# ****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 PROJECT

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 SCOPE OF THE PROJECT

The scope of the Smart Automotive Mechanic Finder using Google map navigator and Clickatell project is to develop a mobile application that allows users to quickly find nearby automotive mechanics based on their location using the Google Maps API. The application will provide users with a list of mechanics, along with their ratings and reviews from other users. Users can then select a mechanic and send a message to them using Clickatell's messaging platform to inquire about their services or schedule an appointment. The application will also allow mechanics to register and create a profile where they can provide information about their services, pricing, and location. The project aims to simplify the process of finding and contactingautomotive mechanics for users, while also providing mechanics with a platform to expand their customer base.

## 1.4 EXISTING SYSTEM

The existing system for finding automotive mechanics typically involves manually searching for service centers 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.

## 1.5 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.

Overall, the proposed system aims to provide a seamless experience for drivers to find and book automotive mechanics, while also helping mechanics manage their appointments and improve their services.

## 1.6 LITERATURE SURVEY

### 1.6.1 OVERVIEW OF RELATED WORK AND RESEARCH

Work and research for Smart Automotive Mechanic Finder using Google map navigator and Clickatell involves exploring the existing systems and technologies for finding mechanics and auto repair services. Some of the related works and research focus on developing location-based systems, using GPS and Google Maps API, to find nearby service providers. Other research focuses on integrating messaging services, such as Clickatell, to allow customers to communicate with service providers and receive updates on their vehicle's repair status. Additionally, some related work explores the use of machine learning and artificial intelligence to optimize the search for service providers and improve the accuracy of service recommendations. Overall, the related work and research highlight the importance of developing efficient and user-friendly systems for finding automotive repair services.

### 1.6.2 ON CLUSTERING MASSIVE TEXT AND CATEGORICAL DATASTREAMSA NEIGHBORHOOD-BASED APPROACH

Text Clustering Clearly Explains The Nature Of Leakage And Possibilities To Avoid Leakage. It Specifies How The Distributor Can “Intelligently” Give Data To Agents In Order To Improve The Chances Of Detecting A Guilty Agent. 6 By Adding Fake Objects To Distributed Set, The Distributor Can Find The Guilt Agent Easily.

### 1.6.3 CO-CLUSTERING DOCUMENTS AND WORDS USING BIPARTITE

For Clustering Of Linked Document Collections Clustering Can Be Applied To Any Database Relation Having Attributes Which Are Such That Changes In A Few Of Their Values Do Not Affect The Applications. The Tuples Attributes Within A Tuple, Bit Positions In An Attribute, And Specific Bit Values Are All Algorithmically Determined Under The Control Of A Private Key Known Only To The Owner Of The Data.

# ****2. SYSTEM SPECIFICATION****

## ****2.1 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

## ****2.2 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

## 2.3 SOFTWARE DESCRIPTION

* XAMPP
* CSS
* HTML
* JAVASCRIPT
* PHP
* MySQL
* GOOGLE MAP
* CLICKATELL

### 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 Preprocessor".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.

### 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.

### CHARACTERISTICS OF PHP

### Five important characteristics make PHP's practical nature possible

* Simplicity
* Efficiency
* 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.

## 2.5 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](https://www.javatpoint.com/windows), [Linux](https://www.javatpoint.com/linux-tutorial), 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.

## 2.6 JAVASCRIPT

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server-side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as Realtime collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects, for example:

Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

This means that in the browser, JavaScript can change the way the webpage (DOM) looks. And, likewise, Node.js JavaScript on the server can respond to custom requests from code written in the browser.

## 2.7 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.

## 2.8 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.

## 2.9 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

## 2.10MySQL

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.

## 2.11 HTML

Hypertext Markup Language is the computer language that facilitates website creation. The language, which has code words and syntax just like any other language, is relatively easy to comprehend and, as time goes on, increasingly powerful in what it allows someone to create. HTML continues to evolve to meet the demands and requirements of the Internet under the guise of the World Wide Web Consortium, the organization that designs and maintains the language; for instance, with the transition to [Web 2.0](https://www.investopedia.com/terms/w/web-20.asp).Hypertext is the method by which Internet users navigate the web. By clicking on special text called hyperlinks, users are brought to new pages. The use of hyper means it is not linear, so users can go anywhere on the Internet simply by clicking on the available links. Markup is what HTML tags do to the text inside of them; they mark it as a specific type of text. For example, markup text could come in the form of boldface or italicized type to draw specific attention to a word or phrase.

## 2.12 HYPERTEXT MARKUP LANGUAGE BASICS

At its core, HTML is a series of short codes typed into a text-file. These are the tags that power HTML’s capabilities. The text is saved as an HTML file and viewed through a web browser. The browser reads the file and translates the text into a visible form, as directed by the codes the author used to write what becomes the visible rendering. Writing HTML requires tags to be used correctly to create the author’s vision.The tags are what separate normal text from HTML code. Tags are the words between what are known as angle-brackets, which allow graphics, images, and tables to appear on the webpage. Different tags perform different functions. The most basic tags apply formatting to text. As web interfaces need to become more dynamic, Cascading Style Sheets (CSS) and JavaScript applications may be used. CSS makes web pages more accessible, and JavaScript adds power to basic HTML.

## 2.13 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.PHP 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 it. The development tools and database servers that require licensing costs have limited programming resources compared to open-source development tools, which have an enormous and fast growing dedicated and knowledgeable community that extends around the world. There has been disagreement about which tool is better. Naturally, the developer who is more familiar with one tool over the other will stand behind the tool that he or she has experience with. With our experience, we have found that, PHP and MySQL are the best development tools. When developed correctly, applications can be built with clean and simple usability, complex functionality, speed, power and scalability.

## 2.14 GOOD DEVELOPMENT PRACTICES

* One item to note: No matter how great the development tools are that you are developing with, if the application is not developed correctly, the end result may not be worthwhile.
* PHP and MySQL are built so that anyone with a minimum amount of programming knowledge can put together a browser-based web application.
* This easy entry to PHP development can also create problems when an inexperienced programmer takes on a large Web Application project and does not consider security implications, scalability and the SQL execution time when there are large concurrent connections.

## 2.15 EXAMPLES OF MESSY CODING AND SLOPPY SQL QUERIES

* Not using coding guidelines
* Mixing HTML with PHP functions
* Coding inconstancies
* Using different styles of function calls

## 2.16 RELEASES AND MILESTONES

|  |  |
| --- | --- |
| **DATE** | **MILESTONES** |
| 6/8/1995 | PHP 1.0 was officially released |
| 11/1/1997 | PHP 2.0 was released |
| 6/6/1998 | PHP 3.0 was released by ZeevSuraski and Andi Gutmans with a rebuilt parser |
| 5/22/2000 | PHP 4.0 was released and powered by Zend Engine 1.0 |
| 2/27/2002 | PHP 4.3 was released - this was a major release - PHP reached over 10,000,000 domains |
| 7/13/2004 | PHP 5.0 was released - another major milestone - powered by Zend Engine 2.0 - PHP True Object was implemented |
| 6/30/2009 | PHP 5.3 was released - support of anonymous functions and closures introduced, major improvement in object handling and functions |

# 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.

## 3.2 SYSTEM DESIGN

Design is multi-step process that focuses on data structure software architecture, procedural details, (algorithms etc.) and interface between modules. The design process also translates the requirements into the presentation of software that can be accessed for quality before coding begins.

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.

## 3.3 INPUT DESIGN

Input design is the process of converting user -originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system

## 3.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system form any end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system,the output design determines the input to be given to the application.

## 3.5 SYSTEM ARCHITECTURE

The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design. The output of this design process is a description of the software architecture.

* It defines an abstraction level at which the designers can specify the functional and performance behaviour of the system.
* It evaluates all top –level designs
* It develops and documents top-level designs for the external and internal interface
* It develops all preliminary versions of users documentation
* It defines and documents preliminary test requirements and the schedule for software integration.

System architecture is a conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure and behaviour of the system.

Fig3.1 System Architecture Architecture

Design Representation:

Architecture design can be represented using the following models.

