ROAD RESCUE

Submitted by

CHIRAG THAKUR [RA2111003010071]
S RAHUL [RA2111003010099]

Under the Guidance of

Dr. Santhanakrishnan C

Assistant Professor (Sr.G), Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUE – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. **RA2111003010071** Certified to be the bonafide work done by **CHIRAG THAKUR** of II Year/IV Sem B.Tech Degree Course in the **Practical Software Software Engineering and Project Management 18CSC206J** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**, Kattankulathur during the academic year 2022 – 2023.

11/5/23

M. Pushpalatha

LAB INCHARGE

Name of the Faculty: Dr. C Santhanakrishnan

Designation: Assistant Professor (Sr.G)

Department of Computing Technologies

SRMIST – KTR.

Head of the Department

Name of the Faculty: Dr.M.Pushpalatha

Designation: Professor and Head

Department of Computing Technologies

SRMIST – KTR.

Date:

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF TABLES	
	LIST OF ABBREVIATIONS	
1	**INTRODUCTION**	**1**
2	**STAKEHOLDERS & PROCESS MODELS**	**2**
	• 2.1 STAKEHOLDERS	**2**
	• 2.2 PROCESS MODELS	**2**
3	**IDENTIFYING REQUIREMENTS**	**4**
	• 3.1 FUNCTIONAL REQUIREMENTS	**4**
	• 3.2 NON-FUNCTIONAL REQUIREMENTS	**4**
4	**PROJECT PLAN & EFFORT**	**5**
	• 4.1 PROJECT PLAN	**5**
	• 4.2 EFFORT AND COST ESTIMATION	**6**
5	**WORK BREAKDOWN STRUCTURE & RISK ANALYSIS**	**7**
	• 5.1 WORK BREAKDOWN STRUCTURE	**7**
	• 5.2 RISK ANALYSIS	**8**
	o 5.2.1 SWOT ANALYSIS	**8**
	o 5.2.2 RMMM	**8**
6	**SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM**	**9**
	• 6.1 SYSTEM ARCHITECTURE DIAGRAM	**9**
	• 6.2 USE CASE	**9**
	• 6.3 CLASS DIAGRAM	**10**
7	**DATA FLOW DIAGRAM**	**11**
	• 7.1 DATA FLOW LEVEL 0 DIAGRAM	**11**
	• 7.2 DATA FLOW LEVEL 1 DIAGRAM	**11**
8	**ENTITY RELATIONSHIP DIAGRAM**	**12**
9	**SEQUENCE & COLLABORATION DIAGRAM**	**13**
	• 9.1 SEQUENCE DIAGRAM	**13**

- 9.2 COLLABORATION DIAGRAM 13

10 DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE **14**

- 10.1 TESTING FRAMEWORK 14
  - 10.1.1 TYPES OF TESTING 14
  - 10.1.2 METHODOLOGY 14
  - 10.1.3 TEST DELIVERABLES 15
  - 10.1.4 TOOLS REQUIRED 15
- 10.2 USER INTERFACE 16

11 TEST CASES & REPORTING **17**

- 11.1 FUNCTIONAL TEST CASES 17
- 11.2 NON-FUNCTIONAL TEST CASES 18

12 ARCHITECTURE/DESIGN/FRAMEWORK/IMPLE-MENTATION **19**

- 12.1 ARCHITECTURAL DESIGN 19
- 12.2 ARCHITECTURAL DESIGN 19
- 12.3 IMPLEMENTATION 20

CONCLUSION

REFERENCES

APPENDIX

- CODE
- SAMPLE SCREENSHOTS

ABSTRACT

Our society is heavily reliant on transportation, and the automobile is a crucial aspect of it. However, owning a car comes with a host of problems, ranging from mechanical breakdowns to accidents, theft, and more. Towing services, car washes, and insurance coverage are just a few examples of the services that exist to address these issues. Nevertheless, in today's world, it is critical to provide a platform that integrates all of these services in one place, allowing users to access them with ease. The purpose of this project is to develop a comprehensive web application that provides users with a one-stop-shop for all of their automobile-related needs. The platform will include features such as towing service request, car wash booking, insurance coverage plans, and accident coverage options. This project aims to simplify the car-ownership experience for users by providing a user-friendly platform to access all of these services.

Keywords:

Automobile Services; Roadside assistance; Towing; Vehicle Maintenance; Vehicle Care

LIST OF FIGURES

FIGURE	TITLE	PAGE NO
2.1	Waterfall Model	3
5.1	Work Breakdown Structure	7
6.1	System Architecture	9
6.2	Use Case Diagram	9
6.3	Class Diagram	10
7.1	Data flow level 0 diagram	11
7.2	Data flow level 1 diagram	11
8.1	Entity Relation Diagram	12
9.1	Sequence Diagram	13
9.2	Collaboration diagram	13

LIST OF TABLES

FIGURE	TITLE	PAGE NO
4.1	Project Management Plan	5
4.2	Effort And Cost Estimation	6
5.1	Swot Analysis	12
5.2	RMMM	12
10.1	Tools Required	15
11.1	Functional Test Cases	17
11.2	Non-Functional Test Cases	18

LIST OF ABBREVIATIONS

S.NO	Abbreviation	Meaning
1	ANOVA	Analysis of Variance
2	CI	Confidence Interval
3	CSV	Comma Separated Value
4	FMEA	Failure Modes and Effects Analysis
5	MSE	Mena Squared Error
6	OR	Odds Ratio
7	QA	Quality Assurance
8	RMC	Root Mean Square

CHAPTER 1
INTRODUCTION

In today's world, where everyone is constantly on the move, transportation has become an integral part of our lives. With the increase in the number of vehicles on the roads, it is not surprising that breakdowns, accidents, and other unforeseen events have become more common. These incidents can be stressful and dangerous, leaving drivers stranded and in need of immediate assistance. However, finding a reliable and efficient road side assistance service can be difficult and time-consuming, as drivers may not know where to turn or have to navigate through multiple phone numbers and websites. In such situations, it is crucial to have access to reliable and efficient roadside assistance services that can provide quick and timely support. The aim of this project is to develop a web-based application that provides a comprehensive range of services to car owners and drivers, including towing, car washing, flood coverage, and accident coverage. This application is designed to offer a one-stop solution for all the vehicle-related needs of its users. It aims to provide a user-friendly interface that is easy to navigate and offers quick access to a wide range of services. The web-based application is designed to be accessible from anywhere, at any time. It is compatible with a wide range of devices, including smartphones, tablets, and desktop computers. It also incorporates the latest security features to ensure the safety and privacy of user information. Overall, this project aims to revolutionize the way people think about vehicle-related services. By offering a comprehensive range of services, it aims to provide a one-stop solution for all the vehicle-related needs of its users. With its user-friendly interface and advanced features, this application is poised to become a game-changer in the world of roadside assistance services. Our website has a user-friendly interface that enables customers to easily navigate through the various services offered. The homepage provides customers with all the basic information they need, along with an option to book any service they require. The website also has a blog page that contains news and information about the latest developments in the transportation industry. In addition, we have also included a contact page on our website where customers can get in touch with us to clarify any doubts or issues they may have. To ensure the security of customer information, we have also implemented a login page where customers can create an account and manage their bookings and other details. Overall, our website aims to provide customers with a convenient and reliable platform for all their vehicle-related needs. With our range of services and user-friendly interface, we hope to make the process of owning and maintaining a vehicle easier and hassle-free for our customers.

CHAPTER 2

STAKEHOLDERS & PROCESS MODEL

2.1 STAKEHOLDERS

A Road Rescue project involves several stakeholders who can be affected by the project's outcome. Let's look at some of the key stakeholders in a Road Rescue project:

- **Customers:** The end-users who will use the website to book services, read the blog, and contact the service providers. Their feedback and user experience are crucial to the success of the website.
- **Service Providers:** The businesses or individuals who offer services through the website. Their satisfaction with the platform and the booking process is essential to maintaining their participation in the website.
- **Project Managers:** The individuals responsible for managing the project, ensuring that it stays on schedule and within budget, and communicating with stakeholders.
- **Investors:** The individuals or organizations that have invested capital in the project and have a vested interest in its success. They expect a return on their investment, and their satisfaction with the website's performance is critical.
- **Website Developers:** The team responsible for designing, developing, and testing the website. They must ensure that the website meets the requirements of the stakeholders and is functional and user-friendly.
- **Support Staff:** The individuals responsible for handling customer inquiries, complaints, and technical issues related to the website. They ensure that the customers receive prompt and effective support and that their issues are resolved satisfactorily.

