SECURE PROGRAMMING FOR WEB

Final Report

2018 – 2019

M. Sc CYBER SECURITY

NATIONAL COLLEGE OF IRELAND

By

Sriram Kalyanraman

X18128394

TABLE OF CONTENTS

Executive Summary

The Website named Caféagape is a platform used by general public customers to know all the coffee types and book their seats in the store. This website was built with major security features with reference to the OWASP top 10 security 2017 amendment. Here the user after providing the user credential and thus the credentials are validated so that the user gets an access to view and book their place. There are two actor functionalities present who has two distinctive roles. The roles of the actors are

- Customer – The customer can login into the website to view and explore the website. New customer can be added into the database and reserve his/her place.
- Admin – The admin has the major role to maintain the website and maintain the database. The admin has a role to view the database contents which was entered by the customer and also the admin can edit the data from the database using the web Graphical User Interface (GUI).

1 Background

This website is majorly focused on security factor. The basic functional features present in the website is to provide the access to the DBMS by the one of the actor. There are two functional actors who has two different roles. The roles of the actors are described below.,

- CUSTOMER: The customer can login into the website to view and explore the website. New customer can be added into the database and reserve his/her place.
- ADMIN: The admin has the major role to maintain the website and maintain the database. The admin has a role to view the database contents which was entered by the customer and also the admin can edit the data from the database using the web Graphical User Interface (GUI).

On a secondary note, this website is also built on business perspective like server and client validation, value authentication, analyzing, session management and to track all the happenings on the website. Every single action performed in the website are store in as a cookie in the user's browser.

1.1 Aims

The aim of the website is to provide a user friendly platform. There are many key features available for the customer actor. The admin actor also has functionality to edit and view the database information. The key functionalities of the website are as follows.,

- Add new user
- Book a place
- Alter the database contents
- Modify Customer details
- Track the happenings on the website

Once the customer logs into the computer or a new user is validated and created, the customer has the full privilege to access all the respective authorized information in the website.

1.2 Technologies

Building a website consist of many technologies involved. The basic requirements of the developer a personal computer with undisruptive internet connection and a web server for hosting the website. The development technologies used in building the website are PHP, HTML, CSS, JavaScript, Bootstrap and MySQL.

1.3 Hardware Architecture

The hardware is the most important requirement for building the website because the website can be compiled and run on a platform which must have a strong foundation on the supporting host system. The website code is built on an IDE(Integrated Development Environment) named

Sublime Text 3 and the code is hosted in a web server which maintains platforms like Apache server, Mysql database and Firezilla.

1.4 Software Architecture

The software architecture is the skeleton of a web application where the technologies that are used in building the website is done to make the website protected. the network, and each of these can have a substantial impact on the quality, performance, maintainability and overall success of the application. The API and technologies used in building the website are

1.4.1 PHP:

PHP Hypertext Pre-processor (PHP) is a programming language which was created for developing a website with basic coded functionalities. For a developer to code website in php, he/she must have a knowledge about server packages and DBMS.

The main features of PHP technology are as follows:

- The PHP algorithm supports HTML and CSS as the client side coding and corelates the functionalities given in the front end.
- PHP is used as a bridge between the client and the server. [1]

1.4.2 MySql Databases:

The MySql database API provides universal data access from the Java programming language. The MySqli database provides the information about the contents that are fed into by the users and the other actors during validation in the website. It gives universal access to all the flat files present in the database. There are many other functionalities present in addition to the information schema present in the database by default. It provides an environment where the user has the capacity to create and delete any number databases.

1.4.3 HTML(Hyper Text Markup Language)

HTML (Hyper Text Markup Language) is a coding language which used to develop the front end of the website. It is create using the predefined tags which are used to develop the divisional contents in the webpage. HTML is a GUI based line of coding where all the line of code the developer consticts, it will be displayed in the webpage.

1.4.4 Bootstrap

Bootstrap is a predefined set of functions which includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins. Bootstrap helps the developers not to create

the website from scratch. Using bootstrap, more creative and responsive websites. Basically Bootstraps are developed using minimalised Javascript command. [2]

1.4.5 CSS

CSS stands for Cascading Style Sheet. CSS is used to design the website. It is a dependant platform because the CSS cannot be run standalone. It requires HTML to explicit its features.

1.4.6 JavaScript

JavaScript is a client side code which is used for validation of forms. This technology of code is to frame the functionalities in the front end for validation purposes. It forms a bridge between the user and the interface because the caution scripts are done using the javascript commands.[3]

2 System Requirements

2.1 Requirements

This section will give the description of the requirements of the user over the website application. There are many functionalities added to the website where the user can check and using the GUI and also using the GUI the user can book the seat according to the website requirement. The website also requests the user for certain credentials which are mandatory for the database contents.

2.1.1 Functional and Security requirements

RANKING	USE CASE	ACTOR
High	View the web contents	User
High	Login \ Authenticate	User \ Admin
High	Database retrieval	Admin
Medium	Modify Database Contents	Admin
Medium	Add Customer	User
Medium	Add Bookings	User
Medium	List all Bookings	Admin
Medium	Compromise Confidentiality	Attacker
Medium	Compromise Integrity	Attacker
Medium	Compromise Availability	Attacker
Medium	Secure Database Credentials	Admin
Medium	Ensure Availability	Web Server

Table 1: Functional and Security Requirements

This section will describe and identify what the application must do, what benefit the application will provide to the user when the user itself starts the application.

- When ever the user enters into the website, the user is initially redirected to the login page of the website. If the user is already present in the database, the user(customer) is then redirected to the booking page where the customer can book a place in the store using the GUI of the website. But if the user is not available, the user login credential is checked and is redirected to signup page where the user customer feeds in the required credentials and then the input validation process is done to ensure that the data fed into the database by the user is valid and then an alert message is shown to the user customer showing that the data is added into the database. Then the user

has the privilege to book and view all the page in the website. After all process is done, the user can log out of the website.

- The website comprises of input fields, dropdown tabs, checkboxes which makes the website to be more user friendly.
- The functional requirements of the website are the following:
    - Add Customer
    - Modify Customer
    - Customer place booking
    - View all the bookings
    - Modify the booking database

- The Security Functional requirements of the website are the following
    - Client side and server side validation
    - Password Hashing algorithm (BCRYPT)
    - Password analyzer
    - Two-factor authentication
    - Ensure non-repudiation
    - Prepared Statements
    - CSRF prevention (Cross Site Request Forgery)
    - Session Management
    - SSL Certification (Secure Socket Layer)
    - Admin Role
    - Security question validation
    - Broken Authentication
    - XXE Prevention (XML eXternal Entity)
    - Database login credentials

2.1.2 User requirements

The user requires a user friendly GUI where the user can scan through the website and explore the contents in it. Some of the basic requirements required for the user are.,

- Personal computer
- Undisruptive internet connection
- Web browser
- Firewall activated
- Account credentials

2.2 Environmental requirements

The environmental requirements for the developer in this website development is a webserver, continuous internet connection, an activated firewall in the browser, XAMPP server manager for managing the server and the database and the styling must be done to make sure that the

user easily accesses the website. Some of the pictorial references are given for easier references.

