1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Smart Automotive Mechanic Finder is a web-based application that enables car owners to quickly locate nearby auto repair shops and mechanics using the Google Maps API. The application aims to provide an easy and efficient solution to help car owners find reputable and reliable auto mechanics to perform maintenance and repair services for their vehicles. In addition, the application integrates with Clickatell SMS Gateway to send SMS notifications to the mechanics when a customer requests their services.

The application features a user-friendly interface that allows car owners to search for mechanics based on their location, services offered, and ratings from previous customers. The system also allows mechanics to register and create a profile to advertise their services, view and accept service requests, and update their availability.

1.2 OBJECTIVE OF THE SYSTEM

The objective of the Smart Automotive Mechanic Finder using Google map navigator and Clickatell project is to develop a web-based application that allows users to easily locate and contact nearby automotive mechanics through the use of Google Maps and Clickatell API. The project aims to provide a convenient and efficient solution for vehicle owners to find reliable and affordable mechanics in their local area, while also helping mechanics to increase their visibility and customer base. The system will utilize a database of mechanics' information and services to provide users with the most relevant search results based on their location and specific requirements. The application will also incorporate a messaging system using Clickatell API to allow users to directly communicate with mechanics and schedule appointments.

1.3 EXISTING SYSTEM

The existing system for finding automotive mechanics typically involves manually searching for service centres in a particular area or relying on recommendations from friends and family. This process can be time-consuming and may not always yield the best results. Additionally, there is often a lack of transparency in the pricing and quality of services offered by these mechanics.

The Smart Automotive Mechanic Finder using Google Map Navigator and Clickatell project aims to address these issues by providing an automated system that allows users to search for mechanics based on their location and view information such as service pricing, reviews, and ratings from other users. This system leverages the power of Google Maps to help users find nearby mechanics and Clickatell for communication purposes, making the process of finding and contacting an automotive mechanic more efficient and convenient.

1.4 PROPOSED SYSTEM

The proposed system for Smart Automotive Mechanic Finder using Google map navigator and Clickatell project is an improved version of the existing system. It aims to provide a more efficient and user-friendly way for drivers to find automotive mechanics in their area. The system will make use of Google Maps API to display the location of nearby mechanics and allow users to easily navigate to them. Clickatell will be integrated to enable users to send a text message to the selected mechanic to book an appointment.

The proposed system will have a user-friendly interface where users can search for mechanics by location, service type, and ratings. The system will also have a review and rating system to help users make informed decisions on which mechanic to choose. The system will provide notifications to users when their appointment is confirmed or if there are any changes to the appointment.

Additionally, the proposed system will have an administrative dashboard for the mechanics where they can manage their appointments, view customer feedback, and update their profile. This will allow mechanics to efficiently manage their workload and improve customer satisfaction.

1.4.1 Advantage of Proposed System

- Efficient Mechanic Finder: The system provides an efficient way to find mechanics for automotive repairs. With the help of Google maps, the user can easily locate the nearest mechanic, and the Clickatell integration enables seamless communication with the mechanic.
- Time-Saving: The system saves time for both the user and the mechanic. The user can quickly locate a nearby mechanic and make an appointment, and the mechanic can view the appointment and be prepared for the repair job.
- Cost-effective: The system helps the user find the most cost-effective mechanic for their needs. The user can view the mechanic's ratings and reviews and choose the one that fits their budget.
- User-Friendly: The system is user-friendly and easy to use. The user interface is intuitive and requires minimal training.
- Automated Reminders: The system can send automated reminders to the user and mechanic about the scheduled appointment. This feature ensures that the user and the mechanic are prepared for the repair job, and the appointment is not missed.
- Improved Communication: The system provides improved communication between the user and the mechanic. The Clickatell integration allows the user to communicate with the mechanic via text messages, which is a convenient and quick way to ask questions and get updates on the repair job.

2.SYSTEM REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10, Linux or macOS
- **Database Management System:** MySQL
- **Web Server:** Apache or Nginx
- **Programming Languages and Frameworks:**
  PHP 7.0 or higher, HTML, CSS, JavaScript and Laravel Framework 6.0 or higher
- Google Maps API key and Clickatell API key
- Text editors or integrated development environment (IDE) such as Visual Studio Code or PhpStorm

2.2 HARDWARE REQUIREMENTS

- **Processor:** Intel Core i3 or equivalent
- **RAM:** 4 GB or more
- **Hard Disk Space:** 20 GB or more
- **Internet Connection:** Broadband or high-speed internet connection

2.3 LANGUAGE SPECIFICATION

2.3.1 INTRODUCTION TO PHP

PHP started out as a small open-source project that evolved as more and more people found out how useful it was. Rasmus Leadoff unleashed the first version of PHP way back in 1994.PHP is a recursive acronym for "PHP: Hypertext Pre-processor". PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with several popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and

Microsoft SQL Server. PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time. PHP supports many major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time. PHP is forgiving PHP language tries to be as forgiving as possible. PHP Syntax is C-Like.

2.3.2 COMMON USES OF PHP

- PHP Performs System Functions, I.E., From Files on A System It Can Create, Open, Read, Write, And Close Them.
- PHP Can Handle Forms, I.E., Gather Data from Files, Save Data to A File, Through Email You Can Send Data, Return Data to The User.
- You Add, Delete, Modify Elements Within Your Database Through PHP.
- Access Cookies Variables and Set Cookies.
- Using PHP, You Can Restrict Users to Access Some Pages of Your Website.

2.3.3 CHARACTERISTICS OF PHP

Five important characteristics make PHP's practical nature possible

- Simplicity
- Security
- Flexibility
- Familiarity

2.4 XAMPP

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was

developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

2.4.1 COMPONENTS OF XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/ programming languages such as **PHP** and **Perl**. X denotes Cross-platform, which means that it can work on different platforms such as Windows, Linux and macOS.

Many other components are also part of this collection of software and are explained below.

Cross-Platform: Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.

Apache: It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

MariaDB: Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS,

developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.

PHP: It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for Hypertext Processor. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.

Perl: It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.

phpMyAdmin: It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.

OpenSSL: It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol. Presently version 0.9.8 is a part of XAMPP.

XAMPP Control Panel: It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.2.1 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial.

Webalizer: It is a Web Analytics software solution used for User logs and provide details about the usage.

Tomcat: Version 7.0.42 is currently being used in XAMPP. It is a servlet based on JAVA to provide JAVA functionalities.

Filezilla: It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files. Its recently updated version is 0.9.41.

2.5 JAVASCRIPT

JavaScript is a compact, object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. And Livewire enables you to create server-based applications similar to common gateway interface (CGI) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form input, and page navigation.

For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

2.6 CLICKATEL

Clickatell is a global mobile communications company that provides messaging and voice services through various channels, including SMS, WhatsApp, Viber, and more. It allows businesses to connect with their customers and employees through mobile messaging, with features like two-way messaging, delivery reports, and personalized messaging. Clickatell also provides APIs and integration options for developers to build messaging functionality into their own applications.

2.7 GOOGLE MAP

Google Maps is a web-based mapping service developed by Google. It provides a comprehensive mapping solution, allowing users to view maps, get directions, and explore businesses and other points of interest. Google Maps uses satellite imagery, street maps, 360-degree panoramic views of streets, real-time traffic conditions, and route planning to help users navigate their way around the world. It also offers features such as the ability to search for nearby businesses and attractions, save locations for future reference, and share maps and directions with others. Google Maps is available on desktop and mobile devices, and is widely used for both personal and business purposes.

2.8 CSS: CASCADING STYLE SHEETS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification

2.9 MySQL : STRUCTURED QUERY LANGUAGE

MySQL is an open-source relational database management system. As with other relational databases, MySQL stores data in tables made up of rows and

columns. Users can define, manipulate, control, and query data using Structured Query Language, more commonly known as SQL. A flexible and powerful program, MySQL is the most popular open-source database system in the world. As part of the widely used LAMP technology stack (which consists of a Linux-based operating system, the Apache web server, a MySQL database, and PHP for processing), it's used to store and retrieve data in a wide variety of popular applications, websites, and services. MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL is free and open-source software under the terms of the GNU General Public License and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Wideners forked the open-source MySQL project to create MariaDB.

2.10 HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext one point to another point. We can navigate through the information based on out interest and

preference. A markup language is simply a series of items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop

HTML provides tags (special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics, fonts, different sizes, colour, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

2.10.1 ADVANTAGE

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent
- HTML tags are not case-sensitive.

2.11 PHP AND MYSQL DEVELOPMENT

PHP is a fast and feature-rich open-source scripting language used to develop Web Applications or Internet / Intranet Applications. MySQL is a powerful open-source database server built based on a relational database management system (RDBMS) and can handle a large concurrent database connection. When combined, talented PHP and MySQL developers can build very powerful and scalable Web / Internet / Intranet Applications and MySQL are referred to as development tools. PHP and MySQL are Open Source, meaning that they are free development tools, and there is a large community of dedicated volunteer programmers who contribute to make improvements and are continuously adding features to development tools and database servers.

3.SYSTEM DESIGN

3.1 INTRODUCTION

System analysis is the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose.

Computer software design changes continuously as new methods; better analysis and broader understanding evolved. Software Design is at relatively early stage in its revolution.

Therefore, Software Design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However, techniques for software designs do exist, criteria for design qualities are available and design notation can be applied

3.2 SYSTEM ARCHITECTURE

System architecture is the structure and organization of a software system or application. It defines how different components of a system work together and how data flows between them. In the case of Smart Automotive Mechanic Finder using Google map navigator and Clickatell project, the system architecture includes three main components: the user interface, the database, and the backend logic. It defines how different components of a system work together and how data flows between them. quantitative nature that are normally associated with more classical engineering disciplines.

