CHAPTER 1

INTRODUCTION

1.1 Introduction to DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by possibly many concurrent users. Databases are generally separated into application areas. For example, one database may contain Human Resource (employee and payroll) data; another may contain sales data; another may contain accounting data; and so on. Databases are managed by a DBMS. Many Database Systems are being used which are in turn managed by many other Database Management Systems. A Database Management System (DBMS) is a set of programs that manages any number of databases. Basically, DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

1.2 Brief Outline of Project

Insurance Policy data management system is a web based project which is developed for tracking the details of the insurance policy, customer details and company details. This series of web pages is an online insurance analysis and information management system that provides easy access of information regarding the people and resources of insurance. User can view their own personal details when login into the Policy Holder module. This project is useful for any kind of insurance company to manage the insurance details, to sanction the insurance for customer, process the insurance policy details and all kind of insurance process through online. The Insurance management system is a complete solution for organizations, which need to manage insurance for their vehicles, equipment, buildings, and other resources. This insurance management system can efficiently manage the company, records, provides instant access and one that improves the productivity. It will show details about insurance and

its types, also it will show the details about different duration schemes to the corresponding insurance type or 2 insurance policy. The main objective of the developed system is to allow admin users to register insured persons with their name, date of birth, residence address, medical history and also policy details.

1.3 Project Goal

In the proposed Life Insurance Management System, all the work will be digitalized and is done via computers and internet. All the details regarding the insurance holder and schemes will be added via computer and the information data is being saved in servers. Backup should be there in case if by chance any of the information will be lost. Time consume will be reduced and users will get any easy way to access their insurance related information and new upcoming schemes. Users just have to click on the button and just have to wait for some moments and they get an easy access to their information. The proposed system is for making easier to manage policy holder details, agent details, policy details, claimant details and payment details. The proposed system is designed to eliminate the drawbacks of the existing system. It is designed by keeping to eliminate the drawbacks of the present system in order to provide a permanent solution to the problems. The primary aim of the new system is to speedup transactions. This insurance management system will be developed for 7 managing the insurance management system. The overall system is control through the main menu.

1.4 Scope

The scope of our project 'LIC Management System' is like any other conventional management system i.e. we can store the details for the employees working in the company, customers of the company and also check the details of the policies registered. The user can also view a detailed policy data view. Our project can be implemented in daily life since mail is commonly used.

CHAPTER 2

REQUIREMENT SPECIFICATION

Some of the basic requirements for the development of this project are as follows:

2.1 Hardware Requirements

Processor	:	intel core i5
Hard Disk	:	20 GB, 80 GB, 160 GB or above
Monitor	:	15 VGA colour, 1366*768 resolution
RAM	:	2GB or above
Input Device	:	Keyboard and Mouse

2.2 Software Requirements

Operating System	:	Windows 10
Tool used	:	XAMPP
Front end used	:	HTML/CSS/PHP
Back end used	:	SQL

CHAPTER 3

PROBLEM DESCRIPTION

LIC MANAGEMENT SYSTEM

The problem tackled in the project is to handle the policy data using database management system. This project would focus on both front-end as well as backend for systematic working. Data input would be given from the front-end by users. The front-end would be a HTML form.

- Relation between client and his policies is a one to many relationship, but policy type to clients is a many to many relationship.
- Data would be handled at the back-end using different tables and relations using MySQL.
- A policy taken by a client has attributes like premium, sum assured, date of commencement, etc.
- A client has attributes including personal details as well as details about the policy he/she has taken.
- A policy type contains attributes describing the type of policies like premium based on the mode, risk cover.
- There would be many other tables where records of policies taken by different clients would be present depending on its status like active, lapsed, etc. The developed system should allow admin users to register insured persons with their name, date of birth, residence address, medical history and also policy details. After registering all the insured persons, website should provide management facilities like delete unwanted persons' data. And also should provide awareness to the visitors about micro insurance through articles.

The tables are as follows

Table 3.1 - AGENT

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Agent_code	VARCHAR(10)	PRIMARY KEY	Code of agent
Agent_name	VARCHAR(150)	NOT NULL	Name of agent

DOB	DATE	NOT NULL	Date of birth
Address	VARCHAR(80)	NOT NULL	Adress of agent
Pincode	INT(6)	NOT NULL	Pincode of agent
Branch	VARCHAR(50)	NOT NULL	Agent branch
Contact_Num	BIGINT(10)	NOT NULL	Phone number

Table 3.2 - CUSTOMER

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Customer_Num	BIGINT(10)	PRIMARY KEY	Customer number
First_Name	VARCHAR(50)	NULL	First name
Middle_Name	VARCHAR(50)	NULL	Middle name
Last_Name	VARCHAR(50)	NULL	Last name
Gender	CHAR(1)	NULL	Gender of customer
DOB	DATE	NULL	Date of birth
Address	VARCHAR(70)	NULL	Address of customer
Pincode	INT(6)	NULL	Pincode of customer
Contact_Num	BIGINT(10)	NULL	Contact number
Mother_Name	VARCHAR(150)	NULL	Mother name
Mother_Status	VARCHAR(10)	NULL	Mother-status whether alive or dead
Father_Name	VARCHAR(150)	NULL	Father name
Father_Status	VARCHAR(10)	NULL	Father-status whether alive or dead
Marital_Status	CHAR(1)	NULL	Marital status whether married or single
Spouse	VARCHAR(150)	NOT NULL	Spouse name

Table 3.3 - POLICY

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Policy_Num	INT(15)	PRIMARY KEY	Policy number
Customer_Num	BIGINT(10)	NOT NULL	Customer number
Agent_code	VARCHAR(10)	NOT NULL	Code of agent
DOC	DATE	NOT NULL	Date of contact
Product	VARCHAR(50)	NOT NULL	Product name
Sum_Assured	INT(10)	NOT NULL	Total sum assured
Pay_Period	INT(2)	NOT NULL	Pay period
Ins_Period	INT(2)	NOT NULL	Insurance period

Table 3.4 - PREMIUM

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Policy_Num	INT(15)	PRIMARY KEY	Policy number
Premium	INT(10)	NOT NULL	Premium method
Mode	VARCHAR(3)	NOT NULL	Mode of payment
Last_date	DATE	NOT NULL	Last date of payment

Table 3.5 - PAID PREMIUM

COLUMN NAME	DTATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Receipt-Num	INT(20)	PRIMARY KEY	Receipt number

Reciept_Date	DATE	NOT NULL	Receipt date
Policy_Num	INT(15)	NOT NULL	Policy number

Table 3.6 -UNPAID PREMIUM

COLUMN NAME	DTATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Policy_Num	INT(15)	PRIMARY KEY	Policy number
Lateness	INT(10)	NOT NULL	Lateness of payment
Fine	INT(11)	NOT NULL	Fine for lateness

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

An Entity Relationship Diagram is a data modelling technique that graphically illustrates an information system entity and the relationships between those entities.

