BLUE-COLLAR SERVICES

LIST OF CONTENTS

S.NO. **CHAPTER** **PAGE NO.**

LIST OF FIGURES

ABSTRACT

Blue-collar jobs require manual labour and their services can be hired by anyone. The purpose of the project is to develop an IRS (Information Retrieval System) for all manual services required by the common man in his day to day life. Our project provides self-employment to the workers and renders services to the user. Working on this project gives us a better experience that can be used in developing real time projects in coming future. Various phases of this project involve sample databases with several tables having information about electricians, plumbers, gardeners, masons, painters, carpenters, tailors and garage services. We will be developing a standard web page that is linked to the databases to retrieve authenticated worker information which can be viewed by the users. Users can also give feedback of workers based on their performance. Based on the feedback, admin can either revoke or retain the services of a worker. This makes the worker render a committed and reliable service.

CHAPTER-1

INTRODUCTION

Services of electricians, plumbers, gardeners, masons, painters, carpenters, tailors, garage services are all called Blue-collar services. The manual services of these workers are required by almost everyone at some time or the other. Various phases of this project involve sample databases with tables having information about workers in different categories, and their services can be utilized by any user who has registered in our website.

1.1. PROBLEM STATEMENT

In every household, there is perennial need for something or the other to be repaired or maintained. Most of the people now-a-days face a common problem of finding labour to repair such problems. We need prompt solutions to our problems and information which is reliable. Hence a portal where authenticated worker information is available is much sought after.

1.2. SCOPE

Our project is an online website which can be accessed by anyone. Users must register themselves to get the services of their interest done. Users have access to all the categories of workers and their information. Users make a note of this information and directly contact the workers according to their requirement. Workers can get employed through our website. Workers make a request to the Admin to include their information in the website. The Admin manually verifies their data and includes their services. He also maintains the user and worker databases. Worker Feedback is an important aspect of our website. Based on the feedback given by the users, the admin can either revoke or retain the services of a worker. This makes the worker render a committed and reliable service.

1.3. OBJECTIVES

One of the main objectives of our project is to establish a services portal which is of mutual use to both the worker and the user and to make the life of the common man easy. It also gives an opportunity to workers to showcase their skills. Working on this project gives us a better experience that can be used in developing real time projects in coming future. Based on the feedback, the services of the worker can either be revoked or retained. This makes the worker render a committed and reliable service.

CHAPTER-2

SYSTEM ANALYSIS

2.1. EXISTING SYSTEM

An extensive manual search is required to be done in order to contact labour and find out information about them.

Disadvantages:

- In the present scenario, a person must visit various individual sites separately to get his job done.
- The user may have to collect information from various sources like friends and neighbours.
- Proximity of workers to users is not known unless a direct contact is made with the workers.
- This takes a lot of time and the information may not be reliable.
- Moreover depending on the distance of the workplace the worker may not be willing to lend his services.
- Worker information is not verified.
- The feedback of workers is also not available.

2.2. PROPOSED SYSTEM

We will be developing a standard mobile- friendly web application to retrieve worker information which can be viewed by the users.

Advantages:

- The proposed system maintains a repository of all types of work related information at one place and enables one to make direct contact with the worker.
- Workers can get self-employed through our system.
- As the worker himself seeks entry into the system there is a greater sense of responsibility on his part.
- Anyone can browse through all the details and gather information regarding workers without any delay.
- Since the address of the worker is available a user can contact a worker nearer to his house.
- Information about workers is manually verified and only authenticated workers are enlisted in our system.
- User feedback of workers is also available.
- Based on the feedback, the admin can either revoke or retain the services of a worker. This creates a sense of commitment within the workers.
- We have used the mobile-friendly website builder – Mobirise. This makes our website easily accessible from mobile phones also.

2.3. SOFTWARE REQUIREMENT

Front End: HTML, CSS, JS

Back End: PHP, SQL

Database: MySQL

Tools: Mobirise (mobile-friendly website builder)

Web Hosting: Infinityfree.net

Operating system: Cloud Based

CHAPTER-3

3.1. SYSTEM DESIGN AND ARCHITECTURE

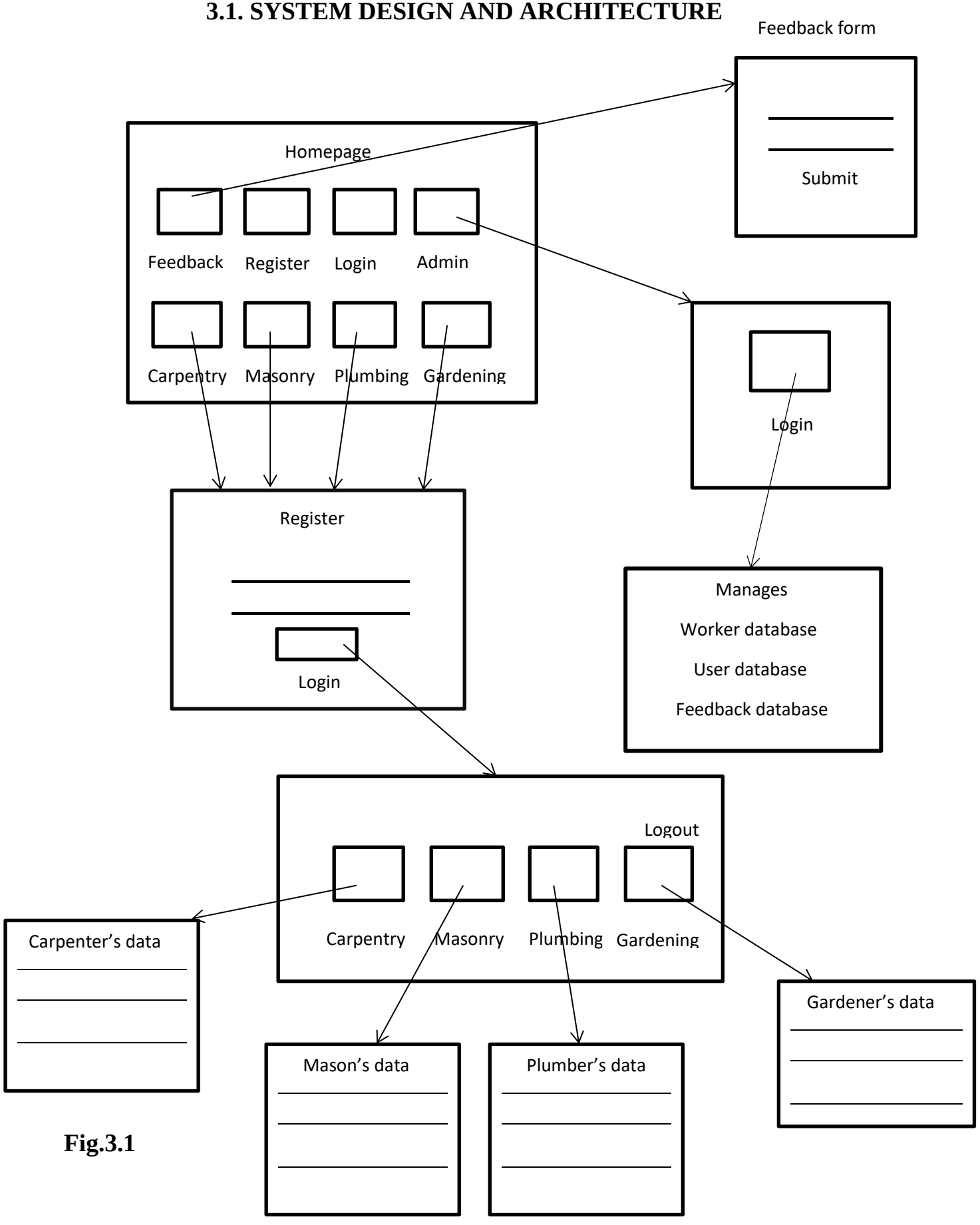


Fig.3.1

There is an initial homepage in Fig.3 consisting of the following buttons: Contact Us, About Us, Feedback, Register, Login and Admin. The different categories of workers are also displayed on the homepage.

Contact Us: Workers seek entry into our system by clicking this button.

About Us: It gives the information about website developers.

Feedback: It allows users to give feedback about worker's performance through google forms.

Register: A new user/worker needs to register to gain entry into the system and is redirected to worker's data.

Login: A registered user/worker can login to the system to seek information.

Admin: Only the admin has access to this module. He manages the worker's data, user's data and feedback about worker's performance given by the user.

After successful login a page displays different worker categories. Each category contains the name, address, experience, and phone no. of workers who have registered with the website. After viewing information the user logs out from the website.

3.2. MODULES

The system has the following modules:

Registration and login: A new user/ worker can register himself with the website and an existing user/worker can log in to the website with the help of a password.

