A REPORT ON

Boomerang- Trip Planner and Advisor

PROJECT REPORT

BACHELOR OF TECHNOLOGY

(Computer Science & Engineering)

SUBMITTED BY

Himani Chaudhary

Jasjit Singh Rudra

DECEMBER 2020

ABSTRACT

The Project Titled as '**Boomerang (Trip Planner and Advisor)**'is designed as a better alternative to the standard Tour Guiding and Trip Planning System. The main objective of this Project to design a product for the users where they can book hotels, trip packages, know about various places ,have a tour recommended to users and most importantly, provide a secure platform and easy to use interface. It eliminates the need to move from one app to another to get the sources we want.

Boomerang allows users to browse various Trip Packages and help users to get a tour worth their money. Many popular destinations are available in Trip Packages at affordable prices. The project consists of a user website, an administrative panel (CMS panel) and app. Through administrative panel, new users can be added, deleted or user credentials can be modified.Through the website, new users can register themselves. Users can check out various places Boomerang offers Trip Packages to. Moreover, users can register complaints/queries through the website.

Trip Planner includes various modules such as User Sign-in and Login, Trip recommender, Plan-a-trip, Trip Packages, Hotel Booking, Payments and Settings. The methodology of each module is as follows:-

A) Sign-in and Login

On the start of the python app, the user is shown a login screen where he/she can enter his/her credentials to gain access to the features of this program. Non registered users can click on sign-in button where they have to give certain information after which they will be registered and can use other modules. The sign-in information is stored on a table in a database hosted through a local server. From the same sign-in table, the username and password is checked and user can gain access if the input matches with the data stored.

B) Trip Recommender

In this module, the user can enter certain keywords or tags that he/she wants to see at the destination they want to visit. A list of tags corresponding to places is stored in a table and if three or more tags match, that recommended place is shown to the user.

C) Trip Packages

When the destination is chosen or even if user wants to check rates of certain tourist packages, Trip Packages is the module to go to. Various Trip Packages to different continents and places are available and users can select the one that they like.

D) Hotel Booking

After the destination is set or if user wants to search for certain hotels or book premium suites, then they can use Hotel Booking module.

E) Payments

After the selection of hotel and a trip package, the user is guided to payments tab where he/she can pay for the trip. Users enter their credit/debit card information to pay for the services.

F) Settings

The Settings module is used to update profile information of users. User can change his username or password from settings.

TECHNOLOGIES USED

HTML: HTML is used for creating elements in the web pages and creating the web pages.
CSS: For designing and making the web page looks attractive.
BOOTSTRAP: It is used for creating the layout. Helps inincreasing the development speed, and responsible for the responsiveness of the website.
JAVASCRIPT: JavaScript has been used for the working of various bootstrap features.
PHP: PHP is basically used to make all the calculations and making all the server side processes.
MySQL: The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically.
Python: It is a very powerful, high level programming language used to design the Boomerang Application and program its various modules.

List of Figures

CONTENTS

1. Introduction

1.1 Problem Definition

Considering the hectic lifestyle most of us have today, a vacation sounds like the perfect way to relax while also getting a chance to explore new places, cultures and cuisines. The positive effects vacations/trips can have on us are underrated. To put it in other words, people associate vacations only to a holiday from work, but vacations are scientifically proven to have various health benefits, such as:

- **Improved Physical Health**
  Stress contributes to heart disease and high blood pressure. Taking vacations at appropriate intervals can help reduce risk of coronary heart disease or heart attacks.
- **Improved Mental Health**
  According to neuroscientists, brain structure is altered by chronic exposure to Cortisol, which is the stress hormone and also a major contributing factor to anxiety and depression. Vacation helps heal the mind and body in a way that it couldn't if it was still under pressure.
- **Greater Well Being**
  It has been found in a study that the subjects' physical complaints, like, quality of sleep and mood had improved as compared to before vacation.
- **Increased Mental Power**
  After the vacation, the workers are found to be more productive, focused and alert .
- **Improved Familial Relationships**
  Vacation gives us a chance to spend time with our loved ones and this helps in building strong relationships with them.
- **Decreased Burn-Out**
  Workers or employees who frequently go on a vacation are less likely to experience a burnout which makes them more creative and productive than their colleagues who are over-worked and under-rested.

Besides, these benefits, a vacation also makes us more confident as we interact with new people and also enhances our knowledge about the places, the languages, food etc.

The major problem that we face is the planning of the trip/vacation is research. We need to research potential vacation locations, select the one that suits the best after research about the cost, transport, route, weather etc. have been done. All this research work makes planning a vacation stressful and to save themselves from the stress, some people simply give up on the vacation.

This is when the role of trip planning services comes into play. The trip planning services i.e. websites and applications, make the research work easier for us. We can decide upon a location within a few minutes while also knowing about the various landmark places to visit, the climatic conditions of the place etc. Not only this, they help us book the mode of transportation, flights and rental cars for instance, and also a place to stay. So, basically, there is a whole team researching for your perfect vacation. Sounds impressive? Definitely, but what we turn a blind eye to is the fact that some of these websites might actually loot us rather than help us.

- They might offer free airline tickets/free amusement park tickets
- What they offer to us as a discount might be extra money flowing into their accounts
- They might hide certain overhead costs which might go over-budget for us.
- They might add fake reviews
- They demand money almost 60 days prior to the trip
- Their social media links might not be working
- The website may have too many errors (which is common if translations are used)
- They might ask for payment outside of the website where you found the booking
- They accept wire transfers or gift cards
- Show edited pictures of a different and cheaper place.

Taking these problems into consideration, the trip planning project was made for providing a secure way for the customers/users to make bookings for their trips where they can also find the legitimate pictures and deals for the locations. The website and application are made with a mindset of benefitting the customer.

1.2 Project Overview and Specifications

Boomerang actively employees Python 3 (for application), HTML 5/CSS 3 and Bootstrap 3.4.1 for its front end design. PHP has been used at the backend to implement various features in the website such as sessions, sign-up/login, settings and payment portal. The website is hosted using WAMP local server and WAMP database is used for storage of data. MySQL queries have been used for operations on data.

Detailed Specifications are given below:

1.3Software Requirements:

- Python 3.x

- Mozilla Firefox*, Google Chrome*or any other desktop web browser* -
  *(versions released after 2014)

- Pycharm IDE for running python GUI. (IDLE or any other Python interpreter may also be used)

- Any modern 64 bit operating system

1.4Hardware Requirements:

- Intel Pentium or above processor
- 512 Mb RAM or higher
- Mouse and Keyboard for inputs

2 Literature Survey

2.1 Existing Systems

There are several companies that are providing similar technologies and interface with their platforms for Trip Planning and Advice. Few of them are listed below:

1) MakeMyTrip Limited

MakeMyTrip Limited is an Indian online travel company founded in 2000. MakeMyTrip provides online travel services including flight tickets, domestic and international holiday packages, hotel reservations, rail, and bus tickets. Most of the projects at MakeMyTrip use Java to implement searching operations.

2) Trivago

Trivago is an online hotel and accommodation search website which compares prices for over 3 million hotels and apartments on 180+ booking sites in 190 countries worldwide. It has integrated over 175 million hotel ratings and 10 million hotel reviews for its users to find the most ideal hotel for its users.Trivago is operated on a hybrid infrastructure of both on premise machines and clusters on Google Cloud for comparison purposes.

3) Road Trippers

Roadtrippers

Roadtrippers is a web based software application, mobile app, and content provider that helps travelers plan road trips. The software lets users discover independently owned points of interest in the United States and Canada. The Roadtrippers application lets travelers plan trips, calculate time and gas expense, and choose from over 5 million independent locations in the United States to visit.The application relies on MapBox for location and tracking.

4) OYO

OYO

Oyo Rooms, also known as Oyo Hotels & Homes, is an Indian hospitality chain of leased and franchised hotels, homes and living spaces. OYO consists of budget hotels, lounges and rooms which can be easily booked from its website or app. Its website uses the following services to be hosted and its features to run on:-
Elastic Compute Cloud (EC2) Cloud Server & Hosting,Content Delivery Network (CDN) andAmazon CloudFront.

5) ixigo

ixigo

ixigo is an Indian AI-based travel e-commerce website founded in 2007. It aggregates and compares real-time travel information, prices and availability for flights, trains, buses, and hotel availability for its users in an environment similar to MakeMyTrip.

2.2 Proposed System

Boomerang project is designed for ease in making travel decisions, use reliable sources and provide a unified platform. Many people avoid traveling due to certain problems like booking of hotels, determining locations, making plans on what to do on trips, etc. To counter this, Boomerang provides carefully designed Trip packages and has the feature of Trip recommendations to provide its users ease of selection and a unified platform to complete all trip related processes in lesser time.

Boomerang application is designed using python, which is a very powerful and fast language. It provides more secure way of online transactions than other languages such as PHP.

2.3 Feasibility Study

a) Technical Feasibility: Boomerang (website + cms panel) can run on any modern browser, whose versions are released in/after 2014, as they can support HTML 5 and its features. The Python app is also not system dependent and is portable. There are no specific hardware requirements, but RAM higher than 512 Mb and Intel Pentium/Similar or above processor is recommended. A keyboard and Mouse is required for input

b) Operational Feasibility: Boomerang has a self-explanatory user interface and should face minimum resistance from users.

c) Economic Feasibility: There are minimal hardware and software costs for Boomerang as it is not system heavy. Cost of training for website and application maintenance would be average.

d) Legal Feasibility: Boomerang (website) follows all W3 standards of website design and thus should pass required quality tests/standards for software design.

e) Schedule Feasibility: A team size of 5 professionals for design/updates, 3 for maintenance and 10 for customer relations would be required during launch of project.

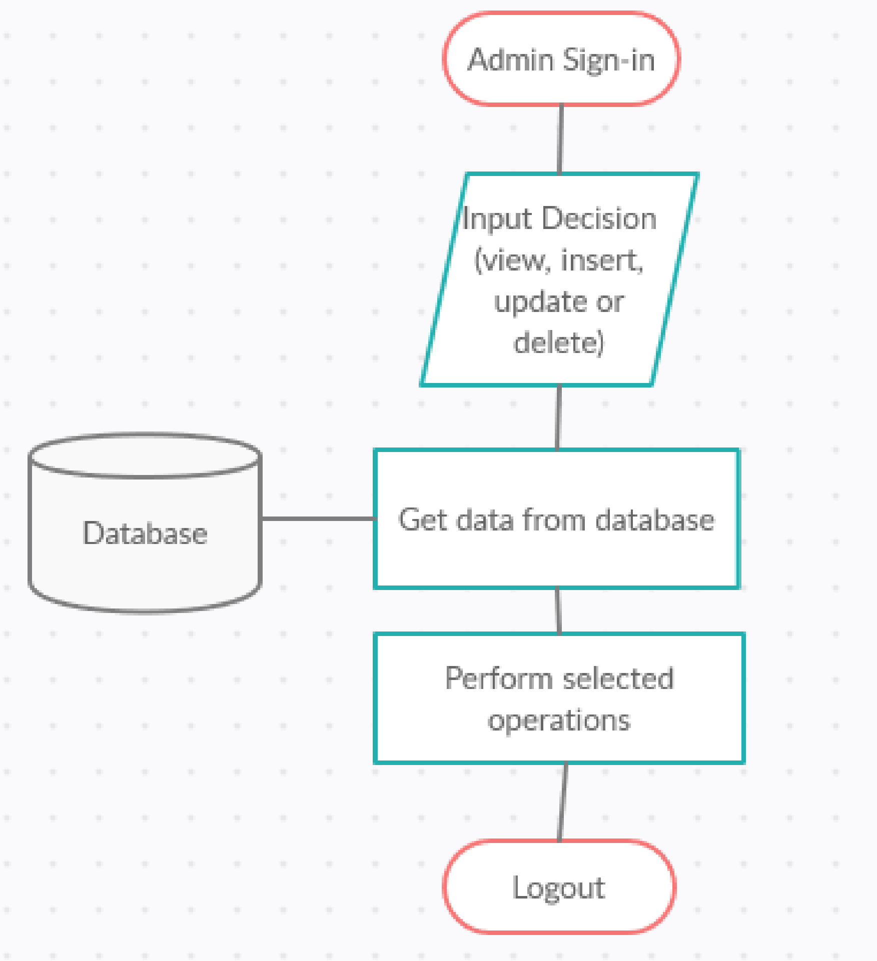
3. System Analysis/ Design

3.1 Requirement Specification

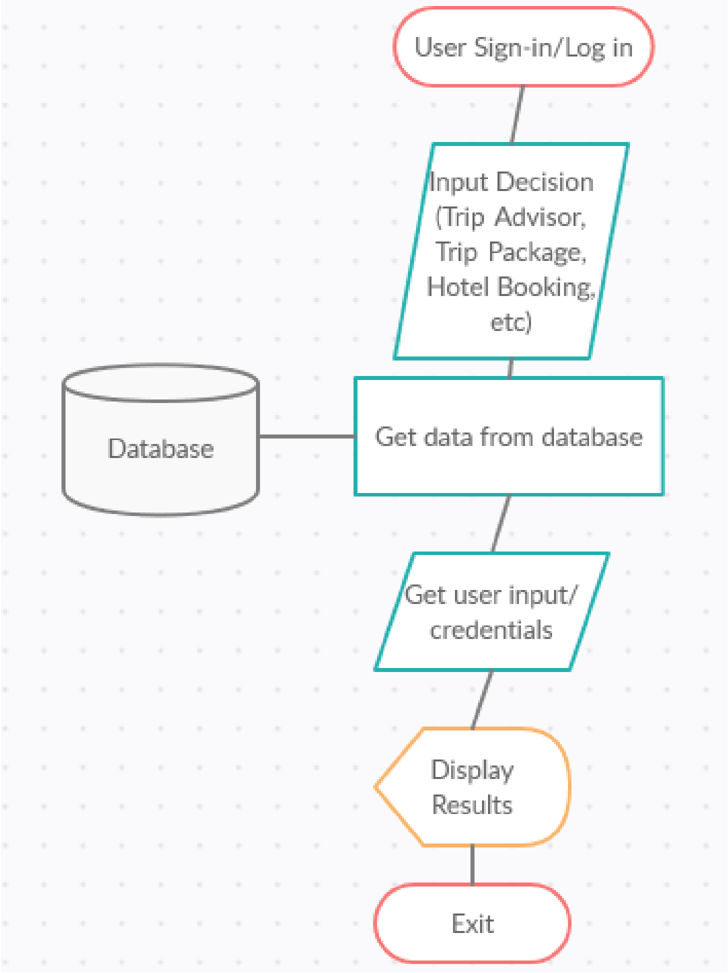
Please refer to page 2

3.2 Flowcharts / DFDs / ERDs

1) Flowchart

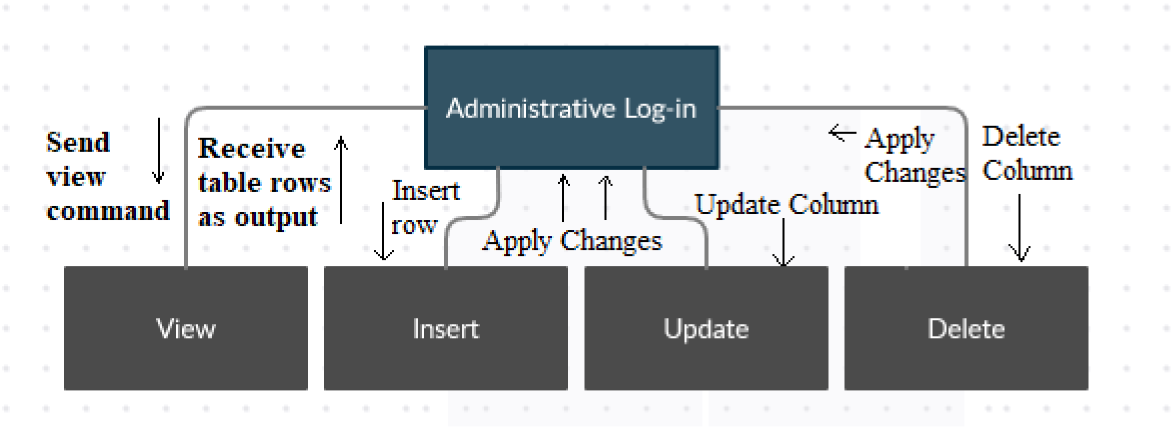

Admin Sign-in
Input Decision (view, insert, update or delete)
Database
Get data from database
Perform selected operations
Logout

