Shri B.V.V.Sangha's

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS)

BAGALKOT-587 103.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

For the academic year

2019-2020

Project Report

On

"Insurance policy management"

Project Coordinator
Prof.Suvarna Hanji

Head of the Department
Dr.Saboji

Submitted by:

Name
Vivek S Halakatti

USN
2BA18CS081

INDEX

Sl. Chapters

1)ABSTRACT

Our proposed project aims to build and manage a database that can be very beneficial for an insurance company. The insurance company needs to keep track of all the details of its target companies, agents, policyholders, their premium payments and the various products they offer. In the project, we intend to use the concepts involved in handling the data acquired from insurance companies, agents, and employees using MySQL and PHP. The database will include multiple tables which will be managed efficiently. Some functions will run which would classify the policies based on their status that whether they are active, lapsed, matured, etc. According to those classifications, different tables are created in database and according to functions, records are grouped in respective tables.

INTRODUCTION

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 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.

PROBLEM DESCRIPTION

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 back- end 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.

SYSTEM SPECIFICATIONS

An attractive and methodical Insurance Policy Management System requires the amalgamation and utilization of modern technologies like the PHP, CSS and MySQL.

PHP

PHP - It is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is an acronym for "PHP: Hypertext Preprocessor".

In the simplest terms, php is the acronym of “hypertext preprocessor”. It is the most commonly used server-side scripting language and is used predominantly for the purpose of web development.

PHP is the most extensively and commonly used server site on the web as it is known to rule or cover almost 82 % of the entire web today. Apart from its use in web development, PHP is used as the general purpose programming language as well.

The computer language PHP was used for the first time in history in the year 1995 and was known to be developed a year prior by a great computer scientist Rasmus Lerdof. Originally, PHP stood for “Personal Home Page” which was changed to “hypertext preprocessor” in the following years.

So, this was just a sneak peek in the computer language PHP but the exact meaning of hypertext preprocessor shall be disclosed in the further section of this blog.

From an eternity, most of the people have the same doubt as to “what do you mean by hypertext preprocessor?” as of now, no

exact meaning of hypertext preprocessor has been found out, but its function is quite clear.

The hypertext preprocessor is mainly used for the purpose of extraction of data from a database and then provides it to the web page. On the contrary, it is very often used for the purpose of connection with the database, retrieval of data, addition or for the purpose of updating the content.

This particular speciality of the hypertext preprocessor is said to make the computer language PHP the complete and the ideal language for the purpose of development of a large scale website

HTML

This is used for styling purpose. HTML coding is just a structure and CSS is applied to dictate the look and feel. Font size, font color, font style styling of images, page layout, and more are determined by CSS.

Hypertext Markup Language (**HTML**) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page.

HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997

MySQL

MySQL - It provides us a way to integrate and manage the database for the policy system by using the various commands to handle the queries.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.

- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.
- MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.
- MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.
- MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

XAMPP

XAMPP – This is a software used to connect php files and the database on a local server.

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

Its usage

The most obvious characteristic of XAMPP is the ease at which a WAMP webserver stack can be deployed and instantiated. Later some common packaged applications that could be easily installed were provided by Bitnami.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the World Wide Web.

A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress.

It is also possible to connect to localhost via FTP with an HTML editor.

PROBLEM FORMULATION

Existing System

In the existing Life Insurance Management System, the work is done by hand. All the details for the insurance such as cash information or age related important information was collected into the hard copy and by chance if any of the document get missed up or get harmed then whole of the information will be missed, resulted into the major loss for the user.

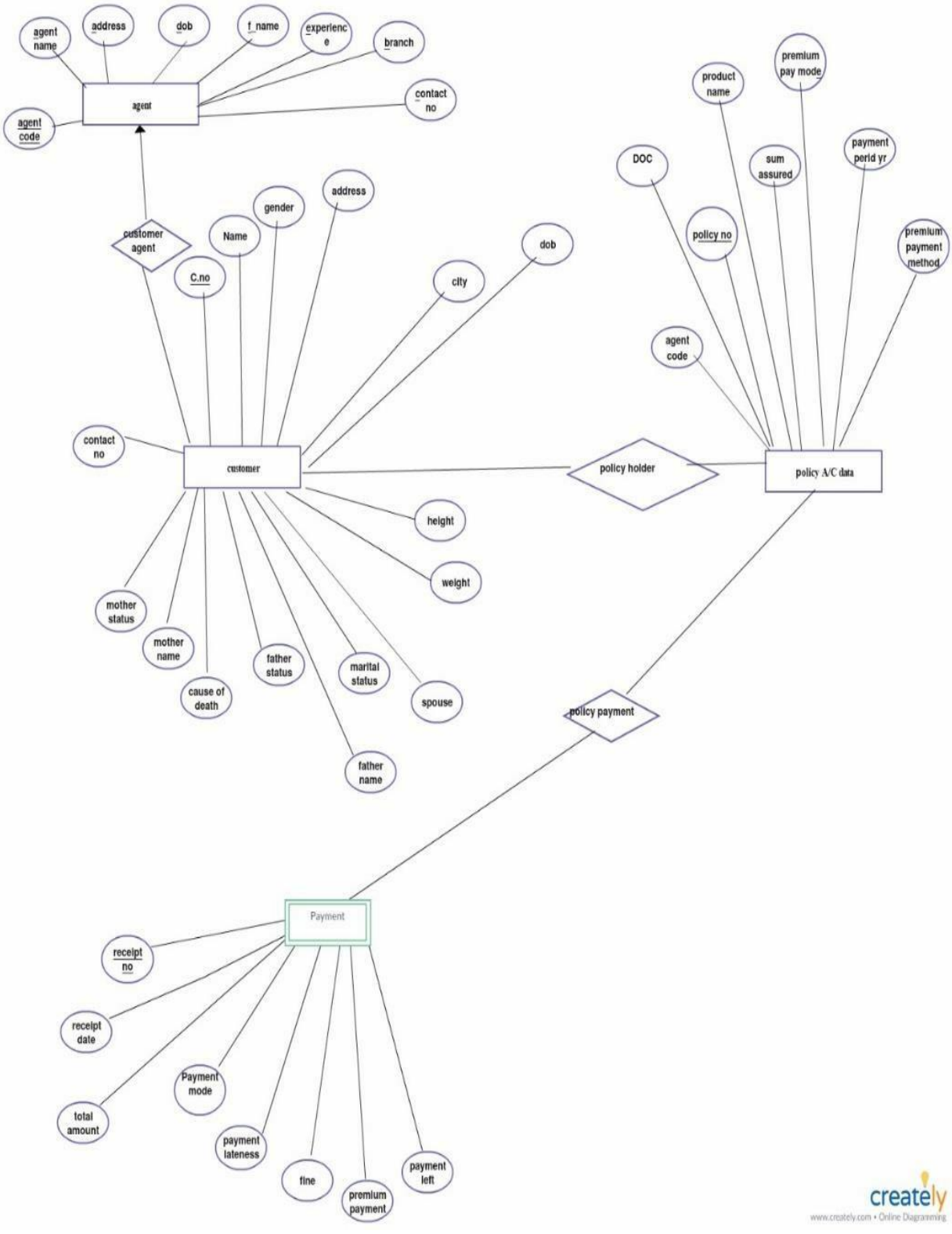
Also adding all the details manually will take a lot of time and also a lot more chances of entering the information wrong. And also sending details from one place to another will not be any task because in case while sending one important document from one place to another, it get lost, then also it will proved to a great loss both for the user as well as for the organization.