Administration: The admin module centralizes all activities to be done in this application. The Admin authenticates worker information and maintains the database. Admin can add, remove and edit information (insert, delete and update) related to workers. He also has access to worker's feedback given by the users. Depending on the feedback the admin can either revoke or retain the services of a worker.

User: A new user has to register and login. A registered user can login with his registered name and password. The user can view information relating to any of the category of workers. A user can give feedback about the worker's performance.

Worker: A new worker has to register and login. A registered worker can login with his registered name and password. The worker can view information relating to any of the category of workers. If he wants to register himself as a worker/edit his information, he contacts the admin who manages the workers information.

3.3. ADDITIONAL FEATURES

To make the system more user-friendly the following features are added:The language translator feature where the user now has an option of browsing the website in English, Hindi or Telugu. The visitor counter is added to know the number of hits to the website. The website has been submitted to google and on authenticating the site it is made available on the google search engine.

3.4. DATA FLOW DIAGRAM

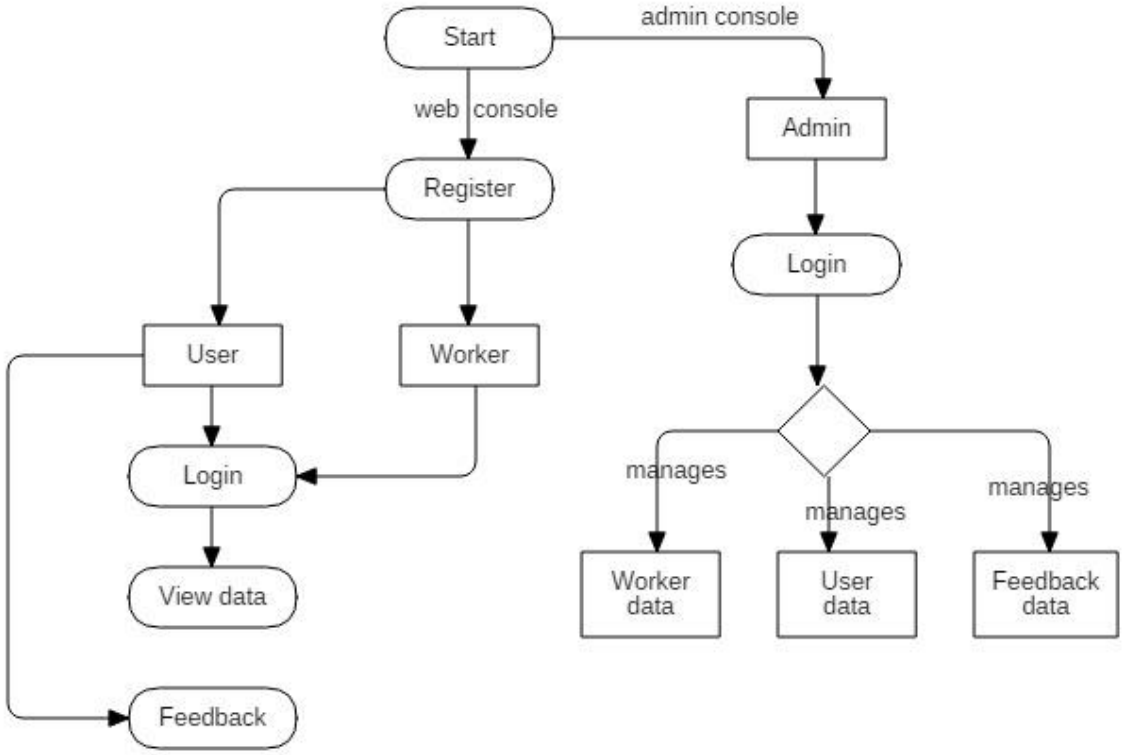


Fig.3.4 DFD for Blue Collar Services

A user or worker has to initially register himself with the website. They now have access to worker's database consisting of the name, phone no., address and experience of the worker. Users make a note of this information and directly contact the workers according to their requirement. Workers can get employed through our website. Workers make a request to the Admin to include their information in the website. The Admin manually verifies their data and includes their services. He also maintains the user, worker and feedback databases. Worker Feedback is an important aspect of our website. Based on the feedback given by the users, the admin can either revoke or retain the services of a worker. This makes the worker render a committed and reliable service.

3.5. UML DIAGRAMS

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems.

UML includes the following diagrams namely: Use case diagram, Class diagram, Sequence diagram, Activity diagram etc.

The goal of UML can be defined as a simple modeling mechanism to model all possible practical systems in today's complex environment.