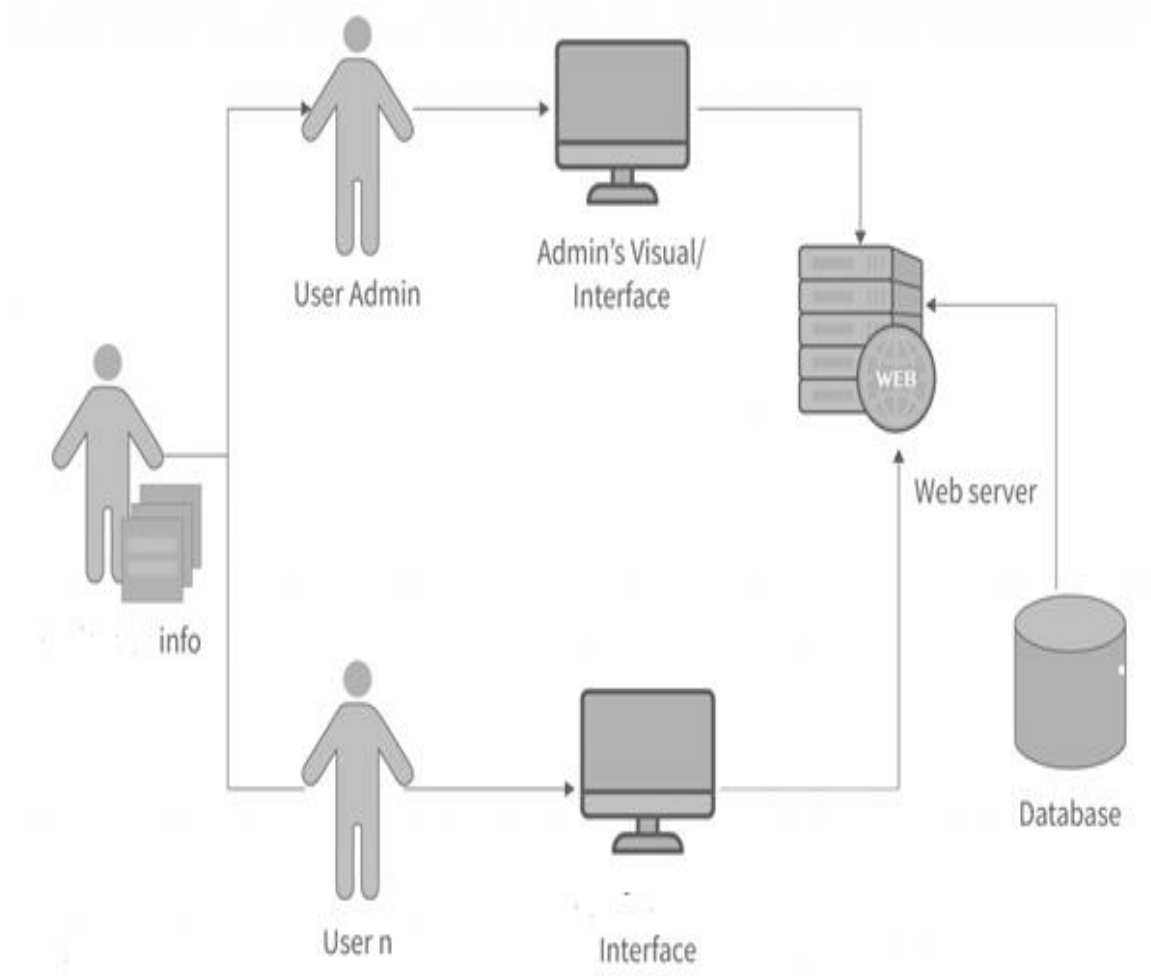


Fig 3.3 System Architecture Diagram

3.3 Module Description

Modules:

- USER MODULE
- MECHANIC MODULE
- DATABASE MODULE
- CLICKATEL OTP MODULE

- MOBILE APPLICATION MODULE
- WEB HOSTING MODULE
- GOOGLE MAP INTEGRATION MODULE

3.3.1 USER MODULE

The user module allows users to register, log in, and log out. Users benefit from being able to sign on because this associates content they create with their account and allows various permissions to be set for their roles. The user module supports user roles, which can be set up with fine-grained permissions allowing each role to do only what the administrator permits. Each user is assigned one or more roles. By default, there are three roles: anonymous a user who has not logged in and authenticated a user who is registered, and administrator a signed in user who will be assigned site administrator permissions.

3.3.2 MECHANIC MODULE

The mechanic module is a core part of the Smart Automotive Mechanic Finder system. It allows mechanics to register and create profiles, including their location and services offered. The module also enables customers to search for nearby mechanics based on their location, view mechanic profiles, and make service requests. The module is integrated with Google Maps to provide customers with accurate location information, and Clickatell for SMS notifications and reminders. The mechanic module helps connect customers with mechanics in a fast, convenient, and reliable way.

3.3.3 DATABASE MODULE

The database for the Development Project that identifies each Participating Party and identifies certain information about each Participating Party such as its Notice Address, Party Project Leader, Technical Coordinators and Designated Executive.

3.3.4 CLICKATEL OTP MODULE

One Time Passwords (OTPs) are a necessity in today's technological society. They help to protect users from fraud and protect their accounts and private information through SMS authentication. The system requires some refinement when used with mobile devices, but Google may just be the company to achieve that.

With the release of Google's Android O, the company looks to change the way that OTPs and SMS authentication are handled. The second developer preview, which released in May 2017, contains a slew of new application programming interface (API) code that facilitates OTPs**.**

3.3.5 MOBILE APPLICATION MODULE

The decisions to build a native application with Java and use a modular design are discussed. The application includes five modules: catalogue search, in-building navigation, a barcode scanning feature, and up to date notifications of circulating technology availability. A sixth module, Amazon recommendations, that is not included in the version of the app that was released is also discussed. The article also reports on the findings of two rounds of usability testing and the plans for future development of the app.

3.3.6 WEB HOSTING MODULE

The web hosting module for the Smart Automotive Mechanic Finder project involves the storage and management of the web. This module includes the deployment of the web application on the hosting server and ensuring that the web application is available to users 24/7.

3.3.7 GOOGLE MAP INTEGRATION MODULE

Google Maps is usually the basis of any map application for displaying data with a location reference. With custom markers, lines, colours, polygons, and images, you can create a layout that is tailored to your company.

Places provides comprehensive location data for over 100 million locations around the world. Users receive important guidance and decision-making assistance in the form of addresses.

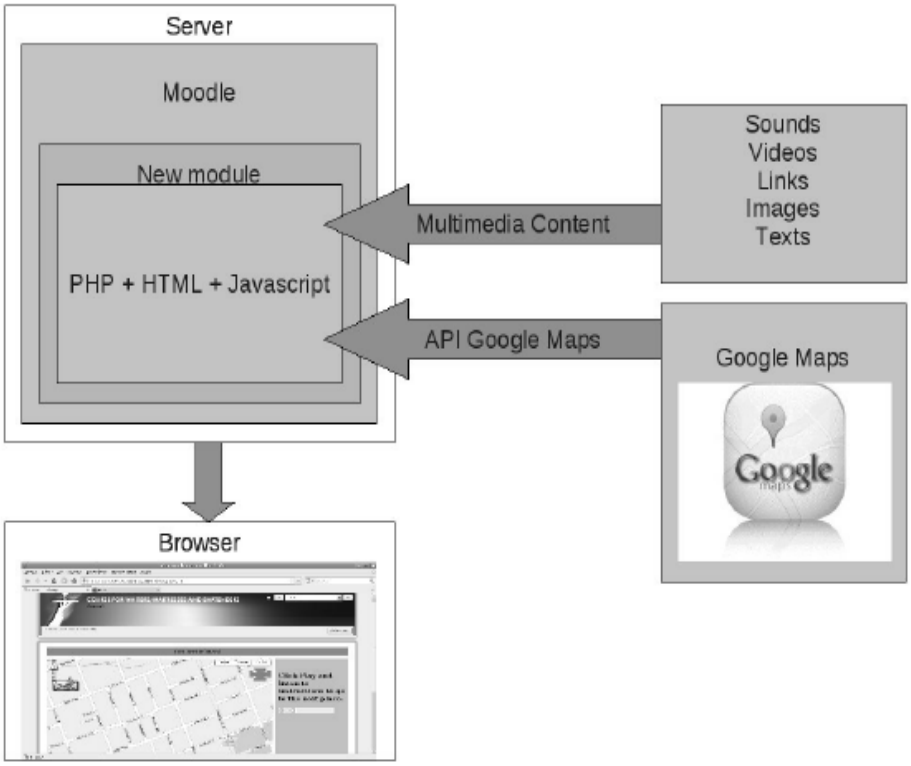


Fig 3.2.7 Google Map Integration module

The system also integrates the Clickatell API to provide an SMS messaging feature, allowing users to send a text message to the mechanic directly from the system, requesting a service or making an appointment. The mechanic can then respond to the message through the system, providing a more convenient and efficient communication channel for both parties.

The proposed system addresses the limitations of the existing system, such as the inconvenience of manually searching for nearby mechanics and the lack of a streamlined communication channel between the user and the mechanic. The Smart Automotive Mechanic Finder system offers a more user-friendly and efficient solution, which can save time and effort for vehicle owners, while also

benefiting the automotive mechanics by providing them with a broader customer base.

