	12:30
Delivery Type	pickupservice
Pickup Address	Mayur Vihar Phase 1 Delhi
Service Request Date	2019-04-04 23:27:58
Admin Status	Completed

Service Completion Date	2019-04-04 23:29:40
Admin Remark	Service done
Service By	Naveen
Service Charge	1200
Parts Charge	1100
Other Charge(if any)	100
Total Amount	2400

2021 © Vehicle Service Managment System

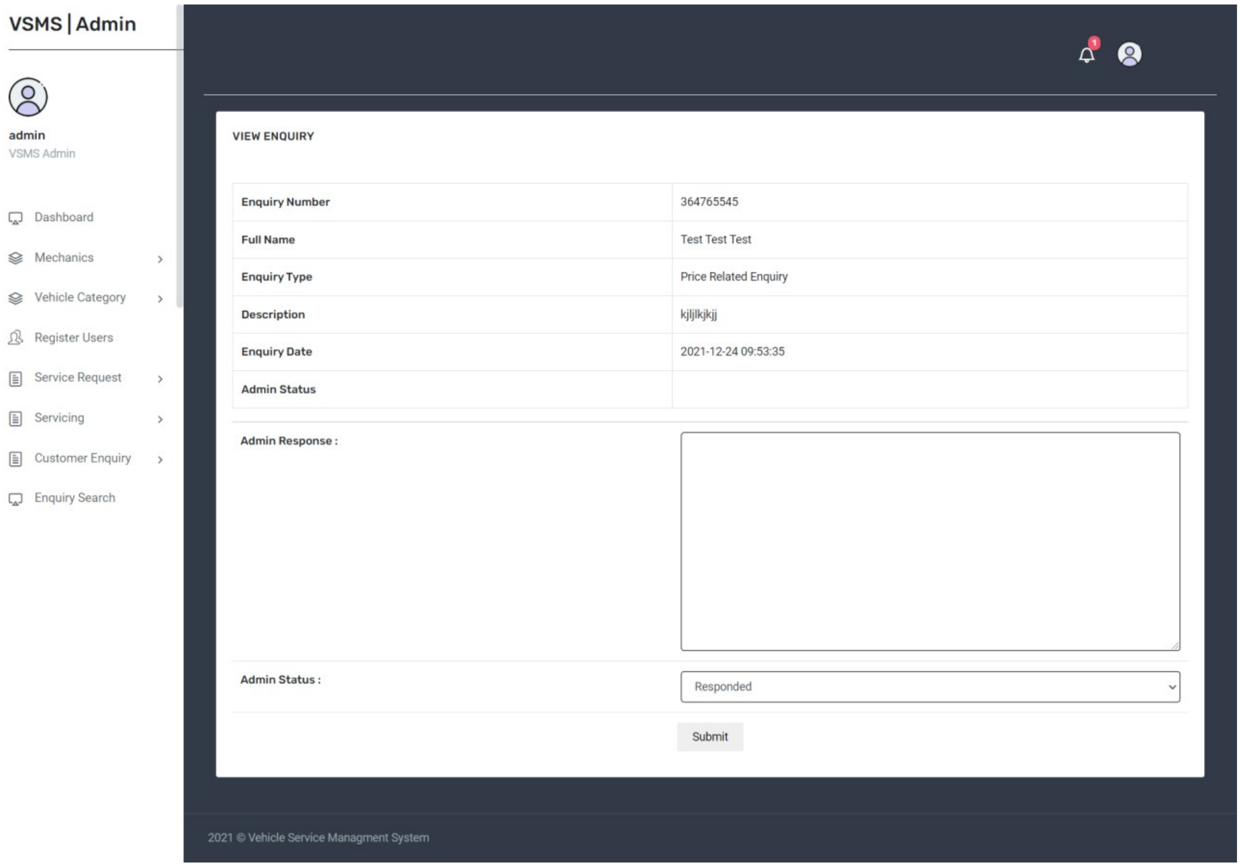
Customer Enquiry



NOT RESPONDED ENQUIRY

S.NO	Enquiry Type	Enquiry Number	Full Name	Mobile Number	Email	Enquiry Date	Action
1	Price Related Enquiry	364765545	Test Test Test	7878787878	testtest@gmail.com	2021-12-24 09:51:58	View Details

View Customer Enquiry



VIEW ENQUIRY

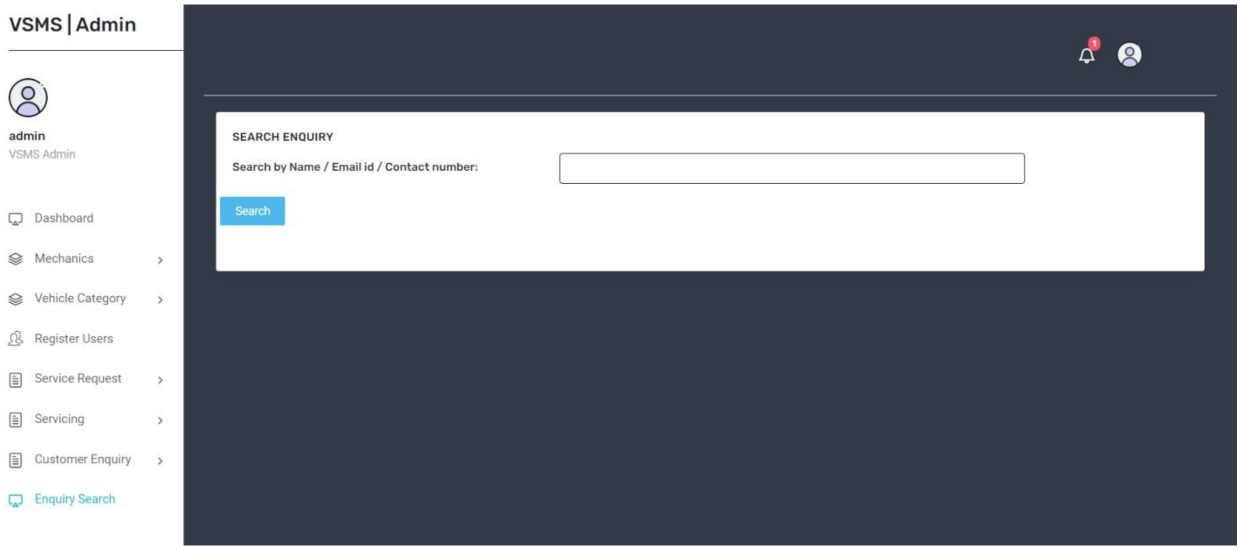
Enquiry Number	364765545
Full Name	Test Test Test
Enquiry Type	Price Related Enquiry
Description	kjljlkjkjj
Enquiry Date	2021-12-24 09:53:35
Admin Status	

Admin Response :

