**Table of Contents**

[**INTRODUCTION** 1](#_Toc515101164)

[**OBJECTIVES** 1](#_Toc515101165)

[**REQUIREMENT ANALYSIS** 2](#_Toc515101166)

[**Existing system** 2](#_Toc515101167)

[**Praposed system** 2](#_Toc515101168)

[**SYSTEM ANALYSIS** 3](#_Toc515101169)

[**Admin Module** 3](#_Toc515101170)

[**Proposer module** 3](#_Toc515101171)

[**Agent module** 3](#_Toc515101172)

[**HARDWARE & SOFTWARE REQUIREMENT** 3](#_Toc515101173)

[**Hardware requirements** 3](#_Toc515101174)

[**Software requirements** 4](#_Toc515101175)

[**TECHNOLOGIES USED** 4](#_Toc515101176)

[**Hyper text markup language(html)** 4](#_Toc515101177)

[**Mysql** 5](#_Toc515101178)

[**Javascript** 5](#_Toc515101179)

[**Css (cascading style sheets)** 6](#_Toc515101180)

[**D3.js** 7](#_Toc515101181)

[**SOFTWARE REQUIREMENT SPECIFICATION** 8](#_Toc515101182)

[**Introduction** 8](#_Toc515101183)

[**Purpose** 8](#_Toc515101184)

[**Functional requirements** 8](#_Toc515101185)

[**Non-functional requirements** 9](#_Toc515101186)

[**SYSTEM DESIGN** 10](#_Toc515101187)

[**Dfd\_0 level:** 10](#_Toc515101188)

[**Dfd\_2 level:** 10](#_Toc515101189)

[**DFD-2 level:** 11](#_Toc515101190)

[**DATABASE TABLE** 12](#_Toc515101191)

[**SYSTEM IMPLEMENTATION** 14](#_Toc515101192)

[**Coding** 14](#_Toc515101193)

[**TESTING** 18](#_Toc515101194)

[**Test cases and Results** 19](#_Toc515101195)

[**SNAPSHOTS:** 20](#_Toc515101196)

[**VALIDATION** 28](#_Toc515101197)

[**CONCLUSION** 28](#_Toc515101198)

[**FUTURE ENHANCEMENT** 29](#_Toc515101199)

[**BIBLIOGRAPHY** 29](#_Toc515101200)

# **INTRODUCTION**

A website is a collection of related WebPages, including multimedia content, typically identified with a common domain name and published on at least web server.

In today’s world life is full of risks so everyone wants to make himself secure, everyone wants to keep them on the safe side policy plays a vital role and provide them insurance about their money security, health security, accidental security etc.

People mostly keep cats, dogs, fishes and birds as pets. Most people pet animals for their love for them while others pet them for their need for instance for security purpose, companionship, etc. However, whatever be the case pets eventually become an integral part of the family.

The insurance policy covers animals such as Cats and Dogs owned by different individuals and which are used for commercial and for personal purpose against the risk of permanent total disablement or death due to the accident and/or any diseases which the animal may contract during the policy period

This system has been proposed to maintain the record of the policy holder, agent details, policy detail, payment details. So, this will be developed to manage the insurance of the clients and their relative records.

# **OBJECTIVES**

* This project has provided easy attractive web pages to help the users to operate with minimal computer knowledge.
* This site gives all the information about the pet insurance to provide better service for the proposer.
* It provides the full details about the insurance policy and related information about the insured pet, agents, and proposer etc.
* The main objective of the developed system is to allow admin users to register insured pet with their name, date of birth, residence address, and also policy details. After giving registering all the insured pet, website should provide management facilities like delete unwanted pets data.
* It provide facilities like verify claim, verify agent, verify policy and verify feedback for the admin.
* Efficiency with respect to time and work.
* This project reduces workload of proposer and admin.

# **REQUIREMENT ANALYSIS**

## **Existing system**

* Now a days the firm is doing its all work manually.
* Any user can view the information about pet.
* All entries regarding insurance of pets are done manually in register.
* Now the firm wants all its work computerized.
* With using manual system it is very difficult to maintain the register.
* It is Time consuming.
* It is very slow process.
* It is difficult to retrieve the data.
* It is difficult to find the error.
* In existing system there is chance of data redundancy that is data may be recorded twice.
* It is difficult to maintain large numbers of records manually.
* Less Security.

## **Proposed system**

* Easy to retrieve any kind of data.
* Reduce the time.
* Easy to store client information, pet information etc.
* System Security and Authorization.
* Reduce the paper work.
* The proposed system is user friendly.
* The system will accept the necessary inputs and generates the required output in the form of report.
* Allows user to get registered from their places and transact for the required insurance policy.
* The system makes the overall project management much easier and flexible.
* Huge amount of data can be stored in a single system, entries can be made very quick and accurate. Hence time can be saved.

# **SYSTEM ANALYSIS**

**Modules present**

This project has following modules.

## **Admin Module**

Admin is a super user of our website. Admin has the power to verify policy, verify agent, verify claim. Admin can view the feedback posted in this website and he can also delete the feedback if he is not interested with it.

## **Proposer module**

Proposer role is to first register into the site and then he has to login through a given page. After he login on the page he will move to buy policy, then he must select the pet which he wanted to insure, a page of payment details and challan will be displayed. Proposer has to take the printout of the displayed challan an he has to pay the displayed amount through the challan in the bank.

## **Agent module**

Agent must register to the given registration page. After registration, only if the admin permits he can be the agent and his name will be displayed on the ‘Know Agent’ page.

# **HARDWARE & SOFTWARE REQUIREMENT**

## **Hardware requirements**

**Server**

Platform : Windows 7 and higher

System : Pentium V 2.8 GHz

RAM : 2GB

Hard disk : 200GB

**Client**

Platform : Windows 7 and higher

System : Pentium V 2.8 GHz

Ram : 2.1 GB

Hard disk : 200GB

## **Software requirements**

Web server : XAMPP Server

Back end : MYSQL

Server side scripting : PHP

Client side scripting : HTML

# **TECHNOLOGIES USED**

## **Hyper text markup language(html)**

**Hyper Text Markup Language** is a basis for creating web page. The internet is latest technology for exchanging information. Even the corporate sector is investing a large amount of money in creating a website and making their presence felt. Hyper Text Markup Language was envisioned to be format for allowing people using different computers to share information seamlessly over the network.

A markup language such as Hyper Text Markup Language is simple a collection of codes called elements which are used to indicate the structure and often the document. A web browser that renders the document interprets the meaning of these codes to figure out how to structure or display a document. Is a Hyper Text Markup Language used to write hypermedia documents for the world wide web(www). Hyper Text Markup Language is a subnet of generalized markup language.

**Hypertext Preprocessor (PHP)**

**What is PHP?**

* PHP is an acronym for "PHP: Hypertext Preprocessor"
* PHP is a widely-used, open source scripting language
* PHP scripts are executed on the server
* PHP is free to download and use

**What is a PHP File?**

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code are executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

**Why PHP?**

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases
* PHP is free.
* PHP is easy to learn and runs efficiently on the server side

## **Mysql**

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).

Databases 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, Supervisors, 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.

**MySQL is a database.**

The data in MySQL is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows. Databases are useful when storing information categorically. A company may have a database with the following tables: "Employees", "Products", "Customers" and "Orders".

## **Javascript**

JavaScript is a programming language used to make web pages interactive. It is what gives a page life—the interactive elements and animation that engage a user. If you've ever used a search box on a home page, checked a live baseball score on a news site, or watched a video, JavaScript is probably running through it.

JavaScript, on the other hand, is a text-based programming language meant to run as part of a web-based application. When first developed, it was intended to be a compliment to Java. But JavaScript took on a life of its own as one of the three pillars of web development—the other two being HTML and CSS. Unlike Java applications, which need to be compiled before they can run in a web-based environment, JavaScript was purposely designed to integrate into HTML. All major web browsers support [JavaScript](https://www.thoughtco.com/javascript-programming-4133476), though most give users the option of disabling support for it.

JavaScript is designed to run within the HTML structure of a website and is often used multiple times. If you're writing code, your JavaScript will be more easily accessed if you place them in separate files. (Using a .JS extension helps identify them.) You then link the JavaScript to your HTML by inserting a tag. The same script can then be added to several pages just by adding the appropriate tag into each of the pages to set up the link.

## **Css (cascading style sheets)**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**Advantages of CSS**

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. Soits a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
* **Offline Browsing** − CSS can store web applications locally with the help of an offline catche.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.
* **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

## **D3.js**

D3.js is a JavaScript library used to create interactive visualizations in the browser. The D3.js library allows us to manipulate elements of a webpage in the context of a data set. These elements can be **HTML, SVG,** or **Canvas elements** and can be introduced, removed, or edited according to the contents of the data set. It is a library for manipulating the DOM objects. D3.js can be a valuable aid in data exploration, it gives you control over your data's representation and lets you add interactivity.

**D3.js Features**

D3.js is one of the best data visualization framework and it can be used to generate simple as well as complex visualizations along with user interaction and transition effects. Some of its salient features are listed below:

* Extremely flexible.
* Easy to use and fast.
* Supports large datasets.
* Declarative programming.
* Code reusability.
* Has wide variety of curve generating functions.
* Associates data to an element or group of elements in the html page.

**D3.js Benefits**

D3.js is an open source project and works without any plugin. It requires very less code and comes up with the following benefits −

* Great data visualization.
* It is modular. You can download a small piece of D3.js, which you want to use. No need to load the whole library every time.
* Easy to build a charting component.
* DOM manipulation

# **SOFTWARE REQUIREMENT SPECIFICATION**

## **Introduction**

This is the Software Requirement Specification (SRS) of the Project “PET INSURNACE”. Here it is used to specify the requirement for the initial implementation of the system.