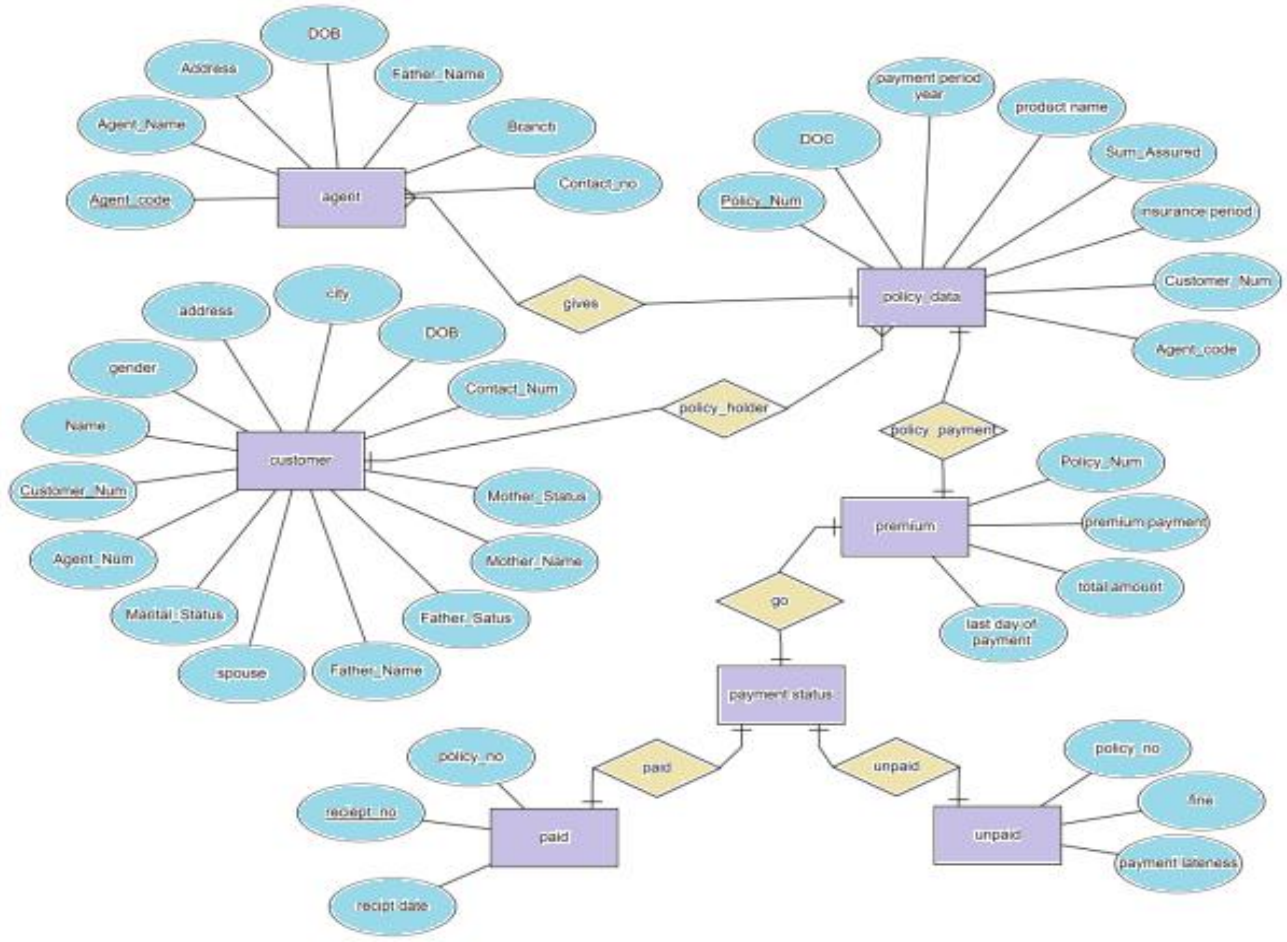

Fig 4.1: ER diagram

The Fig. 4.1, is the Entity - Relationship diagram which has been used to create our Database.

4.2 Schema Diagram

A database is a skeleton structure that represents the logical view of the entire database.