1. Structural model: Illustrate architecture as an ordered collection of program components.
2. Dynamic model: Specifies the behavioural aspect of the software architecture and indicates how the structure or system configuration changes as the function changes due to change in the external environment.
3. Process model: Focuses on the design of the business or technical process which must be implemented in the system.
4. Functional model: Represents the functional hierarchy of a system.
5. Framework model: Attempts to identify repeatable architectural design patterns encountered in similar types of application. This leads to an increase in the level of abstraction.

# 3.6 ****MODULE****

This web application consists of two modules admin module, user module:

## ****3.6.1 ADMIN MODULE****

This module contains dashboard, users count and feedbacks and allows adding, deleting, and modifying of new, jobs available. Admin can add city, stream, and more details about jobs. Admin can change his password and change the privacy and contact number on the user side.

## 3.6.2 ADMIN IMPORTANT PHP FILES

* Login\_pg.php – This file contains a login page.
* Index.php - This file contains a dashboard
* Admin\_header.php -this file is header nav bar and menu of the admin side
* Admin\_footer.php -this file contains admin footer
* Db\_con.php – this the only file which connects to the database mysql

Full\_time.php, Part\_time.php – these pages are show the jobs details to the admin Demo connection for the above pages Include () function,

**Fig 3.2 Admin Module**

## ****3.6.3 USER MODULE****

Users should register themselves, they can login using their email and password. Then they can search job according to their interest. User can contact to the employer for jobs by clicking on the jobs displayed.

## 3.6.4 USERS IMPORTANT PHP FILES

* Login\_home.php– This is the first page of the user side login page
* Seek.php– This is page for job seekers for the user side
* Search\_jb.php– search page of the user side
* Gmail\_login.php– the page for gmail login using google api
* Head.php– this file contains nav bar for the all-user side pages
* Foot.php– this is the footer for the all-user pages
* Seek\_results.php– this is page for show the search result of the jobs
* Backend-search.php– this file helps to search keywords for the search\_jb.php page

This all are very important files or pages in the admin side.

Demo connection for the above pages

Include () function,

**Fig 3.2 User Module**

# 3.7 USE CASE DIAGRAM

Use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process. A use case describes a function that a system performs to achieve the user’s goal. A use case must yield an observable result that is of value to the user of the system. Use cases contain detailed information about the system, the system’s users, relationships between the system and the users, and the required behaviour of the system. Use cases do not describe the details of how the system is implemented. Each use case describes a particular goal for the user and how the user interacts with the system to achieve that goal. The use case describes all possible ways that the system can achieve, or fail to achieve, the goal of the user.

You can use cases for the following purposes:

* Determine the requirements of the system
* Describe what the system should do
* Provide a basis for testing to ensure that the system works as intended

In models that depict businesses, use cases represent the processes and activities of the business. In models that depict software systems, use cases represent the capabilities of the software. Each use case must have a unique name that describes the action that the system performs. Use case names are often short phrases that start with a verb, such as Place Order Online.

Use-case diagrams are helpful in the following situations:

* Before starting a project, you can create use-case diagrams to model a business so that all participants in the project share an understanding of the workers, customers, and activities of the business.
* While gathering requirements, you can create use-case diagrams to capture the system requirements and to present to others what the system should do.
* During the analysis and design phases, you can use the use cases and actors from your use-case diagrams to identify the classes that the system requires.
* During the testing phase, you can use use-case diagrams to identify tests for the system.

The following topics describe model elements in use-case diagrams:

* + [**Use cases**](https://www.ibm.com/docs/en/SS8PJ7_9.7.0/com.ibm.xtools.modeler.doc/topics/cuc.html)  
    A use case describes a function that a system performs to achieve the user’s goal. A use case must yield an observable result that is of value to the user of the system.
  + [**Actors**](https://www.ibm.com/docs/en/SS8PJ7_9.7.0/com.ibm.xtools.modeler.doc/topics/cactor.html)  
    An actor represents a role of a user that interacts with the system that you are modelling. The user can be a human user, an organization, a machine, or another external system.
  + [**Subsystems**](https://www.ibm.com/docs/en/SS8PJ7_9.7.0/com.ibm.xtools.modeler.doc/topics/csubsys.html)  
    In UML models, subsystems are a type of stereotyped component that represent independent, behavioural units in a system. Subsystems are used in class, component, and use-case diagrams to represent large-scale components in the system that you are modelling.
    - [**Relationships in use-case diagrams**](https://www.ibm.com/docs/en/SS8PJ7_9.7.0/com.ibm.xtools.modeler.doc/topics/crelsme_ucd.html)  
      In UML, a relationship is a connection between model elements. A UML relationship is a type of model element that adds semantics to a model by defining the structure and behaviour between the model elements.

As the following figure illustrates, a use case is displayed as an oval that contains the name of the use case.

You can add the following features to use cases:

* Attributes that identify the properties of the objects in a use case
* Operations that describe the behaviour of objects in a use case and how they affect the system
* Documentation that details the purpose and flow of events in a use case

## Actors

An actor represents a role of a user that interacts with the system that you are modeling. The user can be a human user, an organization, a machine, or another external system. You can represent multiple users with a single actor and a single user can have the role of multiple actors. Actors are external to the system. They can initiate the behaviour described in the use case or be acted upon by the use case. Actors can also exchange data with the system.

# 3.8 ER-DIAGRAM:

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses [data modeling](https://www.techtarget.com/searchdatamanagement/definition/data-modeling) techniques that can help define business processes and serve as the foundation for a [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database).

# 3.8.1 IMPORTANCE OF ERDS AND THEIR USES

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database) is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later. However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.

ERDs are generally depicted in one or more of the following models:

* A conceptual data model, which lacks specific detail but provides an overview of the scope of the project and how [data sets](https://www.techtarget.com/whatis/definition/data-set) relate to one another.
* A logical data model, which is more detailed than a conceptual data model, illustrating specific [attributes](https://www.techtarget.com/whatis/definition/attribute) and relationships among [data points](https://www.techtarget.com/whatis/definition/data-point). While a conceptual data model does not need to be designed before a logical data model, a physical data model is based on a logical data model.
* A physical data model, which provides the blueprint for a physical manifestation -- such as a relational database -- of the logical data model. One or more physical data models can be developed based on a logical data model.

There are five basic components of an entity relationship diagram. Similar components will be designated by the same shape. For example, all entity types might be enclosed in a rectangle, while all attributes are enclosed in a diamond. The components include:

1. Entities, which are objects or concepts that can have data stored about them. Entities refer to tables used in databases.
2. Attributes, which are properties or characteristics of entities. An ERD attribute can be denoted as a [primary key](https://searchsqlserver.techtarget.com/definition/primary-key), which identifies a unique attribute, or a [foreign key](https://www.techtarget.com/searchoracle/definition/foreign-key), which can be assigned to multiple attributes.
3. The relationships between and among those entities.
4. Actions, which describe how entities share information in the database.
5. Connecting lines

For example, an ERD representing the information system for a company's sales department might start with graphical representations of entities such as the sales representative, the customer, the customer's address, the customer's order, the product, and the warehouse. (See diagram above.) Then lines or other symbols can be used to represent the relationship between entities, and text can be used to label the relationships.

The three main cardinalities are:

1. A one-to-one relationship (1:1). For example, if each customer in a database is associated with one mailing address.
2. A one-to-many relationship (1:M). For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.
3. A many-to-many relationship (M: N). For example, at a company where all call centre agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

# 3.9 DATABASE NAME:

## 3.9.1 CREATING A MYSQL TABLE USING MYSQLI AND PDO

We have already learned about creating databases in MySQL from PHP in this article. The steps to create table are like creating databases. The difference is instead of creating a new database we will connect to existing database and create a table in that database. To connect to an existing database, we can pass an extra variable “database name” while connecting to MySQL. The CREATE TABLE statement is used to create a table in MySQL.

## 3.9.2 VARCHAR:

Holds a variable length string that can contain letters, numbers, and special characters. The maximum size is specified in parenthesis.

## 3.9.3 INT:

The INTEGER data type accepts numeric values with an implied scale of zero. It stores any integer value between -2147483648 to 2147483647.

The attributes that are used along with data types in this article are:

## 3.9.4 NOT NULL:

Each row must contain a value for that column, null values are not allowed.

## 3.9.5 PRIMARY KEY:

**Used to** uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number.

## 3.9.6DATABASE TABLES:

Loop testing is a type of white-box testing technique that is used to test individual loops (such as for, while, and do-while loops) within a software program. It involves testing the loop with various sets of input values to ensure that it executes correctly and does not result in any unexpected behaviors or errors.