Flow Chart- Administrative Panel

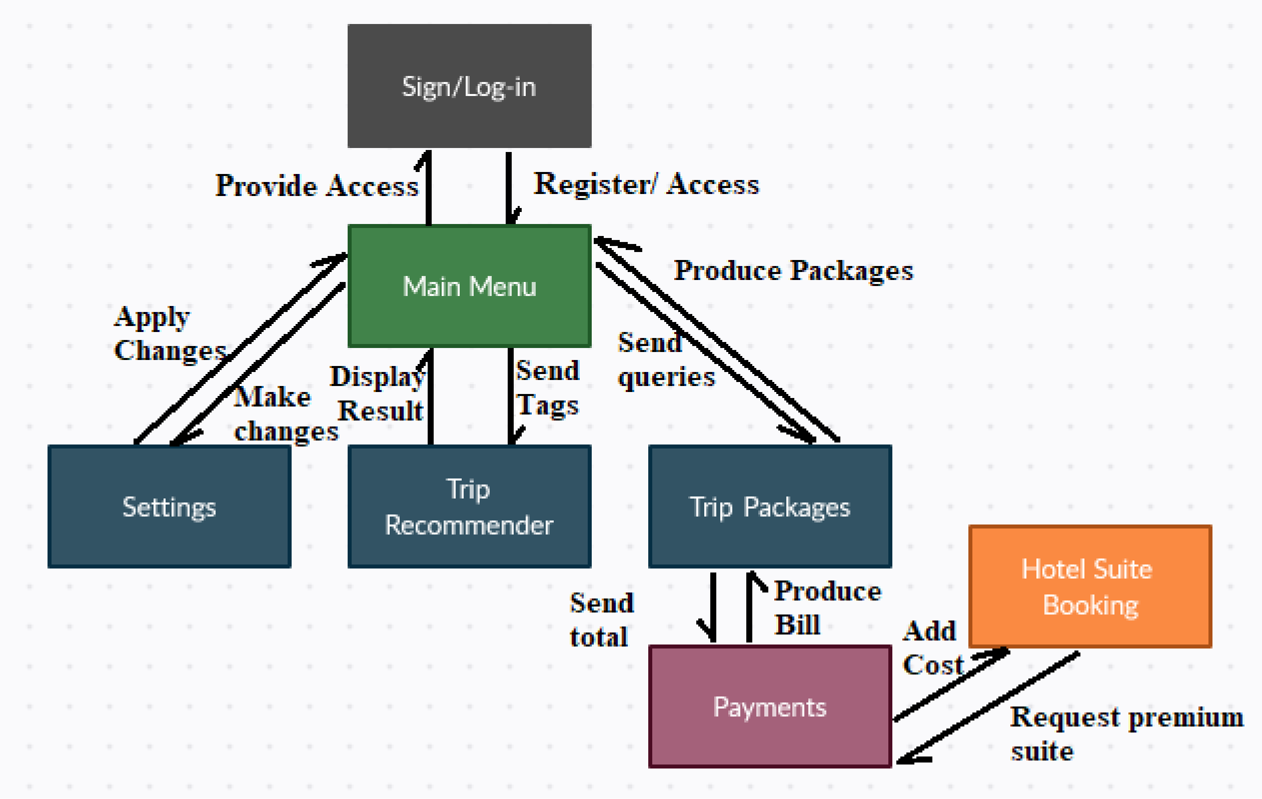

User Sign-in/Log in
Input Decision (Trip Advisor, Trip Package, Hotel Booking, etc)
Database
Get data from database
Get user input/ credentials
Display Results
Exit

Flow Chart- Python Application

2) DFD



DFD CMS Panel



DFD Python App

3.3 Design and Test Steps / Criteria

Boomerang (website + cms panel) was designed using HTML 5, CSS 3 and Bootstrap framework for frontend and PHP for backend processes. Data of website was stored using MySQL database.

Test Steps for website included:

a) Verification of Input- Under this step, form inputs were tested, and it was checked whether form action was correct and required input field were being used properly. Input fields such as phone number were checked upon that they can only have length of 10 characters.
b) Database verification- Under this step, inputs inserted into the tables was checked. For example: checking of primary keys, whether two users can have same e-mail id, etc.)
c) Testing of SQL queries: Whether correct MySQL queries have been used.

Boomerang (app) was designed using Python with external libraries such as mysqliconnector, which were tested using print statements to observe results. Mailing system was tested using Gmail SMTP server.

3.4Algorithms and Pseudo Code

A)For Website and CMS panel

3.4.1 Complaints

a) Algorithm

1) Start
2) Get USER credentials
3) Get USER message
4) Register message to database.
5) Exit.

b) Pseudo Code

```
Input (Name,Email);
Input (Message);
If (Name OR Email OR Message == NULL)
       { Print(“Please enter all credentials”); }
Else
{
 Get mysqli_connection("localhost","root","","boomerang");
  Insert into table(Name,Email,Message);

 Print (“Your query/complain has been registered”);
 }
```

3.4.2 Sign-up

a) Algorithm

1) Start
2) Get USER credentials
3) Register USER credentials to database.
4) Exit

b) Pseudo Code

```
     Input Credentials;

     If (Credentials == NULL)
            { Print(“Please enter all credentials”); }
     Else
     {
      Get mysqli_connection("localhost","root","","boomerang");
       Insert into table(Credentials);
Gotouserpage;      }
```

3.4.3 Log-in

1) Start
2) Get USER credentials

3) Apply linear search and get matching credentials.
4) If, no credentials match, display Wrong Credentials
5) Else, ‘Log-in’
6) Exit

b) Pseudo Code

```
    Input Credentials;

    If (Credentials == NULL)
          { Print(“Please enter all credentials”); }
    Else
    {
     Get mysqli_connection("localhost","root","","boomerang");
     Select credentials where credentials == Credentials;
                   if(Selected rows == 1)
                   {
                            Start Session;
                   Gotouserpage;      }
                   Else
                   { Print (“Invalid Credentials”); }
```

3.4.4 Update

1) Start
2) SELECT * from table where username is required username.
3) UPDATE Table
4) Exit

b) Pseudo Code

```
    Input Password;

     Get mysqli_connection("localhost","root","","boomerang");
UPDATE employees set First='$empfirst',Last='$emplast',Email='$empmail',Location='$emploc',PIN='$emppin',Location='$emploc',Phone='$empph',Username='$empuser',Password='$emppass' where Username=='$empuser'";

Return;
```

3.4.5 Delete

a) Algorithm

1) Start
2) SELECT * from table where username is required username.
3) DELETE from Table
4) Exit

b) Pseudo Code

```
Input choice;

     Get mysqli_connection("localhost","root","","boomerang");
```

```
"DELETE FROM 'login' WHERE 'Username'='$id'";
Return;
```

B) For App

3.4.6 Sign-up

a) Algorithm

1) Start
2) Get USER credentials
3) Register USER credentials to database.
4) Exit

b) Pseudo Code

```
    Input Credentials;

    If (Credentials == NULL)
           { Print(“Please enter all credentials”); }
    Else
    {
     Get mysqli_connection("localhost","root","","boomerang");
      Insert into table(Credentials);
Gotouserpage;      }
```

3.4.7 Log-in

1) Start
2) Get USER credentials
3) Apply linear search and get matching credentials.
4) If, no credentials match, display Wrong Credentials
5) Else, ‘Log-in’
6) Exit

b) Pseudo Code

```
    Input Credentials;

    If (Credentials == NULL)
           { Print(“Please enter all credentials”); }
    Else
    {
     Get mysqli_connection("localhost","root","","boomerang");
     Select credentials where credentials == Credentials;
                        if(Selected rows == 1)
                        {
                        Gotouserpage;      }
                        Else
                        { Print (“Invalid Credentials”); }
```

3.4.8 Settings

1) Start
2) Get USER password
3) Apply linear search and get matching password.
4) If, no password match, display Wrong password
5) Else, GET new password
6) Replace password with new password.
7) Exit

b) Pseudo Code

```
Input Password;

If (Password == NULL)
      { Print("Please enter all credentials"); }
Else
{
 Get mysqli_connection ("localhost","root","","boomerang");
 Select password where password == Password;
                  if(Selected rows == 1)
                  { Input new password;
                     Password = new password;
                  Return;}
                  Else
                  { Print ("Invalid Credentials"); }
```

3.4.9 Tag Search

a) Algorithm

1) Start
2) GET user choices
3) Search corresponding values in database
4) Display value with most matches.
5) Exit

b) Pseudo Code

```
Input choice;


      Get mysqli_connection ("localhost","root","","boomerang");
SELECT Country, Count(*) FROM locationsearch WHERE Tags LIKE %s LIMIT 1
            myconnect.execute(findloc, [("%"+tag+"%")])
result = myconnect.fetchall()
                        Print (result);
                        Return;
```

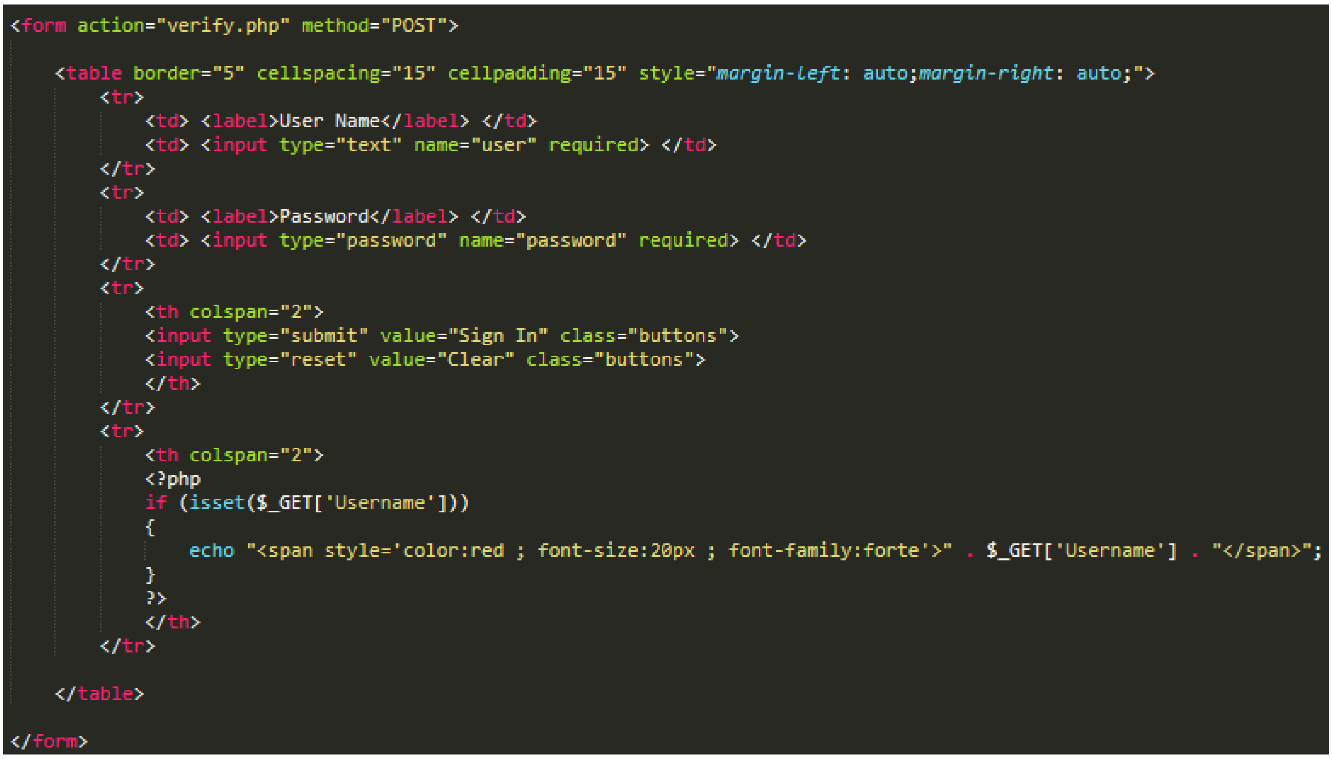
3.5 Testing Process

a) For verification of Input- Different types of inputs, such as alphabets in phone numbers, not using '@' in mail addresses, etc. were used in order to prevent unwanted inputs by users.

b) For database verification- Insertion of NULL values, duplicate values (for primary key), etc. were inputted to achieve desired stored results in tables.

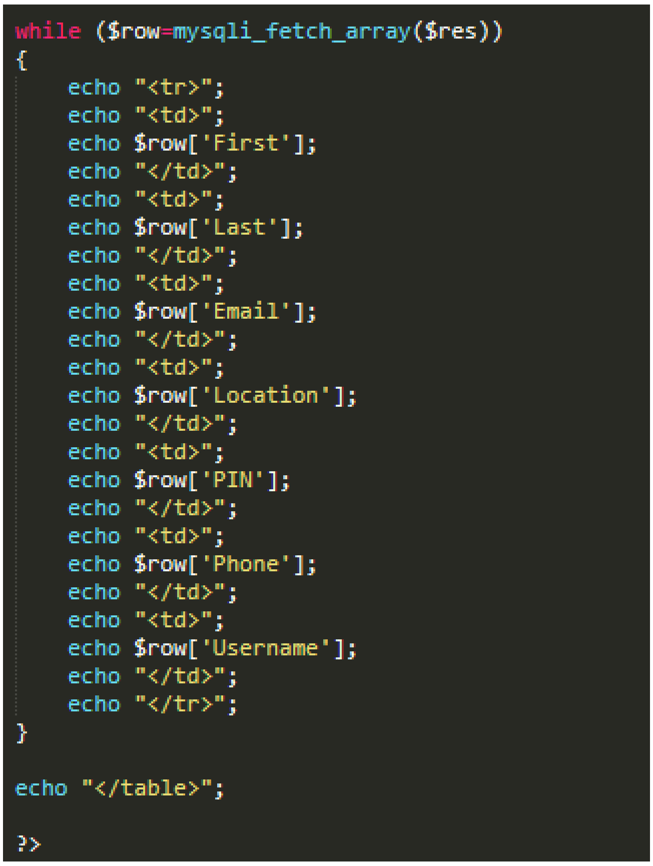
c) SQL queries: To check whether correct MySQL queries have been used, output statements were inserted in various parts of code to check out whether each query gets the desired result.