Admin Status : Responded

2021 © Vehicle Service Managment System

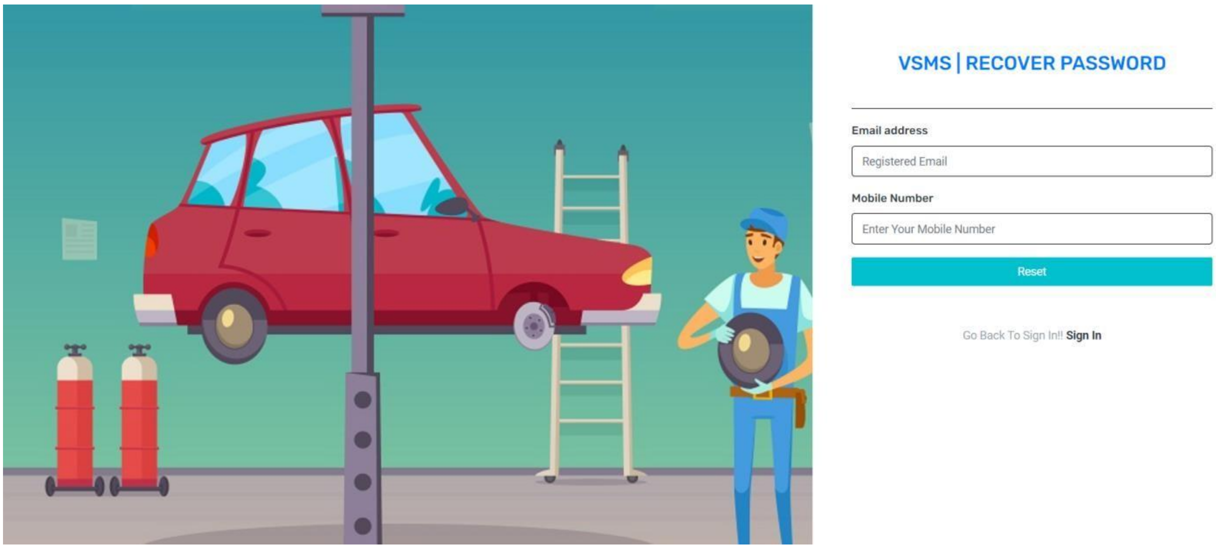
Enquiry Search



SEARCH ENQUIRY

Search by Name / Email id / Contact number:

Forgot Password



VSMS | RECOVER PASSWORD

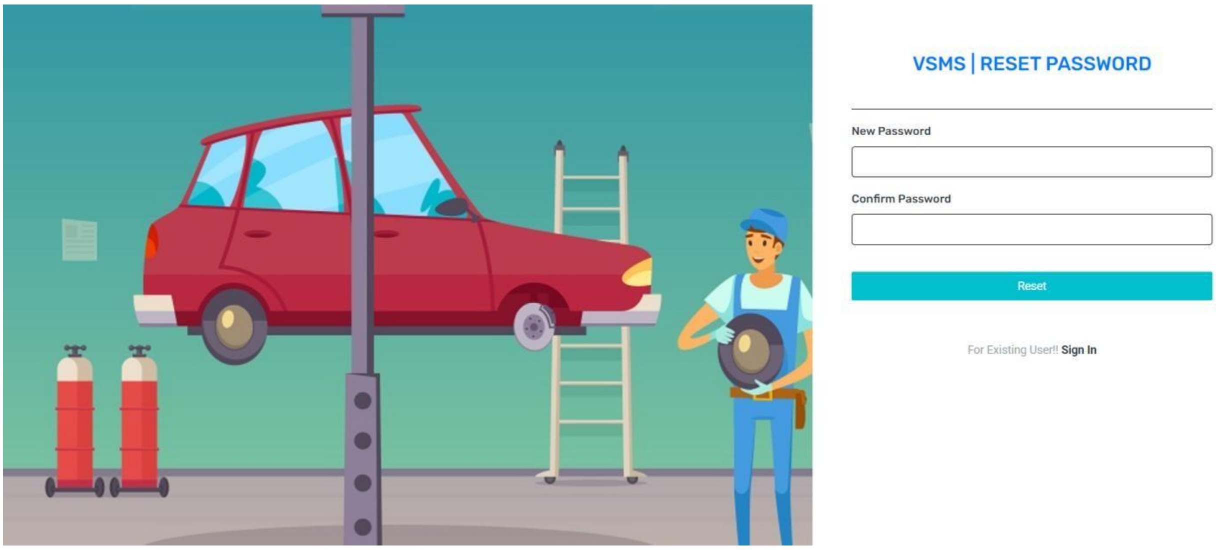
Email address

Registered Email

Mobile Number

Enter Your Mobile Number

Reset Password



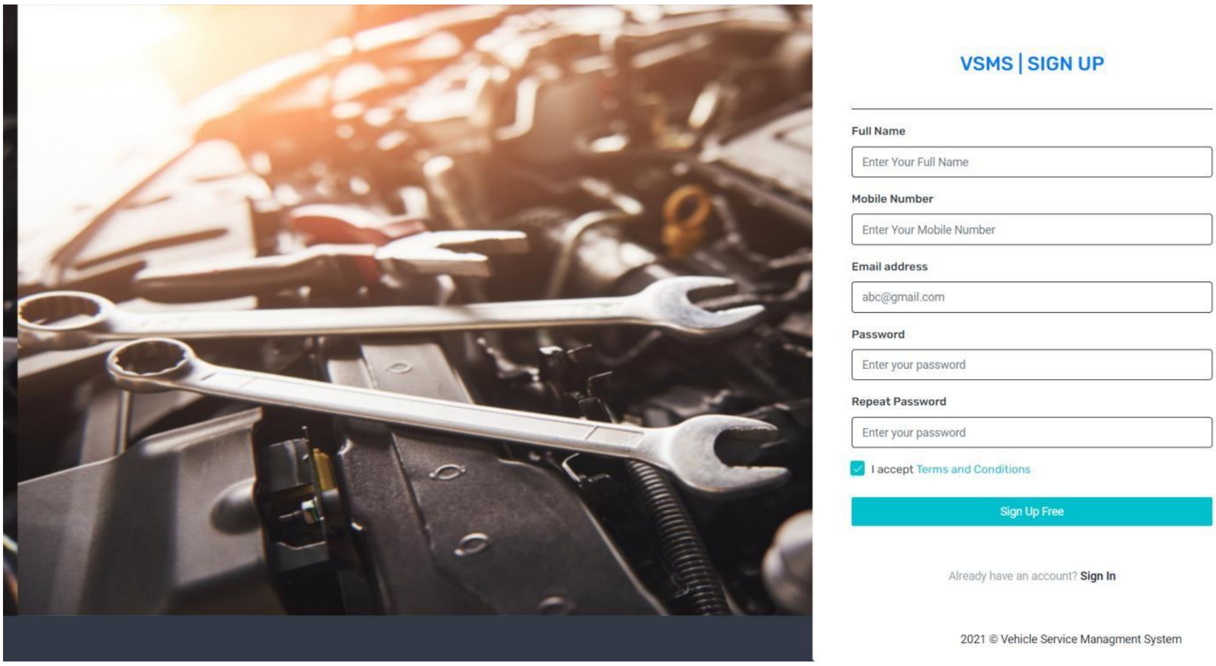
VSMS | RESET PASSWORD

New Password

Confirm Password

User Panel

Sign Up Page



VSMS | SIGN UP

Full Name

Enter Your Full Name

Mobile Number

Enter Your Mobile Number

Email address

abc@gmail.com

Password

Enter your password

Repeat Password

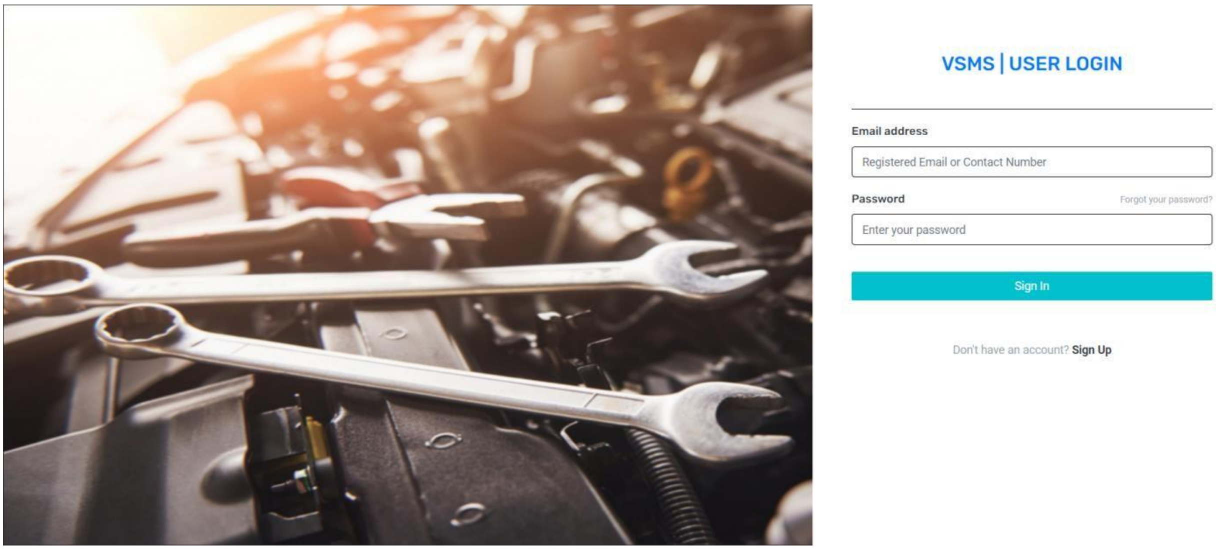
Enter your password

☑ I accept Terms and Conditions

Already have an account? **Sign In**

2021 © Vehicle Service Managment System

Sign in



VSMS | USER LOGIN

Email address

Registered Email or Contact Number

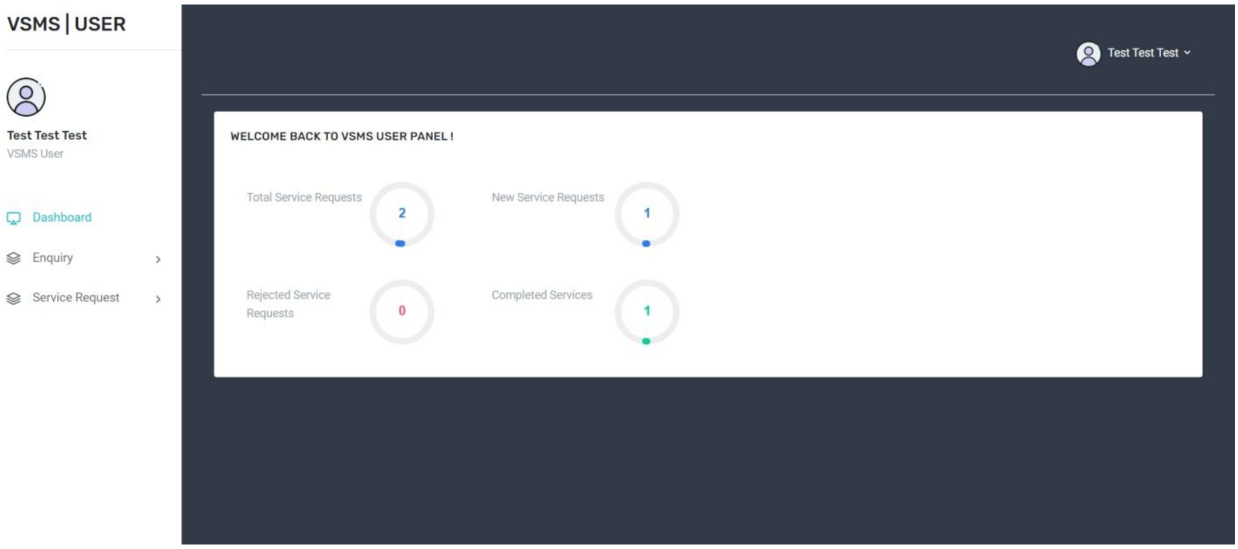
Password

Forgot your password?

Enter your password

Don't have an account? **Sign Up**

Dashboard



WELCOME BACK TO VSMS USER PANEL !