2.2 PROCESS MODEL

Selection of Methodology: Waterfall

The waterfall methodology is sometimes referred to as the "traditional" or "sequential" approach to project management.

Here are the typical phases of the waterfall methodology:

- **Requirements gathering:** In this phase, the project team identifies the requirements and needs of the project stakeholders. This phase often involves research, interviews, surveys, and other techniques to identify the requirements for the project.
- **Design:** In this phase, the project team creates a detailed design of the system or software to be developed. This phase may include creating wireframes, mock-ups, and other design
  documents that specify the user interface, system architecture, and other technical details.
- **Implementation:** In this phase, the actual development work begins. The project team writes the code, tests it, and integrates it into the overall system.
- **Testing:** In this phase, the project team conducts comprehensive testing of the software or system to ensure that it meets the requirements and works as intended. This phase may include functional testing, integration testing, and other types of testing.

- **Deployment:** In this phase, the software or system is deployed to the production environment, where it is made available to users or customers.
- **Maintenance:** In this phase, the project team provides ongoing maintenance and support for the software or system. This may include fixing defects, adding new features, and addressing other issues that arise over time.

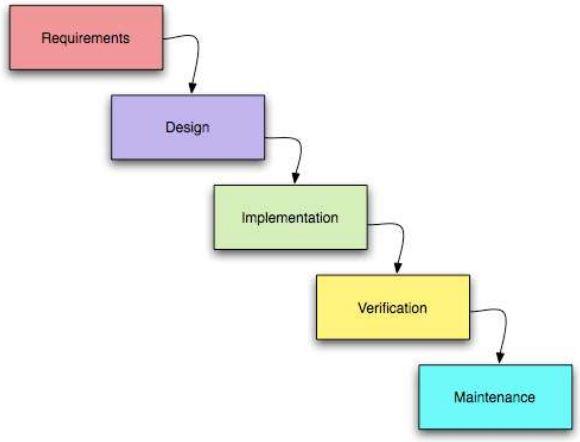


Fig 2.1 Waterfall Model

CHAPTER 3

|IDENTIFYING REQUIREMENTS

3.1 Functional Requirements:

- **User registration:** Users should be able to create an account on the platform, providing their name, contact information, and payment details.
- **Service provider registration**: The service provider should be able to register on the platform, providing their name, contact information, and details of the services they offer.
- **Service requests:** Users should be able to request road side assistance services through the platform, specifying their location, the type of vehicle they have, and the service required.
- **Service matching:** The platform should be able to match service requests with available mechanics in the user's location who are capable of providing the required service.
- **Real-time communication:** The platform should provide a mechanism for real-time communication between users and mechanics, enabling them to exchange information about the service request and coordinate the service.
- **Service tracking:** Users should be able to track the status of their service request in real-time, providing updates on the estimated time of arrival and the progress of the service.
- **Service rating:** Users should be able to rate and provide feedback on the service provided by the mechanic, which can be used to improve the quality of service.

3.2 Non-Functional Requirements:

- **Performance:** The platform should be designed to handle a large volume of requests, providing fast response times and minimal downtime.
- **Security:** The platform should be designed with security in mind, using encryption and other security measures to protect user and mechanic data, as well as payment information.
- **Reliability:** The platform should be designed to be reliable, with robust error handling and fault tolerance, and a backup and recovery plan in place to minimize downtime in the event of a system failure.
- **Scalability:** The platform should be designed to be scalable, able to handle a growing number of users and mechanics as the platform expands.
- **Compatibility:** The platform should be designed to be compatible with a wide range of web browsers, operating systems, and devices, ensuring that it can be used by the broadest possible audience.
- **Maintainability:** The platform should be designed to be maintainable, with clean and well-organized code, good documentation, and a clear separation of concerns.

CHAPTER 4

PROJECT PLAN & EFFORT

4.1 Project Management Plan:

Focus Area	Details
Schedule Management	• Governance Framework • Project Team Structure • Roles & Responsibilities of Team • Project Closure
Scope Management	• Scope Statement • Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) • Define Deliverable • Requirement Change Control
Schedule Management	• Define Milestones • Schedule Control
Cost Management	• Estimate Effort • Assign Team • Budget Control
Quality Management	• Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting • Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management	• Estimate and Manage the need • People: People & Skills Required • Finance: Budget Required • Physical: Facilities, IT Infrastructure
Stakeholder	• Identifying, • Analyzing • Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]

Risk Management	• Identifying • Analyzing • Prioritizing project risks
Procurement Management	Adhering to organization procurement process

Table 4.1 Project Management Plan

4.2 Effort and Cost Estimation:

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	create the design for front end	Drawing analyzing the basic design and understanding how the UI will look.	3	2000
	coding the front end	Using logical analytical techniques to design the front end using various tools	6	6000
	merging front end end to back end	Merging the front to back end (database) to store the data	1	1000
Creating and managing a backend to store and analyze data	Creating and managing a backend to store and analyze data	creating a structure using dbms software and analyzing the data	4	4000

Table 4.2 Effort and Cost Estimation

CHAPTER 5
WORK BREAKDOWN STRUCTURE & RISK ANALYSIS

5.1 Work Breakdown Structure:

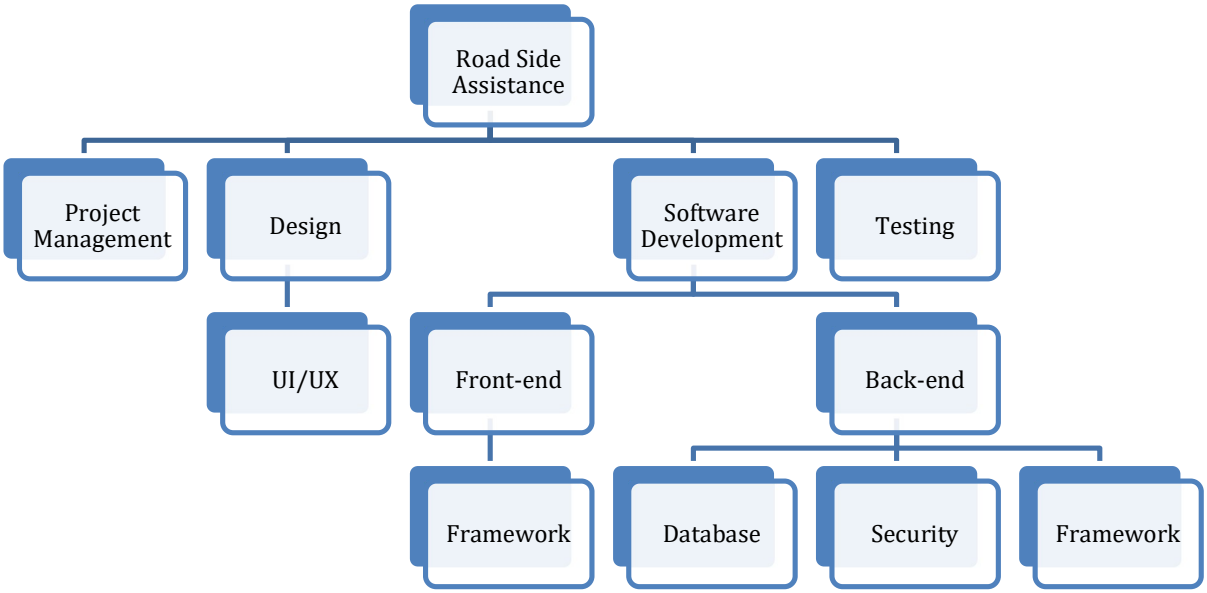


Fig 5.1 Work Breakdown Structure

1) Road Side Assistance
2) Project Management
3) Design
    a) UI/UX
4) Software Development
    a) Front-end
        (i) Framework
    b) Back-end
        (i) Database
        (ii) Security
        (iii)Framework
5) Testing

5.2 Risk Analysis:

5.2.1 SWOT Analysis:

STRENGTHS	WEAKNESS
▪ User friendly interface. ▪ Skilled team members. ▪ Brings jobs to the market. ▪ Data analysis is done.	▪ Lack in staff. ▪ Unreliable third-party software is used. ▪ Poor hosting services.
OPPORTUNITIES	**THREATS**
▪ Different job roles. ▪ Usage of new technology. ▪ Innovative marketing strategies.	▪ Security system concerns and hackers. ▪ Competitors. ▪ Lack of regulation. ▪ Advancements in technology.