USE CASE

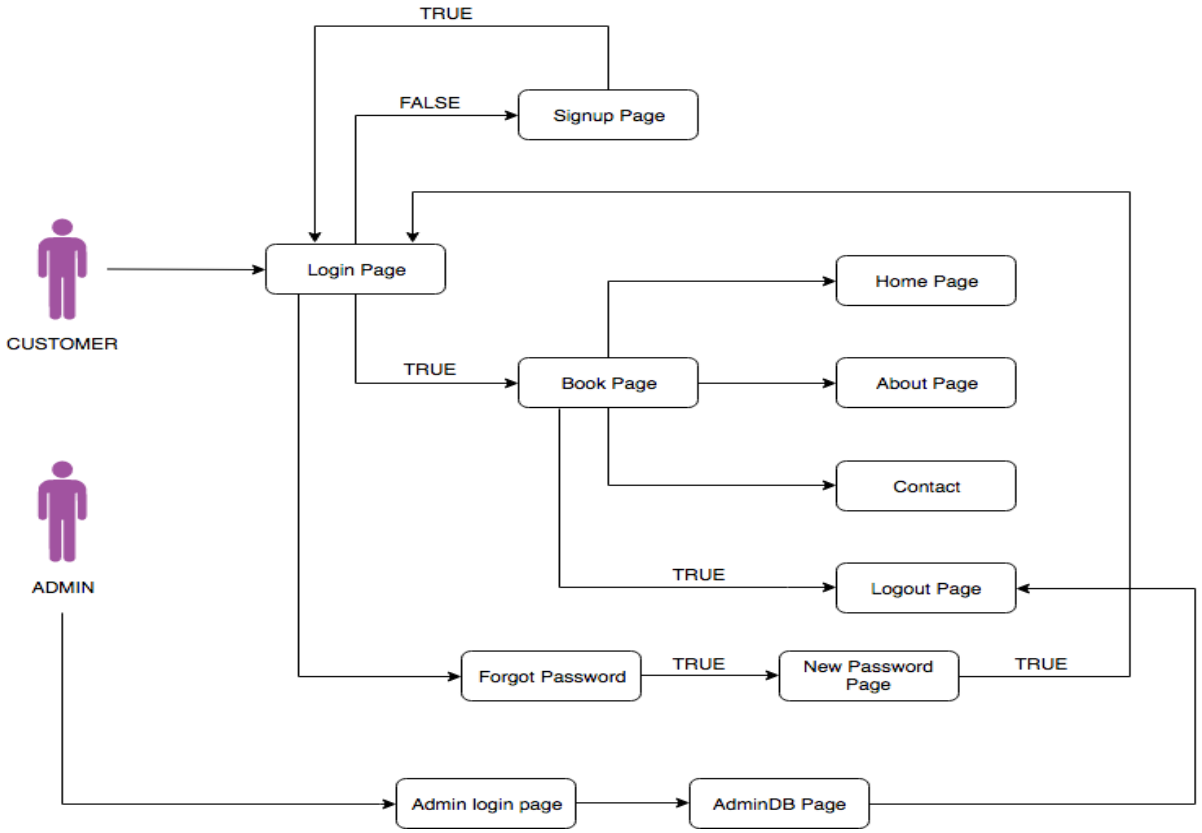

Fig 1 – Use Case diagram

SEQUENCE DIAGRAM

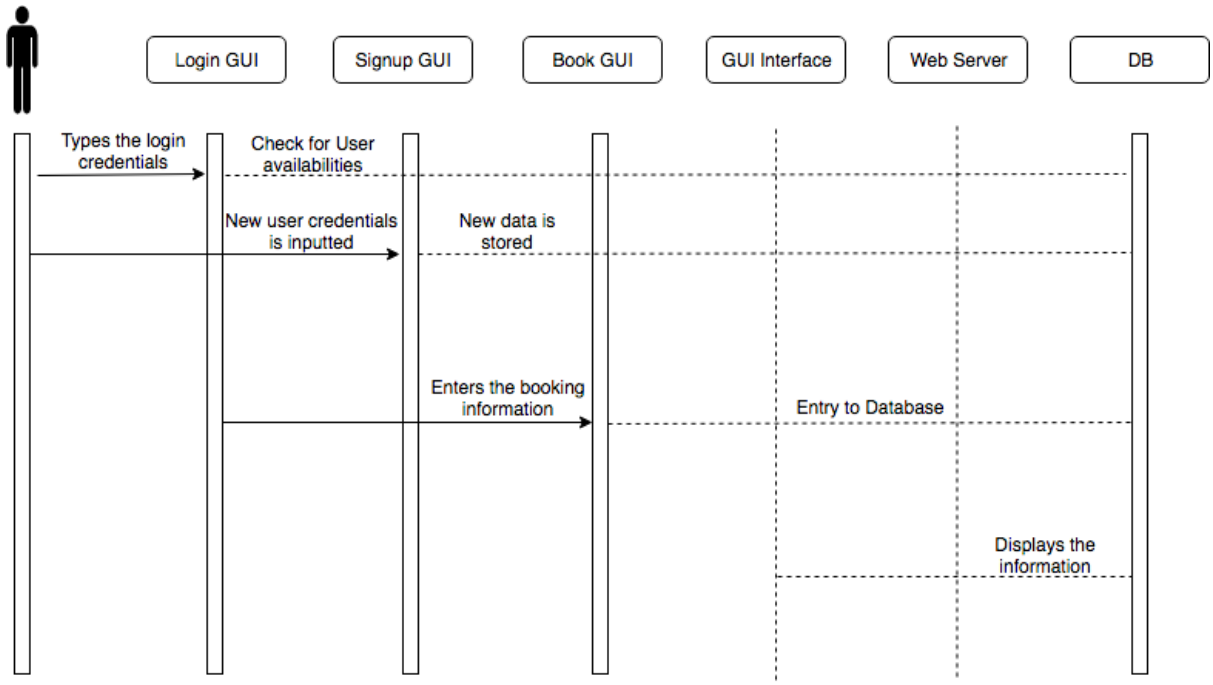

Fig 2 – Sequence diagram

CLASS DIAGRAM

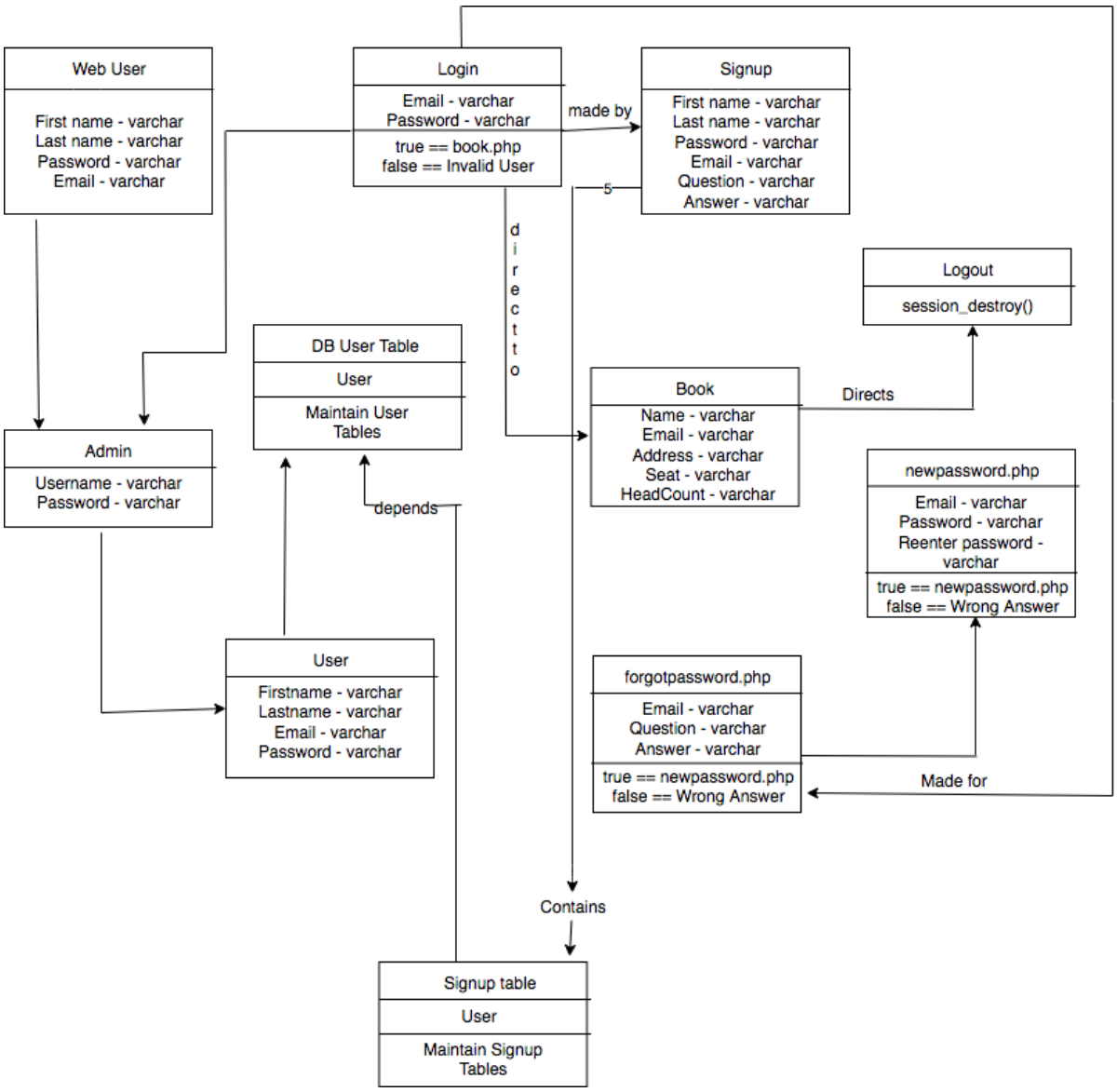

Fig 3 – Class diagram

INFRASTRUCTURE DIAGRAM

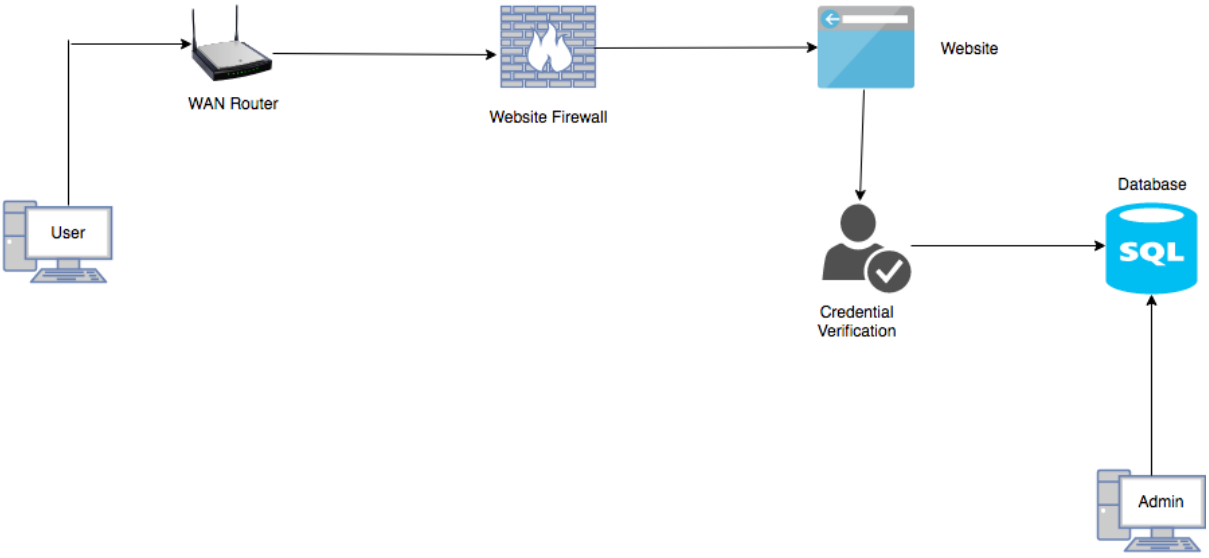

Fig 3 – Infrastructure diagram

2.3 Implementation

This website contains 13 PHP pages, one CSS page and Javascripts inscribed in the PHP files.

2.3.1 Index.php

This PHP page is the first and the foremost page the customer user is redirected to. This page contains the login form where the user types in the credentials requested by the website. If the condition provided by the user is true, the user will be redirected to book.php or the user is given alert for the failure of login.

```
    <form action="indexvalidate.php" method="post" class="register-form" onsubmit="return validation()">
      <div class="form">
      <input type="email" name="email" placeholder="Type in the email id" id="mail" autocomplete="off" required />
      <span id="emailid"></span>
      </div>
      <div class="form">
      <input type="password" placeholder="Type in the password" name="pass" id="pass" autocomplete="off" required />
      <span id="password"></span>
      </div>
      <div class="rem">
      <input type="checkbox" name="remember"> Remember Me </input>
      </div>

      <!-- <div class="form">
      <input type="email" name="emailid" placeholder="Type again the email id" id="mail" autocomplete="off" />
      <span id="emailid"></span>
      </div> -->
  <center><p id="question"></p><div class="form"><input id="ans" type="text" autocomplete="off"></div>
   <div id="message">Type correct answer to verify.</div>
   <div id="success"></div>
   <div id="fail"></div>
   <div class="form">
   <input type="submit" value='Login' name='operation'>
   <center><a href = "forgotpass.php">Forgot password</a></center>
 </div></center>


    </form>

  </div>
  </div>

  <footer>
      <div class="container">
          <div class="row">
              <div class="col-lg-12 text-center">
                  <p>Copyright &copy; Cafe Agape 2018</p>
              </div>
          </div>
      </div>
  </footer>
```

Fig 4 – index.php

2.3.2 Signup.php

This pages helps the new customer user to login into the website. Every time the user gives the credentials in the signup page, the data are stored in the database.

```
  <form action="signup.php" method="post" class="register-form" onsubmit="return validate()">
<div class="form">
    <input type="text" placeholder="First Name" name="firstname" id="fname" autocomplete="off" required />
    <span id="firstname"></span>
</div>
<div class="form">
    <input type="text" placeholder="Last Name" name="lastname" id="lname" autocomplete="off" required />
    <span id="lastname"></span>
</div>
<div class="form">
    <input type="password" id="pass" placeholder="Enter password" name="password"  autocomplete="off" required onkeyup="checkPassword(this.v
        alue)" />
    <span id="password"></span>
    <progress value="0" max="100" id="strength" style="width: 230px"></progress>
</div>
<div class="form">
    <input type="password" placeholder="Re-enter password" name="repassword" id="repass" autocomplete="off" required />
    <span id="repassword"></span>
</div>
<div class="form">
    <input type="email" name="emailid" placeholder="Type in the email id" id="mail" autocomplete="off" required />
    <span id="emailid"></span>
</div>
<div class="dropdown">
        <label>Security Question</label>
    <select type="question" name="question" class="question">
    <option value="What is your birth city?">What is your birth city?</option>
    <option value="What is your pet name?">What is your pet's name?</option>
    <option value="What is your Mother name?">What is your Mother's name?</option>
    </select>
</div>
<div class="form">
    <input type="text" name="answer" placeholder="Type in the answer" autocomplete="off" id="ans" required />
    <span id="answer"></span>
</div>
<center><div class="g-recaptcha" data-sitekey="6LcFZ3sUAAAAAKn5e_K97Vgpi8H8D9J6rtgKd4_z"></div></center>

<div class="form">
    <input type="submit" name="submit" value="submit" class="btn btn-signup" autocomplete="off" />
</div>
<input type="hidden" name="_token" class="form-control" value="<?php echo $_session['_token'];?>"/>
  </form>
   <script src='https://www.google.com/recaptcha/api.js'></script>
</div>
</div>
```

Fig 5 – signup.php

2.3.3 Forgotpass.php

If the user forgets the password, the user gets validated using the email id and the security question and answer which is then validated referring the database.

```
  <form action="forgotpass.php" method="post" class="register-form">
    <div class="form">
    <input type="email" name="email" placeholder="Type in the email id" id="mail" autocomplete="off" required />
    <span id="email"></span>
    </div>
     <center><div class="dropdown">
        <label>Security Question</label>
    <select type="question" name="question" class="question">
    <option value="What is your birth city?">What is your birth city?</option>
    <option value="What is your pet name?">What is your pet's name?</option>
    <option value="What is your Mother name?">What is your Mother's name?</option>
    </select>
</div></center>
<div class="form">
    <input type="text" name="answer" placeholder="Type in the answer" autocomplete="off" required />
    <span id="answer"></span>
</div>
    <div class="form">
        <input type="hidden" name="_token" class="form-control" value="<?php echo $_session['_token'];?>"/>
    <input type="submit" name="submit" value="submit" class="btn btn-signup" autocomplete="off" />
    </div>
  </form>
</div>
</div>
```

Fig 6 – forgotpass.php

2.3.4 Reenter.php

If the answer provided by the customer user, the page is redirected to reenter.php where the user enters the new password and it is stored in the database.

```
  <form action="reenter.php" method="post" class="register-form">
    <div class="form">
    <input type="email" name="email" placeholder="Enter E-mail ID" id="mail" autocomplete="off" required />
    <span id="email"></span>
    </div>
    <div class="form">
    <input type="password" name="password" placeholder="New Password" id="mail" autocomplete="off" required />
    <span id="email"></span>
    </div>
<div class="form">
    <input type="password" name="repassword" placeholder="Re-enter New Password" autocomplete="off" required />
    <span id="answer"></span>
</div>
    <div class="form">
        <input type="hidden" name="_token" class="form-control" value="<?php echo $_session['_token'];?>"/>
    <input type="submit" name="submit" value="submit" class="btn btn-signup" autocomplete="off" />
    </div>
  </form>
</div>
</div>
```

Fig 7 – reenter.php

2.3.5 book.php

After the user is validated, the customer user can book the place in the physical store. The data is then stored in the database.

```
  <form action="book.php" method="post" class="register-form">
    <input type="text" name="name" placeholder="Name" required />
    <input type="email" name="email" id="email" placeholder="Email" required />
    <input type="text" name="address" placeholder="Address" required />
    <div class="dropdown">
        <label>Select Seat</label>
    <select type="seat" name="seat" class="seat">
    <option value="Executive 1">Executive 1</option>
    <option value="Executive 2">Executive 2</option>
    <option value="Executive 3">Executive 3</option>
    <option value="Premium Cabin 1">Premium Cabin 1</option>
    <option value="Premium Cabin 2">Premium Cabin 2</option>
    <option value="cabin 1">Cabin 1</option>
    <option value="cabin 2">Cabin 2</option>
    <option value="cabin 3">Cabin 3</option>
    <option value="normal 1">Normal 1</option>
    <option value="normal 2">Normal 2</option>
    <option value="normal 3">Normal 3</option>
    <option value="normal 4">Normal 4</option>
    <option value="normal 5">Normal 5</option>
    </select>
</div>


    <div class="dropdown">
        <label>Head Count</label>
    <select type="seat" name="headcount" class="seat">
    <option value="One">One</option>
    <option value="Two">Two</option>
    <option value="Three">Three</option>
    <option value="Four">Four</option>
    <option value="Five">Five</option>
    <option value="Six">Six</option>
    </select>
</div>

    <input type="hidden" name="_token" class="form-control" value="<?php echo $_session['_token'];?>"/>
    <input type="submit" value="book" name="book" class="btn btn-signup"/>
  </form>
</div>
</div>
```

