1. INTRODUCTION

1.1 Aim

Emergency Electronic Banking (EEB), project aims to develop a comprehensive platform that provides the location and information of business correspondent agents (BC agents) across various regions. The project's goal is to enhance financial inclusion by making it easier for individuals and businesses to access banking services through BC agents. By leveraging modern technology and data-driven solutions, the project seeks to bridge the gap between banks and underserved communities.

1.2 Purpose

The purpose of this document is to present a detailed description of the **Emergency Electronic Banking (EEB).** It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and clients.

1.3 Scope

The future of the Emergency Electronic Banking (EEB) project is promising, with potential for growth and impactful developments. This includes expanding to new regions, partnering with more financial institutions, improving the platform's features and functionality, leveraging data for insights, collaborating with organizations, integrating with digital payment systems, providing financial education, and advocating for supportive policies. These efforts aim to ensure greater financial inclusion and empower underserved communities with accessible banking services.

1.4 About Project

The Emergency Electronic Banking (EEB) project is designed to address the challenges of financial inclusion by providing a comprehensive platform that enables individuals and businesses to easily locate and access the services of business correspondent agents (BC agents). These agents act as intermediaries for banks and financial institutions, bringing banking services to underserved areas and communities.

The project's core objectives revolve around creating a user-friendly web application that centralizes the information of BC agents. The platform allows users to search for agents based on their location, specific services required, or other relevant criteria. It also provides a means for users to communicate with BC agents, inquire about services, and schedule appointments.

The project emphasizes the expansion of financial services to underserved regions. It aims to build a centralized database that contains accurate and up-to-date information about registered BC agents, including their contact details, service areas, and operational hours. This information is verified to ensure the legitimacy of the agents and protect users from fraudulent activities.

The future scope of the project is promising. It includes potential expansion to new regions, collaboration with a wider range of financial institutions, and the incorporation of new features and functionalities based on user feedback and emerging technologies. The project also envisions leveraging data analytics to gain insights into user preferences and demand for specific banking services.

Partnerships and collaborations with government agencies, non-profit organizations, and technology providers are crucial aspects of the project's future scope. These collaborations can amplify the impact of the project and foster innovation in the field of financial inclusion.

to the development of supportive policies and regulations.

In conclusion, the Emergency Electronic Banking (EEB) project endeavors to enhance financial inclusion by providing a convenient platform for locating and accessing the services of BC agents. By leveraging technology, data-driven solutions, and partnerships, the project aims to bridge the gap between banks and underserved communities, fostering economic

Furthermore, integrating Emergency Electronic Banking (EEB) platform with digital payment systems, offering financial education resources, and engaging in policy advocacy are additional opportunities for the project's growth. These efforts aim to empower underserved communities, promote financial literacy, and contribute growth and empowering individuals and businesses with improved access to banking services.

2. LITERATURE SURVEY

2.1 Existing System

In current scenario, suppose a user wants to get a banking facility he/she will visits nearby bank or ATM, but if they are at a remote area it is difficult, so many people are facing problem with banking without knowing the exact information. At the end they exit without getting any facility.

Limitation of Existing System

- 24/7 unavailability
- Cash provided only to people with all account details
- Door step banking
- Service charges may apply

To avoid all these limitations and make the working more efficient, the system needs to be computerised.

2.2 Proposed System

The project Emergency Electronic Banking (EEB) is a website developed in a way to overcome the existing problem faced by the users. The website mainly helps user to find a BC outlet to get the banking facility at the remote places where there will be no bank's or ATM's. This website provides BC outlet location, BC agent information. This website is developed to make this process easy, time saving for the users

Expected Advantages Of Proposed System:

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features.

- Ensure data accuracy' s.
- Minimum time needed for the various processing.
- Greater efficiency, User friendliness and interactive.
- Better service and Minimum time required.
- No service charges applied

2.3 Technologies Used

2.3.1 Hyper Text Markup Language

HTML Document Is

HTML document are plain-text (also known as ASCII) files that can be created using any text editor (e.g. Emacs or vi on UNIX machines; SimpleText on a Macintosh, Notepad on a Windows machine). You can also use word-processing software if you remember to save your document as "text only with line breaks".

Tags Explained

An element is a fundamental component of the structure of a text document. Some examples of elements are heads, tables, paragraphs, and lists. Think of it this way: you use HTML tags to mark the elements of a file for your browser. Elements can contain plain text, other elements, or both.

To denote the various elements in an HTML document, you use tags HTML tags consist of a left angle bracket (<), a tag name, and a right angle bracket (→). Tags are usually paired (eg. <H1> and </H1>) to start and end the tag instruction. The end tag looks just like the start tag except a slash (/) precedes the text within the brackets.

Some elements may include an attribute, which is additional information that is included inside the start tag For example, you can specify the alignment of images (top, middle or bottom) by including the appropriate attribute with the image source HTML code.

Note: HTML is not case sensitive. <title> is equivalent to <TITLE> or <TITLE> Not all tags are supported by all World Wide Web browsers. If a browser does no support a tag, it will simply ignore it Any text placed between a pair of unknown tags will still be displayed, however.

2.3.2 Introduction To Java Script

JavaScript is a client-side programming languages that you can use for creating interactive web pages JavaScript uses almost the same ideas present in Java. You can learn JavaScript to embed JavaScript code in HTML pages and the Web browser interprets it accordingly. JavaScript used to be known initially as Mocha and was originally developed by Brendan Eich of Netscape. Later on it was renamed to Live Script, and finally to JavaScript.

Although it shares many of the features and structures of the full Java language, it was developed independently. JavaScript can interact with HTML source code and can help you to spice up your website with dynamic content. It is an open language which you can use without purchasing a license.

It is normally used in the form of client-side JavaScript and is implemented as part of a web browser to offer enhanced user interfaces and dynamic websites. However, it is used outside web pages too.

What Can You Do with JavaScript?

- Show alert messages wherever necessary
- Prompts user for input
- Get confirmation from users by using confirm dialog
- Write Change HTML tags on document Read HTML tags from document
- Print a web page
- Open or close browser windows
- Bind Us bind user events
- Calculation areas on pages
- Dynamically changing background and text colors, or "buttons". To look at the URL history and take action based on it.
- Dynamic forms that include built-in error checking

JavaScript plays a prominent role for advertisement by using rotating ads or banners script and may set a cookie on your computer indicating you have seen the advertisement. JavaScript is used in e-commerce sites and allows adding items to your shopping cart, processing forms and submitting orders to be shipped. It can also be used to display dynamic content to users while shopping

2.3.3 Introduction to PHP

The past five years have been fantastic in terms of the explosive growth of the Internet and the new ways in which people are able to communicate with one another Spearheading this phenomenon has been the World Wide Web (WWW), with thousands of new sites being launched daily and consumers being consistently offered numerous outstanding services via this new communication medium. With this exploding market has come a great need for new technologies and developers to learn these technologies Chances are that if you are reading this paragraph, you are one of these Web developers or are soon to become one. Then you've heard of the great new technology called PHP.

Characteristics of PHP:

As you may have realized, the PHP languages revolves around the central theme of practicality. PHP is about providing the programmer with the necessary tools to get the job done in a quick and efficient fashion

Five important characteristics make PHP's practical nature possible:

- Familiarity
- Simplicity
- Security
- Flexibility
- Efficiency
- It is free (so it makes interesting)

1. Familiarity:

Programmers from many backgrounds will find themselves already accustomed to the PHP languages Many of the language's constructs are borrowed from C and Perl, and in many cases PHP code is almost indistinguishable from that found in the typical C or Pascal program. This minimizes the learning curve considerably.

II. Simplicity:

A PHP script can consist of 10,000 lines or one line; whatever you need to get the job done. There is no need to include libraries, special compilation directives, or anything of the sort. The PHP engine simply begins executing the code after the first escape sequence (<?) and continues until it passes the closing escape sequence (?>). If the code is syntactically correct, it will be executed exactly as it is displayed.

III. Efficiency:

Efficiency is an extremely important consideration for working in a multi-user environment such as the WWW.PHP 4.0 introduced resource allocation mechanisms and more pronounced support for object- oriented programming, in addition to session management features. Reference counting has also been introduced in the latest version, eliminating unnecessary memory allocation.

IV. Security:

PHP provides developers and administrators with a flexible and efficient set of security safeguards. These safeguards can be divided into two frames of reference: system level and application level.

I. System-Level Security Safeguards
II. Application-Level Security Safeguards

V. Flexibility:

Because PHP is an embedded language, it is extremely flexible towards meeting the needs of the developer. Although PHP is generally touted as being used in conjunction solely with HTML, it can also be integrated alongside languages like JavaScript, WML, XML, and many others. Additionally, as with most other mainstream languages, wisely planned PHP applications can be easily expanded as needed. Browser dependency is not an issue because PHP scripts are compiled entirely on the server side before being sent to the user. In fact, PHP scripts can be sent to just about any kind of device containing a browser, including cell phones, Personal digital assistant (PDA) devices, pagers, laptops, not to mention the traditional PC. People who want to develop shell-based applications can also execute.