Table 5.1 SWOT Analysis

5.2.2 RMMM:

RESPONSE	STRATEGY	EXAMPLES
AVOID	Risk avoidance is a strategy where the project team takes action to remove any threats from the project.	▪ Changing the execution plan ▪ Extending the timeline
TRANSFER	Risk transference involves shifting the risk threat and impact to a third party.	▪ Issuing Contracts ▪ Insurance
MITIGATE	Risk mitigation is the strategy where team takes an action to reduce risk occurrence.	▪ Increasing Testing ▪ Reducing Complexity
ACCEPT	The team acknowledges the risk and its impact and decide to deal with it when it occurs.	▪ Resource Management ▪ Budget Management

Table 5.2 RMMM

CHAPTER 6
SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM

6.1 System Architecture:

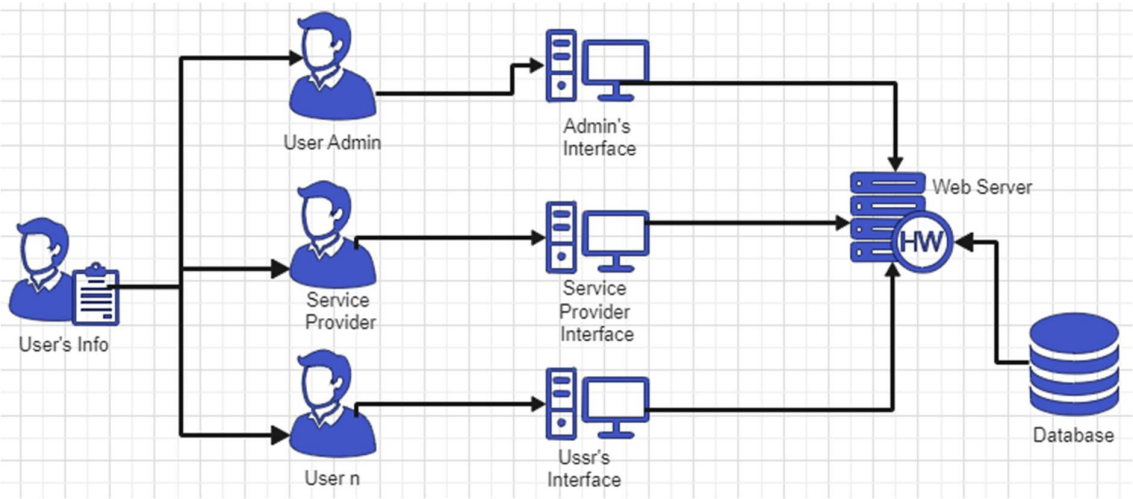


Fig 6.1 System Architecture

6.2 Use case diagram:

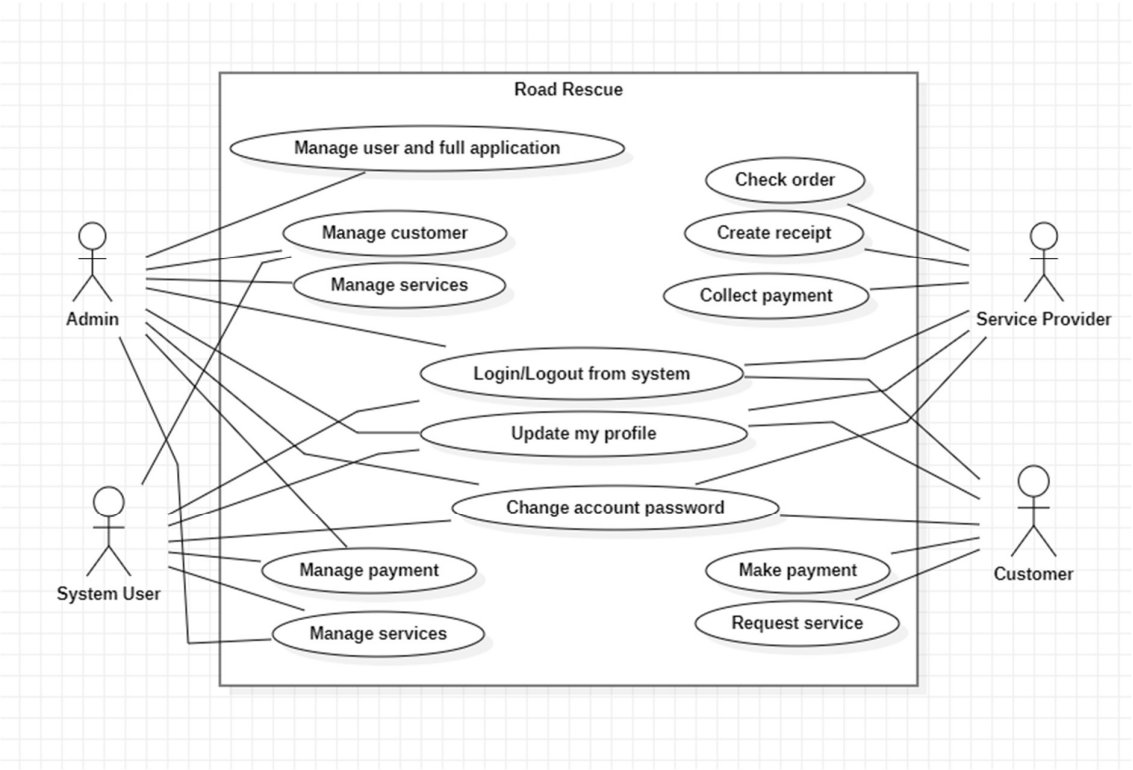


Fig 6.2 Use Case Diagram

6.3 Class diagram:

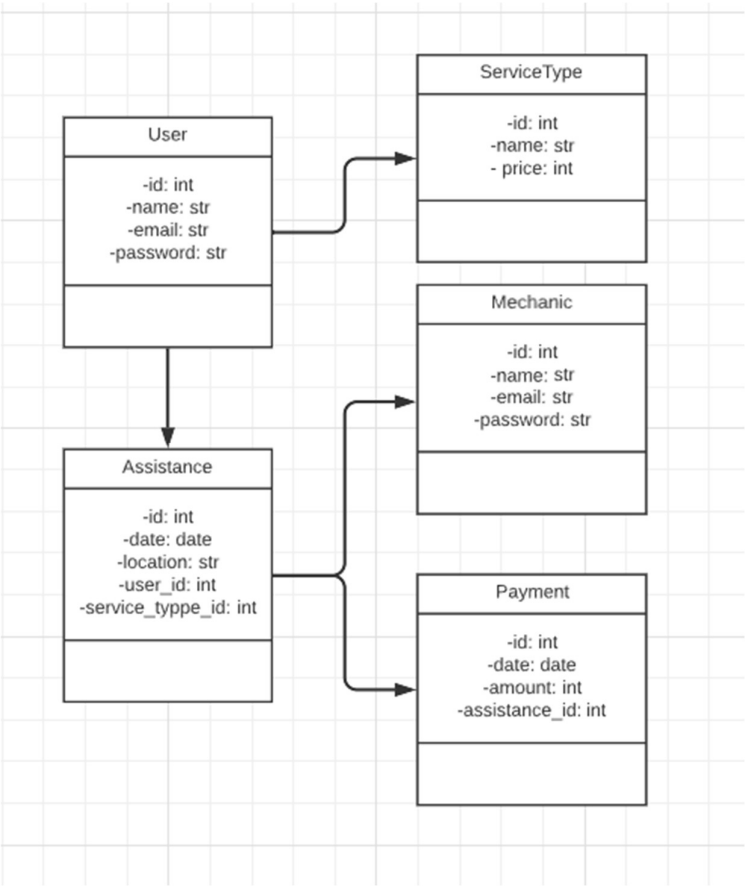


Fig 6.3 Class Diagram

CHAPTER 7
DATA FLOW DIAGRAM

7.1 Data flow level 0 diagram:

Fig 7.1 Data Flow level 0 Diagram

7.2 Data flow level 1 diagram:

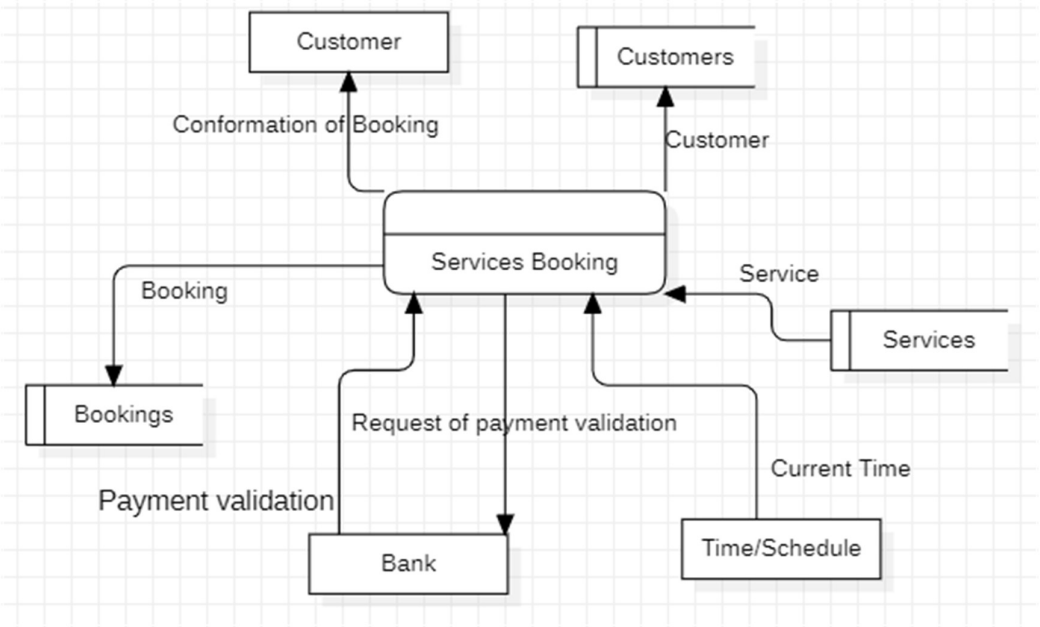


Fig 7.2 Data Flow level 1 Diagram

CHAPTER 8
ENTITY RELATIONSHIP DIAGRAM

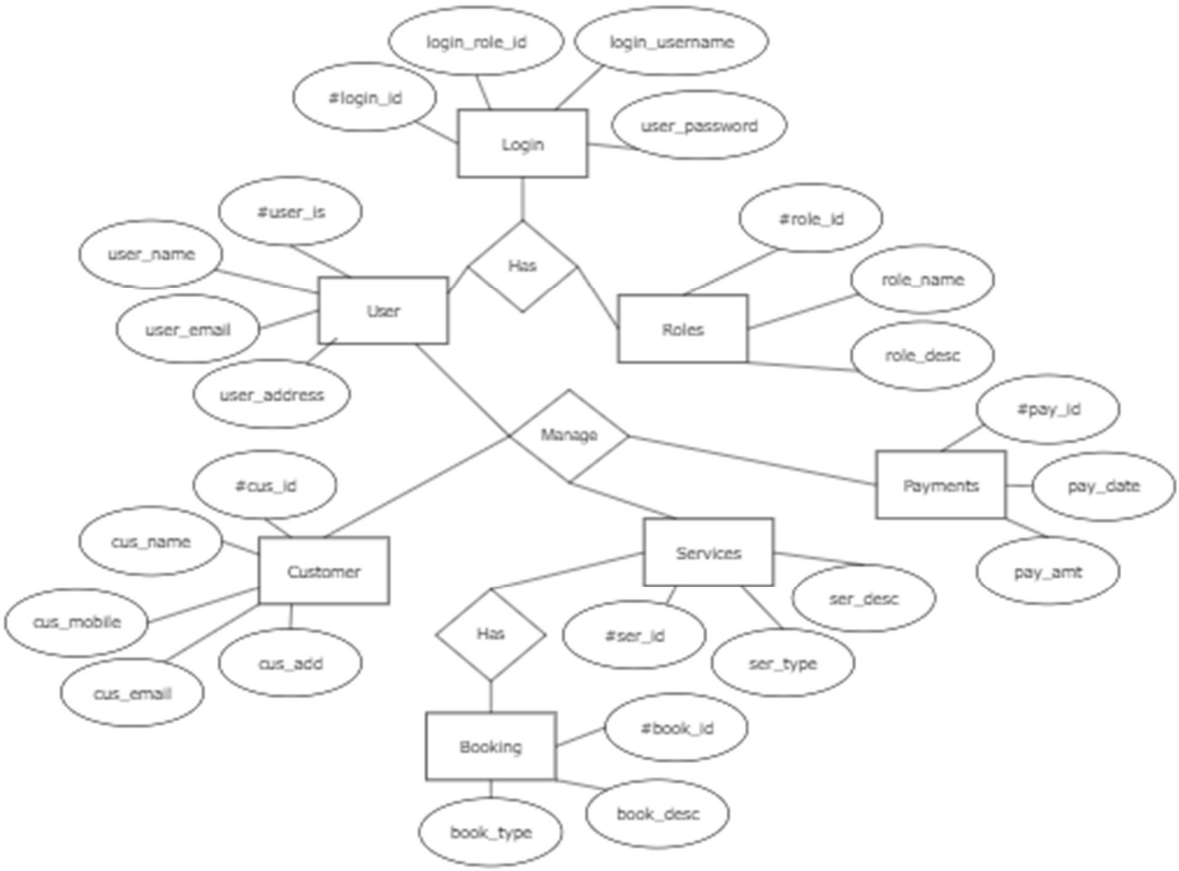


Fig 8.1 Entity Relationship Diagram

CHAPTER 9
SEQUENCE & COLLABORATION DIAGRAM

9.1 Sequence diagram:

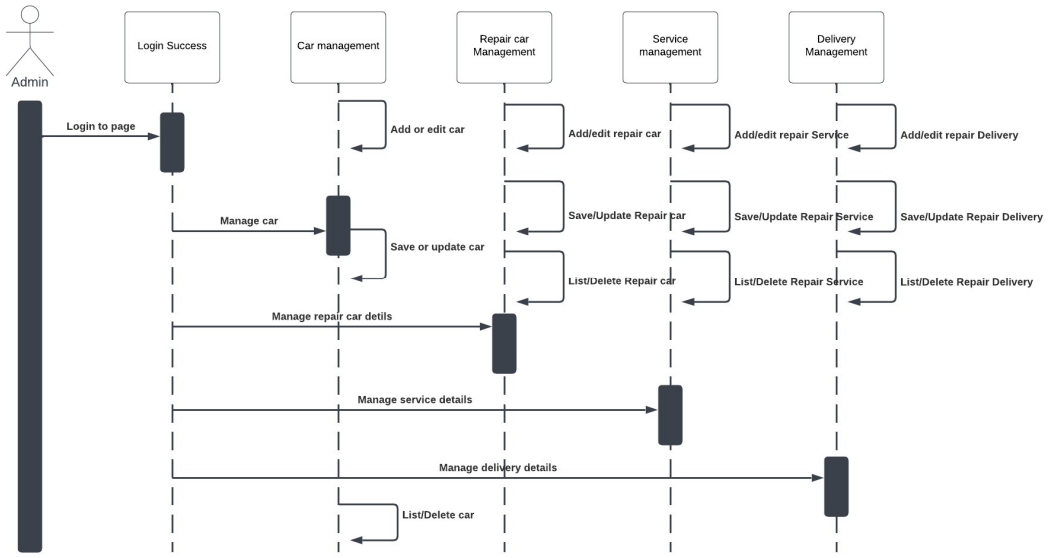


Fig 9.1 Sequence Diagram

9.2 Collaboration diagram:

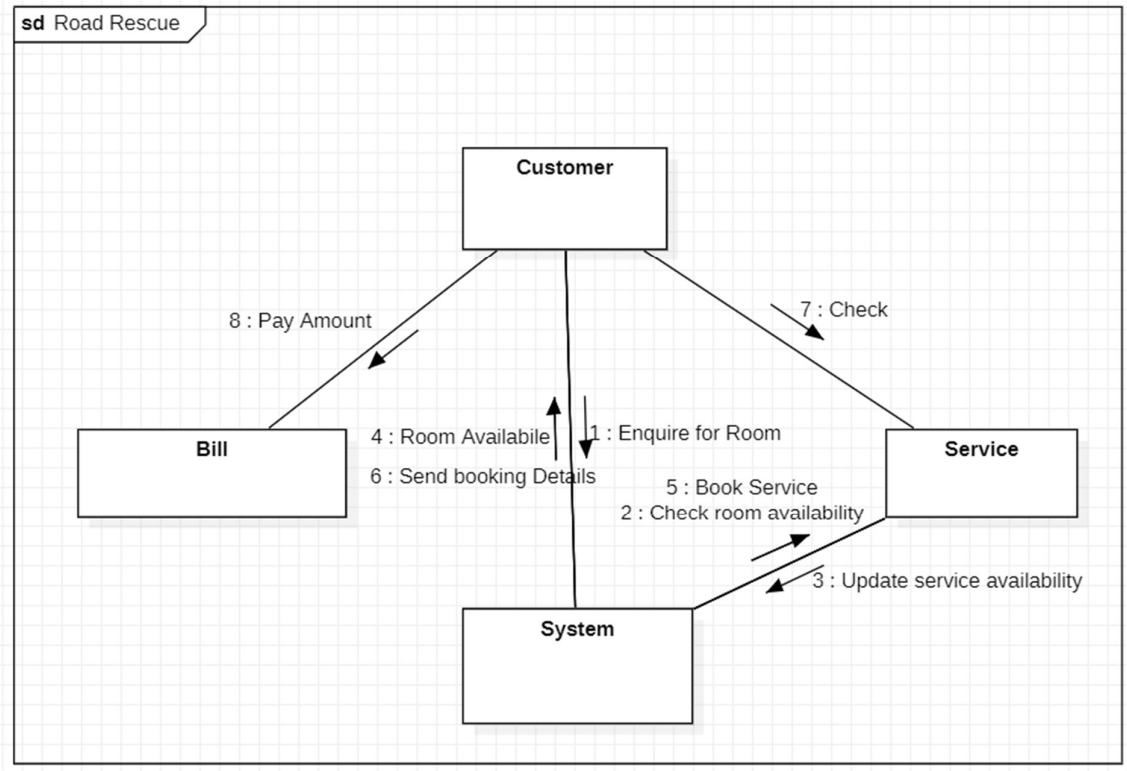


Fig 9.2 Collaboration Diagram

CHAPTER 10
DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE

10.1 Testing Framework:

Scope of testing:

The scope of testing for the Road Rescue website includes all the features and functions that have been implemented, including user registration, login, service request, service provider search, service provider selection, service provider response, service completion, payment processing, and customer support.

10.1.1 Types of testing:

The following types of testing will be performed as part of the testing framework:

- Unit testing: This will involve testing individual units of code to ensure that they are working as expected.
- Integration testing: This will involve testing the integration of different components of the website to ensure that they are working together correctly.
- System testing: This will involve testing the entire system to ensure that it meets the functional and non-functional requirements.
- Acceptance testing: This will involve testing the website from the perspective of the end-users to ensure that it meets their requirements.
- Performance testing: This will involve testing the website's performance under different load conditions to ensure that it can handle the expected traffic.
- Security testing: This will involve testing the website's security features to ensure that it is protected against common security threats.
- Regression testing: This will involve testing the website after changes or updates have been made to ensure that existing functionality has not been affected.

10.1.2 Methodology:

The testing framework will follow an iterative and incremental approach, with testing performed at each stage of the development process. The testing process will be carried out by a dedicated testing team, with support from the development team. The testing methodology will involve the following steps:

- Test planning: This will involve defining the scope of testing, identifying test cases, and developing a test plan.
- Test design: This will involve creating test scenarios and test cases based on the functional and non-functional requirements.
- Test execution: This will involve running the tests and recording the results.