Here is an example of a loop testing table for Smart Automotive Mechanic Finder using Google map navigator and Clickatell project:

| **Loop Identifier** | **Test Case #** | **Test Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| For loop 1 | 1 | i = 1 | True | True | Pass |
|  | 2 | i = 3 | True | True | Pass |
|  | 3 | i = 0 | False | False | Pass |
| For loop 2 | 1 | j = 10 | True | True | Pass |
|  | 2 | j = -5 | False | False | Pass |
|  | 3 | j = 0 | False | False | Pass |
| While loop 1 | 1 | k = 1 | True | True | Pass |
|  | 2 | k = 5 | True | True | Pass |
|  | 3 | k = 0 | False | False | Pass |
| While loop 2 | 1 | l = 10 | True | True | Pass |
|  | 2 | l = -5 | False | False | Pass |
|  | 3 | l = 0 | False | False | Pass |

In this table, we have identified four loops within the software program (two for loops and two while loops) and created test cases for each loop. For each test case, we have provided the input values, the expected output, and the actual output. We have also indicated whether the test case passed or failed based on whether the actual output matched the expected output.

By performing loop testing, we can ensure that each individual loop within the program is functioning correctly and that the program as a whole is robust and free from errors.

## 3.10 ALGORITHMS:

### For Owner login page:

Step 1: Start (click login for owner register)

Step 2: Enter phone number in given place

Step 3: click otp icon

Step 4: Enter otp at given place

Step 5: click login icon

Step 6: End

### For Mechanic login page:

Step 1: Start (click login for owner register)

Step 2: Enter phone number in given place

Step 3: click otp icon

Step 4: Enter otp at given place

Step 5: click login icon

Step 6: End

# 4. IMPLEMENTATION AND TESTING

## 4.1 CODING

### 4.1.1 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><br>

         </a>

      </div>

      <div class="d-flex">

         <a class="noHover" href="user\_login.php">

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

         </a>

         <a class="noHover" href="mech\_login.php">

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

         </a>

      </div>

   </div>

</div>

<?php

include('components/footer.php');

?>

### 4.1.2 user\_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 rel="shortcut icon" href="./assets/images/car-care.png" type="image/x-icon"/>

    <!-- Styles and Fonts -->

    <!-- Bootstrap -->

    <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">

    <!-- Custom CSS -->

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

    <!-- Font Icon -->

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

    <!-- Main CSS -->

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

    <script type="text/javascript">

        //password match

        function Validate() {

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

            var confirmPassword = document.getElementById("confirm\_password").value;

            if (password != confirmPassword) {

                alert("Passwords do not match.");

                return false;

            }

            return true;

        }

        //password checkbox

        function showPassword() {

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

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

                x.type = "text";

            } else {

                x.type = "password";

            }

        }

    </script>

    <?php

    if( empty(session\_id()) && !headers\_sent()){

        session\_start();

    }

    if (isset($\_SESSION['name'])) {

        // Create connection

        $conn = mysqli\_connect("localhost", "root", "", "repairspot");

        // Check connection

        if (!$conn) {

            die("Connection failed" . mysqli\_connect\_error());

        }

        $name = $\_SESSION['name'];

        $sql = "SELECT \* FROM user\_details where mob\_num='".$\_SESSION['mob\_num']."'";

        $result = $conn->query($sql);

        if ($result->num\_rows > 0) {

            header("Location: https://localhost/Quick-Mechanist/user\_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');

        ?>

        <section class="signup mt-3 mb-0">

            <div class="container">

                <div class="signup-content">

                    <div class="signup-form">

                        <h2 class="form-title">Owner-Login</h2>

                        <form id="user\_login" method="POST" action="./backend/user\_login\_back.php">

                            <div class="form-group">

                                <div class="d-flex">

                                    <input type="tel" maxlength="10" name="mob\_num" id="mob\_num" placeholder="Phone Number" />

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

                                        Get OTP

                                    </button>

                                </div>

                                <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">

                        <figure>

                            <img src="./assets/images/undraw\_bike\_ride.svg" alt="sing up image">

                        </figure>

                        <a href="https://localhost/Quick-Mechanist/mech\_login.php" class="signup-image-link">Mechanic Login</a>

                    </div>

                </div>

            </div>

        </section>

    </div>

    <?php

    include('components/justFooter.php');

    ?>

</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>

    var otp = generateOTP();

    function sendOTP() {

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

        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 there! Here's OTP for Quick Mechanist. Please don't share the OTP - ${otp}`

                    },

                    {

                        "channel": "sms",

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

                        "to": "919944622435",

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

                    }

                ]

            });

            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!!")

                }

            };

            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>

### 4.1.3 user\_register.php

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">

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

    <title>Owner Registration</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" />

    <!-- Font Icon -->

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

    <!-- Main css -->

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

    <script type="text/javascript">

        //password checkbox

        function showPassword() {

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

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

                x.type = "text";

            } else {

                x.type = "password";

            }

        }

    </script>

    <?php

    session\_start();

    if (isset($\_SESSION['name'])) {

        header("Location: https://localhost/Quick-Mechanist/user\_dashboard.php");

    }

    ?>

    <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>

        <div class="navbar" style="background-color: #dee2e6;">

            <div class="w-75 m-auto d-flex justify-content-between align-self-center">

                <a href="/" style="text-decoration: none; color: rgba(33, 37, 41, 0.75);">

                    <div class="icon align-self-center">

                        <img src="./assets/images/car-care.png" alt="" srcset="" class="logoImg">

                        <h5>

                            Mechanist

                        </h5>

                    </div>

                </a>

                <div class="menu">

                    <ul class="mb-0">

                        <li><a href="./index.php"> HOME</a></li>

                        <li><a href="./uncode/services.php"> SERVICES</a></li>

                        <li><a href="./uncode/contact.php"> CONTACT</a></li>

                    </ul>

                </div>

            </div>

        </div>

        <section class="signup mt-3 mb-0">

            <div class="container">

                <div class="signup-content">

                    <div class="signup-form">

                        <h2 class="form-title">Owner</h2>

                        <form name="my" id="user\_registration\_form" method="POST" action="./backend/user\_register\_back.php">

                            <div class="form-group">

                                <input type="text" name="name" id="name" placeholder="Name" />

                            </div>

                            <div class="form-group">

                                <div class="d-flex">

                                    <input type="tel" name="mob\_num" id="mob\_num" maxlength="10" placeholder="Phone Number" />

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

                                        Get OTP

                                    </button>

                                </div>

                                <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" type="button" value="Register" onclick="return verifyOTP()" />

                            </div>

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

                        </form>

                    </div>

                    <div class="signup-image">

                        <figure>

                            <img src="./assets/images/undraw\_automobile\_repair.svg" alt="sing up image">

                        </figure>

                        <a href="https://localhost/Quick-Mechanist/mech\_register.php" class="signup-image-link">Register as Mechanic</a>

                    </div>

                </div>

            </div>

        </section>

    </div>

    <div>

        <footer class="d-flex flex-wrap justify-content-between align-items-center py-4 my-0 border-top">

            <div class="container d-flex w-100 justify-content-between">

                <div class="col-md-4 d-flex align-items-center">

                    <a href="https://afranzio.github.io" target="\_blank" class="mb-0 me-2 mb-md-0 text-muted text-decoration-none lh-1">

                        <span class="mb-0 mb-md-0 text-muted">

                            © <span id="year"></span>

                        </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">

                                <path d="M5.026 15c6.038 0 9.341-5.003 9.341-9.334 0-.14 0-.282-.006-.422A6.685 6.685 0 0 0 16 3.542a6.658 6.658 0 0 1-1.889.518 3.301 3.301 0 0 0 1.447-1.817 6.533 6.533 0 0 1-2.087.793A3.286 3.286 0 0 0 7.875 6.03a9.325 9.325 0 0 1-6.767-3.429 3.289 3.289 0 0 0 1.018 4.382A3.323 3.323 0 0 1 .64 6.575v.045a3.288 3.288 0 0 0 2.632 3.218 3.203 3.203 0 0 1-.865.115 3.23 3.23 0 0 1-.614-.057 3.283 3.283 0 0 0 3.067 2.277A6.588 6.588 0 0 1 .78 13.58a6.32 6.32 0 0 1-.78-.045A9.344 9.344 0 0 0 5.026 15z" />