In current system any customer who wants to buy any life insurance policy has to contact Insurance Agent or visit the company directly. It will takes lots of extra time of customer. If customer wants to compare two policies than he has to one by one compare the details of policy manually and then decide which policy to buy.

Customer has to maintain record of each and every policy owned by him manually or enter the details in excel sheet. It is very tedious task for the customer to remember on which month he has to pay policy premium and on which date his policy ends.

Life Insurance Company also has to give advertisement about new policy plans either in News Paper or in TV channels. So it is very costly for the Life Insurance Company. Customer has to pay policy premium by visiting branch office or contacting insurance agent.

ER Diagram of existing system



Following are disadvantages of Existing System

After studying existing system we found following problems and weaknesses in the system:

1) Current system is manual.
2) It is time consuming because it takes lots of time in searching accurate detail about various life insurance policy plans.
3) It is very difficult to locate the agents in order to buy policy.
4) Customer has to maintain the records of policy hold by them manually. It is very difficult for the customer to remember when he has to pay policy premium and what the date of his policy maturity is.
5) Online Policy Premium Payment is not possible.
6) It is costly in terms of advertisement and marketing for Life insurance Company

Objectives

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. It also helps the customer to view their own insurance status information. The web pages provide easy links for easy navigation in the system. A visitor with minimum knowledge of web browsing/surfing can access the site very easily. 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. Due to dynamic nature of features, the members, admin members should be able to understand the provided facilities. After getting details of all the

insured people, system should be able to delete unwanted persons' data. If the policy holder wants to view the information about their own policy details, he/she can login to policy status page by using the necessary details already given by insurance company and view their own details and also they give feedback to the insurance company. An online help documentation will be provided to help the users and visitors in using the facilities.

- To computerize the Insurance System.
- To reduce Data Redundancy.
- To reduce the cumbersome job of maintaining several documents.
- To eliminate the delays in report generation for insurance policies.
- To facilitate faster searching of information by insurance companies and concerned parties.
- Thus, reducing time, energy and cost.
- To give assurance to the policy holders about maintain Data Privacy and Security.

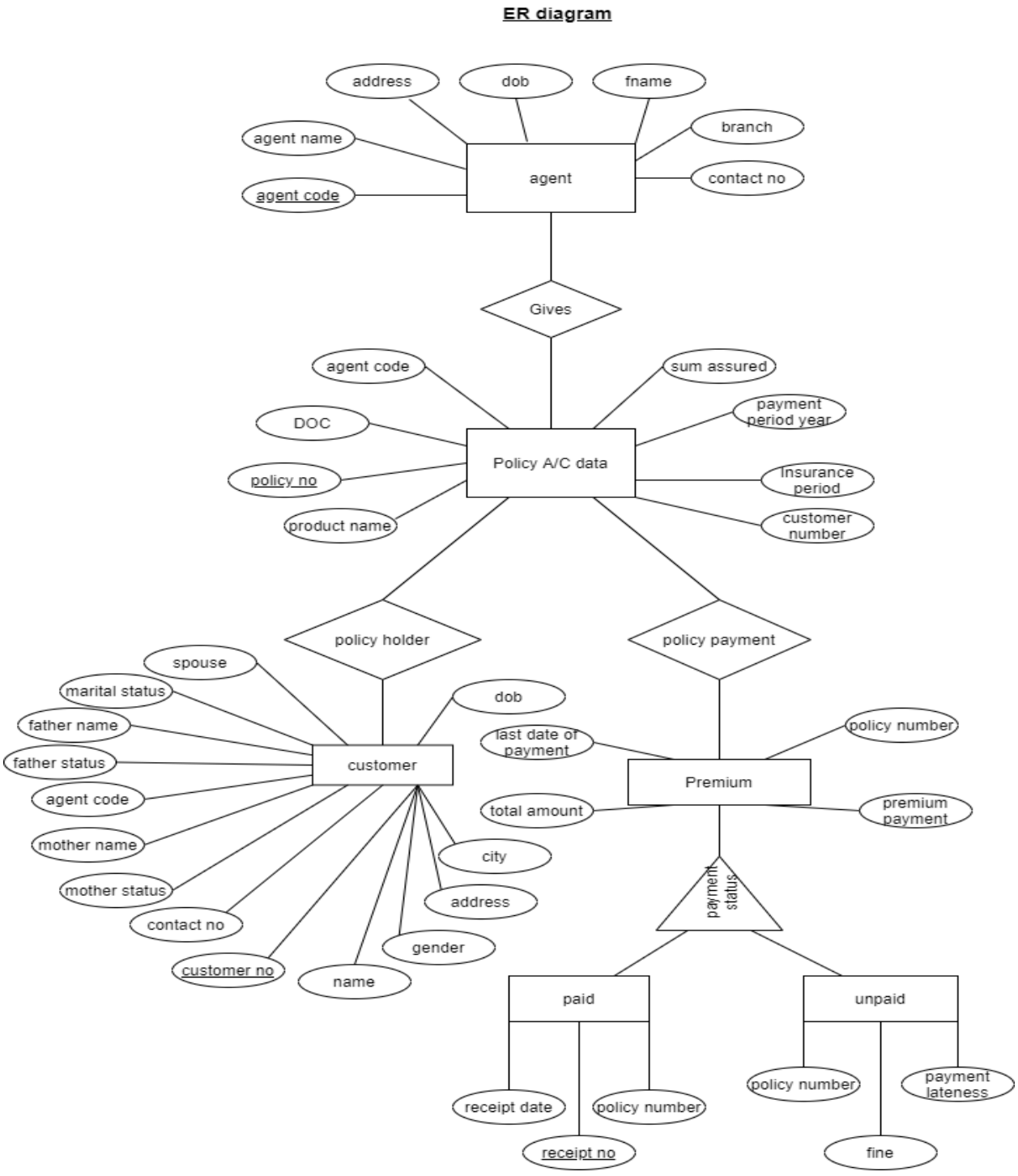
Proposed System

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 managing the insurance management system. The overall system is control through the main menu. The report is prepared for the schemes and implemented by the concerned officials.