- Test reporting: This will involve documenting the test results and communicating them to the development team.
- Test closure: This will involve reviewing the testing process and identifying areas for improvement.

10.1.3 Test deliverables:

The following test deliverables will be produced as part of the testing framework:

- Test plan: This will document the scope of testing, the types of testing to be performed, and the testing methodology to be followed.
- Test scenarios: This will document the various scenarios that will be tested.
- Test cases: This will document the specific test cases that will be run.
- Test scripts: This will document the scripts that will be used to automate the testing process.
- Test results: This will document the results of the tests, including any defects or issues that were identified.

10.1.4 Tools Required:

To implement the testing framework for the Road Rescue project, the following tools may be useful:

- Selenium Web Driver: A popular automation testing tool used for web applications.
- Apache JMeter: A performance testing tool used for load testing, stress testing, and functional testing.
- Jenkins: A continuous integration and continuous delivery (CI/CD) tool used for automating software development processes.
- Word Template: A Word template is a pre-designed document format that can be used to create consistent and professional-looking test plans and other project documentation.

Category	Methodology	Tools Required
Functional Requirements	Manual	Word Template
Integration testing	Manual	Selenium Web Driver
Acceptance testing	Manual	User acceptance testing checklist
Performance testing	Automatic	Apache JMeter
Security testing	Manual	OWASP ZAP
Regression testing	Automatic	Jenkins CI/CD pipeline

Table 10.1 Tools Required

10.2 User Interface:

1. **Home Page:**

- This will be the main entry point for users, providing them with all the necessary information about the service.
- The page will have a clear and concise messaging that conveys the purpose of the service and what it offers to users.
- There will be a prominent call-to-action button to book a service, making it easy for users to get started with the service.
- The page will be optimized for different devices and screen sizes, ensuring a seamless experience for users on desktop and mobile devices.

2. **Blog Page:**

- This page will contain all the latest news and updates about the service, as well as additional information about the company and its offerings.
- The page will be designed to be visually appealing and easy to navigate, with clear and concise messaging about each blog post.
- Users will be able to easily find and access the blog posts they are interested in, with options to sort or filter the posts by topic or date.

3. **Contact Page:**

- This page will provide users with a way to contact the company directly, whether it's through email, phone, or a contact form.
- The page will be designed with clear and concise messaging that conveys the purpose of the page and how users can get in touch with the company.
- The contact form will be easy to use and understand, with fields for users to enter their name, email address, and message.

4. **Login Page:**

- This page will allow users to log in to their account and access any features or information that requires authentication.
- The page will have clear and concise messaging that conveys the purpose of the page and how users can log in to their account.
- The login form will be easy to use and understand, with fields for users to enter their username and password.
- The page will be designed to be secure and protect user data, with options for users to reset their password if necessary.