Testing for CMS Panel



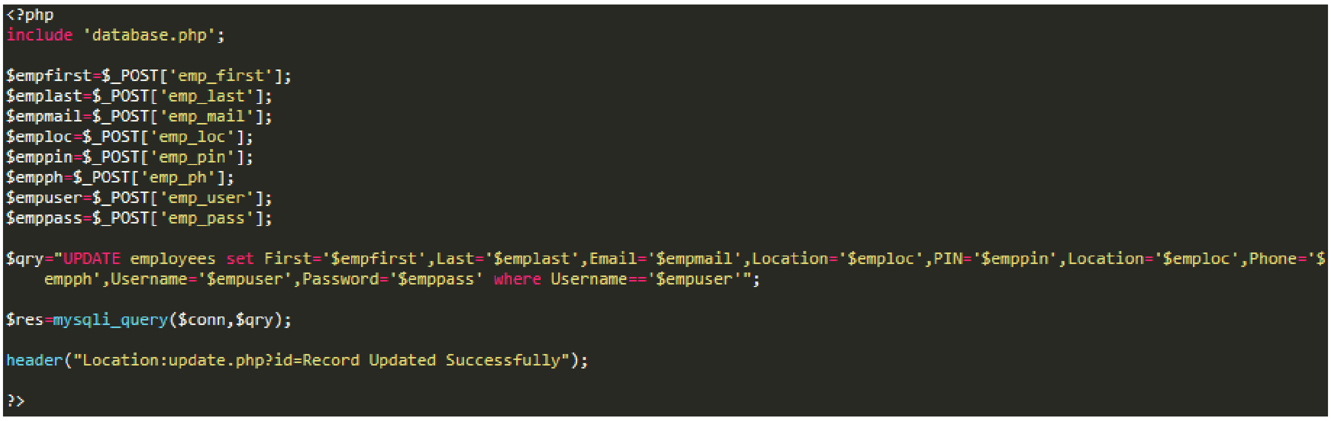
```
<form action="verify.php" method="POST">

    <table border="5" cellspacing="15" cellpadding="15" style="margin-left: auto;margin-right: auto;">
        <tr>
            <td> <label>User Name</label> </td>
            <td> <input type="text" name="user" required> </td>
        </tr>
        <tr>
            <td> <label>Password</label> </td>
            <td> <input type="password" name="password" required> </td>
        </tr>
        <tr>
            <th colspan="2">
            <input type="submit" value="Sign In" class="buttons">
            <input type="reset" value="Clear" class="buttons">
            </th>
        </tr>
        <tr>
            <th colspan="2">
            <?php
            if (isset($_GET['Username']))
            {
                echo "<span style='color:red ; font-size:20px ; font-family:forte'>" . $_GET['Username'] . "</span>";
            }
            ?>
            </th>
        </tr>

    </table>

</form>
```

Log-in form was tested upon various inputs and echoing results



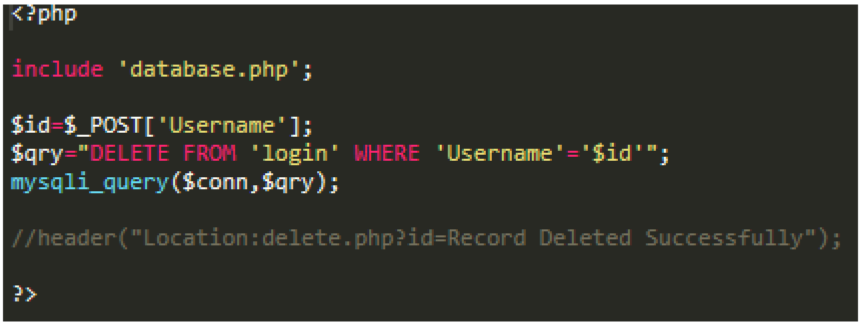
```
while ($row=mysqli_fetch_array($res))
{
    echo "<tr>";
    echo "<td>";
    echo $row['First'];
    echo "</td>";
    echo "<td>";
    echo $row['Last'];
    echo "</td>";
    echo "<td>";
    echo $row['Email'];
    echo "</td>";
    echo "<td>";
    echo $row['Location'];
    echo "</td>";
    echo "<td>";
    echo $row['PIN'];
    echo "</td>";
    echo "<td>";
    echo $row['Phone'];
    echo "</td>";
    echo "<td>";
    echo $row['Username'];
    echo "</td>";
    echo "</tr>";
}

echo "</table>";

?>
```

View records were tested by matching values with table contents of database



```
<?php
include 'database.php';

$empfirst=$_POST['emp_first'];
$emplast=$_POST['emp_last'];
$empmail=$_POST['emp_mail'];
$emploc=$_POST['emp_loc'];
$emppin=$_POST['emp_pin'];
$empph=$_POST['emp_ph'];
$empuser=$_POST['emp_user'];
$emppass=$_POST['emp_pass'];

$qry="UPDATE employees set First='$empfirst',Last='$emplast',Email='$empmail',Location='$emploc',PIN='$emppin',Location='$emploc',Phone='$
    empph',Username='$empuser',Password='$emppass' where Username=='$empuser'";

$res=mysqli_query($conn,$qry);

header("Location:update.php?id=Record Updated Successfully");

?>
```

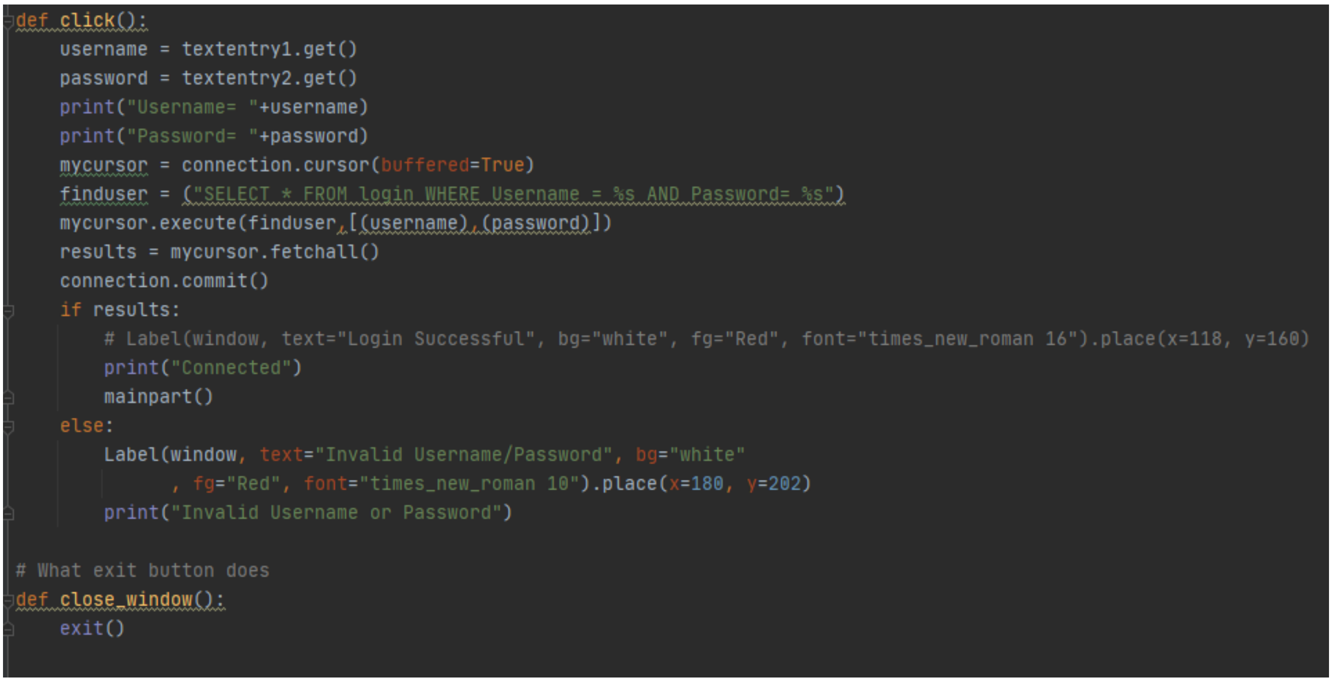
Update records were tested by matching updated record with the database



```
<?php

include 'database.php';

$id=$_POST['Username'];
$qry="DELETE FROM 'login' WHERE 'Username'='$id'";
mysqli_query($conn,$qry);

//header("Location:delete.php?id=Record Deleted Successfully");

?>
```

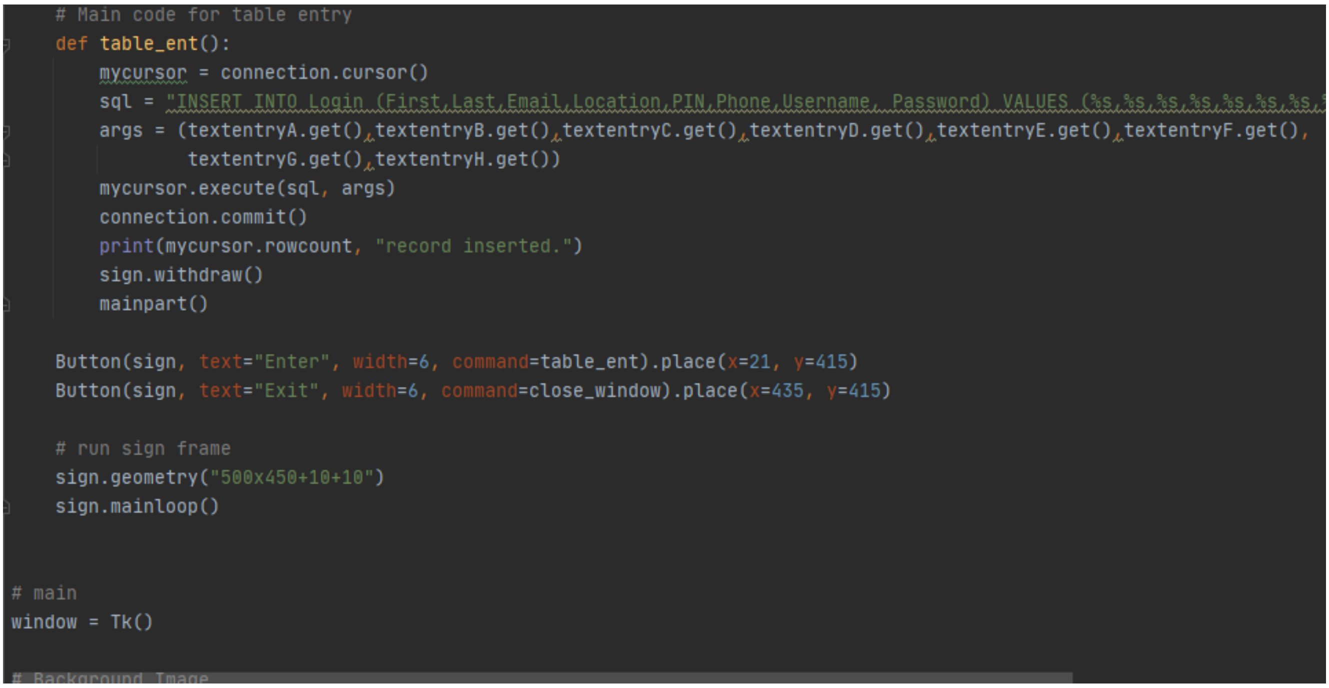
Deleterecords were tested by matching updated record with the database

Testing for App



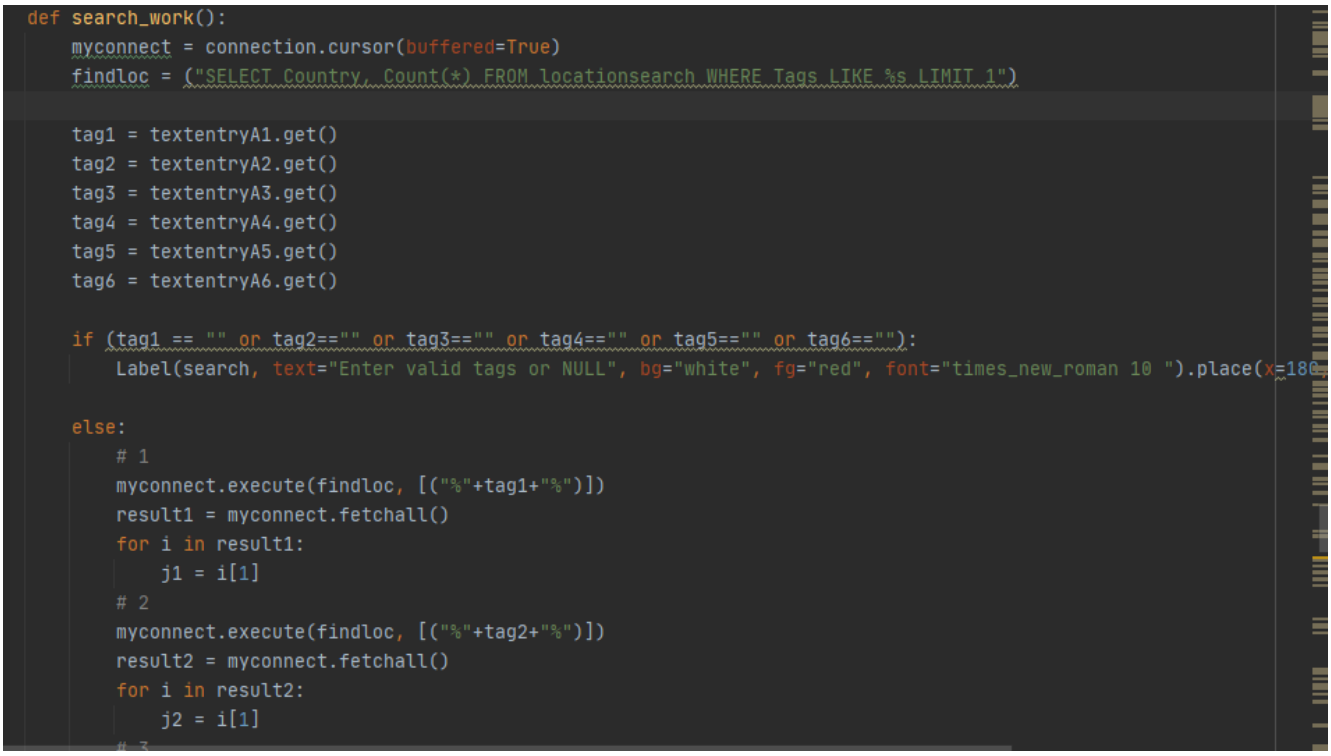
```
def click():
    username = textentry1.get()
    password = textentry2.get()
    print("Username= "+username)
    print("Password= "+password)
    mycursor = connection.cursor(buffered=True)
    finduser = ("SELECT * FROM login WHERE Username = %s AND Password= %s")
    mycursor.execute(finduser,[(username),(password)])
    results = mycursor.fetchall()
    connection.commit()
    if results:
        # Label(window, text="Login Successful", bg="white", fg="Red", font="times_new_roman 16").place(x=118, y=160)
        print("Connected")
        mainpart()
    else:
        Label(window, text="Invalid Username/Password", bg="white"
              , fg="Red", font="times_new_roman 10").place(x=180, y=202)
        print("Invalid Username or Password")

# What exit button does
def close_window():
    exit()
```