Proposed system ER Diagram



Creation of Tables

Agent

```
CREATE TABLE `agent` (
  `Agent_code` varchar(10) NOT NULL,
  `Agent_name` varchar(150) NOT NULL,
  `DOB` date NOT NULL,
  `Address` varchar(80) NOT NULL,
  `Pincode` int(6) NOT NULL,
  `Branch` varchar(50) NOT NULL,
  `Contact_Num` bigint(10) NOT NULL
)
```

Customer

```
CREATE TABLE `customer` (
  `Customer_Num` bigint(10) NOT NULL,
  `First_Name` varchar(50) NOT NULL,
  `Middle_Name` varchar(50) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Gender` char(1) NOT NULL,
  `DOB` date NOT NULL,
  `Address` varchar(70) NOT NULL,
  `Pincode` int(6) NOT NULL,
  `Contact_Number` bigint(10) NOT NULL,
  `Mother_Name` varchar(150) NOT NULL,
  `Mother_Status` varchar(10) NOT NULL,
  `Father_Name` varchar(150) NOT NULL,
  `Father_Status` varchar(10) NOT NULL,
  `Marital_status` char(1) NOT NULL,
  `Spouse` varchar(150) DEFAULT NULL
)
```

Paid premium

```
CREATE TABLE `paid_premium` (
  `Receipt_Num` int(23) NOT NULL,
  `Receipt_Date` date NOT NULL,
  `Policy_Num` int(15) NOT NULL
)
```

Policy data

```
CREATE TABLE `policy_data` (
  `Policy_Num` int(15) NOT NULL,
  `Customer_Num` bigint(10) NOT NULL,
  `Agent_code` varchar(10) NOT NULL,
  `DOC` date NOT NULL,
  `Product` varchar(50) NOT NULL,
  `Sum_Assured` int(10) NOT NULL,
  `Pay_Period` int(2) NOT NULL,
  `Ins_Period` int(2) NOT NULL
)
```

Premium

```
CREATE TABLE `premium` (
  `Policy_Num` int(15) NOT NULL,
  `Premium` int(10) NOT NULL,
  `Mode` varchar(3) NOT NULL,
  `Last_date` date NOT NULL
)
```

Unpaid premium

```
CREATE TABLE `unpaid_premium` (
  `Policy_Num` int(15) NOT NULL,
  `Fine` int(10) NOT NULL,
  `Lateness` int(11) NOT NULL)
```

IMPLEMENTATION

Home Page

The home page contains the links to data and registration pages. Clicking on any one of the link as per the requirement of the user will direct the user to the required data or registration page.

(a) Agent Registration Page

This page takes the details of the agents employed in the company and stores the data in agent table of database.

(b) Customer Registration Page

This page takes the details of the customers/clients of the company and stores the data in the customer table of database.

(c) Policy Registration Page

This page takes the details of the policies registered by the customer and stores the data in the 'policy_data' table of the database

(d) Premium Payment Page

This page takes the Policy Number as input and enables the admin to make premium payment of the customer and makes changes between 'paid_premium' and 'unpaid_premium' tables of the database.

(e) Agent data

This shows the data of agents employed in the company from the agent table.

(g)Customer data storage

This shows the data of customers/clients of the company from the customer table. Customer number in the table is primary key.

(f) Policies data storage view

This shows the policies registered by the clients and their policy details from the 'policy_data' table.

(g) Premium Details view

This shows the details of the Paid and Unpaid premium of all the policies of all the customers.

(h) Detailed Policy Data

This shows a very detailed information about the selected policy registered along with the details of the client in 'policy_data' table

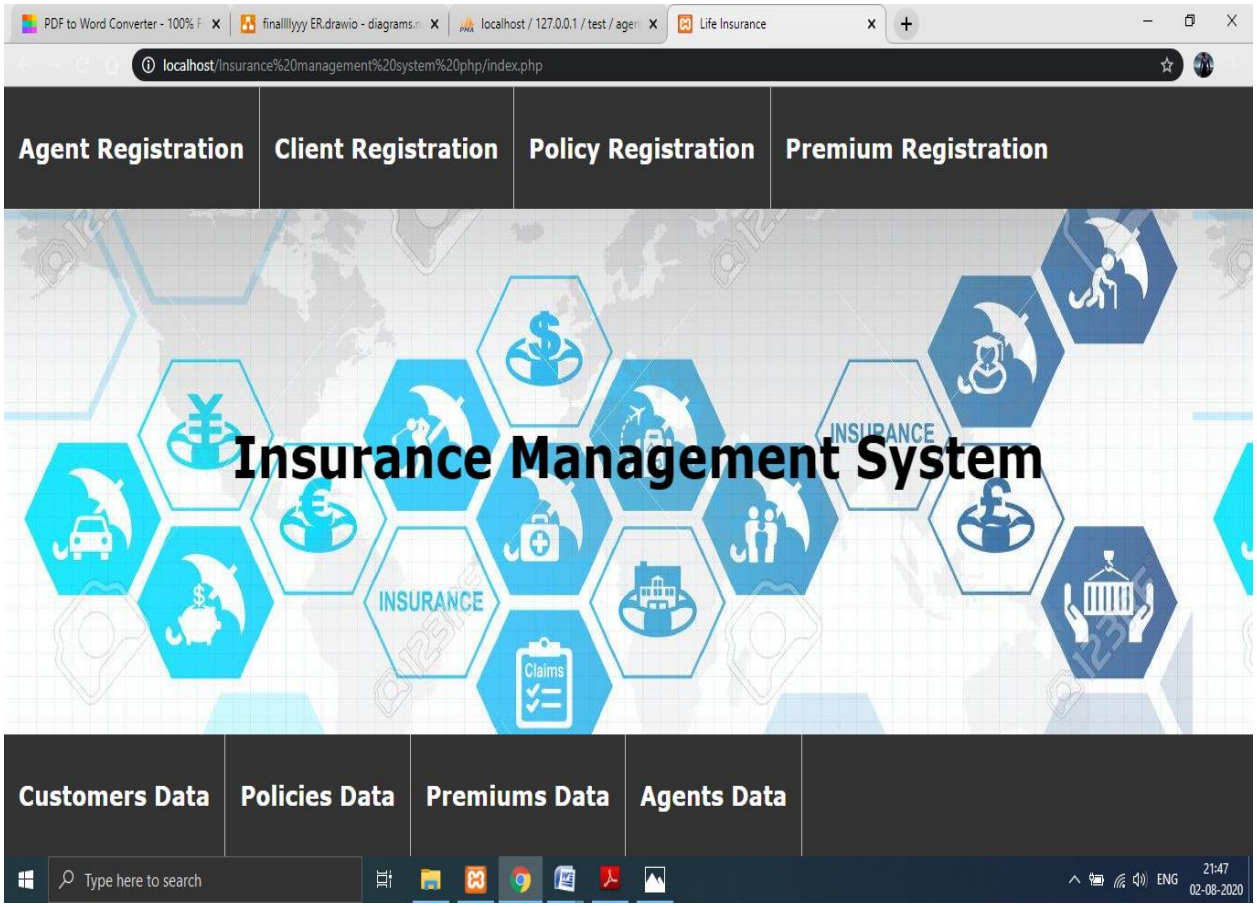
DESCRIPTION OF MODULES

There are 4 base modules used in this project:-

1) connection.php – This file is common for all and included in all the other pages used in this project. This file is to connect the database to the page in which query is going to be run making changes in the database.
2) input.php – Each file used for inserting the data in the database includes this file. It is used to insert data in the database. It includes the connection.php file to connect to database, fetches inputs from the file where user (admin) gives the input and runs the insert query.
3) modified.php – This file is used to display the contents of the database. For each table, there is different modified.php file. This connects to database and shows the data fetched from the database.
4) delete.php – This file is used to run delete query in database. The link is provided in the modified.php with each tuple to delete that entry.