Overall, the user interface will be designed with the user in mind, providing a seamless and intuitive experience for users to access the service and its features.

CHAPTER 11
TEST CASES & REPORTING

11.1 Functional Test Cases:

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Book a service	Verify that a service can be booked	1. Log into the System	A new service will be booked	New booked service is displayed	Pass	Success
			2. Navigate to the profile	-	Navigation possible	Pass	Success
			3. Enter valid customer information	It should be created and saved to the system	Service is created and saved	Pass	Success
2.	See an booked service	Verify that booked service is showing	1. Log into the System	The user should be able to navigate to the service details page	Navigation possible	Pass	Success
			2. Search for an existing service in the system using various search criteria	The search results should display the relevant service(s) and the user should be able to select and view the details	Relevant service and details displayed	Pass	Success
			3. Navigate to the details for the selected service	The service details should be displayed, including customer information	Service has all informatio n	Pass	Success

Table 11.1 Functional Test Cases

11.2 Non-Functional Test Cases:

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remark
1.	Performance Testing (Production Environment)	Verify system response time during high traffic periods	1. Generate a large number of services and save them to the system	System should respond within 2-5 seconds for eachrequest, even during high traffic periods	Optimum Response	Pass	Success
			2. Simulate high levels of concurrent user activity on the system	Swift usage	No server down	Pass	Success
			3. Monitor system response times and log any errors or exceptions	-	-		
2.	Security Testing	Verify user authentication and authorization	1. Attempt to log in to the system with invalid or unauthorized credentials	System should deny access and display an error message if the user is not authorized or if the credentials are invalid	Denied access for invalid credentials	Pass	Success
			2. Attempt to access restricted areas or perform unauthorized actions in the system	System should deny access and display an error message if the user attempts to perform unauthorized actions	Restricted access for invalid credentials	Pass	Success

Table 11.2 Non-Functional Test Cases

CHAPTER 12
ARCHITECTURE / DESIGN / FRAMEWORK / IMPLE-MENTATION

12.1 Architectural Design:

The Road Rescue can be designed using a layered architecture that consists of three main layers:

- Presentation Layer: This layer is responsible for handling user interactions and displaying the user interface to the end-user. It includes the user interface, such as the invoice and bill generation forms, which are developed using a client-side scripting language like JavaScript, HTML, and CSS.

- Application Layer: This layer manages the application logic and business rules. It includes the modules responsible for processing the data, such as product and customer information, generating invoices, and calculating GST. This layer can be implemented using a server-side programming language such as Java, Python, or PHP.

- Data Layer: This layer is responsible for storing the data used by the application. It includes the database management system that stores the product and customer information, as well as the invoices generated by the system. This layer can be implemented using a relational database management system such as MySQL, Oracle, or PostgreSQL.

12.2 Architectural Design:

- The Road Rescue can be developed using a web application framework, such as Django or Ruby on Rails.