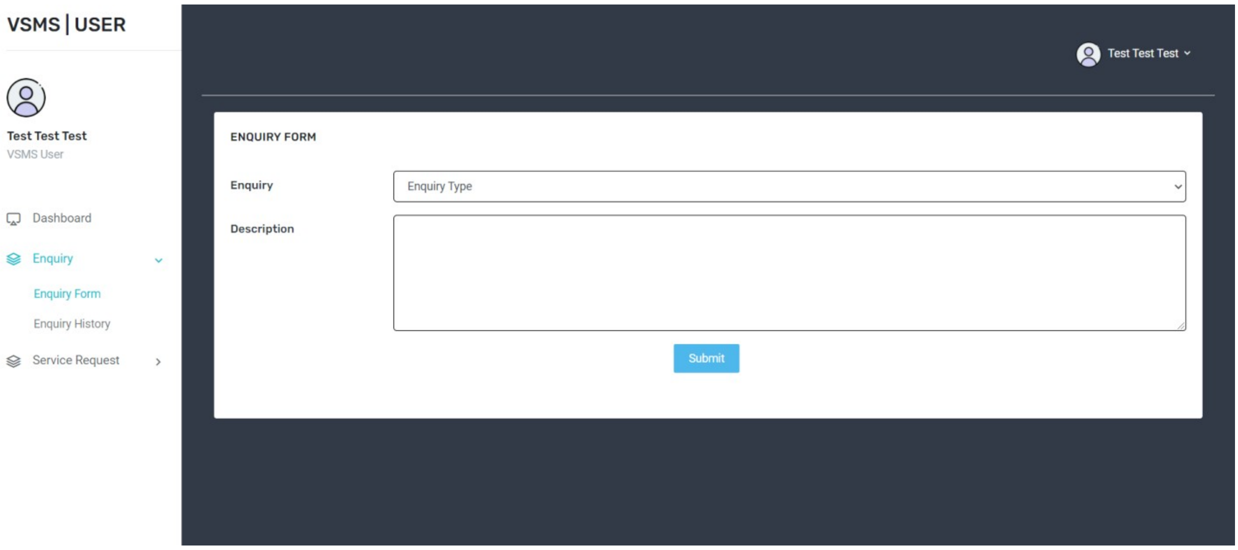
Total Service Requests 2

New Service Requests 1

Rejected Service Requests 0

Completed Services 1

Enquiry Form

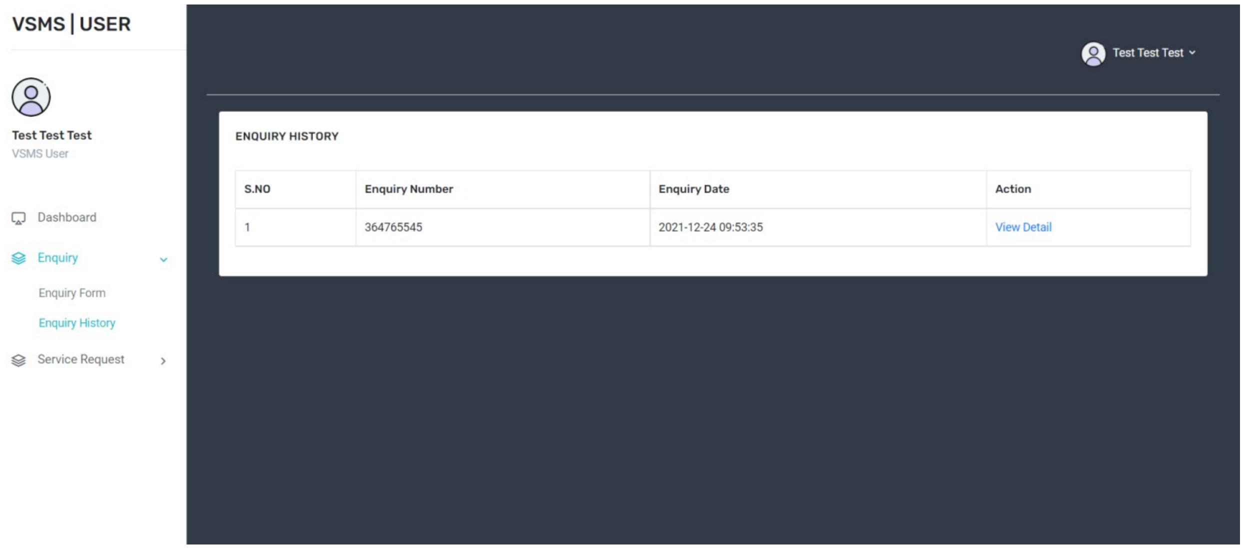


ENQUIRY FORM

Enquiry: Enquiry Type

Description

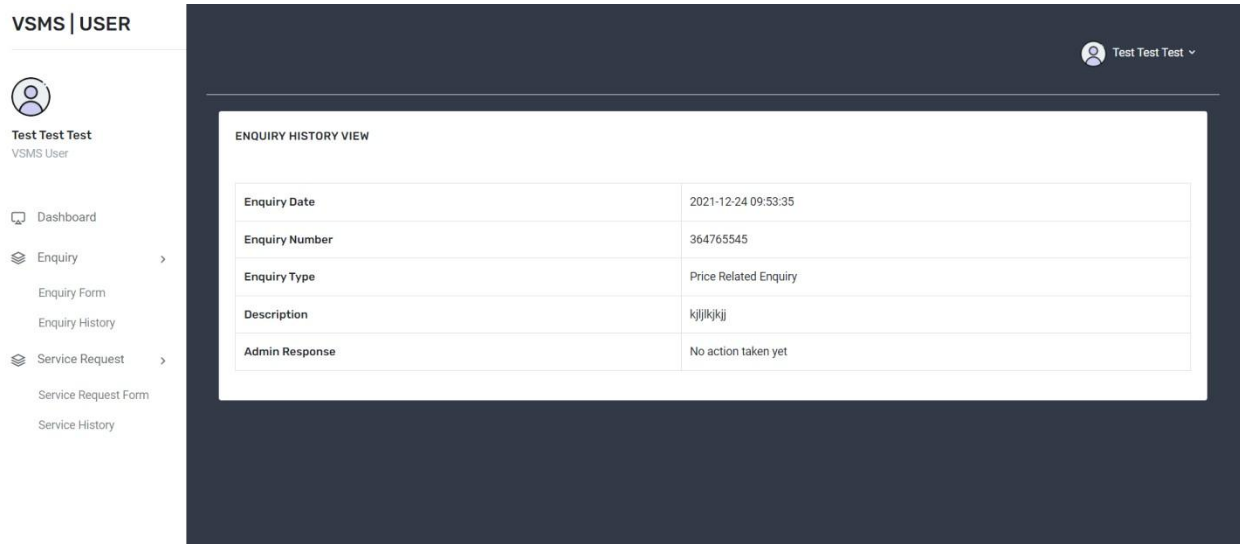
Enquiry History



ENQUIRY HISTORY

S.NO	Enquiry Number	Enquiry Date	Action
1	364765545	2021-12-24 09:53:35	View Detail

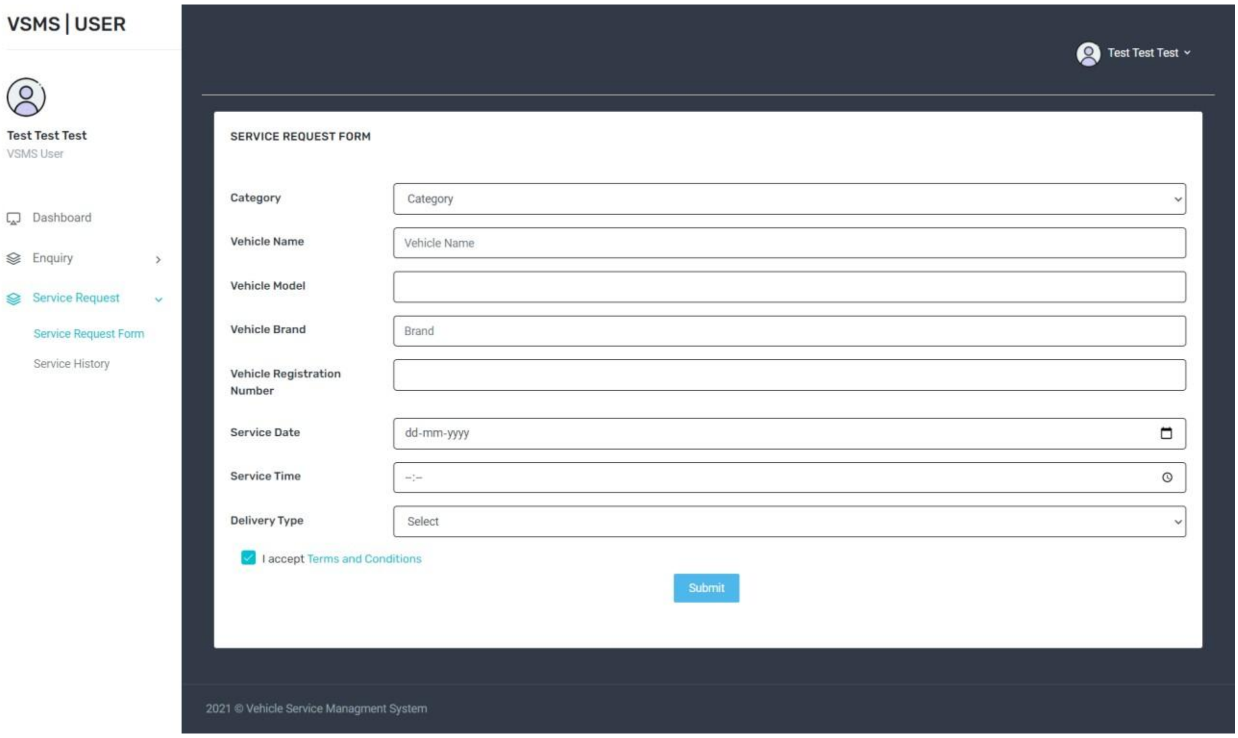
View Enquiry History



ENQUIRY HISTORY VIEW

Enquiry Date	2021-12-24 09:53:35
Enquiry Number	364765545
Enquiry Type	Price Related Enquiry
Description	kjljlkjkjj
Admin Response	No action taken yet

Service Request Form



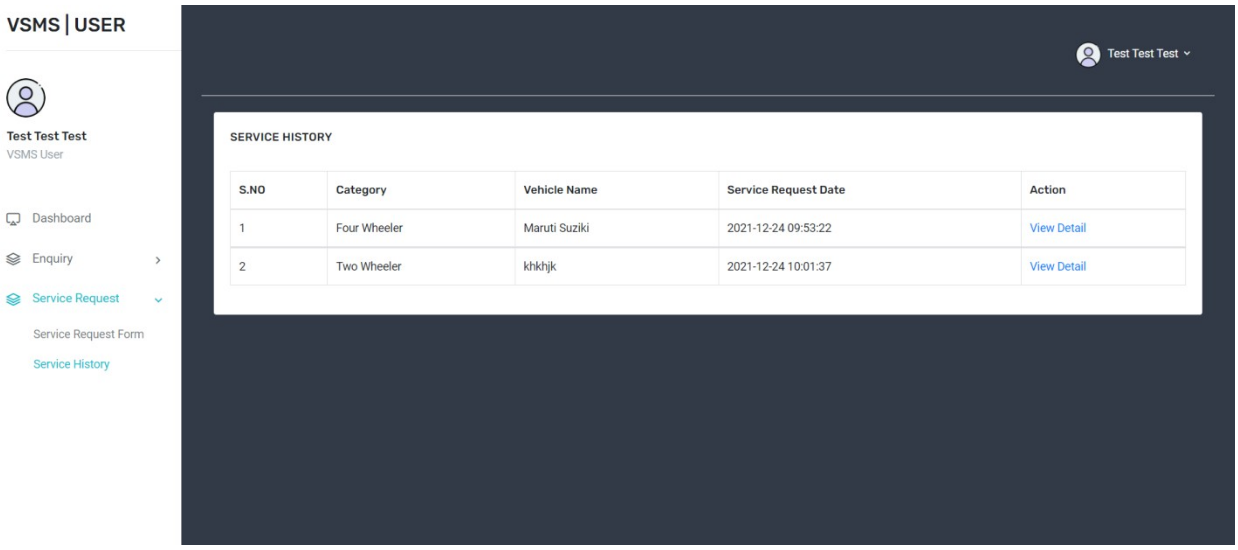
SERVICE REQUEST FORM

Category	Category
Vehicle Name	Vehicle Name
Vehicle Model	
Vehicle Brand	Brand
Vehicle Registration Number	
Service Date	dd-mm-yyyy
Service Time	--:--
Delivery Type	Select