2.3.4 Introduction to My-SQL

A database is a structure that comes in two flavours: a flat database and a relational database. A relational database is much more oriented to the human mind and is often preferred over the gabble-de-gook flat databases that are just stored on hard drives like a text file. MySQL is a relational database. In a relational structured database, there are tables that store data. The columns define which kinds of information will be stored in the table. An individual column must be created for each type of data you wish to store (i.e. Age, Weight, and Height).

On the other hand, a row contains the actual values for these specified columns. Each row will have I value for each and every column. For example, a table with columns (Name, Age, Weight-lbs) could have a row with the values (Bobs, 65, 165). If all this relational database talk is too confusing, don't despair. We will talk about and show a few examples in the coming lessons.

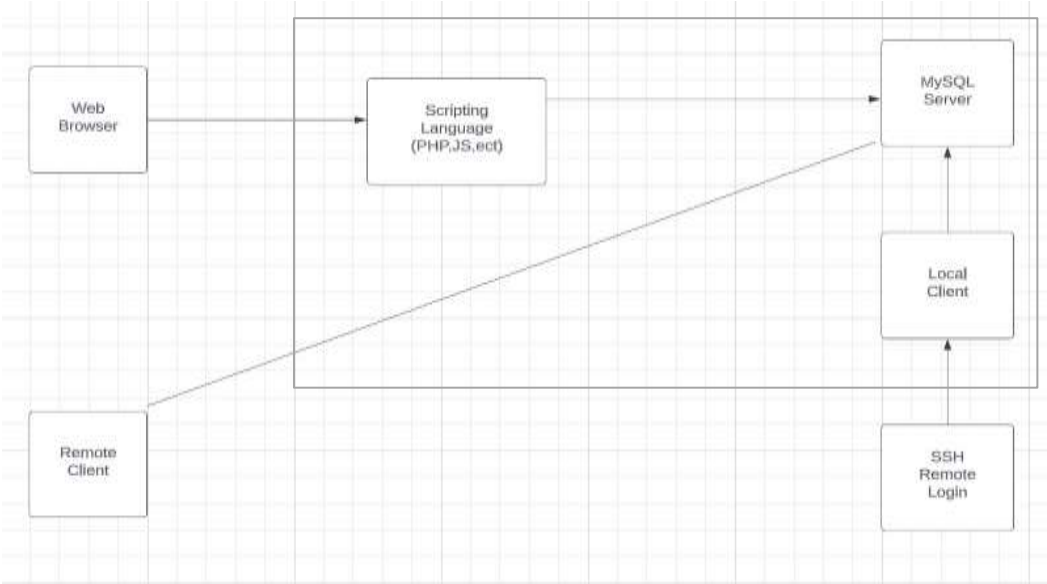
Database are most useful when it comes to storing information that fits into logical categories. For example, say that you wanted to store information of all the employees in a company. With a database you can group different parts of your business into separate tables to help store your information logically. Example tables might be: Employees, Superlatives, and Customers. Each table would then contain columns specific to these three areas. To help store information related to each employee, the Employees table might have the following columns: Hire, Date, Position, Age, and Salary.

PHPMyAdmin

Also supplied by most hosting services is phpMyAdmin (you can also install it anywhere you want, as it's open source and free). This tool will allow you to view all the MySQL database, tables, and entries, as well as perform SQL queries remotely through a web browser. Although we will be teaching how to create databases, tables and all other MySQL tasks through PHP, we encourage you to learn about phpMyAdmin, It's easy-to-use interface will allow you to do many common MySQL tasks quickly and easily.

My-SQL working

My-SQL is a database server program and as such is installed on one machine, but can "serve the database to a variety of locations. To explain look at the following diagram.



The MySQL Server is installed on a Server and can be accessed directly via various client interface, which send SQL statements to the server and then display the results to a user. Some of these are.

A Local Client - a program on the same machine as the server. An example of this is the command line My-SQL client software we will be using in the rest of the My-SQL workshops

A Scripting Languages - can pass SQL queries to the server and display the result

A Remote Client -a program on a different machine that can connect to the server and run SQL statements

3. HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Minimum Hardware Requirements

Server

Processor	:	Pentium-II or higher
Processor Speed	:	533 MHZ
RAM	:	32 MB (64 MB recommended)
Hard disk	:	20 GB (min)

Client

Processor	:	Pentium-II or higher
Processor Speed	:	533 MHZ
RAM	:	32 MB (64 MB recommended)
Hard disk	:	20 GB (min)

3.2 Minimum Software Requirements

Web server	:	XAMPP Servera
Back end	:	MYSQL
Server-side scripting	:	PHP. Codeigniter
Client-side scripting	:	HTML, CSS, Bootstrap, JavaScript

4. SYSTEM DESIGN

4.1 Modules:

This **"Emergency Electronic Banking"** Application has three modules

1. Admin
2. Bank
3. User

1.Admin

Admin is super user and administrator role is created for BC agents. Admin is the one who is responsible for adding up BC agents, adding bank, adding agent information and their location.

2.Bank

Bank get's the ability to be admin in future days, but for now the developers plays the roll agent. Bank get the information about the banking transaction done through the BC agents. And the information about the users who are getting these facility through the help of EEB website.

3.User

User gets registered within the application and then login using the email id and the password which is generated by the user itself. The user will avail the facilities of the website and get's the required information.

4.1.1 Functional and Non-Functional Requirements

Functional Requirements:

Admin

1. Admin can maintain the list of categories
2. Admin is responsible to maintain the agents
3. Admin is responsible to maintain the database

Bank

Bank plays a temporary role of agent

Bank is responsible for all the transactions done through BC agents

Bank also get's the benefit from this website

User

User needs to register to the application to login into the website

User can view all the features

User can get the BC agent information and their Google map location

User can get the required facility once they reached the BC outlet

Non-Functional Requirements

1. Correctness:

Since this project is used to provide the actual and correct information about details of the particular project which were done by students

2. Reliability:

The system as to provide the correct information under any situation, In case of any error in input, or operation, system should reflect proper message or give proper helping information

3. Robustness:

It's vital that the system should be a fault tolerant with respect to illegal user input. Error Checking must be built in the system to prevent system failure,

4. Maintainability:

The project will be used for a long time, it must be easy to maintain and easy to incorporate future changes. The design if the system should be module based and changing the design of the one module should not affect the proper operation of the other module.

5. Portability:

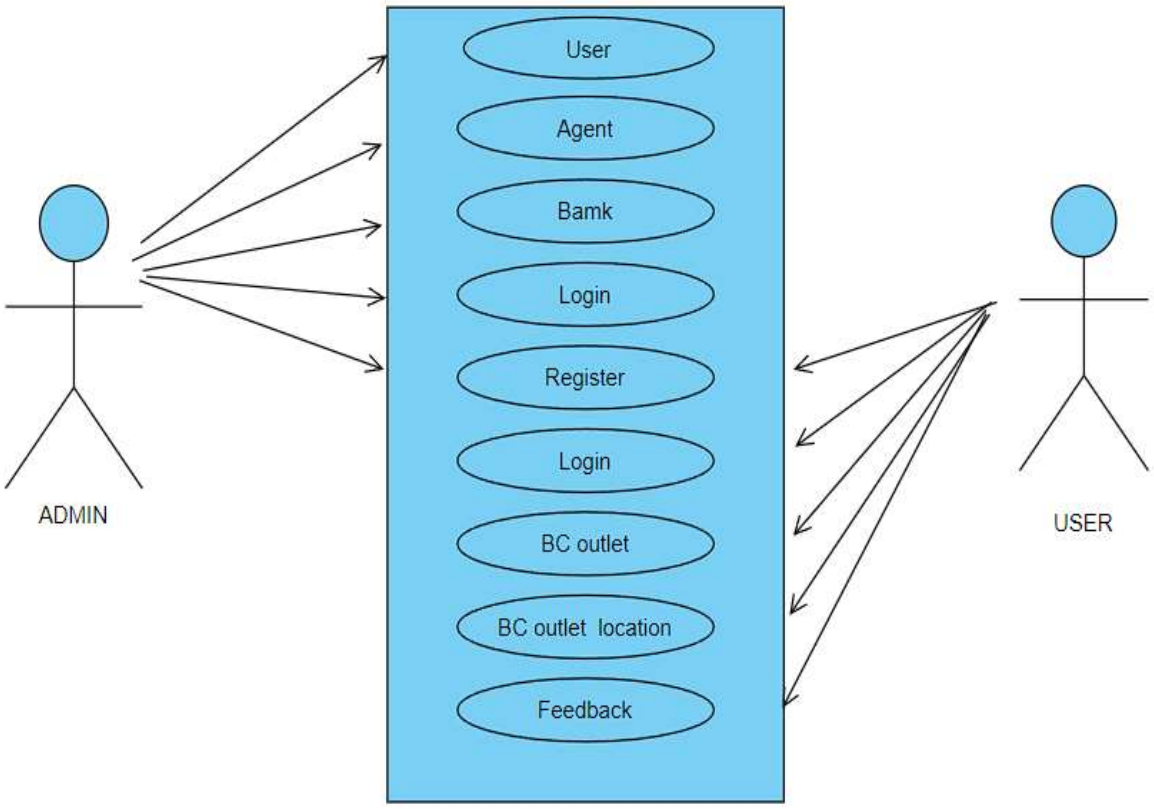
The system should be portable so as to can run in any web browser with very little or no modifications

6. Security:

All security precautions are taken to make the product more reliable, only valid i.e., registered persons can access it.