- The chosen framework will provide pre-built modules and libraries that can be used to simplify the development process.

- The framework will also provide security features, such as input validation, to prevent errors and data loss.

12.3 Implementation:

The implementation process for the Road Rescue involves the following steps:

- Design the user interface and application logic using wireframes and flowcharts.
- Choose a web application framework and set up the development environment.
- Implement the presentation layer using HTML, CSS, and JavaScript.
- Develop the application layer using the chosen server-side programming language and web application framework.
- Implement the data layer using a database management system.
- Integrate the different layers of the system, test the application, and debug errors. Deploy the application on a server or a cloud platform, such as Amazon Web Services (AWS) or Microsoft Azure.

It is important to note that the implementation process will vary depending on the chosen framework, programming language, and database management system. Additionally, it is essential to follow software engineering best practices, such as version control and testing, to ensure the quality of the system and the reliability of the results.

CONCLUSION:

In conclusion, this project involved the development of a website for booking services. The user interface of the website included a homepage for booking services, a blog page for news and information, a contact page for reaching out to the company, and a login page for registered users. The functional requirements for the website were tested manually using a Word template, while the non-functional requirements were tested using automated tools like Selenium and JMeter. The testing process included both positive and negative test cases and was aimed at ensuring the website met all the specified requirements.

The stakeholders involved in the project were the project manager, the development team, the testing team, and the end-users. The project was developed using the waterfall methodology, and the system architecture included components like the web server, application server, database server, and client browsers. The testing framework used for the project included an introduction, scope of testing, types of testing, methodology, and test deliverables. The tools required for testing included Selenium, JMeter, and a Word template for manual testing.

The testing process was divided into functional and non-functional testing categories, with the functional testing being tested using manual testing methodology and the non-functional testing being tested using automated testing tools. The test plan and test case report were generated for both categories, and the status of each test case was tracked in a tabular format. The testing process helped to identify and rectify defects in the website and ensured that the website was ready for deployment.

Overall, the project was a success and met all the requirements set out for it. The website was fully functional and user-friendly, and the testing process helped to ensure its reliability and quality. The project can serve as a useful reference for similar projects in the future, and the lessons learned can be applied to improve the development and testing processes in future projects.

REFERENCES:

1. https://www.w3schools.com/
2. https://getbootstrap.com/
3. https://jquery.com/
4. https://nodejs.org/
5. https://expressjs.com/
6. https://www.mongodb.com/
7. https://github.com/
8. https://www.figma.com/
9. https://www.adobe.com/products/photoshop.html
10. https://slack.com/

APPENDIX

SAMPLE CODE

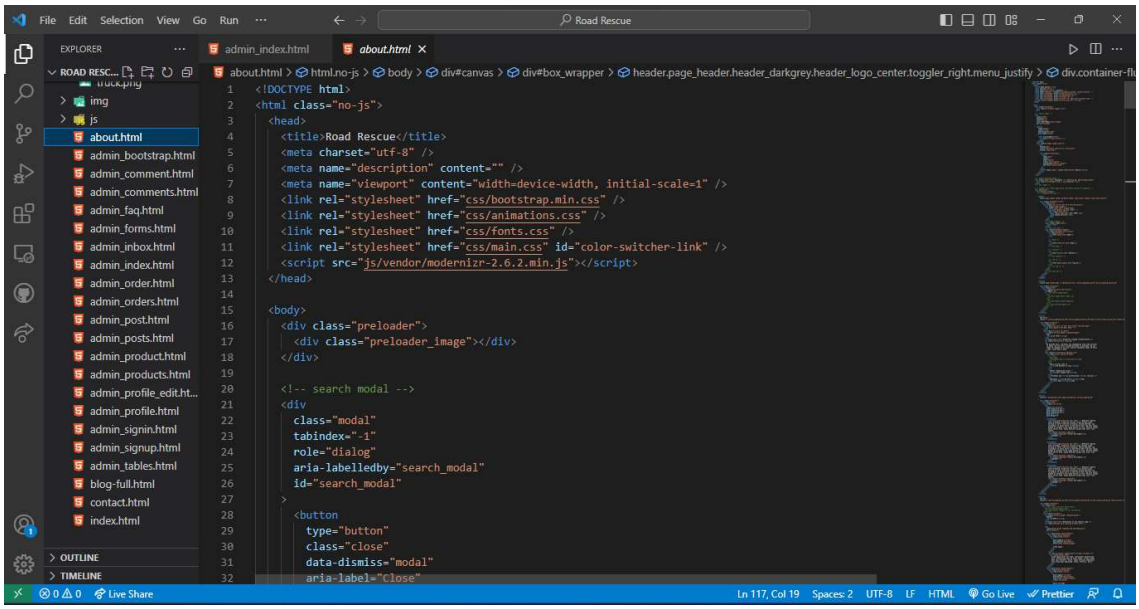
HOME PAGE:

```html
<!DOCTYPE html>
<html class="no-js">
  <head>
    <title>Road Rescue</title>
    <meta charset="utf-8" />
    <meta name="description" content="" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="css/bootstrap.min.css" />
    <link rel="stylesheet" href="css/animations.css" />
    <link rel="stylesheet" href="css/fonts.css" />
    <link rel="stylesheet" href="css/main.css" id="color-switcher-link" />
    <script src="js/vendor/modernizr-2.6.2.min.js"></script>
  </head>

  <body>
    <div class="preloader">
      <div class="preloader_image"></div>
    </div>

    <!-- search modal -->
    <div
      class="modal"
      tabindex="-1"
      role="dialog"
      aria-labelledby="search_modal"
      id="search_modal"
    >
      <button
        type="button"
        class="close"
        data-dismiss="modal"
        aria-label="Close"
```

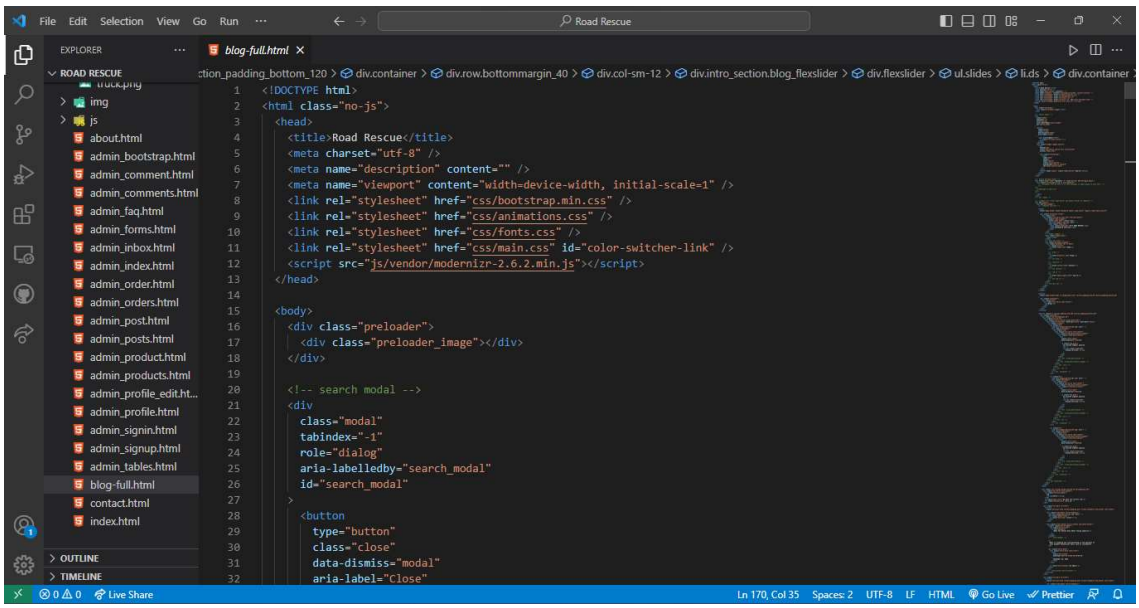
CONTACT PAGE:

```html
<!DOCTYPE html>
<html class="no-js">
  <head>
    <title>Road Rescue</title>
    <meta charset="utf-8" />
    <meta name="description" content="" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="css/bootstrap.min.css" />
    <link rel="stylesheet" href="css/animations.css" />
    <link rel="stylesheet" href="css/fonts.css" />
    <link rel="stylesheet" href="css/main.css" id="color-switcher-link" />
    <script src="js/vendor/modernizr-2.6.2.min.js"></script>
  </head>

  <body>
    <div class="preloader">
      <div class="preloader_image"></div>
    </div>

    <!-- search modal -->
    <div
      class="modal"
      tabindex="-1"
      role="dialog"
      aria-labelledby="search_modal"
      id="search_modal"
    >
      <button
        type="button"
        class="close"
        data-dismiss="modal"
        aria-label="Close"
```