Software Requirement Specification bridges the gap between the client or the user and the system developer. This document describes the user needs accurately.

SRS is the official statement of what is required of the system developers and it includes both user requirements for the system and detailed specification of system requirements.

## **Purpose**

The purpose of this report is to define the requirement of the “PET INSURANCE”. In detail, this report will provide the general description of the project, product perspective and overview of requirements, general constraints and a user view of product use. In addition, it will also provide the specific requirements and functionalities needed for this project such as interface, functionality and performance requirements. The purpose of the Software Requirement Specification is to properly document the requirement of the user necessary to build this application.

## **Functional requirements**

**Admin:**

* Admin can view and verify the policy.
* Admin can view and verify the claim.
* Admin can view and verify the agent.
* Admin can delete the feedback if he does not want to display it .

**Proposer:**

* Proposer need to register with valid details to get benefits from this website.
* Proposer can see his activated policy.
* Proposer can track his claim of the activated policy.
* Proposer can view the agent details.
* Proposer can give his feedback.

**Agent:**

* Agent needs to register with his valid details to become the agent of the insurance company.

## **Non-functional requirements**

**Correctness:**

Since this project is used to provide the actual and correct information about details of the particular project which were done by students. The admin will have more privileges on the database; the system should always provide correct response and the data in all the database should always be constantly updated with the latest information.

**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.

**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.

**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.

**Security:**

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

# **SYSTEM DESIGN**

## **Dfd\_0 level:**

Proposer

Admin

## **Dfd\_2 level:**

Customer registration

Proposer

Insured pet

Claim policy

Agent

## **DFD-2 level:**

Admin

Claim policy

Admin

Agent

Feedback

# **DATABASE TABLE**

**Admin:**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Username | Varchar(50) |
| Password | Varchar(100) |

**Agent:**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Agent-id | Int(11) |
| Fname | Varchar(200) |
| Mname | Varchar(200) |
| Lname | Varchar(100) |
| Dob | Date |
| Gender | Varchar(10) |
| Full\_address | Varchar(500) |
| State | Varchar(200) |
| City | Varchar(100) |
| Mobile\_no | Varchar(10) |
| Email | Varchar(100) |
| Access | Int(11) |

**Claim\_policy**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Claim\_id | Int(11) |
| Insurance\_id | Varchar(20) |
| Claim\_date | Date |
| Reason | Varchar(100) |
| Message | Text |
| Status | Int(11) |
| Admin\_message | Text |

**Customer\_registration:**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Customer\_id | Int(11) |
| Fname | Varchar(50) |
| Mname | Varchar(50) |
| Lname | Varchar(50) |
| Dob | Date |
| Gender | Varchar(10) |
| Full\_address | Varchar(200) |
| State | Varchar(50) |
| City | Varchar(50) |
| Mobile\_no | Varchar(15) |
| Email | Varchar(100) |
| Username | Varchar(100) |
| Password | Varchar(50) |
| Cpassword | Varchar(50) |

**Payment:**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Id | Int(11) |
| Challan\_no | Varchar(50) |
| Customer\_id | Varchar(100) |
| Insurance\_id | Varchar(20) |
| Payment\_date | Date |

**Insured\_pet:**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| Active | Int(11) |
| Insurance\_id | Varchar(20) |
| Pet\_type | Varchar(50) |
| Customer\_id | Int(11) |
| Pet\_name | Varchar(100) |
| Pet\_gender | Varchar(10) |
| Pet\_SN | Varchar(5) |
| Pet\_dob | Date |
| Pet\_cost | Int(11) |
| Is\_injured | Varchar(5) |
| Is vet | Varchar(5) |
| Is\_insured | Varchar(5) |
| Invest | Varchar(100) |
| Term | Int(11) |

# **SYSTEM IMPLEMENTATION**

**System Implementation**

The implementation is the stage of the project where the theoretical design is converted into working system. The goal is to provide the logical order for a creation of the modules and creation of the system.

The implementation of any software requires perfect satisfaction of the user. The proposed system takes into account the various difficulties faced by the users in operating system and provides user friendly operations.

The system is implemented in the server and checked for its performance and accuracy. Successful implementation of the software is accomplished. This project is to be done using PHP as front end and My-SQL as back end. This project use HTML controls.

Implementation activities are primarily environmental. They deal with the realities of particular machines, systems, languages, compilers, tools, developers, and clients necessary to translate a design into working code.

After the system is implemented and conversion is complete, user usually conducts a review of the system. The new system needs less man power and provides information on time. Saves data entry duplication work. It also provides the locking system and password protection so it is reliable.

## **Coding**

**Login Page:**

<?php

session\_start();

if(isset($\_GET['adminlogout']))

{

session\_unset();

session\_destroy();

session\_write\_close();

setcookie(session\_name(),'',0,'/');

session\_regenerate\_id(true);

header("location:cpanel.php");

}

if(isset($\_GET['logout']))

{

session\_start();

session\_unset();

session\_destroy();

session\_write\_close();

setcookie(session\_name(),'',0,'/');

session\_regenerate\_id(true);

header("location:index.php");

}

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta http-equiv="Content-Type"content="text/html; charset=utf-8" />

<title>Pet insurance - login</title>

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

<link href="./CSS/styles1.css" rel="stylesheet" type="text/css" />

<meta charset='utf-8'>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="CSS/stylesheet.css">

<script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>

<script src="js/script.js"></script>

</head>

<body>

<div class="head-container"><div id=logo><imgsrc="res/Logo.jpg"/></div>

</div>

<div id='cssmenu'>

<ul>

<li class='mnu-home active'><a href='index.php'><i class="white fa fa-home"></i>Home</a></li>

<li><a href='agent.php'><i class="white fa fa-photo"></i>Agent Registration</a></li>

<li CLASS="has-sub"><a href='#'><i class="white fa fa-user-plus"></i>Proposer</a>

<ul>

<li><a href="User\_Registration.php" >Registration</a></li>

<li><a href="view\_agent\_customers.php" >Know Agents</a></li>

</ul></li>

<li CLASS="has-sub"><a href=''><i class="white fa fa-graduation-cap"></i>Buy Policy</a>

<ul>

<?php

$mysqli = new mysqli("localhost", "root", "", "Insurance");

if($mysqli->connect\_error)

{

die("Connection Faild ".$mysqli->connect\_error);

}

$sql="select \* from pet";

$result1=$mysqli->query($sql);

if($result1->num\_rows>0)

{

while($row1=$result1->fetch\_assoc())

{

echo'<li><ahref="buyinsurance.php?pid='.$row1['pid'].'">'.$row1['title'].'</a>

</li>';

}

}

?>

</ul></li>

<li><a href='feedback\_form.php'><i class="white fa fa-pencil-square-o"></i>Feedback</a></li>

<?php

if(isset($\_SESSION['username']))

echo'<li><a href="proposer\_panel.php"><i class="white fa fa-address- card"></i>Proposer Panel</a>

</li>';

?>

<li class='has-sub'><a href='#'><i class="white fa fa-info-circle"></i>About Us</a>

<ul>

<li><a href='About\_us.php'>About Us</a></li>

<li><a href='Notification.php'>Rules and Regulations</a></li>

</ul></li>

<li><a href='Contact\_us.php'><i class="white fa fa-phone-square"></i>Contact Us</a></li>\

<?php

if(isset($\_SESSION['username']))

echo'<li><a href="login.php?logout=1"><i class="white fa fa-sign-in"></i> Logout</a>';

else

echo'<li><a href="login.php"><i class="white fa fa-sign-in"></i> Login</a>

</li>';

?>

</ul>

</div>

<body>

<?php

if(isset($\_GET['pid']))

{

$pid=$\_GET['pid'];

$type=$\_GET['type'];

}

if(!isset($\_POST['btn\_submit']))

{

echo'

<form method="post" action="">

<div class="main-container">

<div class="form-style">

<div class="left-container"></div>

<div class="right-container">

<div style="padding:20px;">

<label style="width:100%"><span style="text-align:right">Username:</span><input type="text" class="input-field" name="user\_name" value="" /></label>

<label style="width:100%"><span style="text-align:right">Password:</span><input type="password" class="input-field" name="password" value="" /></label>

<br>

<br>

<label style="width:100%; min-width:50px; text-align:center"><span style="text-align:right" ></span><input type="submit" style="width:80%;" name="btn\_submit" value="Login" /></label>

<div style="text-align:right">

Not <a href="User\_Registration.php">Registered?</a>

</div>

</div>

</div>

</div>

</div>

</form>';

}

else

{

$mysqli = new mysqli("localhost", "root", "", "Insurance");

if($mysqli->connect\_error)

{

die("Connection Faild ".$mysqli->connect\_error);

}

$username=test\_input($\_POST['user\_name']);

$password=test\_input($\_POST['password']);

$sql="select \* from customer\_registration where user\_name='".$username."' and password='".$password."'";

$result1=$mysqli->query($sql);

if($result1->num\_rows>0)

{

$\_SESSION['username']=$username;

$\_SESSION['password']=$password;

if(isset($\_GET['pid']))

{

header("location:buyinsurance.php?pid=".$pid."&type=".$type);

}

else

header("location:proposer\_panel.php");

}

else{

echo'<script type="text/javascript">alert("Incorrect Username or Password"); window.location="login.php";</script>';

}

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}?>

# **TESTING**

Testing is the integral part of software development. Poor testing methodologies lead to unstable products and unpredictable development times. It is very essential to have a testing plan in place to ensure that the product delivered is robust and stable and is delivered in a predictable timeline.

Standard development methodologies describe a set of general testing mechanisms which must be incorporated in the product development lifecycle. These mechanisms start from testing very small of code piece by piece to testing the whole application functionality in the end.