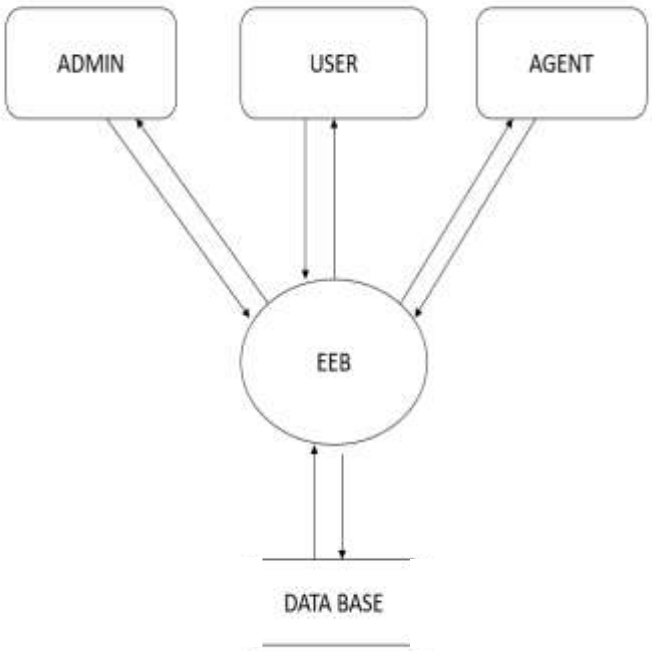
4.2 Use Case Diagram

A use case diagram is a graphic depiction of the interaction among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.



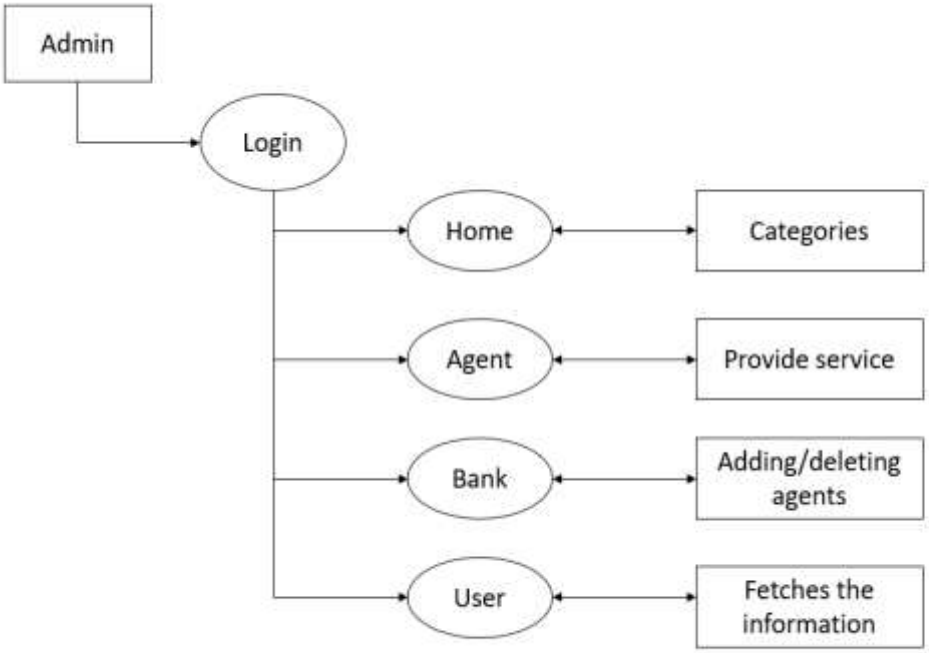
4.3 Data Flow Diagram

Understanding software documents require the logical structure analysis of diagrammatic notations, such as data flow diagrams. A Data Flow Diagram (DFD) represents the functional dependencies within a system: it shows how output values in a computation are derived from input values. DFDs show how information flows around a system, they: Represent a situation from the viewpoint of the data; Are techniques to assist analysis of processes in the system.