                            </svg>

                        </a>

                    </li>

                    <li class="ms-3">

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

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

                                <path d="M8 0C5.829 0 5.556.01 4.703.048 3.85.088 3.269.222 2.76.42a3.917 3.917 0 0 0-1.417.923A3.927 3.927 0 0 0 .42 2.76C.222 3.268.087 3.85.048 4.7.01 5.555 0 5.827 0 8.001c0 2.172.01 2.444.048 3.297.04.852.174 1.433.372 1.942.205.526.478.972.923 1.417.444.445.89.719 1.416.923.51.198 1.09.333 1.942.372C5.555 15.99 5.827 16 8 16s2.444-.01 3.298-.048c.851-.04 1.434-.174 1.943-.372a3.916 3.916 0 0 0 1.416-.923c.445-.445.718-.891.923-1.417.197-.509.332-1.09.372-1.942C15.99 10.445 16 10.173 16 8s-.01-2.445-.048-3.299c-.04-.851-.175-1.433-.372-1.941a3.926 3.926 0 0 0-.923-1.417A3.911 3.911 0 0 0 13.24.42c-.51-.198-1.092-.333-1.943-.372C10.443.01 10.172 0 7.998 0h.003zm-.717 1.442h.718c2.136 0 2.389.007 3.232.046.78.035 1.204.166 1.486.275.373.145.64.319.92.599.28.28.453.546.598.92.11.281.24.705.275 1.485.039.843.047 1.096.047 3.231s-.008 2.389-.047 3.232c-.035.78-.166 1.203-.275 1.485a2.47 2.47 0 0 1-.599.919c-.28.28-.546.453-.92.598-.28.11-.704.24-1.485.276-.843.038-1.096.047-3.232.047s-2.39-.009-3.233-.047c-.78-.036-1.203-.166-1.485-.276a2.478 2.478 0 0 1-.92-.598 2.48 2.48 0 0 1-.6-.92c-.109-.281-.24-.705-.275-1.485-.038-.843-.046-1.096-.046-3.233 0-2.136.008-2.388.046-3.231.036-.78.166-1.204.276-1.486.145-.373.319-.64.599-.92.28-.28.546-.453.92-.598.282-.11.705-.24 1.485-.276.738-.034 1.024-.044 2.515-.045v.002zm4.988 1.328a.96.96 0 1 0 0 1.92.96.96 0 0 0 0-1.92zm-4.27 1.122a4.109 4.109 0 1 0 0 8.217 4.109 4.109 0 0 0 0-8.217zm0 1.441a2.667 2.667 0 1 1 0 5.334 2.667 2.667 0 0 1 0-5.334z" />

                            </svg>

                        </a>

                    </li>

                    <li class="ms-3">

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

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

                                <path d="M0 1.146C0 .513.526 0 1.175 0h13.65C15.474 0 16 .513 16 1.146v13.708c0 .633-.526 1.146-1.175 1.146H1.175C.526 16 0 15.487 0 14.854V1.146zm4.943 12.248V6.169H2.542v7.225h2.401zm-1.2-8.212c.837 0 1.358-.554 1.358-1.248-.015-.709-.52-1.248-1.342-1.248-.822 0-1.359.54-1.359 1.248 0 .694.521 1.248 1.327 1.248h.016zm4.908 8.212V9.359c0-.216.016-.432.08-.586.173-.431.568-.878 1.232-.878.869 0 1.216.662 1.216 1.634v3.865h2.401V9.25c0-2.22-1.184-3.252-2.764-3.252-1.274 0-1.845.7-2.165 1.193v.025h-.016a5.54 5.54 0 0 1 .016-.025V6.169h-2.4c.03.678 0 7.225 0 7.225h2.4z" />

                            </svg>

                        </a>

                    </li>

                </ul>

            </div>

        </footer>

    </div>

    </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>

    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}`

                    },

                    {

                        "channel": "sms",

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

                        "to": "919944622435",

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

                    }

                ]

            });

            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!!")

                }

            };

            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("user\_registration\_form").submit();

        } else {

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

            $(".otpError").show();

            return false;

        }

    }

</script>

</html>

### 4.1.4 user\_dashboard.php

<?php

session\_cache\_limiter('private, must-revalidate');

session\_cache\_expire(120);

session\_start();

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

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

}

?>

<!DOCTYPE html>

<html>

<head>

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

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

  <title>Owner</title>

  <!-- Styles and Fonts -->

  <!-- Bootstrap -->

  <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">

  <!-- Custom CSS -->

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

  <!-- Font Icon -->

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

  <!-- Main CSS -->

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

  <style>

    body {

      font-family: Arial, Helvetica, sans-serif;

    }

    \* {

      box-sizing: border-box;

    }

    /\* Button used to open the contact form - fixed at the bottom of the page \*/

    .open-button {

      background-color: #555;

      color: white;

      padding: 16px 20px;

      border: none;

      cursor: pointer;

      opacity: 0.8;

      position: center;

      bottom: 23px;

      right: 28px;

      width: 280px;

    }

    /\* The popup form - hidden by default \*/

    .form-popup {

      position: absolute;

      left: 15%;

      top: 15%;

      display: none;

      z-index: 9;

      width: 75%;

      margin: auto;

    }

    /\* Add styles to the form container \*/

    .form-container {

      max-width: 500px;

      padding: 10px;

      background-color: white;

      border: 3px solid #f1f1f1;

      margin: auto;

    }

    /\* Full-width input fields \*/

    .form-container input[type=text],

    .form-container input[type=password] {

      width: 100%;

      padding: 15px;

      margin: 5px 0 22px 0;

      border: none;

      background: #f1f1f1;

    }

    /\* When the inputs get focus, do something \*/

    .form-container input[type=text]:focus,

    .form-container input[type=password]:focus {

      background-color: #ddd;

      outline: none;

    }

    /\* Set a style for the submit/login button \*/

    .form-container .btn {

      background-color: #04AA6D;

      color: white;

      padding: 16px 20px;

      border: none;

      cursor: pointer;

      width: 100%;

      margin-bottom: 10px;

      opacity: 0.8;

    }

    /\* Add a red background color to the cancel button \*/

    .form-container .cancel {

      background-color: red;

    }

    /\* Add some hover effects to buttons \*/

    .form-container .btn:hover,

    .open-button:hover {

      opacity: 1;

    }

    /\* table for current orders \*/

    table,

    th,

    td {

      border: 2px solid green;

    }

    .blur,

    .cur-req,

    .cur-req form,

    .cur-req form table,

    .odr-htry,

    .odr-htry table {

      text-align: center;

      align-items: center;

      justify-content: center;

      width: 100%;

      overflow: hidden;

    }

    .cur-req form table,

    .odr-htry table {

      width: 75%;

      margin: auto;

    }

    h2 {

      margin-top: 2rem;

    }

    body {

      overflow: hidden;

    }

    .blurBox {

      width: 100vw;

      height: 100vh;

      position: absolute;

      z-index: 8;

      background-color: #cccccc6e;

      -webkit-filter: blur(2px);

      -moz-filter: blur(2px);

      -o-filter: blur(2px);

      -ms-filter: blur(2px);

      filter: blur(2px);

      display: none;

    }

    @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;

      }

    }

  </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>

      <button class="open-button" onclick="openForm()">REQUEST TO REPAIR</button>

      <div class="form-popup" id="myForm">

        <form action="./backend/user\_dashboard\_back.php" method="POST" class="form-container">

          <h1 style="border-bottom: 1px solid #f1f1f1;" class="mb-3">REPAIR REQUEST</h1>

          <!-- <label for="name"><b>Name</b></label> -->

          <input type="text" value="<?Php echo $\_SESSION['name']; ?>" name="name" readonly>

          <!-- <label for="location"><b>Landmark</b></label> -->

          <input type="text" placeholder="Landmark" name="location" required autocomplete="off">

          <!-- <label for="vehicle\_type"><b>Vehicle</b></label> -->

          <input type="text" placeholder="Bike, Car, Truck etc" name="vehicle\_type" required autocomplete="off">

          <!-- <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");