Fig 8 – book.php

2.3.6 admin.php

This page has the functionality of the admin to login and modify the contents of the database which the user has provided.

```
<form action="admin.php" method="post" class="register-form">
  <div class="form">
  <input type="text" name="name" placeholder="Type in the Admin Name" id="mail" autocomplete="off" required />
  <span id="email"></span>
  </div>
  <div class="form">
  <input type="password" placeholder="Type in the password" name="pass" id="pass" autocomplete="off" required />
  <span id="password"></span>
  </div>

  <div class="form">
  <input type="submit" name="submit" value="login" class="btn btn-signup" autocomplete="off" />
  </div>
</form>
</div>
```

Fig 9 – admin.php

2.3.7 admindb.php

Here in this page, the admin has the role to view all the bookings done by the customer user and can modify the data which will be reflected in the database.

```
    <table class = "table">
        <tr>
        <th> ID</th>
        <th> Name </th>
        <th> Email </th>
        <th> Address </th>
        <th> Seat </th>
        <th> HeadCount </th>
        <th> Action</th>
    </tr>
    <?php
        $i = 1;

        while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
            $name = $row['Name'];
            $email = $row['Email'];
            $add = $row['Address'];
            $seat = $row['Seat'];
            $headcount = $row['HeadCount'];
            $id = $row['id'];
        ?>
            <tr>
                <td><?php echo $i; ?></td>
                <td><?php echo $name; ?></td>
                <td><?php echo $email; ?></td>
                <td><?php echo $add; ?></td>
                <td><?php echo $seat; ?></td>
                <td><?php echo $headcount; ?></td>
                <td>
                    <a href="admindb.php?delete=<?php echo $id; ?>" onclick="return confirm('Are you sure?');">Booking Taken</a>
                </td>
            </tr>

            <?php
                $i++;
            }
            if(isset($_GET['delete'])) {
                $delete_id = $_GET['delete'];

                mysqli_query($dbconnect, "DELETE FROM book WHERE id = '$delete_id'");
            }
            ?>
</table>
```

Fig 10 – admindb.php

2.3.8 home.php

In this page, it contains details of the physical store cafeagape.

```
    <div class="row">
        <div class="box">
            <div class="col-lg-12">
                <hr>
                <h2 class="intro-text text-center">Know us
                    <strong>worth sipping a cup of cafe</strong>
                </h2>
                <hr>
                <img class="img-responsive img-border img-left" src="img/intro-pic.jpg" alt="">
                <hr class="visible-xs">
                <p>At the heart of our culture is a keen appetite for contributing back into the growth of the community, and with roots in
                ministry, we're dedicated to supporting those in need. Good food and good people will build a stronger, safer, and kinder
                community. Join us in backing the multitude of impactful charitable organizations in our area, including Coalition for the
                Homeless, Arnold Palmer Medical Center, Foundation for Foster Children, Elevate Orlando, and the Florida Hospital and
                Foundation.</p>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="box">
            <div class="col-lg-12">
                <hr>
                <h2 class="intro-text text-center">
                    <strong>to showcase our content</strong>
                </h2>
                <hr>
                <p>We serve Expresso, Americano, Cappucino, Latte, Flat White, Mocha, Hot Chocolate, Speciality Drinks, </p>
                <p>Soya Milk, Coconut Milk, Almond Milk, Flavour Cafe, Tea, Herbal Tea</p>
            </div>
        </div>
    </div>
```

Fig 11 – home.php

2.3.9 about.php

this page contains the about details of the physical store cafeagape.

```
<div class="container">

    <div class="row">
        <div class="box">
            <div class="col-lg-12">
                <hr>
                <h2 class="intro-text text-center">About
                    <strong>Cafe Agape</strong>
                </h2>
                <hr>
            </div>
            <div class="col-md-6">
                <img class="img-responsive img-border-left" src="img/slide-2.jpg" alt="">
            </div>
            <div class="col-md-6">
                <p>Tradition and great flavour are the heart and soul of Irish cuisine. Since 2015 Cafe Agape has believed in sharing those
                traditions. We invite the people of Dublin to take part in sharing- with us and with each other-the Cafe Agape way of dining.
                If you follow along with our menu, you will experience tasty breakfast and brunch creations, enjoy our lunch menus which are
                inspired to tantalise the taste buds. With the help of your server, enjoy the most important ingredients in your dining
                experience: food, friends, and family.</p>
            </div>
            <div class="clearfix"></div>
        </div>
    </div>
```

Fig 12 – admin.php

2.3.10 contact.php

This page contains the contact details of the physical store cafeagape.

```
<div class="container">

    <div class="row">
        <div class="box">
            <div class="col-lg-12">
                <hr>
                <h2 class="intro-text text-center">Contact
                    <strong>Cafe Agape</strong>
                </h2>
                <hr>
            </div>
            <div class="col-md-8">
                <!-- Embedded Google Map using an iframe - to select your location find it on Google maps and paste the link as the iframe
                src. If you want to use the Google Maps API instead then have at it! -->
                <iframe width="100%" height="400" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="http://maps.google.com/
                maps?hl=en&amp;ie=UTF8&amp;ll=37.0625,-95.677068&amp;spn=56.506174,79.013672&amp;t=m&amp;z=4&amp;output=embed"></iframe>
            </div>
            <div class="col-md-4">
                <p>Phone:
                    <strong>123.456.7890</strong>
                </p>
                <p>Email:
                    <strong><a href="mailto:name@example.com">abe@cafeagape.com</a></strong>
                </p>
                <p>Address:
                    <strong>5 Belvedere Place
                        <br>Mount Joy, Dublin 1</strong>
                </p>
            </div>
            <div class="clearfix"></div>
        </div>
    </div>
```

Fig 13 – contact.php

2.3.11 CSS functionalities

The following screenshots provide the styling command used to style the website page.

```css
body {
    font-family: "Open Sans","Helvetica Neue",Helvetica,Arial,sans-serif;
    background: url('../img/bg.jpg') no-repeat center center fixed;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    background-size: cover;
    -o-background-size: cover;
}

progress {
    color: blue;
}

h1,
h2,
h3,
h4,
h5,
h6 {
    text-transform: uppercase;
    font-family: "Josefin Slab","Helvetica Neue",Helvetica,Arial,sans-serif;
    font-weight: 700;
    letter-spacing: 1px;
}

p {
    font-size: 1.25em;
    line-height: 1.6;
    color: #000;
}

hr {
    max-width: 400px;
    border-color: #999999;
}

.brand,
.address-bar {
    display: none;
}

.navbar-brand {
    text-transform: uppercase;
    font-weight: 900;
    letter-spacing: 2px;
}

.navbar-nav {
    text-transform: uppercase;
    font-weight: 400;
    letter-spacing: 3px;
}

.img-full {
    min-width: 100%;
}
```

Fig 14 – css1

```css
.brand-before,
.brand-name {
    text-transform: capitalize;
}

.brand-before {
    margin: 15px 0;
}

.brand-name {
    margin: 0;
    font-size: 4em;
}

.tagline-divider {
    margin: 15px auto 3px;
    max-width: 250px;
    border-color: #999999;
}

.box {
    margin-bottom: 20px;
    padding: 30px 15px;
    background: #fff;
    background: rgba(255,255,255,0.9);
}

.intro-text {
    text-transform: uppercase;
    font-size: 1.25em;
    font-weight: 400;
    letter-spacing: 1px;
}

.img-border {
    float: none;
    margin: 0 auto 0;
    border: #999999 solid 1px;
}

.img-left {
    float: none;
    margin: 0 auto 0;
}

footer {
    background: #fff;
    background: rgba(255,255,255,0.9);
}

footer p {
    margin: 0;
    padding: 50px 0;
}
```

Fig 15 – css2

```css
@media screen and (min-width:768px) {
    .brand {
        display: inherit;
        margin: 0;
        padding: 30px 0 10px;
        text-align: center;
        text-shadow: 1px 1px 2px rgba(0,0,0,0.5);
        font-family: "Josefin Slab","Helvetica Neue",Helvetica,Arial,sans-serif;
        font-size: 5em;
        font-weight: 700;
        line-height: normal;
        color: #fff;
    }

    .top-divider {
        margin-top: 0;
    }

    .img-left {
        float: left;
        margin-right: 25px;
    }

    .address-bar {
        display: inherit;
        margin: 0;
        padding: 0 0 40px;
        text-align: center;
        text-shadow: 1px 1px 2px rgba(0,0,0,0.5);
        text-transform: uppercase;
        font-size: 1.25em;
        font-weight: 400;
        letter-spacing: 3px;
        color: #fff;
    }

    .navbar {
        border-radius: 0;
    }

    .navbar-header {
        display: none;
    }

    .navbar {
        min-height: 0;
    }

    .navbar-default {
        border: none;
        background: #fff;
        background: rgba(255,255,255,0.9);
    }
```

Fig 15 – css3

```css
    .nav>li>a {
        padding: 25px;
    }

    .navbar-nav>li>a {
        line-height: normal;
    }

    .navbar-nav {
        display: table;
        float: none;
        margin: 0 auto;
        table-layout: fixed;
        font-size: 1.25em;
    }
}

.form{
    background: transparent;
    background-size: cover;
    position: relative;
    z-index: 1;
    max-width: 360px;
    margin: 0 auto 0;
    padding: 20px;
    text-align: center;
    align-items: center;

}

.form input{
    font-family: 'Flamenco', cursive;
    outline: 1;
    background: #E6E3E1;
    width: 100%;
    border: 0;
    margin: 0 auto 15px;
    padding: 10px;
    box-sizing: border-box;
    font-size: 14px;
}

.rem{
    font-family: 'Flamenco', cursive;
    outline: 1;
    background: transparent;
    text-align: center;
    width: 100%;
    border: transparent;
    /*margin: 0 auto 15px;*/
    /*padding: 10px;*/
    box-sizing: border-box;
    font-size: 14px;
}
```

Fig 16 – css4

```css
.dropdown{
    font-family: 'Flamenco', cursive;
    outline: 1;
    background: #E6E3E1;
    width: 100%;
    border: 0;
    margin: 0 0 auto;
    padding: 15px;
    box-sizing: border-box;
    font-size: 14px;
    text-align: left;
}

.seat{
    font-family: 'Flamenco', cursive;
    outline: 1;
    background: #E6E3E1;
    width: auto;
    border: 0;
    margin: 0 auto 15px;
    padding: 10px;
    box-sizing: border-box;
    font-size: 14px;
}


.btn{
    display: inline-block;
    padding: 10px 30px;
    font-weight: lighter;
    text-decoration: none;
    text-transform: uppercase;
    border-radius: 200px;
    transition: background-color 0.2s, border 0.2s, color 0.2;
}

.btn-login{
    background-color: transparent;
    color: #875338;
    margin-right: 15px;
    border: 1px solid #875338;
}

.btn-login:hover{
    background-color: #fff;
}

.btn-signup{
    background-color: transparent;
    color: #000;
    border: 1px solid #000;
}
```

Fig 17 – css5

```css
.btn-signup:hover{
    background-color: #875338;
}

table, th, td {
    border-top: 1px solid black;
    border-right: 1px solid black;
    border: 1px solid black;
    border-collapse: collapse;
    width: 100%
    color: #875338;
    font-family: 'Flamenco', cursive;
    table-layout: auto;
    text-align: left;
}



@media screen and (min-width:1200px) {
    .box:after {
        content: '';
        display: table;
        clear: both;
    }
}
```

Fig 18 – css6

3 SECURITY IMPLEMENTATION

There are numerous security features involved in this website to ensure the safety of the user credentials for not getting tampered from any external or third party attacks. The security features implemented in this website are listed below.,

- Client and Server side authentication
- Password Hashing algorithm (BCRYPT)
- Password analyzer
- Two-factor authentication
- Ensure non-repudiation
- Prepared Statements
- CSRF prevention (Cross Site Request Forgery)
- Session Management
- SSL Certification (Secure Socket Layer)
- Admin Role
- Security question validation
- Broken Authentication
- XXE Prevention (XML eXternal Entity)
- Database Login Credentials

3.1 CLIENT AND SERVER SIDE AUTHENTICATION

3.1.1 SERVER SIDE AUTHENTICATION

The server side authentication is done to all the pages where the user feeds in the credentials and in places where user data are displayed out.