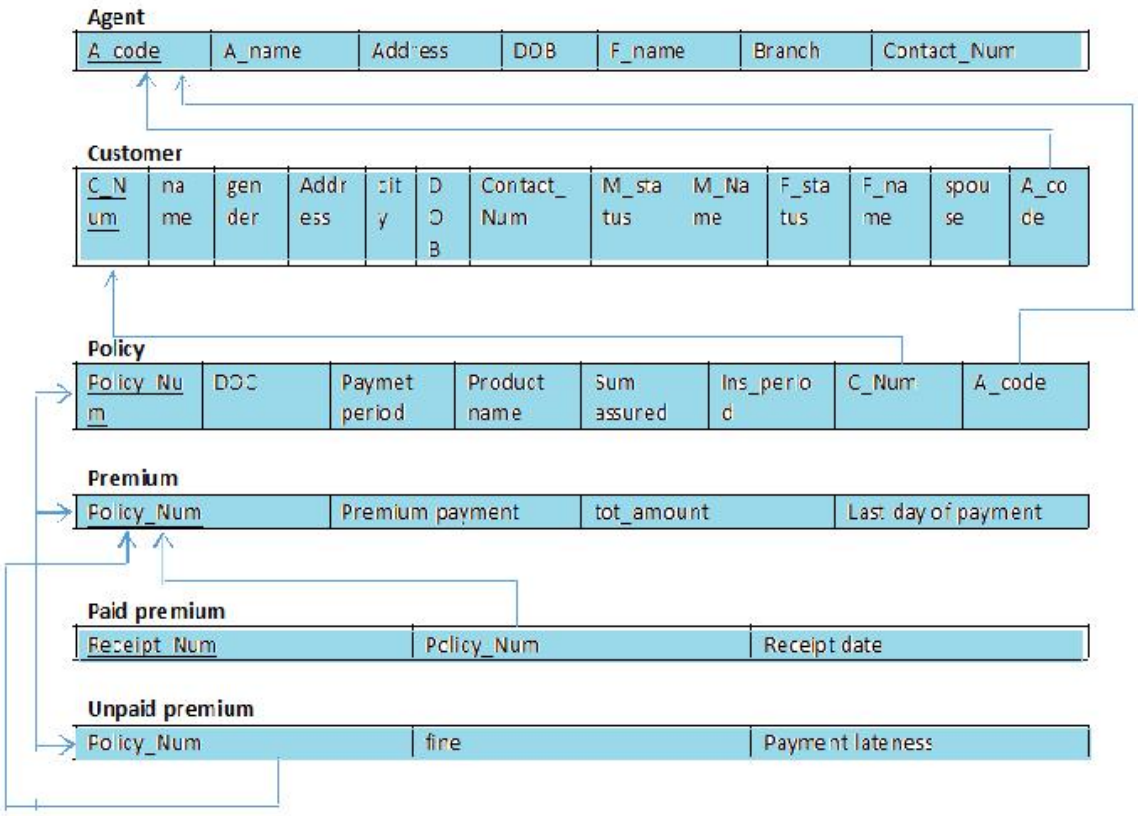

Fig. 4.2: Schema diagram

The Fig. 4.2, is the Schema diagram which has been used to create our Database.

4.3 Normal Form

4.3.1 First Normal Form

A Relation is in first normal form if and only if the following conditions are satisfied,

1. Contains only atomic values.
2. There no repeating groups.

4.3.2 Second Normal Form

A Relation schema R is in second normal form if and only if the following conditions are satisfied,

1. R is in 2NF.
2. All non-key attributes are fully functionally dependent on primary key.

4.3.3 Third Normal Form

A Relation schema R is in third normal form if and only if the following conditions are satisfied,

1. R is in 2NF.
2. There are no transitive functional dependencies.

All the tables in LIC MANAGEMENT SYSTEM satisfy all the three normal form

CHAPTER 5

IMPLEMENTATION

5.1 Introduction to software used

PHPMYADMIN is used for implementing HTML,CSS and tomcat Apache server. MYSQL server is used for the database and MYSQL workbench is used to manage the server and create the database.

5.1.1 PHP

PHP is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML.PHP scripts can only be interpreted on a server that as PHP installed .

5.1.2 MYSQL

MYSQL is an open source relational database management system .The application is used for a wide range of purpose, including data warehousing ,e-commerce

SQL supports a number of client and utility programs ,command-line programs and administration tools such as MYSQL Workbench.

5.2 Source code

Connection.php

```
<?php
        $servername = "localhost";
        $username = "root";
        $password = "";
```

```
$conn = mysqli_connect($servername , $username , $password,"test") or die("unable to connect to host");
?>
```

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>LIC</title>
    <link rel="stylesheet" type="text/css" href="index/css/style1.css">
</head>
<body>
<header>
    <div class="main">

        <ul>
            <li class="active"> <a href="agent/agent.php"> Agent regstration </a></li>
            <li> <a href="client/client.php"> Customer regstration </a></li>
            <li> <a href="policy/policy.php"> Policy regstration </a></li>
            <li> <a href="premium/premium.php"> Premium regstration </a></li>
            <li> <a href="agent/modified1.php"> Agent data </a></li>
            <li> <a href="client/modified1.php"> Customer data </a></li>
            <li> <a href="policy/modified1.php"> Policy data </a></li>
            <li> <a href="premium/modified1.php"> Premium data </a></li>
        </ul>
    </div>
    <div class="title">
        <h1> LIFE INSURANCE CORPORATION </h1>
```

```
            </div>
            <div class="button">
                <a href="#" class="btn"> Login </a>
            </div>
        </header>
        </body>
        </html>
```

agent/input.php

```
<?php
            include "../connection.php";
            $ac=$_POST['Agent_code'];
            $an=$_POST['Agent_Name'];
            $d=$_POST['DOB'];
            $a=$_POST['Address'];
            $p=$_POST['Pincode'];
            $con=$_POST['Contact_Number'];
            $br=$_POST['Branch'];
            $query="insert into agent(Agent_code,Agent_name,DOB, Address, Pincode,
Branch, Contact_Num) values('$ac','$an','$d','$a',$p,'$br',$con)";
            mysqli_query($conn,$query) or die($query."Can't Connect to Query...");
?>
```

agent.php

```
<html>
  <head>
    <title>Registration Form</title>
  </head>
  <body>
    <link href = "reg.css" type = "text/css" rel = "stylesheet" />
            <link href = "../style3.css" type = "text/css" rel = "stylesheet" />
```

```
<ul>
<li style="float:right;"><a href="../index.html">Back to homepage</a></li>
</ul>
<h2>Agent</h2>
<form name="form1" action='modified.php' method = 'POST' enctype = "multipart/form-data" >
<div class = "container">
<div class = "form_group">
<label>Agent Code:</label>
<input type = "text" name = "Agent_code" id="button" required pattern="[0-9]{3}[A-Z a-z]{3}[0-9]{3}" placeholder="enter agent code"/>
</div>
<div class = "form_group">
<label>Name:</label>
<input type = "text" name = "Agent_Name" id="button" value = "" required placeholder="enter agent Name"/>
</div>
<div class = "form_group">
<label>Date of Birth:  </label><input type = "date" name = "DOB" id="button" value = "" required placeholder="DOB" />
</div>
<div class = "form_group">
<label>Address:</label>
<input type = "text" name = "Address" id="button" value = "" required placeholder="enter agent Address"/>
</div>
<div class = "form_group">
<label>Pincode: </label>
<input type = "text" name = "Pincode" id="button" value = ""  required placeholder="enter Pincode"/>
</div>
<div class = "form_group">
```

```
            <label>Branch: </label>
            <input type = "text" name = "Branch" id="button" value = ""  required placeholder="enter Branch"/>
        </div>

                        <div class = "form_group">
            <label>Contact Number: </label>
            <input type = "text" name = "Contact_Number" id="button" value = ""  required pattern="[0-9]{10}" placeholder="agent phone no" />
        </div>
                        <div class = "form">
            <input type = "submit" id="butt" value = "submit"/>
        </div>
                        <div class = "form">
            <input type = "reset" id="butt" value = "reset"/>
        </div>
      </div>
    </form>
  </body>
</html>
```

agent/delete.php

```
<?php
include "../connection.php";
if(isset($_GET['id'])){
$sql = "delete from agent where Agent_code = '".$_GET['id']."'";
$result = mysqli_query($conn,$sql);
}
header('Location:modified1.php');
?>
```

agent/modified.php

```
<?php
include "input.php";