☑ I accept Terms and Conditions

2021 © Vehicle Service Managment System

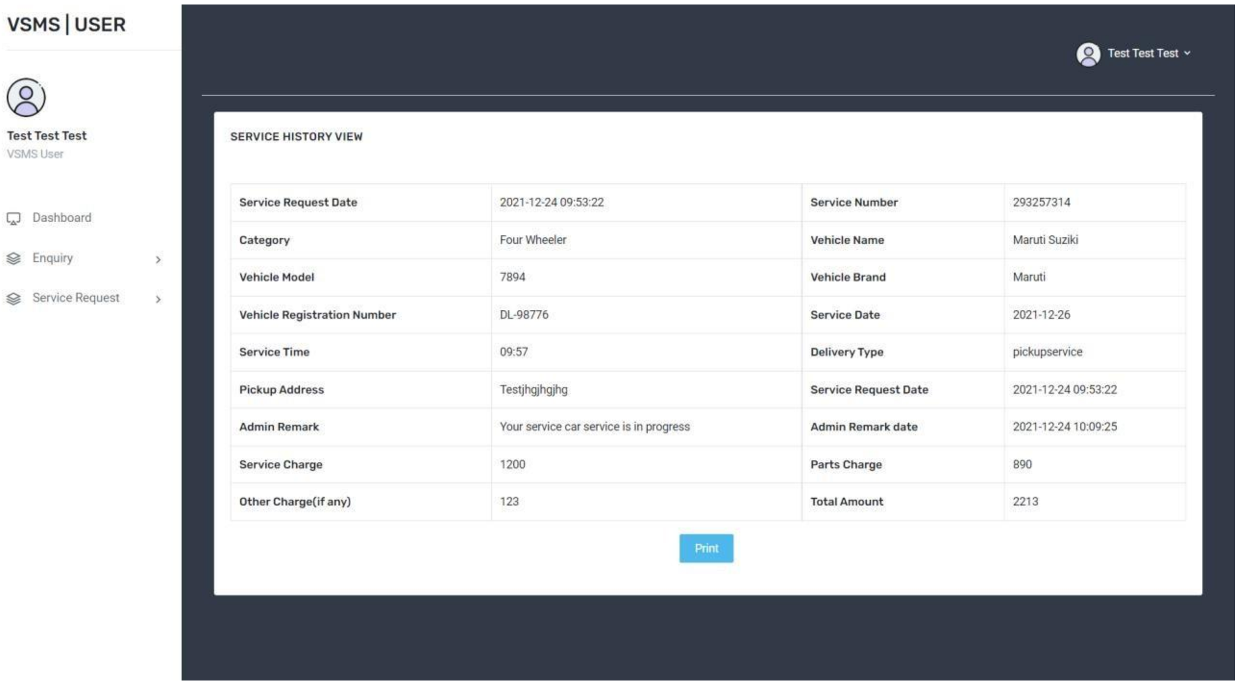
Service History



SERVICE HISTORY

S.NO	Category	Vehicle Name	Service Request Date	Action
1	Four Wheeler	Maruti Suziki	2021-12-24 09:53:22	View Detail
2	Two Wheeler	khkhjk	2021-12-24 10:01:37	View Detail

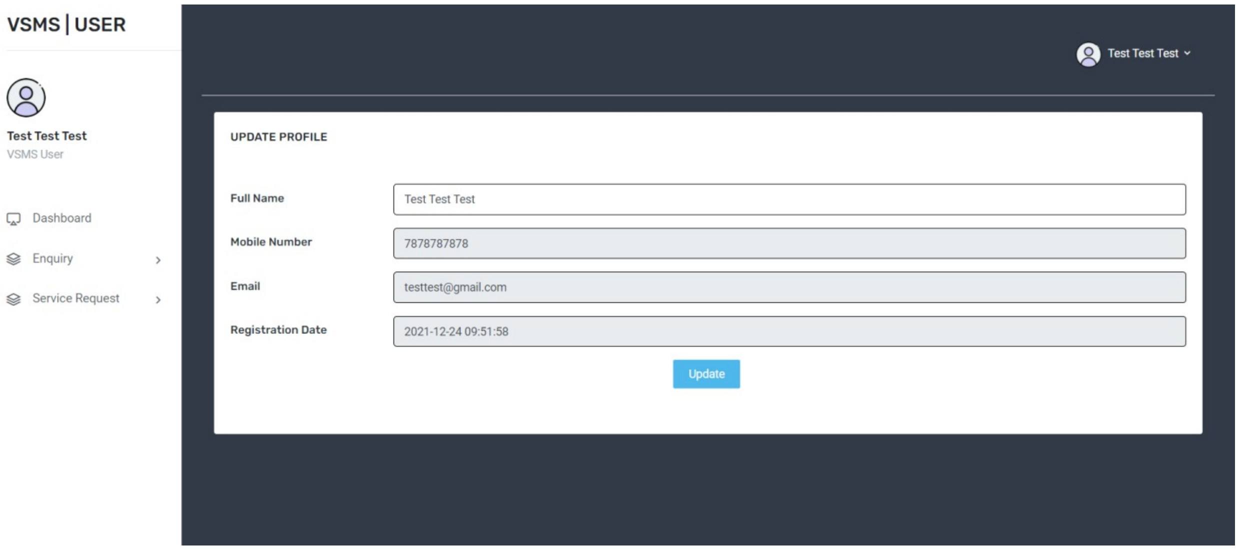
View Service History



SERVICE HISTORY VIEW

Service Request Date	2021-12-24 09:53:22	Service Number	293257314
Category	Four Wheeler	Vehicle Name	Maruti Suziki
Vehicle Model	7894	Vehicle Brand	Maruti
Vehicle Registration Number	DL-98776	Service Date	2021-12-26
Service Time	09:57	Delivery Type	pickupservice
Pickup Address	Testjhgjhgjhg	Service Request Date	2021-12-24 09:53:22
Admin Remark	Your service car service is in progress	Admin Remark date	2021-12-24 10:09:25
Service Charge	1200	Parts Charge	890
Other Charge(if any)	123	Total Amount	2213