            // Check connection

            if (!$conn) {

              die("Connection failed" . mysqli\_connect\_error());

            }

            $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();

          }

          ?>

        </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'];

            $sql = "SELECT order\_id,approved\_mech\_name,mech\_mobile\_num,order\_status,approved\_datetime FROM mech\_approved where user\_name='$name'";

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

            if ($result && $result->num\_rows > 0) {

              echo "<table><tr><th>ORDER ID</th><th>MECHANIC EMAIL</th><th>MECHANIC MOBILE NUM</th><th>ORDER STATUS</th><th>APPROVED DATE&TIME</th></tr>";

              // output data of each row

              while ($row = $result->fetch\_assoc()) {

                echo "<tr><td>" . $row["order\_id"] . "</td><td>" . $row["approved\_mech\_name"] . "</td><td>" . $row['mech\_mobile\_num'] . "</td><td>" . $row["order\_status"] . "</td><td>" . $row["approved\_datetime"] . "</td></tr>";

              }

              echo "</table>";

            } else {

              echo " NO ORDERS FOUND";

            }

            $conn->close();

          }

          ?>

        </form>

      </div>

      <div class="odr-htry">

        <h2>ORDER HISTORY</h2>

        <?php {

          // Create connection

          $conn = mysqli\_connect("localhost", "root", "", "repairspot");

          // Check connection

          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'";

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

          if ($result->num\_rows > 0) {

            echo "<table><tr><th>ORDER ID</th><th>ORDER CANCELLED/COMPLETED<br>DATE&TIME</th><th>LOCATION</th><th>VEHICLE TYPE</th><th>VEHICLE PROBLEM</th><th>ORDER STATUS</th></tr>";

            // output data of each row

            while ($row = $result->fetch\_assoc()) {

              echo "<tr><td>" . $row["order\_id"] . "</td><td>" . $row["last\_updated"] . "</td><td>" . $row["user\_request\_place"] . "</td><td>" . $row["vehicle\_type"] . "</td><td>" . $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>

  function openForm() {

    document.getElementById("myForm").style.display = "block";

    document.getElementById("blurBox").style.display = "block";

  }

  function closeForm() {

    document.getElementById("myForm").style.display = "none";

    document.getElementById("blurBox").style.display = "none";

  }

</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;

    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>

### 4.1.5 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"/>

    <!-- Font Icon -->

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

    <!-- Main css -->

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

    <script type="text/javascript">

        //password match

        function Validate() {

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

            var confirmPassword = document.getElementById("confirm\_password").value;

            if (password != confirmPassword) {

                alert("Passwords do not match.");

                return false;

            }

            return true;

        }

        //password checkbox

        function showPassword() {

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

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

                x.type = "text";

            } else {

                x.type = "password";

            }

        }

    </script>

    <?php

    session\_start();

    if (isset($\_SESSION['name'])) {

        // Create connection

        $conn = mysqli\_connect("localhost", "root", "", "repairspot");

        // Check connection

        if (!$conn) {

            die("Connection failed" . mysqli\_connect\_error());

        }

        $name = $\_SESSION['name'];

        $sql = "SELECT name FROM mech\_details where name='".$\_SESSION['name']."'";

        $result = $conn->query($sql);

        if ($result->num\_rows > 0) {

            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');

    ?>

        <section class="signup mt-3 mb-0">

            <div class="container">

                <div class="signup-content">

                    <div class="signup-form">

                        <h2 class="form-title">Mechanic-Login</h2>

                        <form id="mech\_login" method="POST" action="./backend/mech\_login\_back.php">

                            <div class="form-group">

                                <div class="d-flex">

                                    <input type="tel" maxlength="10" name="mob\_num" id="mob\_num" placeholder="Phone Number" />

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

                                        Get OTP

                                    </button>

                                </div>

                                <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>

                </div>

            </div>

        </section>

    </div>

    <?php

    include('components/justFooter.php');

    ?>

</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>

    var otp = generateOTP();

    function sendOTP() {

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

        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 there! Here's OTP for Quick Mechanist. Please don't share the OTP - ${otp}`

                    },

                    {

                        "channel": "sms",

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

                        "to": "919944622435",

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

                    }

                ]

            });

            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!!")

                }

            };

            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>

### 4.1.6 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;

            border-radius: 0;

            border-bottom: 1px solid #999;

            font-size: inherit;

            padding-left: 25px;

        }

        .form-select:focus {

            border-bottom: 1px solid #999;

        }

        .mech\_other\_service {

            display: none;

        }

    </style>

    <script type="text/javascript">

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

        function checkvalue() {

            var service = document.getElementById('mech\_service\_type').value;

            if (service == 'stype') {

                document.getElementById("other\_service").style.display = 'none';

            }

            if (service == 'others') {

                document.getElementById("other\_service").style.display = 'block';

            } else {

                document.getElementById("other\_service").style.display = 'none';

            }

        }

        //password match

        function Validate() {

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

            var confirmPassword = document.getElementById("confirm\_password").value;

            if (password != confirmPassword) {

                alert("Passwords do not match.");

                return false;

            }

            return true;

        }

        //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');

        ?>

        <section class="signup mt-3 mb-0">

            <div class="container">

                <div class="signup-content">

                    <div class="signup-form">

                        <h2 class="form-title">Mechanic</h2>

                        <form id="mech\_registration\_form" name="my" method="POST" action="./backend/mech\_register\_back.php">

                            <div class="form-group">

                                <input type="text" name="name" id="name" placeholder="Name" />

                            </div>

                            <div class="form-group">

                                <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>

                            </div>

                            <div class="form-group mech\_other\_service" id="other\_service">

                                <input type="text" name="mech\_other\_service" id="mech\_other\_service" placeholder="If Others:" />

                            </div>

                            <div class="form-group">

                                <div class="d-flex">

                                    <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()">

                                        Get OTP

                                    </button>

                                </div>

                                <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>

                            <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">

                        <figure>

                            <img src="./assets/images/undraw\_automobile\_repair.svg" alt="sing up image">

                            <!-- <img src="./assets/images/signup-image.jpg" alt="sing up image"> -->

                        </figure>

                        <a href="https://localhost/Quick-Mechanist/user\_register.php" class="signup-image-link">Owner Register</a>

                    </div>

                </div>

            </div>

        </section>

    </div>

    <div>

        <?php

        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>

    var otp = generateOTP();

    function sendOTP() {

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

        var name = $("#name").val();

        var number = $("#mob\_num").val();

        var service\_type = $('#mech\_service\_type').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 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}`

                        }

                    ]

                });

            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!!")

                }

            };

            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\_registration\_form").submit();

        } else {

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

            $(".otpError").show();

            return false;

        }

    }

</script>

</html>

### 4.1.7 mech\_dashboard.php

<?php

session\_cache\_limiter('private, must-revalidate');

session\_cache\_expire(61200);

session\_start();

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

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

}

include('backend/radius\_calculator.php');

?>

<html>

<head>

    <title>Mechanic Dashboard</title>

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

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

    <!-- Styles and Fonts -->

    <!-- Bootstrap -->

    <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">

    <!-- Custom CSS -->

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

    <!-- Font Icon -->

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

    <!-- Main CSS -->

    <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;

            padding: 5px;

            height: 100%;

            box-sizing: border-box;

            word-wrap: break-word;

            word-break: break-all;

            text-align: center;

            pointer-events: none;

        }

        .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;

            }

        }

    </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());

                    }

                    $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='PENDING'";

                    $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>";

                                echo "<tr> <td><input class='tableInput' type='text' name='order\_id' value='" . $row["order\_id"] . "' /></td> <td>" . $row["last\_updated"] . "</td> <td><input class='tableInput' type='text' name='user\_name' value='" . $row["name"] . "' /></td> <td><textarea class='tableInput' type='text' name='user\_request\_place'>" . $row["user\_request\_place"] . "</textarea></td> <td><input class='tableInput' type='text' name='vehicle\_type' value='" . $row["vehicle\_type"] . "' /></td> <td><input class='tableInput' type='text' name='vehicle\_problem' value='" . $row["vehicle\_problem"] . "' /></td> <td><a href='" . $gmapwith . "' target=\_blank> Google Map </a></td> <td>" . $actionButton . "</td> </tr>";

                            }

                        }

                        echo "</table>";

                    } else {

                        echo "0 results";