Login was tested using wrong credentials, uppercase/lowercase credentials and printing results



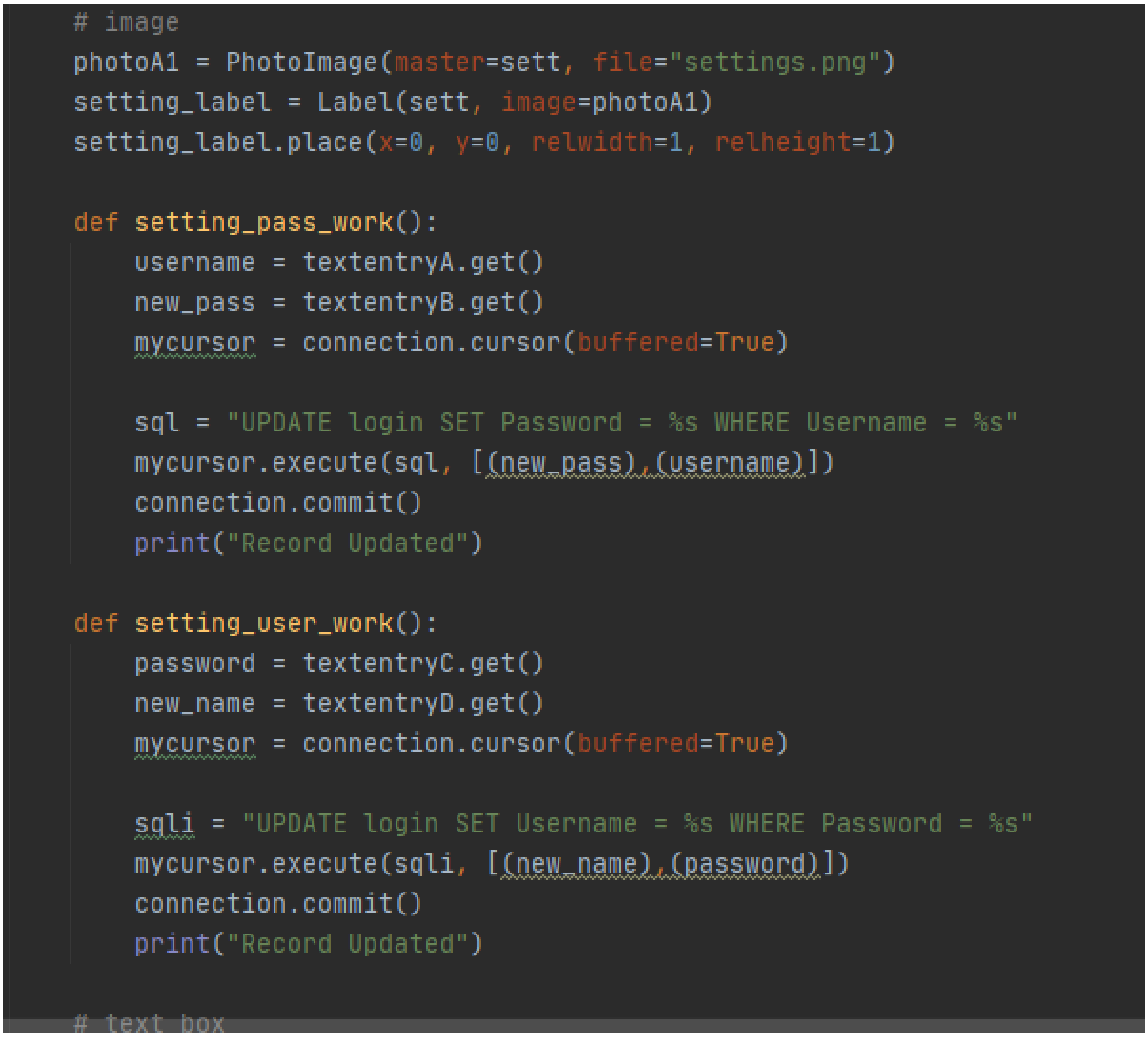
```
    # Main code for table entry
    def table_ent():
        mycursor = connection.cursor()
        sql = "INSERT INTO Login (First,Last,Email,Location,PIN,Phone,Username, Password) VALUES (%s,%s,%s,%s,%s,%s,%s,
        args = (textentryA.get(),textentryB.get(),textentryC.get(),textentryD.get(),textentryE.get(),textentryF.get(),
                textentryG.get(),textentryH.get())
        mycursor.execute(sql, args)
        connection.commit()
        print(mycursor.rowcount, "record inserted.")
        sign.withdraw()
        mainpart()

    Button(sign, text="Enter", width=6, command=table_ent).place(x=21, y=415)
    Button(sign, text="Exit", width=6, command=close_window).place(x=435, y=415)

    # run sign frame
    sign.geometry("500x450+10+10")
    sign.mainloop()


# main
window = Tk()

# Background Image
```

Form credentials and actions were tested under various inputs.



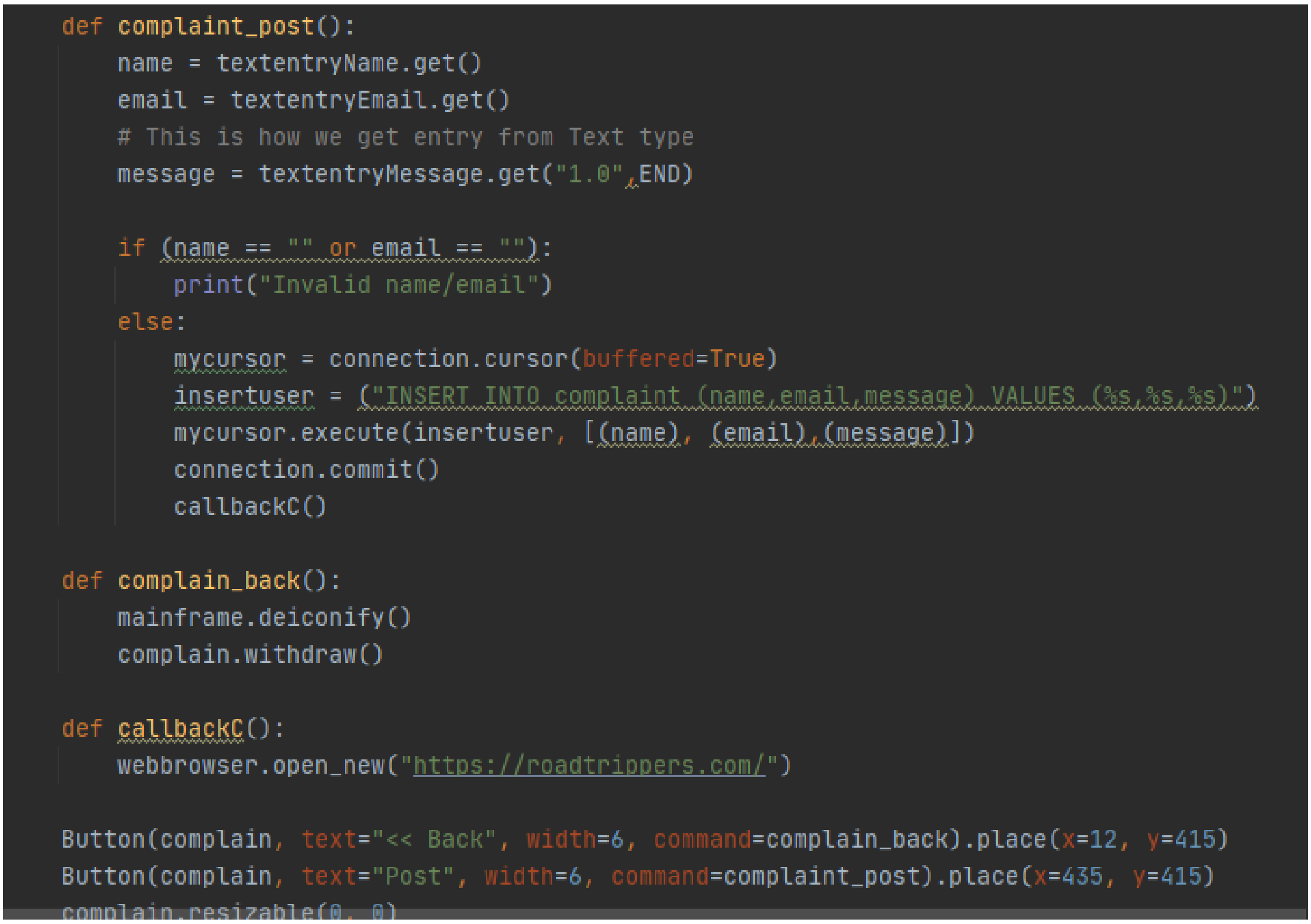
```
def search_work():
    myconnect = connection.cursor(buffered=True)
    findloc = ("SELECT Country, Count(*) FROM locationsearch WHERE Tags LIKE %s LIMIT 1")

    tag1 = textentryA1.get()
    tag2 = textentryA2.get()
    tag3 = textentryA3.get()
    tag4 = textentryA4.get()
    tag5 = textentryA5.get()
    tag6 = textentryA6.get()

    if (tag1 == "" or tag2=="" or tag3=="" or tag4=="" or tag5=="" or tag6==""):
        Label(search, text="Enter valid tags or NULL", bg="white", fg="red", font="times_new_roman 10 ").place(x=180,

    else:
        # 1
        myconnect.execute(findloc, [("%"+tag1+"%")])
        result1 = myconnect.fetchall()
        for i in result1:
            j1 = i[1]
        # 2
        myconnect.execute(findloc, [("%"+tag2+"%")])
        result2 = myconnect.fetchall()
        for i in result2:
            j2 = i[1]
        # 3
```

Tag search was tested by using equal matching tags of 2 locations, printing results of each match, etc.



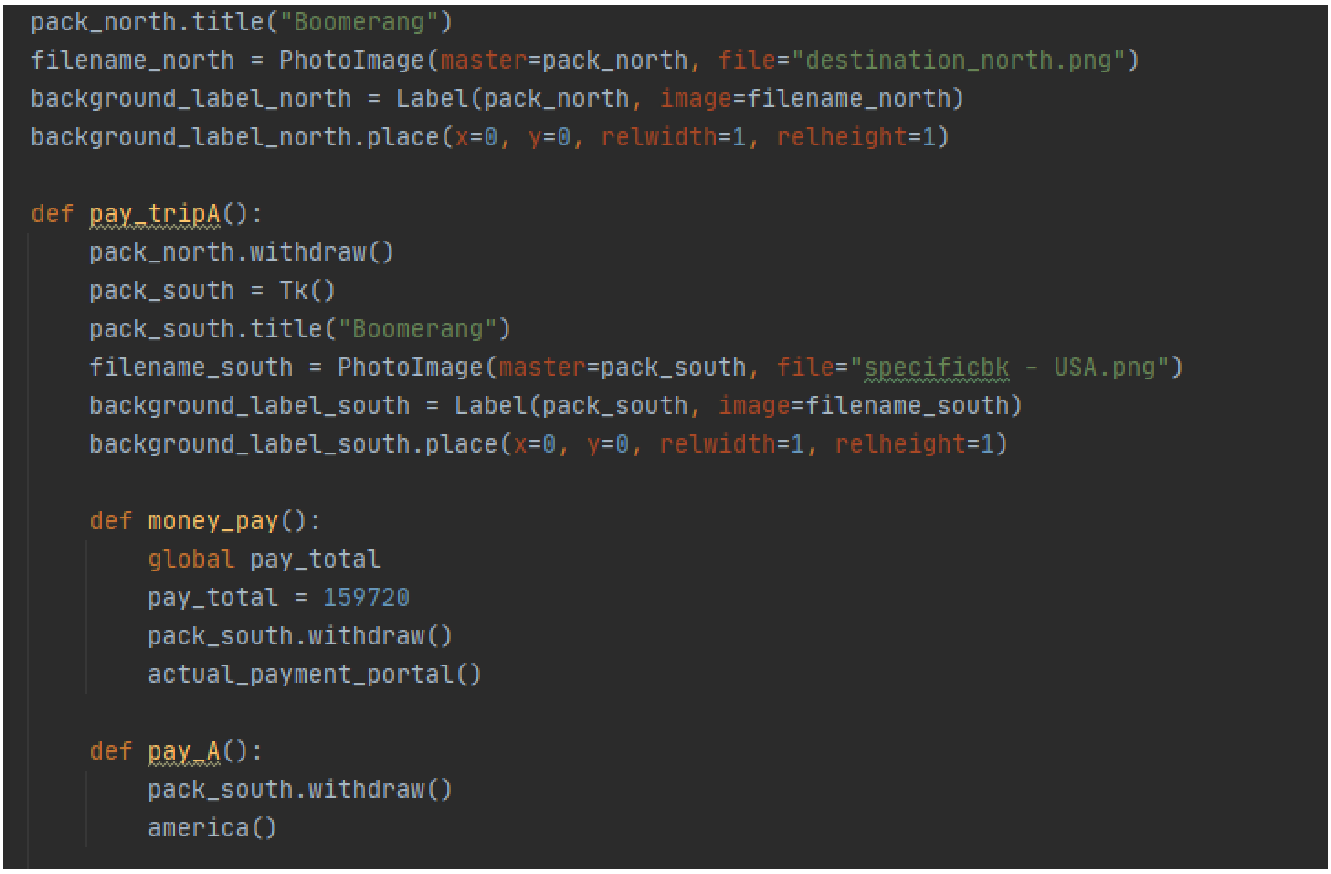
```python
# image
photoA1 = PhotoImage(master=sett, file="settings.png")
setting_label = Label(sett, image=photoA1)
setting_label.place(x=0, y=0, relwidth=1, relheight=1)

def setting_pass_work():
    username = textentryA.get()
    new_pass = textentryB.get()
    mycursor = connection.cursor(buffered=True)

    sql = "UPDATE login SET Password = %s WHERE Username = %s"
    mycursor.execute(sql, [(new_pass),(username)])
    connection.commit()
    print("Record Updated")

def setting_user_work():
    password = textentryC.get()
    new_name = textentryD.get()
    mycursor = connection.cursor(buffered=True)

    sqli = "UPDATE login SET Username = %s WHERE Password = %s"
    mycursor.execute(sqli, [(new_name),(password)])
    connection.commit()
    print("Record Updated")

# text box
```