3.5.1. USE CASE DIAGRAM

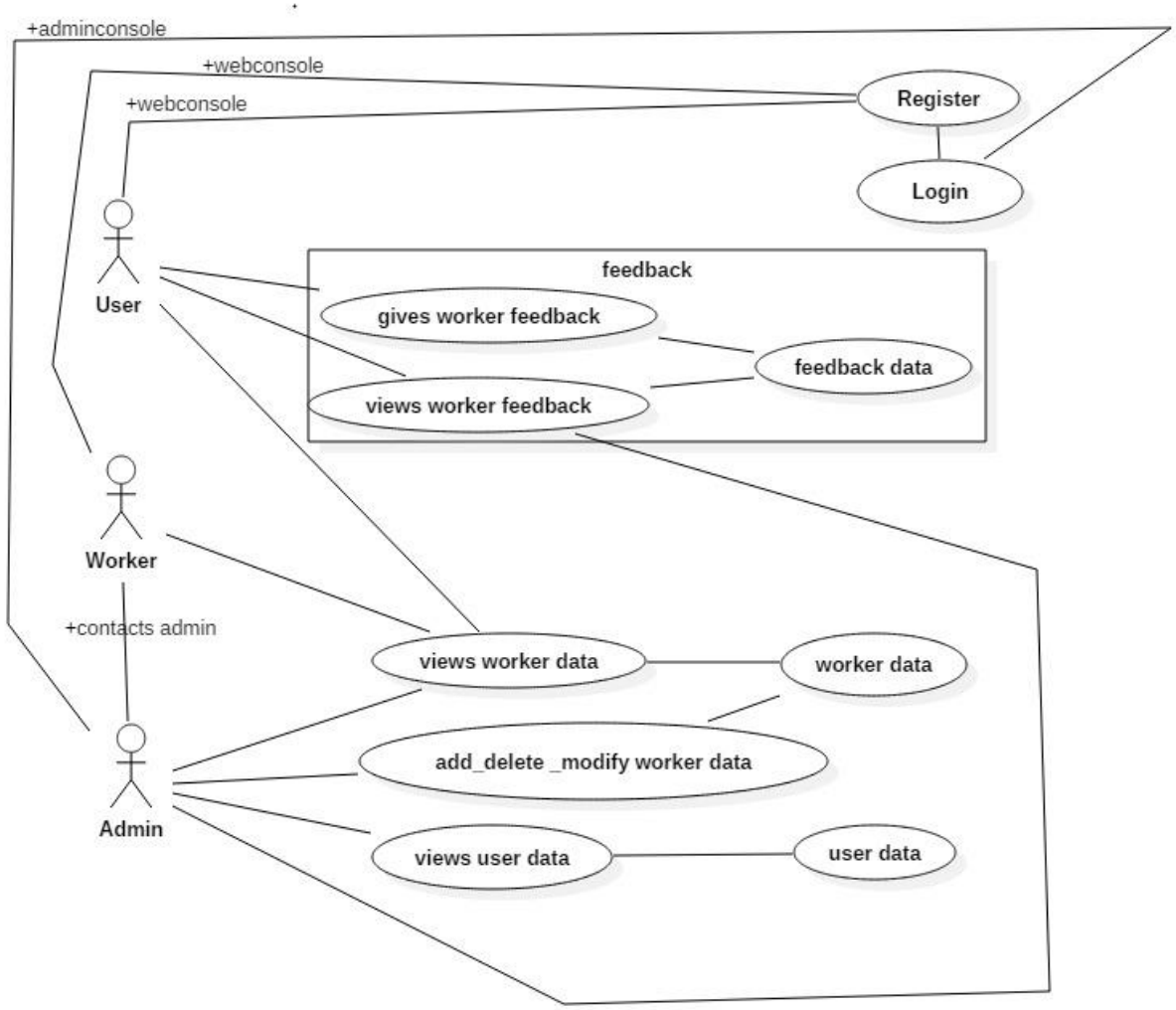


Fig.3.5.1 Use Case Diagram for Services

3.5.2. CLASS DIAGRAM

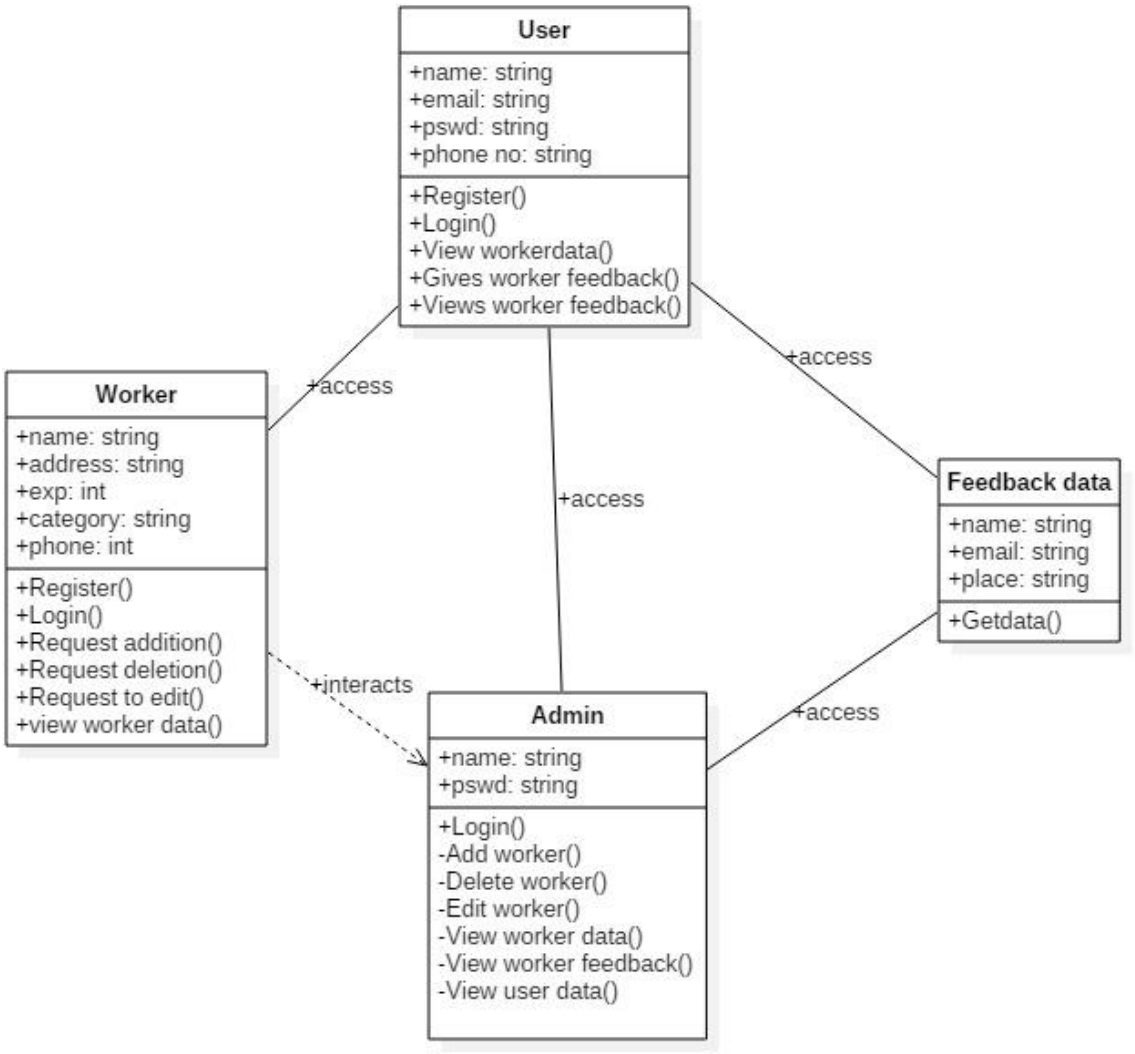


Fig.3.5.2 Class Diagram for User, Worker and Admin

3.5.3. SEQUENCE DIAGRAM

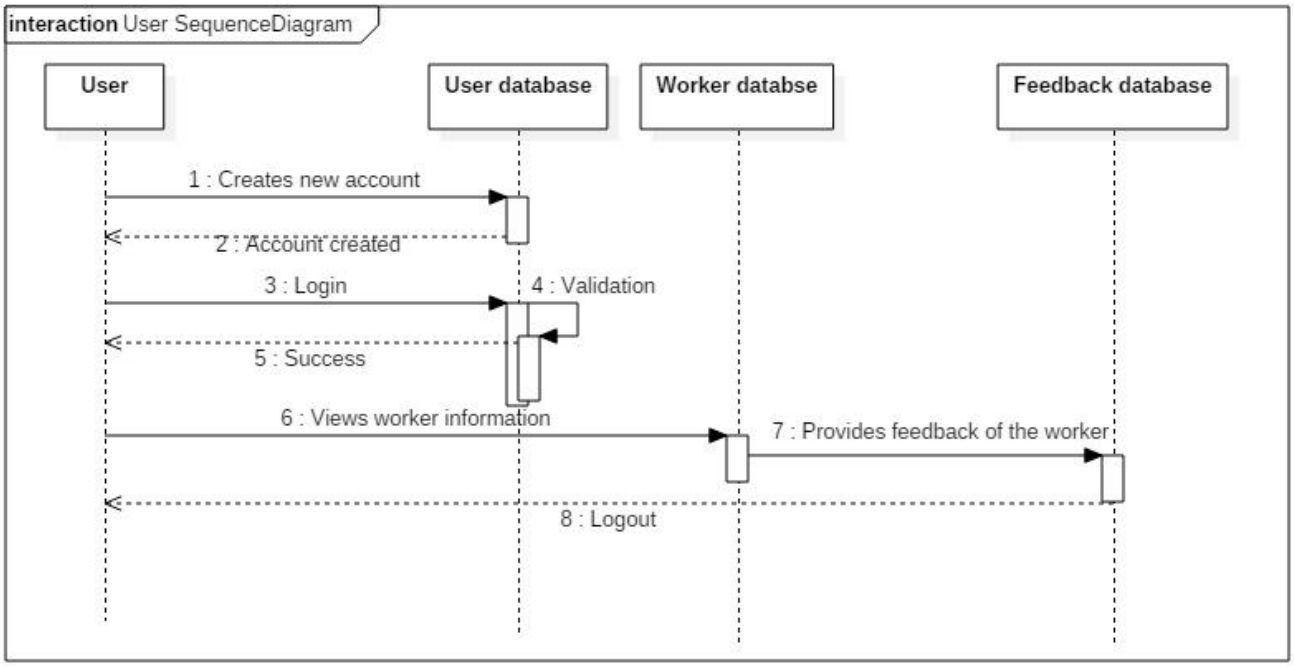


Fig.3.5.3.1 Sequence Diagram for User

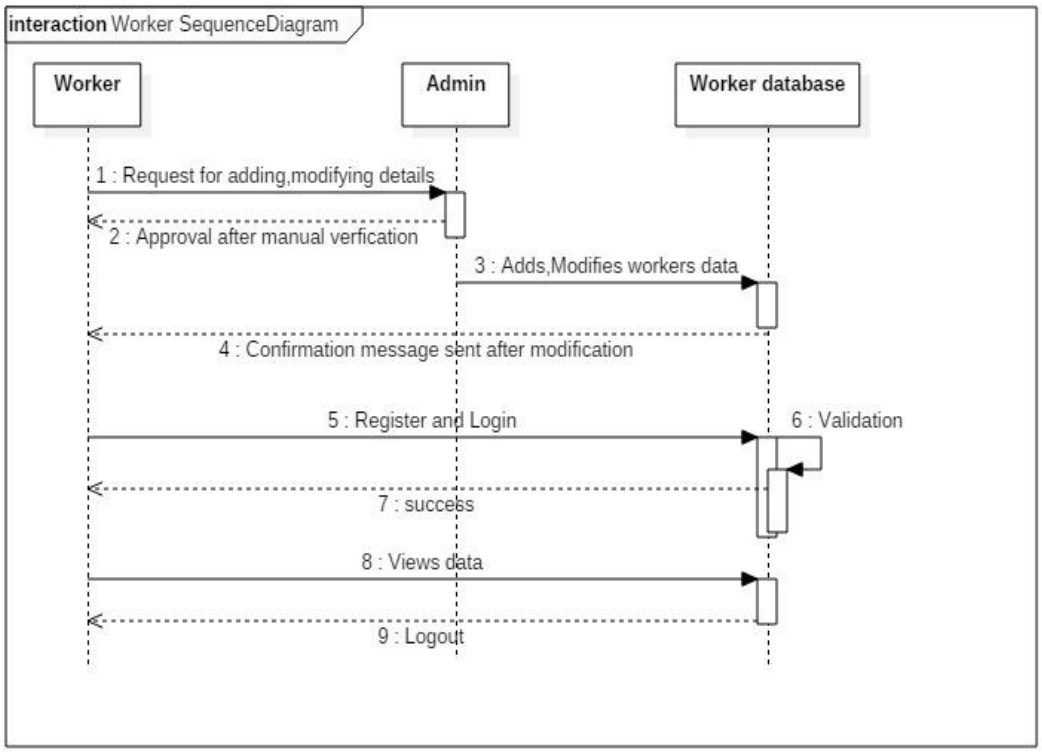


Fig.3.5.3.2 Sequence Diagram for Worker

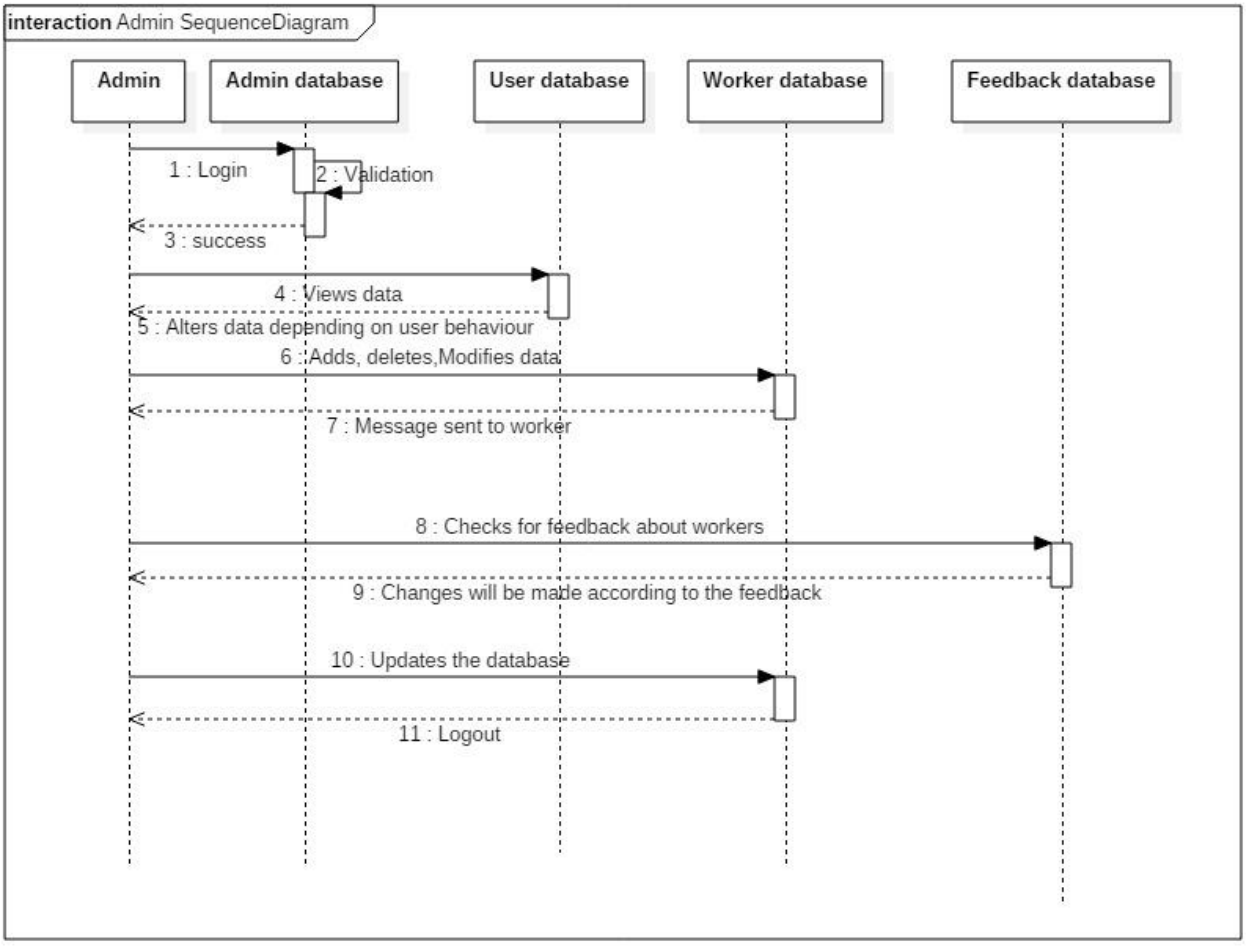


Fig.3.5.3.3 Sequence Diagram for Admin

3.5.4. ACTIVITY DIAGRAM

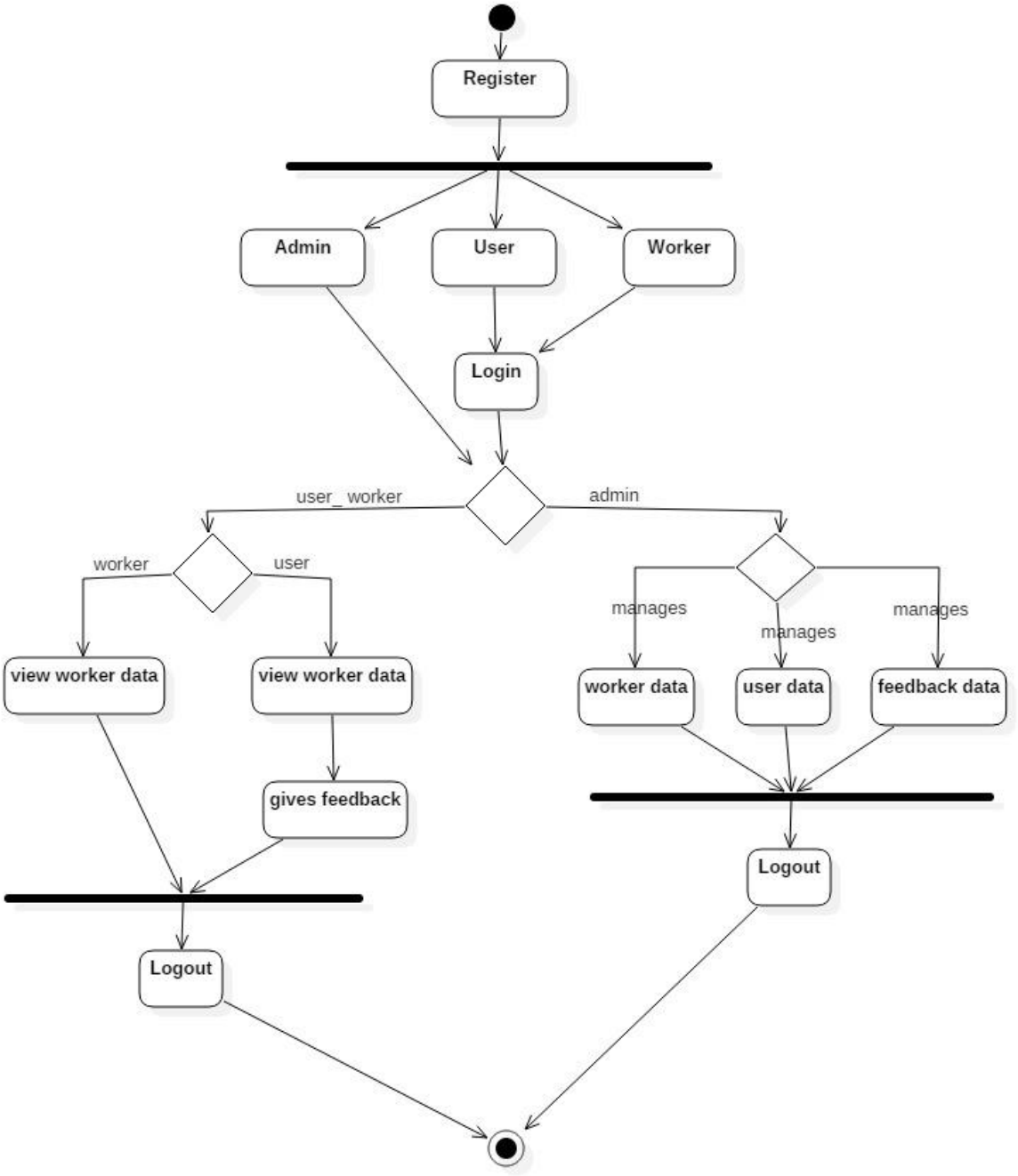


Fig.3.5.4 Activity Diagram for Services

CHAPTER-4

IMPLEMENTATION

The user interface which takes input or where the output is displayed to the user is done using HTML, CSS and JS. The module to which all the input of the front end goes or from where all the output comes to front end after processing through various algorithms is done using PHP and SQL. The database where the information is stored is done using MySQL .The service that allows individuals to post a website on to the internet is done using infinityfree.net. The website is built using Mobirise tool.