3.1.1.1 Login validation

In this page, the user feeds in the credentials for logging into the website. The server checks for the correctness of the user input from the database and accordingly allows and rejects the user.

```php
<?php
include("csrf.php");
if (isset($_POST['operation'])) {

    $username = "sriram";
    $password = "rSDO0Ir40Iwz8CVX";
    $hostname = "localhost";

    $dbhandle = mysqli_connect($hostname, $username, $password) or die("Could not connect to DB");
    $selected = mysqli_select_db($dbhandle, "cafeagape");
        // session_start();

        $email = $_POST['email'];
        $pass = $_POST['pass'];


        $sql = "SELECT password from signup where email='".$email."'";


        $run = mysqli_query($dbhandle, $sql);


        $row = $run->fetch_assoc();

        if($email == $sql) {
            if (isset($_POST['remember'])) {
            setcookie('email', $email, time()+60*60*7);
            setcookie('pass', $pass, time()+60*60*7);
        }
        session_start();
        $_SESSION['email'] = $email;
        header("location: book.php");
    }


        if (password_verify($pass,$row["password"])) {

            $_SESSION['email'] = $email;
            header("location: book.php");
        }
        else{

        header("location: index.php?error=1");
            }
      }else {
        header("location: index.php");
      }
?>
```

Fig 19 – login validation

3.1.1.2 Signup validation

The server here checks the user input and validates whether the data is already present in the database and also checks for the user input from server end

```php
<?php
include('csrf.php');

$mysqli = new mysqli('localhost','sriram','rSDO0Ir40Iwz8CVX','cafeagape');


  if (isset($_POST['submit'])) {
      // session_start();
      $sql = "INSERT INTO signup(firstname, lastname, password, email, Question, Answer) VALUES (?,?,?,?,?,?)";
      $stmt = $mysqli->prepare($sql);
      $stmt->bind_param("ssssss", $firstname, $lastname, $password, $email, $question, $answer);

       $firstname = mysqli_real_escape_string($mysqli, $_POST['firstname']);
      $lastname = mysqli_real_escape_string($mysqli, $_POST['lastname']);
      $password = mysqli_real_escape_string($mysqli, $_POST['password']);
      $password2 = mysqli_real_escape_string($mysqli, $_POST['repassword']);
      $email = mysqli_real_escape_string($mysqli, $_POST['emailid']);
      $question = mysqli_real_escape_string($mysqli, $_POST['question']);
      $answer = mysqli_real_escape_string($mysqli, $_POST['answer']);


      if ($password == $password2) {

          $password = password_hash($password, PASSWORD_BCRYPT);

      $stmt->execute();
       header("location: index.php");

  } else {
      echo "<script>alert('Passwords doesn't match');</script>";
  }
  $stmt->close();
  $mysqli->close();

}
?>
```

Fig 20 – signup validation

3.1.1.3 Forget password validation

Here the server validates whether the email is present in the database and then it validates question and answer input by the user.

```php
<?php
include("csrf.php");

    $dbconnect = mysqli_connect('localhost', 'root', '', 'cafeagape');

if (isset($_POST['submit'])) {


        // session_start();

        $email = mysqli_real_escape_string($dbconnect, $_POST['email']);
        $question = mysqli_real_escape_string($dbconnect, $_POST['question']);
        $answer = mysqli_real_escape_string($dbconnect, $_POST['answer']);

        $sql = "SELECT Question, Answer FROM signup WHERE email='".$email."'";


        $run = mysqli_query($dbconnect, $sql);


        $row = $run->fetch_assoc();

        if($question == $row["Question"] && $answer == $row['Answer']) {
        header("location: reenter.php");
    } else {
        echo "<script>alert('Wrong Answer!!');</script>";
    }
}
?>
```

Fig 21 – forgot password validation

3.1.1.4 New password validation

The page here checks for the matching of the password and the verification of the email present in the database. And then the new password is updated in the database.

```php
<?php
include("csrf.php");

$dbconnect = mysqli_connect('localhost', 'root', '', 'cafeagape');

if(isset($_POST['submit'])) {

    $email = mysqli_real_escape_string($dbconnect, $_POST['email']);
    $pass = mysqli_real_escape_string($dbconnect, $_POST['password']);
    $repass = mysqli_real_escape_string($dbconnect, $_POST['repassword']);

    $sql = "SELECT email FROM signup WHERE email='".$email."'";



        $run = mysqli_query($dbconnect, $sql);


        $row = $run->fetch_assoc();
        // echo "$email";
if(mysqli_num_rows($email) == 0){
        // echo "$email";
    if($pass == $repass) {
        $pass = password_hash($pass, PASSWORD_BCRYPT);
        $sql1 = "UPDATE signup SET password='$pass' WHERE email='$email'";

        $run1 = mysqli_query($dbconnect, $sql1);

        // echo "hello";
        echo "<script>alert('Password Changed')";
        header("location: index.php");
    } else {
        echo "<script>alert('Password update unsuccessful!');</script>";
    }
    }
    else {
        echo "<script>alert('User doesnot exist!');</script>";
    }
}

?>
```

Fig 22 – new password validation

3.1.1.5 Booking validation

Here the server checks for the correctness of the input values provided by the customer user. And post validation, the data is stored in the database.

```php
<?php
include("csrf.php");
    session_start();
    $mysqli = new mysqli('localhost','sriram','rSDO0Ir40Iwz8CVX', 'cafeagape');
    if(isset($_POST['book'])) {
    $sql = "INSERT INTO book(Name, Email, Address, Seat, HeadCount) VALUES (?,?,?,?,?)";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param("sssss", $name, $email, $add, $seat, $head);

    $name = mysqli_real_escape_string($mysqli, $_POST['name']);
    $email = mysqli_real_escape_string($mysqli, $_POST['email']);
    $add = mysqli_real_escape_string($mysqli, $_POST['address']);
    $seat = mysqli_real_escape_string($mysqli, $_POST['seat']);
    $head = mysqli_real_escape_string($mysqli, $_POST['headcount']);
    // $types = mysqli_real_escape_string($mysqli, $_POST['types']);
      $stmt->execute();
      echo "<script>alert('Booking Successful...')</script>";

  } else {
  }
?>
```

Fig 22 – Booking page validation

3.1.2 CLIENT SIDE AUTHENTICATION

Even though the server side is validated, the client needs some measures to verify and so javascript has been inscribed in the PHP file for validation.

3.1.2.1 Login validation

```javascript
function validation(){

    var email = document.getElementById('mail').value;
    var pass = document.getElementById('pass').value;

    if(email == ""){
      document.getElementById('emailid').innerHTML = "**Please fill the email_id field font-weight-bold".fontcolor("red");
      return false;

    }

    else if(email.indexOf('@') <= 0){

    document.getElementById('emailid').innerHTML = "** @ Position invalid".fontcolor("red").fontcolor("red");
    return false;


  }

  else if((email.charAt(email.length-4)!='.') && (email.charAt(email.length-3)!='.')){

    document.getElementById('emailid').innerHTML = "** . Position invalid".fontcolor("red");
    return false;

  }




  else if(pass == ""){
      document.getElementById('password').innerHTML = "**Please fill the password field".fontcolor("red");
      return false;
  }


  else if((pass.length <=8) || (pass.length > 20)){

    document.getElementById('password').innerHTML = "**Password must be between 8 and 20".fontcolor("red");
    return false;
  }
}

  </script>
```

Fig 23 – Login page validation

3.1.2.2 Signup validation

```
function validate(){

  var first = document.getElementById('fname').value;
  var last = document.getElementById('lname').value;
  var email = document.getElementById('mail').value;
  var pass = document.getElementById('pass').value;
  var repass = document.getElementById('repass').value;
  var answer = document.getElementById('ans').value;

  if(first == ""){

    document.getElementById('firstname').innerHTML = "**Please fill the first name ".fontcolor("red");
    return false;
  }

  else if(!isNaN(first)){

    document.getElementById('firstname').innerHTML = "**Only characters are allowed".fontcolor("red");
    return false;

  }

  else if((first.length <=3) || (first.length > 20)){

    document.getElementById('firstname').innerHTML = "**Firstname's length must be between 3 and 20".fontcolor("red");
    return false;
  }


  else if(last==""){

    document.getElementById('lastname').innerHTML = "**Please fill the last name".fontcolor("red");
    return false;
  }


  else if(!isNaN(last)){

    document.getElementById('lastname').innerHTML = "**Only characters are allowed".fontcolor("red");
    return false;

  }

  else if((last.length <=2) || (last.length > 20)){

    document.getElementById('lastname').innerHTML = "**Lastname's length must be between 2 and 20".fontcolor("red");
    return false;
  }


  else if(email==""){

    document.getElementById('emailid').innerHTML = "**Please fill the email field".fontcolor("red");
    return false;
  }
```

Fig 24 – Signup page validation 1

```
    else if(email.indexOf('@') <= 0){

      document.getElementById('emailid').innerHTML = "** @ Position invalid".fontcolor("red");
      return false;


    }

    else if((email.charAt(email.length-4)!='.') && (email.charAt(email.length-3)!='.')){

      document.getElementById('emailid').innerHTML = "** . Position invalid".fontcolor("red");
      return false;

    }

    else if(pass==""){

      document.getElementById('password').innerHTML = "**Please fill the password field".fontcolor("red");
      return false;
    }

    else if((pass.length <=8) || (pass.length > 20)){

      document.getElementById('password').innerHTML = "**Password must be between 8 and 20".fontcolor("red");
      return false;
    }

    else if(pass!=repass){


      document.getElementById('repassword').innerHTML = "**Passwords don't match".fontcolor("red");
      return false;

    }

    else if(repass==""){

      document.getElementById('repassword').innerHTML = "**Please enter the password confirmation".fontcolor("red");
      return false;
    }

    else if(answer==""){
      document.getElementById('answer').innerHTML = "**Please enter the answer".fontcolor("red");
      return false;
    }
  }
</script>
```

Fig 25 – Signup page validation 2

3.1.2.3 Forgot password validation

```
<?php
include("csrf.php");

    $dbconnect = mysqli_connect('localhost', 'root', '', 'cafeagape');

if (isset($_POST['submit'])) {


        // session_start();

        $email = mysqli_real_escape_string($dbconnect, $_POST['email']);
        $question = mysqli_real_escape_string($dbconnect, $_POST['question']);
        $answer = mysqli_real_escape_string($dbconnect, $_POST['answer']);

        $sql = "SELECT Question, Answer FROM signup WHERE email='".$email."'";


        $run = mysqli_query($dbconnect, $sql);


        $row = $run->fetch_assoc();

        if($question == $row["Question"] && $answer == $row['Answer']) {
        header("location: reenter.php");
    } else {
        echo "<script>alert('Wrong Answer!!');</script>";
    }
}
?>
```

Fig 26 – Forgot password validation

3.1.2.4 New password validation

```php
<?php
include("csrf.php");

$dbconnect = mysqli_connect('localhost', 'root', '', 'cafeagape');

if(isset($_POST['submit'])) {

    $email = mysqli_real_escape_string($dbconnect, $_POST['email']);
    $pass = mysqli_real_escape_string($dbconnect, $_POST['password']);
    $repass = mysqli_real_escape_string($dbconnect, $_POST['repassword']);

    $sql = "SELECT email FROM signup WHERE email='".$email."'";



        $run = mysqli_query($dbconnect, $sql);


        $row = $run->fetch_assoc();
        // echo "$email";
if(mysqli_num_rows($email) == 0){
        // echo "$email";
    if($pass == $repass) {
        $pass = password_hash($pass, PASSWORD_BCRYPT);
        $sql1 = "UPDATE signup SET password='$pass' WHERE email='$email'";

        $run1 = mysqli_query($dbconnect, $sql1);

        // echo "hello";
        echo "<script>alert('Password Changed')";
        header("location: index.php");
    } else {
        echo "<script>alert('Password update unsuccessful!');</script>";
    }
    }
    else {
        echo "<script>alert('User doesnot exist!');</script>";
    }
}

?>
```

Fig 27 – New password validation

3.1.2.5 Book page validation

```php
<?php
include("csrf.php");
    session_start();
    $mysqli = new mysqli('localhost','sriram','rSDO0Ir40Iwz8CVX', 'cafeagape');
    if(isset($_POST['book'])) {
    $sql = "INSERT INTO book(Name, Email, Address, Seat, HeadCount) VALUES (?,?,?,?,?)";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param("sssss", $name, $email, $add, $seat, $head);

    $name = mysqli_real_escape_string($mysqli, $_POST['name']);
    $email = mysqli_real_escape_string($mysqli, $_POST['email']);
    $add = mysqli_real_escape_string($mysqli, $_POST['address']);
    $seat = mysqli_real_escape_string($mysqli, $_POST['seat']);
    $head = mysqli_real_escape_string($mysqli, $_POST['headcount']);
    // $types = mysqli_real_escape_string($mysqli, $_POST['types']);
      $stmt->execute();
      echo "<script>alert('Booking Successful...')</script>";

  } else {
  }
?>
```

Fig 28 – Book page validation

3.2 PASSWORD HASHING (BCRYPT)

In this website the user password is hashed into BCRYPT algorithm which is rated to be the second most secure way of hashing and salting the password.

```
if ($password == $password2) {

    $password = password_hash($password, PASSWORD_BCRYPT);

$stmt->execute();
 header("location: index.php");
```

Fig 29 – Password hashing

3.3 PASSWORD ANALYZER

This feature in the website analyses the strength of the password the customer user input. This functionality combines with the javascript validation and shows the strength using the indication.

```
<script type="text/javascript">
var pass = document.getElementById("pass")
pass.addEventListening('keyup', function() {
    checkPassword(pass.value)
})
function checkPassword(password) {
    var strengthBar = document.getElementById("strength")
    var strength = 0;
    if (password.match(/[a-zA-Z0-9][a-zA-Z0-9]+/)) {
        strength += 1
    }
    if (password.match(/[~<>?]+/)) {
        strength += 1
    }
    if (password.match(/[!@#$%^&*()]+/)) {
        strength += 1
    }
    if (password.length > 5) {
        strength += 1
    }
    if (password.length == 0) {
        strength == 0;
    }
    switch(strength) {
        case 0:
        strengthBar.value = 0;
        break
        case 1:
        strengthBar.value = 20;
        break
        case 2:
        strengthBar.value = 40;
        break
        case 3:
        strengthBar.value = 60;
        break
    }
}
```