                    }

                    $conn->close();

                }

                ?>

            </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']; ?>">

                <?php {

                    // Create connection

                    $conn = mysqli\_connect("localhost", "root", "", "repairspot");

                    // Check connection

                    if (!$conn) {

                        die("Connection failed" . mysqli\_connect\_error());

                    }

                    $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']);

                                echo "<tr> <td><input class='tableInput' type='text' name='order\_id' value='" . $row["order\_id"] . "' /></td> <td>" . $row["last\_updated"] . "</td> <td><input class='tableInput' type='text' name='user\_name' value='" . $row["name"] . "' /></td> <td><textarea class='tableInput' type='text' name='user\_request\_place'>" . $row["user\_request\_place"] . "</textarea></td> <td><input class='tableInput' type='text' name='vehicle\_type' value='" . $row["vehicle\_type"] . "' /></td> <td><input class='tableInput' type='text' name='vehicle\_problem' value='" . $row["vehicle\_problem"] . "' /></td> <td>" . "$distance[0] $distance[1]" . "<br><a href='" . $gmapwith . "' target=\_blank> Google Map </a></td> <td>" . $actionButton . "</td> </tr>";

                            }

                        }

                        echo "</table>";

                    } else {

                        echo "0 results";

                    }

                    $conn->close();

                }

                ?>

            </form>

            <div class="content">

            </div>

        </div>

    </div>

    <?php

    include('components/justFooter.php');

    ?>

</body>

</html>

### 4.1.8 mech\_mobile.php

<?php

 $conn=mysqli\_connect("localhost","root","","repairspot");

  if (!$conn)

  {

   die(" connection failed".mysqli\_connect\_error());

   echo "database connection failed";

  }

   //if submit triggers..

    else

  {

      $mech\_email=$\_POST['name'];

      $query="SELECT mob\_num from mech\_details where email='$mech\_email'";

      $result=mysqli\_query($conn,$query);

      if($result->num\_rows > 0)

      {

           while ($row=$result->fetch\_assoc())

           {

             echo $row['mob\_num'];

           }

      }

      //if mob num not present

      else

     {

      echo " nil";

     }

     $conn->close();

   }

?>

### 4.1.9 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>

                        <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>

### 4.1.10 repairspot.sql

-- phpMyAdmin SQL Dump

-- version 4.8.5

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Apr 06, 2023 at 05:51 PM

-- Server version: 10.1.38-MariaDB

-- PHP Version: 5.6.40

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `repairspot`

--

-- --------------------------------------------------------

--

-- Table structure for table `mech\_approved`

--

CREATE TABLE `mech\_approved` (

  `order\_id` int(11) NOT NULL,

  `user\_name` varchar(50) NOT NULL,

  `approved\_mech\_name` char(50) NOT NULL,

  `vehicle\_type` varchar(50) NOT NULL,

  `vehicle\_problem` varchar(100) NOT NULL,

  `location` varchar(200) NOT NULL,

  `order\_status` varchar(30) NOT NULL,

  `approved\_datetime` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

  `mech\_mobile\_num` varchar(15) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- --------------------------------------------------------

--

-- Table structure for table `mech\_details`

--

CREATE TABLE `mech\_details` (

  `latitude` varchar(20) NOT NULL,

  `longitude` varchar(20) NOT NULL,

  `otp` varchar(6) NOT NULL,

  `service\_type` varchar(20) NOT NULL,

  `service\_type\_others` varchar(20) NOT NULL,

  `name` varchar(25) NOT NULL,

  `mob\_num` varchar(13) NOT NULL,

  `mech\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `mech\_details`

--

INSERT INTO `mech\_details` (`latitude`, `longitude`, `otp`, `service\_type`, `service\_type\_others`, `name`, `mob\_num`, `mech\_id`) VALUES

('13.1429', '79.9075', '', 'car\_mechanic', '', 'JosephMechanic', '9444009634', 14);

-- --------------------------------------------------------

--

-- Table structure for table `user\_booking\_request`

--

CREATE TABLE `user\_booking\_request` (

  `name` varchar(50) NOT NULL,

  `user\_request\_place` varchar(200) DEFAULT NULL,

  `vehicle\_type` varchar(20) DEFAULT NULL,

  `vehicle\_problem` varchar(200) DEFAULT NULL,

  `request\_status` varchar(20) DEFAULT NULL,

  `approved\_mech\_name` varchar(55) NOT NULL,

  `order\_id` int(20) NOT NULL,

  `last\_updated` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

  `latitude` varchar(11) NOT NULL,

  `longitude` varchar(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `user\_booking\_request`

--

INSERT INTO `user\_booking\_request` (`name`, `user\_request\_place`, `vehicle\_type`, `vehicle\_problem`, `request\_status`, `approved\_mech\_name`, `order\_id`, `last\_updated`, `latitude`, `longitude`) VALUES

('Joseph', 'Darling Showroom Thiruvallur', 'Car i20 CDI', 'Puncher', 'PENDING', 'JosephMechanic', 1044, '2023-04-06 12:06:57', '13.1430891', '79.9084724');

-- --------------------------------------------------------

--

-- Table structure for table `user\_details`

--

CREATE TABLE `user\_details` (

  `name` varchar(25) NOT NULL,

  `otp` int(11) NOT NULL,

  `mob\_num` varchar(13) NOT NULL,

  `user\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `user\_details`

--

INSERT INTO `user\_details` (`name`, `otp`, `mob\_num`, `user\_id`) VALUES

('Joseph', 6428, '9944622435', 1);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `mech\_approved`

--

ALTER TABLE `mech\_approved`

  ADD PRIMARY KEY (`order\_id`);

--

-- Indexes for table `mech\_details`

--

ALTER TABLE `mech\_details`

  ADD PRIMARY KEY (`mech\_id`);

--

-- Indexes for table `user\_booking\_request`

--

ALTER TABLE `user\_booking\_request`

  ADD PRIMARY KEY (`order\_id`);

--

-- Indexes for table `user\_details`

--

ALTER TABLE `user\_details`

  ADD PRIMARY KEY (`user\_id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `mech\_details`

--

ALTER TABLE `mech\_details`

  MODIFY `mech\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;

--

-- AUTO\_INCREMENT for table `user\_booking\_request`

--

ALTER TABLE `user\_booking\_request`

  MODIFY `order\_id` int(20) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=1045;

--

-- AUTO\_INCREMENT for table `user\_details`

--

ALTER TABLE `user\_details`

  MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

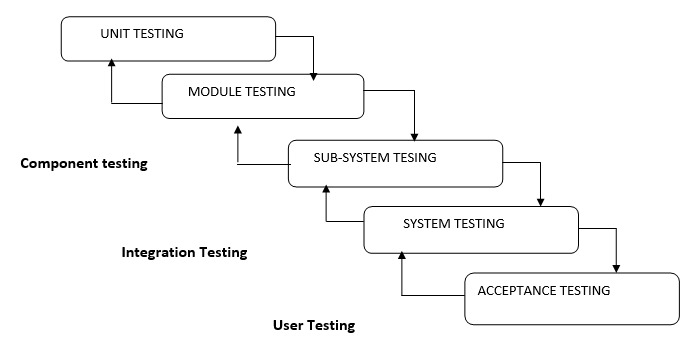
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

## 4.2 STEMTESTING

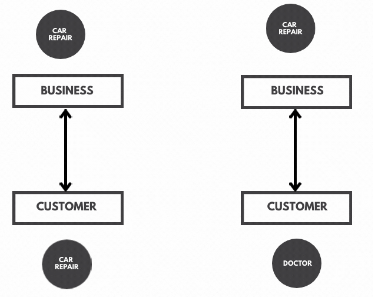
The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress is done by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



### 4.2.1UNITTESTING

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.

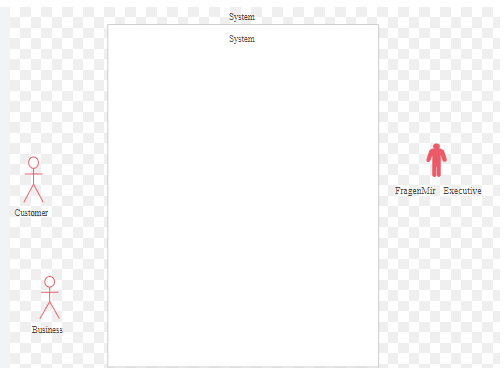


### 4.2.2 WHITE BOXTESTING

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.