1. Unit Testing
2. Integration Testing
3. System Testing
   * Validation Testing
   * Black Box Testing
   * White Box Testing
4. Acceptance Testing

## **Test cases and Results**

**Login Page**

|  |  |  |
| --- | --- | --- |
| **Input** | **Result** | **Conclusion** |
| 1.User name : User Input  2.Password : User Input | Admin and Proposer are allowed to access the system or an error message is displayed. | A user has to give a valid username and password in order to access the system. If user name or password is incorrect then error message is displayed. |

**Proposer Registration Page**

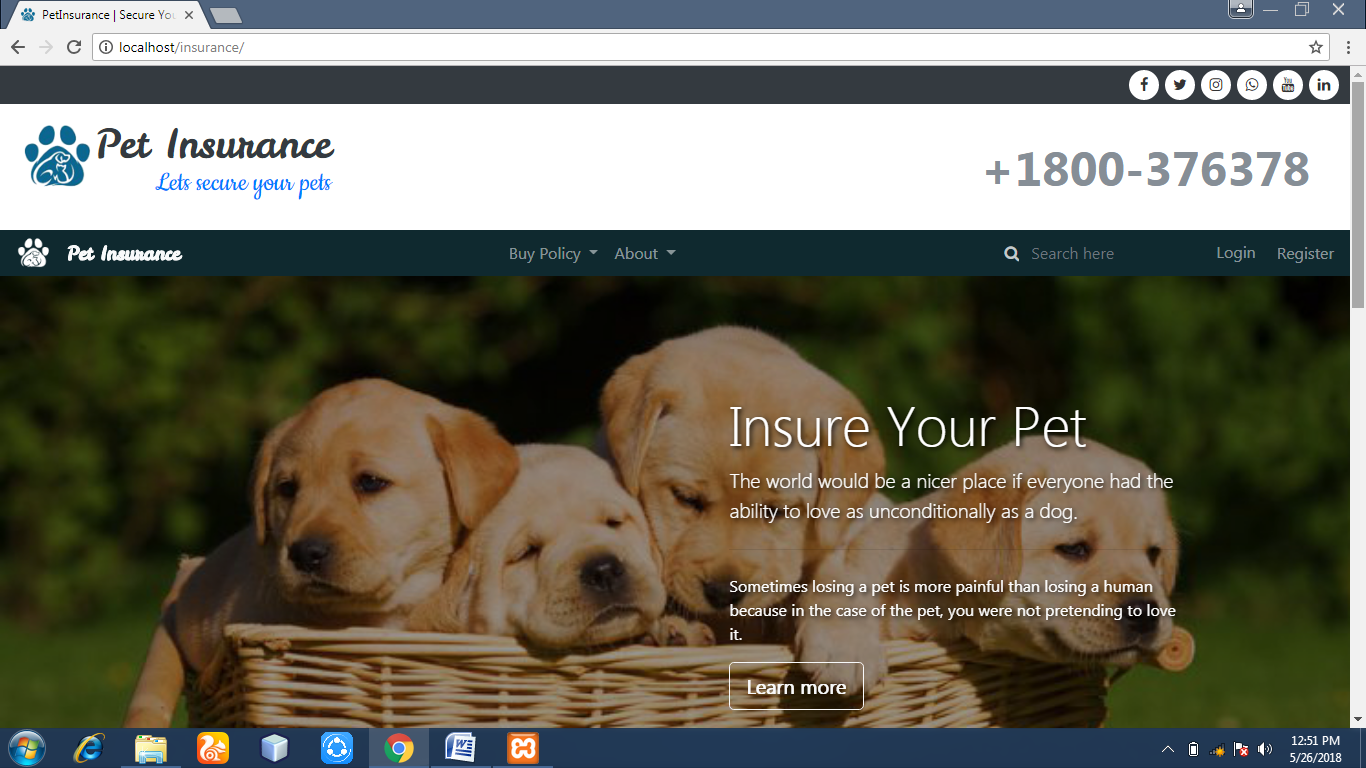
|  |  |  |
| --- | --- | --- |
| **Input** | **Result** | **Conclusion** |
| 1.Proposer Name: User Input  2.Date of Birth: User Input  3.Gender: User Input  4.Full Address of User: User Input  5.State: User Input  6.City: User Input  7.Mobile No: User Input  8.Email: User Input  9.Username: User Input  10.Password: User Input  11.Confirm Password: User Input | Praposer is registered to the application. | A proposer can be registered to the application if he provided a valid details or otherwise error messageis displayed to provide the valid details. |

**Agent Registration Page**

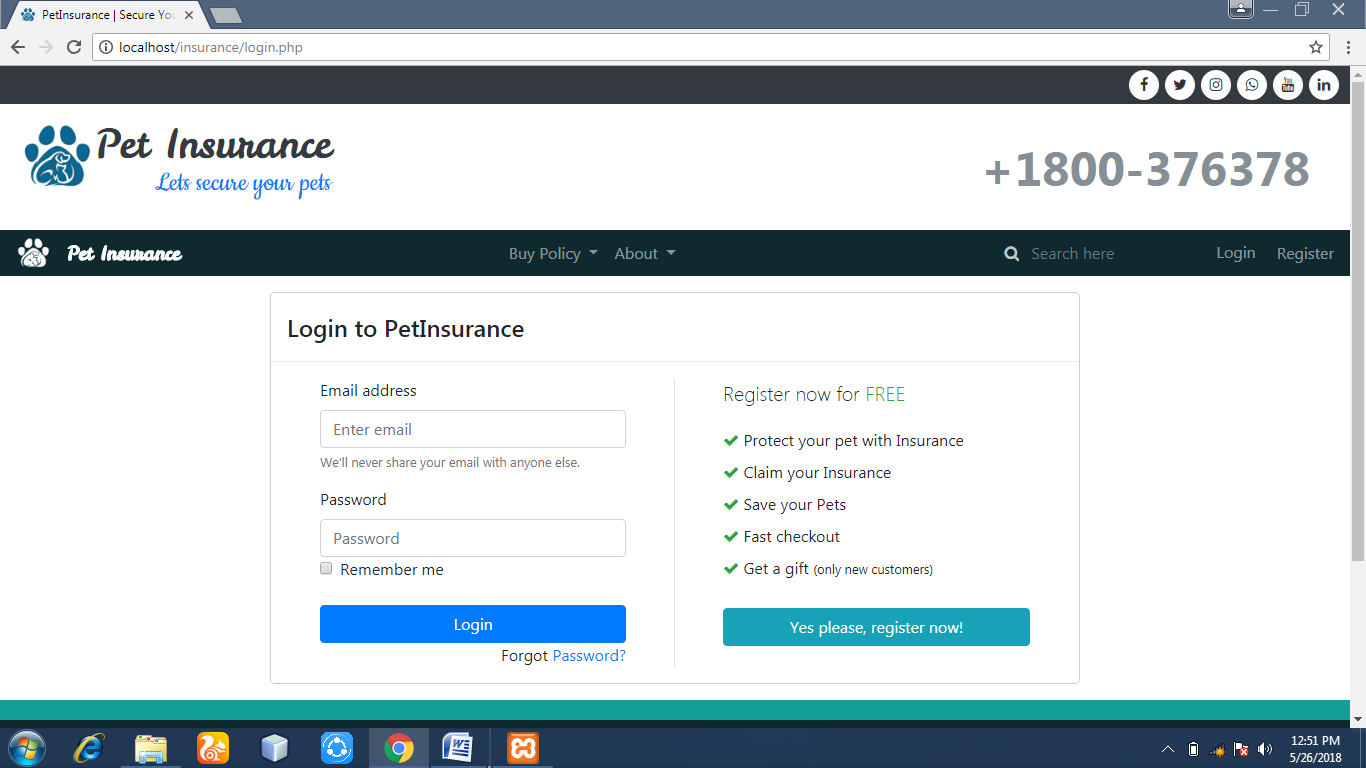
|  |  |  |
| --- | --- | --- |
| **Input** | **Result** | **Conclusion** |
| 1.Agent Name: User Input  2.Date of Birth: User Input  3.Gender: User Input  4.Full Address of User: User Input  5.State: User Input  6.City: User Input  7.Mobile No: User Input  8.Email: User Input | Agent is registered to the application. | Aagent can be registered to the application if he provided a valid details or otherwise error message is displayed to provide the valid details. |