Settings was testing by inputting various types of inputs and using print statements



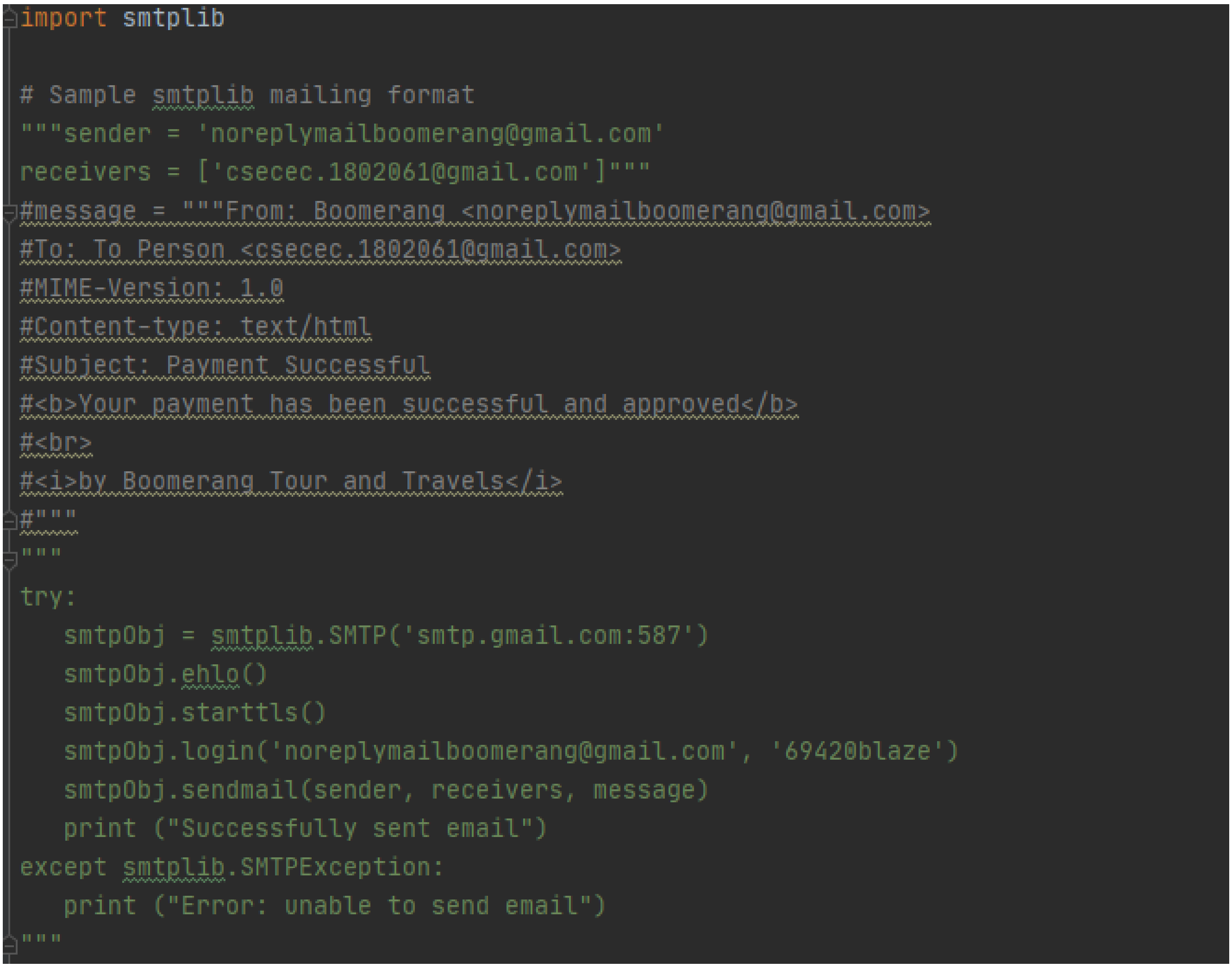
```
def complaint_post():
    name = textentryName.get()
    email = textentryEmail.get()
    # This is how we get entry from Text type
    message = textentryMessage.get("1.0",END)

    if (name == "" or email == ""):
        print("Invalid name/email")
    else:
        mycursor = connection.cursor(buffered=True)
        insertuser = ("INSERT INTO complaint (name,email,message) VALUES (%s,%s,%s)")
        mycursor.execute(insertuser, [(name), (email),(message)])
        connection.commit()
        callbackC()

def complain_back():
    mainframe.deiconify()
    complain.withdraw()

def callbackC():
    webbrowser.open_new("https://roadtrippers.com/")

Button(complain, text="<< Back", width=6, command=complain_back).place(x=12, y=415)
Button(complain, text="Post", width=6, command=complaint_post).place(x=435, y=415)
complain.resizable(0, 0)
```

Complaints was tested by confirmed entered messages by checking into table in database



```
pack_north.title("Boomerang")
filename_north = PhotoImage(master=pack_north, file="destination_north.png")
background_label_north = Label(pack_north, image=filename_north)
background_label_north.place(x=0, y=0, relwidth=1, relheight=1)

def pay_tripA():
    pack_north.withdraw()
    pack_south = Tk()
    pack_south.title("Boomerang")
    filename_south = PhotoImage(master=pack_south, file="specificbk - USA.png")
    background_label_south = Label(pack_south, image=filename_south)
    background_label_south.place(x=0, y=0, relwidth=1, relheight=1)

    def money_pay():
        global pay_total
        pay_total = 159720
        pack_south.withdraw()
        actual_payment_portal()

    def pay_A():
        pack_south.withdraw()
        america()
```

Package payments were tested by matching corresponding values to package actual cost, hotel costs, etc.

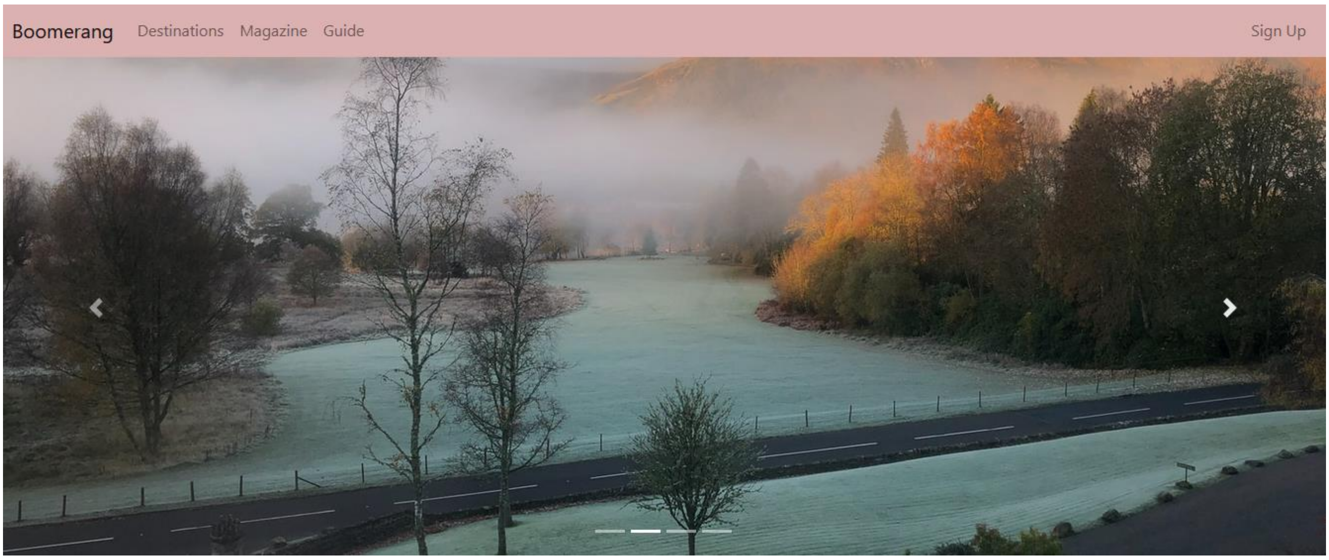


```
import smtplib

# Sample smtplib mailing format
"""sender = 'noreplymailboomerang@gmail.com'
receivers = ['csecec.1802061@gmail.com']"""
#message = """From: Boomerang <noreplymailboomerang@gmail.com>
#To: To Person <csecec.1802061@gmail.com>
#MIME-Version: 1.0
#Content-type: text/html
#Subject: Payment Successful
#<b>Your payment has been successful and approved</b>
#<br>
#<i>by Boomerang Tour and Travels</i>
#"""
"""
try:
   smtpObj = smtplib.SMTP('smtp.gmail.com:587')
   smtpObj.ehlo()
   smtpObj.starttls()
   smtpObj.login('noreplymailboomerang@gmail.com', '69420blaze')
   smtpObj.sendmail(sender, receivers, message)
   print ("Successfully sent email")
except smtplib.SMTPException:
   print ("Error: unable to send email")
"""
```

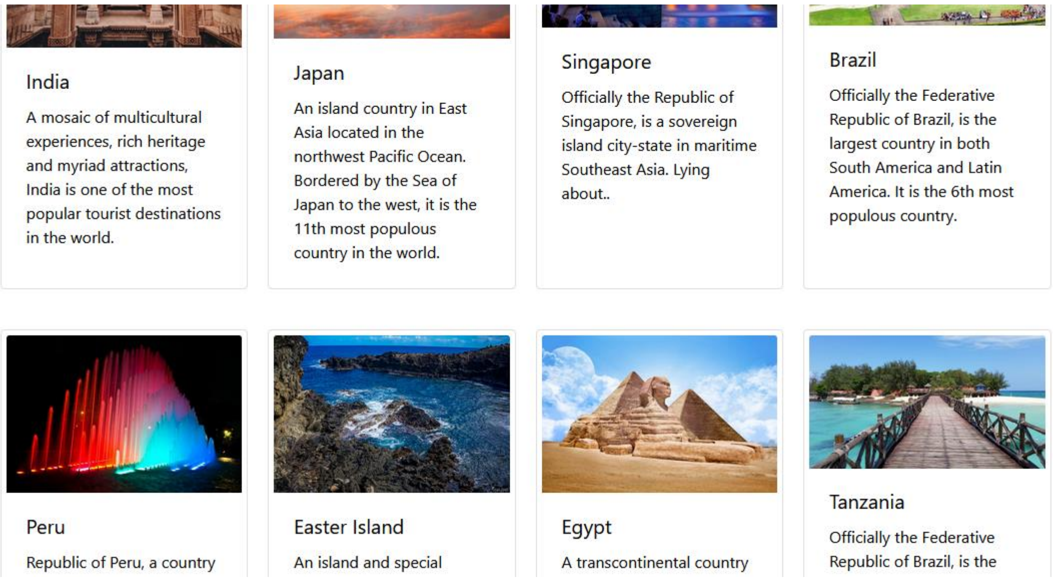
Python mailer was tested by using sample email ids and upon receiving were confirmed.

4 Results/Outputs

a) Website



Homepage



India

A mosaic of multicultural experiences, rich heritage and myriad attractions, India is one of the most popular tourist destinations in the world.

Japan

An island country in East Asia located in the northwest Pacific Ocean. Bordered by the Sea of Japan to the west, it is the 11th most populous country in the world.

Singapore

Officially the Republic of Singapore, is a sovereign island city-state in maritime Southeast Asia. Lying about..

Brazil

Officially the Federative Republic of Brazil, is the largest country in both South America and Latin America. It is the 6th most populous country.

Peru

Republic of Peru, a country

Easter Island

An island and special

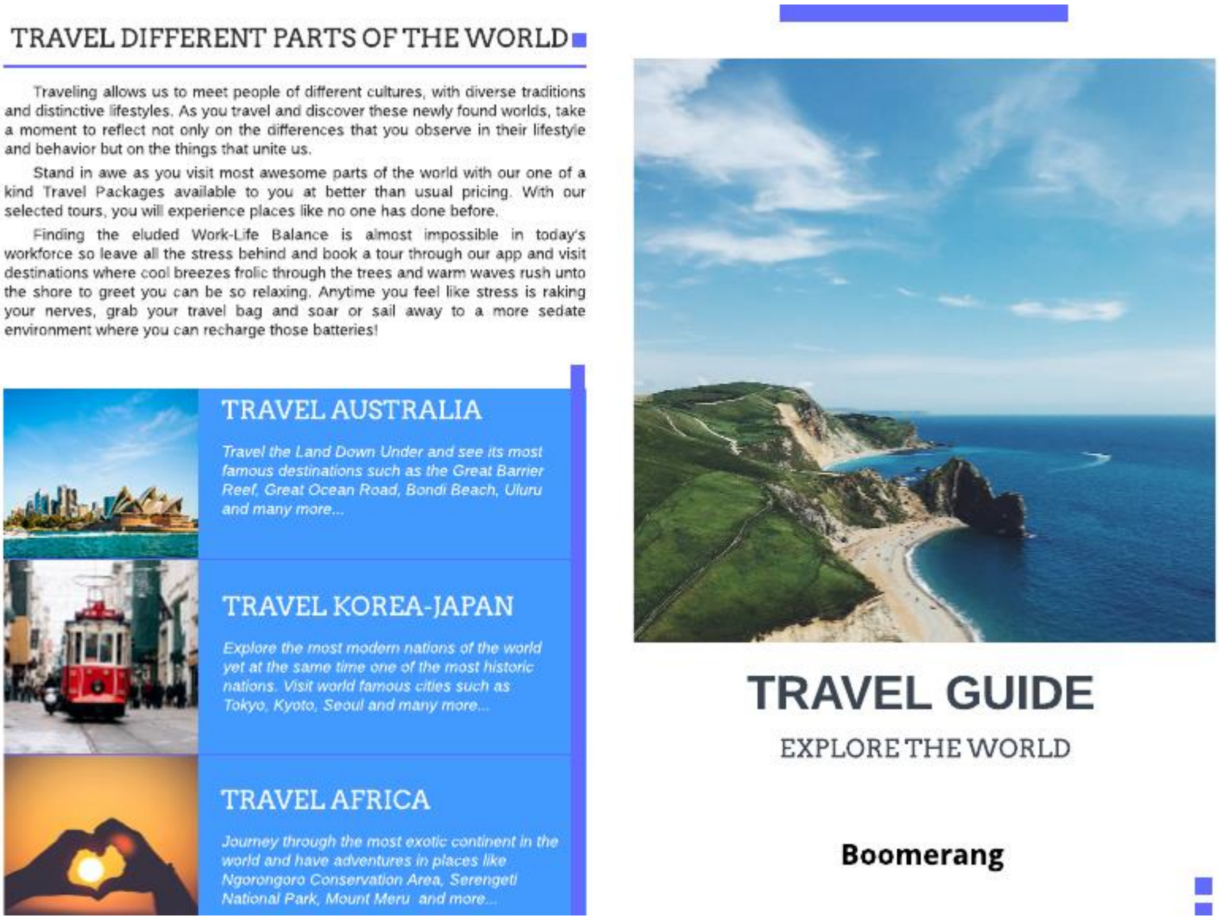
Egypt

A transcontinental country

Tanzania

Officially the Federative Republic of Brazil, is the

Travel Magazine



TRAVEL DIFFERENT PARTS OF THE WORLD

Traveling allows us to meet people of different cultures, with diverse traditions and distinctive lifestyles. As you travel and discover these newly found worlds, take a moment to reflect not only on the differences that you observe in their lifestyle and behavior but on the things that unite us.