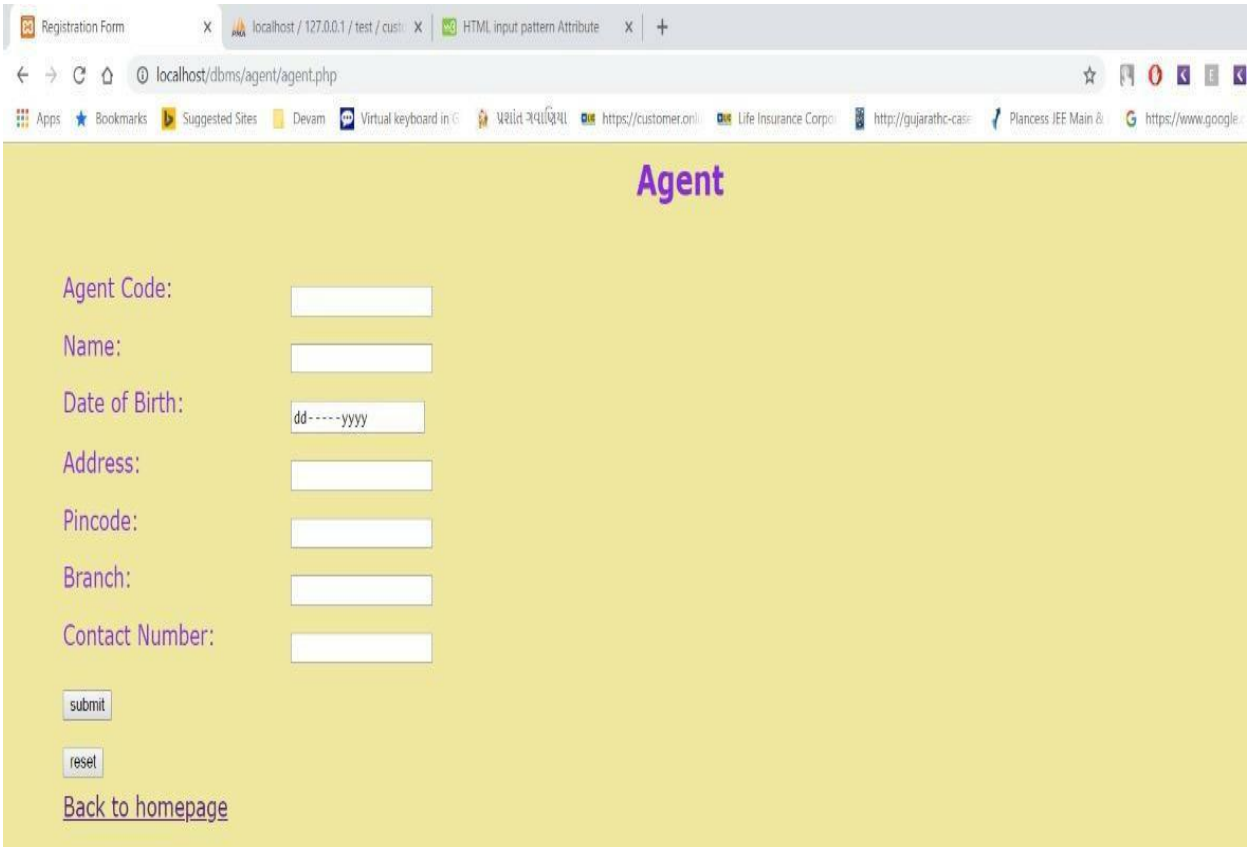
Execution Snapshots

HomePage



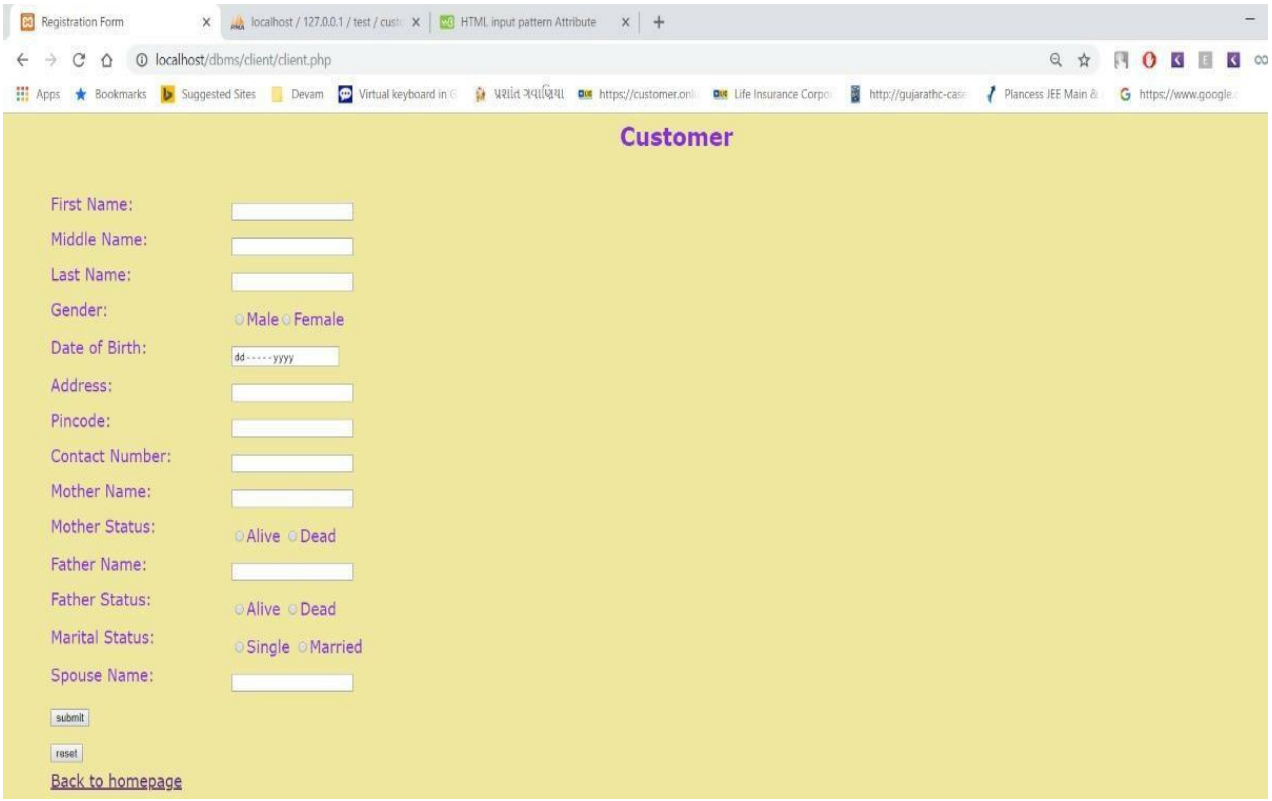
This page consists of links reaching all the other pages for registration and showing the data stored of agents, customers, their policies and their premiums.