# **SNAPSHOTS**

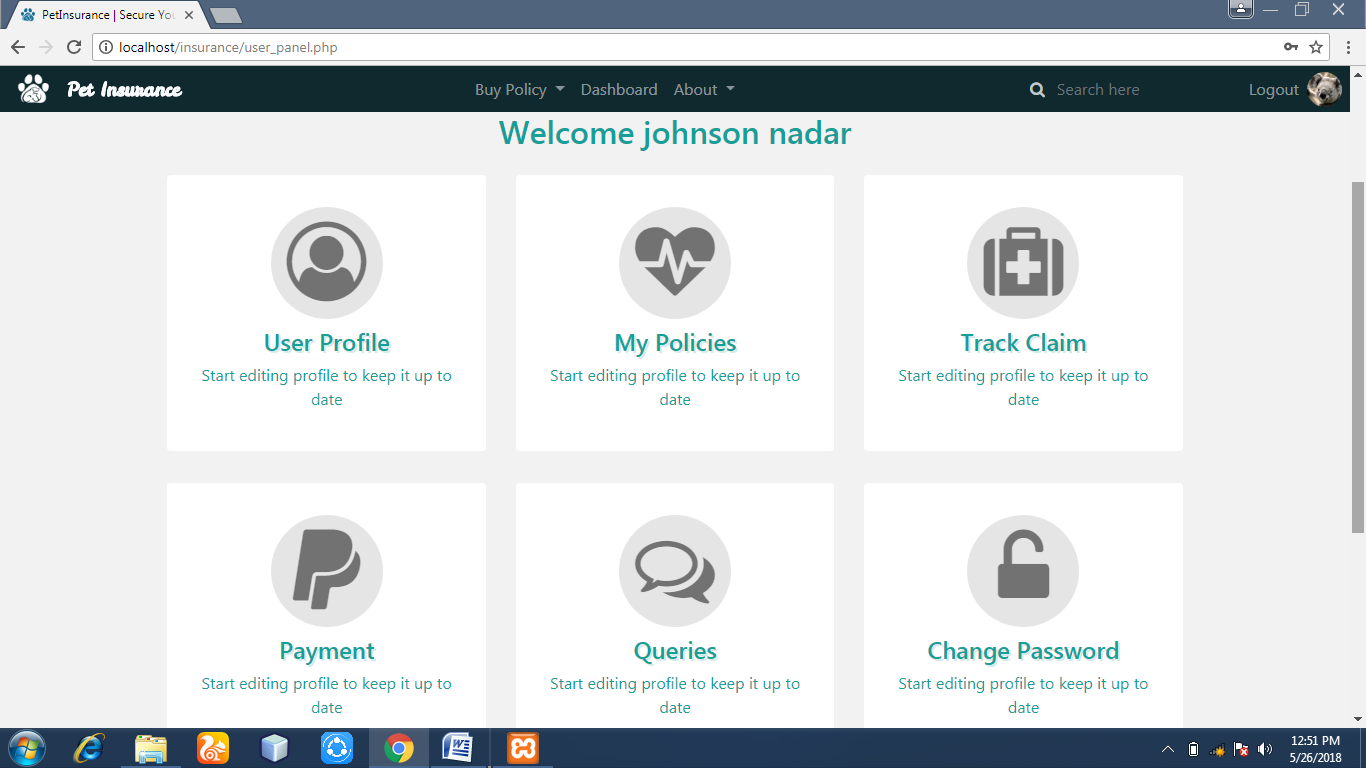
**Home page:**

****

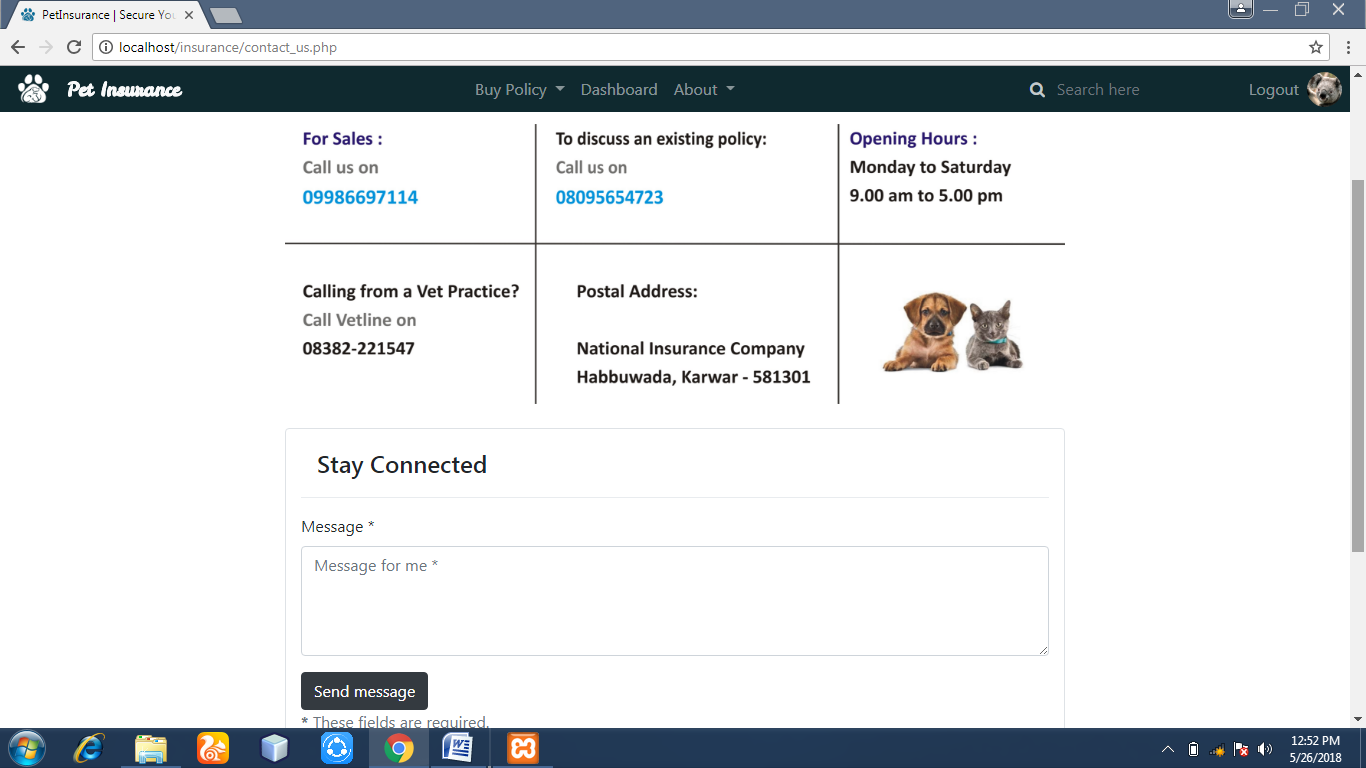
**Login Page:**

****

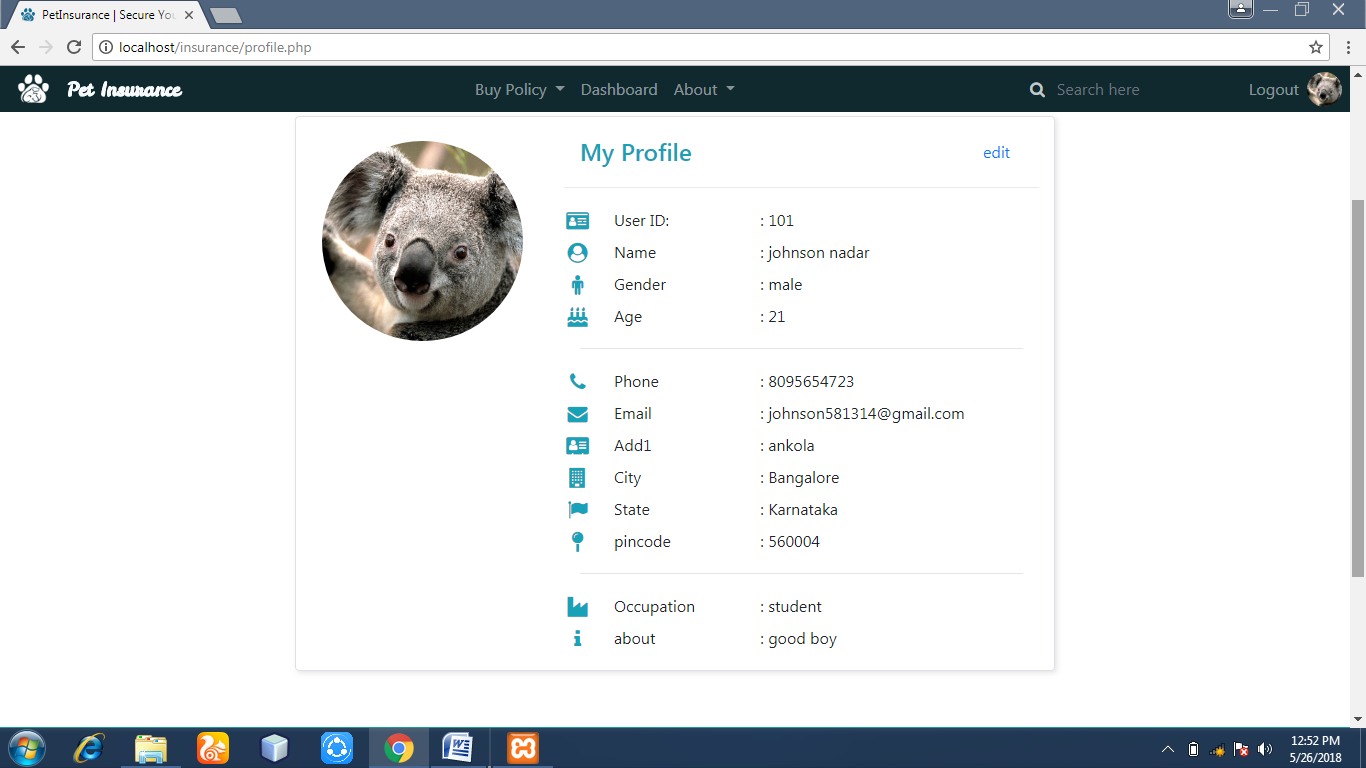
**Proposer Dashboard Page:**



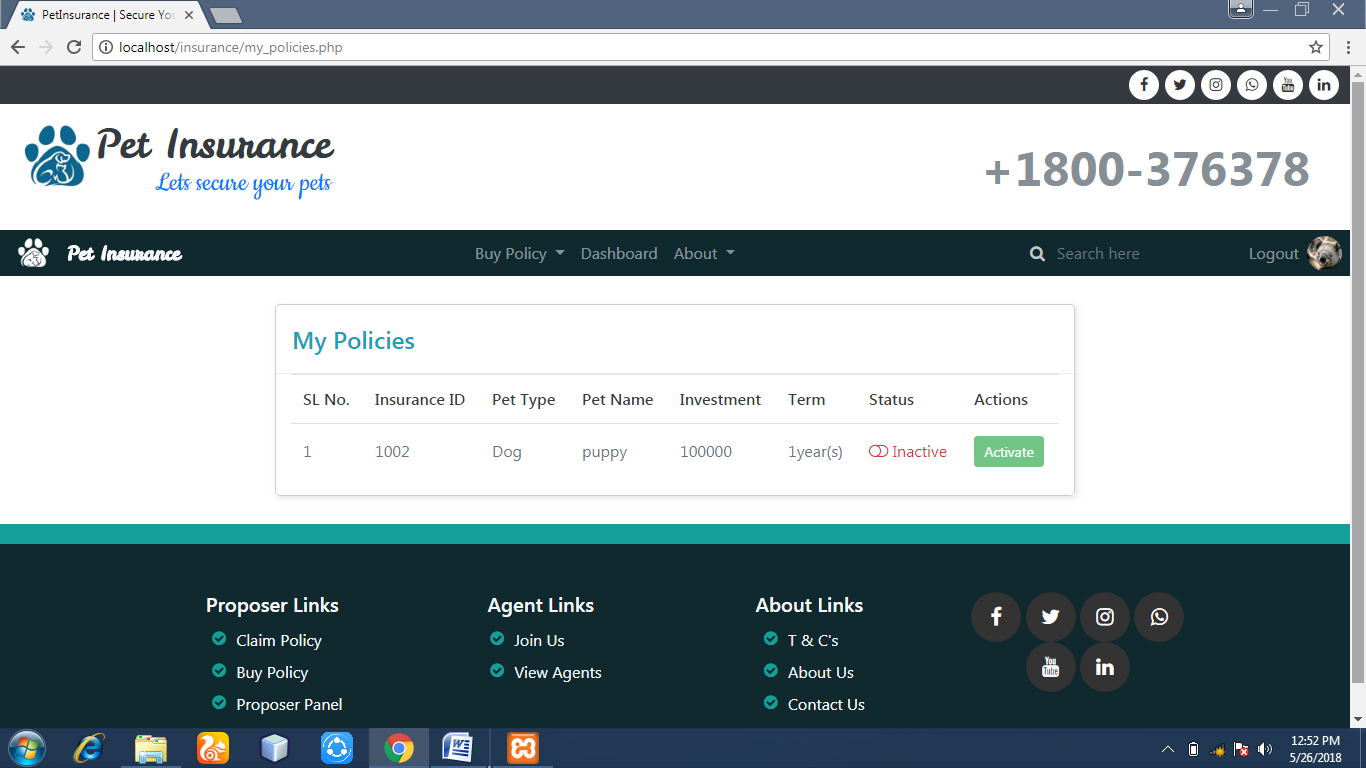
**Contact Us Page:**



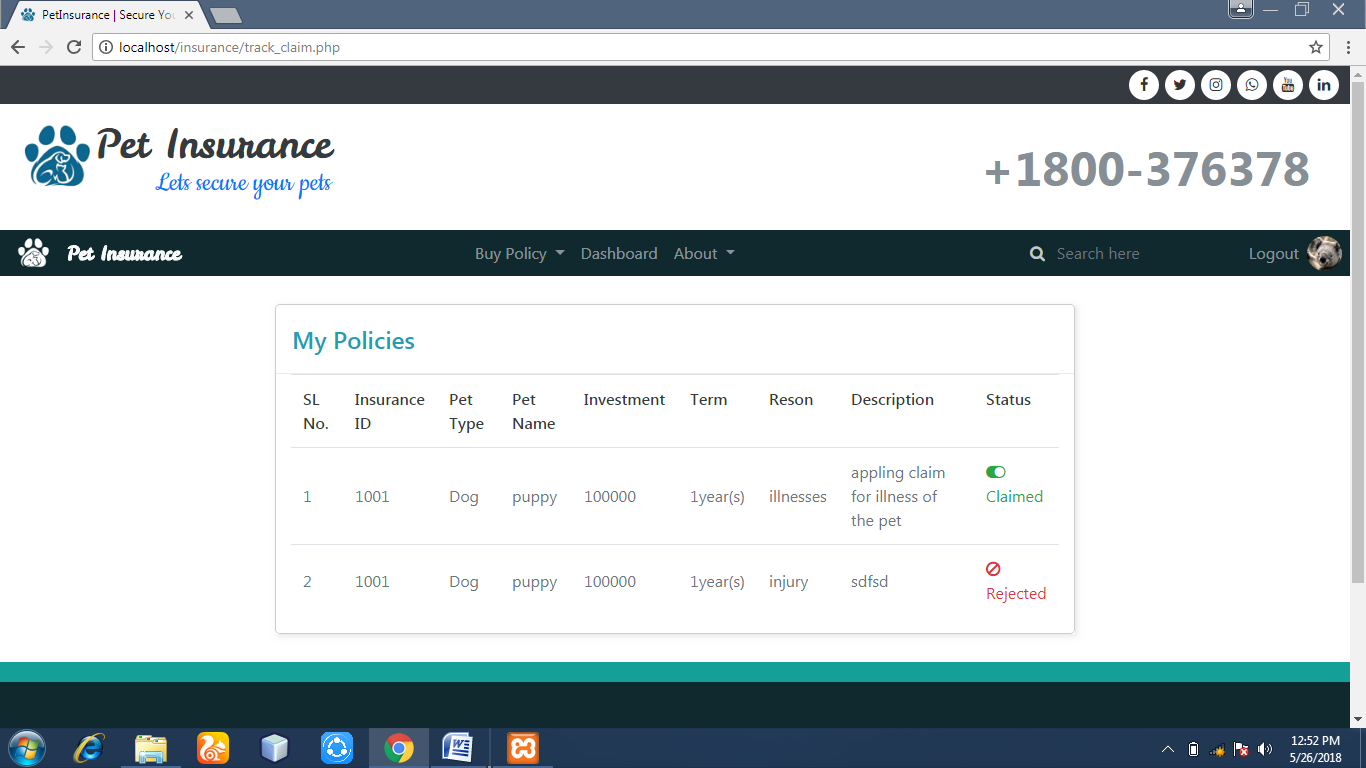
**User Profile Page:**

****

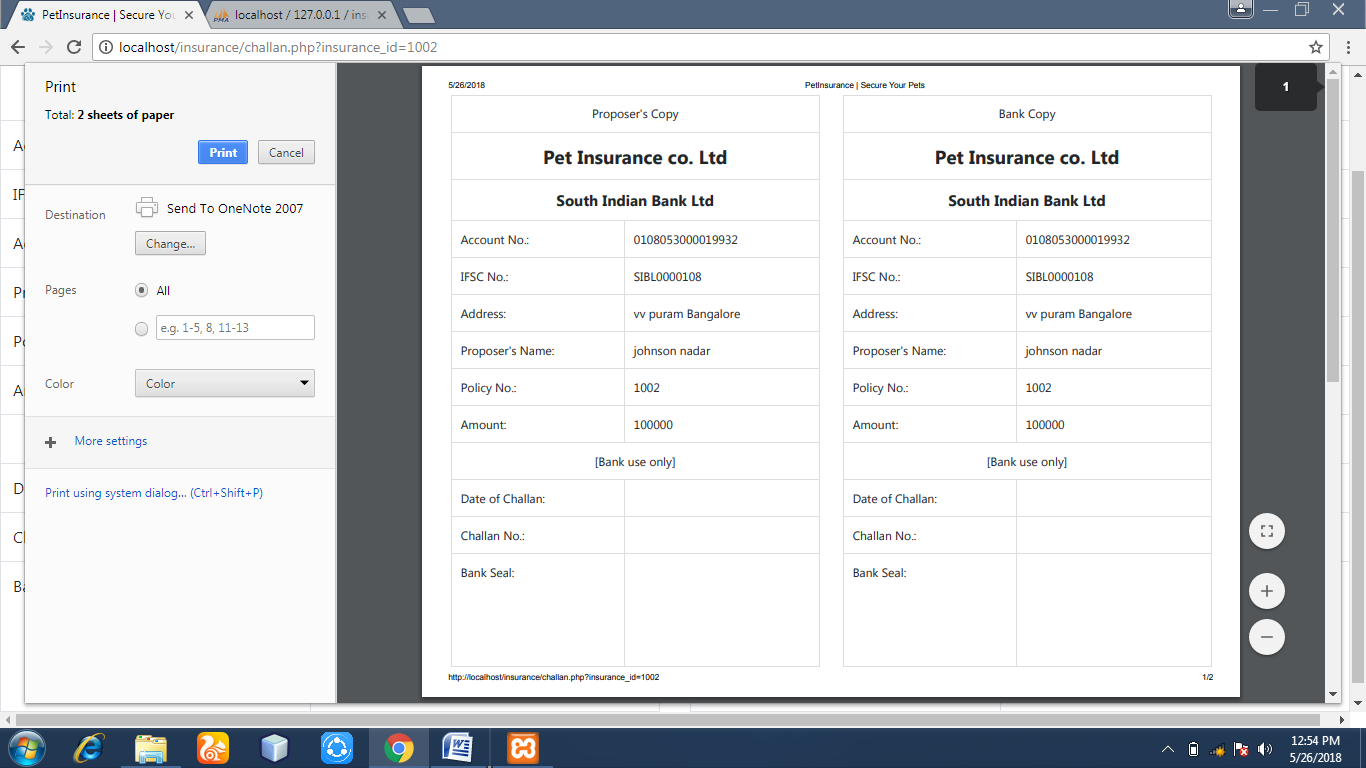
**My Policies Page:**



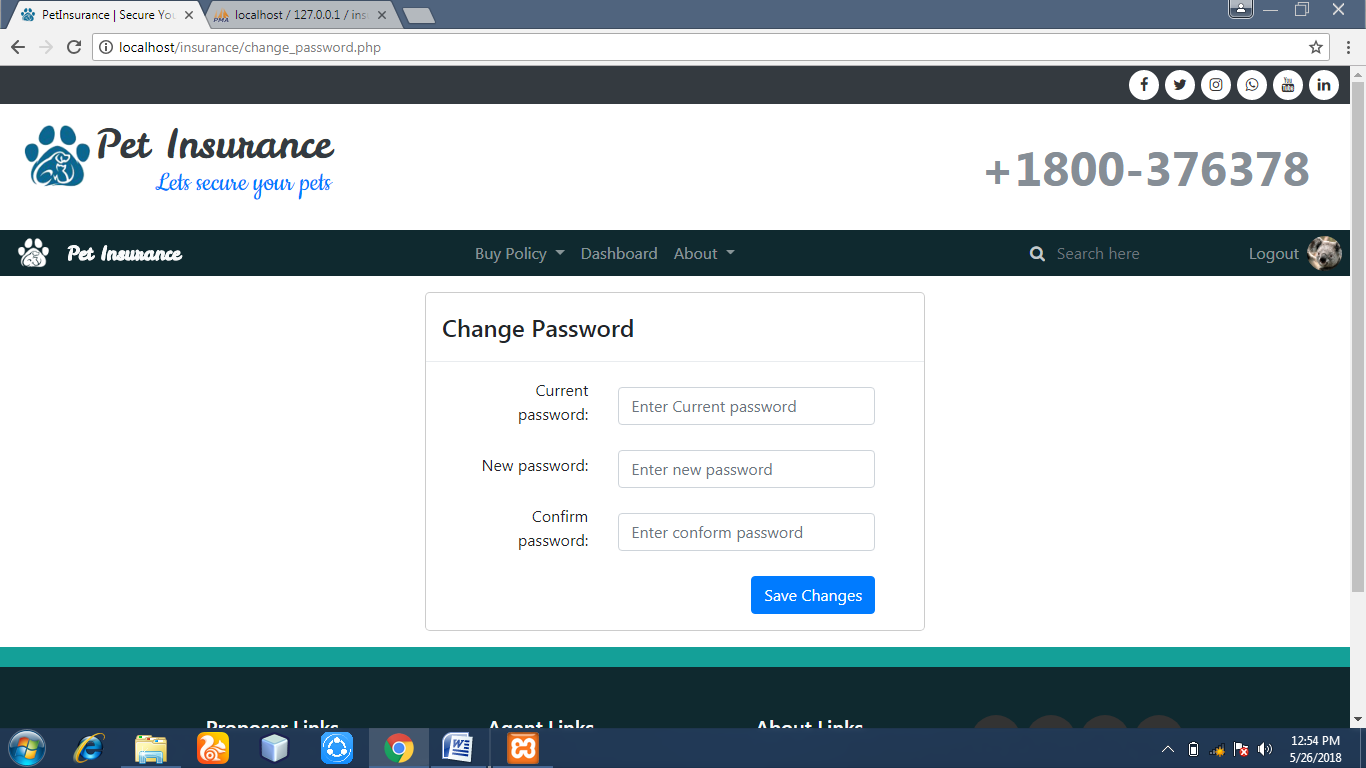
**Claim Page:**

****

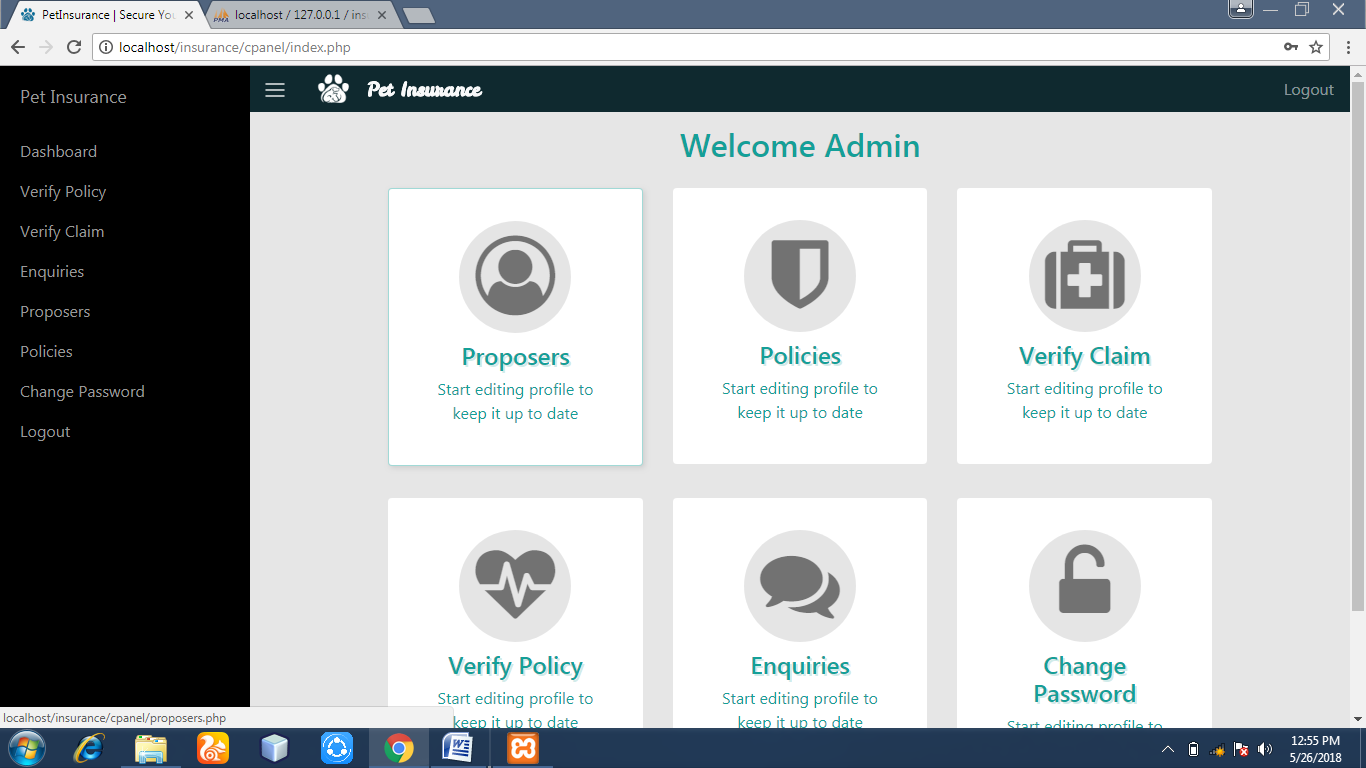
**Print Challan:**



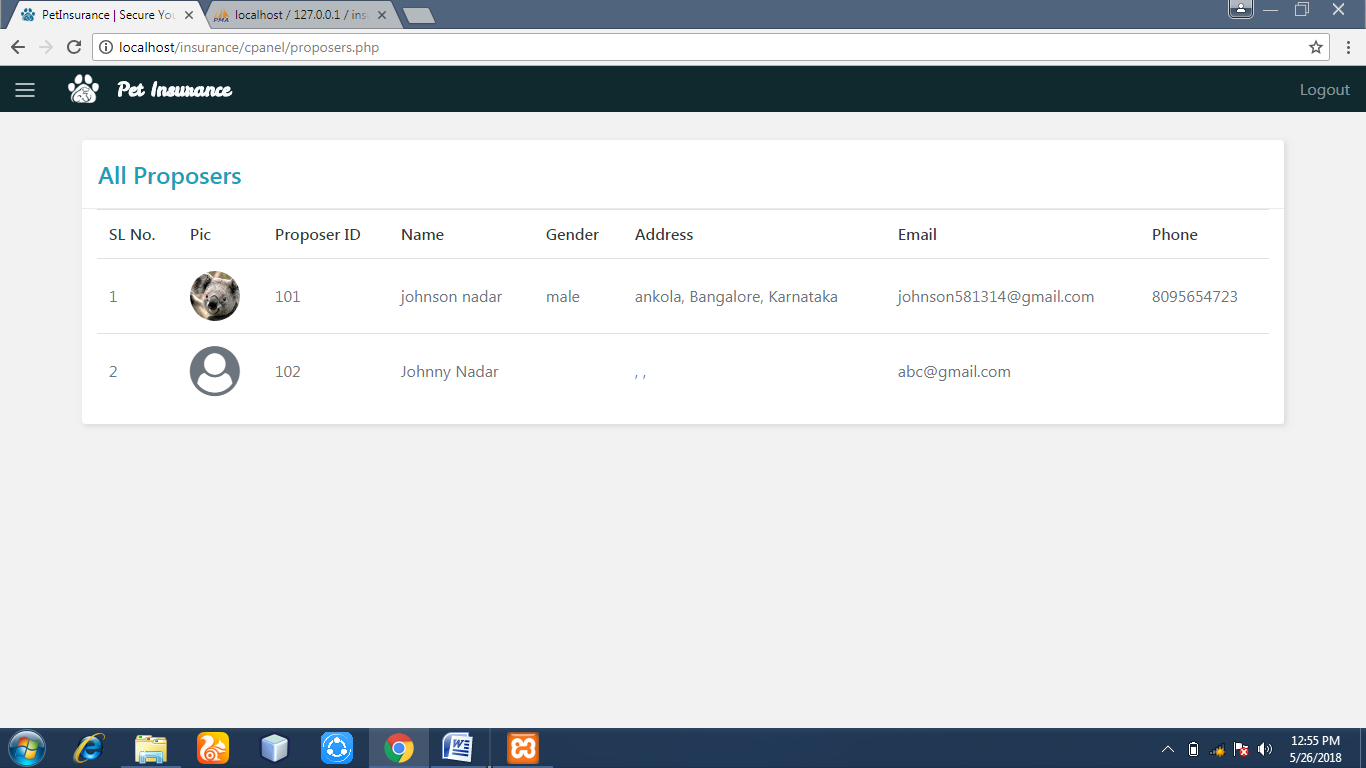
**Change Password**



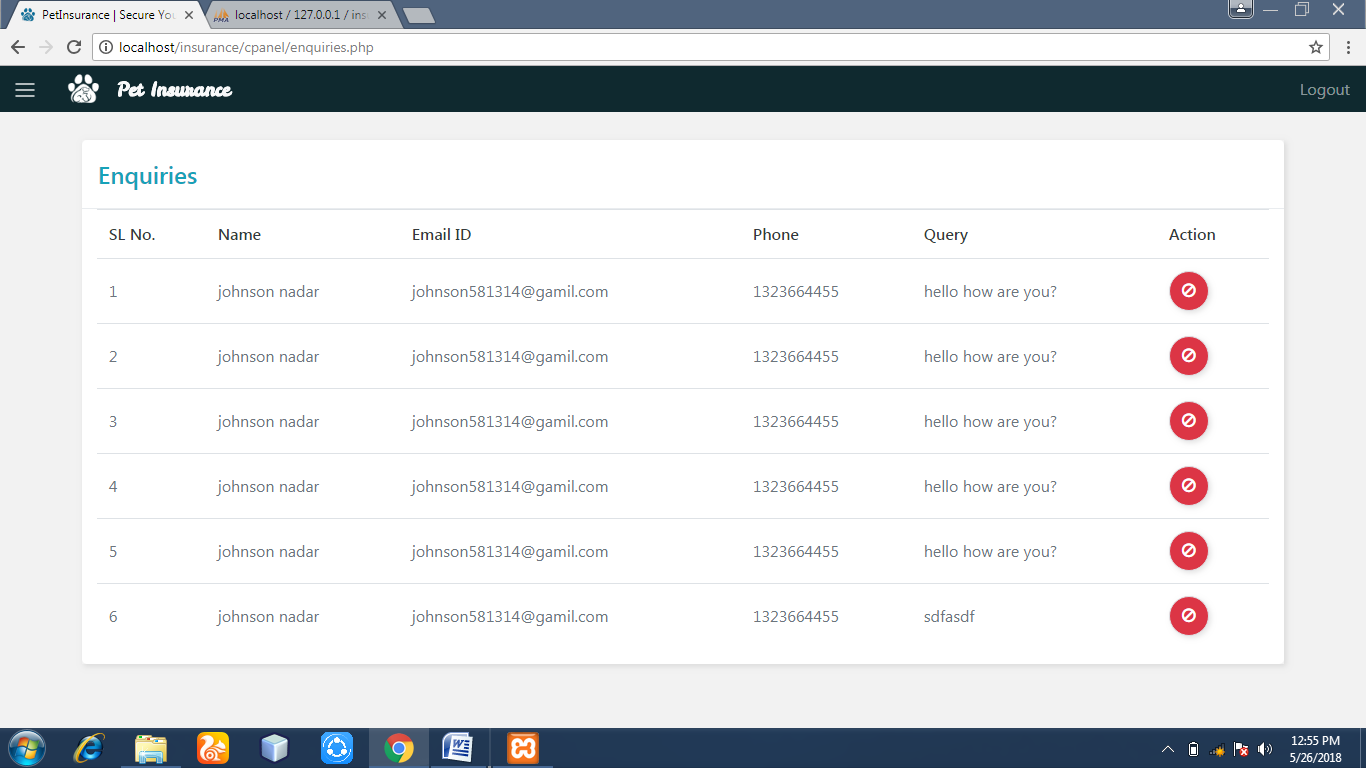