4.1. SAMPLE CODE

Login code

```
<form name="form1" action=" " method="post">
   <div class="row row-sm-offset">
                    <div class="col-xs-12 col-md-6">
                     <div class="form-group">
                        <tr>
                <div style="margin-left: 50%">
                      <table>
                           <td>Name</td>
                          <td><input type="text" name="t1"></td>
                        </table >
               </div>
                          </tr>
                     </div>
                   </div>
                   <div class="col-xs-12 col-md-6">
                     <div class="form-group">
                       <tr>
                                <td>Password</td>
                           <td><input type="password" name="t2"></td>
                       </tr>
                     </div>
                   </div>
                        <table style="margin: auto;">
                         <div style="margin-left: 45%">
                        <td colspan="2" align="center"><input type="submit"     class="btn btn-primary"
name="submit1" value="LOGIN"></td>
                       </div>
                       </table>
</form>
```

Registration code

```
<form action=" register.php" method="post" data-form-title="Registration">

                <input type="hidden"
value="SeGyeLf9MPU1pnOaqb4FuwGSv3gOCSAhN7AAdH68MZ2xc8ecaCUSWb5hLN9dkKFOu3ddw
m4CaB2sXUmLuuaJ/4TDId294eXPJ0koZm7ELmv902I8MX4qp2yLHyNNZVhB" data-form-
email="true">
                <div class="row row-sm-offset">
                  <p>New member</p>
                  <div class="col-xs-12 col-md-4">
                    <div class="form-group">
                     <label class="form-control-label" for="form1-c-name">Name<span class="form-
asterisk">*</span></label>
                      <input type="text" class="form-control" name="name" required="" data-form-
field="Name" id="form1-c-name">
                    </div>
                  </div>
                   <div class="col-xs-12 col-md-4">
                    <div class="form-group">
                     <label class="form-control-label" for="form1-c-name">Email<span class="form-
asterisk">*</span></label>
                      <input type="text" class="form-control" name="email" required="" data-form-
field="Email" id="form1-c-name">
                    </div>
                  </div>

                  <div class="col-xs-12 col-md-4">
                    <div class="form-group">
                      <label class="form-control-label" for="form1-c-phone">Phone<span
class="form-asterisk">*</span></label>
                      <input type="tel" class="form-control" name="phone" required="" data-form-
field="Phone" id="form1-c-phone">
                    </div>
                  </div>
                 <div class="col-xs-12 col-md-4">
                    <div class="form-group">
                      <label class="form-control-label" for="form1-c-name">Password<span
class="form-asterisk">*</span></label>
                      <input type="password" class="form-control" name="password" required=""
data-form-field="Password" id="form1-c-name">
                    </div>
                  </div>

                </div>

                <div><button type="submit" class="btn btn-primary"  >Register</button></div>

                 </form>
```

Admin code

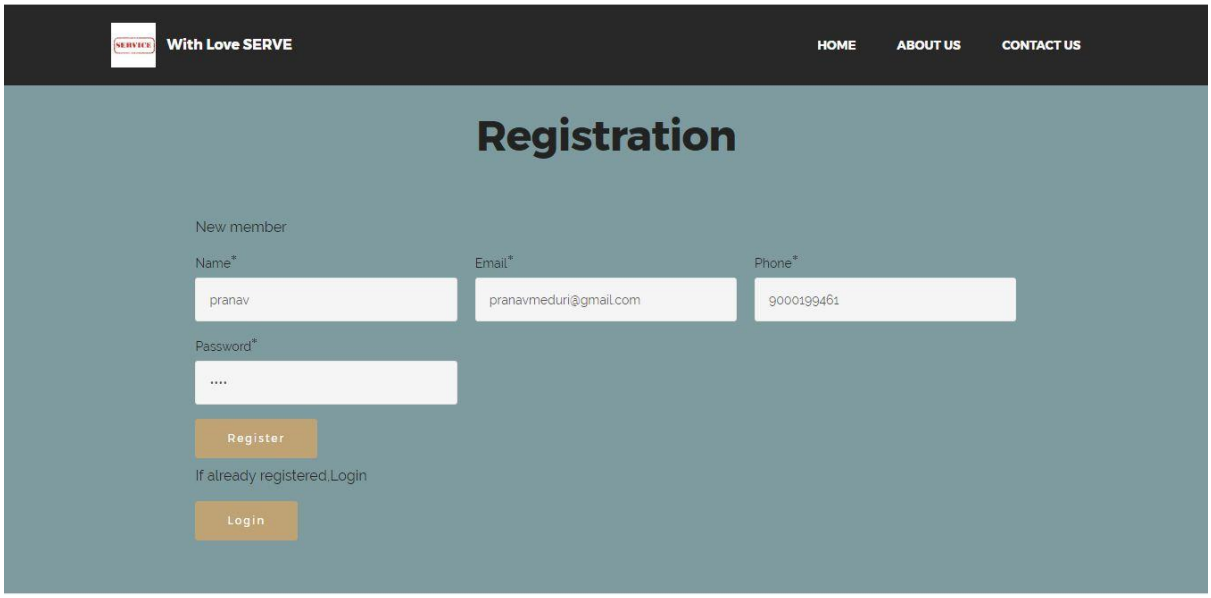
```
<body>
<form name="form1" action="adminc1.php" method="post">
<table>
    <tr>
      <td>Name<td>
      <td><input type="text" name ="t1"></td>
   </tr>
    <tr>
      <td>Phone<td>
      <td><input type="text" name ="t2"></td>
   </tr>
   <tr>
      <td>Address<td>
    <td>  <textarea name="t3"></textarea></td>

   </tr>
     <tr>
      <td>Exp<td>

       <select name="t4">
<?php
  for($i=0;$i<=30;$i++)
{
 echo "<option value=".$i.">".$i." yrs</option>";
}
?>

     </select>
   </tr>
  <tr>
      <td><colspan="2" align="center" ><input type="submit" name="submit1" value="insert"></td>
       <td><colspan="2" align="center" ><input type="submit" name="submit2"
value="delete"></td>
      <td><colspan="2" align="center" ><input type="submit" name="submit4"
value="update"></td>
       <td><colspan="2" align="center" ><input type="submit" name="submit3"
value="display"></td>

  </tr>
</table>
</form>

</body>
```

```
<body>
<form name="form1" action="adminc.php" method="post">
</tr>
  <tr>
     <td><colspan="2" align="center" ><input type="submit" name="submit" value="back"></td>

      </tr>
</table>
</form>
</body>
</html>
 <?php

include "db.php";

if(isset($_POST["submit1"]))
{
$add=addslashes($_POST["t3"]);
echo $add;
$sql="INSERT INTO table4 (name,phone,address,exp) values
('{$_POST["t1"]}','{$_POST["t2"]}','{$add}','{$_POST["t4"]}')";
echo $sql;

if (mysqli_query($conn, $sql)) {
   echo "Inserted successfully";
} else {
   echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
}

if(isset($_POST["submit2"]))
{
  $sql = "DELETE FROM table4 WHERE name='{$_POST["t1"]}' ";

if (mysqli_query($conn, $sql)) {
   echo " Deleted successfully";
} else {
   echo "Error : " . mysqli_error($conn);
}

}

if(isset($_POST["submit3"]))
{
echo "<table>";
   $result = mysqli_query($conn,"SELECT * FROM table4 ");


 echo "<table border =1  align=center  style=width:50% margin:auto  >";
    echo "<tr>";
echo "<th>";  echo 'sno';    echo"</th>";
```