User Profile



UPDATE PROFILE

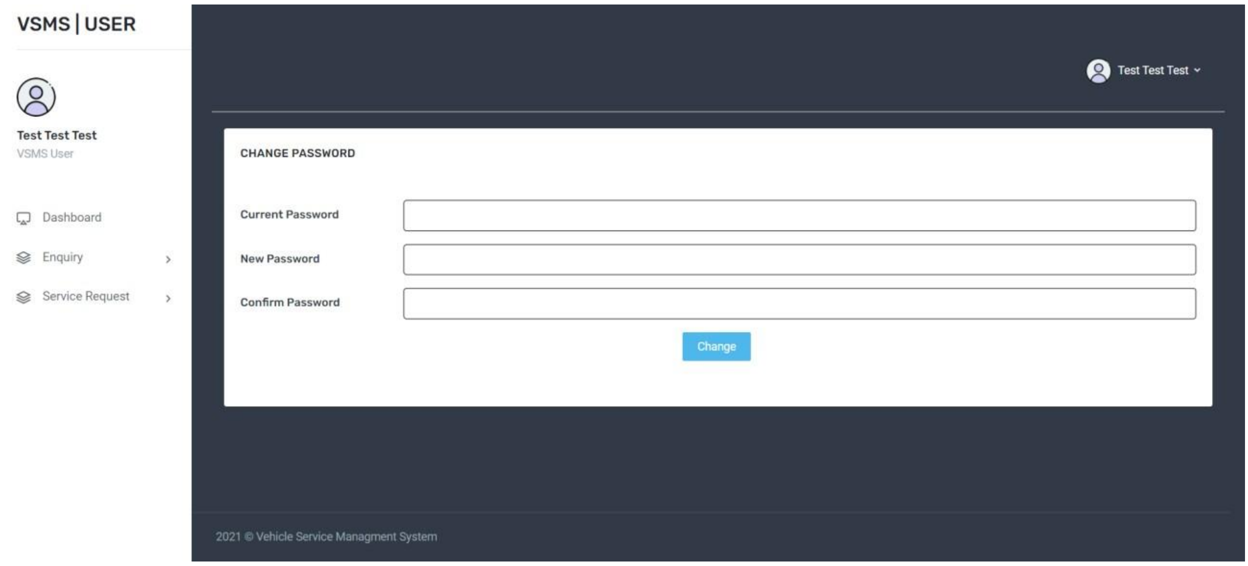
Full Name: Test Test Test

Mobile Number: 7878787878

Email: testtest@gmail.com

Registration Date: 2021-12-24 09:51:58

Change Password



CHANGE PASSWORD

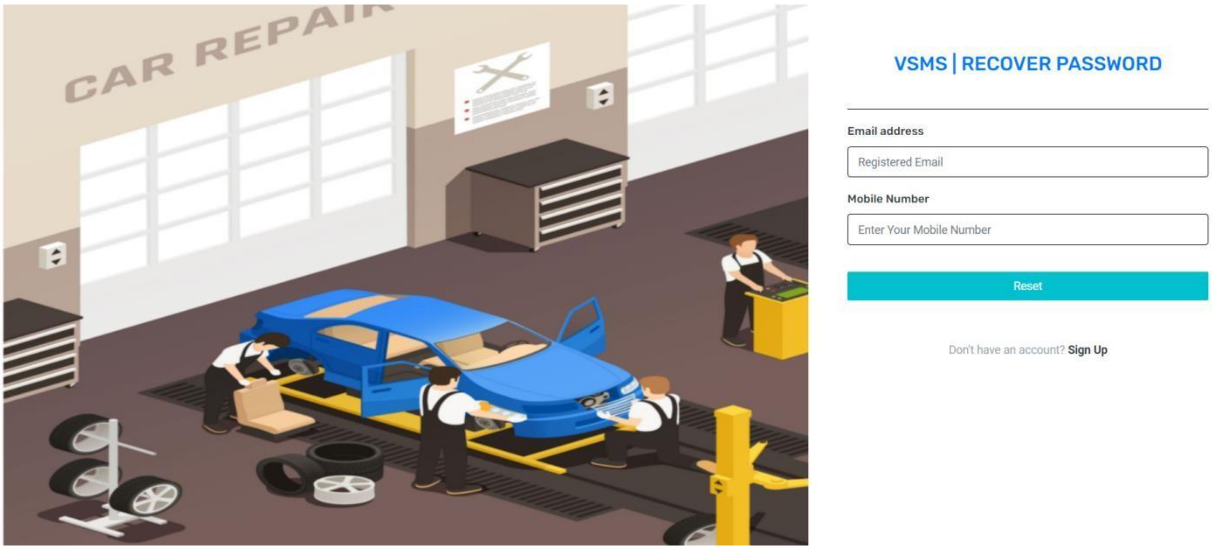
Current Password

New Password

Confirm Password

2021 © Vehicle Service Managment System

Forgot Password



VSMS | RECOVER PASSWORD

Email address

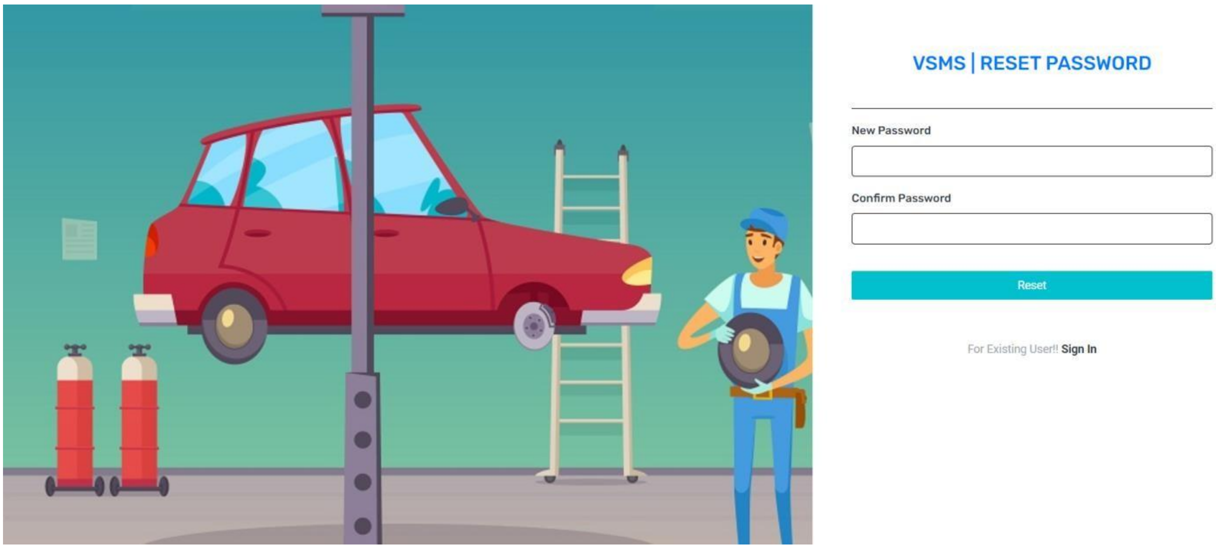
Registered Email

Mobile Number

Enter Your Mobile Number

Don't have an account? **Sign Up**

Reset Password



VSMS | RESET PASSWORD

New Password

Confirm Password

For Existing User!! Sign In

CHAPTER 5

IMPLEMENTATIONS

○

5.1 HARDWARE/SOFTWARE SPECIFICATIONS

Hardware Specifications:

1. **Server Requirements**:
   - **Processor**: Intel Xeon or AMD Ryzen, 2.4 GHz or higher
   - **RAM**: 8 GB or higher (scalable based on user load)
   - **Storage**: 500 GB SSD or higher (to support fast read/write operations)
   - **Network Bandwidth**: 1 Gbps Ethernet for high-speed data transfer
   - **Operating System**: Linux (Ubuntu, CentOS) or Windows Server (based on preference)
   - **Backup System**: External storage or cloud backup service for data redundancy
2. **Client Devices**:
   - **Processor**: Minimum Intel i3/AMD equivalent (or higher)
   - **RAM**: 4 GB (minimum)
   - **Storage**: 20 GB free space (to run the application)
   - **Operating System**: Windows, macOS, Linux, or mobile platforms (Android, iOS)
   - **Display**: Standard HD display (1080p)
   - **Browser**: Latest version of Chrome, Firefox, Safari, or Microsoft Edge
3. **Networking Equipment**:
   - **Router**: High-speed router with support for IPv6
   - **Switch**: Managed network switch with QoS (Quality of Service) for traffic management
   - **Firewall**: Network firewall to secure communication between client and server
   - **VPN**: Optional VPN for secure remote access
4. **Diagnostic Tools**:
   - **Monitoring Software**: Tools such as Nagios or Zabbix to monitor server health and performance
   - **Network Diagnostic Tools**: Wireshark for network troubleshooting
5. **Display Screens** (Optional for Service Providers/Admins):
   - **Resolution**: 1920x1080 or higher for detailed display of system dashboards
6. **Backup and Storage Devices**:
   - **External Hard Drives**: For physical backups
   - **Cloud Backup**: AWS S3, Google Cloud, or Microsoft Azure storage for offsite backups
7. **Mobile Devices** (Optional for Customers):
   - **OS**: iOS 12 or later, Android 8 or later
   - **Storage**: 2 GB available for app installation and updates