$sql = "select * from agent";
$result = mysqli_query($conn,$sql);
?>
<html>
   <body>
      <link href = "../style.css" type = "text/css" rel = "stylesheet" />
                  <link href = "registration.css" type = "text/css" rel = "stylesheet" />
                  <table width = "100%" border = "1" cellspacing = "1" cellpadding = "1">
         <tr>
            <td>Agent Code</td>
            <td>Agent Name</td>
            <td>DOB</td>
            <td>Address</td>
            <td>Pincode</td>
            <td>Branch</td>
                                    <td>Contact Number</td>
            <td colspan = "2">Action</td>
         </tr>
         <?php

                  while($row = mysqli_fetch_object($result)){



         ?>
                        <tr>
                              <td>
                                    <?php echo $row->Agent_code;?>
```

```
                    </td>
                    <td>
                        <?php echo $row->Agent_name;?>
                    </td>
                    <td>
                        <?php echo $row->DOB;?>
                    </td>
                    <td>
                        <?php echo $row->Address;?>
                    </td>
                    <td>
                        <?php echo $row->Pincode;?>
                    </td>
                    <td>
                        <?php echo $row->Branch;?>
                    </td>
                    <td>
                        <?php echo $row->Contact_Num;?>
                    </td>
                    <td> <a href="delete.php?id =
                        <?php echo $row->Agent_code;?>" onclick="return
confirm('Are You Sure')">Delete
                    </a>
                    </td>
                </tr>
            <?php } ?>
    </table>
<?php header('Location: modified1.php')?>;
  </body>
</html>
```

agent/modified1.php

```
<?php
include "../connection.php";



$sql = "select * from agent";
$result = mysqli_query($conn,$sql);
?>
<html>
   <body>
                <link href = "reg.css" type = "text/css" rel = "stylesheet" />
            <link href = "../style1.css" type = "text/css" rel = "stylesheet" />
                <ul>
                        <li style="float:right;"><a href="../index.html"> Back to
homepage</a></li>
                </ul>
                <h1><center>Agents Data</center></h1>


                <table width = "100%" border = "1" cellspacing = "1" cellpadding = "1">
         <tr>
            <td>Agent Code</td>
            <td>Agent Name</td>
            <td>DOB</td>
            <td>Address</td>
            <td>Pincode</td>
            <td>Branch</td>
                                <td>Contact Number</td>
            <td colspan = "2">Action</td>
         </tr>
         <?php
```

```
while($row = mysqli_fetch_object($result)){

?>
        <tr>
            <td>
                <?php echo $row->Agent_code;?>
            </td>
            <td>
                <?php echo $row->Agent_name;?>
            </td>
            <td>
                <?php echo $row->DOB;?>
            </td>
            <td>
                <?php echo $row->Address;?>
            </td>
            <td>
                <?php echo $row->Pincode;?>
            </td>
            <td>
                <?php echo $row->Branch;?>
            </td>
            <td>
                <?php echo $row->Contact_Num;?>
            </td>
            <td> <a href="delete.php?id=<?php echo $row->Agent_code;?>" onclick="return confirm('Are You Sure')">Delete
            </a>
```

```
                    </td>
                </tr>
            <?php } ?>
        </table>
    </body>
</html>
```

input/client.php

```
<?php
        include "../connection.php";
        $fn=$_POST['First_Name'];
        $mn=$_POST['Middle_Name'];
        $d=$_POST['DOB'];
        $ln=$_POST['Last_Name'];
        $g=$_POST['Gender'];
        $a=$_POST['Address'];
        $p=$_POST['Pincode'];
        $con=$_POST['Contact_Number'];
        $mon=$_POST['Mother_Name'];
        $mos=$_POST['Mother_Status'];
        $fan=$_POST['Father_Name'];
        $fas=$_POST['Father_Status'];
        $ms=$_POST['Marital_Status'];
        $s=$_POST['Spouse'];
        $query="insert into
customer(First_Name,Middle_Name,Last_Name,Gender,DOB,Address,Pincode,Contact_Nu
mber, Mother_Name, Mother_Status,Father_Name, Father_Status, Marital_status, Spouse)
values('$fn','$mn','$ln','$g','$d','$a',$p,$con,'$mon','$mos','$fan','$fas','$ms','$s')";
        mysqli_query($conn,$query) or die($query."Can't Connect to Query...");
?>
```

client.php

```
<html>
  <head>
    <title>Registration Form</title>
  </head>
  <body>
    <link href = "reg1.css" type = "text/css" rel = "stylesheet" />
            <link href = "../style3.css" type = "text/css" rel = "stylesheet" />
            <ul>
                <li style="float:right;"><a href="../index.html"> Back to homepage</a></li>
            </ul>
            <h2>Customer</h2>
    <form name = "form1" action='modified.php' method = 'POST' enctype = "multipart/form-data" >
      <div class = "container">
        <div class = "form_group">
          <label>First Name:</label>
          <input type = "text" name = "First_Name" value = "" required placeholder="enter F-name" />
        </div>
        <div class = "form_group">
          <label>Middle Name:</label>
          <input type = "text" name = "Middle_Name" value = "" required placeholder="enter M-name" />
        </div>
        <div class = "form_group">
          <label>Last Name:</label>
          <input type = "text" name = "Last_Name" value = "" required placeholder="L-name" />
        </div>
                <div class = "form_group">
```

```
<label>Gender:   </label><input type = "radio" name = "Gender" value = "M" required />Male<input type = "radio" name = "Gender" value = "F" required />Female

</div>

<div class = "form_group">

<label>Date of Birth:     </label><input type = "date" name = "DOB" value = "" required placeholder="enter dob" />

</div>

<div class = "form_group">

<label>Address:</label>

<input type = "text" name = "Address" value = "" required placeholder="enter address" />

</div>


<div class = "form_group">

<label>Pincode: </label>

<input type = "text" name = "Pincode" value = ""  required placeholder="enter pincode" />

</div>

<div class = "form_group">

<label>Contact Number: </label>

<input type = "text" name = "Contact_Number" value = ""  required pattern="[0-9]{10}" placeholder="enter customer no" />

</div>

<div class = "form_group">

<label>Mother Name: </label>

<input type = "text" name = "Mother_Name" value = ""  required placeholder="enter mother name" />

</div>

<div class = "form_group">

<label>Mother Status: </label>

<input type = "radio" name = "Mother_Status" value = "A" required />Alive <input type = "radio" name = "Mother_Status" value = "D" required />Dead
```