3.4 ER-DIAGRAM

An ER diagram, or entity-relationship diagram, is a visual representation of the relationships between entities in a database. ER diagrams are an essential tool for designing and understanding databases, as they provide a clear and concise view of the relationships between entities and their attributes. They are often used in the early stages of database design to help identify the entities that will be included in the database and how they relate to each other.

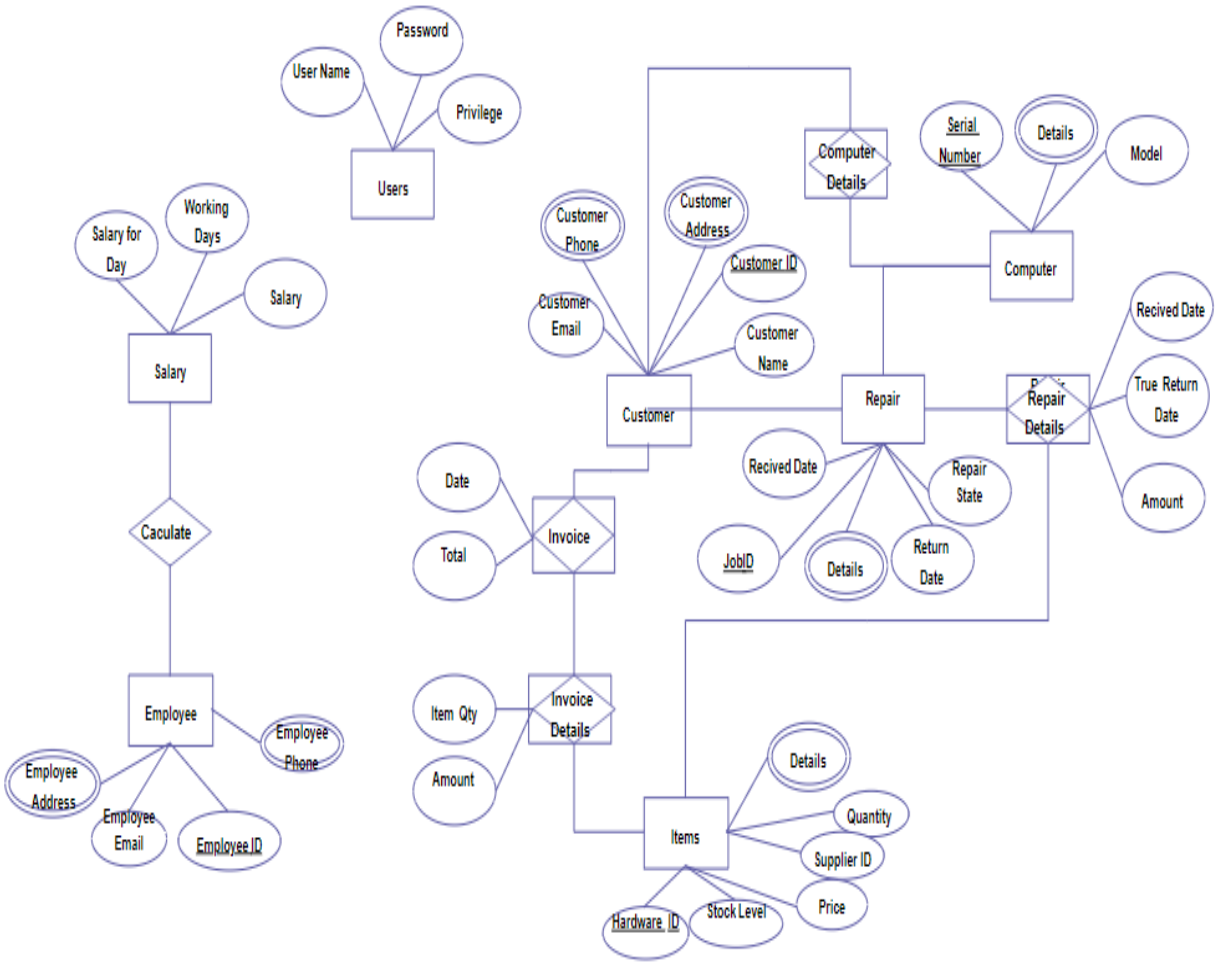


Fig 3.4 ER-DIAGRAM

4.SYSTEM TESTING AND IMPLEMENTATION

4.1. INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

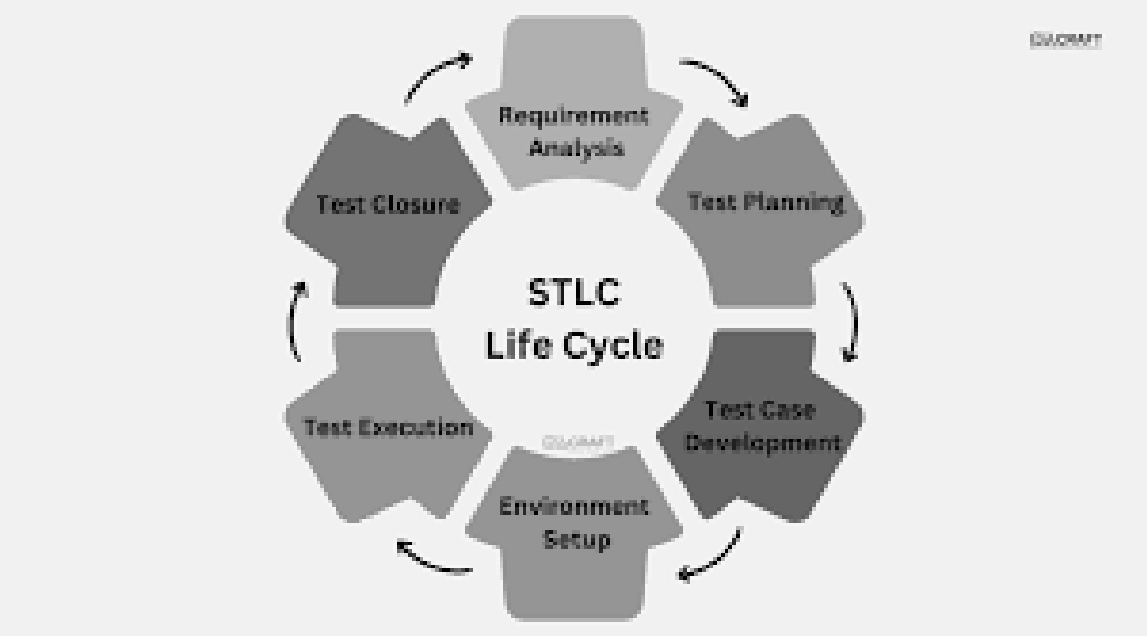


Fig 4.1 Software testing cycle

- Test Planning: This is the first phase in the software testing cycle where the testing team defines the scope, objectives, and approach for testing the software. This phase involves identifying the testing requirements, creating test cases, and designing test scenarios.
- Test Design: In this phase, the testing team creates detailed test cases based on the requirements specified in the planning phase. The test cases are designed to cover all possible scenarios and user actions to ensure that the software is thoroughly tested.

- Test Execution: This is the phase where the actual testing of the software is performed. The testing team executes the test cases designed in the previous phase and identifies any defects or issues that need to be fixed.
- Defect Reporting: Any defects or issues identified during the test execution phase are reported to the development team for resolution. The testing team also verifies that the fixes provided by the development team have resolved the issues.
- Test Closure: This is the final phase of the software testing cycle where the testing team provides a report on the testing process, including the test results, test coverage, and defects identified. The team also provides recommendations for improving the software and the testing process in future releases.

4.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

4.2.1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form. We have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

4.2.2. BASIC PATH TESTING

The established technique of flow graph with Cyclamate complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graphs.

Determine the Cyclamate complexity of the resultant flow graph, using formula:

- V (G) =E-N+2 or
- V (G) =P+1 or
- V (G) =Number of Regions
- Where V (G) is Cyclomatic complexity,
- E is the number of edges,
- N is the number of flow graph nodes,
- P is the number of predicate nodes.
- Determine the basis of set of linearly independent paths.

4.2.3 DATA FLOW TESTING

This type of testing selects the path of the program, according to the location of the definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements. This kind of testing was used in this Project.

4.2.4 LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

No	Test Scenario	Expected Result	Test Result
1	Username is correct. Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
2	Username is incorrect. Password is correct.	Username and Password is incorrect.	Username and Password is incorrect.
3	Username is empty. Password is correct.	Username is required.	Username is required.
4	Username is correct. Password is empty.	Password is required.	Password is required
5	Both Username and Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
6	Both Username and Password is empty.	Username and Password is required.	Username and Password is required.
7	Both Username and Password is correct.	Login Successful.	Login Successful.

Table 4.2.4 Loop Testing

4.3 DATA FLOW

- A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however by two separate arrows since these happen at different type.
- A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.

- A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data in the beginning process.
- A Data flow to a data store means update (delete or change).
- A data Flow from a data store means retrieve or use.
- A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

4.3.1 DATA FLOW DIAGRAM

A data flow diagram is a graphical representation of the flow of data through a system. It shows how information enters and exits the system, as well as the processes and storage locations that handle the data. DFDs are used to visualize and analyze information systems, and can help identify areas for improvement and optimization.