Stand in awe as you visit most awesome parts of the world with our one of a kind Travel Packages available to you at better than usual pricing. With our selected tours, you will experience places like no one has done before.

Finding the eluded Work-Life Balance is almost impossible in today's workforce so leave all the stress behind and book a tour through our app and visit destinations where cool breezes frolic through the trees and warm waves rush unto the shore to greet you can be so relaxing. Anytime you feel like stress is raking your nerves, grab your travel bag and soar or sail away to a more sedate environment where you can recharge those batteries!

TRAVEL AUSTRALIA

Travel the Land Down Under and see its most famous destinations such as the Great Barrier Reef, Great Ocean Road, Bondi Beach, Uluru and many more...

TRAVEL KOREA-JAPAN

Explore the most modern nations of the world yet at the same time one of the most historic nations. Visit world famous cities such as Tokyo, Kyoto, Seoul and many more...

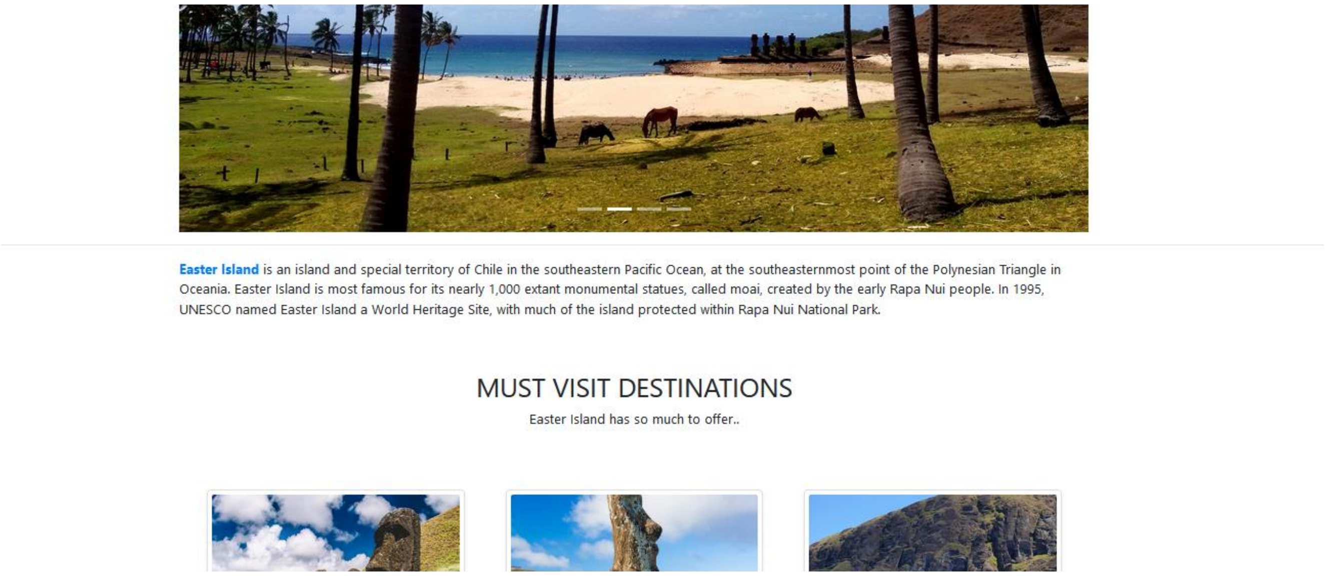
TRAVEL AFRICA

Journey through the most exotic continent in the world and have adventures in places like Ngorongoro Conservation Area, Serengeti National Park, Mount Meru and more...

TRAVEL GUIDE

EXPLORE THE WORLD

Boomerang

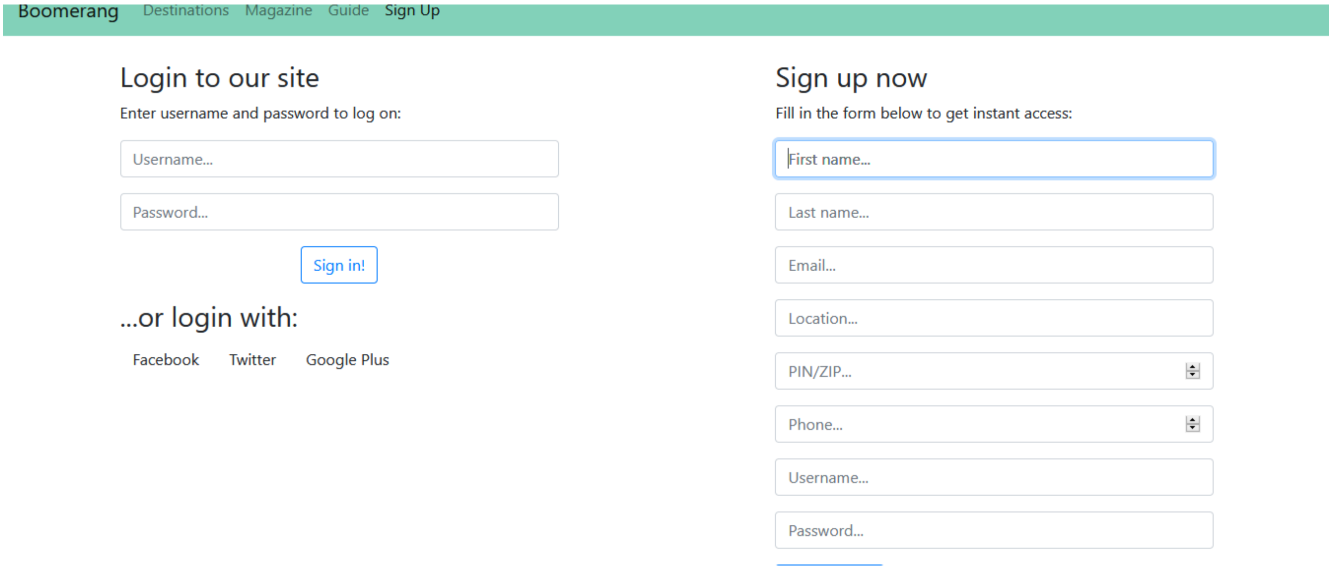


Easter Island is an island and special territory of Chile in the southeastern Pacific Ocean, at the southeasternmost point of the Polynesian Triangle in Oceania. Easter Island is most famous for its nearly 1,000 extant monumental statues, called moai, created by the early Rapa Nui people. In 1995, UNESCO named Easter Island a World Heritage Site, with much of the island protected within Rapa Nui National Park.

MUST VISIT DESTINATIONS

Easter Island has so much to offer..

Travel Guide page



Login to our site

Enter username and password to log on:

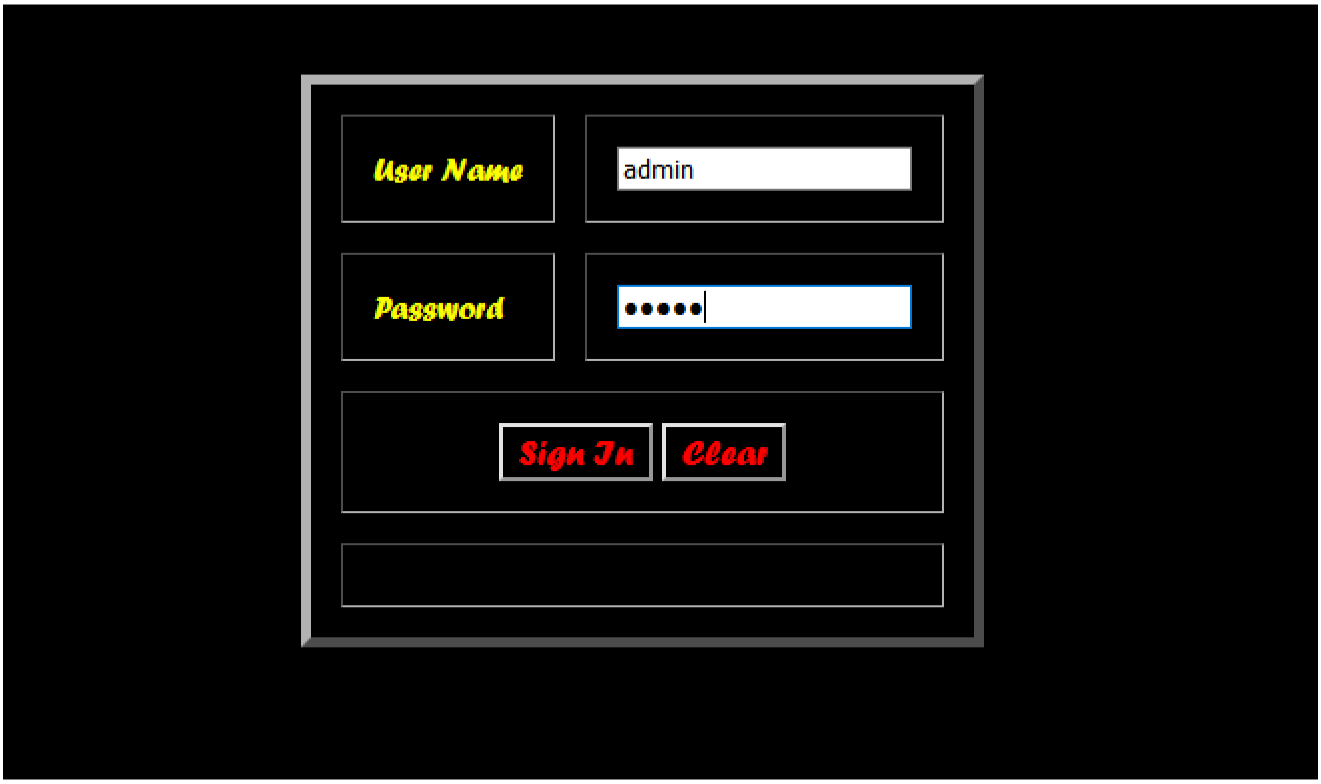
...or login with:

Sign up now

Fill in the form below to get instant access:

Sign-up/Login page

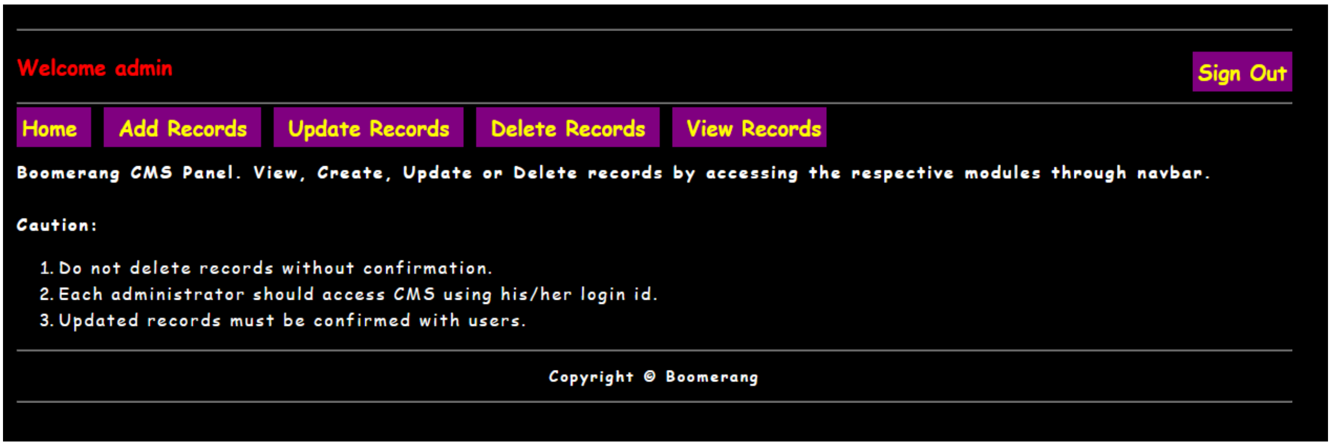
b) CMS Panel



User Name: admin

Password: •••••

Login- initial screen



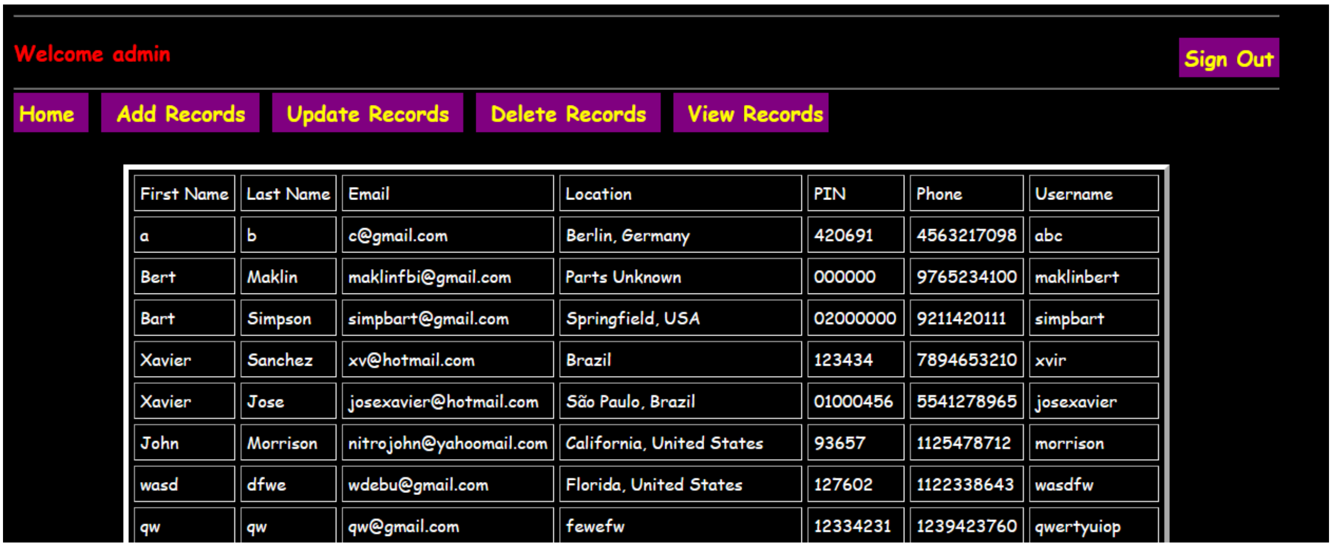
Boomerang CMS Panel. View, Create, Update or Delete records by accessing the respective modules through navbar.