Agent Registration Page



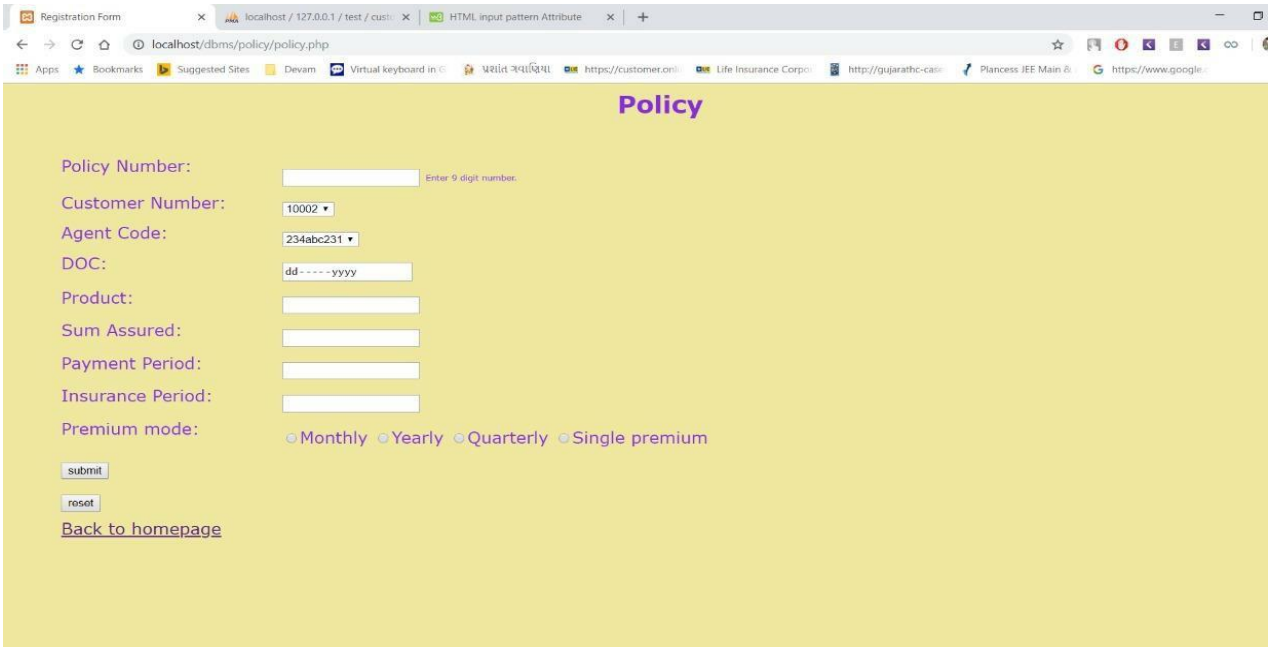
This page is the form for entering the data of Agents employed in the company. The data is inserted into agent table.

Customer Registration Page



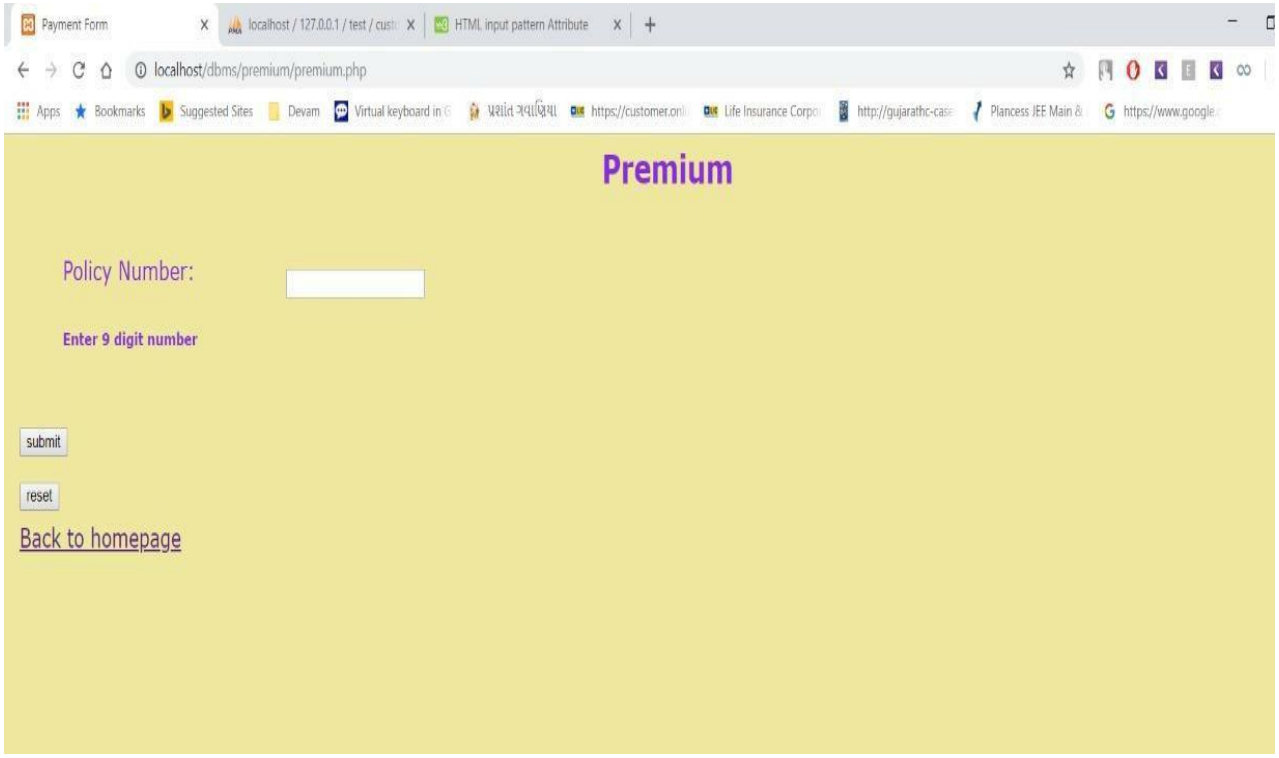
This page inserts the data of customers who have taken policies in customer table. Customer Number is generated automatically in auto-increment.