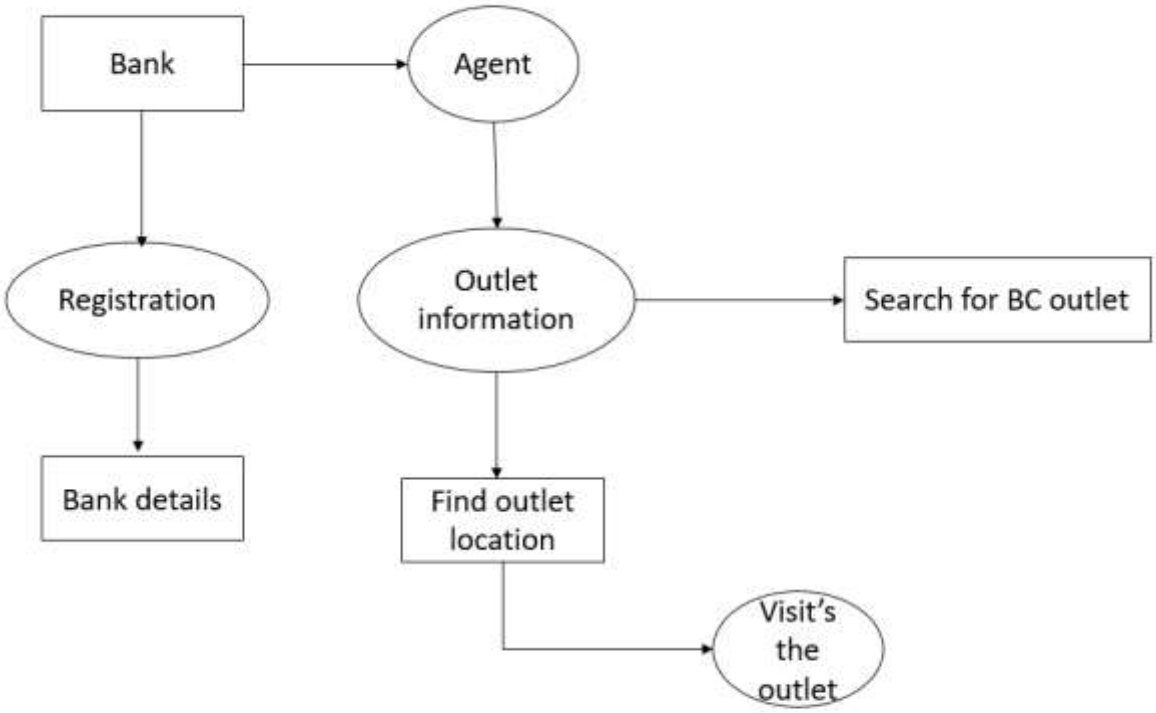


Level 1

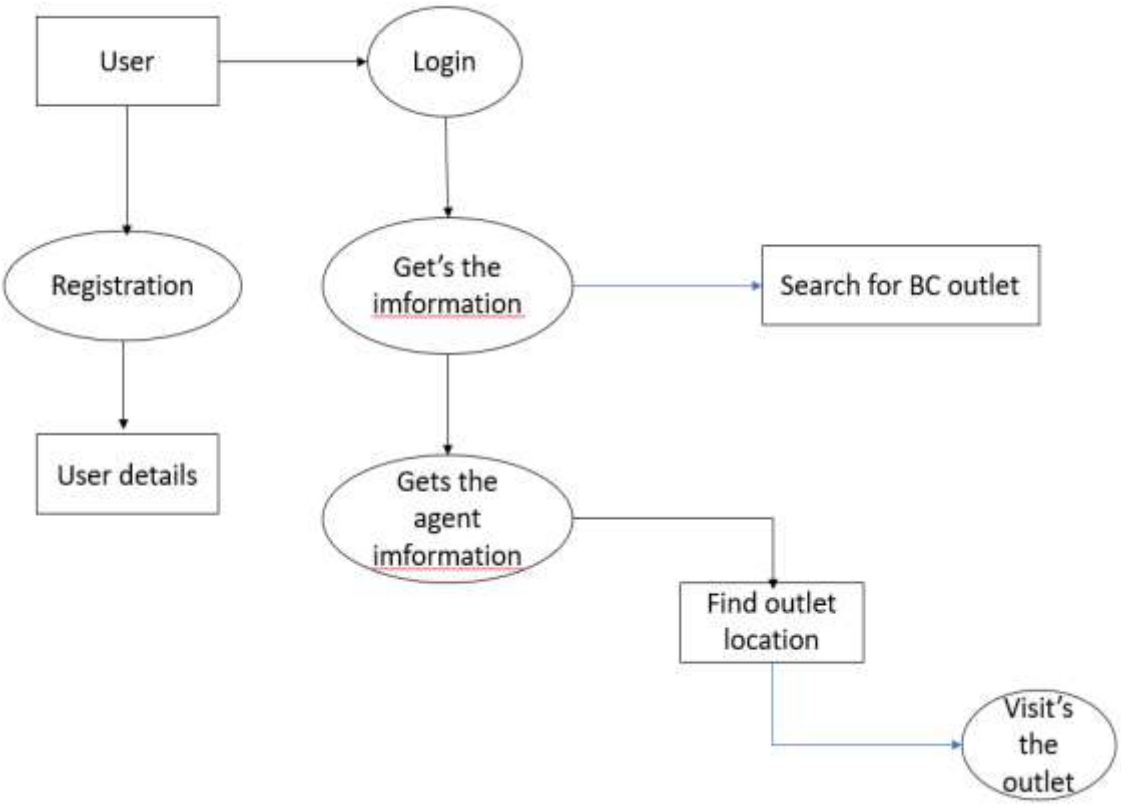
Data Flow Diagram For Admin



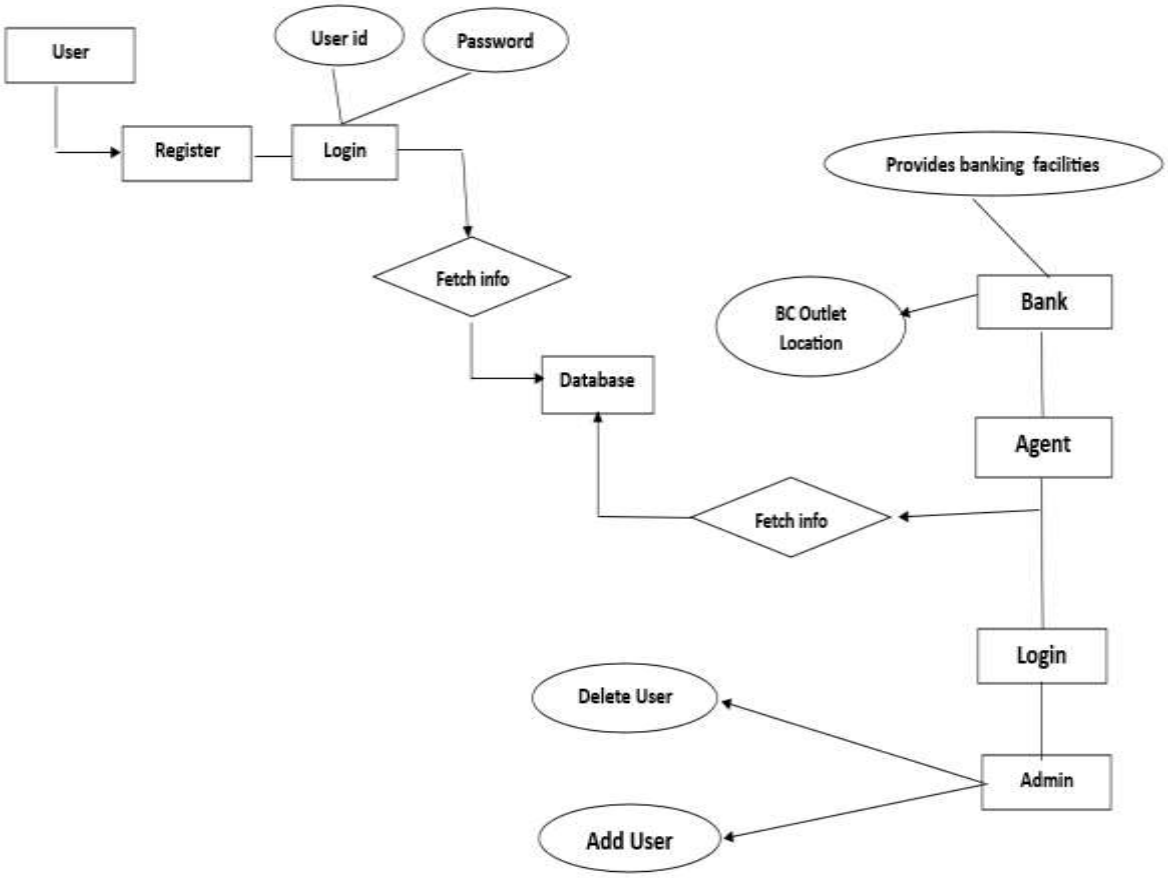
Data Flow Diagram For Bank



Data Flow Diagram For User



4.4 E-R Diagram



4.5 Data Base Table

Agent's Table:

Column name	Data type
Id	Int(100)
Agent_name	Varchar(30)
Agent_contact	Bigint(10)
Agent_address	Varchar(100)

Bank Table:

Column name	Data type
Sl_no	Int(100)
Bank_name	Varchar(30)
Branch	Varchar(30)
BC_outlet	Varchar(30)
BC_loc	Varchar(1000)
Agent_name	Varchar(30)
Agent_contact	Bigint(10)

User Table:

Column name	Data type
Sl no	Bigint(10)
Name	Varchar(30)
Email	Varchar(50)
User_contact	Bigint(10)
User_address	Varchar(100)
Bank_name	Varchar(30)
Acc_no	Bigint(20)
Password	Varchar(15)
Roll_as	Int(1)

5. SYSTEM IMPLEMENTATION

The process of putting the development system to actual use is called system implementation. It includes all activities that plays to use the new system. Once the planning has been completed, the major effort in the department is to ensure that the programs in the system are working properly. The system implementation phase follows the test phase.

The implementation phase of software development involves translation of design specification into source code by using required platform and other tools. The entire software is implemented using the php and html, databases MySQL connection.

In the implementation phase, the project reaches its finishing stage. After the development phase of the SDLC (System Development Life Cycle) is complete, the system is implemented. The software, which was designed in design phase and programmed in development phase of the SDLC, was installed on PC's that required it. The person making use of it is trained during the phase of the SDLC. Moreover, both the hardware and software are tested. The problems that we were unable to simulate were solved by the users. Thus were the main activities performed by us in the course of the project, which lead to its proper completion.

```
<?php
include("includes/header.php");
include("includes/script.php");
include("includes/configuration.php");
?>
<section>
  <div class="Container mt-5 pt-5">
    <div class="row">
      <div class="col-8 col-sm-6 col-md-4 m-auto">
        <div class="card border-3 shadow">
          <div class="card body">
                <svg class="mx-auto my-3" xmlns="http://www.w3.org/2000/svg"
width="50" height="50" fill="currentColor" class="bi bi-person-square"
viewBox="0 0 16 16">
              <path d="M11 6a3 3 0 1 1-6 0 3 3 0 0 1 6 0z" />
              <path d="M2 0a2 2 0 0 0-2 2v12a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V2a2
2 0 0 0-2-2H2zm12 1a1 1 0 0 1 1 1v12a1 1 0 0 1-1 1v-1c0-1-1-4-6-4s-6 3-6 4v1a1
1 0 0 1-1-1V2a1 1 0 0 1 1-1h12z" />
            </svg>
            <?php
            if (isset($_SESSION['status'])) {
            ?>
```

```
                <div class="alert alert-warning alert-dismissible fade show"
role="alert">
                <strong>Hey!</strong> <?php echo $_SESSION['status']; ?>
                <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
                  <span aria-hidden="true">&times;</span>
                </button>
              </div>
            <?php
              unset($_SESSION['status']);
            }
            ?>
            <form action="signupcode.php" method="post">
              <center><input type="text" name="name" id="" class="form-control
col-md-10" placeholder="Enter full name" required> </center>
             <center><input type="email" name="email" id="" class="form-control
my-4 py-2 col-md-10" placeholder="e-mail Id" required></center>
                <center><input type="text" name="contact" id="" class="form-
control my-4 py-2 col-md-10" placeholder="Contact number" required></center>
              <center><input type="text" name="addr" id="" class="form-control
my-4 py-2 col-md-10" placeholder="address" required></center>
                <center><input type="text" name="bankname" id="" class="form-
control my-4 py-2 col-md-10" placeholder="Bank name" required></center>
              <center><input type="text" name="accno" id="" class="form-control
my-4 py-2 col-md-10" placeholder="Account Number" required></center>
             <center><input type="password" name="password" id="" class="form-
control my-4 py-2 col-md-10" placeholder="Password" required></center>
                <center><input type="password" name="confirm_password" id=""
class="form-control my-4 py-2 col-md-10" placeholder="Confirm Password"
required></center>
              <div class="text-center mt-3">
                          <button type="submit" class="btn btn-primary"
name="reguser">Signup</button>
                <a href="login.php" class="nav-link">Have an Account?</a>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

```
<?php
include("includes/header.php");
include("includes/script.php");
include("includes/configuration.php");
?>
<section>
  <div class="Container mt-5 pt-5">
    <div class="row">
      <div class="col-12 col-sm-8 col-md-6 m-auto">
        <div class="card body">
              <svg  class="mx-auto  my-3"  xmlns="http://www.w3.org/2000/svg"
width="50"   height="50"   fill="currentColor"   class="bi   bi-person-square"
viewBox="0 0 16 16">
           <path d="M11 6a3 3 0 1 1-6 0 3 3 0 0 1 6 0z" />
           <path d="M2 0a2 2 0 0 0-2 2v12a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V2a2
2 0 0 0-2-2H2zm12 1a1 1 0 0 1 1 1v12a1 1 0 0 1-1 1v-1c0-1-1-4-6-4s-6 3-6 4v1a1
1 0 0 1-1-1V2a1 1 0 0 1 1-1h12z" />
         </svg>
         <?php
         if (isset($_SESSION['status'])) {
         ?>
              <div  class="alert  alert-warning  alert-dismissible  fade  show"
role="alert">
             <strong>Hey!</strong> <?php echo $_SESSION['status']; ?>
              <button type="button" class="close" data-dismiss="alert" aria-
label="Close">
               <span aria-hidden="true">&times;</span>
             </button>
           </div>
         <?php
           unset($_SESSION['status']);
         }
         ?>
         <form action="logincode.php" method="POST">
            <center><input type="text" name="email" id="" class="form-control
my-4 py-2 col-md-10" placeholder="Email"></center>
            <center><input type="password" name="password" id="" class="form-
control my-4 py-2 col-md-10" placeholder="Password"></center>
           <div class="text-center mt-3">
             <button class="btn btn-primary" name="login_btn">Login</button>
             <a href="signup.php" class="nav-link">New User?</a>
           </div>
         </form>
       </div>
     </div>
   </div>
  </div>
  </div></section>
```