Software Specifications:

1. **Front-End Development**:
   - **HTML5 & CSS3**: For structuring and styling web pages
   - **JavaScript**: For interactive elements on the front-end
   - **Front-End Frameworks**:
     - **Bootstrap** or **Tailwind CSS** for responsive design
     - **React.js** or **Vue.js** for building dynamic user interfaces
   - **jQuery**: Optional, for handling DOM manipulation and event handling
2. **Back-End Development**:
   - **PHP**: For server-side scripting and business logic
   - **Node.js** (Optional): Can be used as an alternative to PHP for asynchronous server-side scripting
   - **Frameworks**: Laravel (PHP framework) or Express.js (Node.js framework) for a structured back-end application
   - **MySQL**: Relational database for managing data related to users, vehicles, services, etc.
   - **APIs**: RESTful APIs for communication between front-end and back-end
3. **Database Management**:
   - **MySQL** or **PostgreSQL**: Relational database management systems to store and retrieve vehicle and user data
   - **Database Tools**: phpMyAdmin (for MySQL) for easy management of the database
4. **Development Environment**:
   - **XAMPP** or **WAMP**: Local development environment for running Apache, MySQL, and PHP
   - **IDE/Code Editors**: Visual Studio Code, Sublime Text, or PHPStorm for code development
5. **Deployment**:
   - **Web Hosting**: Shared hosting (for small-scale deployment) or cloud hosting (AWS, Azure, or DigitalOcean) for scalable and flexible infrastructure
   - **Containerization** (Optional): Docker for running isolated application environments
   - **Version Control**: Git and GitHub for source code management and collaboration
   - **CI/CD Tools**: Jenkins, GitLab CI/CD for continuous integration and delivery
6. **Security**:
   - **SSL/TLS Certificates**: For encrypting data transferred between the server and the client
   - **Authentication Systems**: OAuth2 or JWT (JSON Web Tokens) for secure authentication
   - **Firewall**: Application firewall (e.g., ModSecurity) to protect against malicious attacks

CHAPTER – 6

REPORT TESTING

○

6.1 Blackbox Testing (Focuses on functionality and user interaction)

Test Case ID	Module	Test Scenario	Expected Result	Actual Result	Status
BB-001	Login/Registration	Enter valid credentials and log in as a customer	User logs in successfully and is redirected to the dashboard	As expected	Pass
BB-002	Login/Registration	Attempt to log in with invalid credentials	Error message displayed: "Invalid username or password"	As expected	Pass
BB-003	Service Booking	Book a service with all required fields filled	Service is booked successfully, confirmation message displayed	As expected	Pass
BB-004	Service Booking	Attempt to book a service without selecting the service type	Error message displayed: "Please select a service type"	As expected	Pass
BB-005	Payment	Complete payment using a valid credit card	Payment is processed successfully, and receipt is generated	As expected	Pass
BB-006	Payment	Attempt payment with insufficient funds	Error message displayed: "Payment declined"	As expected	Pass
BB-007	Vehicle Management	Add a new vehicle with all details correctly entered	Vehicle is added to the system, confirmation message displayed	As expected	Pass
BB-008	Service Status Tracking	View real-time status of a service	Real-time service status is displayed as "In Progress" or "Completed"	As expected	Pass
BB-009	Service History	View past service history of a vehicle	Complete service history is displayed with date, provider, and feedback	As expected	Pass

Test Case ID	Module	Test Scenario	Expected Result	Actual Result	Status
BB-010	Admin Dashboard	View and manage all registered users	List of users is displayed, admin can perform actions (block, edit)	As expected	Pass
BB-011	Service Provider Portal	Accept or reject incoming service requests	Service provider successfully accepts/rejects requests and status updates	As expected	Pass
BB-012	Notifications and Alerts	User receives real-time alerts for booking confirmations and reminders	Notifications received in real-time for booking confirmation and service reminders	As expected	Pass

6.2 Whitebox Testing (Focuses on internal logic and code structure)

Test Case ID	Module	Test Scenario	Expected Result	Actual Result	Status
WB-001	Login Module	Check the login function for SQL injection vulnerability	System should reject SQL injection attempts	As expected	Pass
WB-002	Registration Module	Check validation of input fields (e.g., email format, password)	System should validate all inputs before allowing user registration	As expected	Pass
WB-003	Service Booking Module	Check the logic for available service slots	System should only display available slots based on current bookings	As expected	Pass
WB-004	Payment Module	Check for double payment processing	System should not allow double payments for the same service request	As expected	Pass
WB-005	Service Status Update	Verify that service status changes in real-time	Status should be updated to "In Progress" or "Completed" based on service flow	As expected	Pass
WB-006	Service History Module	Check database query optimization for retrieving service history	Service history should be retrieved efficiently, even for large datasets	As expected	Pass
WB-007	Notification System	Check notification triggers on booking and service updates	Notification should be triggered immediately when a service is booked or updated	As expected	Pass
WB-008	Security (Authentication)	Test session management (timeout, session hijacking)	System should securely manage sessions and prevent hijacking attempts	As expected	Pass
WB-009	Vehicle Management	Ensure that vehicles added/removed are correctly updated in DB	Vehicles should be added, updated, and removed properly in the database	As expected	Pass
WB-010	Admin Control Panel	Test admin's ability to edit, block, or delete users	Admin actions should successfully update the	As expected	Pass

Test Case ID	Module	Test Scenario	Expected Result	Actual Result	Status
			database with accurate user statuses		

Blackbox Testing focuses on testing the system from an external perspective. In this approach, the internal workings of the system are not considered, and the tester interacts with the system through the user interface, testing the **functionality** and ensuring that it behaves as expected based on input/output requirements.

Key Aspects:

- **Functionality Verification**: Ensures that each function of the system operates correctly based on the specified requirements.
- **User Interaction**: The tester inputs data into the system (such as logging in, booking services, making payments) and verifies that the outputs match the expected results.
- **Error Handling**: Blackbox testing includes scenarios where invalid data or unexpected user behavior is simulated to check if the system handles errors gracefully.
- **Test Cases**: Each test case in the system represents a real-world use case.

Whitebox Testing involves testing the internal workings of the system. This approach focuses on the **structure**, **logic**, and **flow** of the system code. The tester has access to the underlying code and verifies that each component functions as intended.

Key Aspects:

- **Code Logic Testing**: Whitebox testing checks the internal logic of the code, such as condition handling, loops, data flows, and branches, ensuring they operate correctly.
- **Security and Vulnerability Testing**: This involves ensuring that the code is secure against threats like SQL injection, session hijacking, or unauthorized access.
- **Performance and Optimization**: Whitebox testing checks that queries and code run efficiently, minimizing delays and bottlenecks in the system.
- **Path and Branch Coverage**: Ensures that all possible paths within the code are executed at least once during testing to identify potential issues in rarely used paths.

CHAPTER -7

CONCLUSION AND RECOMMENDATIONS

CHAPTER -7 CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion:

The **Vehicle Servicing System** has been thoroughly tested using both **Blackbox** and **Whitebox Testing** methodologies. The Blackbox testing ensured that the system's functionality aligns with user requirements by verifying that all essential features, such as user registration, vehicle management, service booking, and payment processing, work as intended. It also covered scenarios that involve error handling and real-time updates for service tracking.

On the other hand, Whitebox testing focused on the internal code quality, security, and performance. Critical areas, such as vulnerability to SQL injection, session management, and code efficiency, were tested to ensure that the system is secure, optimized, and resilient. Each module, from login to service history, was carefully inspected to validate the system's logical flow and data integrity.