Policy Registration Page



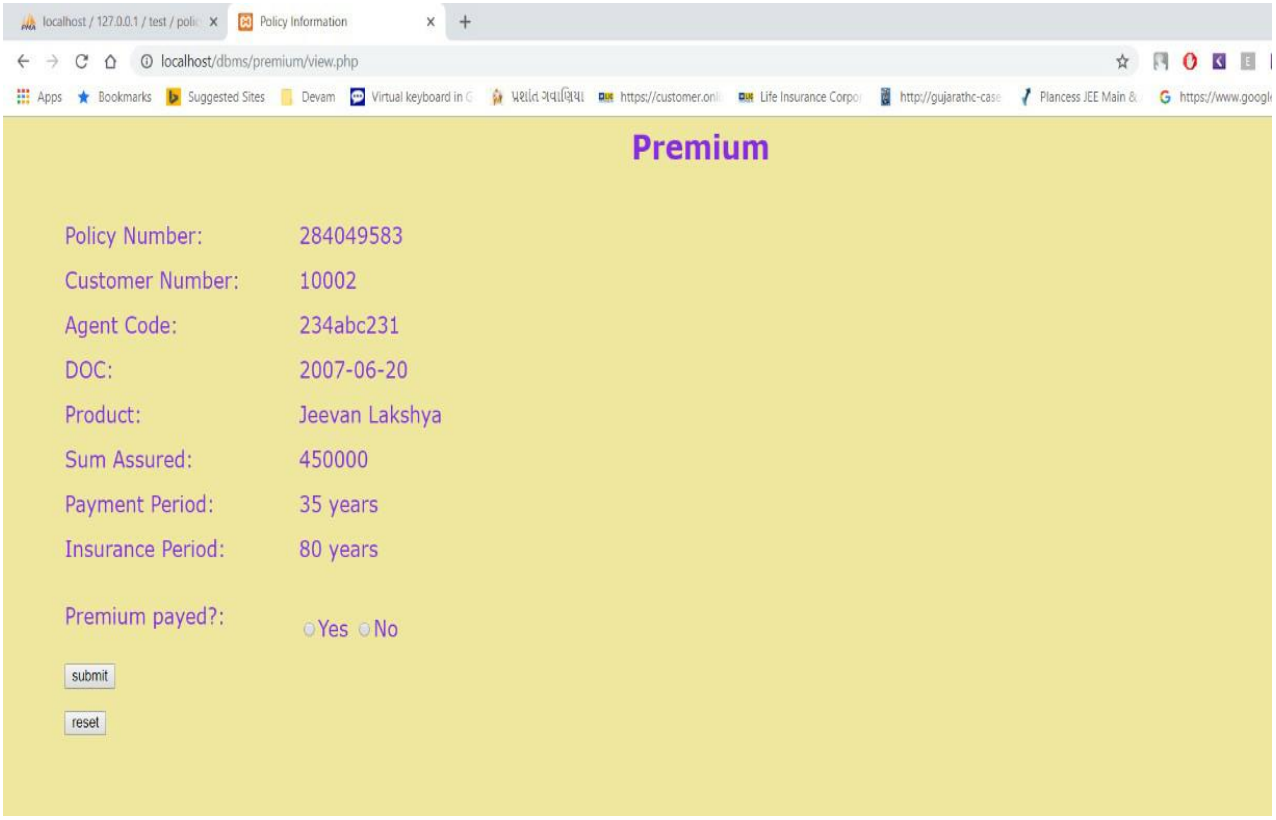
This form inserts the data of policies taken by customers and stores in policy_data table. Calculation of premium is happens in backend based on the mode

Premium Payment Page



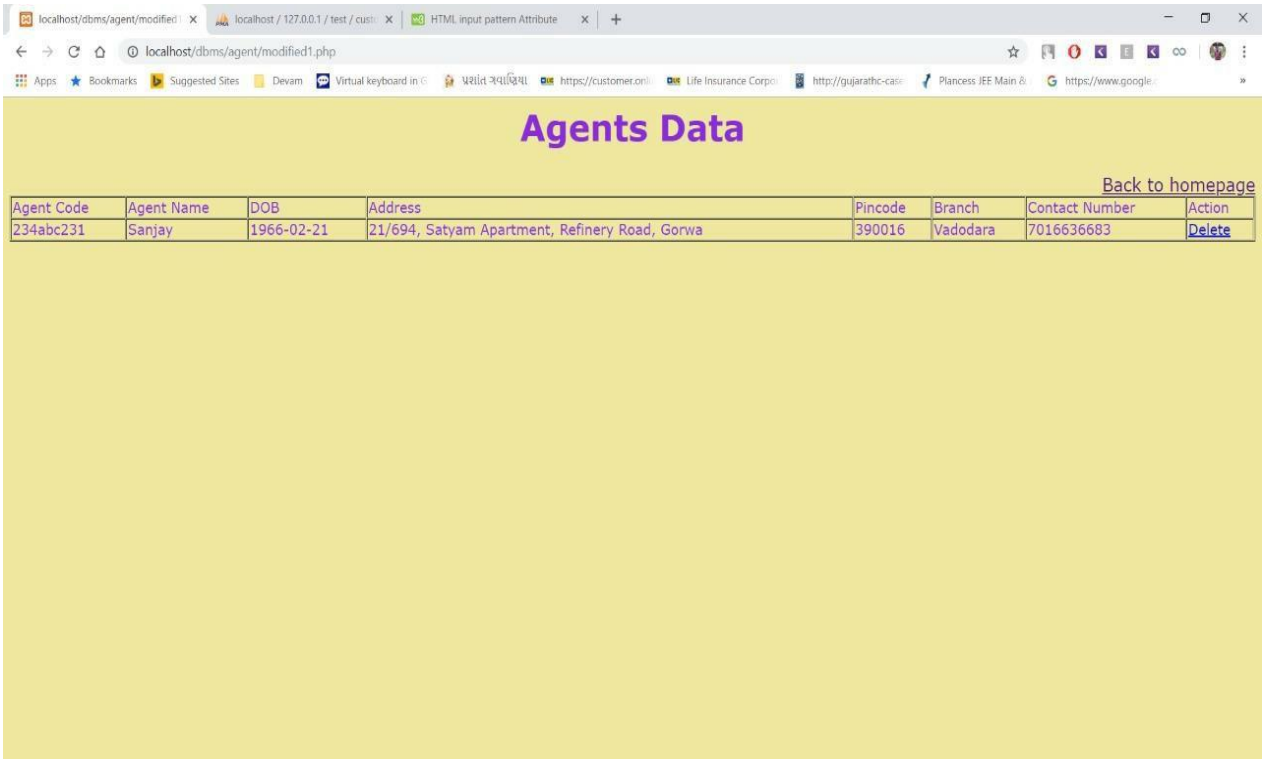
This form leads to another page showing the details of the policy

and asking for payment of premium or not.



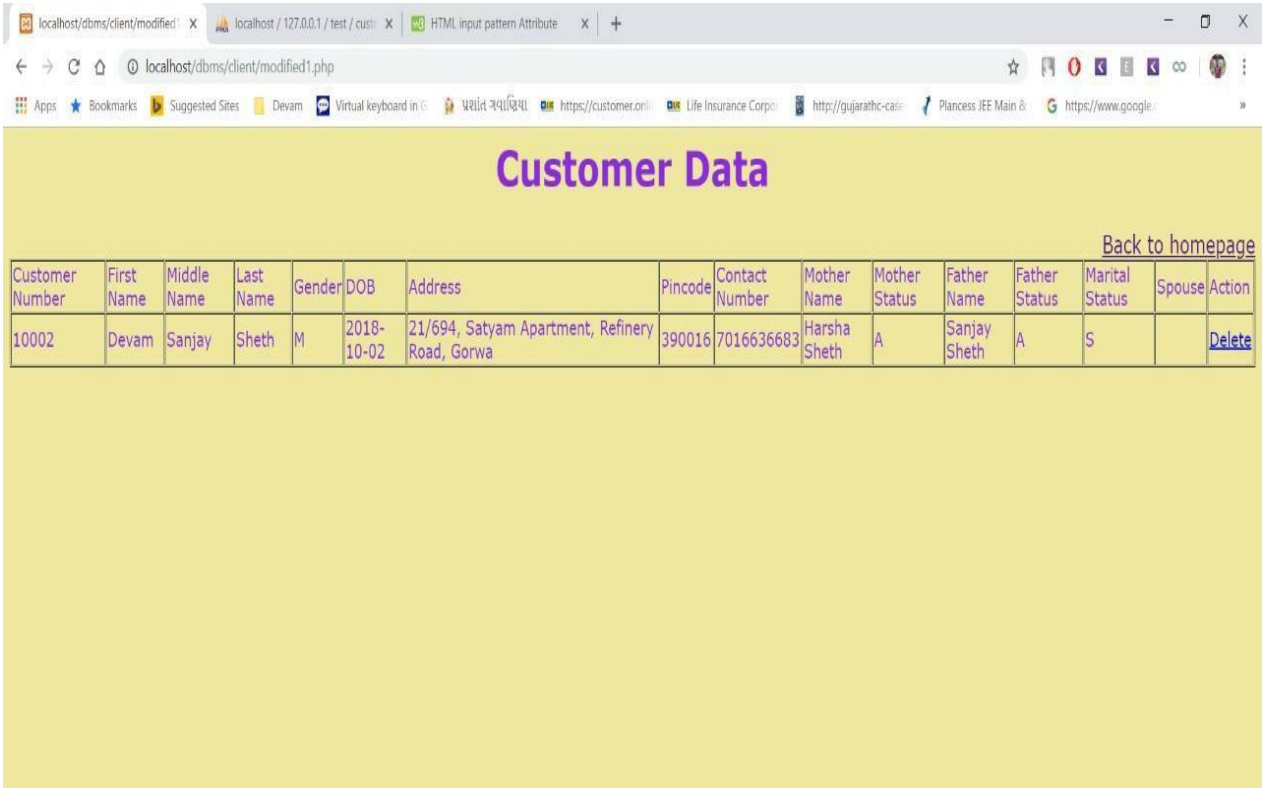
From this page admin confirms that premium is paid or not.