```
<?php
include('authentication.php');
include("assets/includes/header.php");
include("assets/includes/script.php");
include("assets/includes/sidebar.php");
include("assets/includes/topbar.php");
?>
<!--delete case-->
<div    class="modal    fade"    id="deletebankModal"    tabindex="-1"    aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Delete bank</h5>
            <button  type="button"  class="close"  data-dismiss="modal"  aria-
label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <form action="bankcode.php" method="POST">
        <div class="modal-body">
          <input type="hidden" name="delete_bank" class="delete_bankname">
          <p>
            Are you sure you want to delete this Data?
          </p>
        </div>
        <div class="modal-footer">
                    <button    type="button"    class="    btn-secondary"    data-
dismiss="modal">Close</button>
                  <button    type="submit"    name="deletebankbtn"    class="btn-
danger">Yes,Delete</button>
        </div>
      </form>
    </div>
  </div>
</div>
<!-- Modal -->
<div     class="modal     fade"     id="addbankModal"     tabindex="-1"     aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Add Bank</h5>
            <button  type="button"  class="close"  data-dismiss="modal"  aria-
label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
```

```
      <form action="bankcode.php" method="POST">
        <div class="modal-body">
          <div class="form-group">
            <label for="">bank Name</label>
                  <input  type="text"  name="bank_name"  class="form-control"
placeholder="bank Name" required readonly>
          </div>
          <div class="form-group">
            <label for="">Branch</label>
                   <input  type="text"  name="branch"  class="form-control"
placeholder="Branch" required>
          </div>
          <div class="form-group">
            <label for="">BC OUTLET</label>
                  <input  type="text"  name="bc_outlet"  class="form-control"
placeholder="BC OUTLET" required>
          </div>
          <div class="form-group">
            <label for="">BC OUTLET LOCATION</label>
                   <input  type="text"  name="bc_loc"  class="form-control"
placeholder="BC OUTLET LOCATION" required>
          </div>
          <div class="form-group">
            <label for="">Agent Name</label>
                 <input  type="text"  name="agent_name"  class="form-control"
placeholder="Agent name" required>
          </div>
          <div class="form-group">
            <label for="">Agent Contact</label>
                <input  type="text"  name="agent_contact"  class="form-control"
placeholder="Agent Contact" required>
          </div>
        </div>
        <div class="modal-footer">
                    <button   type="button"   class="   btn-secondary"   data-
dismiss="modal">Close</button>
                     <button   type="submit"   name="addbank"   class="   btn-
primary">Save</button>
        </div>
      </form>
    </div>
  </div>
</div>
<div class="card">
  <div class="card-header">
    <h3 class="card-title">BANK DETAILS: </h3>
     <a  href="#"  data-toggle="modal"  data-target="#addbankModal"  class="btn-
secondary float-right">Add BANK</a>
```

```
  </div>
  <table class="table">
    <thead>
      <tr>
        <th scope="col">Sl_no</th>
        <th scope="col">Bank Name</th>
        <th scope="col">Branch</th>
        <th scope="col">bc-Outlet</th>
        <th scope="col">bc-Outlet-Location</th>
        <th scope="col">Agent Name</th>
        <th scope="col">Agent Contact</th>
        <th scope="col">Changes</th>
      </tr>
    </thead>
    <tbody>
      <?php
      $query = "SELECT * FROM bank";
      $query_run = mysqli_query($conn, $query);
      if (mysqli_num_rows($query_run) > 0) {
        foreach ($query_run as $row) {
      ?>
          <tr>
            <td><?php echo $row['sl_no']; ?></td>
            <td><?php echo $row['bank_name']; ?></td>
            <td><?php echo $row['branch']; ?></td>
            <td><?php echo $row['bc_outlet']; ?></td>
            <td><?php echo $row['bc_loc']; ?></td>
            <td><?php echo $row['agent_name']; ?></td>
            <td><?php echo $row['agent_contact']; ?></td>
            <td>
                 <a href="bank-edit.php?branch=<?php echo $row['branch']; ?>"
class=" btn-sm">Edit</a><button type="button" value="<?php echo $row['branch'];
?>" class="btn-sm deletebtn">Delete</button>
          </tr>
        <?php
        }
      } else {
        ?>
        <tr>
          <td>no record found</td>
        </tr>
      <?php
      }
      ?>
    </tbody>
  </table>
</div>
</div>
```