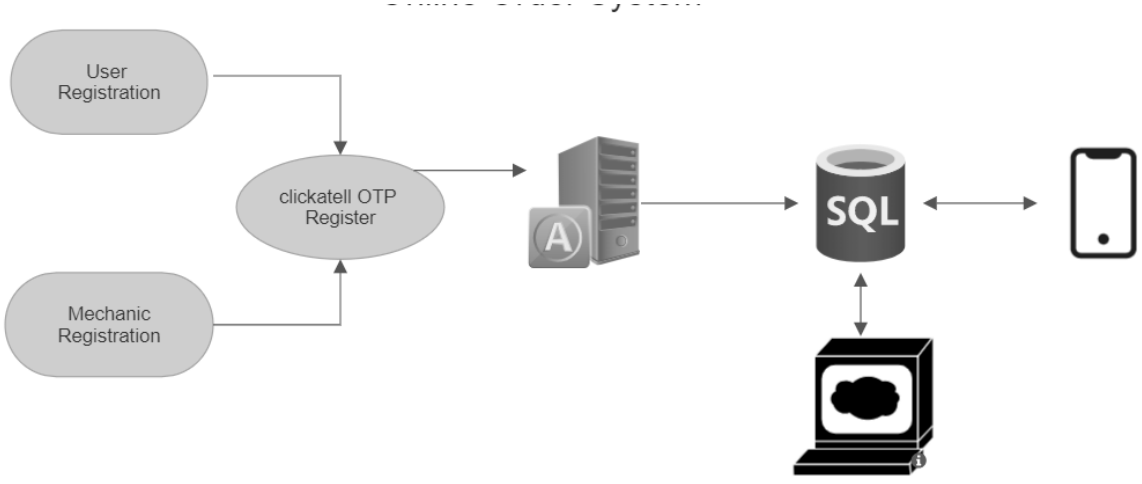


Fig 4.3.1 Data Flow Diagrams

4.3.2 USE CASE DIAGRAM

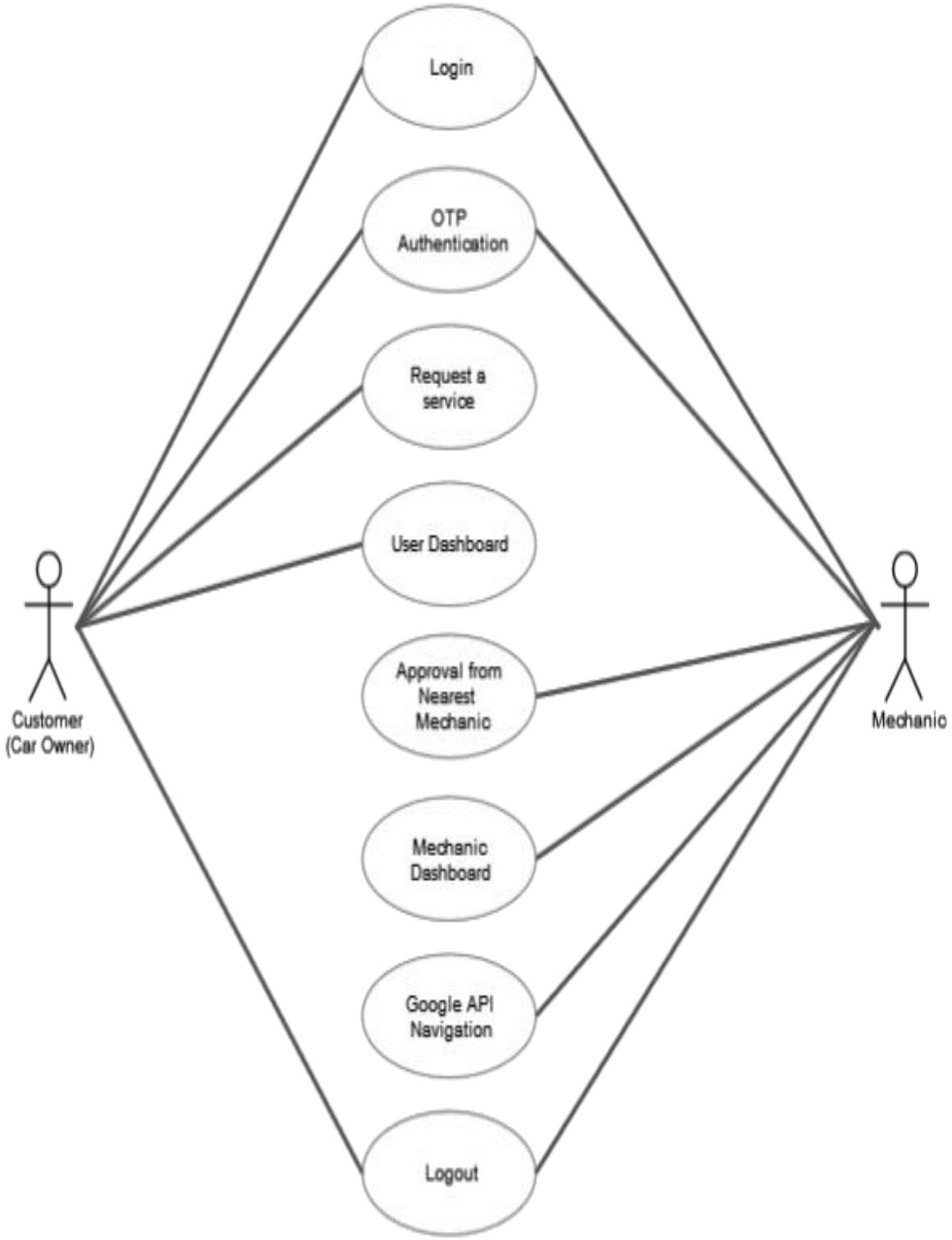


Fig 4.3.2 Use Case Diagram

4.3.3 SEQUENCE DIAGRAM

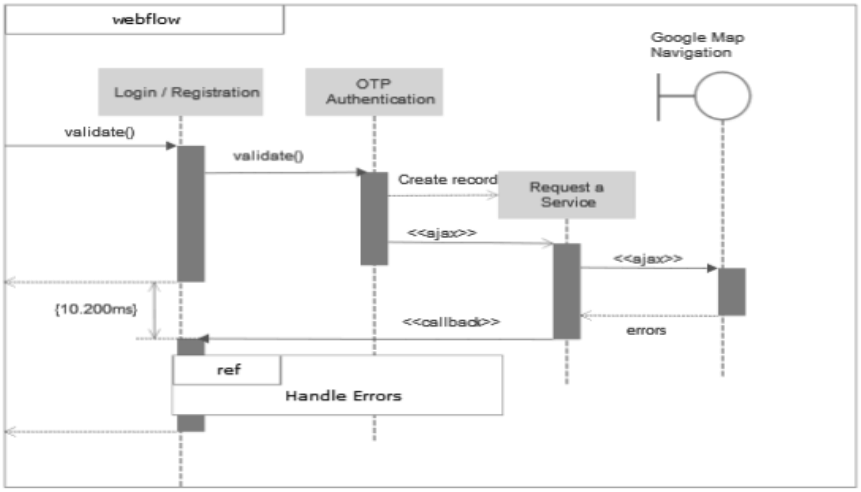


Fig 4.3.3 Sequence Diagram

4.3.4 CLASS DIAGRAM

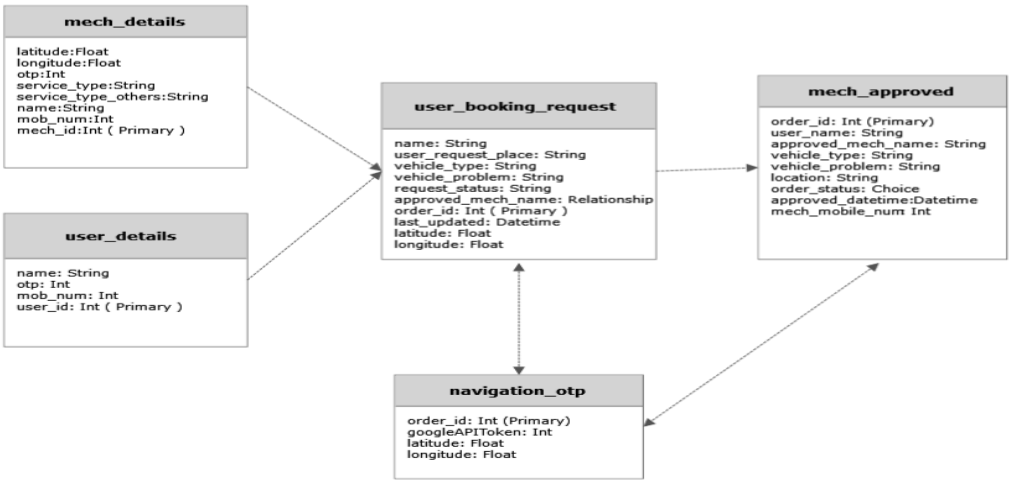


Fig 4.3.4 Class Diagram

4.3.5 SUPPORT SYSTEM

Fig 4.3.5 Support System

4.4 CODING

4.4.1 connection.php

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "repairspot";

$conn = new mysqli($servername,$username,$password,$dbname);

if($conn->connect_error)

die ('connection faild:'.$conn->connect_error);

}

?>
```

4.4.2 index.php

```
<?php

include('components/header.php');