**Admin Dashboard Page:**



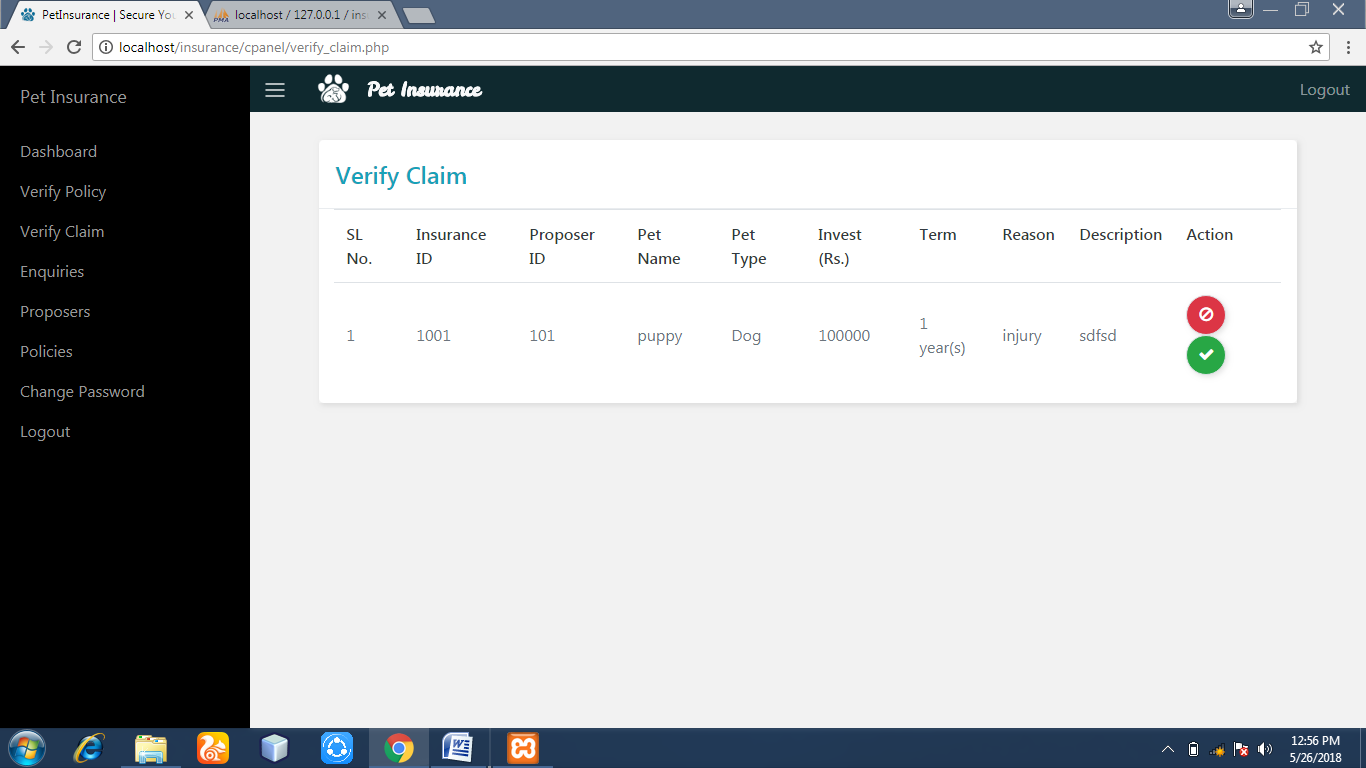
**All Proposers Page:**



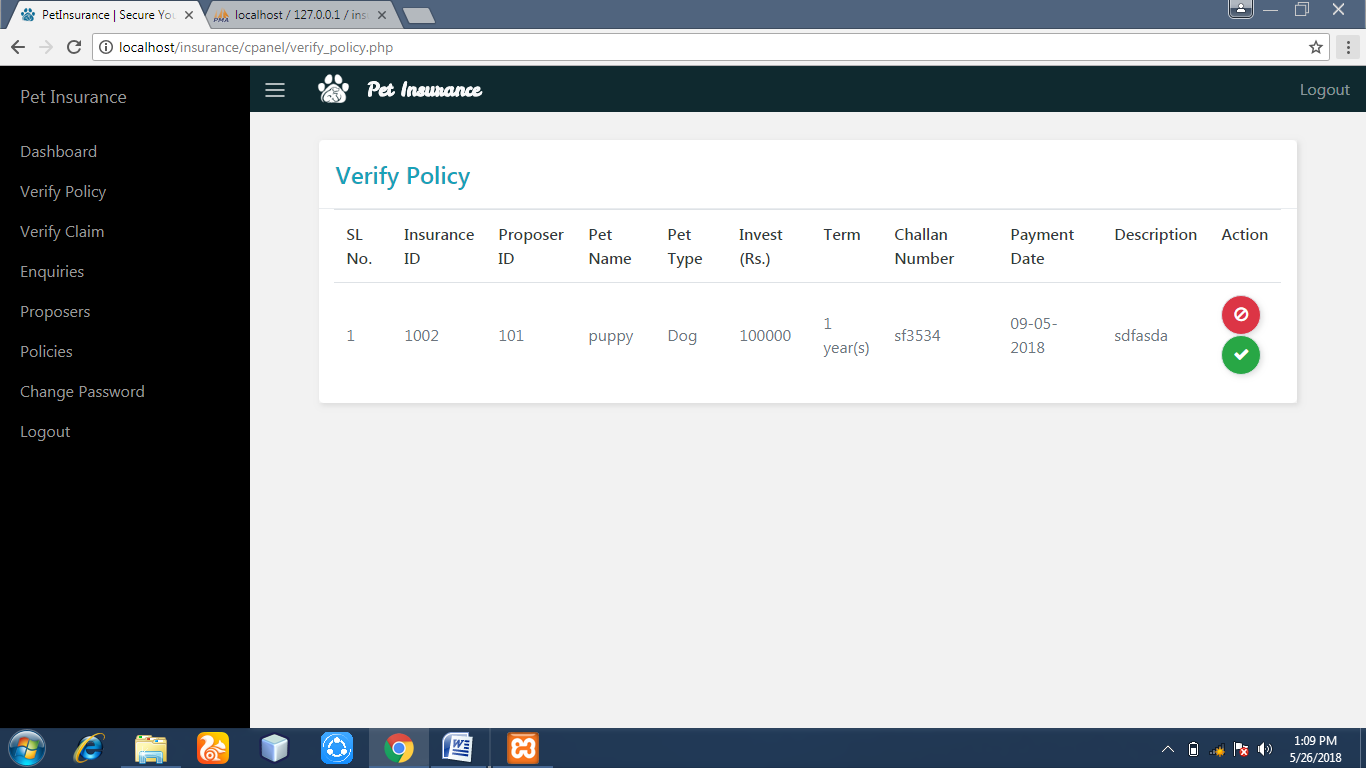
**Enquiries Page:**



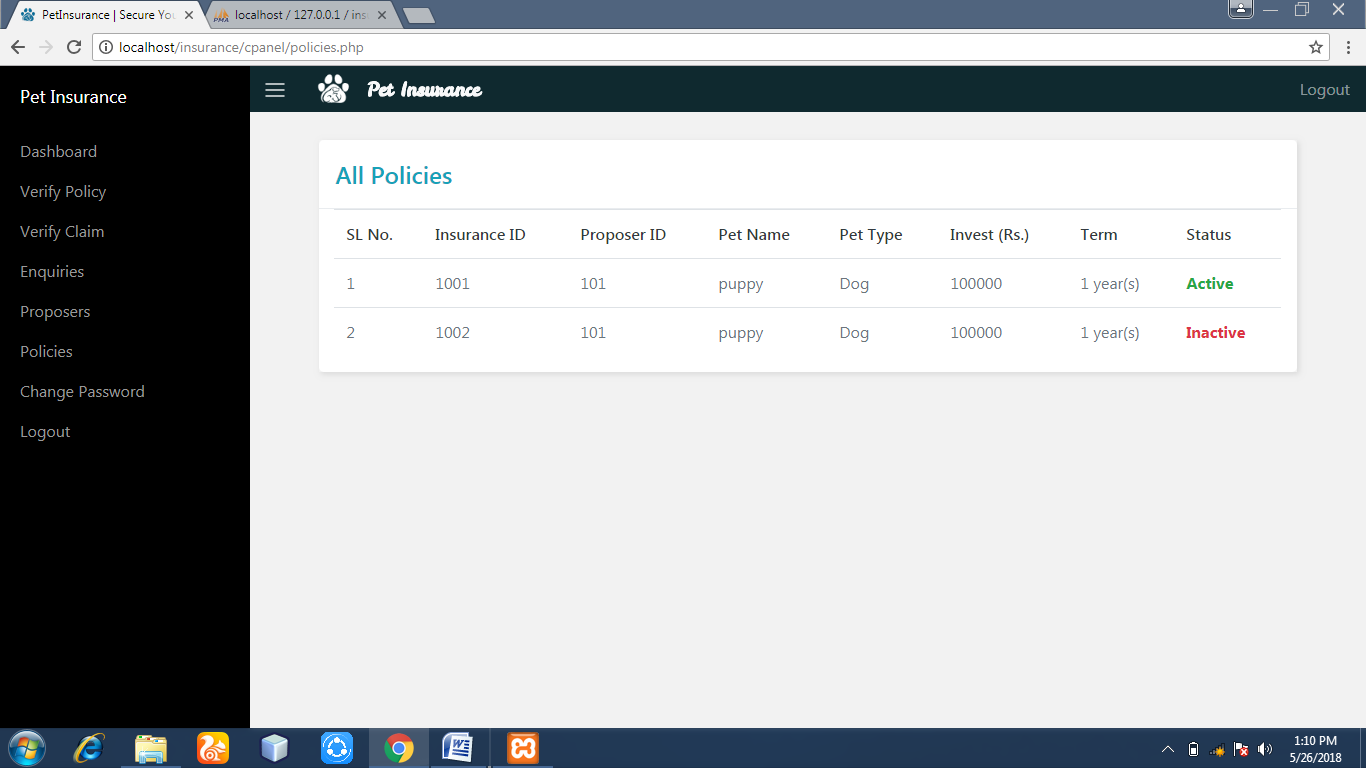
**Verify Claim Page:**



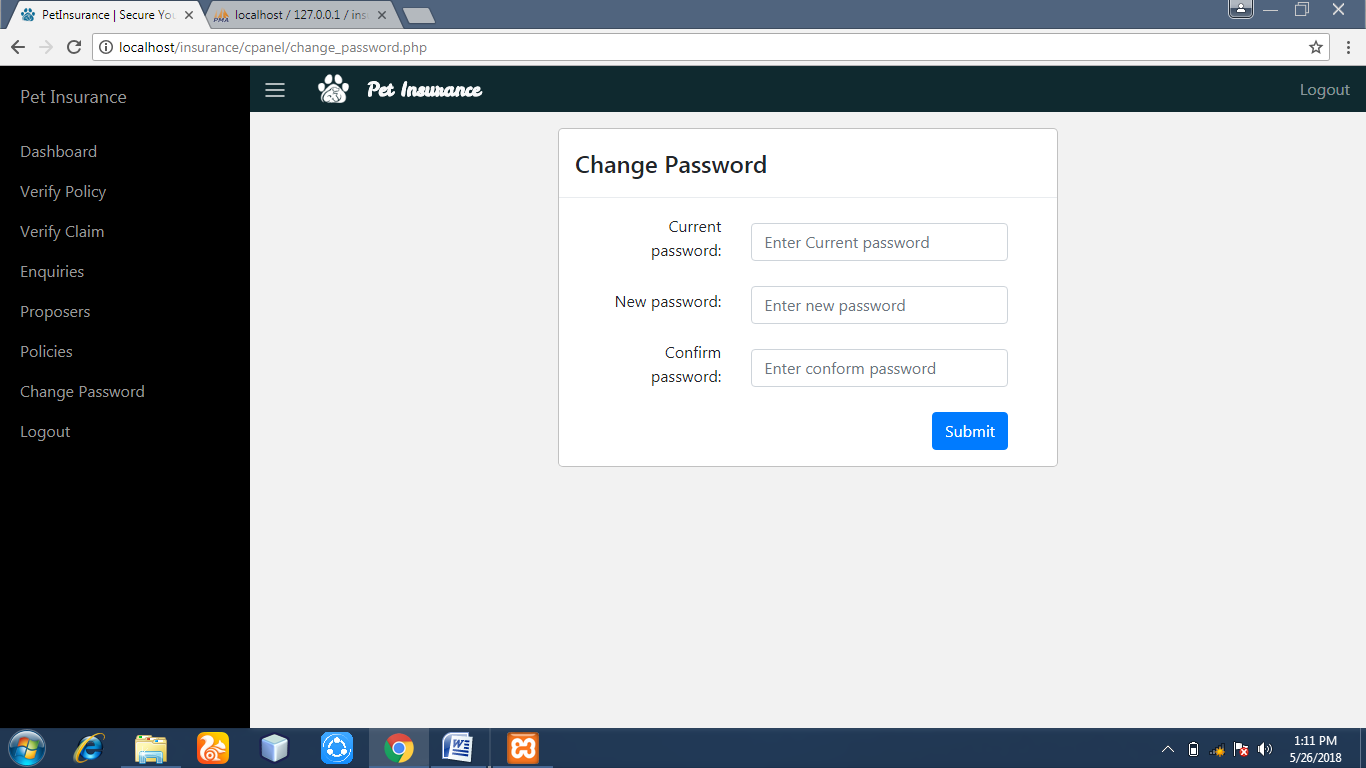
**Verify Policy Page:**



**All Policies Page:**



**Admin Change Password Page:**



# **VALIDATION**

Validation Testing is mainly done to confirm that each value of different fields are falls in their type also check whether any field is blank left i.e ; any field are left without entering any value.

If any fields are blank or empty then message appears like “please fill out this field” ,”your password or username is wrong” etc..