```
<script>
  $(document).ready(function() {
    $('.deletebtn').click(function(e) {
      e.preventDefault();
      var branch = $(this).val();
      //console.log(suitno);
      $('.delete_bankname').val(branch);
      $('#deletebankModal').modal('show')
    });
  });
</script>
```

6. SYSTEM TESTING

Systems should not be tested as a single, monolithic unit. The testing process should therefore proceed in stages where testing is carried out incrementally in conjunction with system implementation. Errors in program components may come to light at a later stage of the testing process. The process is therefore an iterative one with information being fed-back from later stages to carlier parts of the process. The various strategies that were used in testing this software were as follows:

6.1 Unit Testing

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

6.2 Integration Testing

Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

6.3 System Testing

System Testing is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System Testing is a level of the software testing process where complete integrated system/software is tested. The purpose of this test is to evaluate the system compliance with the specified requirements.

6.3.1 White Box Testing

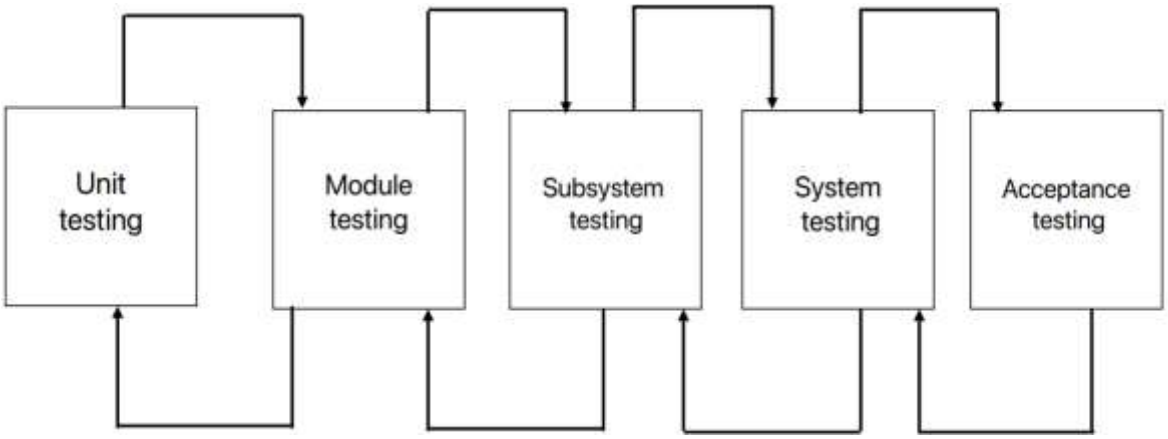
It is a software testing method in which the internal structure /design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions. Also known as structural testing and Glass box Testing.

6.3.2 Black Box Testing

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

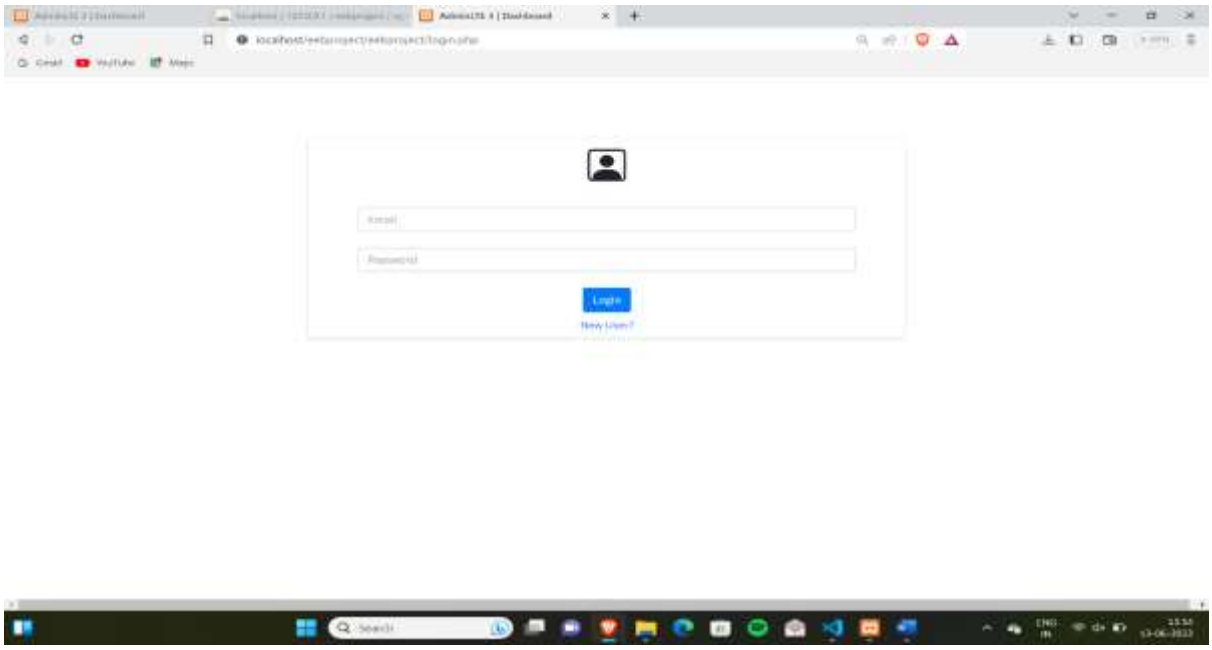
6.4 Acceptance Testing

Acceptance testing is a software testing technique used to determine whether a system or software application meets the required business specifications and is ready for deployment. It is typically performed by end-users, stakeholders, or independent testers to validate that the software meets the predefined acceptance criteria. The primary goal of acceptance testing is to ensure that the software satisfies the needs of the end-users and functions as expected in its intended environment. It focuses on evaluating the overall system's compliance with business requirements, user expectations, and any applicable industry standards or regulations.

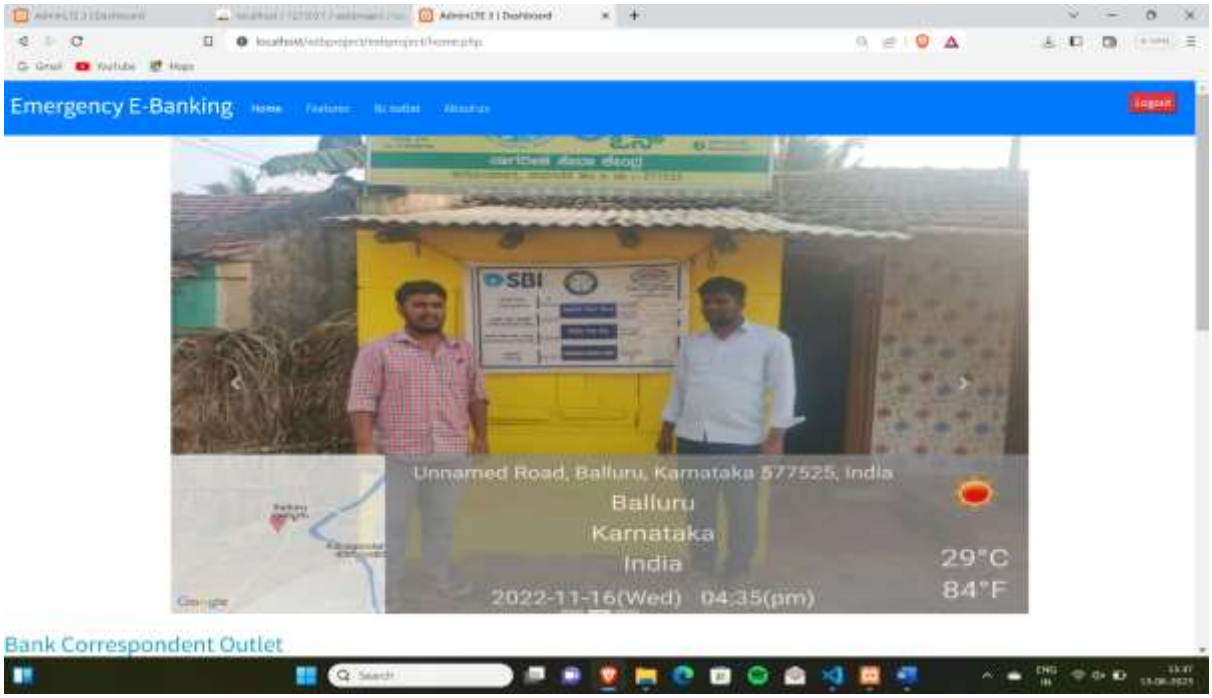