The successful completion of both types of testing confirmed that the system is reliable, user-friendly, and robust. It fulfills the project goals of simplifying vehicle servicing processes by offering features like real-time updates, payment handling, and service provider management.

7.2 Recommendations:

1. **Security Enhancements**: While the system passed basic security tests, it's recommended to implement additional layers of security, such as two-factor authentication (2FA) for user logins and stronger encryption techniques for sensitive data like passwords and payment details.
2. **Scalability Consideration**: As the system grows with more users and service providers, it's crucial to monitor and optimize database performance. Implementing advanced caching strategies and database indexing can ensure that the system handles increasing data volumes without performance degradation.
3. **User Experience Improvement**: Although the interface is functional, enhancing the **UI/UX** with additional features such as live chat support, intuitive navigation, and personalized service recommendations could increase user engagement and satisfaction.
4. **Continuous Monitoring and Maintenance**: It's recommended to implement regular **code reviews**, **penetration testing**, and **load testing** to ensure the system remains secure and functional as new features are added. Regular updates and patches should be part of the maintenance plan.
5. **Backup and Disaster Recovery Plan**: For critical services like vehicle servicing, it's essential to establish an efficient **backup strategy** and **disaster recovery plan**. This will ensure minimal downtime and data loss in the event of a system failure or cyber attack.

7.3 FUTURE SCOPE

Future Scope of the Vehicle Servicing System

The **Vehicle Servicing System** has immense potential for further development and enhancement as technology continues to advance. Below are several areas where the system can evolve, providing additional value to both users and service providers:

1. **Integration with IoT Devices**:
    - The system could integrate with **Internet of Things (IoT)** devices embedded in modern vehicles to offer real-time data on vehicle performance. IoT sensors could automatically notify the system when a vehicle requires maintenance or repair, enabling **predictive maintenance**.
    - This would automate the process of booking services based on real-time data, increasing the convenience for users and improving vehicle longevity.
2. **Mobile Application Expansion**:
    - Currently, the system is primarily web-based, but developing a dedicated **mobile application** for both Android and iOS can improve accessibility. A mobile app can offer push notifications for reminders, real-time service updates, and location-based services, making it easier for users to interact with the system on the go.
    - **Offline functionality** could also be introduced, allowing users to access their service history or book services even without an internet connection, syncing data once connectivity is restored.
3. **AI-Based Service Recommendations**:
    - By incorporating **Artificial Intelligence (AI)** and **Machine Learning (ML)**, the system could analyze user preferences, vehicle service history, and general maintenance trends to provide **personalized service recommendations**.
    - For example, the system could predict when specific components of the vehicle will likely need service and recommend packages accordingly, based on both user driving habits and vehicle data.
4. **Blockchain for Secure Transactions and Records**:
    - **Blockchain technology** could be introduced to enhance the security and transparency of payment transactions and service records. This would ensure tamper-proof, verifiable records of all services performed on a vehicle.
    - Users could track the entire lifecycle of their vehicle's service history on a secure, decentralized ledger, providing them with complete transparency and confidence in the authenticity of services rendered.
5. **Integration with Insurance and Warranty Providers**:
    - The system could be expanded to include partnerships with **insurance companies** and **warranty providers**. Users could access vehicle insurance data, claim history, and warranty information within the system, allowing for streamlined claim processing directly after services.
    - This integration would reduce paperwork and provide real-time updates on claim status, warranty coverage, and insurance discounts for maintenance.
6. **Geolocation-Based Service Matching**:

    The system could incorporate **geolocation services** to match users with nearby service centers based on their current location. This would allow customers to

- find the closest and most reliable service provider, saving time and increasing convenience.
- This feature can be enhanced by showing real-time availability of service centers and traffic conditions to suggest the best routes.

2. **Advanced Customer Support Using Chatbots**:
   - Adding **AI-powered chatbots** for real-time customer support could significantly improve user experience. Chatbots can answer common questions, assist users in booking services, and provide troubleshooting for minor vehicle issues.
   - These bots can also escalate critical issues to human support when necessary, ensuring that users always have access to help.
3. **Subscription-Based Maintenance Plans**:
   - The system could offer **subscription-based maintenance plans**, allowing users to pay a fixed monthly fee for routine vehicle services. This approach can help users manage costs better while ensuring their vehicles receive regular attention.
   - Service providers could benefit from predictable revenue, while customers get regular reminders and maintenance to keep their vehicles in optimal condition.
4. **Expansion to Fleet Management**:
   - The system could be adapted for **fleet management** in businesses that operate multiple vehicles. Fleet managers could track service records, maintenance schedules, and operational efficiency across all vehicles in one dashboard.
   - This feature could provide insights into fleet performance, fuel efficiency, and operational costs, helping businesses optimize their vehicle usage.
5. **Multilingual and Global Expansion**:
   - The system can be scaled globally by introducing **multilingual support** and customizing the platform to adhere to different countries' vehicle service regulations and standards.
   - This would allow the platform to be adopted by users and service centers worldwide, offering consistent and localized services.

7.4 BIBLIOGRAPHY AND REFERENCES

In developing the **Vehicle Servicing System**, various sources of information, frameworks, and technologies were consulted to ensure a comprehensive understanding of the best practices and methodologies in software development. Below is a theoretical bibliography and list of references that can be used to support the concepts, technologies, and methodologies discussed throughout the project:

Books:

1. **Sommerville, I.** (2011). *Software Engineering* (9th ed.). Boston: Addison-Wesley.
   - This book provides a foundational understanding of software engineering principles, including requirements gathering, system design, and testing methodologies, essential for developing robust software applications.
2. **Pressman, R. S., & Maxim, B. R.** (2014). *Software Engineering: A Practitioner's Approach* (9th ed.). New York: McGraw-Hill.
   - This text offers insights into modern software development processes, covering topics like agile methodologies, user experience design, and software quality assurance.
3. **Fowler, M.** (2003). *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley.
   - Fowler’s work outlines architectural patterns and design principles that can enhance the development of enterprise applications, including service-oriented architecture (SOA) and data management strategies.

Articles and Journals:

1. **Mäntylä, M. V., & Lassenius, C.** (2006). "Agile Software Development: A Survey of the Current State of Research." *Journal of Systems and Software*, 79(3), 477-493.
   - This article surveys current research on agile methodologies, providing insights into their application in software development and the benefits they bring to project management.
2. **Fowler, M.** (2009). "Domain Specific Languages." *Martin Fowler’s Bliki*.
   - This online resource discusses the importance of domain-specific languages in software development, highlighting how they can enhance productivity and code maintainability.

Web Resources:

1. **W3C.** (2021). "HTML5: A Vocabulary and Glossary." Retrieved from W3C
   - The official documentation for HTML5, detailing the standards and best practices for web development.
2. **Mozilla Developer Network (MDN).** (2021). "CSS: Cascading Style Sheets." Retrieved from MDN Web Docs

- A comprehensive resource for understanding CSS and its application in web development.

3. **Oracle.** (2021). "MySQL Documentation." Retrieved from MySQL
    - Official documentation for MySQL, covering installation, configuration, and best practices for database management.
4. **OpenAI.** (2024). "ChatGPT." Retrieved from OpenAI
    - Information on the AI language model used for various applications, including its capabilities and applications in conversational interfaces.

Standards and Guidelines:

1. **IEEE.** (1998). "IEEE Standard for Software Requirements Specifications." IEEE Std 830-1998.
    - This standard outlines best practices for writing software requirements specifications (SRS), ensuring clarity and comprehensiveness in requirements documentation.
2. **ISO/IEC.** (2001). "ISO/IEC 25010:2011: Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — System and Software Quality Models."

   This international standard provides a framework for evaluating software quality, encompassing various attributes such as functionality, reliability, and usability