Fig 30 – Password analyser

3.4 TWO-FACTOR AUTHENTICATION

This security is done to make sure that the website is used by human being. There is a mathematical function which supplies algorithmic functions like addition, subtraction questions to the user only when the answer is correct, the login button is enabled orelse the login button is disabled.

```
    <script type="text/javascript">
        $(document).ready(function(){

$('input[type=submit]').attr('disabled','disabled');

var randomNum1;
var randomNum2;

//set the largeest number to display

var maxNum = 20;
var total;

randomNum1 = Math.ceil(Math.random()*maxNum);
randomNum2 = Math.ceil(Math.random()*maxNum);
total =randomNum1 + randomNum2;

$( "#question" ).prepend( randomNum1 + " + " + randomNum2 + "=" );

// When users input the value

$( "#ans" ).keyup(function() {

    var input = $(this).val();
    var slideSpeed = 200;

    $('#message').hide();

    if (input == total) {

        $('input[type=submit]').removeAttr('disabled');
        $('#success').slideDown(slideSpeed);
        $('#fail').slideUp(slideSpeed);

    }

    else {

        $('input[type=submit]').attr('disabled','disabled');
        $('#fail').slideDown(slideSpeed);
        $('#success').slideUp(slideSpeed);

    }

});
});
    </script>
```

Fig 31 – Two factor authentication 1

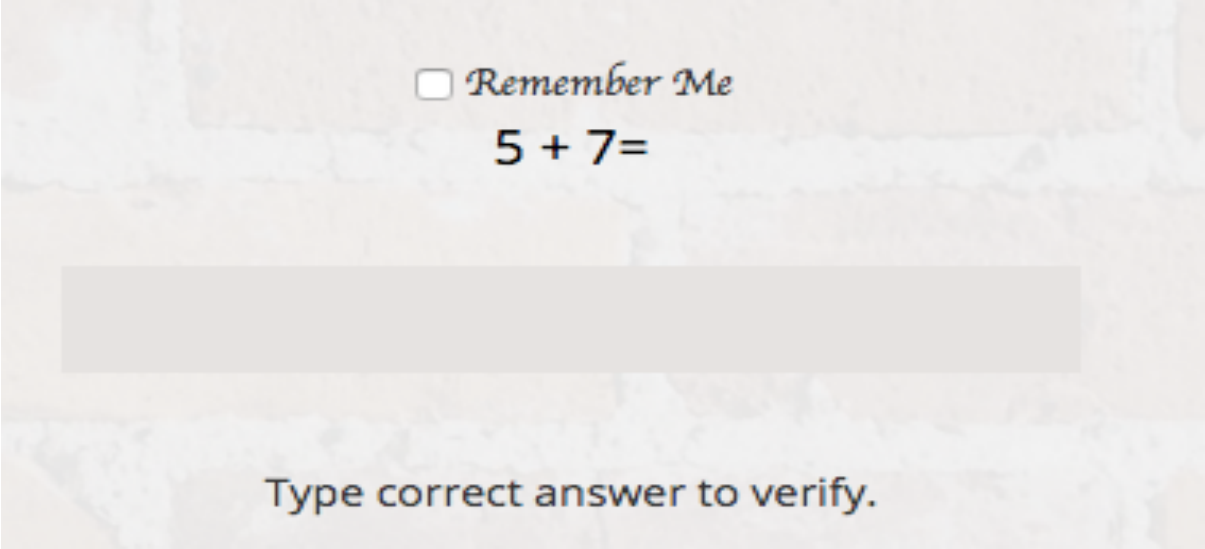

Fig 32 – Two factor authentication 2

3.5 PREPARED STATEMENTS (Preventing SQL Injection)

SQL injection is preferred to be the most performed attack on the website. To prevent SQL injection attack, prepared statements are used so that the attacker cannot add any malicious scripts into the database to fetch the data from the database.

3.5.1 signup prepared statement

```
if (isset($_POST['submit'])) {
    // session_start();
    $sql = "INSERT INTO signup(firstname, lastname, password, email, Question, Answer) VALUES (?,?,?,?,?,?)";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param("ssssss", $firstname, $lastname, $password, $email, $question, $answer);

     $firstname = mysqli_real_escape_string($mysqli, $_POST['firstname']);
    $lastname = mysqli_real_escape_string($mysqli, $_POST['lastname']);
    $password = mysqli_real_escape_string($mysqli, $_POST['password']);
    $password2 = mysqli_real_escape_string($mysqli, $_POST['repassword']);
    $email = mysqli_real_escape_string($mysqli, $_POST['emailid']);
    $question = mysqli_real_escape_string($mysqli, $_POST['question']);
    $answer = mysqli_real_escape_string($mysqli, $_POST['answer']);
```

Fig 33 – signup prepared statement

3.5.2 book page prepared statement

```
if(isset($_POST['book'])) {
$sql = "INSERT INTO book(Name, Email, Address, Seat, HeadCount) VALUES (?,?,?,?,?)";
$stmt = $mysqli->prepare($sql);
$stmt->bind_param("sssss", $name, $email, $add, $seat, $head);

$name = mysqli_real_escape_string($mysqli, $_POST['name']);
$email = mysqli_real_escape_string($mysqli, $_POST['email']);
$add = mysqli_real_escape_string($mysqli, $_POST['address']);
$seat = mysqli_real_escape_string($mysqli, $_POST['seat']);
$head = mysqli_real_escape_string($mysqli, $_POST['headcount']);
```

Fig 34 – book page prepared statement

3.6 CSRF PREVENTION

This security is done to prevent CSRF attack. Here a new key is generated every time the user requests in and out and when the page is refreshed.

```php
<?php
mysqli_connect('localhost','root','');
if ($_SERVER['REQUEST_METHOD']=='post')
{
if (!isset($_post['_token']) || ($_post['_token'] !== $_session['_token']))
{
die('invalid csrf token');
}
}
$_session['_token']=bin2hex(openssl_random_pseudo_bytes(16));
?>
```

Fig 35 – CSRF prevention

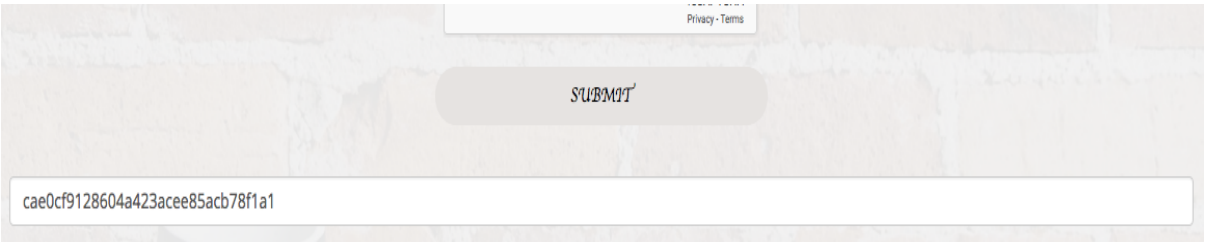

Fig 36 – CSRF key generate

3.7 SESSION MANAGEMENT

This security feature is done to make sure only one user is using the website login at a single time in a single browser and a single computer. Here the cookies are set to the user's email and password and it is stored as cookies in the browser and when the cookie is destroyed, the cookie is erased from the browser's memory.

```
    if($email == $sql) {
        if (isset($_POST['remember'])) {
        setcookie('email', $email, time()+60*60*7);
        setcookie('pass', $pass, time()+60*60*7);
    }
    session_start();
    $_SESSION['email'] = $email;
    header("location: book.php");
}
```

Fig 37 – Session Management

3.8 SSL CERTIFICATION

This is the major security to be done to the website to establish secure website. By using the v3.txt dependencies and accepting the expectations asked by the website. The browser exports the certificate provided by the localhost which is added to the Keychain Access in the certificates section only then the website becomes secure. The server key and root key is generated.

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation,
keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = localhost
```

Fig 38 – v3.txt

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn

[dn]
C=IR
ST=dublin
L=dublin
O=nci
OU=cyber
emailAddress=sriramkalyanraman@gmail.com
CN=sriram kalyanraman
```

Fig 39 – server.conf file

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDapx5EgYEzc3Sj
iLDP03bSUlYKlm4AgxmqoiGRNmLjd+3+eKSm4W1sRdLwZIEAOcNF/lKHUQaQX7Ob
8xLgPi57IjOOpfRnafsb+6xsjwBAjo2OhqVqCEzuhjJtQwtJA7I4KovhWVCb0voe
AXBvUulKNZz74Y09ddxv6AXcGWL+TW6N0qsKaZl0zGPiwYqRTmno6aGYo+63CyoS
BsCKXP4n+4tEz9vGPC7fCj8Gv6LT7sNnOv0zQkIW7eTZsRVAHawCa3YRQKrPJmGG
TI/cpDZwgfuzOeUaJebOK3mdIcvJTHg3dmNK6C3YgzArcKgqlk0bdnq91j1900E2
ekzGfZ+3AgMBAAECggEAKKBelln9uKKeLHiMacEf1cpopET5IW06iqLfW8zgKOWy
udKOvVcrbq+2luYFthYFe27axLvAXXj6tWI/wpyrJEPhkoKxW7msZ7GmgvE7GcOi
97V67K2oJbWXjwejBCkEpzz4iqF+wcU8pfCchILCSACYL5DkHmRU1hIlbJTm+Nrw
XLCk9hxScG/Yf4iq6vArLqmIaTxi1zVql4hNsVtJxVCdWBN3jgSQA0vc9K7+Ur5b
1TcotcJU24A4t0oBry/tl4v6/X4nN8H8nkOXUtRXbQk16SrAfLdwNoq8ecL0QKIm
5ZiuvwmGkx+WyhcrdmNDq1Vgjn1uZze5gGR20GxlGQKBgQDxXhVHw1pm/NZrppE/
u85C6NrX5KvtrK2h029yAYnHKh2LIwIsuwVzoNUDLqGODU5UK9Lf3azkUP0JcpWJ
WQHrIieVVx9v5gepG3dKEY3U6V7PNyXm7i5ZlqM5dUpKcMLAh4jy3cUnAAaujSgH
BhD0SUzi7bg6V2Hnphn+7HO48wKBgQDn6INO072tsN1ftBoKTQlz3WAasHuxKSNS
swp9BiUeYb2Rno8wfqZinKsAKzPRrNvgSHD3iUdbbBCY+sfagSzED8QA0dHqX1m0
JPKLr85pvXK0r7J+jQMII+XIWtBrja0cCVPAh99nM9mT+nNQ6AAvPtSj8jeZjsyo
4deLKGivLQKBgDh1i4DH34OrrPRteBhyc+MO2PIbWWQ1OkBaVv1SAsqFvXdobv/L
l5DelDU0WTIDRslFnwQitGiL0US+euAv49T+nke+o2nnYu9Zr2OyC308QaMzATit
ikYGiJPOLdyP1951VcGJwq5GwUlfV3edIaYi50tF1AxBmHM5VWdeJAulAoGBAL0b
68eh/vDeYwch1IJF48lp4m6hL5x+EWHkwk7FbgtiZLWUeOtwu7l6TC9tC2qV7Q7t
w8D3Xwr8CmTVS5ew8pJG0OxiLS1T59BwuKCUElxDsoAJLiLoVmqS61QASikKdPmB
2ZwMsG515502BEcbiNr9/2RZpmFAA0BMOfouT39ZAoGAahrx0KOLA0MIuG61xFJu
U3VG0JVXHcD3E5bSlEXcz4sImD8mmEBtHiOwiPEMdPSjlkqbZTuY3ruF3zu3Y9dM
d2ZdRX05EP4JuMzWx+uVNDByfLfD8SD0SpHvRUsCF2MKITrfHKCPPOTdzqBDVD5m
zbAceCiFpsUKnqwX9u3AxUY=
-----END PRIVATE KEY-----
```

Fig 40 – Server ley

Fig 41 – rootCA key

Fig 42 – Localhost certificate

3.9 ADMIN ROLE

The role of the admin is to view all the data input by the customer user and the admin also has the privilege to modify the data fed into database by the customer user.

```
<table class = "table">
    <tr>
    <th> ID</th>
    <th> Name </th>
    <th> Email </th>
    <th> Address </th>
    <th> Seat </th>
    <th> HeadCount </th>
    <th> Action</th>
</tr>
<?php
    $i = 1;

    while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        $name = $row['Name'];
        $email = $row['Email'];
        $add = $row['Address'];
        $seat = $row['Seat'];
        $headcount = $row['HeadCount'];
        $id = $row['id'];
    ?>
        <tr>
            <td><?php echo $i; ?></td>
            <td><?php echo $name; ?></td>
            <td><?php echo $email; ?></td>
            <td><?php echo $add; ?></td>
            <td><?php echo $seat; ?></td>
            <td><?php echo $headcount; ?></td>
            <td>
                <a href="admindb.php?delete=<?php echo $id; ?>" onclick="return confirm('Are you sure?');">Booking Taken</a>
            </td>
        </tr>

        <?php
            $i++;
        }
        if(isset($_GET['delete'])) {
            $delete_id = $_GET['delete'];

            mysqli_query($dbconnect, "DELETE FROM book WHERE id = '$delete_id'");
        }
        ?>
</table>
```

Fig 43 – Admin Privilege

3.10 SECURTY QUESTION VALIDATION

This security feature is done on the place where the user changes from old password to new password. It is possible only when the user feeds in the correct answer for the question which was earlier done in the signup by the user.