Agent Data



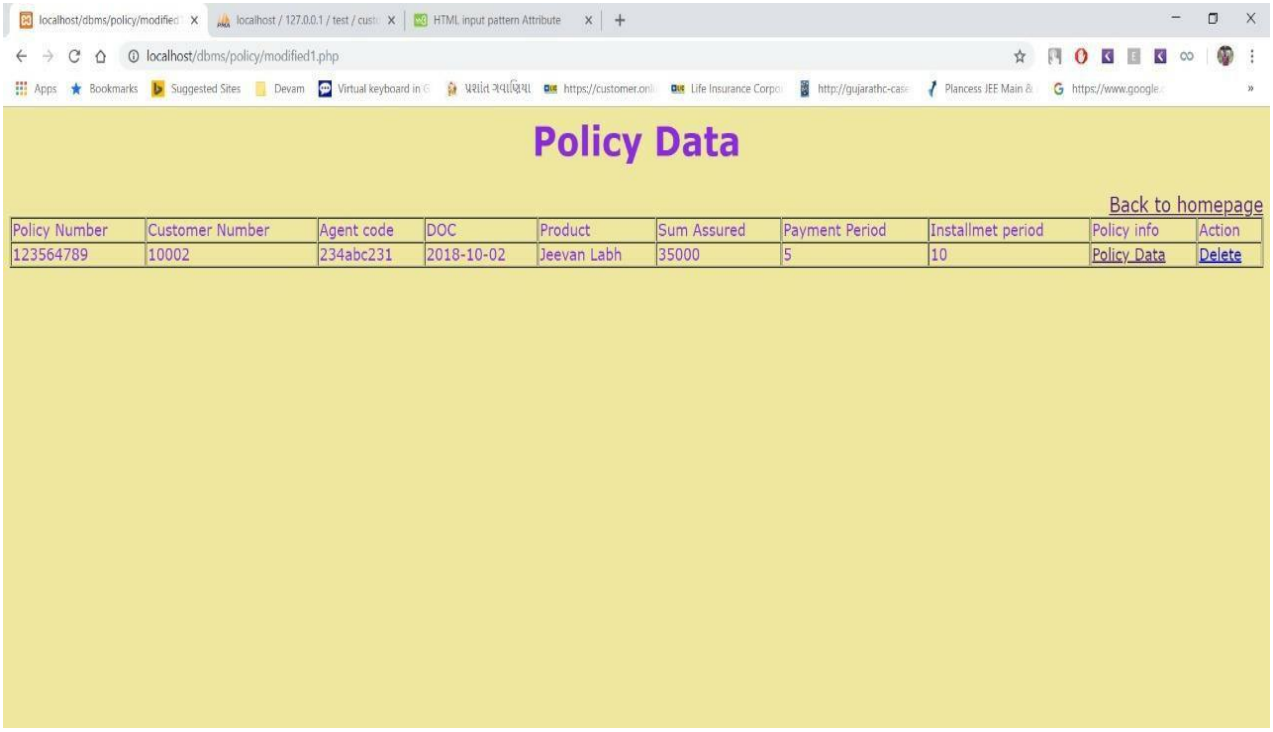
This page shows the data stored in the table of Agent. It shows details of every agent of company and can be deleted also.

Customer Data Storage



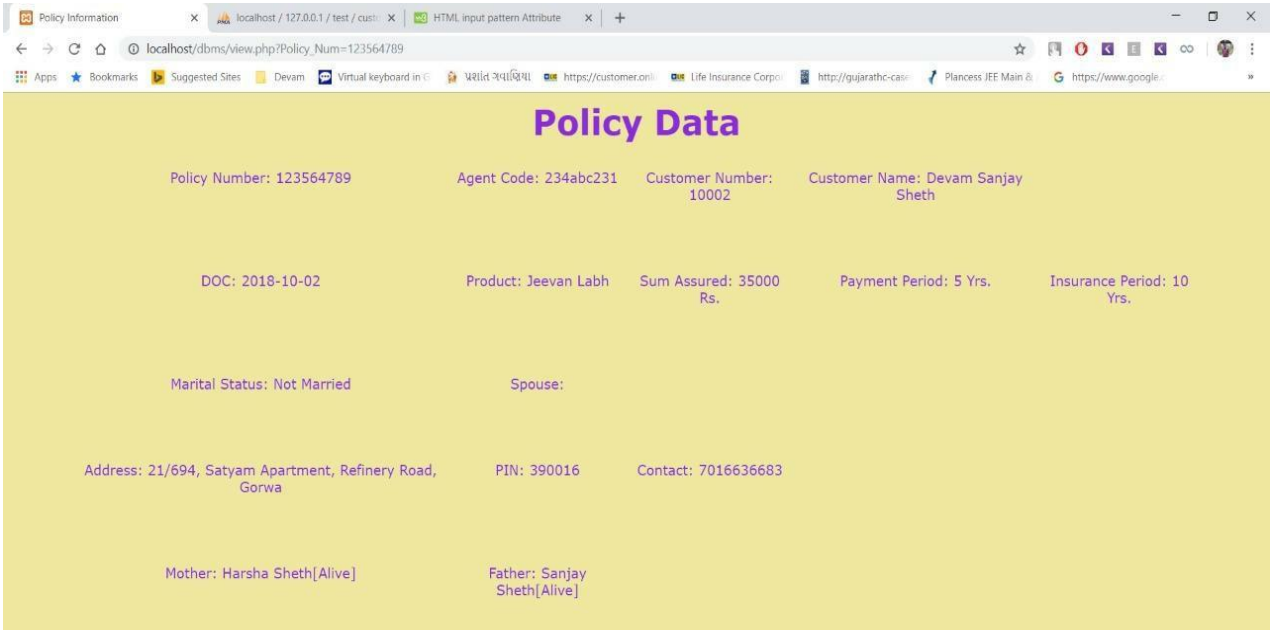
This page shows the data stored in the table of customer. It shows details of every customer who took the policies and it can be deleted also.

