

DETAILED REPORT

FIRST REVIEW

MID ROAD VEHICLE BREAKDOWN ASSISTANCE

GUIDE SIGNATURE

SUBMITTED BY
NAME: BALASANKAR.S

ROLL NO: 2019202005

MCA (REGULAR)-3rd YEAR

MID ROAD VEHICLE BREAKDOWN ASSISTANCE

A PROJECT REPORT

Submitted by

BALASANKAR S – 2019202005

A report of the project submitted to the Faculty of

INFORMATION SCIENCE AND TECHNOLOGY

in partial fulfillment

for the award of the degree

of

MASTER OF COMPUTER APPLICATION



DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY,

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY CHENNAI 600 025

MAY, 2022

ANNA UNIVERSITY
CHENNAI - 600 025
BONAFIDE CERTIFICATE

Certified that this project report titled MID ROAD VEHICLE BREAKDOWN ASSISTANCE is the bonafide work of BALASANKAR S who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation based on which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: CHENNAI
DATE: 07-05-2022

PROJECT GUIDE
DR. E. UMA
ASSISTANT PROFESSOR ([SL.GR](#))
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

DR.S.SRIDHAR
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

MID ROAD VEHICLE BREAKDOWN ASSISTANCE is going to be a good solution for the people who seek help in remote locations with issues in their vehicles. In this application user and helper services are registered by giving their details. User details include their mail, name, phone number, and location of the user. Helper service (Mechanic, towing, fuel) details include their mail, name, phone number, location, type of vehicle mechanic can repair, availability, and ID to verify. This information is for any other purposes used only by this application. Admin verify and approve helper service. When users get any issue, they can get help from the helper chat if the issue is minor and if they need service like a mechanic, towing, or they need fuel then they select the required service. Helper services are listed within their area. Users connect with a helper service, then send their location and tell about their issue in their vehicle. Helper service came to the user's location and do the required service. Pay for the service and give feedback about the service. For admin, feedback is helpful to know about the helper service. If the service is not to the level then the admin has the right to reject the helper service from the application.

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	MOTIVATION AND OBJECTIVE	1
1.2	IMPLEMENTATION PLATFORM/Framework	1
2	RELATED WORK	2
2.1	A CAR BREAKDOWN SERVICE STATION LOCATOR SYSTEM	2
2.2	A SURVEY ON SQL INJECTION ATTACKS, THEIR DETECTION AND PREVENTION TECHNIQUES	2
3	SYSTEM DESIGN	3
3.1	OVERALL ARCHITECTURE	3
3.2	MODULES DESCRIPTION	4
3.2.1	REGISTER AND LOGIN	4
3.2.2	UPDATE PROFILE	4
3.2.3	CHATBOT	4
3.2.4	CONNECT WITH HELPER SERVICE	4
3.2.5	USER CHAT WITH HELPER SERVICE	5
3.2.6	USER DETAILS	5
3.2.7	HELPER SERVICE	5
3.2.8	APPROVE HELPER SERVICE	5
3.2.9	PAYMENT AND FEEDBACK	5
3.3	DETAILED DESIGN WITH FLOW CHART	6
3.3.1	USER	6
3.3.2	HELPER SERVICE	6
3.3.3	ADMIN	7
4	MODULE-BASED OUTPUT SCREENSHOTS	8
	REFERENCES	12

CHAPTER 1

INTRODUCTION

The purpose of MID ROAD VEHICLE BREAKDOWN ASSISTANCE is to ease the people problem who seek help for the issue with the vehicle on the road. The scope of MID ROAD VEHICLE BREAKDOWN ASSISTANCE is to create a convenient and easy-to-use application for users seeking help on the road. We hope to provide a comfortable user experience.

1.1 MOTIVATION AND OBJECTIVE:

The main objective of MID ROAD VEHICLE BREAKDOWN ASSISTANCE is to get service easy when the user has an issue with their vehicle while traveling. MID ROAD VEHICLE BREAKDOWN ASSISTANCE helps helper services to get a business. This project is made as user-friendly as possible so that anyone can use it with little knowledge of system computers. Only approved Helper Services can give service to the user. To ensure the service is good.

1.2 IMPLEMENTATION PLATFORM / FRAMEWORK:

This application is going to develop using HTML, CSS as frontend which helps to style a good user interface and PHP, MYSQL as backend which helps to get data to form web page to the database and vice versa. Visual Studio Code is used as a tool for coding HTML, CSS and PHP. XAMPP server helps to run an MYSQL database. Simple Mail Transfer Protocol (SMTP) is used for sending mail. Google map API is used for location purposes with help of API key and geolocation. The location of the user can be found and the same is used to list the helper service near the user area. The most common attacks, SQL Injection and Cross-Site Scripting (XSS) are prevented in this application

XAMPP: XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl and it helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server.