```
            </div>
                                    <div class = "form_group">
               <label>Father Name: </label>
               <input type = "textbox" name = "Father_Name" value = ""  required
placeholder="enter father name" />
            </div>
                                    <div class = "form_group">
               <label>Father Status: </label>
               <input type = "radio" name = "Father_Status" value = "A" required />Alive
<input type = "radio" name = "Father_Status" value = "D" required/>Dead
            </div>
                                    <div class = "form_group">
               <label>Marital Status: </label>
               <input type = "radio" name = "Marital_Status" value = "S" required />Single
<input type = "radio" name = "Marital_Status" value = "M" required/>Married
            </div>
                                    <div class = "form_group">
               <label>Spouse Name: </label>
               <input type = "textbox" name = "Spouse" value = ""  placeholder="enter spause
name" />
            </div>
                                    <div class = "form">
               <input type = "submit" value = "submit"/>
            </div>
                                    <div class = "form">
               <input type = "reset" value = "reset"/>
            </div>

         </div>
      </form>
   </body>
```

```
</html>
```

client/delete.php

```
<?php
include "../connection.php";
if(is_numeric($_GET['id'])){
$sql = "delete from customer where Customer_Num = '".$_GET['id']."'";
$result = mysqli_query($conn,$sql);
}
header('Location:modified1.php');
?>
```

client/modified.php

```
<?php

include "input.php";
$sql = "select * from customer";
$result = mysqli_query($conn,$sql);
?>
<html>
   <body>
      <link href = "../style3.css" type = "text/css" rel = "stylesheet" />
                  <link href = "reg1.css" type = "text/css" rel = "stylesheet" />
                  <table width = "100%" border = "1" cellspacing = "1" cellpadding = "1">
         <tr>
            <td>Customer Number</td>
            <td>First Name</td>
            <td>Middle Name</td>
            <td>Last Name</td>
            <td>Gender</td>
            <td>DOB</td>
```

```
    <td>Address</td>
    <td>Pincode</td>
    <td>Contact Number</td>
    <td>Mother Name</td>
    <td>Mother Status</td>
    <td>Father Name</td>
    <td>Father Status</td>
    <td>Marital Status</td>
    <td>Spouse</td>
    <td colspan = "2">Action</td>
</tr>
<?php

        while($row = mysqli_fetch_object($result)){


?>
            <tr>
                <td>
                    <?php echo $row->Customer_Num;?>
                </td>
                <td>
                    <?php echo $row->First_Name;?>
                </td>
                <td>
                    <?php echo $row->Middle_Name;?>
                </td>
                <td>
                    <?php echo $row->Last_Name;?>
                </td>
```

```
<td>
        <?php echo $row->Gender;?>
</td>
<td>
        <?php echo $row->DOB;?>
</td>
<td>
        <?php echo $row->Address;?>
</td>
<td>
        <?php echo $row->Pincode;?>
</td>
<td>
        <?php echo $row->Contact_Number;?>
</td>
<td>
        <?php echo $row->Mother_Name;?>
</td>
<td>
        <?php echo $row->Mother_Status;?>
</td>
<td>
        <?php echo $row->Father_Name;?>
</td>
<td>
        <?php echo $row->Father_Status;?>
</td>
<td>
        <?php echo $row->Marital_status;?>
</td>
```

```
                                <td>
                                        <?php echo $row->Spouse;?>
                                </td>
                                <td> <a href="delete.php?id=<?php echo $row-
>Customer_Num;?>" onclick="return confirm('Are You Sure')">Delete
                                </a> </td>
                        </tr>
                <?php } ?>
        </table>
          <?php header('Location:modified1.php');?>
    </body>
</html>
```

client/modified1.php

```
<?php

include "../connection.php";
$sql = "select * from customer";
$result = mysqli_query($conn,$sql);
?>
<html>
    <body>
        <link href = "reg1.css" type = "text/css" rel = "stylesheet" />
                <link href = "../style3.css" type = "text/css" rel = "stylesheet" />
                <ul>
                        <li style="float:right;"><a href="../index.html"> Back to
homepage</a></li>
                </ul>
                <h1><center>Customer Data</center></h1>

                <table width = "100%" border = "1" cellspacing = "1" cellpadding = "1">
```

```
<tr>
    <td>Customer Number</td>
    <td>First Name</td>
    <td>Middle Name</td>
    <td>Last Name</td>
    <td>Gender</td>
    <td>DOB</td>
    <td>Address</td>
    <td>Pincode</td>
    <td>Contact Number</td>
    <td>Mother Name</td>
    <td>Mother Status</td>
    <td>Father Name</td>
    <td>Father Status</td>
    <td>Marital Status</td>
    <td>Spouse</td>
    <td colspan = "2">Action</td>
</tr>
<?php

        while($row = mysqli_fetch_object($result)){


?>
                <tr>
                    <td>
                        <?php echo $row->Customer_Num;?>
                    </td>
                    <td>
                        <?php echo $row->First_Name;?>
```

```
</td>
<td>
        <?php echo $row->Middle_Name;?>
</td>
<td>
        <?php echo $row->Last_Name;?>
</td>
<td>
        <?php echo $row->Gender;?>
</td>
<td>
        <?php echo $row->DOB;?>
</td>
<td>
        <?php echo $row->Address;?>
</td>
<td>
        <?php echo $row->Pincode;?>
</td>
<td>
        <?php echo $row->Contact_Number;?>
</td>
<td>
        <?php echo $row->Mother_Name;?>
</td>
<td>
        <?php echo $row->Mother_Status;?>
</td>
<td>
        <?php echo $row->Father_Name;?>
```

```
                    </td>
                    <td>
                        <?php echo $row->Father_Status;?>
                    </td>
                    <td>
                        <?php echo $row->Marital_status;?>
                    </td>
                    <td>
                        <?php echo $row->Spouse;?>
                    </td>
                    <td> <a href="delete.php?id=<?php echo $row->Customer_Num;?>" onclick="return confirm('Are You Sure')">Delete
                    </a> </td>
                </tr>
            <?php } ?>
        </table>
    </body>
</html>
```

5.3 SQL Stored Procedure

getcustomer()

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getcustomer`()
SELECT * FROM customer$$
DELIMITER ;
```

5.4 SQL Trigger

customer_AFTER_INSERT

```
CREATE TRIGGER `Insert_Trigger`
```

```
 AFTER INSERT ON `customer`
 FOR EACH ROW INSERT INTO triggers
VALUES(null, NEW.Customer_Num, 'intsertd', NOW());
```

customer_AFTER_UPDATE

```
CREATE TRIGGER `Update_Trigger`
 AFTER INSERT ON 'customer'
 FOR EACH ROW INSERT INTO triggers
VALUES(null, NEW.Customer_Num, 'updated', NOW());
```

customer_BEFORE_DELETE

```
CREATE TRIGGER `Delete_Trigger`
 AFTER BEFORE ON 'customer'
 FOR EACH ROW INSERT INTO triggers
VALUES(null, OLD.Customer_Num, 'updated', NOW());
```