7. SNAPSHOTS

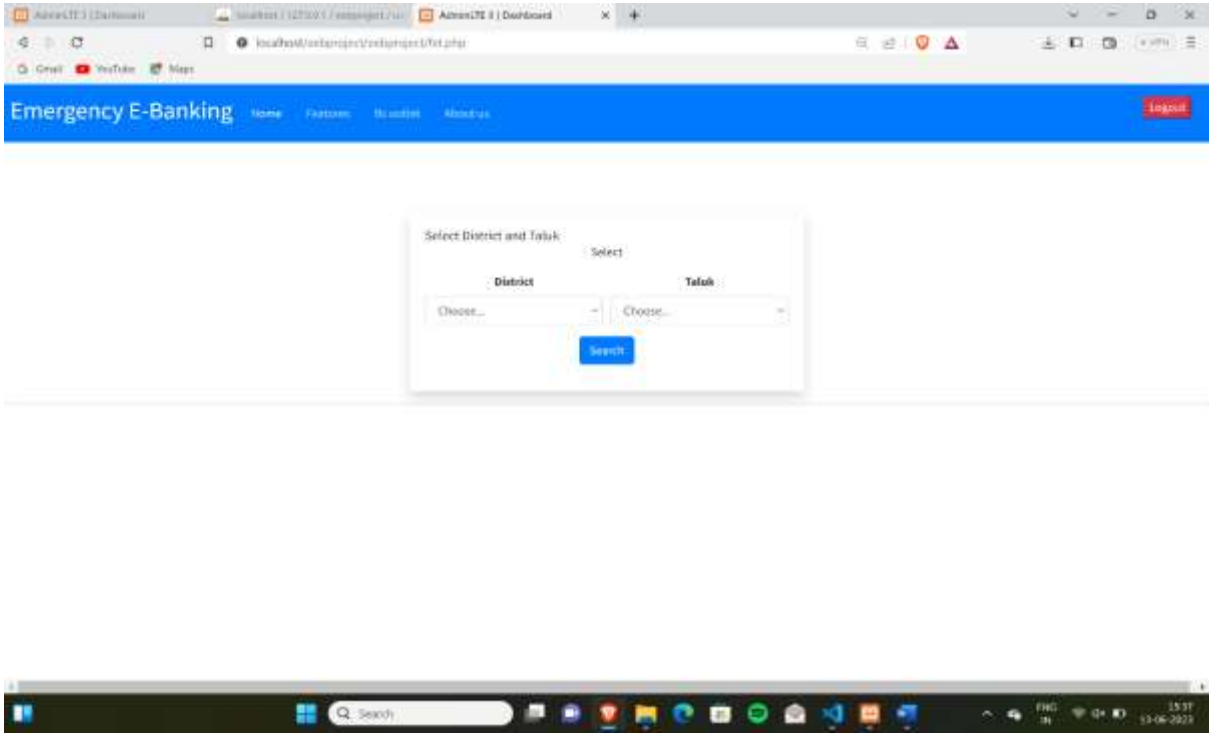
Login Page:

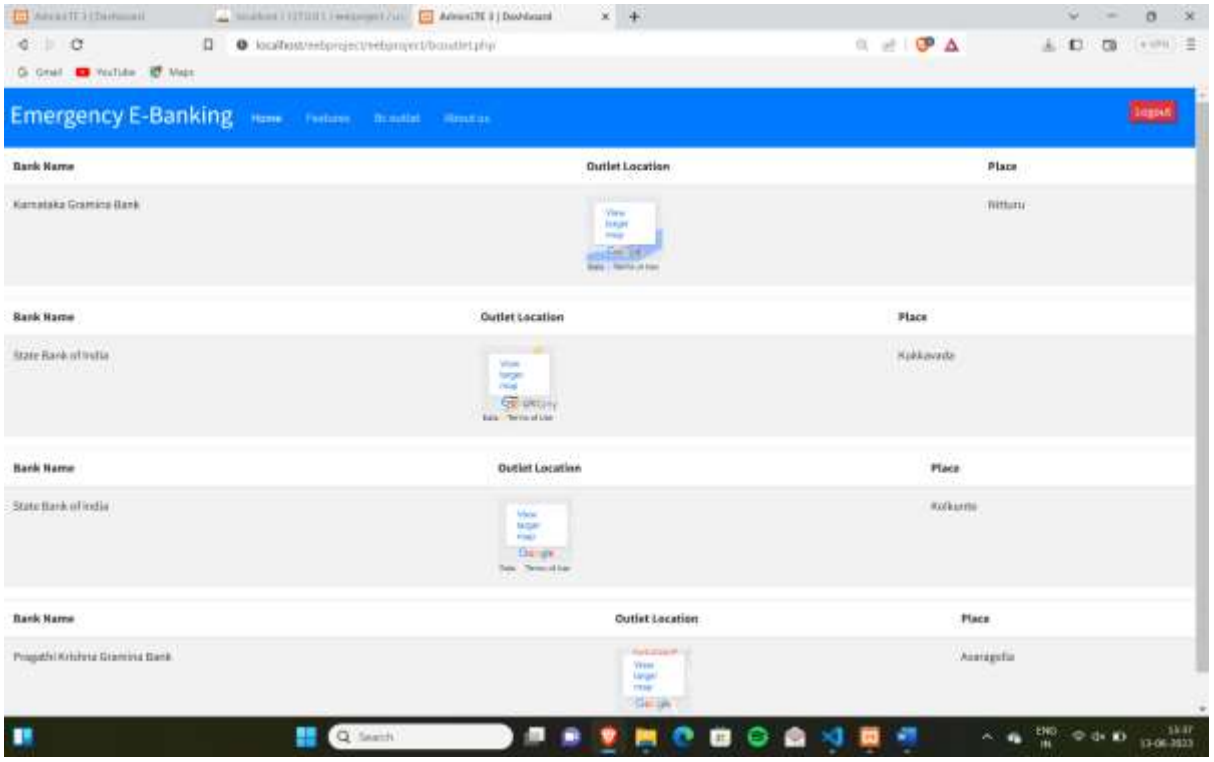


User Home Page:

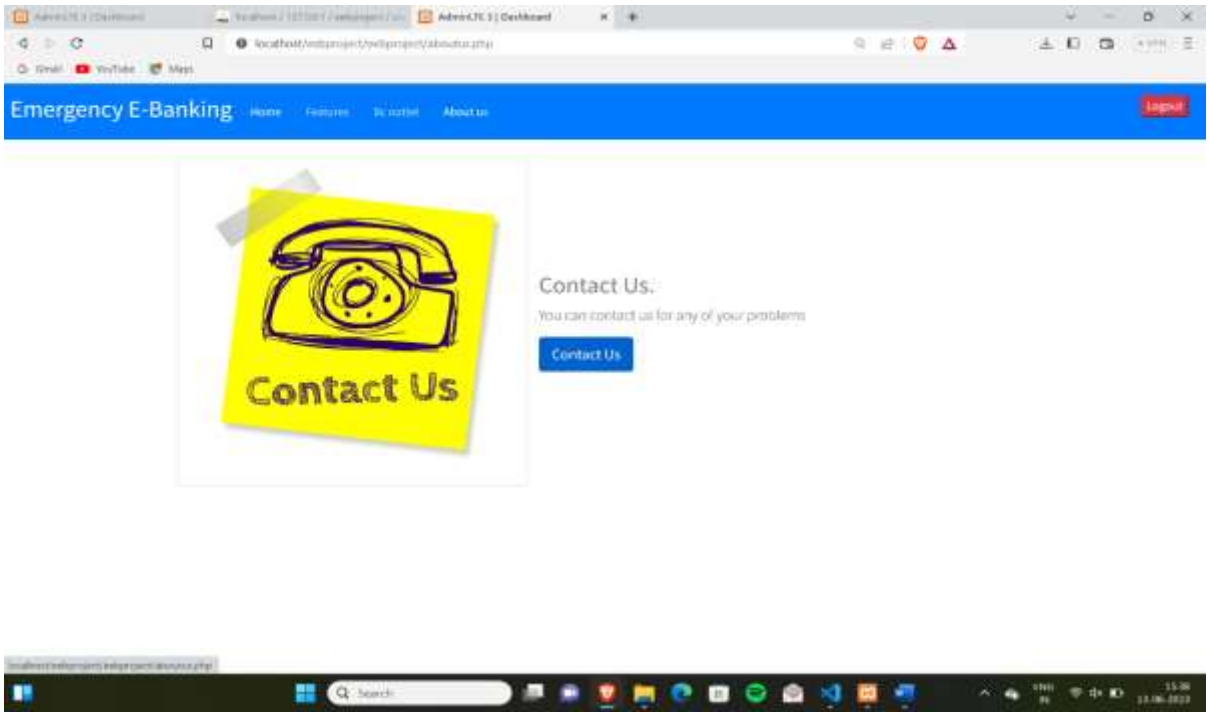


User Feature Page:

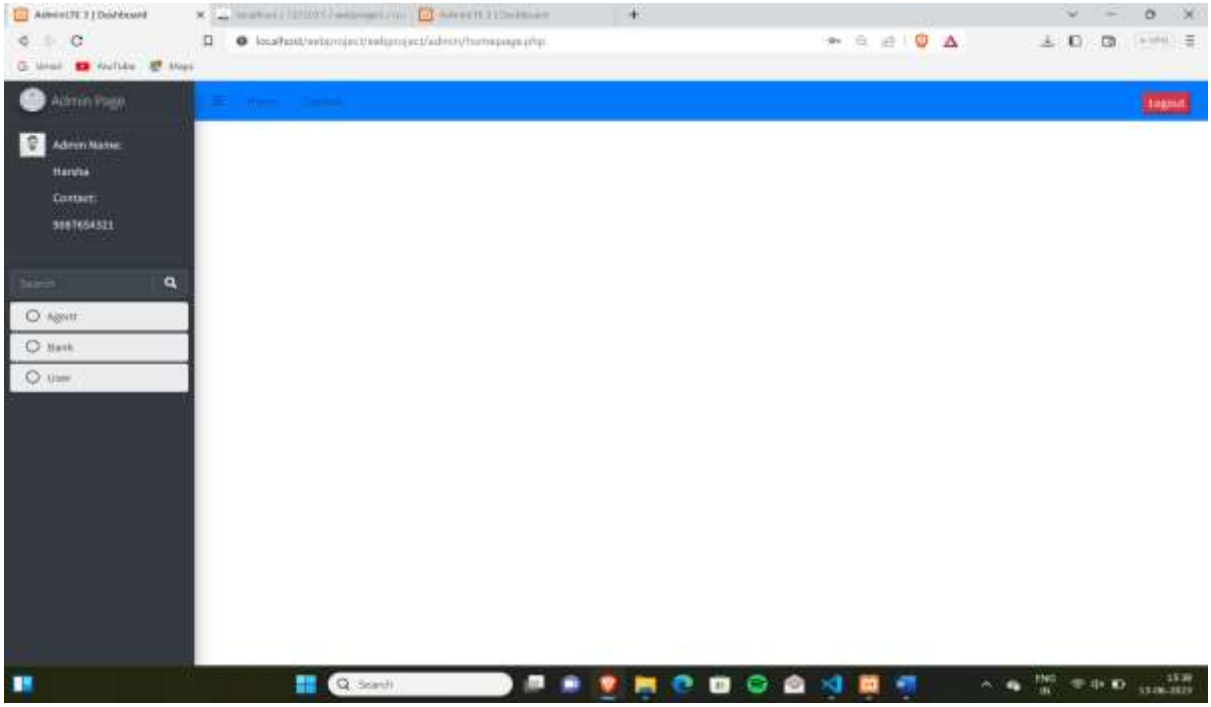


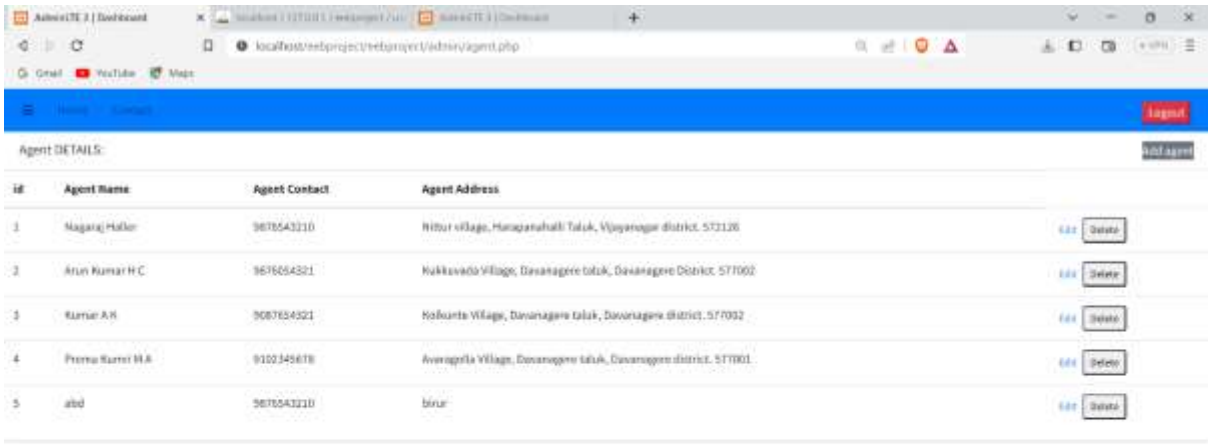


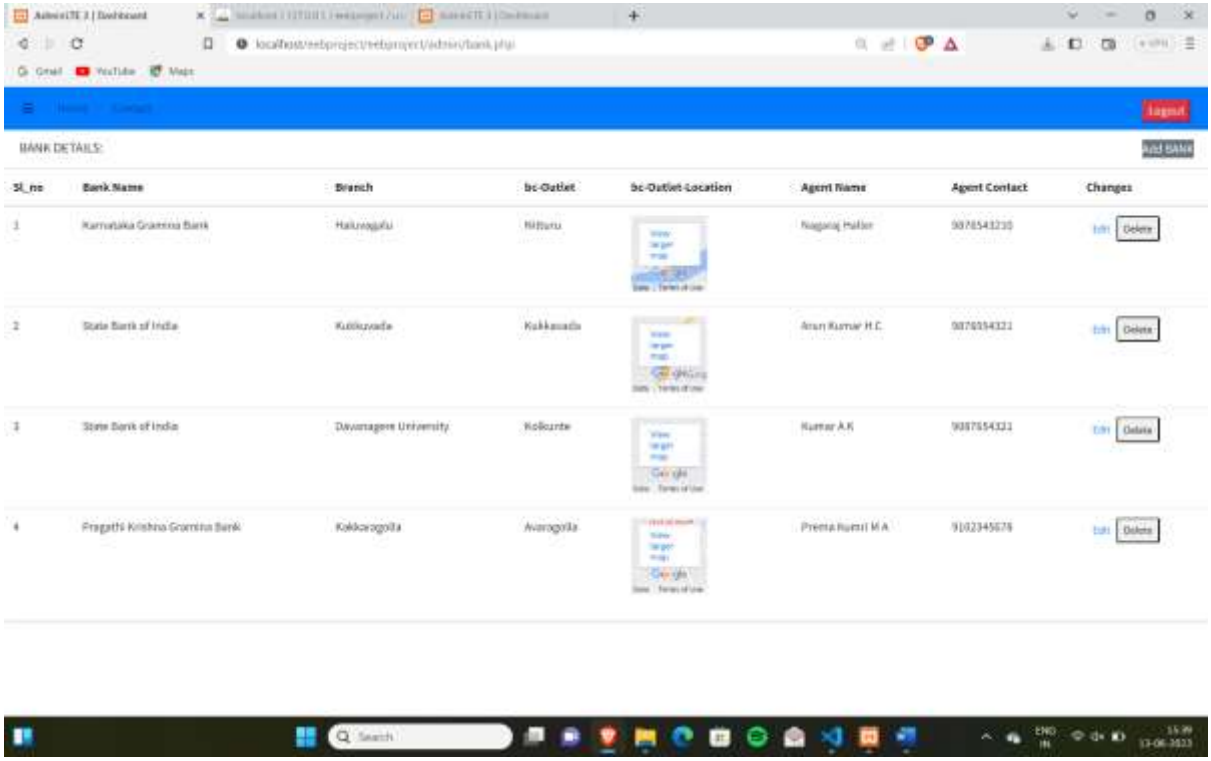
User Contact Us(admin) Page:

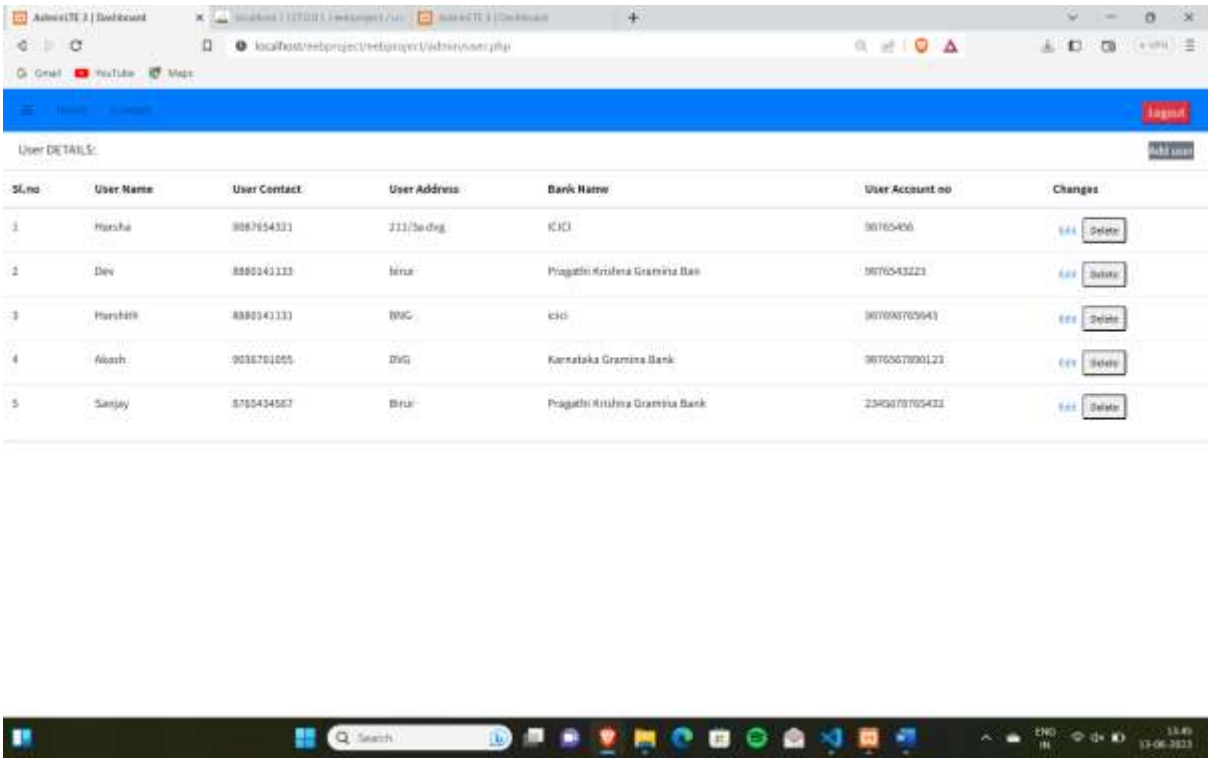


Admin Page:


Admin Page
Admin Name:
Harsha
Contact:
9087654321
Agent
Bank
User
Logout


Agent DETAILS
Logout
id
Agent Name
Agent Contact
Agent Address


BANK DETAILS:
Sl_no
Bank Name
Branch
bc-Outlet
bc-Outlet-Location
Agent Name
Agent Contact
Changes
Karnataka Gramina Bank
Halluvagilu
Nitturu
State Bank of India
Kukkuvada
Kukkavada
State Bank of India
Davanagere University
Kolkunte
Pragathi Krishna Gramina Bank
Kakkaragolla
Avaragolla


User DETAILS:
Sl.no
User Name
User Contact
User Address
Bank Name
User Account no
Changes
Harsha
Dev
Harshith
Akash
Sanjay
Pragathi Krishna Gramina Bank

8. CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The project is to design and implement an online platform to find the nearest BC outlet to get the banking facility . Main idea of implementing of this project is to develop a website where user can search and view the information of the nearest BC outlet and also get it's map location to reach it in a easier way. Emergency Electronic Banking (EEB) is a user friendly graphical user interface through which user can search for the nearest BC outlet retrieved from our database.

The "Emergency Electronic Banking (EEB)" seeks to empower individuals and businesses with easy access to banking services by providing accurate location and information about BC agents. By leveraging technology and establishing a user-friendly platform, the project aims to enhance financial inclusion, bridge the gap between banks and underserved communities, and promote economic growth.

8.2 FUTURE SCOPE:

The basic idea of this Emergency Electronic Banking (EEB) can be enhanced by implementing following advance features:

- . Expansion to New Regions: As the project gains traction and proves its effectiveness, there is potential to expand its coverage to new regions and areas. This would further extend financial inclusion and accessibility to banking services for underserved communities.