```
  <form action="forgotpass.php" method="post" class="register-form">
    <div class="form">
    <input type="email" name="email" placeholder="Type in the email id" id="mail" autocomplete="off" required />
    <span id="email"></span>
    </div>
     <center><div class="dropdown">
        <label>Security Question</label>
    <select type="question" name="question" class="question">
    <option value="What is your birth city?">What is your birth city?</option>
    <option value="What is your pet name?">What is your pet's name?</option>
    <option value="What is your Mother name?">What is your Mother's name?</option>
    </select>
</div></center>
<div class="form">
    <input type="text" name="answer" placeholder="Type in the answer" autocomplete="off" required />
    <span id="answer"></span>
</div>
    <div class="form">
        <input type="hidden" name="_token" class="form-control" value="<?php echo $_session['_token'];?>"/>
    <input type="submit" name="submit" value="submit" class="btn btn-signup" autocomplete="off" />
    </div>
  </form>
</div>
</div>
```

Fig 44 – Security question

3.11 BROKEN AUTHENTICATION

The broken authentication was done in the places where the user provides his/her credentials to the browser. Attacker when tries to page source the code using the browser, the validation is done in another PHP page which restricts the attacker and also the user to view the page. The page redirects to the previous page when the user tries to access the page

3.12 XXE PREVENTION

This security is done to make the website secure from XML eXternal Entity. XXE attacks are done in the places where the validation is done using the username input field. But in this website the validation is done using email address. The email field validation is done using javascript. Only when all the requirements of the javascript is satisfied, the field is verified. The validation blocks values which corresponds to XXE attack.

```
else if(email==""){

  document.getElementById('emailid').innerHTML = "**Please fill the email field".fontcolor("red");
  return false;
}

else if(email.indexOf('@') <= 0){

  document.getElementById('emailid').innerHTML = "** @ Position invalid".fontcolor("red");
  return false;


}

else if((email.charAt(email.length-4)!='.') && (email.charAt(email.length-3)!='.')){

  document.getElementById('emailid').innerHTML = "** . Position invalid".fontcolor("red");
  return false;