CHAPTER 6

SCREENSHOTS

Fig. 6.1: Home Screen

The shows the Home Screen of the LIC

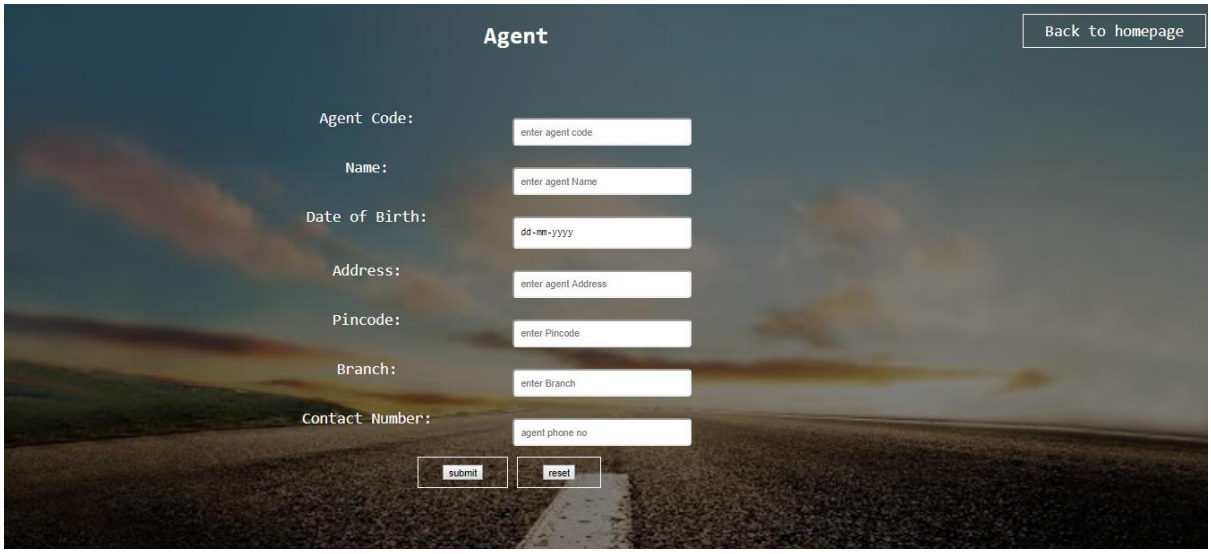

Fig. 6.2: agent Screen

The inserts the data of customers who have taken policies in customer table.

Customer Number is generated automatically in auto-increment.

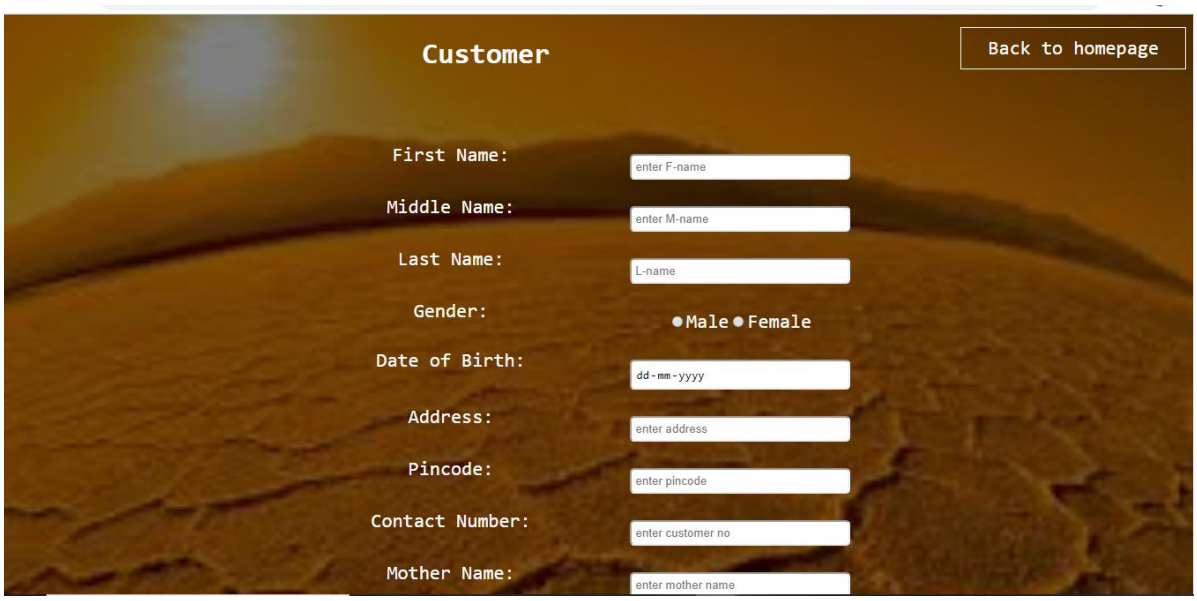

Fig .6.3: customer registration screen

This form inserts the data of policies taken by customers and stores in policy_data table. Calculation of premium is happens in back-end based on the mode.

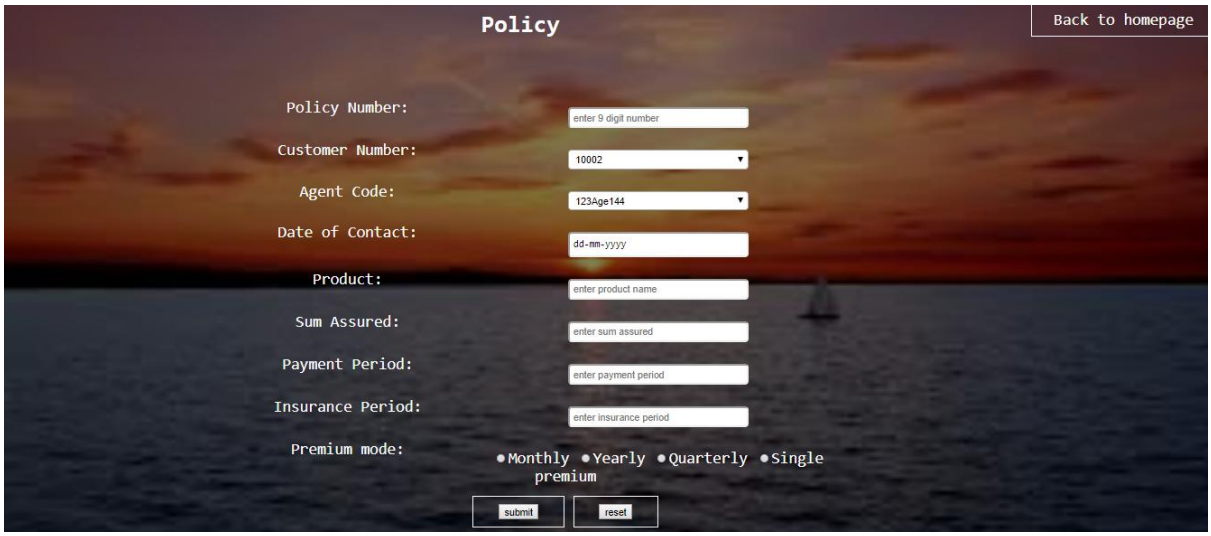

Fig.6.4: Policy Registration Screen

The inserts the data of policies taken by customers and stores in policy_data table. Calculation of premium is happens in back-end based on the mode.