- . Integration with More Financial Institutions: Currently, the project focuses on BC agents associated with specific banks or financial institutions. In the future, there is an opportunity to collaborate with a wider range of financial institutions, including cooperative banks, microfinance institutions, and other organizations, to include their BC agents in the locator platform.

- . Partnerships and Collaborations: The project can explore partnerships with government agencies, non-profit organizations, and technology providers to leverage their expertise, resources, and networks. Collaborations with these entities can amplify the impact of the project, increase its reach, and foster innovation in the financial inclusion domain.

- Integration with Digital Payment Systems: With the increasing adoption of digital payment systems, there is an opportunity to integrate the BC Agent Locator platform with digital wallets, mobile banking apps, and other fintech solutions. This integration would enable users to seamlessly access BC agents' services, make digital transactions, and leverage the convenience of digital payments.

- Financial Education and Empowerment: The project can expand its focus beyond providing location and information about BC agents to include financial education and empowerment initiatives. By offering educational resources, tips, and tools through the platform, users can enhance their financial literacy and make informed decisions regarding their finances.

- Policy Advocacy: As the project generates insights and evidence about the impact of BC agents on financial inclusion, it can engage in policy advocacy efforts. By collaborating with policymakers and regulators, the project can contribute to the development of supportive policies, regulations, and frameworks that promote the growth and effectiveness of BC agent networks.

BIBLIOGRAPHY

Referred Books:

- Beginning PHP and MySQL-W Jason Gilmore Publication, Fourth Edition, 2010.
- PHP and MySQL 24 Hour Trainer-Andrea Tarr Publication, Second Edition, 2011.
- Expert PHP and MySQL-Andrew Curioso, Ronald Bradford, Patrick Galbraith Publication,Fourth Edition 2010.

Referred Websites :

- https://www.w3schools.com
- http://www.tuxradar.com/practicalphp
- http://phpbuddy.com/index.php
- http://www.php.net