?>

<div class="link d-flex justify-content-center">

<div>

<div class="d-flex">

<a class="noHover" href="user_register.php">

<button class="btn">REGISTER AS VEHICLE OWNER</button><br>

</a>

<a class="noHover" href="mech_register.php">
```

<button class="btn">REGISTER AS MECHANIC</button>

<button class="btn">VEHICLE OWNER LOGIN</button>

<button class="btn">MECHANIC LOGIN</button>

<?php

include('components/footer.php');

?>

4.4.3 mech_login.php

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Owner Signin</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp" crossorigin="anonymous">

<link rel="stylesheet" href="./assets/css/styles.css">

<link rel="shortcut icon" href="./assets/images/car-care.png" type="image/x-icon"/>

session_start();

}

header("Location: https://localhost/Quick-Mechanist/mech_dashboard.php");

}else{

if(empty(session_id()) && !headers_sent()){

session_start();

}

Session_destroy();

header('Location: ' . $_SERVER['HTTP_REFERER']);

}

$conn->close();

}

?>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.4/jquery.js" integrity="sha512-6DC1eE3AWg1bgitkoaRM1lhY98PxbMIbhgYCGV107aZlyzzvaWCW1nJW2vDuYQm06hXrW0As6OGKcIaAVWnHJw==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

</head>

```
<body>

<div>

<?php

include('components/navbar.php');

?>

<p class="phError" style="color:red; margin-left: 20px; font-size: 12px; margin-top: 5px;"></p>

</div>

<div class="form-group">

<input type="password" name="otp" id="pass" placeholder="OTP" />

<p class="otpError" style="color:red; margin-left: 20px; font-size: 12px; margin-top: 5px;"></p>

</div>

<div class="form-group form-button">

<input class="form-submit" name="button" type="submit" value="Login" onclick="return verifyOTP()"  />

</div>

</form>

</div>

<div class="signup-image" style="margin-left: 15px">

<figure>

<img src="./assets/images/undraw_bike_ride.svg" alt="sing up image">

</figure>
```

```
<a href="https://localhost/Quick-Mechanist/user_login.php" class="signup-image-link">Owner Login</a>

</div>

var xhr = new XMLHttpRequest(),

body = JSON.stringify({

"messages": [{

"channel": "whatsapp",

xhr.setRequestHeader('Content-Type', 'application/json');

xhr.setRequestHeader('Authorization', 'aGvBybhRR0eNevM7QqSU1g==');

xhr.onreadystatechange = function() {

if (xhr.readyState == 4 && xhr.status == 200) {

console.log('success');

alert("OTP Sent Successfully!!")

}

$(".otpError").html("").hide();

var enteredOtp = $("#pass").val();

if (enteredOtp == otp) {

document.getElementById("verifiedOTP").value = otp;

document.getElementById("mech_login").submit();

} else {

$(".otpError").html('Invalid OTP!')

$(".otpError").show();

return false;
```

```
}


};
xhr.send(body);
} else {
$(".phError").html('Please enter a valid number!')
$(".phError").show();
}
}
function generateOTP() {
var digits = '0123456789';
let OTP = '';
for (let i = 0; i < 4; i++) {
OTP += digits[Math.floor(Math.random() * 10)];
}
return OTP;
}
function verifyOTP() {
$(".otpError").html("").hide();
var enteredOtp = $("#pass").val();
if (enteredOtp == otp) {
document.getElementById("verifiedOTP").value = otp;
document.getElementById("mech_login").submit();
```

```
} else {

$(".otpError").html('Invalid OTP!')

$(".otpError").show();

return false;

}

}

</script>

</html>
```

4.4.4 user_login.php

```
xhr.open('POST', 'https://platform.clickatell.com/v1/message', true);

xhr.setRequestHeader('Content-Type', 'application/json');

xhr.setRequestHeader('Authorization', 'aGvBybhRR0eNevM7QqSU1g==');

xhr.onreadystatechange = function() {

if

let OTP = '';

for (let i = 0; i < 4; i++) {

OTP += digits[Math.floor(Math.random() * 10)];

}

return OTP;

}

function verifyOTP() {

$(".otpError").html("").hide();

var enteredOtp = $("#pass").val();
```

```
if (enteredOtp == otp) {

document.getElementById("verifiedOTP").value = otp;

document.getElementById("mech_login").submit();

} else {

$(".otpError").html('Invalid OTP!')

$(".otpError").show();

return false;

}

}

</script>

</html>
```

4.4.5 mech_register.php

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title></title>

<title> MECHANIC REGISTER</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
crossorigin="anonymous">

<link rel="stylesheet" href="./assets/css/styles.css">
```

<link rel="stylesheet" href="./assets/fonts/material-icon/css/material-design-iconic-font.min.css">

<link rel="stylesheet" href="./assets/css/style.css">

<link rel="shortcut icon" href="./assets/images/car-care.png" type="image/x-icon" />

<style type="text/css">

* {

padding: 0;

margin: 0;

}

.form-select {

border: none;

">

//mechanic service=others...to open textbox

function checkvalue() {

var service = document.getElementById('mech_service_type').value;

if (service == 'stype') {

}

//password checkbox

function myFunction() {

var x = document.getElementById("password");

if (x.type === "password") {

x.type = "text";

} else {

```
x.type = "password";

}

}

</script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.4/jquery.js" integrity="sha512-6DC1eE3AWg1bgitkoaRM1lhY98PxbMIbhgYCGV107aZlyzzvaWCW1nJW2vDuYQm06hXrW0As6OGKcIaAVWnHJw==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

<style>

.phError,

.otpError {

display: none;

}

</style>

</head>

<body>

<div>

<?php

include('components/navbar.php');

?>

<select class="form-select" onchange="checkvalue(this.value)" id="mech_service_type" required name="mech_service" required>

<option value="stype" hidden>Select Service Type</option>
```

<option value="bike_mechanic">

Bike Mechanic

</option>

<option value="car_mechanic">

Car Mechanic

</option>

<option value="others">

Others

</option>

</select>

<input type="text" name="mech_other_service" id="mech_other_service" placeholder="If Others:" />

<input maxlength="10" type="tel" name="mob_num" id="mob_num" placeholder="Phone Number" />

<button type="button" class="btn btn-secondary" style="font-size: 8px;" onclick="sendOTP()">

//mechanic service=others...to open textbox

function checkvalue() {

var service = document.getElementById('mech_service_type').value;

```
if (service == 'stype') {

}

Get OTP

<input type="text" hidden name="latitude" id="latitude">

<input type="text" hidden name="longitude" id="longitude">

<div class="form-group form-button">

<!-- <input class="form-submit" type="submit" id="btnsubmit" name="mech_form_submit" value="Register" onclick="return Validate()" /> -->

<input class="form-submit" type="button" value="Register" onclick="return verifyOTP()" />

</div>

<input class="form-submit" id="verifiedOTP" name="verifiedOTP" hidden="true" />

</form>

</div>

<div class="signup-image">

<!-- <input class="form-submit" type="submit" id="btnsubmit" name="mech_form_submit" value="Register" onclick="return Validate()" /> -->

<input class="form-submit" type="button" value="Register" onclick="return verifyOTP()" />

include('components/justFooter.php');

?>

</div>

</body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
```

qKXV1j0HvMUeCBQ+QVp7JcfGl760yU08IQ+GpUo5hlbpg51QRiuqHAJz8+BrxE/N"
crossorigin="anonymous"></script>

<script>

$(document).ready(function() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showLocation);

} else {

$('#location').html('Geolocation is not supported by this browser.');

}

});

function showLocation(position) {

var latitude = position.coords.latitude;

var longitude = position.coords.longitude;

document.getElementById("latitude").value = latitude;

document.getElementById("longitude").value = longitude;

}

</script>

<script>

"content": `Hello ${name}! - Here's OTP for Quick Mechanist to register as mechanic. Please don't share the OTP - ${otp}`

},

{

"channel": "sms",

```
// "to": "91"+number,

"to": "919944622435",

"content": `Hello ${name}! - Here's OTP for Quick Mechanist to register as mechanic. Please don't share the OTP - ${otp}`

}

function showLocation(position)

var latitude = position.coords.latitude;

var longitude = position.coords.longitude;

document.getElementById("latitude").value = latitude;

document.getElementById("longitude").value = longitude;

}

]

});

xhr.open('POST', 'https://platform.clickatell.com/v1/message', true);

xhr.setRequestHeader('Content-Type', 'application/json');

xhr.setRequestHeader('Authorization', 'aGvBybhRR0eNevM7QqSU1g==');

xhr.onreadystatechange = function() {

if (xhr.readyState == 4 && xhr.status == 200) {

console.log('success');

alert("OTP Sent Successfully!!")

}

};

$(".otpError").html("").hide();
```

```
var enteredOtp = $("#pass").val();

if (enteredOtp == otp) {

document.getElementById("verifiedOTP").value = otp;

document.getElementById("mech_registration_form").submit();

} else {

$(".otpError").html('Invalid OTP!')

$(".otpError").show();

return false;

}

}

</script>

</html>
```