Policies Data Storage View



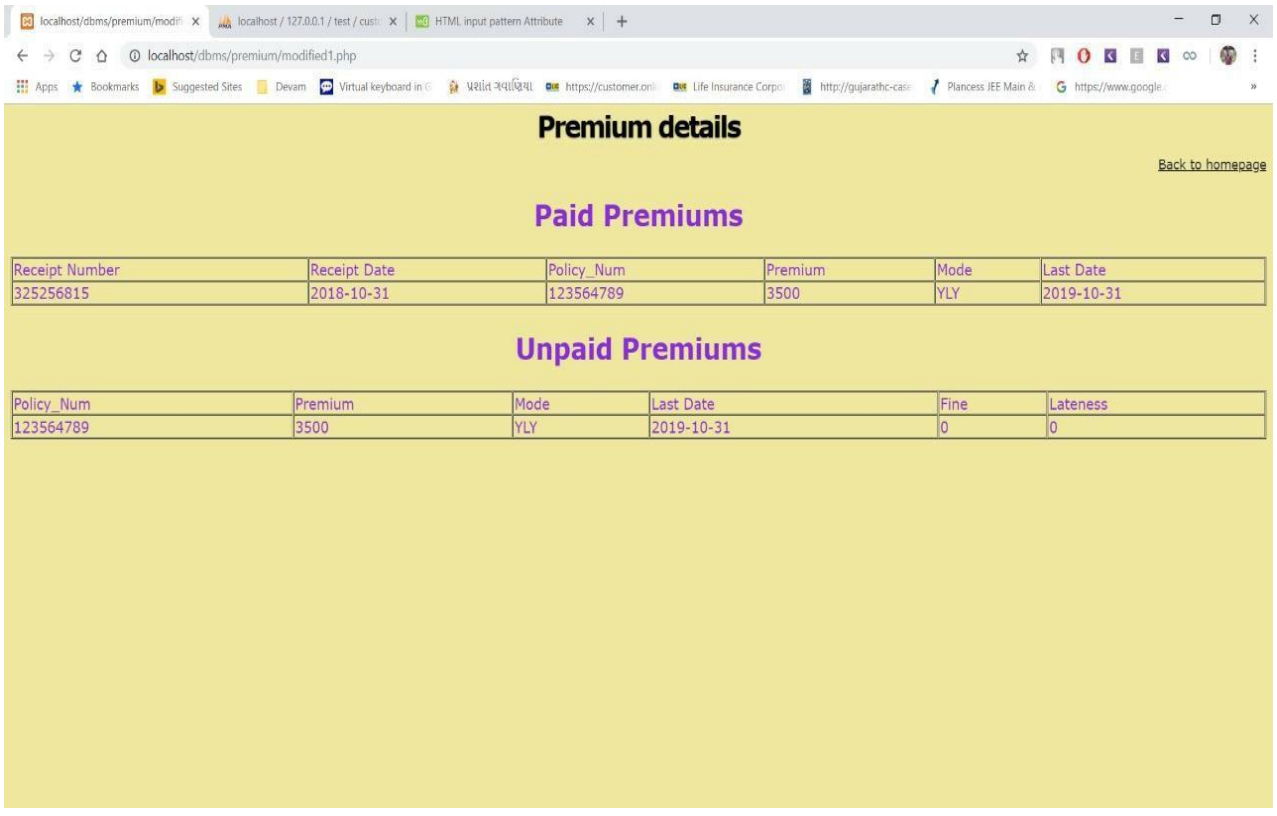
This page shows the data stored in the table of policy_data. It shows details of all the policies and it can be deleted also. The link of 'Policy_data' in a column leads to page showing every details of that specific policy.

Detailed Policy data for each Policy



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

Premium Details View



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

APPLICATIONS

The application of our project 'Insurance Policy Data Management' is like any other conventional management system i.e. we can store the details for the employees working in the company, clients 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.

CONCLUSION & FUTURE DIRECTIONS

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.

APPENDIX

connection.php

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

agent.php -> for input of agent details

```
<html>
  <head>
    <title>Registration Form</title>
  </head>
  <body>
    <link href = "registration.css" type = "text/css" rel = "stylesheet" />
        <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" required
pattern="[0- 9]{3}[A-Z a-z]{3}[0-9]{3}"/>

</div>
<div class = "form_group">

<label>Name:</label>

<input type = "text" name = "Agent_Name" value = "" required />

</div>

<div class = "form_group">

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

</div>

                    <div class = "form_group">

<label>Address:</label>

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

</div>

                    <div class = "form_group">

<label>Pincode: </label>

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

</div>

                    <div class = "form_group">

<label>Branch: </label>

<input type = "text" name = "Branch" value = "" required" />
```

```
            </div>
                                <div class = "form_group">
              <label>Contact Number: </label>
              <input type = "text" name = "Contact_Number" value = ""
required pattern="[0-9]{10}" />
            </div>

                                <div class = "form_group">
              <input type = "submit" value = "submit"/>
            </div>
                                <div class = "form_group">
              <input type = "reset" value = "reset"/>
            </div>
                                <div class = "form_group">
              <label><a href="../index.php">Back to homepage</a></label>
            </div>
          </div>
        </form>
    </body>
</html>
```

Delete.php -> for deletion of agent details

```
<?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');

?>

Modified.php -> for showing details of agents

<?php

include "../connection.php";

$sql = "select * from agent";

$result = mysqli_query($conn,$sql);

?>

<html>

<body>

<link href = "registration.css" type = "text/css" rel = "stylesheet" />

<h1> Agents Data </h1>

 Back to homepage</h4>

<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>
```

- Delete.php, input.php and modified.php is for agent, customer, policies and premiums.
- Connection.php is common file for all
- The given code is for agent only, but customer, premiums and policies have similar code.

REFERENCES

i. https://www.researchgate.net/publication/274642647_Coverage_Criteria_for_Testing_SQL_Queries
ii. https://www.acko.com/articles/general-info/types-of-insurance/
iii. https://www.ey.com/Publication/vwLUAssets/EY_-_Global_insurance_industry_trends_-_policy_management/$FILE/ey-global- insurance-industry-trends.pdf
iv. https://stackoverflow.com/questions/32379038/how-to-generate-entity- relationship-er-diagram-of-a-database-using-microsoft-s
v. https://www.youtube.com/watch?v=rn8T1bWBXcE
vi. https://www.youtube.com/watch?v=kBy-La7DXpg
vii. http://services.lovelycoding.org/insurance-agency-management-system- project/
viii. https://dl.acm.org/citation.cfm?id=1083734
ix. http://ilpubs.stanford.edu:8090/404/
x. https://lagunita.stanford.edu/courses/DB/SQL/SelfPaced/info
xi. https://www.c-sharpcorner.com/UploadFile/52bd60/create-an-html-form- and-insert-data-into-database162/
xii. https://www.coursehero.com/file/22258687/Insurance-Management-System-report/
xiii. https://www.slideshare.net/sangeethavasan/insurance-database- management-system