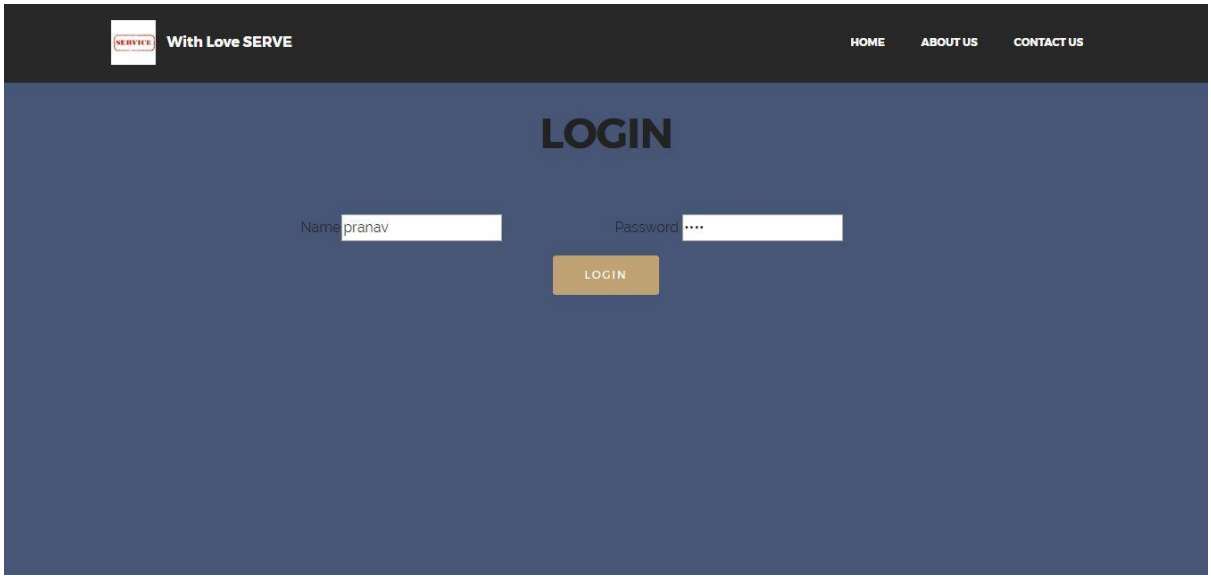
```
    echo "<th>";  echo 'name';    echo"</th>";
     echo "<th>";  echo 'phone';    echo"</th>";
       echo "<th>";  echo 'address';    echo"</th>";
       echo "<th>";  echo 'exp';    echo"</th>";
     echo "</tr>";
  while($row = mysqli_fetch_array($result))
     {

 //   echo $row['sno'] . " " . $row['name']. " " . $row['phone'] . " " . $row['address']. " " . $row['exp'];
//these are the fields that you have stored in your database table employee
 //     echo "<br />";

    echo "<tr>";

   echo "<td>";  echo $row['sno'];    echo"</td>";
    echo "<td>";  echo $row['name'];    echo"</td>";
     echo "<td>";  echo $row['phone'];    echo"</td>";
       echo "<td>";  echo $row['address'];    echo"</td>";
       echo "<td>";  echo $row['exp'];    echo"</td>";
     echo "</tr>";
    }
  echo "</table>";

//} else {
//    echo "0 results";
//}

echo "</table>";

}
  if(isset($_POST['submit4']))
{

    $sql = "UPDATE table4 SET  phone='{$_POST["t2"]}'
,address='{$_POST["t3"]}',exp='{$_POST["t4"]}'   WHERE name='{$_POST["t1"]}' ";

if ($conn->query($sql) === TRUE) {
   echo "Record updated successfully";
} else {
   echo "Error updating record: " . $conn->error;
}

}

mysqli_close($conn);

                                    ?>
```

4.1. TEST CASES

Successful search results can be shown through the following cases.