VS CODE: Visual Studio Code is a good framework to code Html, CSS, JavaScript and PHP

CHAPTER 2

LITERATURE REVIEW

2.1 A car breakdown service station locator system

Khoo Jin Sheng et al (2016) proposed that the system will use the driver's current location to determine the nearest Car Repair Service Providers (CRSP) available and display a list of CRSPs nearby for the driver to choose from. To perform the search for the nearest CRSP, Google Places API for mobile will be used to connect people to places of interest with the power of location awareness on Android. As for the cost of service, the price charged by each CRSP is impossible to be displayed as the root cause of car breakdown must be determined before the cost can be calculated. The scope of this system will focus on searching the nearest CRSP for the drivers, providing help to people who do not possess any mechanic's number in hand. The business deal is between the CRSP and the driver which is out of the system's control.

2.2 A Survey on SQL Injection attacks, their Detection and Prevention Techniques

V. Nithya et al (2013) proposed that most web applications today use a multi-tier design, usually with three tiers: a presentation, a processing and a data tier. The presentation tier is the HTTP web interface, the application tier implements the software functionality and the data tier keeps data structured and answers requests from the application tier. Meanwhile, large companies developing SQL-based database management systems rely heavily on hardware to ensure the desired performance. SQL injection is a type of attack in which the attacker adds Structured Query Language code to the input box of a web form to gain access or make changes to data. SQL injection vulnerability allows an attacker to flow commands directly to web applications underlying database and destroy functionality or confidentiality

CHAPTER 3

SYSTEM DESIGN

This chapter consists of the system design of the project with the overall architecture and the description of the modules used in the project

3.1 OVERALL ARCHITECTURE:

The overall system architecture of the proposed system is shown in Figure 3.1. Helper Service details are stored in the helper service database. Admin verifies and accepts the helper services. Then the helper service details are ready to show to users. Users can get help from the Chabot, in Chabot users send their queries bot will send the actionable data to the user as a response. The other method to help from helper service is for, User to get the nearby helper service with the help of the user's current location from the helper service database. Users can select the required helper service. Once the user connects with the helper service, the User can send the location to the helper service.

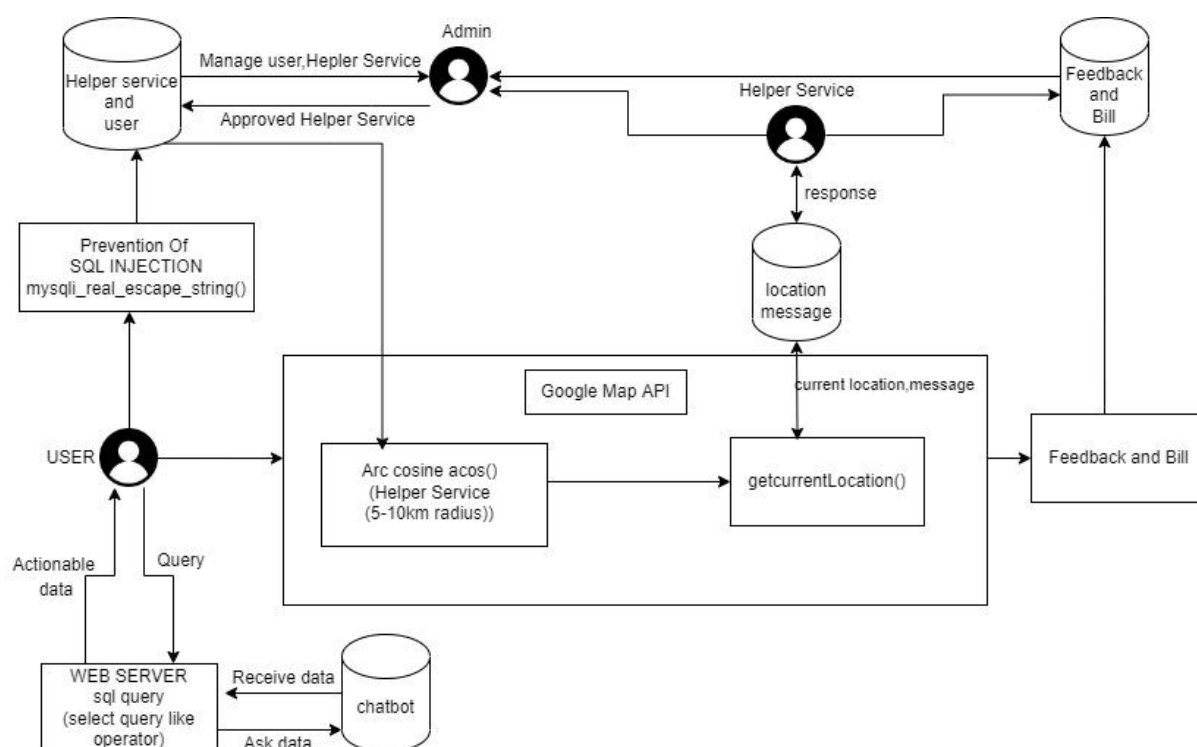


Figure 3.1: Architecture Diagram

Helper service came to the user's location and finish their services. Users need to pay for the service and give feedback on the service. These details are stored in the Feedback and Bill

database. Further, the bill will receive by the user through the mail, sent by the helper service. These details are monitored by the admin.