### 4.2.3 BASIC PATHTESTING

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.

### 4.2.4 CONDITIONALTESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

### 4.2.5 DATA FLOWTESTING

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 variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

### 4.2.6 LOOPTESTING

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

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loop test the innermost loop first and then work out wards.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. |

### 4.2.7User Acceptance Testing

User Acceptance Testing (UAT) is a critical phase in the development process of any software project, including Smart Automotive Mechanic Finder using Google map navigator and Clickatell project. The purpose of UAT is to ensure that the developed software meets the requirements of the end-users and is ready for deployment. In UAT, actual end-users of the system test the software in a simulated production environment to validate that it works as expected and meets their needs.

The following are some key steps involved in UAT for Smart Automotive Mechanic Finder using Google map navigator and Clickatell project:

Define acceptance criteria: The acceptance criteria should be established before the UAT starts. The criteria should include a list of features and functionalities that need to be tested, along with the expected outcomes and results.

Select test users: Identify a group of test users who represent the actual end-users of the system. This group should be diverse, including individuals with different levels of technical knowledge and experience.

Develop test scenarios: Develop test scenarios that will validate that the software meets the acceptance criteria. Test scenarios should simulate real-world scenarios that end-users are likely to encounter.

Execute test scenarios: The test users execute the test scenarios to validate that the software meets the acceptance criteria. The testers should document any issues or bugs they encounter during testing.

Fix identified issues: The development team should address the issues or bugs identified during the UAT and provide a fix for them.

Re-test: After the development team has addressed the issues, the UAT team should retest the software to validate that the fixes work as expected.

Acceptance: If the software meets the acceptance criteria, the UAT team should provide its acceptance, and the software can be deployed to production.

UAT is a critical step in the development process of Smart Automotive Mechanic Finder using Google map navigator and Clickatell project as it ensures that the software is ready for deployment and meets the requirements of the end-users. It is essential to conduct UAT in a simulated production environment to identify any issues or bugs and fix them before the software is deployed to the actual production environment.

# 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.

The proposed system has several advantages over the existing system, including ease of use, real-time tracking, and communication between users and mechanics. The system was developed using various technologies such as HTML, CSS, JavaScript, PHP, MySQL, and Google Maps API.

User acceptance testing was conducted to ensure the usability and effectiveness of the system. The feedback from the test users was positive, indicating that the system is intuitive and helpful in finding nearby mechanics.

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

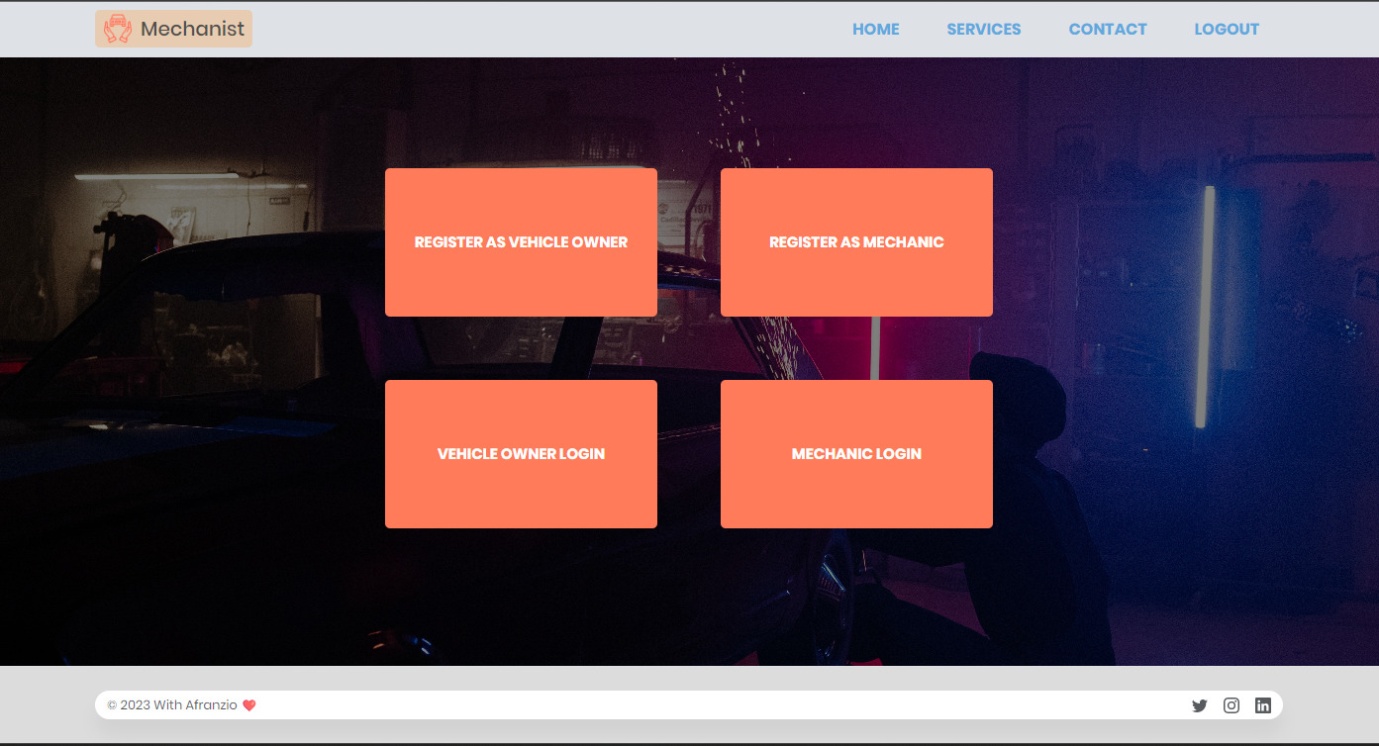
Social media integration: Integrating social media platforms could enable users to share their experiences with the app and also allow the app to provide more personalized recommendations.

Mobile app: Developing a mobile app version of the platform could enable users to access the service on the go and receive push notifications for updates.

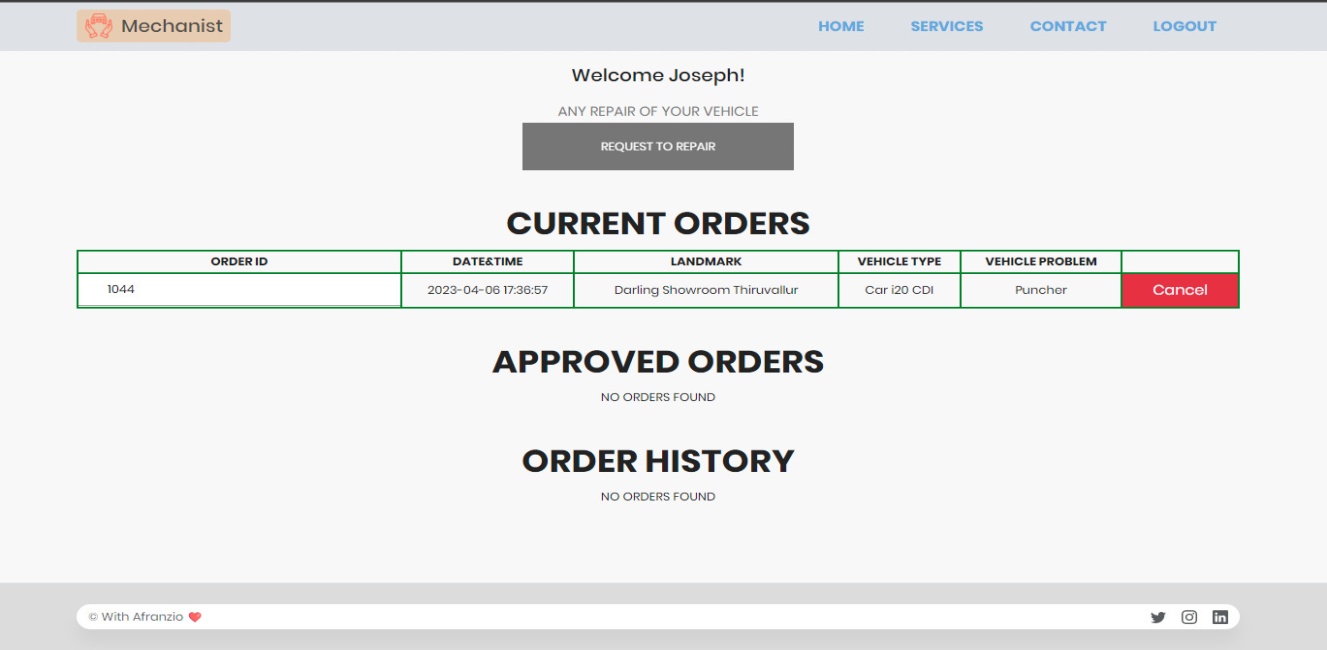
Additional services: Adding more services such as tire replacement, battery replacement, and oil change would expand the platform's offering and make it a one-stop-shop for automotive services.