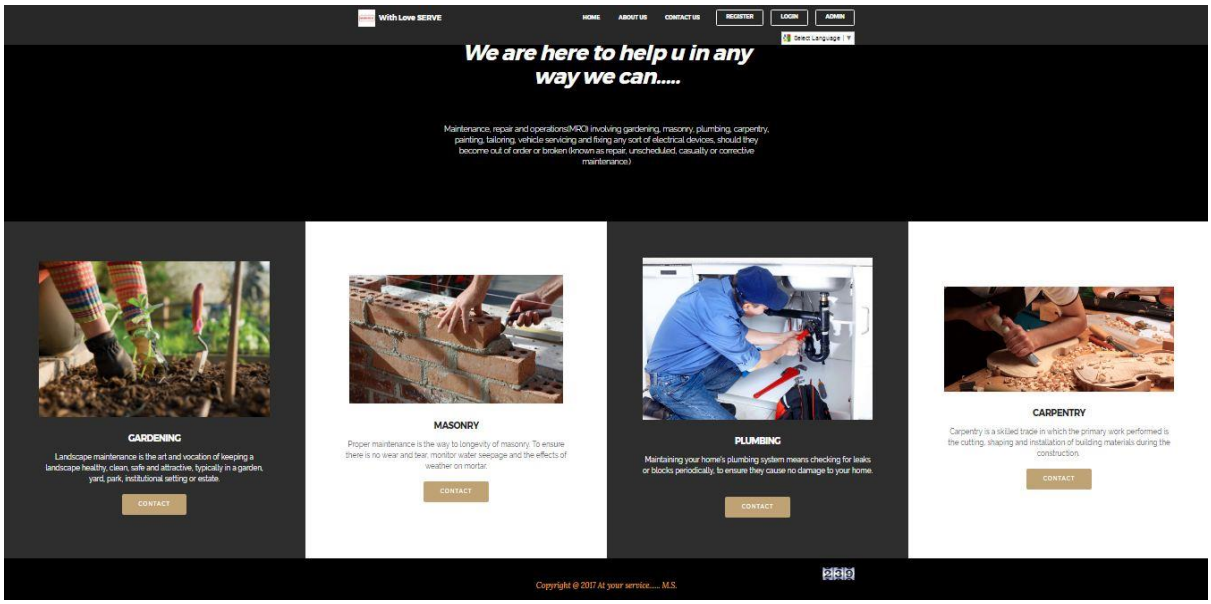


Register



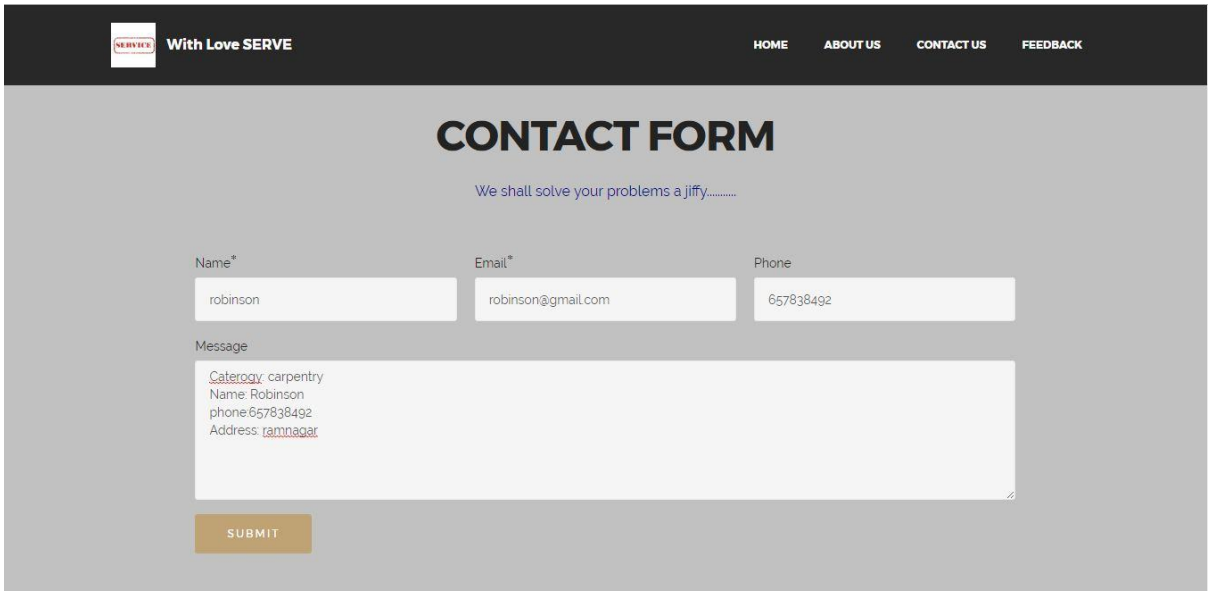
Login

4.3. OUTPUT SCREENS

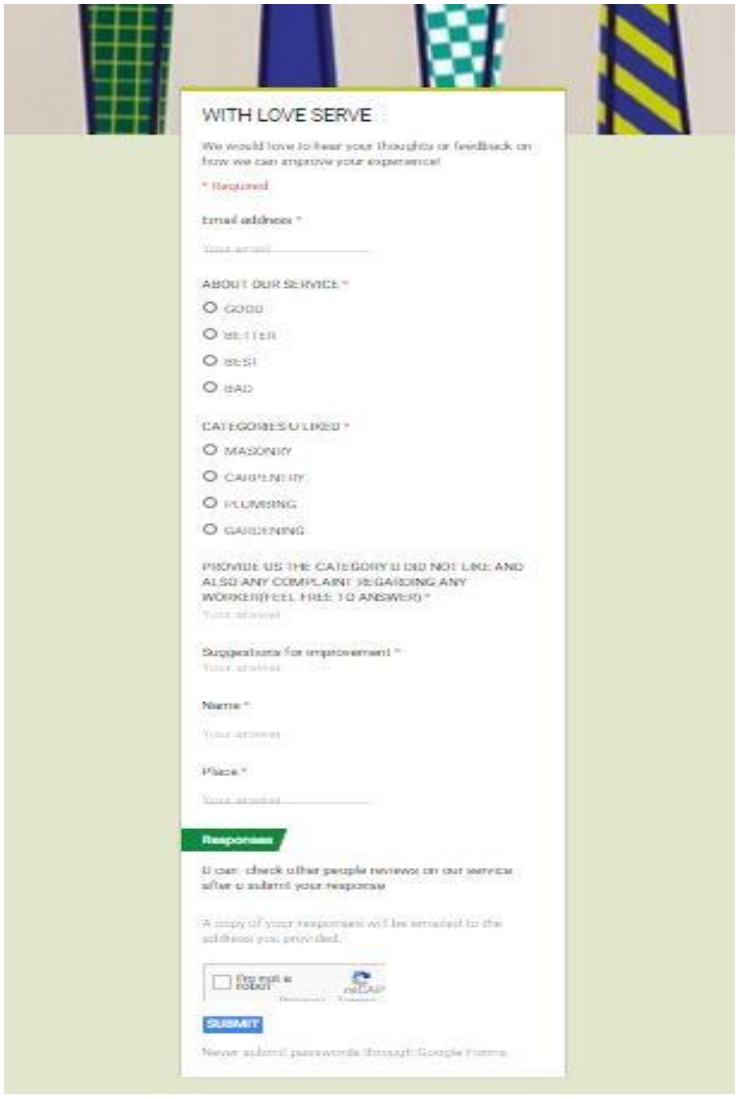


Homepage

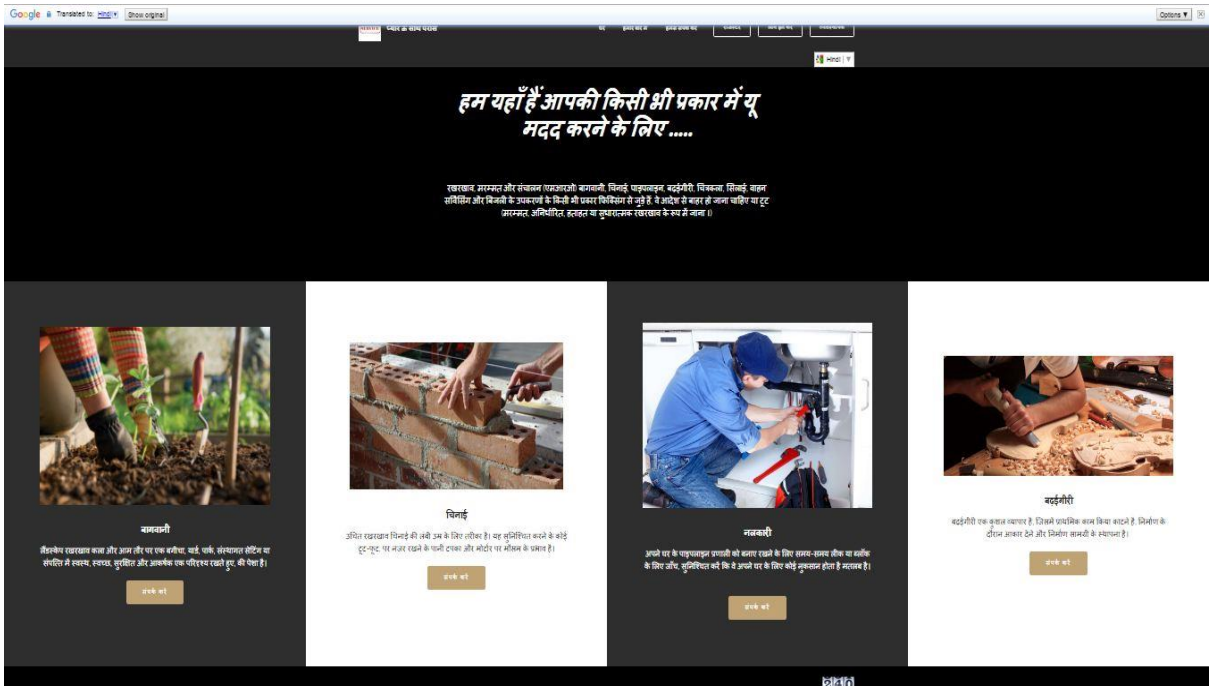
About us



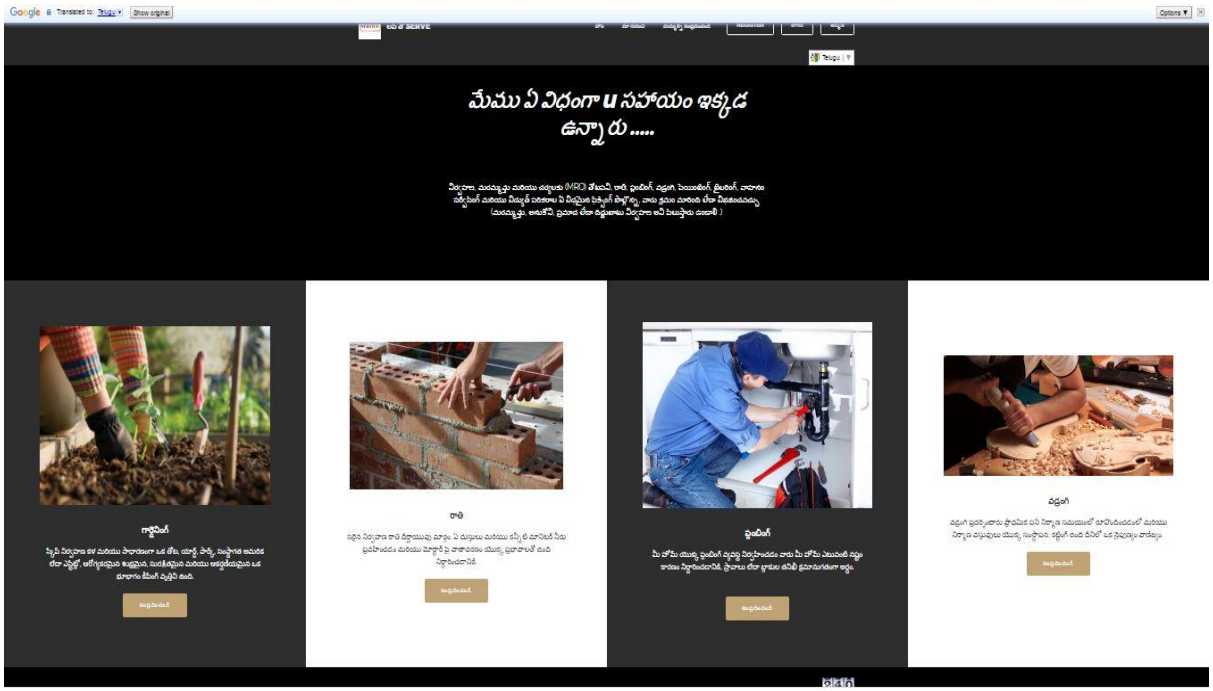
Contact us



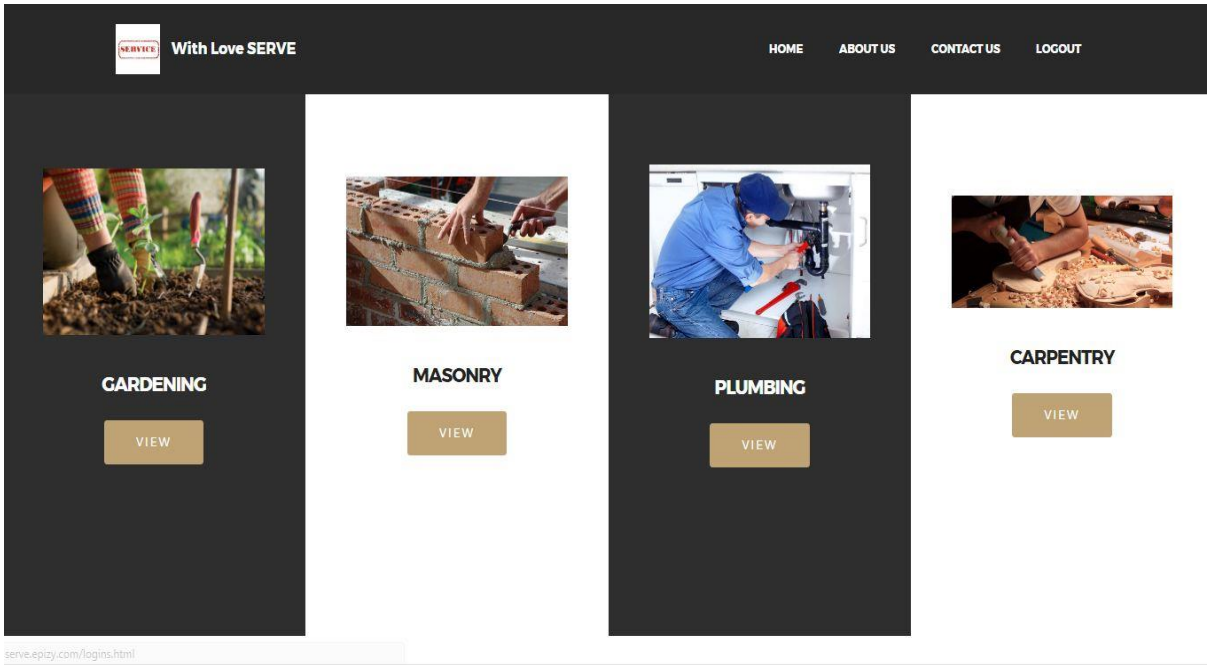
Feedback



Website in Hindi



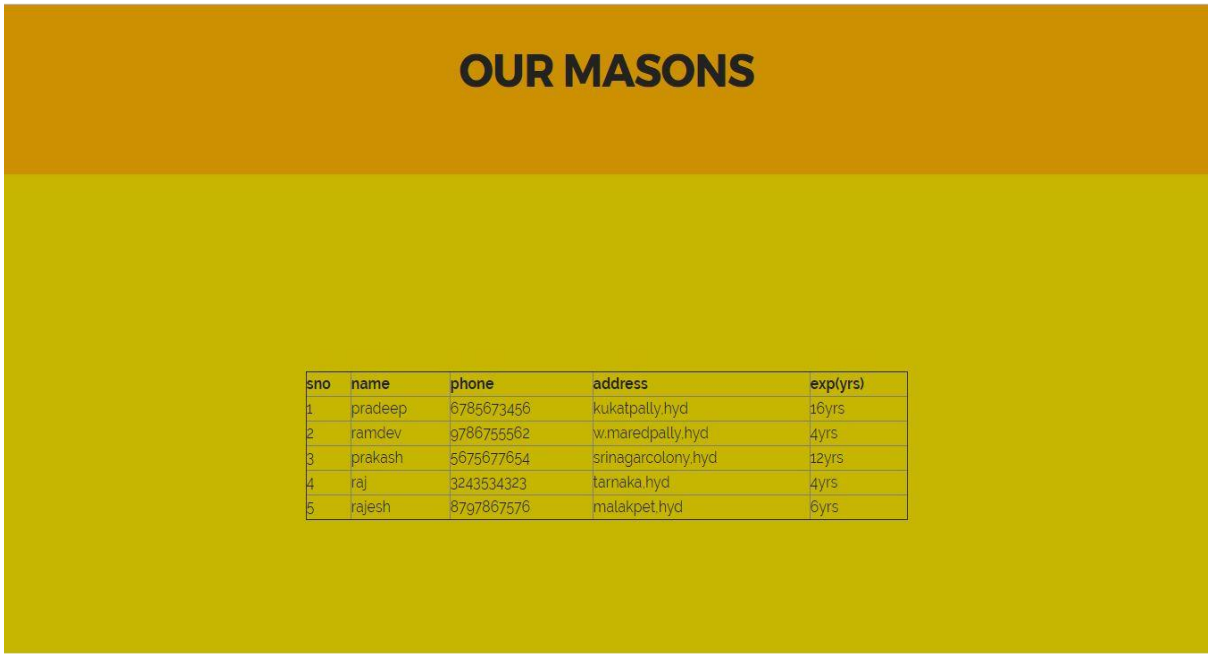
Website in Telugu



User Homepage

sno	name	phone	address	exp(yrs)
1	pranav	9848016120	himayathnagar,hyd	10yrs
2	ramesh	9254643782	nadergul,hyd	11yrs
3	keshav	9878978657	khairatabad,hyd	4yrs
4	raghu	6786908567	secundarabad,hyd	12yrs
5	sriram	4567876545	panjagutta,hyd	9yrs

Gardener’s data



sno	name	phone	address	exp(yrs)
1	pradeep	6785673456	kukatpally.hyd	16yrs
2	ramdev	9786755562	w.maredpally.hyd	4yrs
3	prakash	5675677654	srinagarcolony.hyd	12yrs
4	raj	3243534323	tarnaka.hyd	4yrs
5	rajesh	8797867576	malakpet.hyd	6yrs

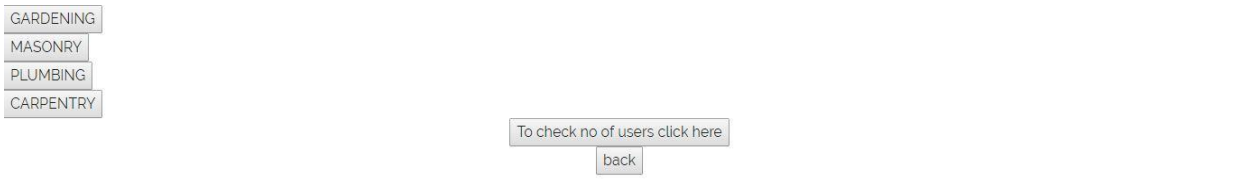
Mason’s data

sno	name	phone	address	exp(yrs)
1	raghu	5676456342	dilshuknagar.hyd	3yrs
2	sriram	9878978767	meerpet.hyd	9yrs
3	lakhshman	8786756034	saidabad.hyd	8yrs
4	rishi	4565456543	kukatpally.hyd	5yrs
5	neel	6756045324	srinagarcolony.hyd	2yrs