}
```

Fig 45 – XXE Prevention

3.13 DATABASE LOGIN CREDENTIAL

```
$username = "sriram";
$password = "rSDO0Ir40Iwz8CVX";
$hostname = "localhost";
```

Fig 46 – Database login credentials

4 GRAPHICAL USER INTERFACE (GUI)

Graphical User Interface plays a major role in the development phase of a website because the user must feel comfortable to use the website. The GUI should not make the user feel difficult to traverse between the pages. The developer must ensure for the correct traversal of pages in the website. The website must be user-friendly and it must not show any hinge in the course of hosting in a public domain. The screenshot of the GUI of the website is shown below., [4]

4.1 Login Page GUI

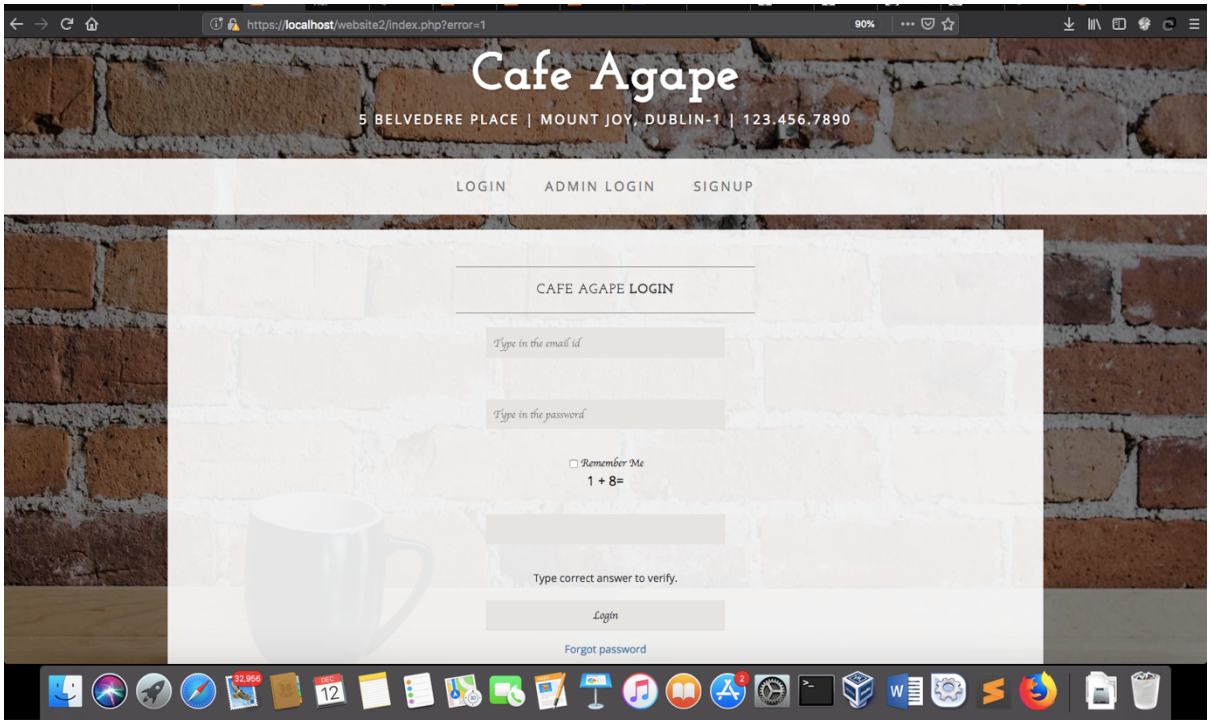

Fig 47 – Login GUI

4.2 Signup GUI

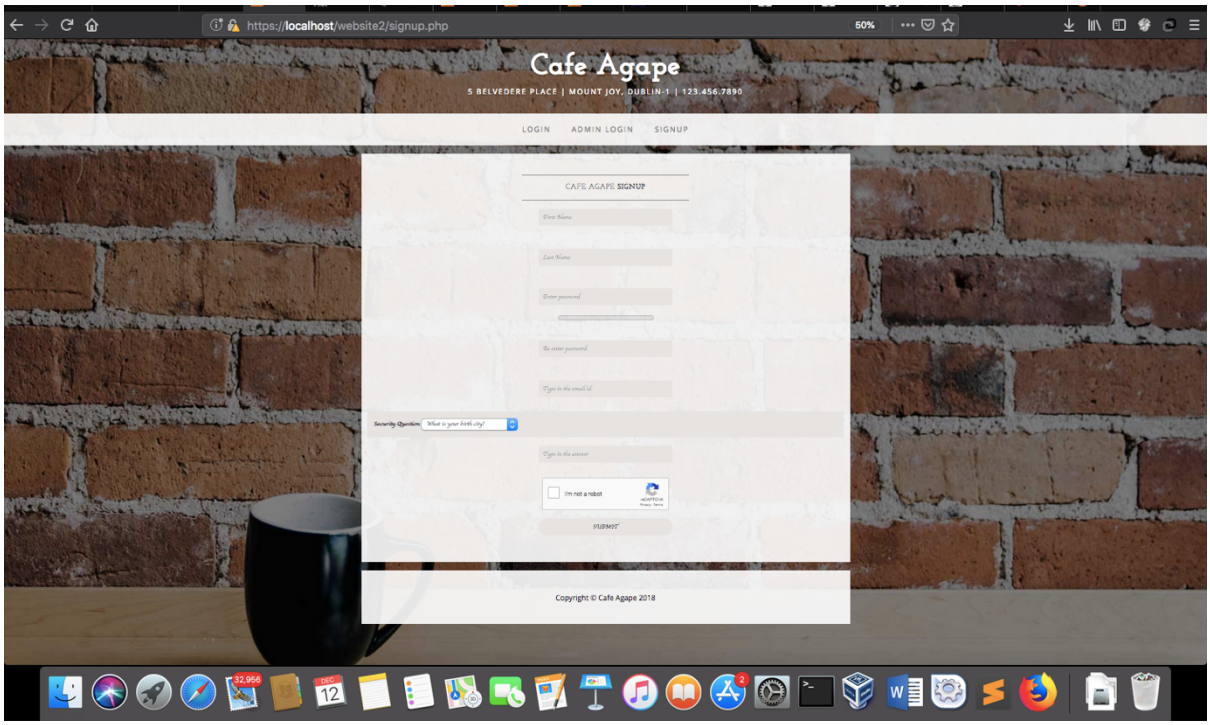

Fig 48 – Signup GUI

4.3 Forget password GUI

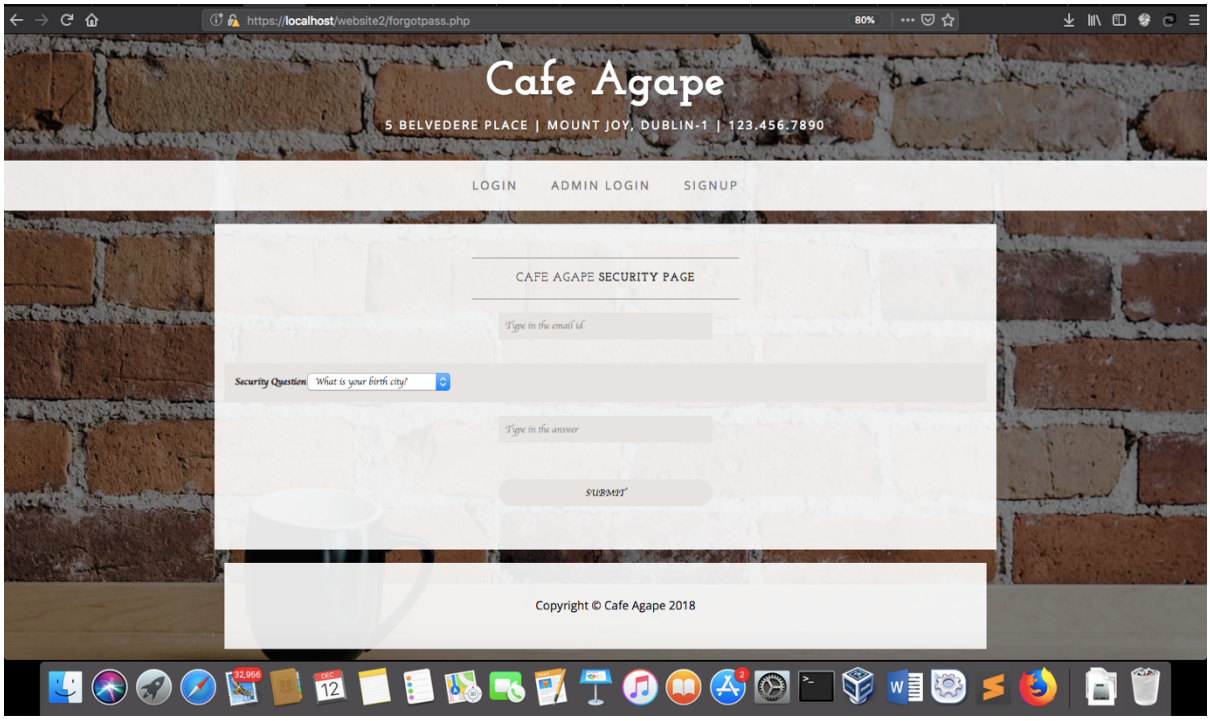

Fig 49 – Forget password GUI

4.4 New password GUI

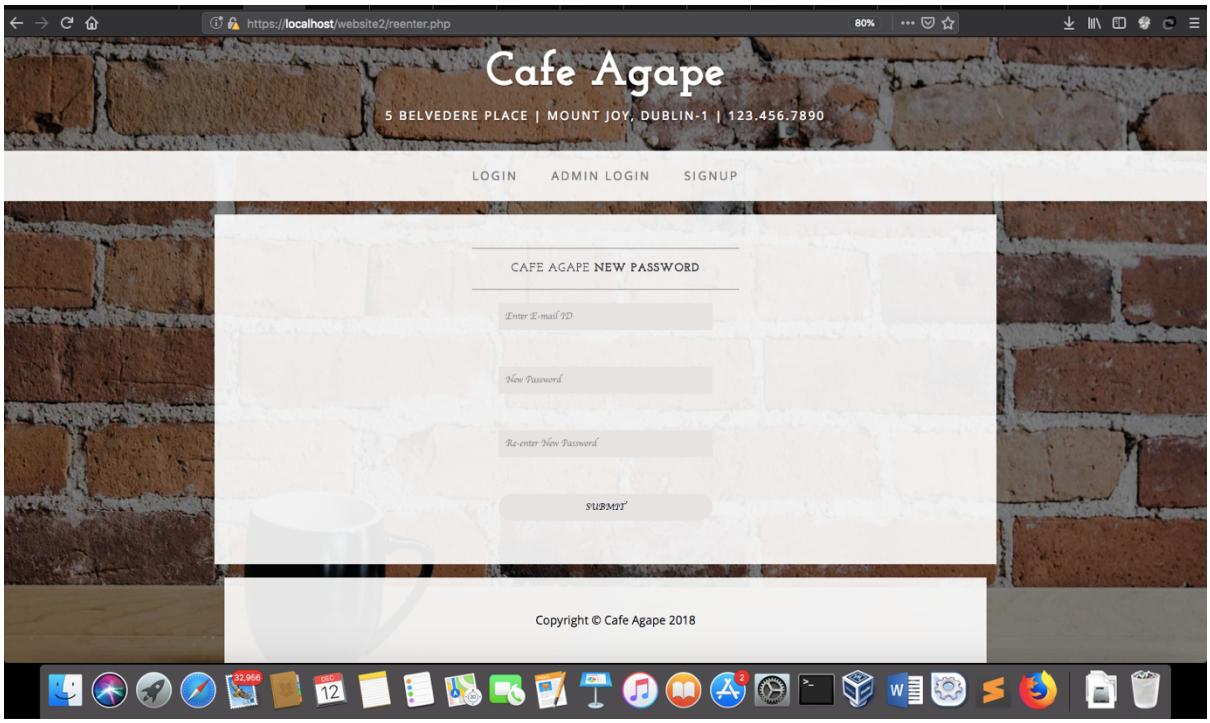

Fig 50 – New password GUI

4.5 Book page GUI

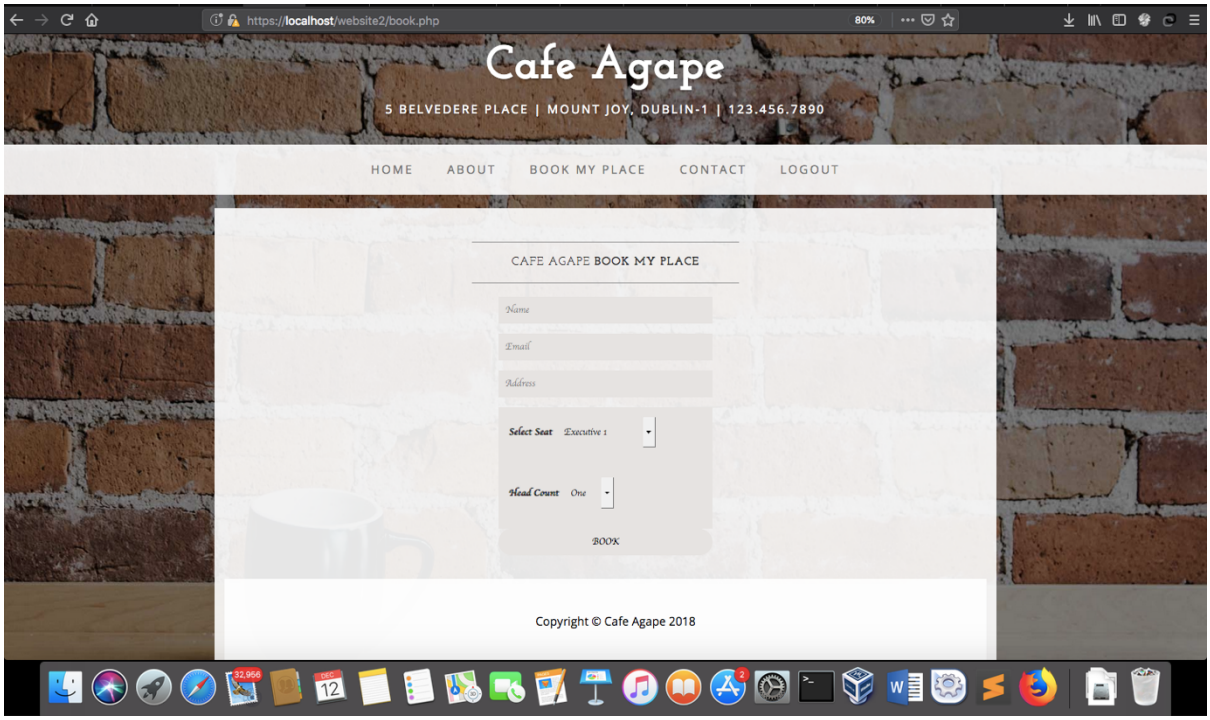

Fig 51 – Book page GUI

4.6 *Home page GUI*

Fig 52 – Home page GUI

4.7 *About page GUI*

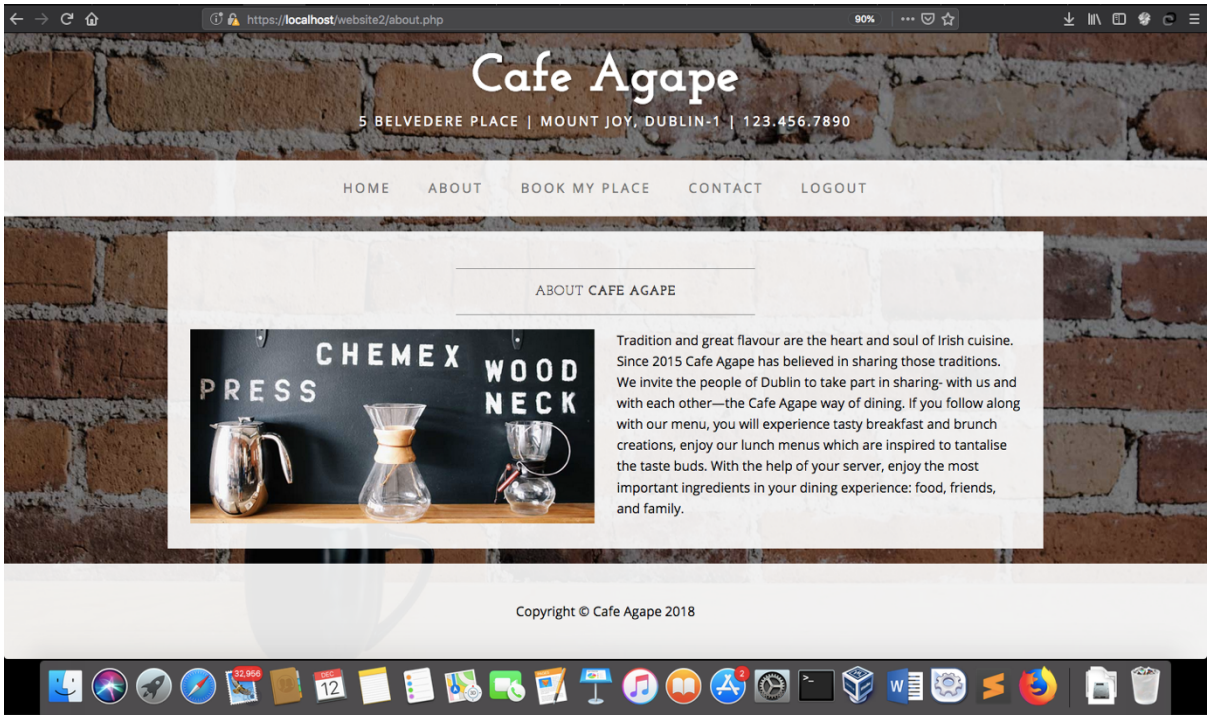

Fig 53 – About page GUI

4.8 *Contact page GUI*

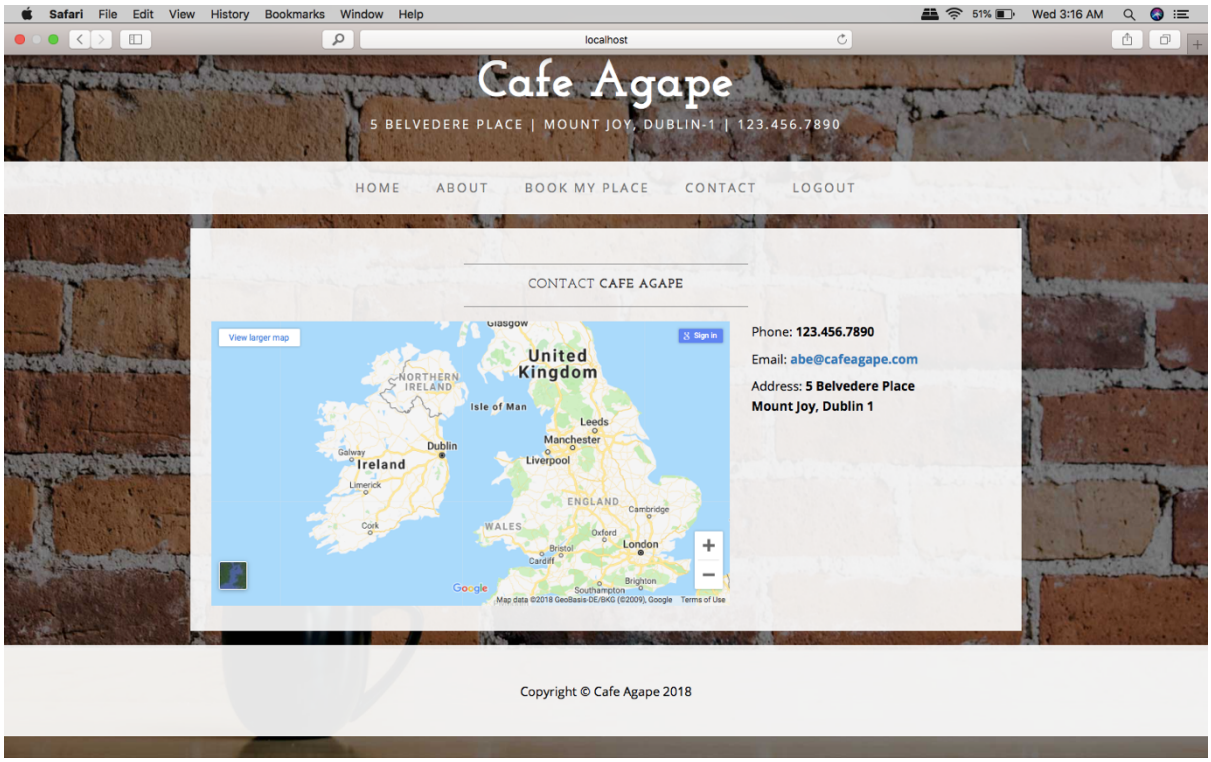

Fig 54 – Contact GUI

4.9 *Admin page GUI*

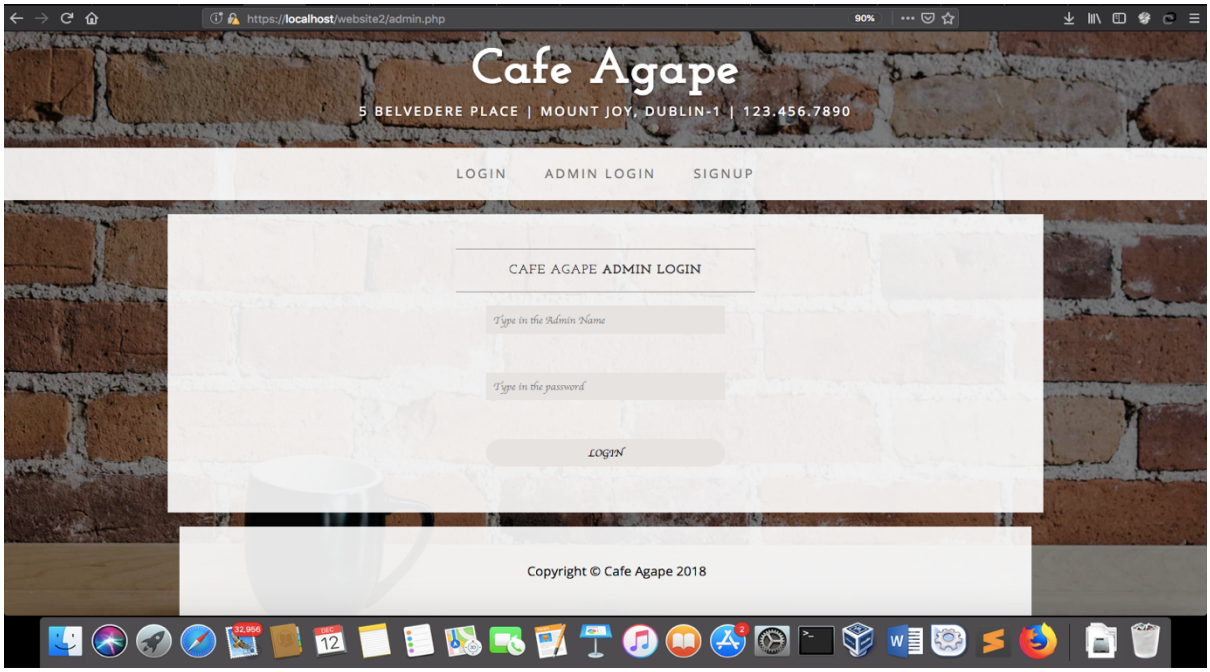

Fig 55 – Admin GUI

4.10 Admin DB page GUI

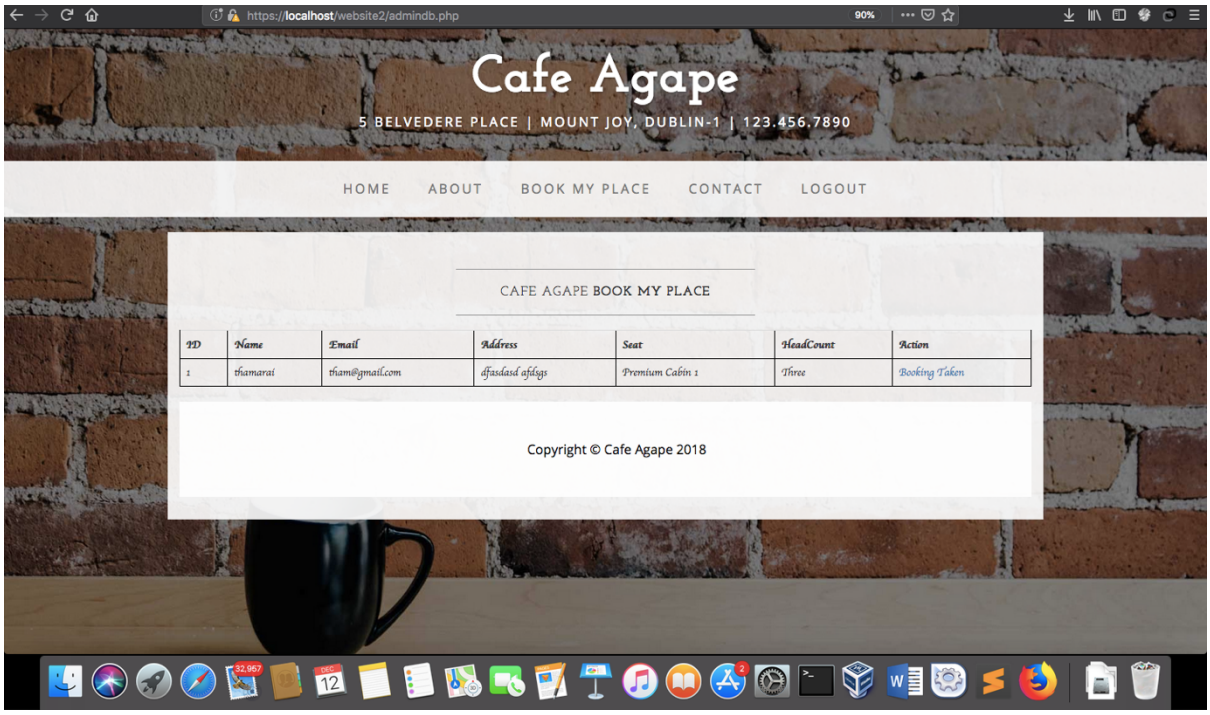

Fig 56 – AdminDB GUI

5 TESTING

Testing is the process finding the correctness of the website. It finds the errors present in the code and also evaluates the functioning of the code and the complexity analysis of the code. There are many ways of testing a project. By performing testing operations the developer comes to know the vulnerabilities, code faults and security patches present in the project.

(Ref: https://searchwindevelopment.techtarget.com/definition/testing)

5.1 PEER CODE REVIEW

Peer Code Review has been done on entire project to find the vulnerabilities and which was informed and then rectified for the wellness of the website.

Code Reviewer – Hem Chandar Sundar (x18113583)			
File Name	**Line of Code**	**Security / code issue found**	**Fixation of the issue**
Book.php	123-140	Dropdown data failure error	Using the name function the error was rectified and hence the data with no error falls in the database
Signup.php	24	No password hashing was done	Password is hashed using BCRYPT algorithm which is preferred as the second most secure way of hashing password
Index.php, signup.php	On general	Command lines was instructed to remove for the quick execution of the code	All the command lines in the code files were removed for the code storage and code execution
Admindb.php	127-135	No role was given to admin	Role has been provided to the admin for modification of the database contents
Index.php	27,28	Session management was not done properly	This error was rectified by setting cookie details and storage of cookie details in the web browser

Contact.php	90	The google map functionality was not executing properly	Since the default browser did not support the gmaps feature, another browser was used to execute the google map
Csrf.php	10	Only one token was generated when CSRF was initially implemented	bin2hex(openssl_random_pseudo_bytes(16) command line was used for continuous refreshing of CSRF token generation

Table 2: Peer Code Review

5.2 FUNCTIONAL TESTING (TOOL BASED TESTINGS)

Functional testing of the website denotes the complete testing of the website components present in the website folder. Initially after coding all components, we performed testing in an online tool called COLLABORATOR SMARTBEAR. It is certified trusted tool which completely scans all the lines of code imported into the website. The functional testing performed here is based on LOC/hr rate.

Review Detail Report

Base Information		Defect Information		Time Information	
ID	3	Number of Defects	0	Created On	2018-12-12 16:26:59
Status	Inspection	Number of Open Defects	0	Finished On	2018-12-12 17:05:06
Title	Cafeagape	Number of Fixed Defects	0		
Creator	Administrator	Number of Tracked Externally Defects	0	Total Person Time	00:03:27
		Number of Comments	0	Wall-Clock Time	00:38:06
				Total Reviewer Time	00:00:05

Fig 57 – Review report

This test initially was done to find the correctness of the code in the host name of the website Cafeagape. All the lines of the file will be tested from top to bottom. But this test doesnot contribute to security testing.