If any fields are to be filled by any character in place of integer then a message appears like “value error”.

# **CONCLUSION**

In this competitive world, faster execution, user satisfaction timely and accurate information have become valuable pre-requisite. This website is developed with the fact in mind, that web based application, are becoming more and more popular.

Our website “PET INSURANCE” is a user friendly web portal designed to bring all the proposer in one platform. Through this project admin can verify policy, claim, agent and feedback. In this website proposer can buy the policy of various pets regularly and be updated with the policy details. Our website is designed keeping in mind that proposer can view the details of his policy very easily.

Through this effortsit may not reach the level of full satisfaction of the proposer, we expect that it nearly satisfy him. The whole experience of working for our project was quite a rewarding one for us as it gave us the opportunity to improve our skills on the finer things.

# **FUTURE ENHANCEMENT**

* In this system, we tried to make a simple application for user.
* We can change our system day by day as per the user requirements will be changed.
* Customer can take policy for pet and give online payment through credit card

# **BIBLIOGRAPHY**

* Bear bibeault, Yehuda katz: jQuery in Action. 3rd Edn, DreamTech India,2008
* RobertW.Sebesta:Programming the Worldwide Web, 4th Edn, Pearson, 2012
* Francis Shanahan:Mashups,WileyIndia, 2012
* Mike Dewar: "Getting Started with D3": O'Reilly Media, 2012
* M.Deitel,P.J.Deitel,A.B.Goldberg:Internet &Internet & World Wide Web How to program,3rd
* Edition, Pearson Education/PHI, 2004

**Web References:**

* <https://www.w3schools.com/pHP/default.asp>
* <https://getbootstrap.com/docs/4.0/getting-started/download/>
* <https://stackoverflow.com/>
* <https://github.com/>
* <https://www.tutorialspoint.com/mysql/index.htm>