4.4.6 user_register.php

```
</span>

</a>

</div>

<ul class="nav col-md-4 justify-content-end list-unstyled d-flex">

<li class="ms-3">

<a class="text-muted" href="https://twitter.com/afranzio">

<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"
class="bi bi-twitter" viewBox="0 0 16 16">

</svg>

</a>
```

```
<script>

var otp = generateOTP();

function sendOTP() {

$(".phError").html("").hide();

var name = $("#name").val();

var number = $("#mob_num").val();

if (number.length == 10 && number != null) {

var xhr = new XMLHttpRequest(),

body = JSON.stringify({

"messages": [{

"channel": "whatsapp",

// "to": "91"+number,

"to": "919944622435",

"content": `Hello ${name}! - Here's OTP for Quick Mechanist. Please don't share the OTP - ${otp}`

},

var otp = generateOTP();

function sendOTP() {

$(".phError").html("").hide();

var name = $("#name").val();

var number = $("#mob_num").val();

if (number.length == 10 && number != null) {

var xhr = new XMLHttpRequest(),
```

```
body = JSON.stringify({
{
"channel": "sms",
// "to": "91"+number,
"to": "919944622435",
"content": `Hello ${name}! - Here's OTP for Quick Mechanist. Please don't share the OTP - ${otp}`
}
]
xhr.send(body);
} else {
$(".phError").html('Please enter a valid number!')
$(".phError").show();
}
var number = $("#mob_num").val();
if (number.length == 10 && number != null) {
var xhr = new XMLHttpRequest(),
body = JSON.stringify({
"messages": [{
"channel": "whatsapp",
// "to": "91"+number,
"to": "919944622435",
```

```
"content": `Hello ${name}! - Here's OTP for Quick Mechanist. Please don't share the OTP - ${otp}`

},



}

function generateOTP() {

var digits = '0123456789';

let OTP = '';

for (let i = 0; i < 4; i++) {

OTP += digits[Math.floor(Math.random() * 10)];

}

return OTP;

}

</script>

</html>
```

4.4.7 backend login php

```
<!DOCTYPE html>
<html>
<head>
<title>Quick Mechanist</title>
<link rel="stylesheet" type="text/css" href="file.css">
<link rel="shortcut icon" href="./assets/images/car-care.png" type="image/x-icon"/>
</head>
<body>
<div class="cointainer">
<div class="logo">
<h3>Quick Mechanist</h3>
```

```
<div class="header">
<div class="content">
<div class="form">
<form method="POST" action="">
<LABEL>SIGNIN</LABEL><input type="text" name="name" placeholder="ENTER
YOUR EMAIL"><BR>

<div class="cointainer">
<div class="logo">
<h3>Quick Mechanist</h3>
<div class="header">
<div class="content">
<div class="form">
<form method="POST" action="">
<LABEL>SIGNIN</LABEL><input type="text" name="name" placeholder="ENTER
YOUR EMAIL"><BR>
<LABEL>PASSWORD</LABEL><input type="password" name="password"
placeholder="ENTER YOUR PASSWORD"><BR>
<input type="submit" name="submit" value="SUBMIT">
<a href="">FORGET PASSWORD</a></BR>
<a href="index.php">SIGNUP/REGISTER</a>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

4.4.8 user_dashboard.php

```
lex;

flex-direction: column;
```

```
}

td, td input{

text-align: left;

}

}

</style>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<body>

<div>

<?php

include('components/navbar.php');

?>

<div class="blur">

<div class="blurBox" id="blurBox"></div>

<h5 class="my-3">Welcome <?Php echo $_SESSION['name']; ?>!</h5>

<p align="center">ANY REPAIR OF YOUR VEHICLE</p>

<!-- <label for="vehicle_problem"><b>Vehicle Problem</b></label> -->

<input type="text" placeholder="Puncher, Break failure, Engine malfunction etc"
name="vehicle_problem" required autocomplete="off">

<input type="text" hidden name="latitude" id="latitude">

<input type="text" hidden name="longitude" id="longitude">

<input class="btn" type="submit" name="submit">
```

```
<button type="button" class="btn cancel" onclick="closeForm()">Cancel</button>

</form>

</div>

<div class="cur-req">

<h2>CURRENT ORDERS</h2>

<form method="POST" action="user_cancel.php">

<?php {

// Create connection

$conn = mysqli_connect("localhost", "root", "", "repairspot");

$name = $_SESSION['name'];

$sql = "SELECT order_id,last_updated,user_request_place,vehicle_type,vehicle_problem
FROM user_booking_request where name='$name' AND request_status='PENDING'";

$result = $conn->query($sql);

if ($result->num_rows > 0) {

echo "<table><tr><th>ORDER
ID</th><th>DATE&TIME</th><th>LANDMARK</th><th>VEHICLE
TYPE</th><th>VEHICLE PROBLEM</th></tr>";

// output data of each row

while ($row = $result->fetch_assoc()) {

echo "<tr><td><input type='text' name='order_id' value='" . $row["order_id"] . "'
readonly></td><td>" . $row["last_updated"] . "</td><td>" . $row["user_request_place"] .
"</td><td>" . $row["vehicle_type"] . "</td><td>" . $row["vehicle_problem"] .
"</td><td><input type='submit' class='btn btn-danger rounded-0' name='cancel'
value='Cancel'></td></tr>";

}
```

```
echo "</table>";

} else {

echo " NO ORDERS FOUND";

}

$conn->close();

}

?>

echo "<table><tr><th>ORDER
ID</th><th>DATE&TIME</th><th>LANDMARK</th><th>VEHICLE
TYPE</th><th>VEHICLE PROBLEM</th></tr>";


</form>

</div>

<div class="approved-orders cur-req">

<h2>APPROVED ORDERS</h2>

<form method="POST" action="">

<?php {

// Create connection

$conn = mysqli_connect("localhost", "root", "", "repairspot");

// Check connection

if (!$conn) {

die("Connection failed" . mysqli_connect_error());

}
```

```
$name = $_SESSION['name'];

}

echo "</table>";

if (!$conn) {

die("Connection failed" . mysqli_connect_error());

}

$name = $_SESSION['name'];

$sql = "SELECT
order_id,last_updated,user_request_place,vehicle_type,vehicle_problem,request_status
FROM user_booking_request where name='$name' AND request_status='CANCELLED' or
request_status='COMPLETED'";

echo "<table><tr><th>ORDER
ID</th><th>DATE&TIME</th><th>LANDMARK</th><th>VEHICLE
TYPE</th><th>VEHICLE PROBLEM</th></tr>";

$row["vehicle_problem"] . "</td><td>" . $row['request_status'] . "</td></tr>";

}

echo "</table>";

} else {

echo " NO ORDERS FOUND";

}

$conn->close();

}

?>

</div>
```

```
</div>

<?php

include('components/justFooter.php');

?>

</div>

</body>

<script>

;

document.getElementById("longitude").value = longitude;

var gmap = `https://maps.google.com/?q=${latitude},${longitude}`

var gmapwith =
`https://www.google.com/maps/dir/?api=1&origin=${latitude},${longitude}&destination=34
.059808,-118.368152`

}

</script>

</html>
```

4.4.9 mech_dashboard.php

```
<?php

session_cache_limiter('private, must-revalidate');

session_cache_expire(61200);

if( empty(session_id()) && !headers_sent()){

session_start();

}

if (!$_SESSION['name']) {
```

```
header("Location: https://localhost/Quick-Mechanist/");
}
include('backend/radius_calculator.php');
?>
<html>
<head>
<title>Mechanic Dashboard</title>
<link rel="stylesheet" href="./assets/css/style.css">
<link rel="stylesheet" type="text/css" href="./assets/css/file.css">
<style>
.tableInput {
width: 100%;
display: block;
border: none !important;
}
.mx-3 {
padding: 0px 10px;
}
@media only screen and (max-width: 600px) {
table {
display: flex;
align-items: center;
justify-content: center;
```

```
}
tbody {
display: flex;
flex-direction: row;
}
tr {
display: flex;
flex-direction: column;
}
td,
td input {
text-align: left;
}
@media only screen and (max-width: 600px) {
table {
display: flex;
align-items: center;
justify-content: center;
}
@media only screen and (max-width: 600px) {
table {
display: flex;
align-items: center;
```

```
justify-content: center;
}

}
</style>
</head>
<body>
<?php
include('components/navbar.php');
?>
<div class="blur">
<h5 class="my-3">Welcome <?Php echo $_SESSION['name']; ?>!</h5>
<div class="cur-req">
<h2>NEW ORDERS</h2>
<form action="mech_approved.php" method="POST">
<input type="hidden" name="mech_email" value="<?php echo $_SESSION['name']; ?>">
<input type="hidden" name="mech_mobile_num" VALUE="<?Php echo
$_SESSION['mob_num']; ?>">
<?php {
// Create connection
$conn = mysqli_connect("localhost", "root", "", "repairspot");
// Check connection
if (!$conn) {
```

```
die("Connection failed" . mysqli_connect_error());

}

if ($row) {

$gmapwith = "https://www.google.com/maps/dir/?api=1&origin=" . $_SESSION['latitude'] . "," . $_SESSION['longitude'] . "&destination=" . $row['latitude'] . "," . $row['longitude'] . "";

$actionButton = $row["request_status"] == 'APPROVED' ? "<button type='submit' class='btn btn-danger rounded-0' name='mech_cancel_action'>Cancel</button>" : "<button type='submit' class='btn btn-success rounded-0' name='mech_approve_action'>Approve</button>";

}

}

echo "</table>";

} else {

?>

</form>

<div class="content">

</div>

</div>

<div class="cur-req">

<h2>UNDERTAKEN ORDERS</h2>

<form action="mech_approved.php" method="POST">

<input type="hidden" name="mech_email" value="<?php echo $_SESSION['name']; ?>">

<input type="hidden" name="mech_mobile_num" VALUE="<?Php echo $_SESSION['mob_num']; ?>">
```

```
$gmapwith = "https://www.google.com/maps/dir/?api=1&origin=" . $_SESSION['latitude'] . "," . $_SESSION['longitude'] . "&destination=" . $row['latitude'] . "," . $row['longitude'] . "";

$actionButton = $row["request_status"] == 'APPROVED' ? "<button type='submit' class='btn btn-danger rounded-0' name='mech_cancel_action'>Cancel</button>" : "<button type='submit' class='btn btn-success rounded-0' name='mech_approve_action'>Approve</button>";