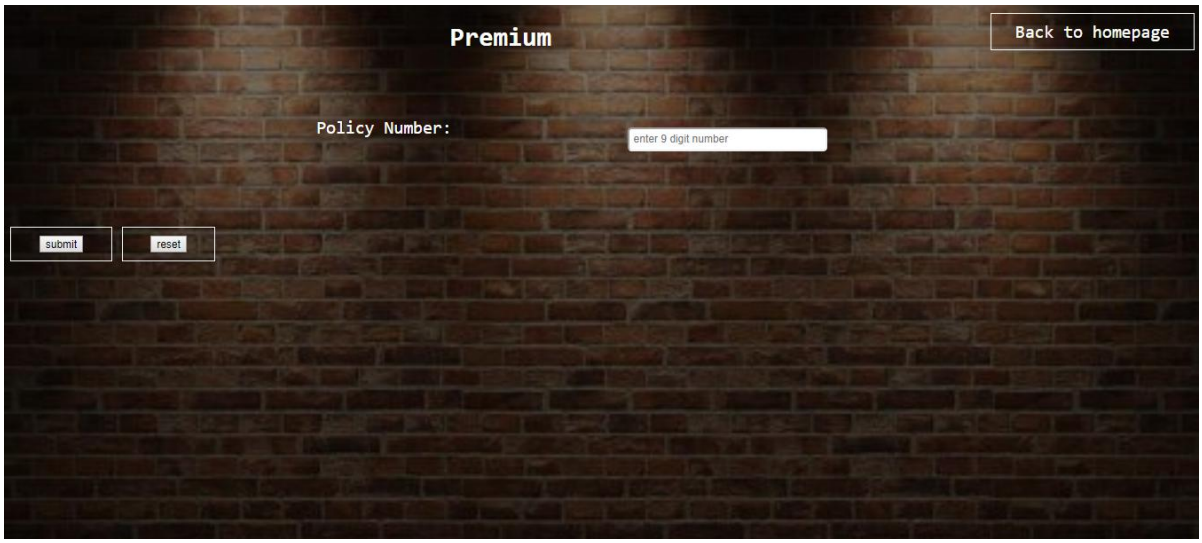

Fig. 6.5: Premium Registration Screen

This form leads to another page showing the details of the policy and asking for payment of premium or not.

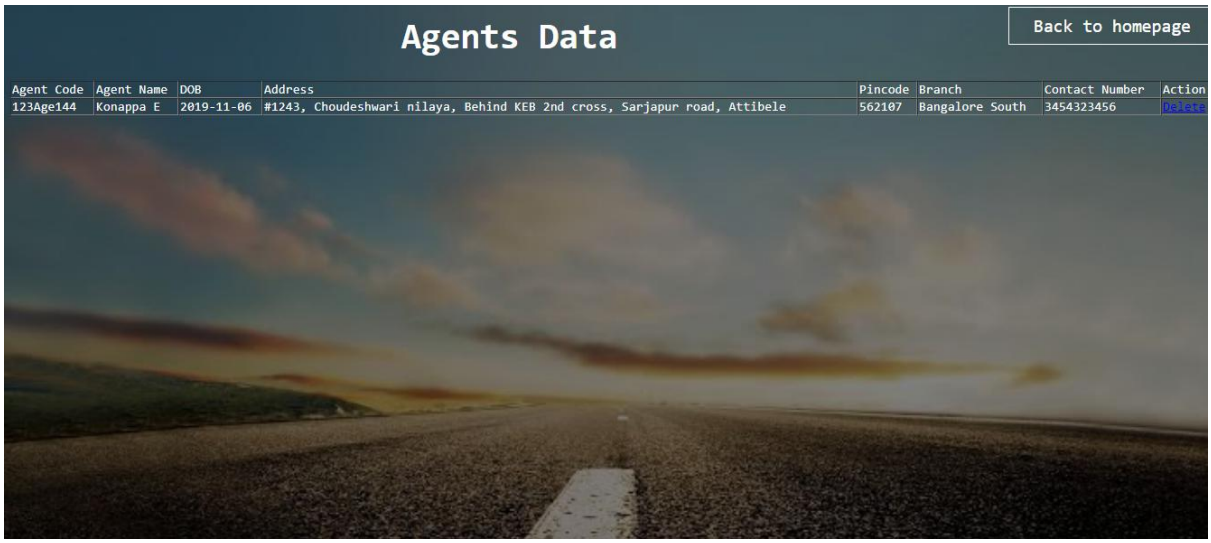

Fig. 6.6: Agent details Screen

This page shows the data stored in the table of Agent. It shows details of every agent of company and can be deleted also.

Fig. 6.7: Customer details Screen

This page shows the data stored in the table of customer. It shows details of every customer who took the policies and it can be deleted also.