Caution:

1. Do not delete records without confirmation.
2. Each administrator should access CMS using his/her login id.
3. Updated records must be confirmed with users.

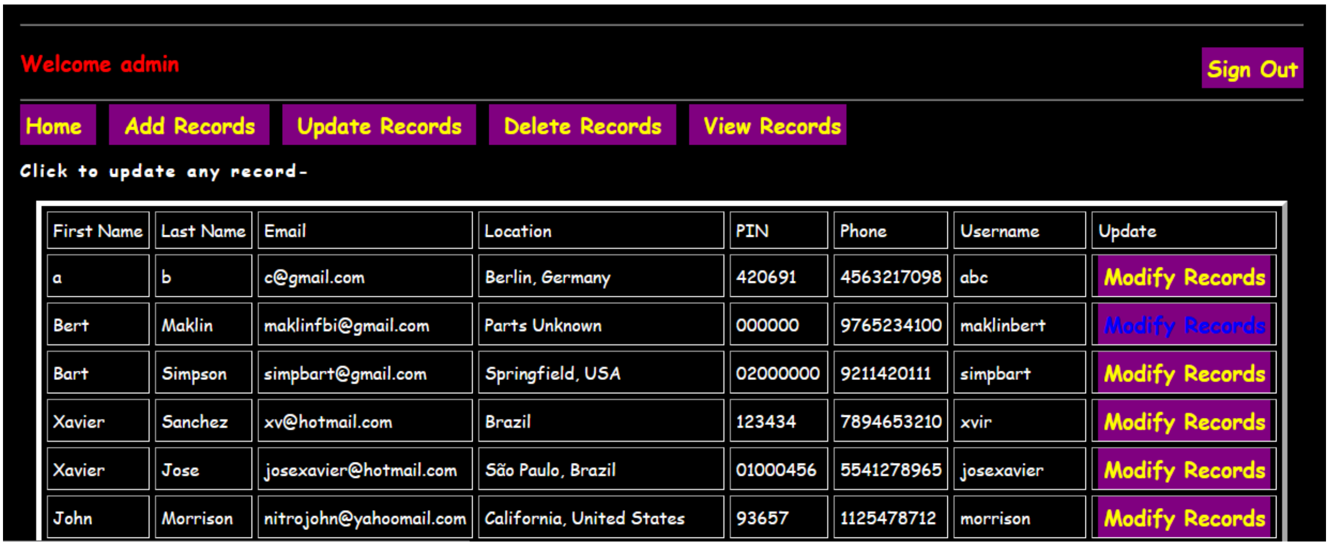
Copyright © Boomerang

Main Menu



First Name	Last Name	Email	Location	PIN	Phone	Username
a	b	c@gmail.com	Berlin, Germany	420691	4563217098	abc
Bert	Maklin	maklinfbi@gmail.com	Parts Unknown	000000	9765234100	maklinbert
Bart	Simpson	simpbart@gmail.com	Springfield, USA	02000000	9211420111	simpbart
Xavier	Sanchez	xv@hotmail.com	Brazil	123434	7894653210	xvir
Xavier	Jose	josexavier@hotmail.com	São Paulo, Brazil	01000456	5541278965	josexavier
John	Morrison	nitrojohn@yahoomail.com	California, United States	93657	1125478712	morrison
wasd	dfwe	wdebu@gmail.com	Florida, United States	127602	1122338643	wasdfw
qw	qw	qw@gmail.com	fewefw	12334231	1239423760	qwertyuiop

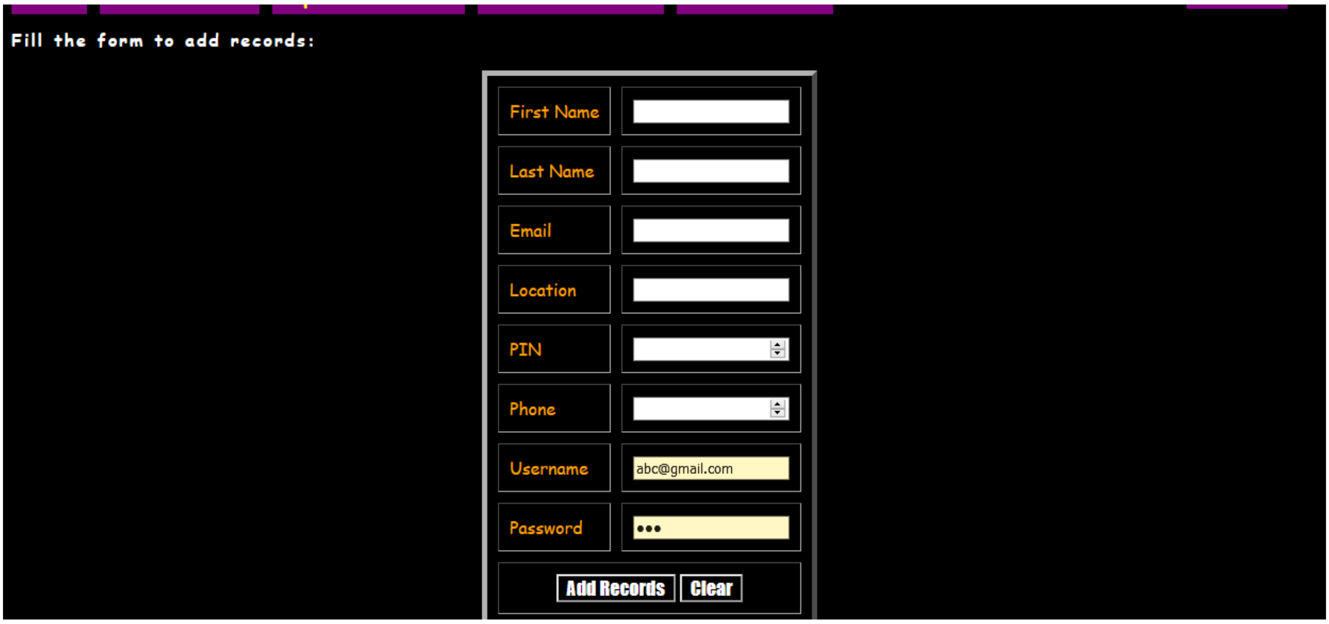
View Records



Click to update any record-

First Name	Last Name	Email	Location	PIN	Phone	Username	Update
a	b	c@gmail.com	Berlin, Germany	420691	4563217098	abc	Modify Records
Bert	Maklin	maklinfbi@gmail.com	Parts Unknown	000000	9765234100	maklinbert	Modify Records
Bart	Simpson	simpbart@gmail.com	Springfield, USA	02000000	9211420111	simpbart	Modify Records
Xavier	Sanchez	xv@hotmail.com	Brazil	123434	7894653210	xvir	Modify Records
Xavier	Jose	josexavier@hotmail.com	São Paulo, Brazil	01000456	5541278965	josexavier	Modify Records
John	Morrison	nitrojohn@yahoomail.com	California, United States	93657	1125478712	morrison	Modify Records

Update Records

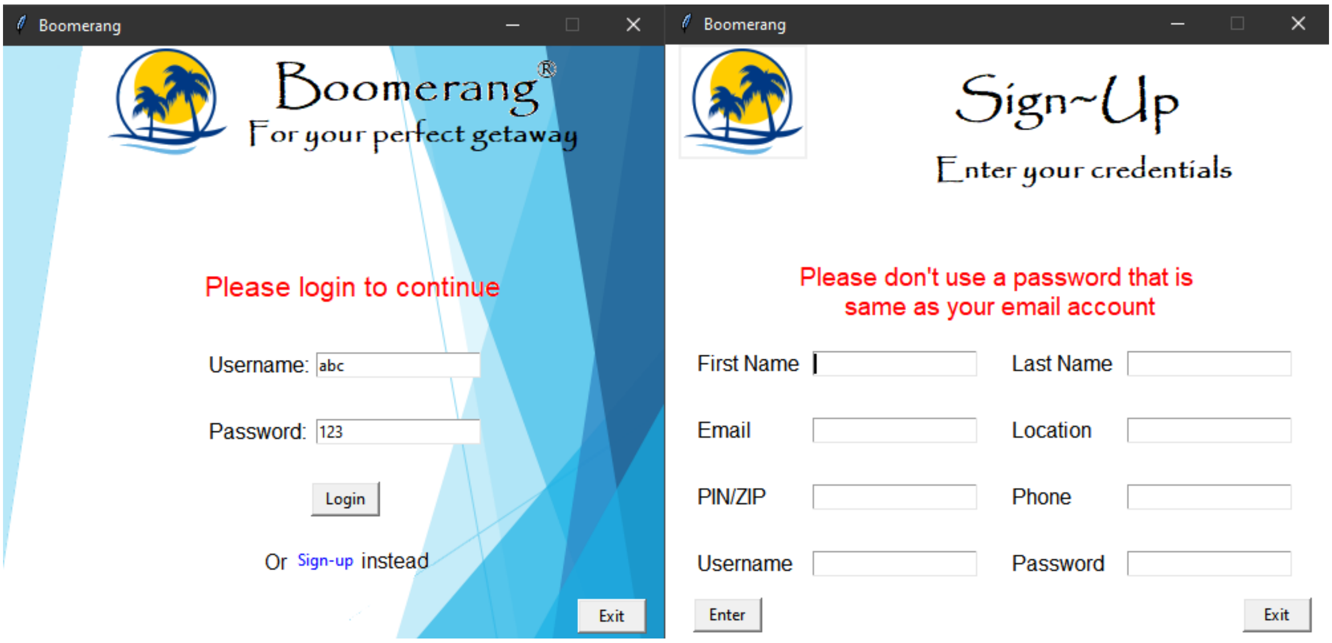


Fill the form to add records:

First Name	
Last Name	
Email	
Location	
PIN	
Phone	
Username	abc@gmail.com
Password	•••