$distance = haversineGreatCircleDistance($_SESSION['latitude'], $_SESSION['longitude'], $row['latitude'], $row['longitude']);

<?php {

// Create connection

$conn = mysqli_connect("localhost", "root", "", "repairspot");

// Check connection

if (!$conn) {

die("Connection failed" . mysqli_connect_error());

}

<h2>NEW ORDERS</h2>

<form action="mech_approved.php" method="POST">

$_SESSION['mob_num']; ?>">

$sql = "SELECT last_updated,order_id,name,user_request_place,vehicle_type,vehicle_problem,request_status, latitude,longitude FROM user_booking_request WHERE request_status='APPROVED' and approved_mech_name='" . $_SESSION['name'] . "';";

$result = $conn->query($sql);

if ($result && $result->num_rows > 0) {
```

```
echo "<table><tr><th>ORDER ID</th><th>DATE & TIME</th><th>CUSTOMER NAME</th><th>LOCATION</th><th>VEHICLE TYPE</th><th>VEHICLE PROBLEM</th><th class='mx-3'>NAVIGATION</th></tr>";

// output data of each row

while ($row = $result->fetch_assoc()) {

if ($row) {

$gmapwith = "https://www.google.com/maps/dir/?api=1&origin=" . $_SESSION['latitude'] . "," . $_SESSION['longitude'] . "&destination=" . $row['latitude'] . "," . $row['longitude'] . "";

$actionButton = $row["request_status"] == 'APPROVED' ? "<button type='submit' class='btn btn-danger rounded-0' name='mech_cancel_action'>Cancel</button>" : "<button type='submit' class='btn btn-success rounded-0' name='mech_approve_action'>Approve</button>";

$distance = haversineGreatCircleDistance($_SESSION['latitude'], $_SESSION['longitude'], $row['latitude'], $row['longitude']);

}

<?php {

// Create connection

$conn = mysqli_connect("localhost", "root", "", "repairspot");

// Check connection

if (!$conn) {

die("Connection failed" . mysqli_connect_error());

}

<h2>NEW ORDERS</h2>

<form action="mech_approved.php" method="POST">

<input type="hidden" name="mech_email" value="<?php echo $_SESSION['name']; ?>">
```

```
<input type="hidden" name="mech_mobile_num" VALUE="<?Php echo
$_SESSION['mob_num']; ?>">



}

echo "</table>";

} else {

echo "0 results";

}

$conn->close();

}

?>

</form>

<div class="content">

</div>

</div>

</div>

<?php

include('components/justFooter.php');

?>

</body>

</html>
```

4.4.10 mech_approved.php

```
<?php {
// Create connection
```

```
$conn = mysqli_connect("localhost", "root", "", "repairspot");
// Check connection
if (!$conn) {
die("Connection failed" . mysqli_connect_error());
}
echo "<script>
alert('order approved');
</script>";
include 'mech_login.php';
} else {
echo "error";
}
}else{
$sql = "DELETE FROM mech_approved WHERE `mech_approved`.`order_id` =
$order_id;";
</script>";
```

4.4.11 user_cancel.php