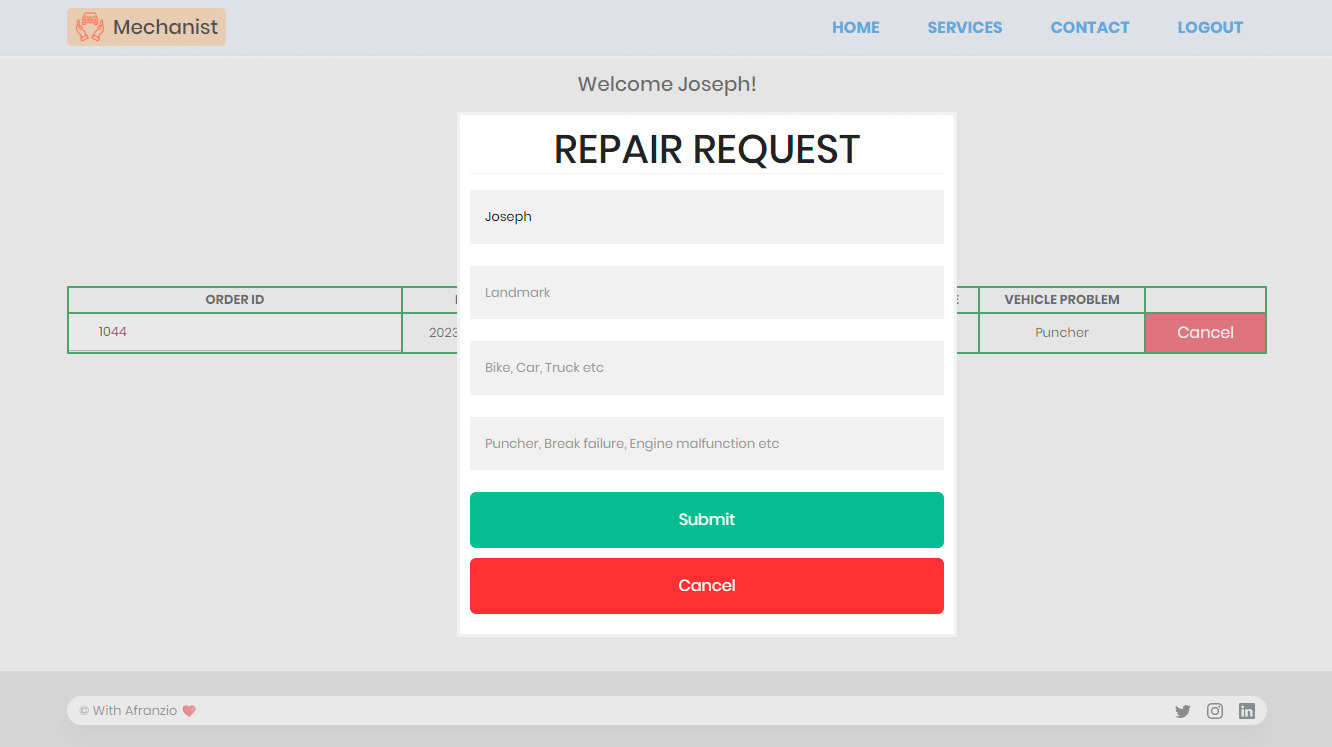
Expansion to other regions: Expanding the platform's availability to other regions and countries could increase its user base and provide more value to users in those areas.

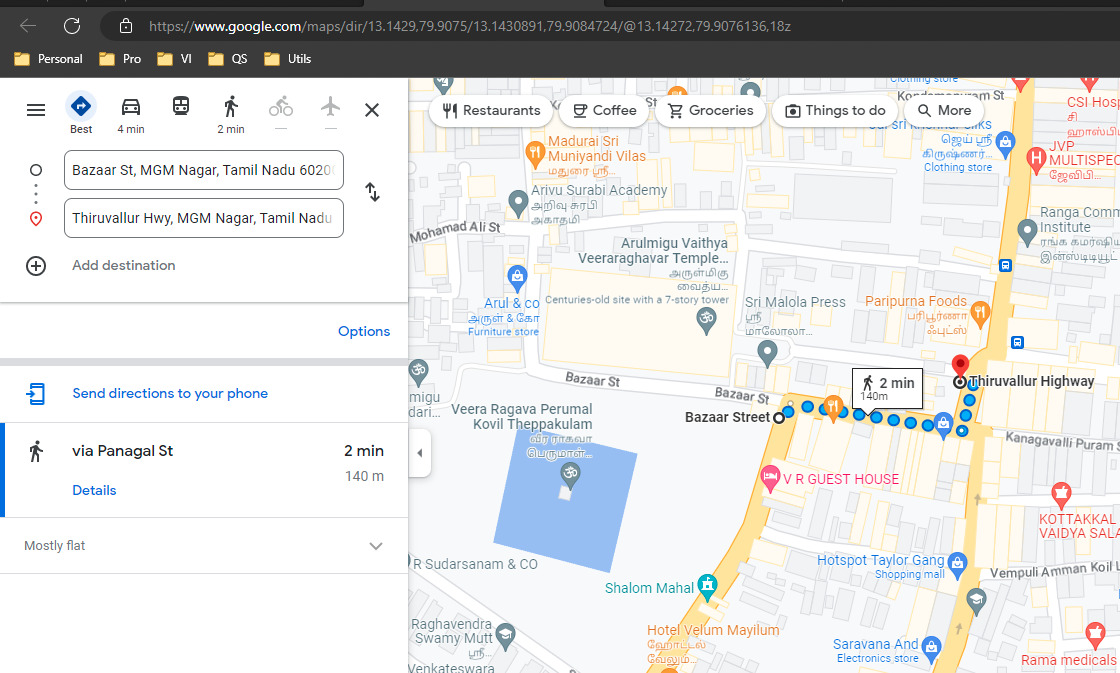
These are just a few examples of potential future enhancements for the Smart Automotive Mechanic Finder using Google Maps and Clickatell project. The possibilities for improvement are endless, and the app could evolve over time to meet the changing needs and demands of its users.APPENDIX



# WhatsApp Image 2023-04-21 at 10.44.55 AM (1).jpeg WhatsApp Image 2023-04-21 at 10.44.56 AM.jpeg WhatsApp Image 2023-04-21 at 10.44.56 AM (1).jpeg WhatsApp Image 2023-04-21 at 10.44.58 AM.jpeg WhatsApp Image 2023-04-21 at 10.44.59 AM (1).jpeg







# REFERENCES

* Google Maps Platform: [https://developers.google.com/maps] (<https://developers.google.com/maps>)
* Clickatell API documentation: [https://www.clickatell.com/developers/api-documentation/] (<https://www.clickatell.com/developers/api-documentation/>)
* "An Intelligent Mechanic Finder System Based on Android Platform" by Zhang Xinxin, Chen Weitao, and Jiang Qian: [<https://www.hindawi.com/journals/jam/2014/579787/>]
* "A Service Recommendation Mechanism for Mobile Automotive Services" by Wei Liu, Hao Su, and Huiqun Yu: [<https://www.sciencedirect.com/science/article/pii/S1877050917320324>]
* "Design and Implementation of Smart Automotive Service Platform Based on Internet of Things“ by Jihua Zhao, Xiaofeng Zhu, and Yan Liu: ([https:/www.sciencedirect.com/science/article/pii/S2351978916301612](https://www.sciencedirect.com/science/article/pii/S2351978916301612%5d))