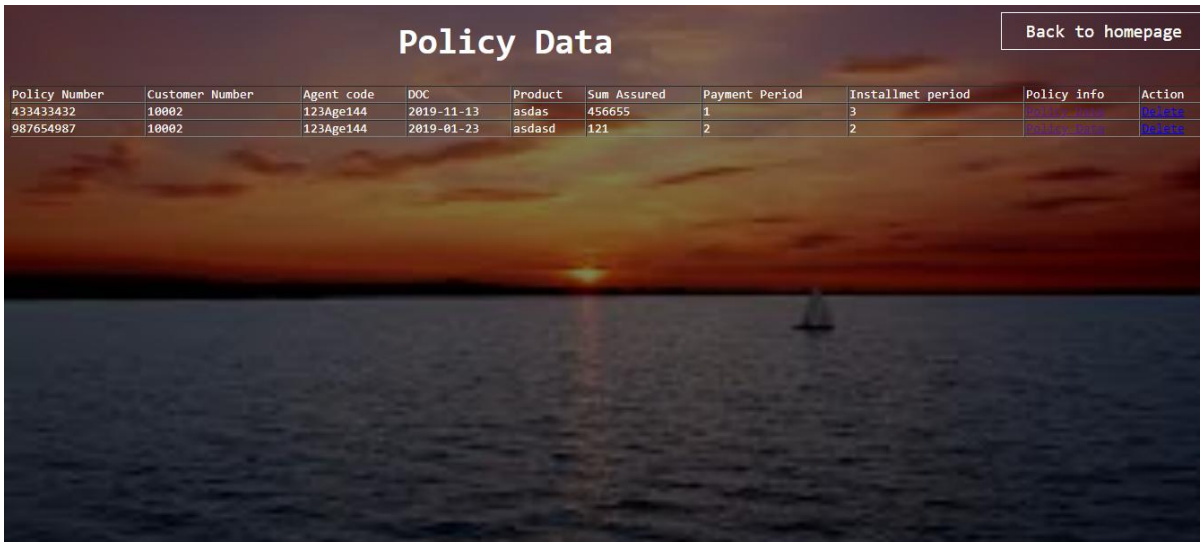

Fig. 6.8: Policy details Screen

This page shows the data stored in the table of policy_data. It shows details of all the policies and it can be deleted also. The link of 'Policy_data' in a column leads

to page showing every details of that specific policy.

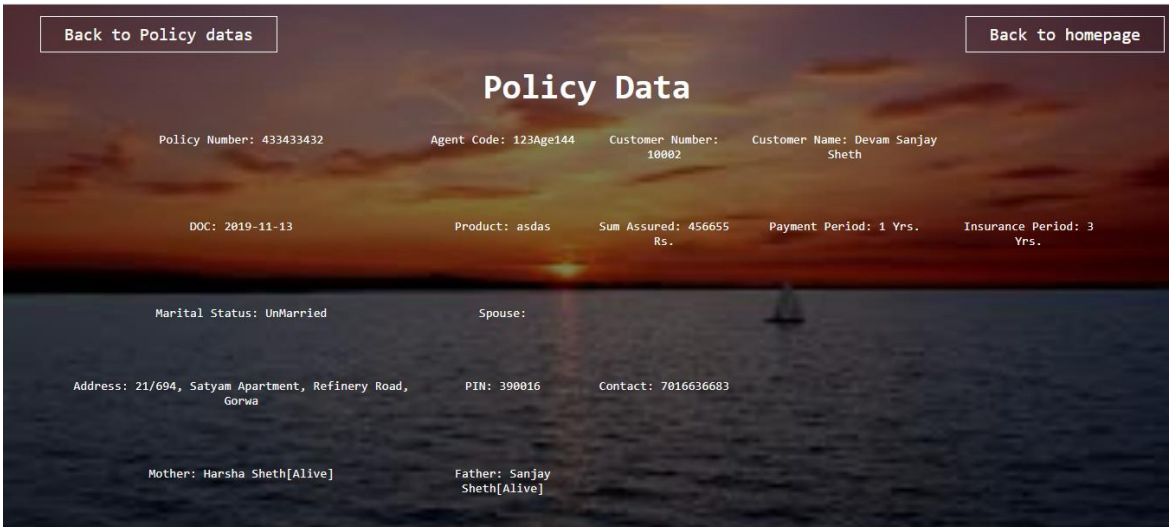

Fig. 6.9: Specify Policy details Screen

This page shows all the details of a specific policy selected in previous page.

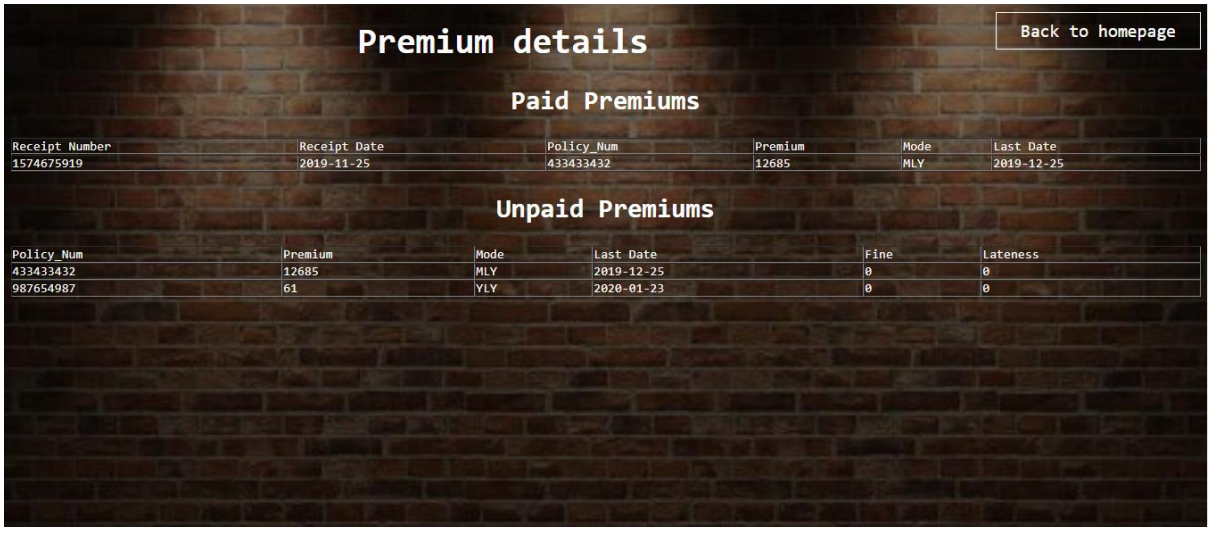

Fig. 7.0: Premium Details Screen

This page shows the data stored in the table of premiums, paid_premiums and unpaid_premiums.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Insurance is the backbone of a country's risk management system. Risk is an inherent part of our lives. The insurance providers offer a variety of products to businesses and individuals in order to provide protection from risk and to ensure financial security. In this project, we have to enhance the way the data is stored and the way we fetch the data from the database. The time required to access data has been reduced. In the existing system, unpaid and paid premiums are stored in one table, which in proposed system are in separate tables. So, whenever the admin needs to fetch the data for the paid and unpaid premiums the time required to sort and fetching data is saved.

For future of this project, we can the same thing for separating policies which are running and which are lapsed. The login for admin and customer can be created to protect the data

REFERENCES

[1] References for CSS and HTML - https://www.w3schools.com

[2] General - https://youtu.be/aIsu9SPcGbU

[3] References for CSS – https://codepen.io

[4] References for CSS and HTML - https://youtu.be/uyaV_EWWRmo

[5] Login Page - https://youtu.be/Vp0ASH8AoQQ