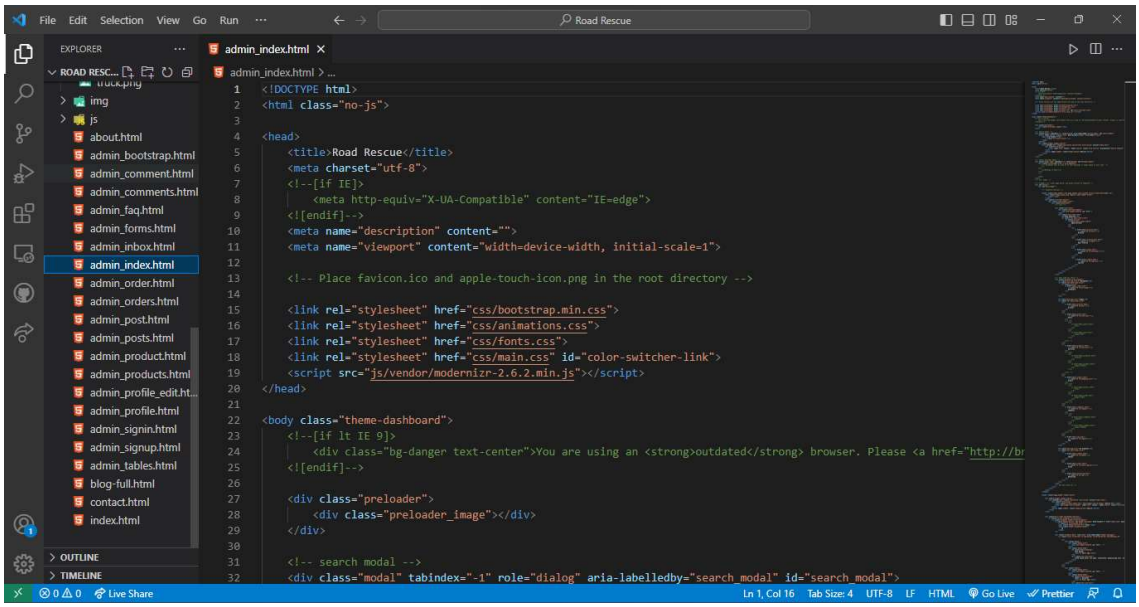
ABOUT PAGE:



BLOG PAGE:



ADMIN PAGE:

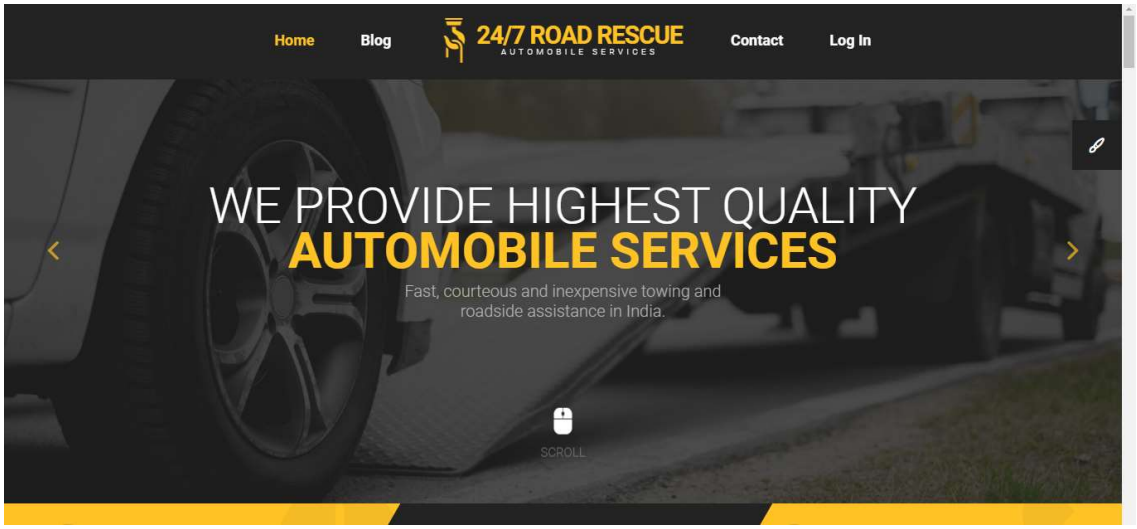

File Edit Selection View Go Run
Road Rescue
EXPLORER
admin_index.html
ROAD RESC...
truck.png
img
js
about.html
admin_bootstrap.html
admin_comment.html
admin_comments.html
admin_faq.html
admin_forms.html
admin_inbox.html
admin_index.html
admin_order.html
admin_orders.html
admin_post.html
admin_posts.html
admin_product.html
admin_products.html
admin_profile_edit.ht...
admin_profile.html
admin_signin.html
admin_signup.html
admin_tables.html
blog-full.html
contact.html
index.html
OUTLINE
TIMELINE
admin_index.html > ...
<!DOCTYPE html>
<html class="no-js">
<head>
<title>Road Rescue</title>
<meta charset="utf-8">
<!--[if IE]>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<![endif]-->
<meta name="description" content="">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Place favicon.ico and apple-touch-icon.png in the root directory -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/animations.css">
<link rel="stylesheet" href="css/fonts.css">
<link rel="stylesheet" href="css/main.css" id="color-switcher-link">
<script src="js/vendor/modernizr-2.6.2.min.js"></script>
</head>
<body class="theme-dashboard">
<!--[if lt IE 9]>
You are using an outdated browser. Please <a href="http://br
<![endif]-->

<!-- search modal -->

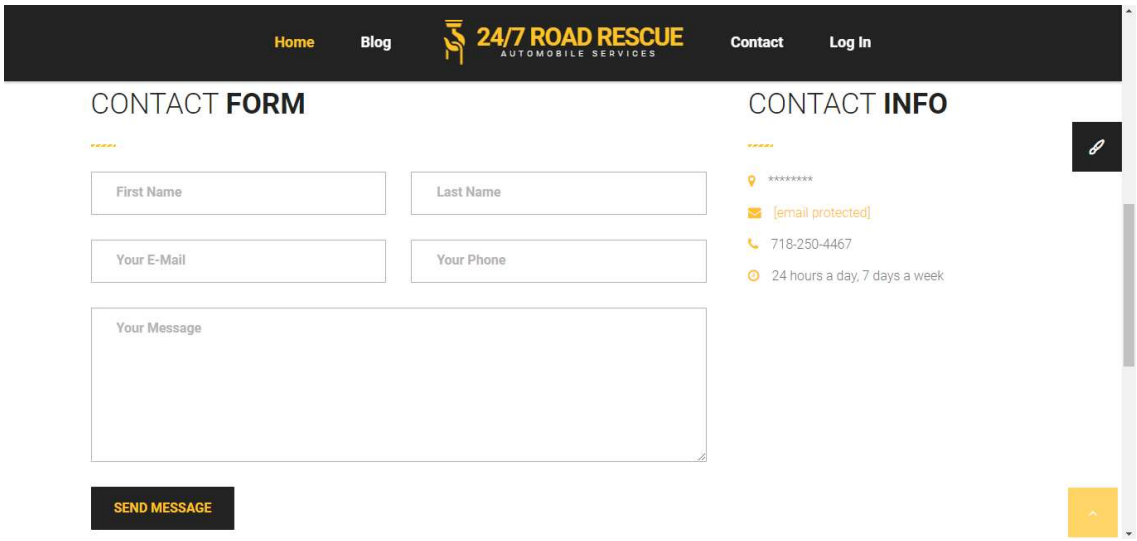
Live Share
Ln 1, Col 16
Tab Size: 4
UTF-8
LF
HTML
Go Live
Prettier