3.2 MODULES DESCRIPTION:

3.2.1 Register and Login:

The user should register in this application. The details required to register are name, email, phone number and password. Then the user can log in to the application by providing their email and password

Helper service should register in this application. The details required to register are name, email, phone number, password, availability, location, type of vehicle and Id Proof. Then Helper service can log in to the application by providing their email and password. Check approved by admin or not. After approval, Helper service can help and provide service to the user

3.2.2 Update Profile:

Users can able to view their details, their name, email and phone number. Users can able to update or change their details in this application.

Helper service can be able to view their details, like their name, email, phone number, availability, location, type of vehicle and id proof. Helper service can able to update or change their details in this application.

3.2.3 Chat Bot:

In the char bot user send their queries and get back the solution from the bot. By using SQL select query which matches the query provided by the user from the database, the bot sends the response as actionable data

3.2.4 Connect with Helper Service:

User selects their problem, the problems listed are Mechanic issue, Towing, Fuel and nearby helper services are listed then connect with the helper service.

3.2.5 User chat with Helper Service and vice versa:

The user sends a request, helper service can accept or reject it. If the issue is solved by the helper service, they can accept else they can reject it.

After accepting, the user chats with them by telling the issue and sends the current location using Google Maps API to the helper service.

3.2.6 User Details:

The user's details are listed to the admin. Admin view all about the user details and manages those details.

3.2.7 Helper Service:

Helper service details are listed to the admin. Admin view all about the helper service details and manages those details.

3.2.8 Approve Helper Service:

Admin verifies the details provided by the helper service and approves the helper service if the details are valid.

3.2.9 Payment and Feedback:

After the service is finished users pay for the service and give feedback about the service. Feedback are listed to the helper service. Good feedback motivates the helper service to give service in much good way, Bad feedback helps helper service to improve their services in comparison to previous services

By getting feedback from the users, admin manages the helper service by continuing or rejecting the helper service. Helper services who get good feedback can continue in the MID ROAD VEHICLE BREAKDOWN ASSISTANCE. Helper services get bad or worst feedback admin has the right to reject the helper service. That helper service can no longer continue in the MID ROAD VEHICLE BREAKDOWN ASSISTANCE.

3.3 DETAILED DESIGN WITH FLOW CHART:

3.3.1 USER:

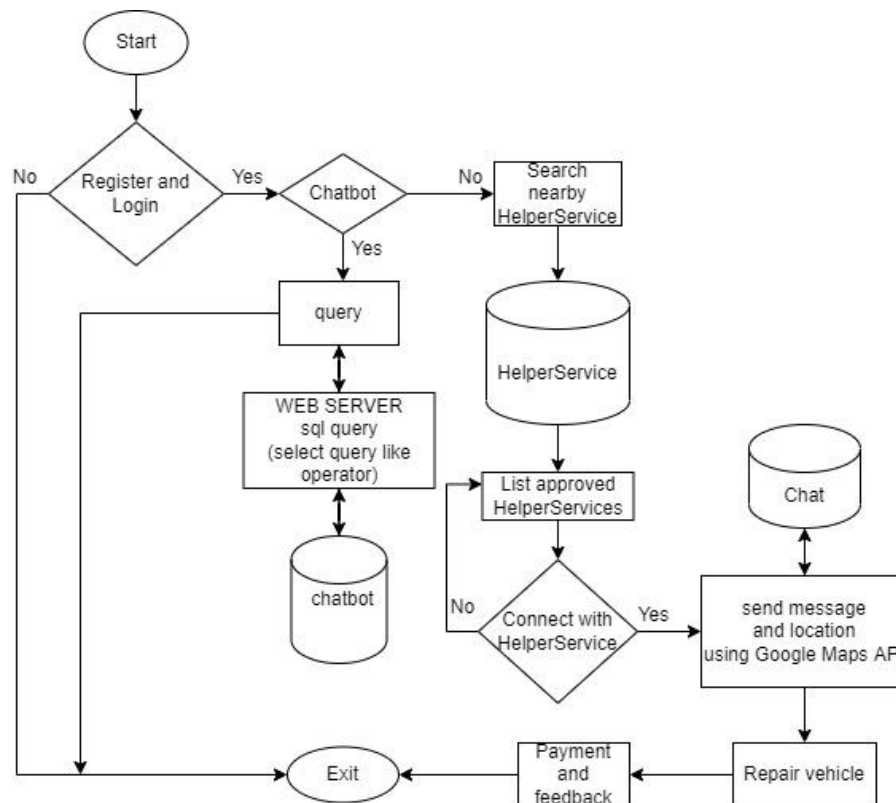


Figure 3.2: User Flow Chart

The flow chart for the user is shown in Figure 3.2. User registration and login. If a user wants to use Chabot, the user has to send the query, Then Chabot responds accordingly using keywords from the user query. If the user wants to connect to the Helper Service, Search for nearby helper service by giving the type of helper service and their current location. The list of helper services available is shown. Select the helper service needed and connect with them. In chat, the user has to send their current location to a helper service so they reach your place and do the service. Finally, pay for the service and provide feedback to the helper service.