Plumber’s data

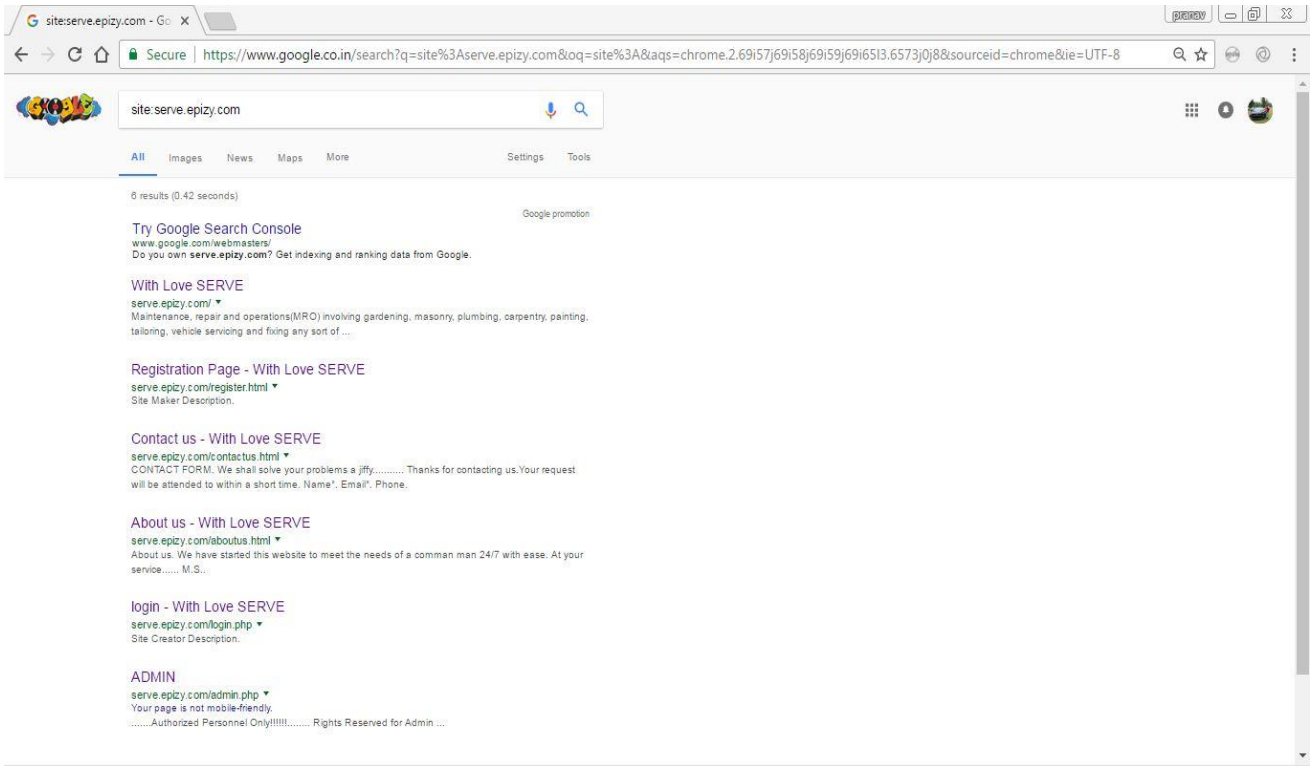
Carpenter’s data

Admin



Admin operations

Admin Adds, Deletes and Manages Worker’s data



Google search engine feature

CHAPTER-5

CONCLUSION

5.1. INFERENCE

We have successfully developed the project which is of great use to both the users as well as the workers. The results show that by the simple click of a button any authenticated worker information is displayed with ease. It provides a platform for the workers to show case their skills and the worker feedback feature creates a sense of commitment and reliability among workers.

5.2. FUTURE ENHANCEMENTS

Through this project users can browse worker information and hire their services. There is a lot of scope for enhancing the features of the website. Users can rank the workers directly instead of giving feedback. Workers can directly include themselves in the website without contacting the admin. Online payment services can be included. More categories of workers can be added depending on requirement. Combo packages can be provided at discounted prices.

REFERENCES

[1] www.w3schools.com/html/

[2] www.w3schools.com/php/

[3] www.w3schools.com/sql/

[4] www.w3schools.com/js/