Materials Summary: Versions of Each File

Custom Statistics		LOC Metrics	
Total Files	13	Num Changelists	1
Code Files	13	LOC (Uploaded)	1936
Num Documents	0	LOC (Reworked)	1936
Images	0	Document Pages	0
URLs	0		
Other Binaries	0		
Inspection Rates		**Defect Density**	
Inspection Rate (Uploaded)	1393920 LOC/hr	Code Defect Density (Uploaded)	0.00 defects/kLOC
Inspection Rate (Reworked)	1393920 LOC/hr	Code Defect Density (Reworked)	0.00 defects/kLOC
Document Inspection Rate	0.00 pages/hr	Document Defect Density	N/A defects/page
Image Inspection Rate	0.00 images/hr	Image Defect Density	N/A defects/image
URL Inspection Rate	0.00 urls/hr		

Fig 58 – Report Summary

5.3 PERFORMANCE TESTING

This type of testing ensures the availability of the website and also the response captured by the tool in executing the website. APACHE JMETER is the performance analysis tool which provides the graphical representations of the performance of the website. In this performance testing, the website is made to hit using 600 anonymous users and tests the availability of the website and also correct happenings in the website that no data contents doesn't get hinged.

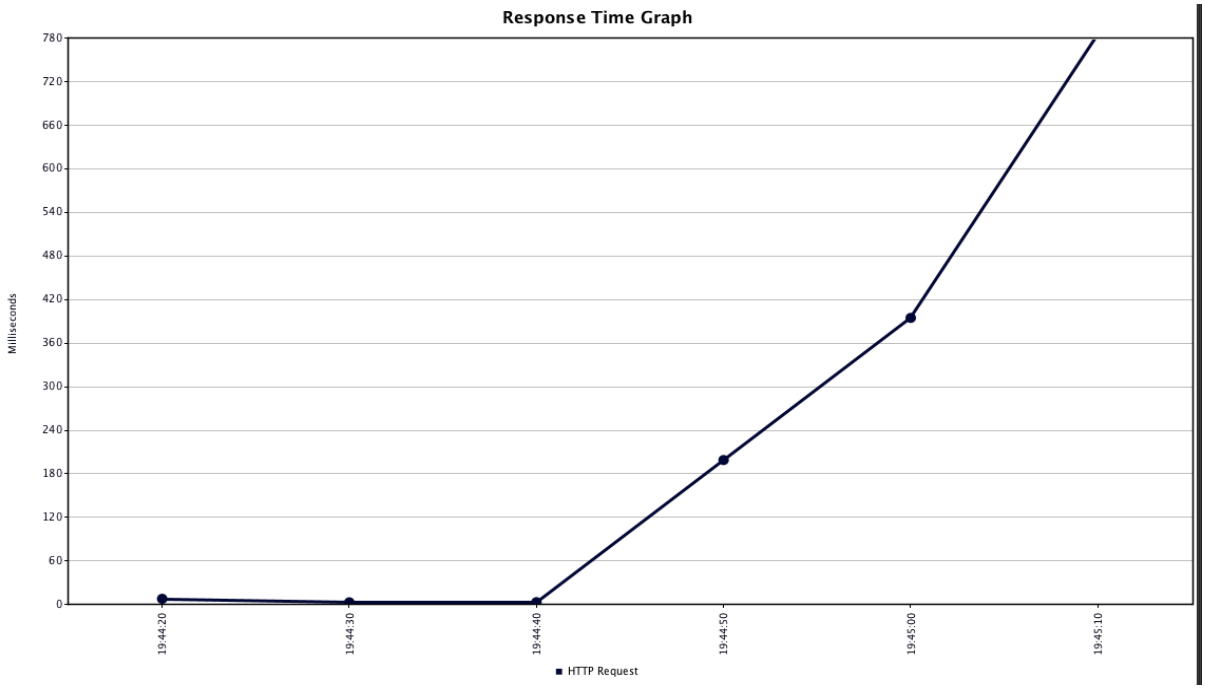

Fig 59 – Performance Graph

5.4 SECURITY TESTING

Security Testing is done for acknowledging the website to be secure a platform for the user to perform activities. Since the website is not publicly hosted, there are not possibilities of performing attacks on the website. Security testing was done using a commercial tool named VISUAL CODE GREPPER V2.1.0. The major vulnerability encountered in the website is XSS attack which can be rectified when the website is publicly hosted.

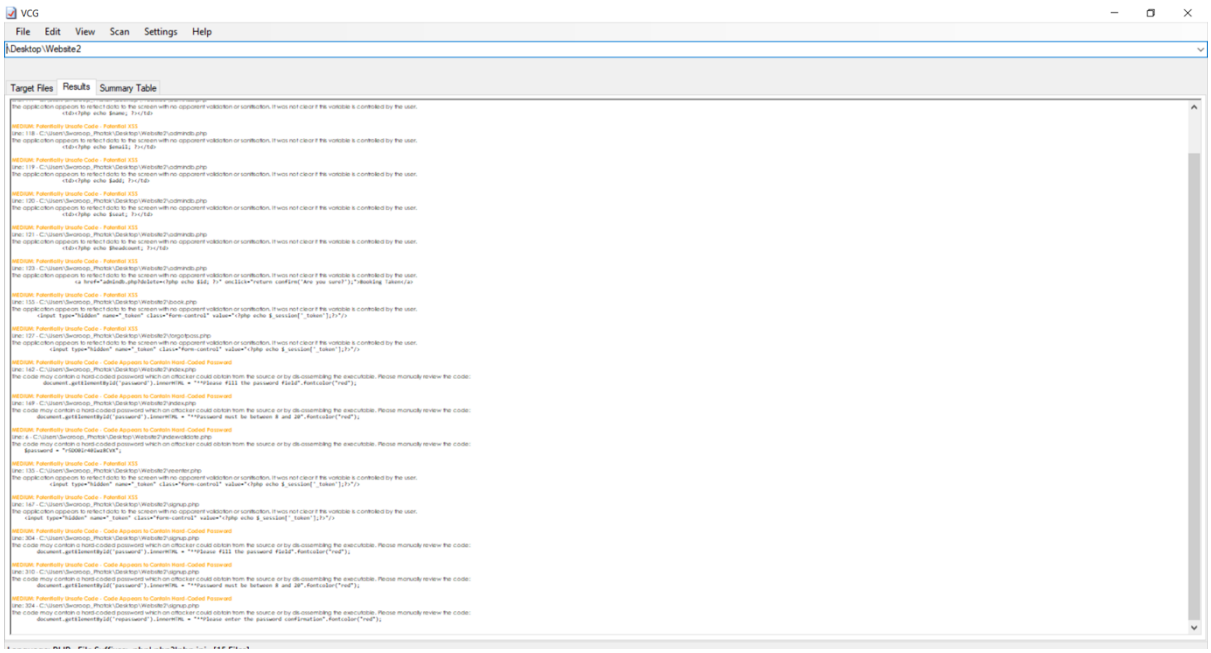

Fig 60 – Security Testing

Since the VISUAL CODE GREPPER tool was not available for Apple Macbook, the security testing was done on a peer's computer which supported this tool.

6 RISK ASSESSMENT

- Since the website is not a publicly hosted, no hard vulnerabilities are found.
- When the website acquires a domain, there are more probabilities for the data to get tampered.
- Tools like Burpsuite, Wireshark can be used to capture the packets from client to the server and the data can be decrypted.
- During post decrement process, the credentials are transmitted through the URL (Uniform Resource Locator) where the attacker can sniff the URL to capture the values transmitting through the URL.
- Server must hold more capacity to handle the load from the user end. For a weak server, DDoS (Distributed Denial of Service) can be performed in the website.
- Recoverability must be assigned to avoid asset destruction
- Database must be monitored regularly and the sanitization process must be done periodically.
- There must be a page source backup anytime to provide uninterrupted service of the website to the end user.
- Website infrastructure must be maintain properly for preventing GUI attacks on the webpage[5][6]

7 CONCLUSION

The website was develop in a way to make the GUI user friendly for the end user. Alongside the security features were also implemented in the website to make sure the website is free from attackers. On applying the security features, the website attains competency to be release on market and also the website was built in an outsourceable manner. The website is given basic and mandatory security functionalities which makes free from simpler and intermediate attacks because the security has been implemented on regard to the OWASP top 10 security 2017 amendments.

8 REFERENCE

[1] tutorialspoint.com, "PHP Introduction," www.tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/php/php_introduction.htm. [Accessed: 11-Dec-2018].

[2] Browser Statistics. [Online]. Available: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp. [Accessed: 11-Dec-2018].

[3] tutorialspoint.com, "Javascript Tutorial," www.tutorialspoint.com. [Online]. Available: https://www.tutorialspoint.com/javascript/index.htm. [Accessed: 12-Dec-2018].

[4] "What is a GUI (Graphical User Interface)?," Computer Hope, 13-Nov-2018. [Online]. Available: https://www.computerhope.com/jargon/g/gui.htm. [Accessed: 11-Dec-2018].

[5] "Risk Management for Your website," Where Our Ethics Come From | IRMI.com. [Online]. Available: https://www.irmi.com/articles/expert-commentary/risk-management-for-your-website. [Accessed: 11-Dec-2018].

[6] Nimrod Luria, "The Website Risk Assessment Tools Every Security Manager Must Use," Infosecurity Magazine, 01-Jun-2015. [Online]. Available: https://www.infosecurity-magazine.com/opinions/website-risk-assessment-tools/. [Accessed: 11-Dec-2018].