3.3.2 HELPER SERVICE:

The flow chart for the helper service is shown in Figure 3.3. Helper service register and log in. Once approved by the admin. The helper service will receive a request from the user. After connecting with the user. In the chat, users send their problems in the form of messages and their current location using Google Maps API. The helper service goes to the

user's location and does the required service. Finally, feedback and payment for the service are collected from the user.

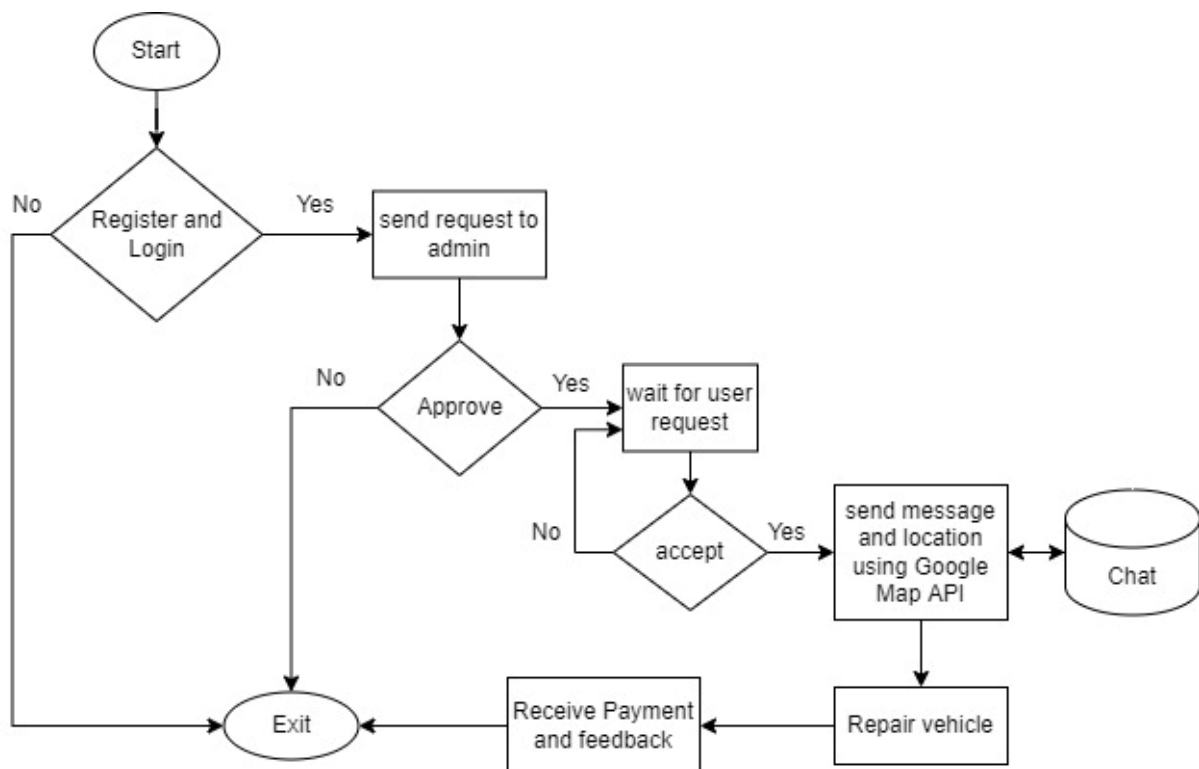


Figure 3.3: Helper Service Flow Chart

3.3.3 ADMIN:

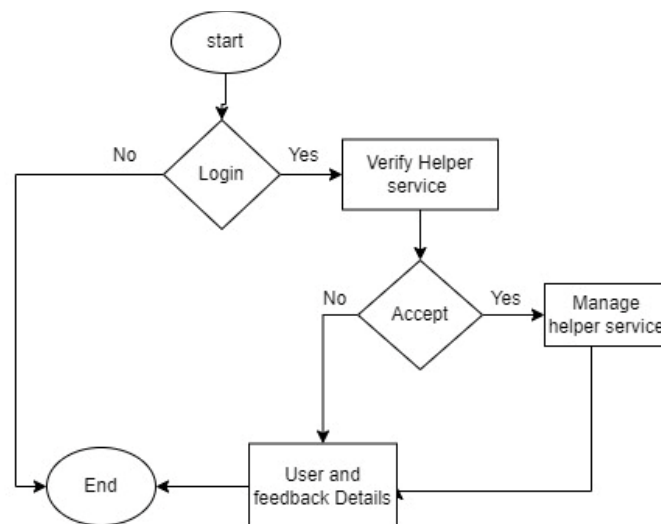


Figure 3.4: Admin Flow Chart

The flow chart for the helper service is shown in Figure 3.4. Admin login and accepts the helper service by verifying the helper service details provided which is by them. Then admin manages user helper service and feedback details.

CHAPTER 4

MODULE-BASED OUTPUT SCREENSHOTS

Home Page:

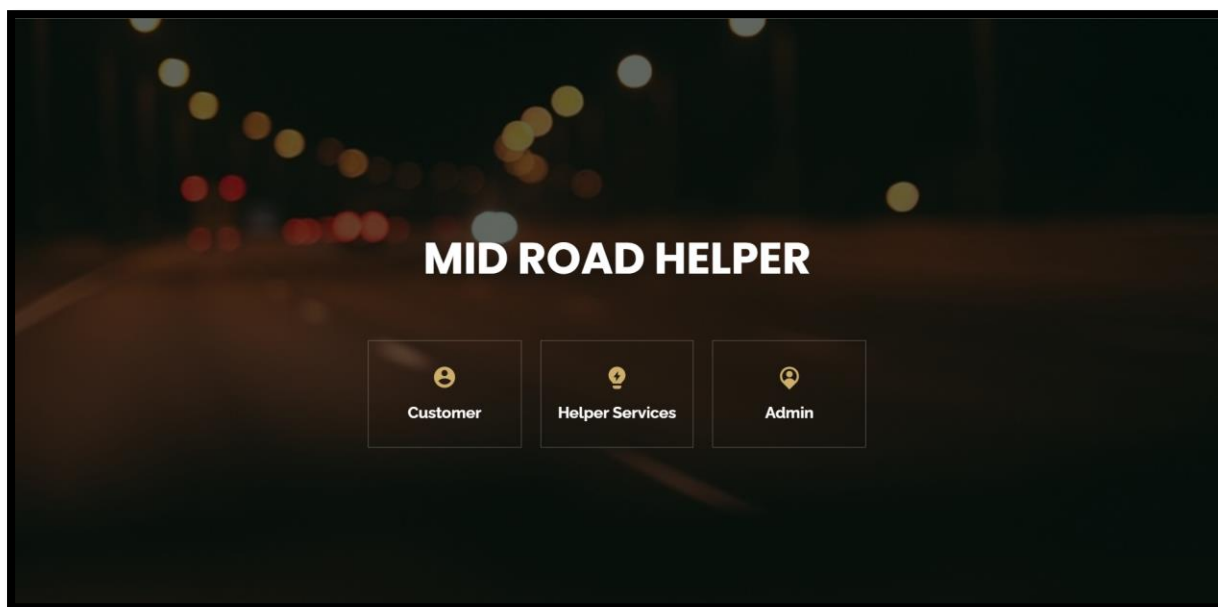


Figure 4.1: Home Screen

There are three types of users: Customers, Helper Service and Admin in this application. The link to navigate to their respective screen is in Figure 4.1 home screen.

Login Page:

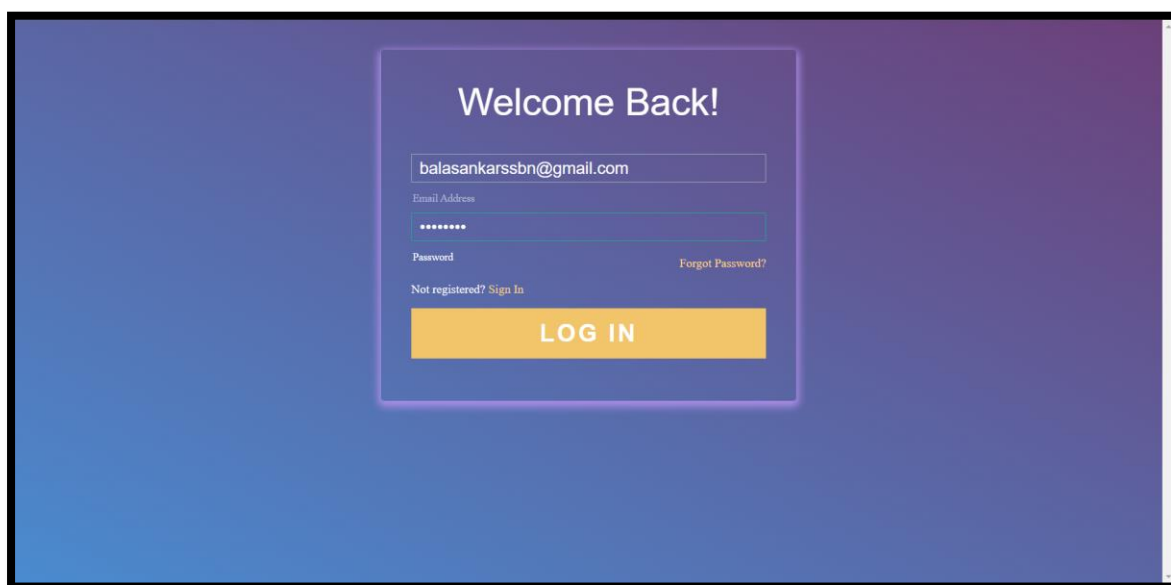


Figure 4.2: User Login Screen

The login screen for the user is in Figure 4.2. Here, users give their mail and their password to log in by clicking the Login button and to register, the user clicks the sign in.

User Registration:

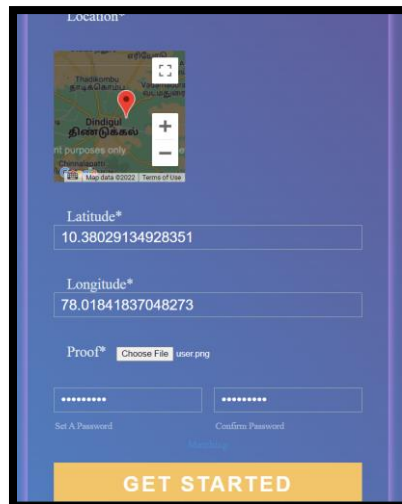
The image shows a 'Sign Up' form on a purple gradient background. The form is titled 'Sign Up' and contains the following fields: 'Your Name' (filled with 'Balasankar'), 'Email Address' (filled with 'balasankarssbn@gmail.com'), 'Number' (filled with '9909874573'), 'Set A Password' (filled with '*****'), and 'Confirm Password' (filled with '*****'). Below the password fields, it says 'Matching'. At the bottom of the form is a yellow 'GET STARTED' button and a link 'Already registered? Log In'.

Figure 4.3: User Registration Screen

The registration screen for the user is in Figure 4.3. Users should register with their details like Name, Email, number and password and click the get started button to register.

Helper Service Registration:

The image shows a 'Sign Up' form for a helper service on a purple gradient background. The form is titled 'Sign Up' and contains the following fields: 'Your Name' (filled with 'Kannan'), 'Business Name' (filled with 'Kanna Mechanics'), 'Email Address' (filled with 'kannamech@gmail.com'), 'Number' (filled with '8900332478'), 'Service*' (with radio buttons for 'Mechanic', 'Towing', and 'Fuel', where 'Mechanic' is selected), and 'Vehicle Type' (filled with 'Car').



Location*

Latitude*

Longitude*

Proof* [Choose File](#) user.png

Set A Password

Confirm Password

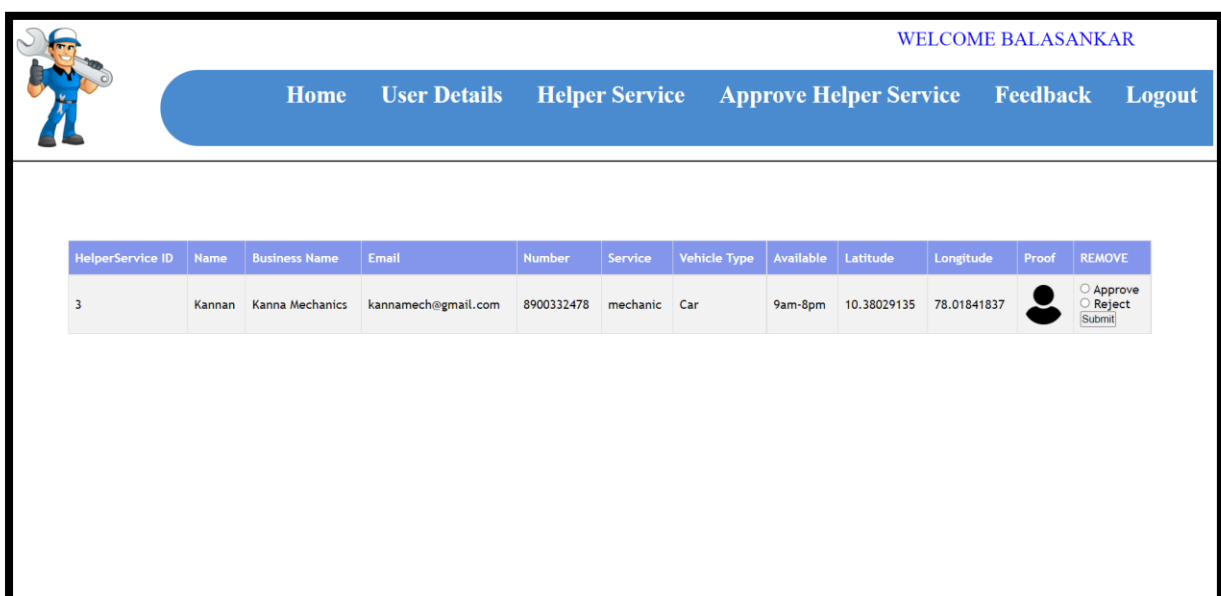
[Back](#)

GET STARTED

Figure 4.4: Helper Service SignUp Screen

The helper service registration screen is shown in Figure 4.4. Helper service should register with the details like name, business name, mail address, number, service, vehicle type, location, proof and password and click the get started button to register.

Admin Approve Helper Service page:



WELCOME BALASANKAR

Home User Details Helper Service Approve Helper Service Feedback Logout


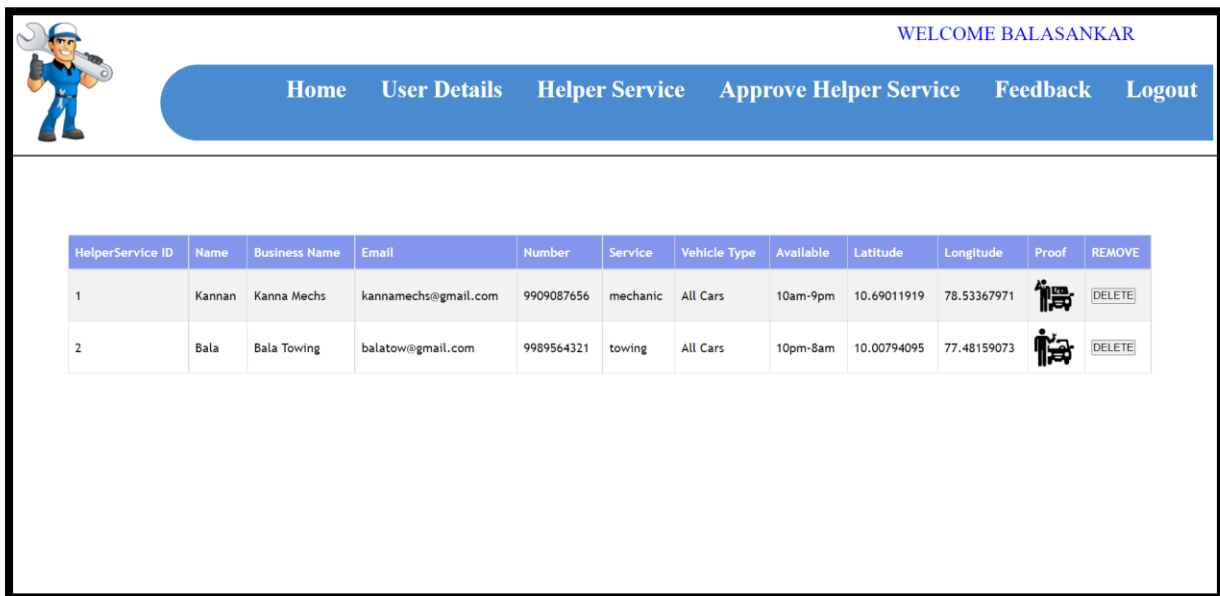
HelperService ID	Name	Business Name	Email	Number	Service	Vehicle Type	Available	Latitude	Longitude	Proof	REMOVE
3	Kannan	Kanna Mechanics	kannamech@gmail.com	8900332478	mechanic	Car	9am-8pm	10.38029135	78.01841837		<input type="radio"/> Approve <input type="radio"/> Reject <input type="button" value="Submit"/>

Figure 4.5: Approves Helper Service Screen

Admin verifies helper service details and approves or rejects helper service. Checking approve or reject radio button and clicking submit is shown in Figure 4.5.

Admin Manage Helper Service Page:



The screenshot shows the 'Admin Manage Helper Service Page'. At the top, there is a welcome message 'WELCOME BALASANKAR' and a navigation bar with links: Home, User Details, Helper Service, Approve Helper Service, Feedback, and Logout. Below the navigation bar is a table listing helper services. The table has columns for HelperService ID, Name, Business Name, Email, Number, Service, Vehicle Type, Available, Latitude, Longitude, Proof, and REMOVE. There are two entries in the table.



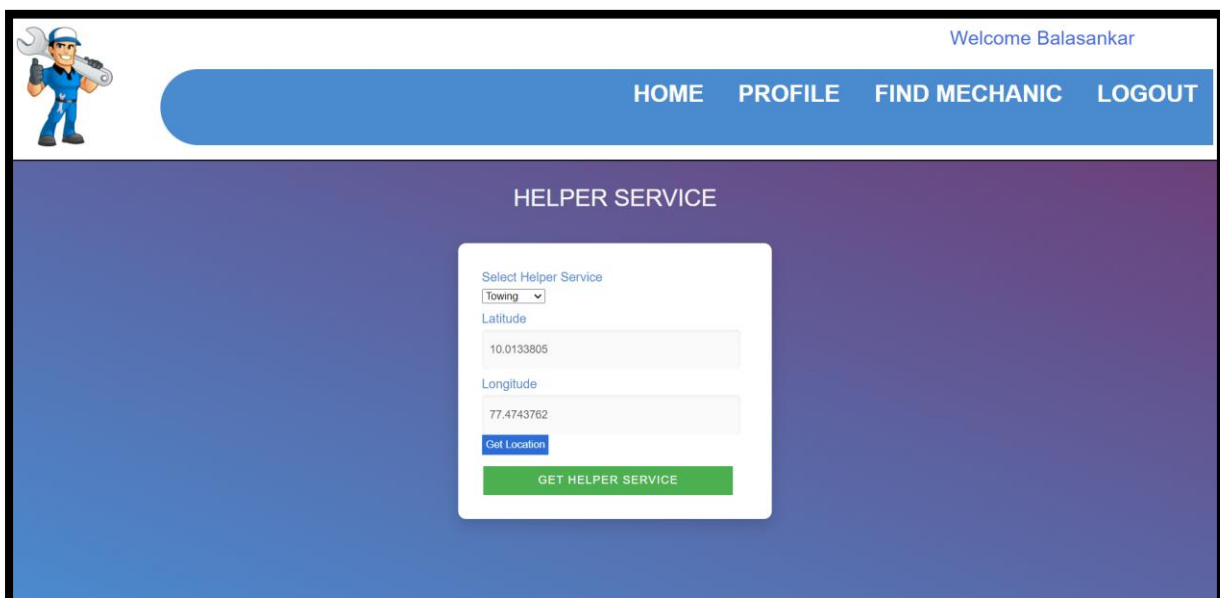
HelperService ID	Name	Business Name	Email	Number	Service	Vehicle Type	Available	Latitude	Longitude	Proof	REMOVE
1	Kannan	Kanna Mechs	kannamechs@gmail.com	9909087656	mechanic	All Cars	10am-9pm	10.69011919	78.53367971		DELETE
2	Bala	Bala Towing	balatow@gmail.com	9989564321	towing	All Cars	10pm-8am	10.00794095	77.48159073		DELETE

Figure 4.6: Helper Service Details Screen

Admin manages helper service by maintaining their details and admin can authority to reject helper service in the application by clicking delete button is shown in Figure 4.6.

Find Helper Service:



The screenshot shows the 'Find Helper Service' screen. At the top, there is a welcome message 'Welcome Balasankar' and a navigation bar with links: HOME, PROFILE, FIND MECHANIC, and LOGOUT. Below the navigation bar is a form titled 'HELPER SERVICE'. The form has a dropdown menu for 'Select Helper Service' with 'Towing' selected. Below the dropdown are input fields for 'Latitude' (10.0133805) and 'Longitude' (77.4743762). There is a 'Get Location' button and a 'GET HELPER SERVICE' button.

Figure 4.7: Find Helper Service Screen

Users search for helper services by selecting the type of helper service they need and the user's current location, then click the get helper service button as shown in Figure 4.7.

Connect with Mechanic:



Search..

HelperService ID	Name	Business Name	Email	Number	Service	Vehicle Type	Available	Latitude	Longitude	SELECT
2	Bala	Bala Towing	balatow@gmail.com	9989564321	towing	All Cars	10pm-8am	10.00794095	77.48159073	<input type="button" value="select"/>

Figure 4.8: Helper Service Select Screen

In Figure 4.8, the list of helper service user needs, with available helper services near the user's location.

REFERENCES:

- [1].khoo jin sheng, ahmad suhaimi baharudin, kamal karkonasasi (2016). "A car breakdown service station locator system". journal of applied engineering research ISSN 0973-4562 volume 11, number 22, pp 11037-11040
- [2].V. Nithya, R. Regan and J. Vijayaraghavan, "A survey of SQL injection attacks, their Detection and Prevention techniques," International Journal of Engineering and Computer Science(IJECS), vol. 2, no. 4, April 2013.
- [3].PHP documentation Available from:
<https://php-legacy-docs.zend.com/manual/php4/en/index>
- [4].Mail Transfer Using PHP Documentation Available from:
<https://www.php.net/manual/en/function.mail.php>
- [5].Google Maps API Tutorial Available from:
<https://developers.google.com/maps/documentation/javascript/overview>
- [6].Database Connection Documentation Available from:
<https://www.javatpoint.com/creating-mysql-database-with-xampp>