Delete Records

c) Python App



Please login to continue

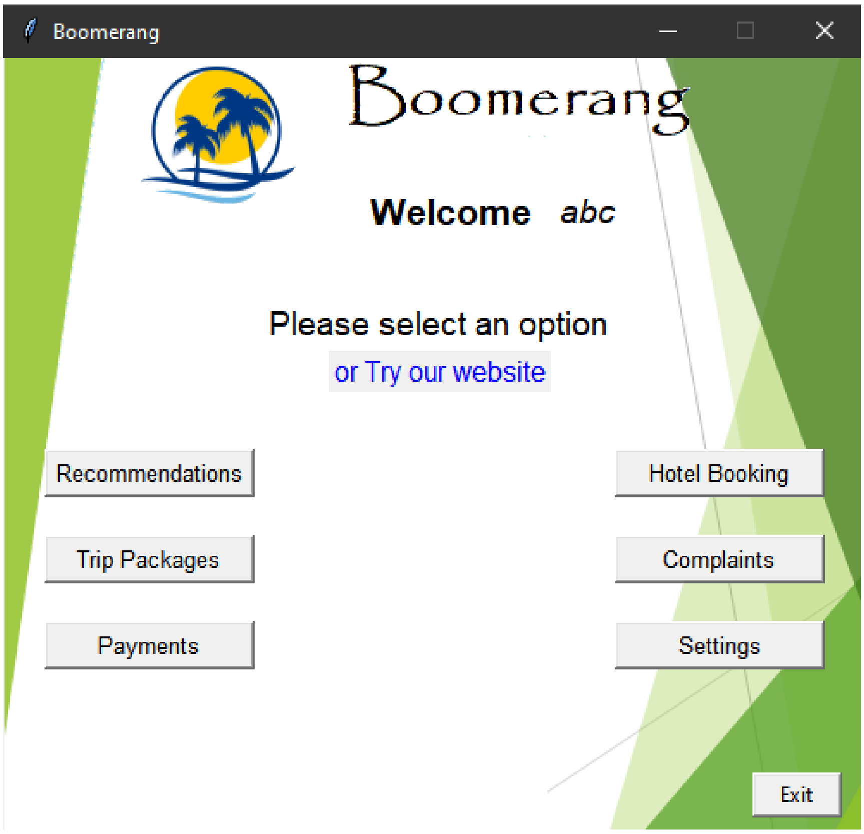
Username: abc

Password: 123

Please don't use a password that is same as your email account

First Name		Last Name	
Email		Location	
PIN/ZIP		Phone	
Username		Password	

Initial Login/Sign-up screen prompt

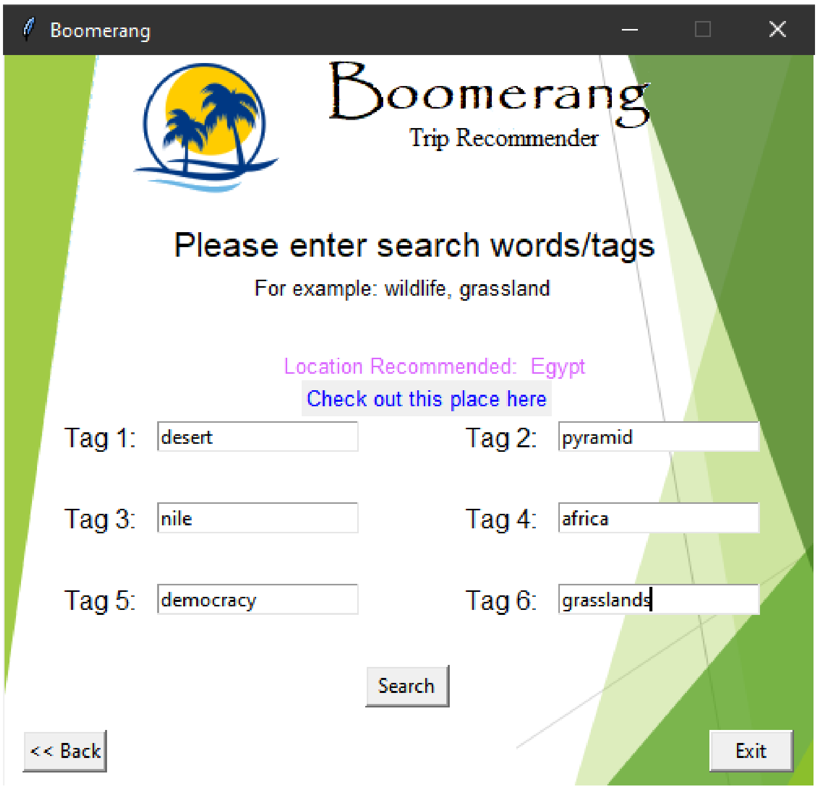


Welcome abc

Please select an option

or Try our website

Main Menu



Please enter search words/tags

For example: wildlife, grassland

Location Recommended: Egypt

Check out this place here

Tag 1: desert

Tag 2: pyramid

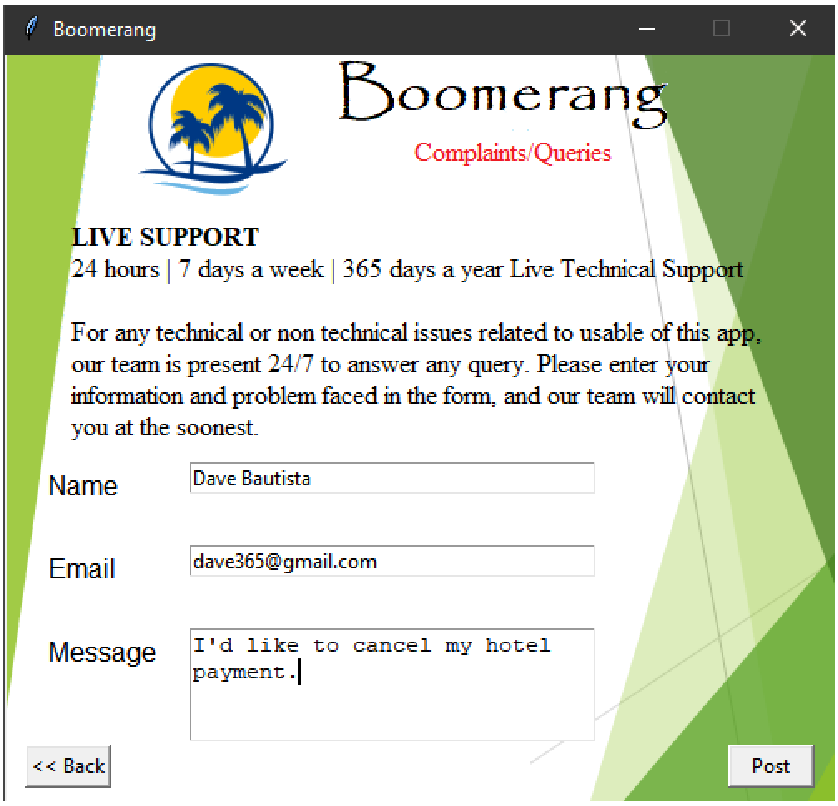
Tag 3: nile

Tag 4: africa

Tag 5: democracy

Tag 6: grasslands

Tag Search



Complaints/Queries

LIVE SUPPORT

24 hours | 7 days a week | 365 days a year Live Technical Support

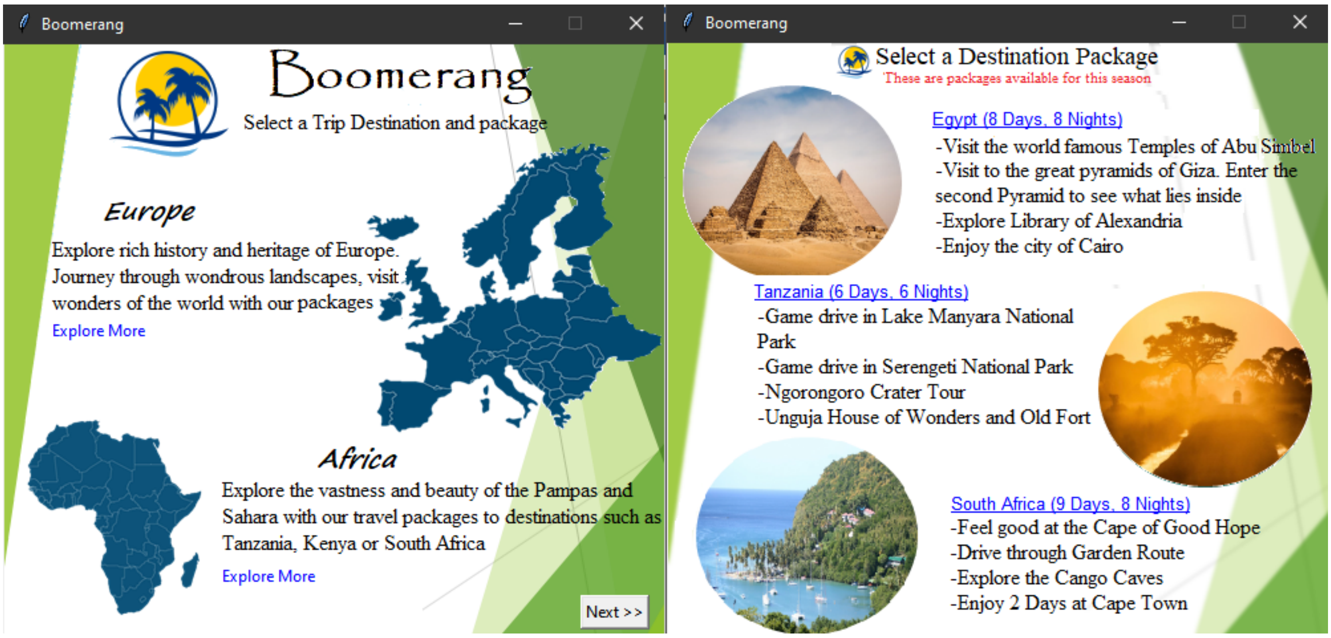
For any technical or non technical issues related to usable of this app, our team is present 24/7 to answer any query. Please enter your information and problem faced in the form, and our team will contact you at the soonest.

Name: Dave Bautista

Email: dave365@gmail.com

Message: I'd like to cancel my hotel payment.

Complains Module



Select a Trip Destination and package

Europe

Explore rich history and heritage of Europe. Journey through wondrous landscapes, visit wonders of the world with our packages

Africa

Explore the vastness and beauty of the Pampas and Sahara with our travel packages to destinations such as Tanzania, Kenya or South Africa

Select a Destination Package

These are packages available for this season

Egypt (8 Days, 8 Nights)

-Visit the world famous Temples of Abu Simbel
-Visit to the great pyramids of Giza. Enter the second Pyramid to see what lies inside
-Explore Library of Alexandria
-Enjoy the city of Cairo

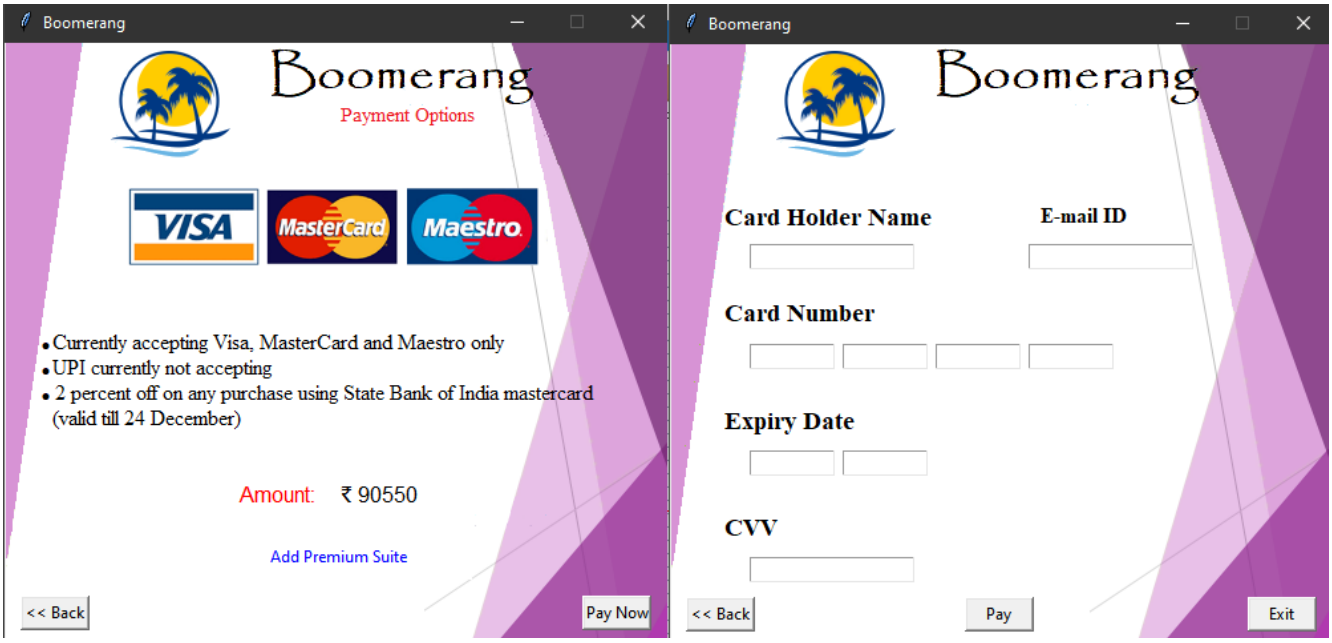
Tanzania (6 Days, 6 Nights)

-Game drive in Lake Manyara National Park
-Game drive in Serengeti National Park
-Ngorongoro Crater Tour
-Unguja House of Wonders and Old Fort

South Africa (9 Days, 8 Nights)

-Feel good at the Cape of Good Hope
-Drive through Garden Route
-Explore the Cango Caves
-Enjoy 2 Days at Cape Town

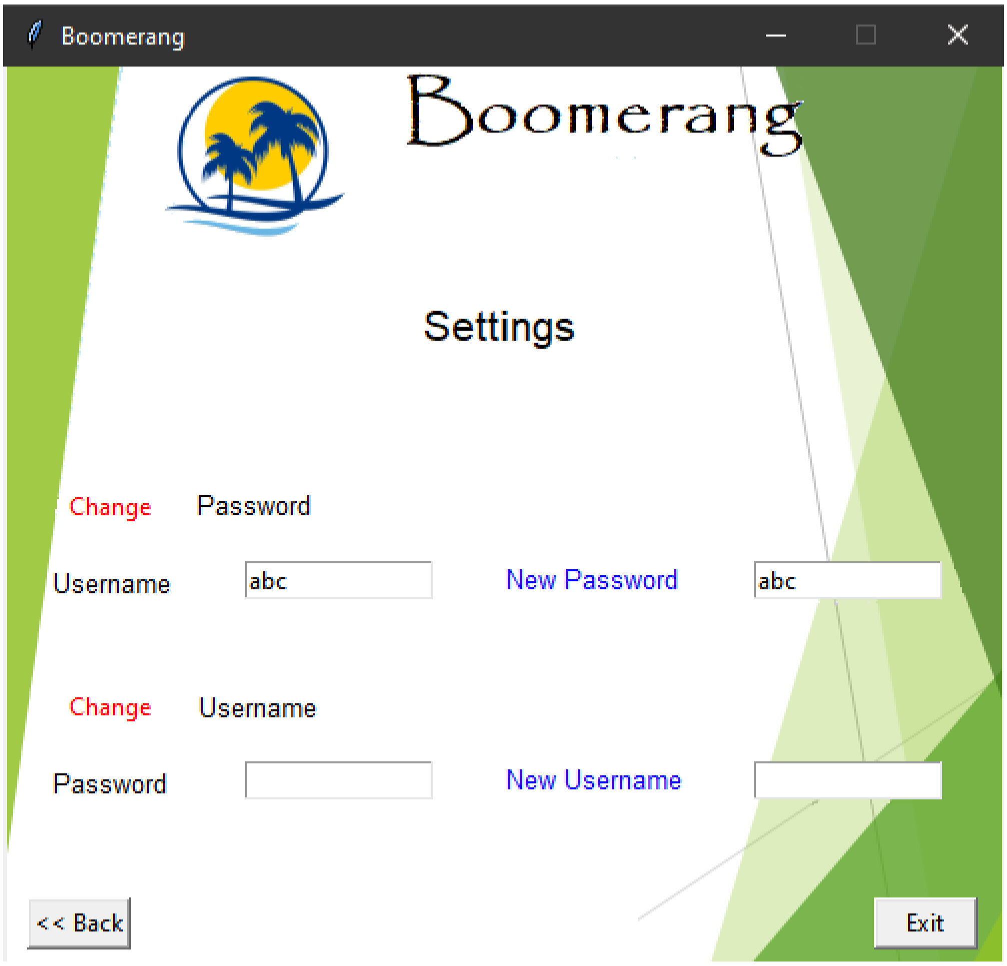
Trip Package Module



- Currently accepting Visa, MasterCard and Maestro only
- UPI currently not accepting
- 2 percent off on any purchase using State Bank of India mastercard (valid till 24 December)

Amount: ₹ 90550

Payments Module



Username abc

New Password abc

Password

New Username

Settings Module

5. Conclusions / Recommendations

E commerce is a rising field which is replacing traditional shopping methods at a very fast pace. Advancement in a field often leads to exploitation unless security is provided and certain standards/rules are proposed. Moreover, purchasing trips online can be risky as there can be cases of cheating, frauds or tours being not up to standards.

Popular Trip planning websites has reached a point where market shares rise and ebb on the basis of discounting. However, the discounting doesn't guarantee customer loyalty. Moreover, due to restrictions of partnerships, there is less freedom of trade.

By having simpler, easy to manage system, it not only decreases maintenance costs but also leaves positive impact on users. By having independent system that rather allows users to choose hotel rooms/ and have a recommended tour place works better and guarantees a loyal customer base.

Boomerang uses following technologies, which allows it to achieve faster loading speed and ease of use. These technologies are as follows:

1) HTML: It provides a mean to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.
2) CSS: It enables the separation of presentation and content, including layout, colors, and fonts.
3) Bootstrap: Bootstrap framework allows the website to be responsive, meaning it can be viewed easily in mobile phones as well as desktop computers, which allows users to access the site from any platform.
4) PHP: Use of PHP allows us to make a website dynamic. It allows us to process forms and run programs on server side.
5) MySQL: It provides an open-sourcerelational database management system.
6) Python: It provides the capability to write complex programs in very simple manner.

Use of these technologies are recommended as they serve many advantages over the others. Thus, by having a simpler, easy to maintain system, initial success and customer satisfaction can be achieved, and the website and application itself can serve as framework for future projects.

References

I. PHP Documentation (https://www.php.net/docs.php)

II. MySQL Documentation (https://dev.mysql.com/doc/)

III. HTML/CSS Documentation (https://www.w3schools.com/)

IV. Bootstrap Documentation (https://getbootstrap.com/docs/3.3/)

V. Python Documentation (https://docs.python.org/3/)

VI. Internshala: Web Development Training (https://trainings.internshala.com/webdevelopment-training)

VII. W3 Schools (https://www.w3schools.com/)

VIII. StackOverflow (https://stackoverflow.com/)

IX. GeekforGeeks (https://www.geeksforgeeks.org/)

X. Google Images

XI. YouTube (https://www.youtube.com/)