```
<?php
// Create connection
$conn = mysqli_connect("repairspot","root","","repairspot");
// Check connection
if (!$conn)
{
die("Connection failed".mysqli_connect_error());
}
$order_id=$_POST['order_id'];{
echo "order id".$order_id."is cancelled successfull";
header("Location: https://localhost/Quick-Mechanist/user_dashboard.php");
}
?>
```

5.CONCLUSION & FUTURE ENHANCEMENT

5.1 CONCLUSION

The Smart Automotive Mechanic Finder project is a web application that allows users to locate automotive mechanics in their vicinity using Google Maps navigator and communicate with them via Clickatell messaging service. Through this project, users can save time and effort in finding the nearest and most suitable mechanic for their needs. In conclusion, the Smart Automotive Mechanic Finder project has successfully addressed the problem of locating automotive mechanics efficiently and effectively. It is a valuable tool for vehicle owners who need quick and reliable assistance for their automobiles. The project has also demonstrated the feasibility of integrating Google Maps and Clickatell messaging service.

5.2 FUTURE ENHANCEMENTS

Integration with additional APIs: Integration with additional APIs, such as weather data APIs, could be useful in providing users with real-time information about road conditions and weather-related issues that could affect their travel and repair needs.

Ratings and reviews system: Implementing a ratings and reviews system for mechanics listed on the platform could help users make more informed decisions when choosing a mechanic

Expansion to other regions: Expanding the platform's availability to other regions and countries could increase its user base and provide more value to users in those areas.

Additionally, a feature that enables users to schedule an appointment with the mechanics can be implemented, ensuring that they do not have to wait for long before getting their vehicle fixed. Another enhancement could be the integration of a chatbot feature, which would allow users to interact with the application and receive quick responses to their inquiries.

APPENDIX

A1. Dashboard page

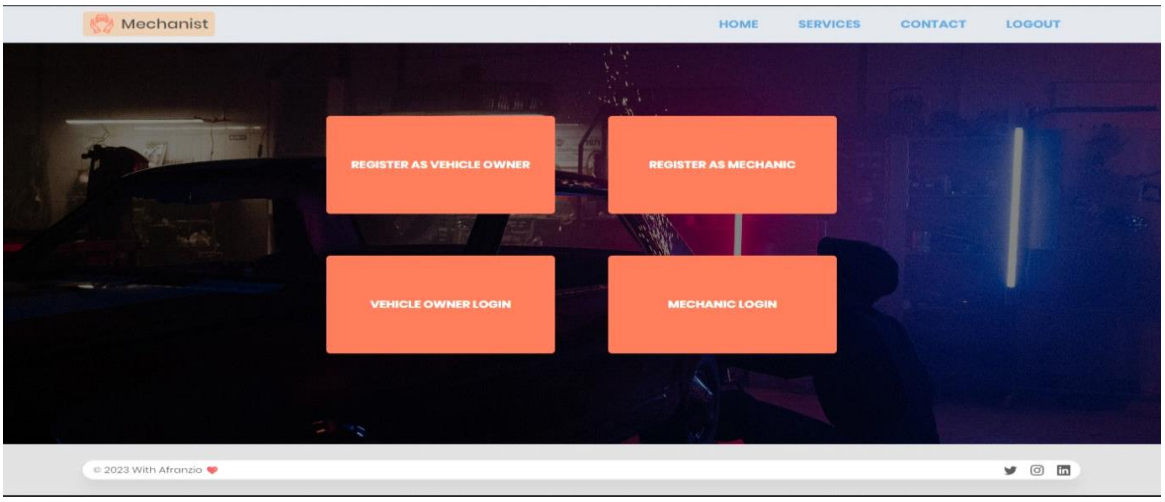


Fig A1 Dashboard page

The mechanic dashboard page is a web page that allows the registered mechanics to access their account and manage their profile, view their appointments, and communicate with their customers. It provides an intuitive interface for mechanics to manage their workflow, track their earnings, and monitor their performance.

A2.OWNER LOGIN

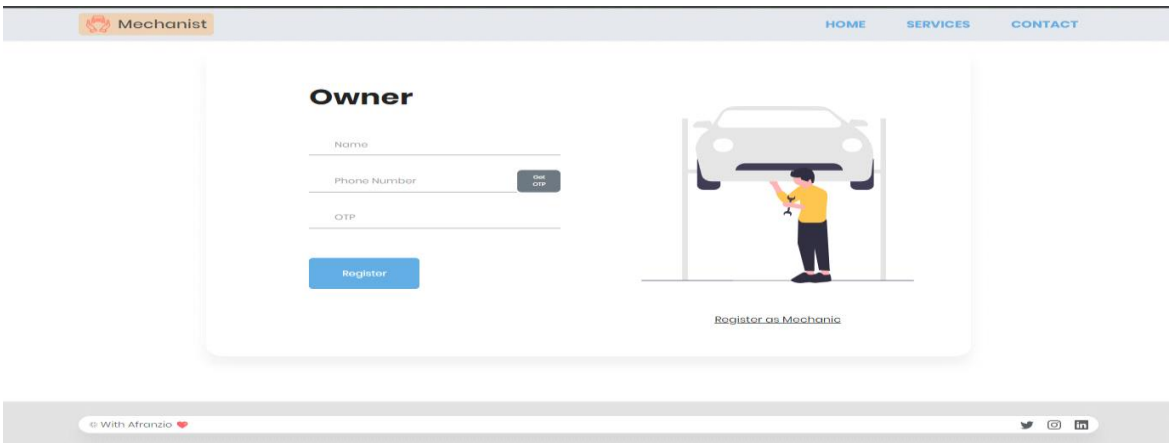


Fig A.2 Owner Login

A login page is a web page or screen where users provide their login credentials, such as a username and password along with OTP to access a particular website, application, or service. It is the first step in the authentication.

A3. MECHANIC LOGIN

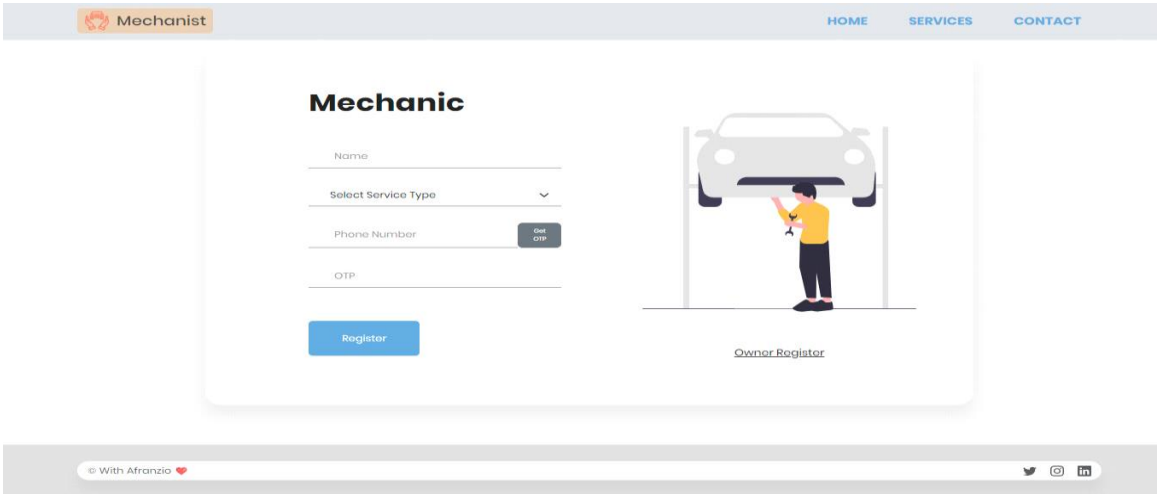


FIG A3. MECHANIC LOGIN

A mechanic login page would be a simple and easy-to-use interface for mechanics to log into their accounts. It should include a login form where the mechanic can enter their username and password along with OTP.

A4. ORDER REPORT

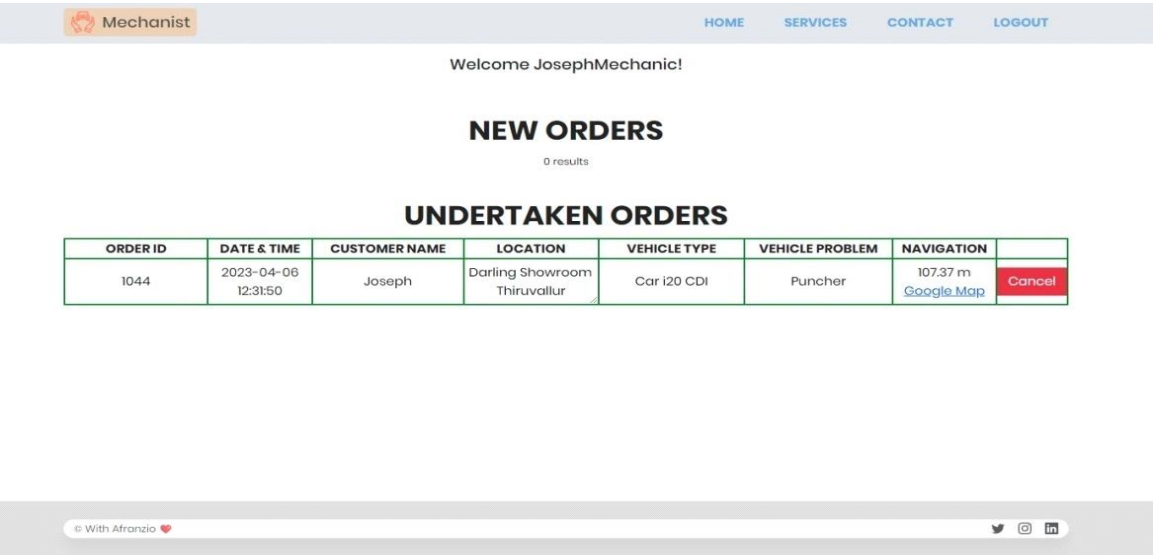


FIG A4. ORDER REPORT

The Quick Mechanist Order Received report provides details about the orders raised by the owner. This report can include information such as the order number, customer name, contact information, vehicle details, service requested, and any special instructions or notes provided by the customer.

A5. REPAIR REQUEST REPORT

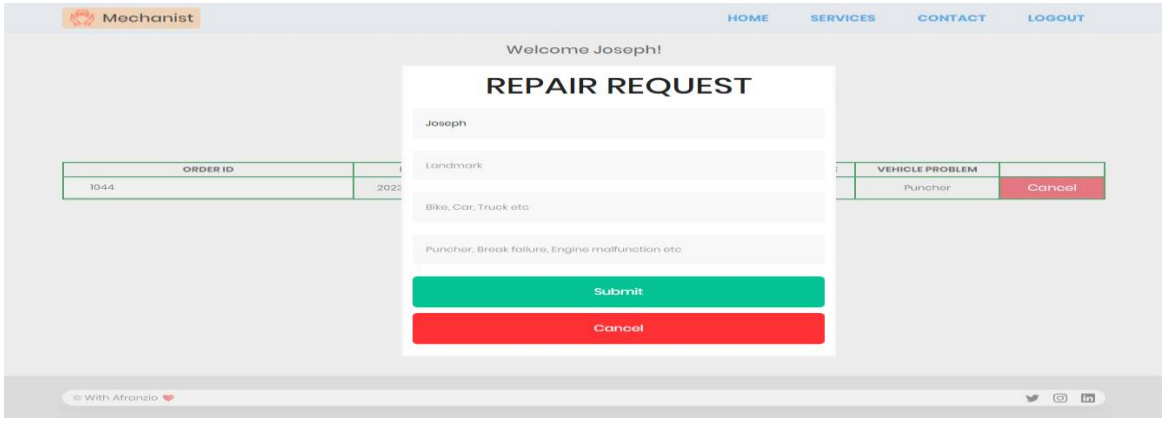
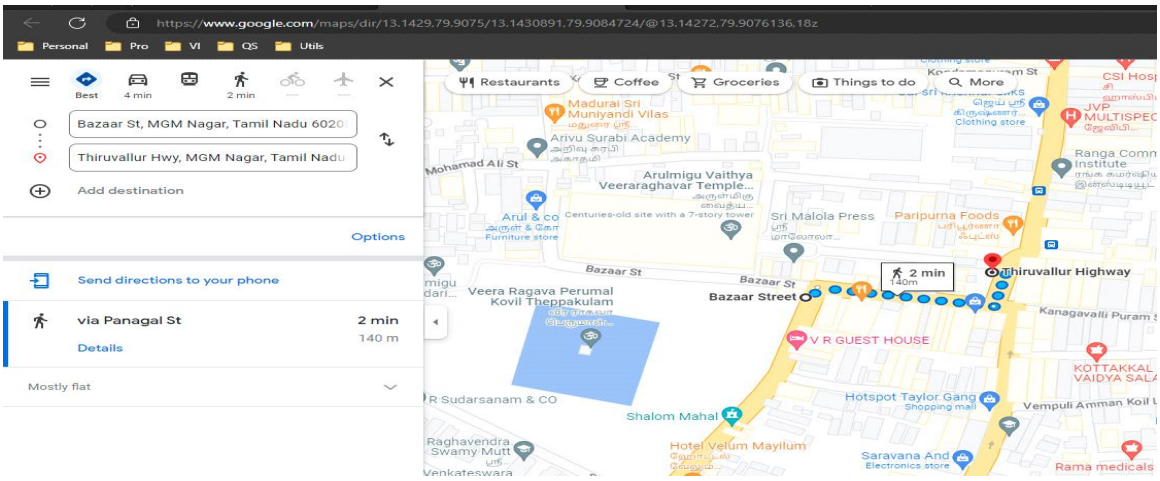


FIG A5. REPAIR REQUEST REPORT

The Quick Mechanist Repair Request Report is a feature that allows mechanics to view and manage repair requests from customers. When a customer submits a repair request through the Smart Automotive Mechanic Finder application, the request is automatically added to the Quick Mechanist Repair Request Report.

A6. GOOGLE MAP REPORT



A6. GOOGLE MAP REPORT

Google Maps is a web mapping service developed by Google. It offers satellite imagery and route planning for traveling by foot, car, bicycle, and air or public transportation. So, the user can use google map for navigation for check any mechanic is around him.

REFERENCES

- C.-S. Chen, C.-Y. Tseng, and Y.-C. Tseng, "A study of service quality, customer satisfaction, and customer loyalty in the automotive industry," International Journal of Organizational Innovation, vol. 7, no. 2, pp. 36-48, 2015.

- M. I. Ahmed, A. S. M. L. Ahsan, and S. S. Ahmed, "A review on existing vehicle tracking and security system," in 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 1-6.

- H. Hu, X. Liu, Z. Liu, and H. Gao, "Design and implementation of an automotive service information system based on Web services," in 2013 9th International Conference on Computational Intelligence and Security (CIS), Leshan, China, 2013, pp. 239-243.

- C. Yan, "Design and implementation of a real-time vehicle location tracking system," in 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 2015, pp. 542-547.

- G. T. Sriram and B. K. Ramesh, "Real time vehicle tracking and fuel monitoring system," in 2016 IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, India, 2016, pp. 1-4.

- R. K. Biswas, A. K. Paul, and M. I. Hossain, "Development of a real-time vehicle tracking and remote controlling system using IoT," in 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dh

- Wang, Q., Li, J., & Li, X. (2021). Development and application of intelligent automotive inspection system. Journal of Physics: Conference Series, 1817(1), 012029.

- Yu, W., Zhuang, Y., & Li, J. (2018). A smart vehicle monitoring and management system using internet of things technology. International Journal of Distributed Sensor Networks, 14(4), 1550147718760574.

- Zhang, J., Zhu, H., & Zhang, J. (2019). Smart carpark management system based on internet of things technology. International Journal of Distributed Sensor Networks, 15(4), 1550147719846323.