SAMPLE SCREENSHOTS

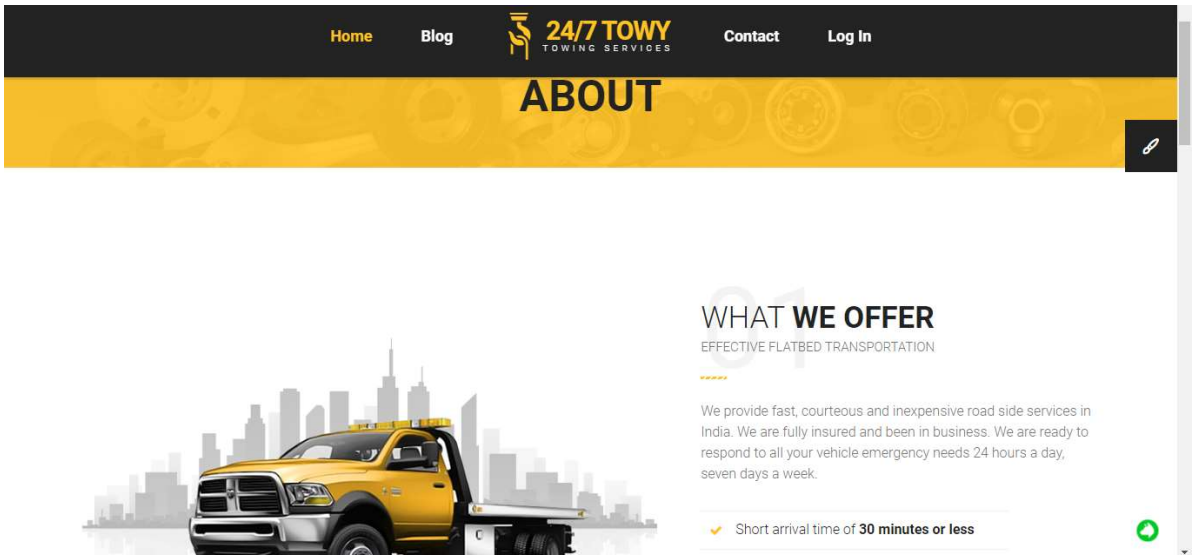
FRONT PAGE:



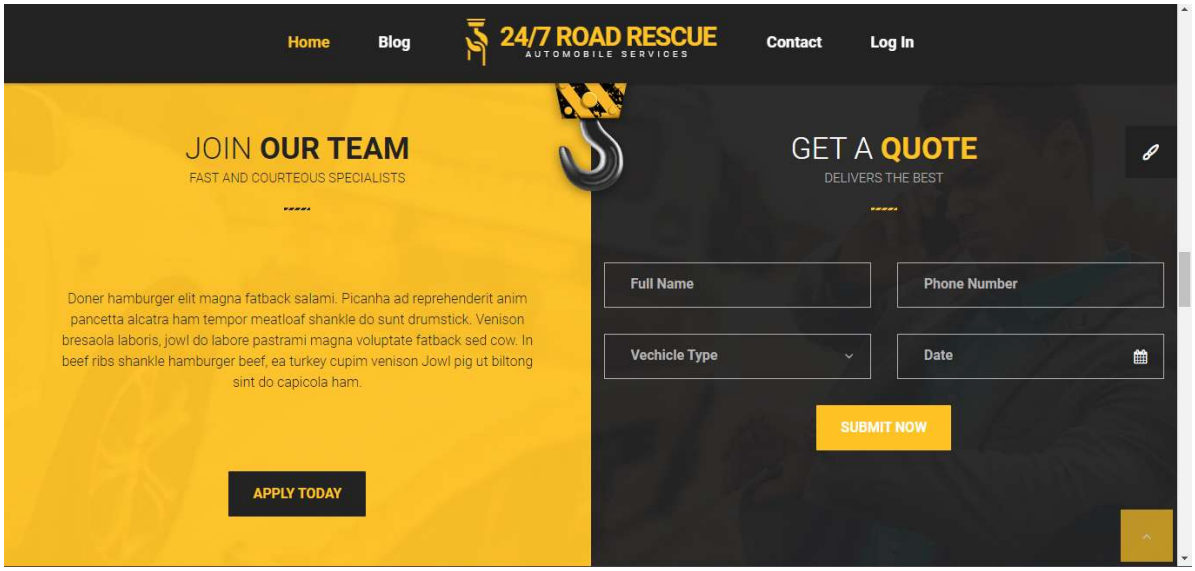
CONTACT PAGE:



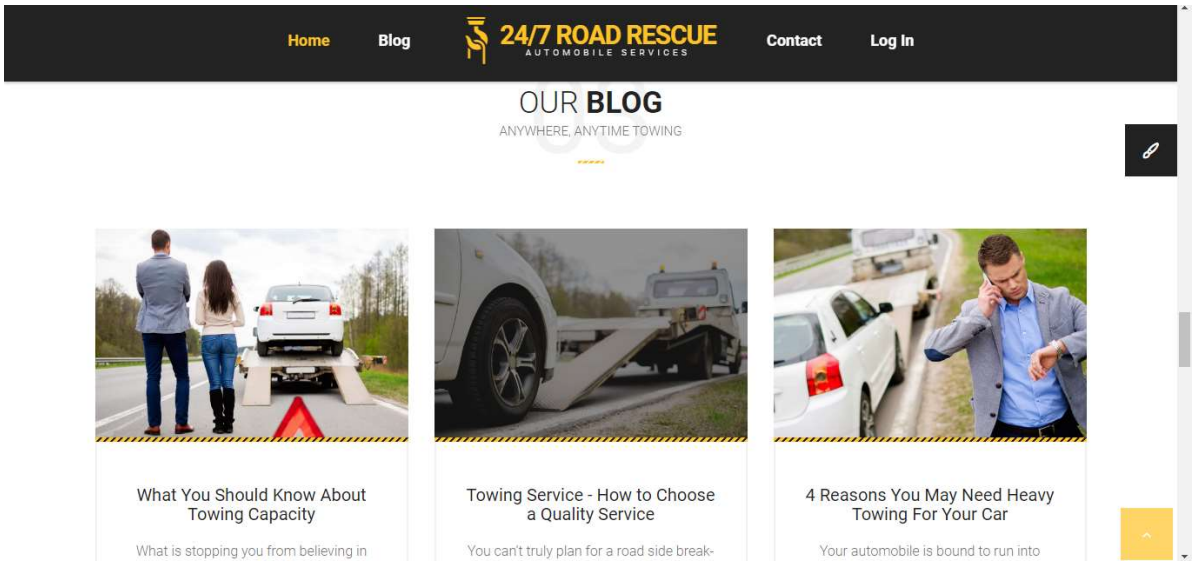
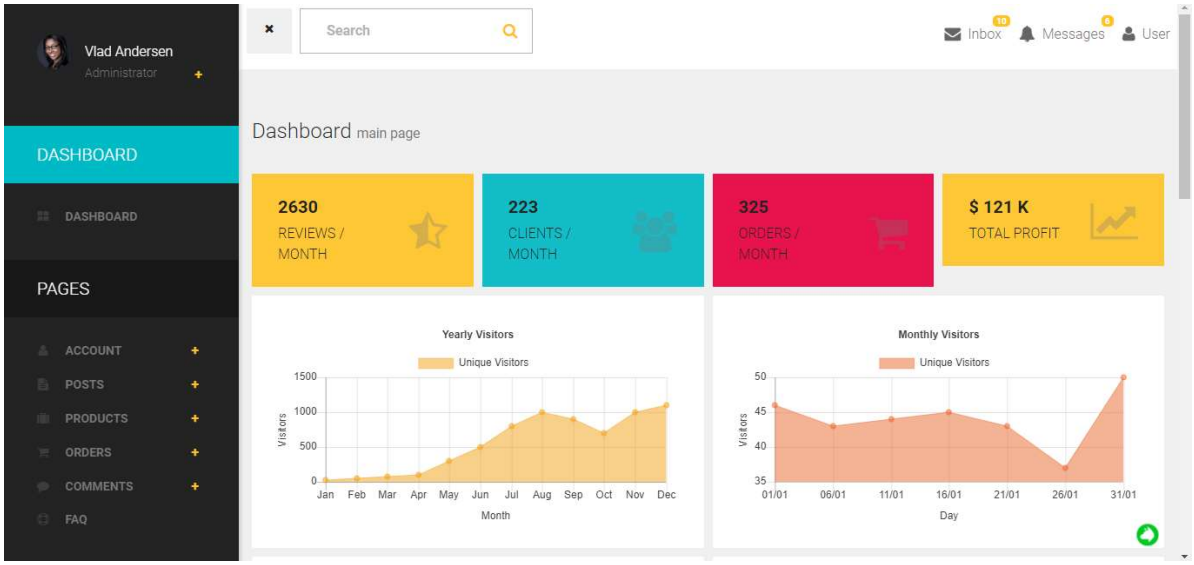
ABOUT:



BOOKING:



BLOG:



ADMIN PAGE: