

A Mini Project Report

PAYING GUEST MANAGEMENT SYSTEM

submitted in partial fulfillment for the requirements of

B.E. (CSE) VI Semester Mini Project

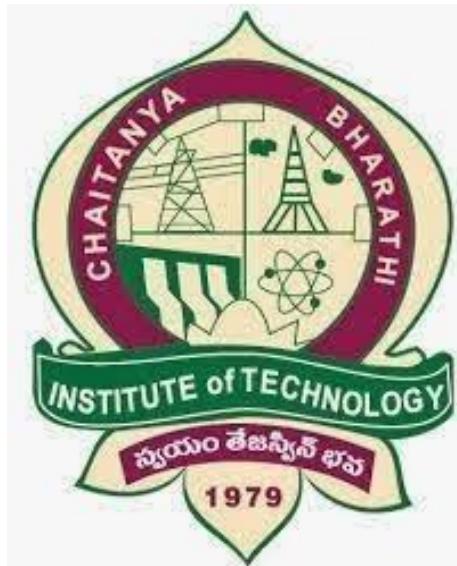
in

COMPUTER SCIENCE AND ENGINEERING

by

SRIDHAR KANDI (160117733051)

D.RAM NITHIN REDDY (160117733038)



Department of Computer Science and Engineering,
Chaitanya Bharathi Institute of Technology (Autonomous),
(Affiliated to Osmania University, Hyderabad)
Hyderabad, TELANGANA (INDIA) – 500 075
MAY-2020

CERTIFICATE

This is to certify that the project titled “**Paying Guest Management System**” is the bonafide work carried out by **Sridhar Kandi (160117733051)** and **D. Ram Nithin Reddy (160117733038)** a student of B.E.(CSE) of Chaitanya Bharathi Institute of Technology, Hyderabad, affiliated to Osmania University, Hyderabad, Telangana(India) during the academic year 2019-2020, submitted in partial fulfillment of the requirements for the **B.E.(CSE)** VI Semester Mini Projects and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Mentor(s)

E.Swathi Asst. prof

Head, CSE Dept.

Swamy Das Prof, DR

Place: Gandipet, Hyderabad

Date: 21.Apr.2020

DECLARATION

I/we hereby declare that the project entitled "Paying Guest Management System". submitted for the B. E (CSE) Mini Projects is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Sridhar Kandi

D. Ram Nithin Reddy

Name(s) and Signature(s) of the Student

Place: Gandipet, Hyderabad

Date: 21.Apr.2020

ABSTRACT

A Guest who pays for the services is a paying guest. Paying guest is a one-stop destination to assist with travel needs and to find the right accommodation as per the needs and budget i.e. bed & breakfast or Homestay, Guest House or Paying Guest. With this project, the user can easily search the paying guest accommodation near the workplace or where we want to stay for a while. Earlier it was difficult to work in a place which is far away and hence we had to lose good opportunities.

In the existing system, users can book the rooms based on his requirements but he may not be fully satisfied with the service because of the environment, food, etc. Also, the user may not know the regional language and communicate with the PG owner effectively. Users may like the PG and want to stay there, but if rooms are not available, he needs to check for another PG.

Here we give detailed information about the services and the environment such as near bus stations, metro stations, all the transport and parks, malls, etc. If the user finds some PG interesting and wants to stay there he can check on the notifications whenever there is availability there. To effectively increase the communication between the user and the PG owner we are using a google translator to translate the languages according to the user's interest.

Acknowledgments

I would like to express our heartfelt gratitude to Mr. G. Vivek, our project guide, for his invaluable guidance and constant support, along with his capable instruction and persistent encouragement.

I am grateful to our Head of Department, Dr. M. Swamy Das, for his steady support and the provision of every resource required for the completion of this project.

I would like to take this opportunity to thank our Principal, Dr. Ravinder Reddy, as well as the management of the institute, for having designed an excellent learning atmosphere.

My thanks are due to all members of the staff and our lab assistants for providing me with the help required to carry out the groundwork of this project.

List Of Figures

- Figure 1: Block diagram
- Figure 2: DFD
- Figure 3: Use case
- Figure 4: Home Page components
- Figure 5: Sign up form
- Figure 6: sign in form
- Figure 7: To show the list of PG Hotels
- Figure 8: Map implementation
- Figure 9: Personal details form
- Figure 10: Payment integrated with Razor Pay.
- Figure 11:Home page function
- Figure 12:register function
- Figure 13:sign in function
- Figure 14:logout function
- Figure 15:pg lists function
- Figure 16:map function
- Figure 17:pay function
- Figure 18:User table part1
- Figure 19: user table part2
- Figure 20:Hotels table
- Figure 21:User details table part1
- Figure 22: User details table part2
- Figure 22 :Django admin log table

Table of Contents

	Title Page	i
	Certificate of the Guide	ii
	Declaration of the Student	iii
	Abstract	iv
	Acknowledgement	v
	List of Figures	vi
1.	INTRODUCTION	1
	1.1 Problem Definition including the significance and objective	1
	1.2 Methodologies	1
	1.3 Outline of the results	2
	1.4 Scope of the project	2
	1.5 Organization of the report	3
2.	LITERATURE SURVEY	4
	2.1 Introduction to the problem domain terminology	4
	2.2 Existing system	4
	2.3 Proposed system	
	2.4 Tools/Technologies used (if required)	5
	2.4.1 Software requirements	5
	2.4.2 Hardware requirements	
3.	DESIGN OF THE PROPOSED SYSTEM/METHOD/ALGORITHM	6
	3.1 Block Diagram	6

	3.2 Module Description	7
4	IMPLEMENTATION OF THE PROPOSED SYSTEM	8
	4.1 Flowcharts / DFDs / ER Diagrams	8-9
	4.2 Design and Test Steps / Criteria	10
	4.3 Algorithms / Pseudo Code	21
	4.4 Dataset description	58
	4.5 Testing Process	59
5.	RESULTS / OUTPUTS AND DISCUSSIONS	61
6.	CONCLUSIONS / RECOMMENDATIONS	70
	6.1 Conclusions	
	6.1.1 Limitations	70
	6.2 Recommendations / Future Work /Future Scope	71
	REFERENCES	71

CHAPTER 1

INTRODUCTION

1.1. Problem Definition and Objectives

The main problem in the existing society is people travel to different places and find very difficult to find a place to live for the few days they stay in cities with all the facilities they need like food, travel facilities, parking, wifi, Air conditioner, etc, some people want to live for days there are many PG's which could be used to stay for months but many PG's will not have a daily basis and people will find it difficult to differentiate these hotels and PG's.

The main objective of this project is to provide a user-friendly web-based system that can serve many issues like provide every single detail of the PG and hotels available and also to make the availability of PG's visible to all the users by our website. PG owners can change and handle the data according to the updates. Users should not face any difficulties in finding the PG's in their comfort zone. Also, there are many PGs who handle on an hourly basis they are not known to many people so it can be handy to them by saving money for the extra hours they usually pay in many hotels.

1.2. Methodologies

We will see many people roaming for a place to live in for a few days in the city and students especially when they come to internships and some coaching stuff to cities. They all find it difficult to search for a place manually. We also find many PG's hostels and hotels giving adds in papers, Tv's, etc. They all need a platform to get things right. They find it very comfortable if they find a place to live first. Also, they need to check all the facilities and requirements they need every PG will not provide the same facilities.

We usually see these statements and problems in many social media noting all the problems they faced while they moved to new cities. A variety of research has been done on the learning of collective behavior.

1.3. Outline of results

So they need a web-based platform that can solve the above-mentioned problem. That is a system that can be helpful for them to minimize the time in searching PG.

The system must allow the user to do their booking online by themselves. Some of the tasks that the system can do are providing a query for arriving date and the duration of staying, providing the number of rooms, view all available rooms and provides the user ability to choose one or more of them, and how many going to be in the single room or double, providing the cost of the booking, asking the users if they want additional services such as dinner or breakfast, storing the user detail like name, address, and telephone, asking the user for confirmation, final confirmation views with the detail of booking and the guests can review or cancel the booking.

1.4. Scope of the Project

The system can be used by many people all over the globe but this is restricted to only a few places in the Hyderabad city, later we will be integrating it to many other places in India.

It can also be used by the students for staying who need to stay in a hostel for various reasons. It can be used by many working people who leave in the city to stay in a PG with reasonable prices.

1.5. Organization of the report

The report is divided into six chapters each chapter gives you some different approach of understanding the project as,

Chapter 1: already gave you the introduction, objectives, scope of the project.

Chapter 2: It describes the existing systems related to this problem and what are the drawbacks in that project and all the related works. It also gives you the hardware and software components required for this system to use.

Chapter 3: It shows you the block diagram of the system. It also explains the details of all the modules present in the system

Chapter 4: It explains to you about the implementation with the help of ER diagrams, Flowcharts, etc. and give you snippets of the code for the modules used to built the system.

Chapter 5: It gives you the information about the results and outputs of the decisions we make such as booking, payment, login/register results, etc.

Chapter 6: It gives you a conclusion of the report including limitations and future enhancements of our project.

CHAPTER 2

LITERATURE SURVEY

2.1. Introduction

This system allows the user to do their booking online by themselves. Some of the tasks that the system can do are providing a query for arriving date and the length of staying, providing the number of rooms, view all available rooms and provides the user the ability to choose one or more of them, recording the number of rooms, view all available rooms and provides the user the ability to choose one or more of them, providing the cost of the booking, asking the users if they want additional service such as dinner or breakfast, storing the user detail like name, address, and telephone, asking the user for confirmation, final confirmation views with the detail of booking and the guests can review or cancel the booking.

2.2. Existing System

In the existing system, users can book the rooms based on his requirements but he may not be fully satisfied with the service because of the environment, food, etc. Also, the user may not know the regional language and communicate with the PG owner effectively. Users may like the PG and want to stay there, but if rooms are not available, he needs to check for another PG.

2.3. Proposed System

Here we give detailed information about the services and the environment such as near bus stations, metro stations, all the transport and parks, malls, etc. If the user finds some PG interesting and wants to stay there he can check on the notifications whenever there is availability there. Searching is effectively done in this. And the cost is calculated on the basis of number of hours he/she has stayed.

2.4. Software Requirement Specification

2.4.1. User Requirements

An active internet connection is required.

2.4.2. Software Requirements

- Any 32-bit or 64-bit operating system that supports Python.
- Python installation (version 3.5)
- Windows or Linux
- Html, CSS, Javascript, Bootstrap Django
- Postgre SQL

2.4.3. Hardware Requirements

- Intel I3 processor or above version
- Tracking user GPS location
- 32-bit or 64-bit processor
- Minimum ram of 1gb is required.

CHAPTER 3

DESIGN OF PROPOSED SYSTEM

3.3 Block diagram

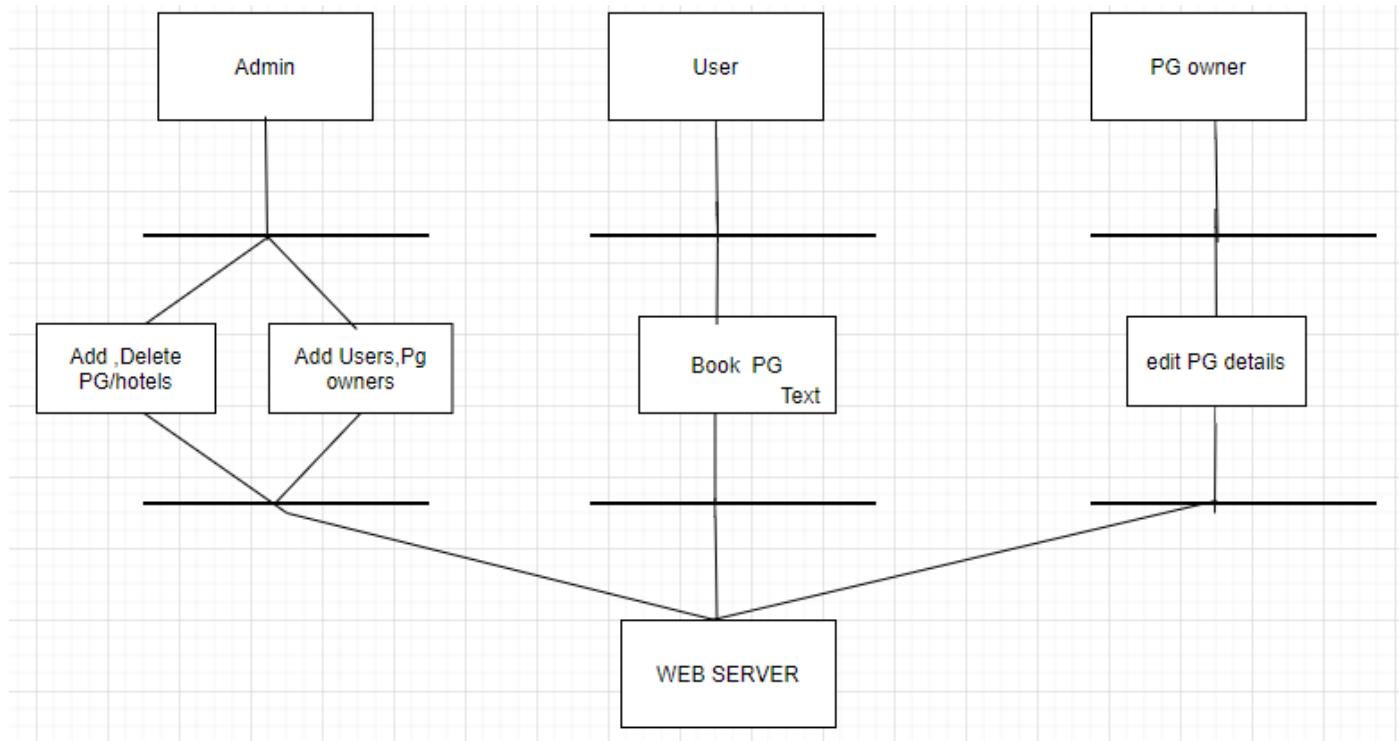


Figure 1: Block diagram

Admin can add and delete the PG's, they can allow users, PG owners to register into the platform.

Users can book the PG according to their requirements specified.

PG owner can make changes to the specifications of the PG.

3.2 Module Description

There are three modules in the proposed system they are :

- 1.Admin**
- 2. User**
- 3.PG owner**

3.2.1 ADMIN:

Admin has 2 main functions they are :

- To maintain the user's login/register details.
- To maintain PG details and issue login for PG owner.
- Admin checks the details of the users and gives him access to the system by providing username and password.
- He also maintains the PG details by adding and deleting the PG's, hotels.
- He maintains the database.

The Admin can also update his profile, change the password and recover the password.

3.2.2 User:

User can do the following activities :

Two types of users 1. Registered users

2. Guest users

- Registered users :
 - Can Book the pg.
 - Update his/her own profile.
- Guest User:
 - In this guest, the user can see only general information like about us, pg details, search pg and contact details.

3.2.3 PG Owner:

The owner can do the following activities:

- The owner can see all detail in brief like total listed pg, total booking, total new booking, total confirmed booking, and total canceled booking.
- The owner can search the booking request by booking number.
- The owner can change or update the availability of rooms and any new features if added to his PG

The owner can also update his profile, change the password and recover the password.

Chapter 4

IMPLEMENTATION OF THE PROPOSED SYSTEM

4.1 Flowcharts / DFDs / ER Diagrams.

4.1.1 DFD

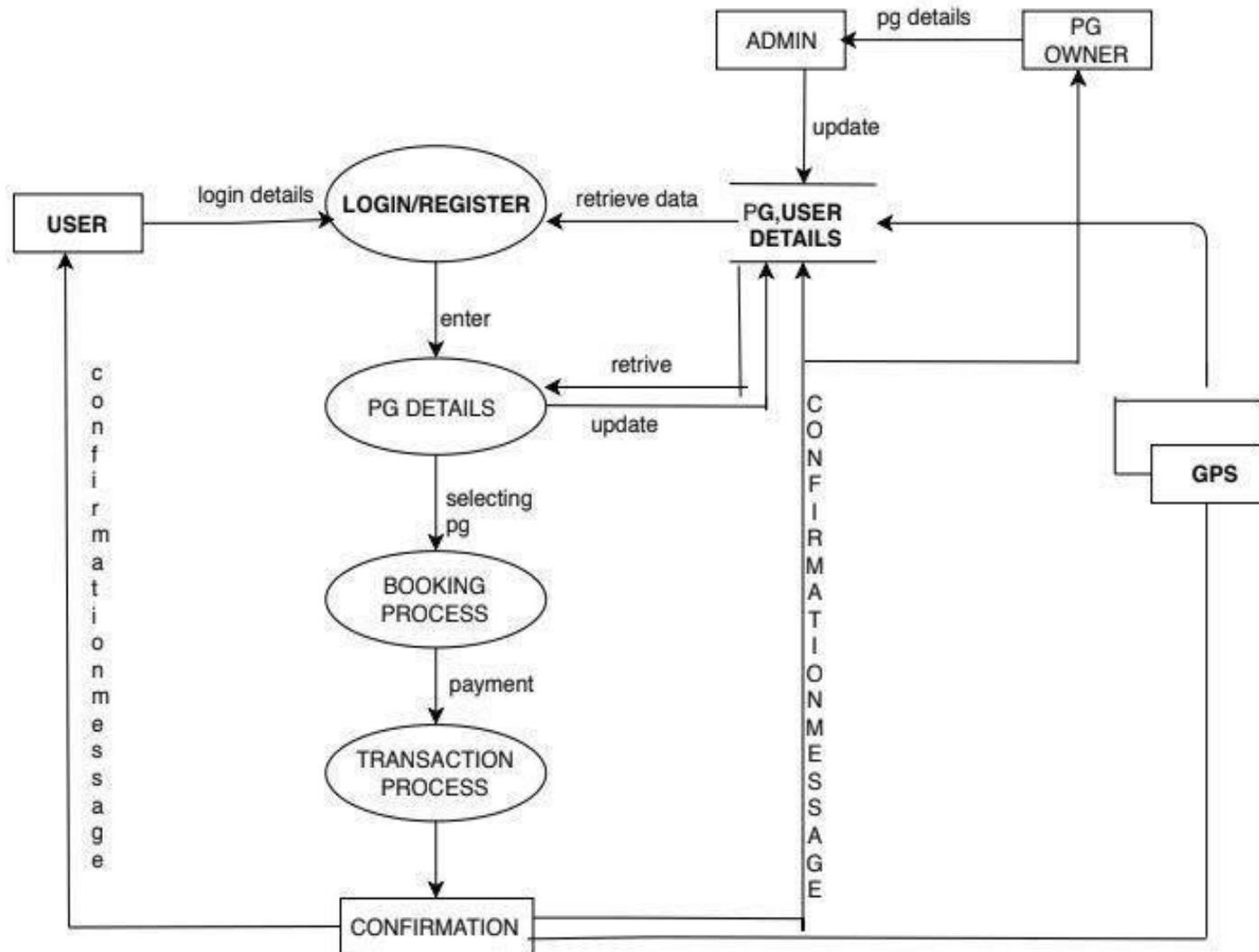


Figure 2: DFD

As the figure clearly shows how the flow of the system takes place.

User will login and search for the PG and book the Pg by making the payment then the confirmation message will be sent to him and the PG owner .

PG owner can update the Pg details by logging in with his credentials.

4.1.3 Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. ... The actors, usually individuals involved with the system defined according to their roles.

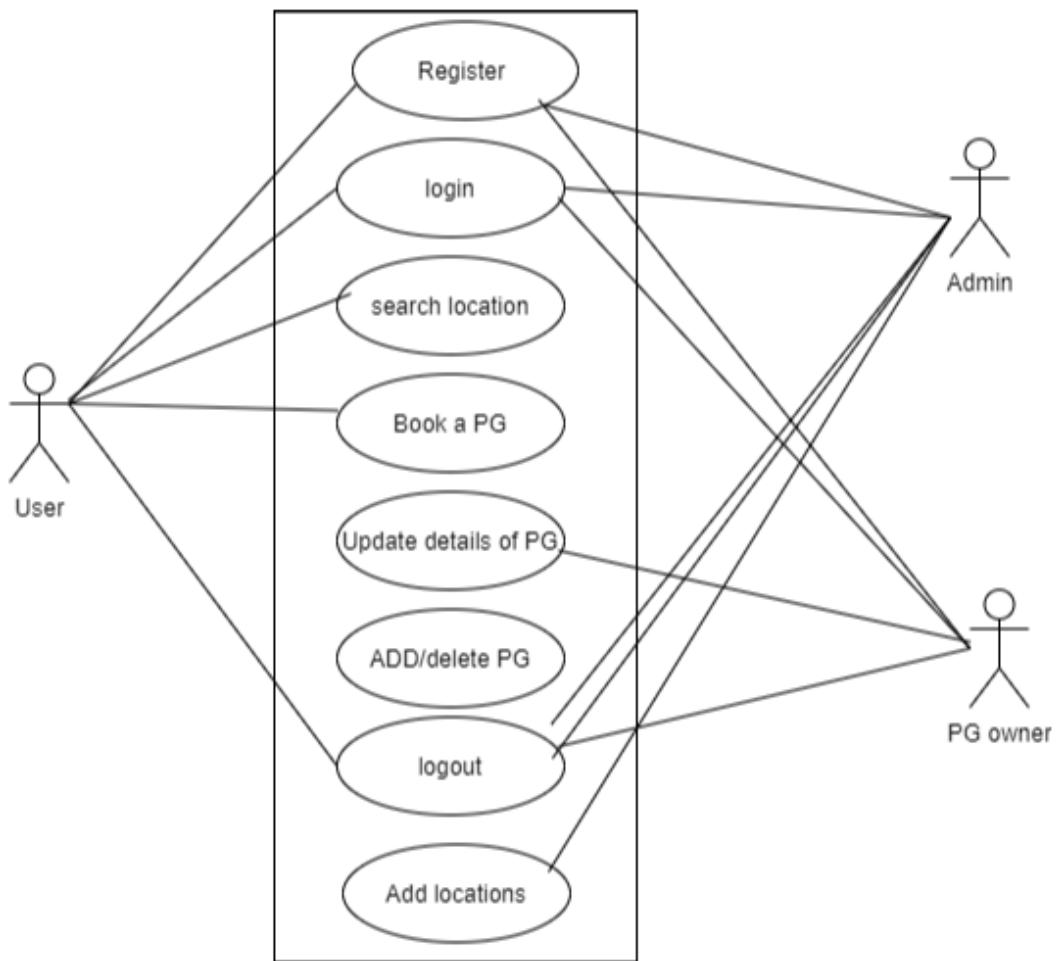


Figure 3: Use Case

This is the Use case diagram of this system it contains three actors and 8 attributes.

User ,PG owner, Admin are the actors.

User does login ,registration, view pg, select pg, select location, book a pg, logout.

Admin registers ,login, add/delete pg's, add locations,logout.

Pg owner registers, login, update details about pg, logout.

4.2 Design and Test Steps / Criteria

The front-end and back-end of the program are bootstrap and python respectively. These are both explored in this section.

4.2.1 Front-End

The front-end of the project is the user interface that has been designed using the Bootstrap widgets. Some of them the Label, Button, Radiobutton, and TextFields widgets have been used in this project. These widgets together constitute the front-end.

```

<div class="menu_content">
  <ul>
    <li class="active"><a href="index.html">Home</a></li>
    <li><a href="about.html">About us</a></li>
    <li><a href="contact.html">Contact</a></li>
    {% if user.is_authenticated%}
    <li>Hello,{{user.first_name}} </li>
    <li><a href="reg/logout">Logout</a></li>
    {% else %}
    <li><a href="reg/register">Register</a></li>
    <li><a href="reg/signin">Login</a></li>
    {% endif %}

  </ul>
</div>
<div class="menu_social">
  <div class="menu_phone ml-auto">Call us: 00-56 445 678 33</div>
  <ul class="d-flex flex-row align-items-start justify-content-start">
    <li><a href="#"><i class="fa fa-pinterest" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-facebook" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-dribbble" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-behance" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-linkedin" aria-hidden="true"></i></a></li>
  </ul>
</div>
</div>

<!-- Home -->

<div class="home">

```

Figure 4: Home Page components

```

<div class="signup-content">
  <div class="signup-form">
    <h2 class="form-title">Sign up</h2>
    <form action='register' method='POST' class="register-form" id="register-form">
      {%- csrf_token %}
      <div class="form-group">
        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
        <input type="text" name="first_name" id="name" placeholder="First Name"/>
      </div>
      <div class="form-group">
        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
        <input type="text" name="last_name" id="name" placeholder="Last Name"/>
      </div>
      <div class="form-group">
        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
        <input type="text" name="username" id="name" placeholder="User Name"/>
      </div>
      <div class="form-group">
        <label for="email"><i class="zmdi zmdi-email"></i></label>
        <input type="email" name="email" id="email" placeholder="Your Email"/>
      </div>
      <div class="form-group">
        <label for="pass"><i class="zmdi zmdi-lock"></i></label>
        <input type="password" name="pass" id="pass" placeholder="Password"/>
      </div>
      <div class="form-group">
        <label for="re_pass"><i class="zmdi zmdi-lock-outline"></i></label>
        <input type="password" name="re_pass" id="re_pass" placeholder="Repeat your password"/>
      </div>
      <div class="form-group">
        <input type="checkbox" name="agree-term" id="agree-term" class="agree-term" />
        <label for="agree-term" class="label-agree-term"><span><span></span></span></span>I agree all statements in <a href="#" class="term-service">Terms of service</a></label>
      </div>
      <div class="form-group form-button">
        <input type="submit" name="signup" id="signup" class="form-submit" value="Register"/>
      </div>
    </center>
    <div>
      {%for message in messages%}
        <h3><font color="red">{{message}}</font></h3>
      {%endfor%}
    </div>
  </div>

```

Activate Windows
Go to Settings to activate Windows.

Figure 5: Signup Form

```

<div class="signin-form">
  <h2 class="form-title">Sign in</h2>
  <form method="POST" class="register-form" id="login-form">
    <div class="form-group">
      <label for="your_name"><i class="zmdi zmdi-account material-icons-name"></i></label>
      <input type="text" name="your_name" id="your_name" placeholder="Your Name"/>
    </div>
    <div class="form-group">
      <label for="your_pass"><i class="zmdi zmdi-lock"></i></label>
      <input type="password" name="your_pass" id="your_pass" placeholder="Password"/>
    </div>
    <div class="form-group">
      <input type="checkbox" name="remember-me" id="remember-me" class="agree-term" />
      <label for="remember-me" class="label-agree-term"><span><span></span></span></span>Remember me</label>
    </div>
    <div class="form-group form-button">
      <input type="submit" name="signin" id="signin" class="form-submit" value="Log in"/>
    </div>
    <center>
      <div>
        {%for message in messages%}
          <h3><font color="red">{{message}}</font></h3>
        {%endfor%}
      </div>
    </center>
  </div>

```

Activate Windows

Figure 6: SignIn Form

```

<form action="pglists" method='POST' id="myform">
    {% csrf_token %}
    {% for dest in dests %}
        <!-- Destination -->
        <div class="destination_item">
            <div class="destination_image">
                

                {% if dest.offer %}
                    <div class="spec_offer text-center"><a href="#">Special Offer</a></div>
                {% endif %}
            </div>
            <div class="destination_content">
                <div class="destination_title">

                    <label ><b> <a href="javascript: void(0)" onclick="document.getElementById('myform').submit();">{{dest.name}}</a></b><label>
                        <input type="hidden" name="hotels" value="{{dest.name}}">
                    </div>
                    <div class="destination_subtitle">
                        <p>{{dest.location}}</p>
                        <label><b><a href="map">Directions</a></b><label>
                    </div>
                    <div class="destination_price">From ${{dest.price}}</div>
                </div>
            </div>
        {% endfor %}
    </form>
</div>

```

Activate Windows
Go to Settings to activate Windo

Figure 7: To show the list of PG Hotels

```

<div id="map"></div>
<script>
    mapboxgl.accessToken = '{ mapbox_access_token }';
    mapboxgl.accessToken = 'pk.eyJ1Ijoi3JpZGhhcJAwMDciLCJhIjoiY2s3b2lra3N4MDVyNzNmXNsZGzsY2htdiJ9.QGZpr-aeWNBms5QBmdHEUA';
    var map = new mapboxgl.Map({
        container: 'map',
        style: 'mapbox://styles/mapbox/streets-v11',
        center: [78.46667, 17.36667],
        zoom: 9
    });

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.3211, 17.3848])
        .setHTML('<h1><b>My hotel</b></h1>')
        .addTo(map);
    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.4, 17.3])
        .setHTML('<h1><b>Taj Banjara</b></h1>')
        .addTo(map);
    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.3169, 17.380799])
        .setHTML('<h1><b>Marriot</b></h1>')
        .addTo(map);
    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.548487, 17.097637])
        .setHTML('<h1><b>Mango Suites</b></h1>')
        .addTo(map);
    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.5, 17.45])
        .setHTML('<h1><b>Radission</b></h1>')
        .addTo(map);

```

Figure 8 : Map Implementation

```

<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Arrival_Date:</b></label>
    <div class="col-sm-10">
        <input type="date" id="birthday" class="form-control" name="arrival">
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Time:</b></label>
    <div class="col-sm-10">
        <input type="time" id="appt" class="form-control" name="time">
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Departure_Date:</b></label>
    <div class="col-sm-10">
        <input type="date" id="birthday" class="form-control" name="departure">
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Gender:</b></label>
    <div class="col-sm-10">
        <div class="radio">
            <label><input type="radio" name="gender" value='Male'>Male</label>
        </div>
        <div class="radio">
            <label><input type="radio" name="gender" value='Female'>Female</label>
        </div>
        <div class="radio">
            <label><input type="radio" name="gender" value='Others'>Others</label>
        </div>
    </div>
</div>

```

Figure 9: Personal Details Form

```

<div class="razorpay-embed-btn" data-url="https://rzp.io/l/ny716WS" data-text="Pay Now" data-color="#528FF0" data-size="large">
<script>
(function(){
    var d=document; var x=!d.getElementById('razorpay-embed-btn-js')
    if(x){ var s=d.createElement('script'); s.defer=!0;s.id='razorpay-embed-btn-js';
    s.src='https://cdn.razorpay.com/static/embed_btn/bundle.js';d.body.appendChild(s); } else{var rzp=window['__rzp__'];
    rzp && rzp.init && rzp.init()} }()
</script>
</div>

```

Figure 10: Payment integrated with Razor Pay.

4.2.1.1 Labels

Labels are the simplest widgets, used to display text at a specified location in the application.

The labels used in this project are as follows:

Table 1: Labels

S.NO	Label	Functionality
------	-------	---------------

1	Home	Returns to the home page.
2	About us	It includes the functionalities and the PG Hotels information.
3	Contact	It displays the contact information of PG Hotels.
4	Register	Ask the user to register.
5	Login	Ask the user to login to the website.
6	Hotel Name	Displays the name of the hotel.
7	Directions	Ask the user to show direction and navigation.
8	Place	It displays the location of the hotel.
9	Price	It displays the price of the hotel.
10	Special Offer	It displays, whether a special offer is included or not.
11	First_name	Ask the user to enter the first_name.
12	Last_name	Ask the user to enter the last_name.
13	Email_Id	Ask the user to enter the email_id.
14	Phone_Number	Ask the user to enter the phone number.
15	Gender	Ask the user to enter the gender.
16	Arrival_Date	Ask the user to enter the arrival date for a hotel.
17	Time	Ask the user to enter the time of reservation.
18	Departure_Date	Ask the user to enter the departure date.
19	Selected_Hotel	Ask the user to enter the selected hotel at booking.
20	Amount	Ask the user to enter the amount to be paid.

4.2.1.2 Buttons

Buttons are associated with commands that they need to execute when pressed. The buttons used in this project are as follows:

Table 2: Buttons

S. No.	Button	Functionality
1	Register	It allows the user to register.
2	Login	It allows the user to login.

3	Search	It allows the user to search hotels in a specified location.
4	Logout	It allows the user to quit the website.
5	Submit	It allows the user to submit our booking details.
6	Make Payment	It allows the user to submit the payment amount.
7	Geo Location	It allows the user to show the exact geo-location..

4.2.1.3 Entries

Entries are used to accept text from the user in the form of a text field. The entries used in this project are as follows:

S. No.	Entries	Functionality
1	User_name	It allows the user to enter a username.
2	First_name	It allows the user to enter the First name.
3	Last_name	It allows the user to enter the Last_name.
4	Email_Id	Allows the user to enter the Email_Id.
5	Phone_Number	Allows the user to enter the Phone_Number.
6	Gender	It allows the user to enter Gender.
7	Arrival_Date	Allows the user to enter the Arrival_Date.
8	Time	It allows the user to enter the Time.
9	Departure_Date	It allows the user to enter the Departure_Date.
10	Search	It allows the user to search for a specific location.
11	Selected_Hotel	It allows the user to enter the selected Hotel.
12	Amount	It allows the user to enter the total amount.
13	Password	It allows the user to enter a password.

4.2.1.4 Radio buttons

Radio buttons are used to allow the user to choose one of many options.

S. No.	Entries	Functionality
1	Gender	It allows the user to choose the gender.

2	Room_type	Allows the user to choose the room type i.e.,single, double, or triple.
3	No_of_guests	It allows the user to choose the number of guests.

4.4.2 Back_End

The back-end of this project consists of user-defined functions that drive the program, and the database which stores the user data and PG Hotels data. The user data and the Hotels related data are stored in the form of tables, which is connected to the PostgreSQL Database.

4.4.2.1. Functions

The functions are the core of the project and move the control of the program between sequential screens in the user interface. The functions defined in the project are as follows:

Table 4: Functions

S. No.	Function
1	Index
2	Register
3	Sign in
4	Logout
5	Pglist
6	Pglists
7	Map
8	Pay

```
from django.shortcuts import render
# Create your views here.

def index(request):
    return render(request, "index.html")
```

Figure 11 : Homepage function

```
def register(request):
    if request.method=='POST':
        first_name=request.POST['first_name']
        last_name=request.POST['last_name']
        username=request.POST['username']
        email=request.POST['email']
        password1=request.POST['pass']
        password2=request.POST['re_pass']

        send_mail('PGHOTEL','Hello {} ,your registration at PG HOTELS is successful '.format(first_name),'kandisridhar123@gmail.com',[email],fail_silently=False,
if password1==password2:
    if User.objects.filter(username=username).exists():
        messages.info(request,'Username Taken')
        return redirect('register')
    else:
        user=User.objects.create_user(username=username,password=password1,email=email,first_name=first_name,last_name=last_name)
        user.save()
        return redirect('signin')
    else:
        messages.info(request,'Password not matching')
        return redirect('register')
    return redirect('/')
else:
    return render(request,'register.html')
```

Figure 12 : register function

```
from django.contrib.auth.models import User,auth
from django.conf import settings
from django.core.mail import send_mail
from django.http import HttpResponseRedirect

# Create your views here.

def signin(request):
    if request.method == 'POST':
        username=request.POST['your_name']
        password=request.POST['your_pass']
        user=auth.authenticate(username=username,password=password)

        if user is not None:
            auth.login(request,user)
            return HttpResponseRedirect("pglist")
        else:
            messages.info(request,'invalid credentials')
            return HttpResponseRedirect('signin')

    else:
        return render(request,'signin.html')
```

Figure 13 : sign in function

```
def logout(request):
    auth.logout(request)
    return HttpResponseRedirect('/')
```

Figure 14: logout function

```

from django.shortcuts import render
from .models import pghotels
from .models import userdetailsxx
from django.contrib.auth.models import User,auth
# Create your views here.

def pglist(request):
    if request.method=='POST':
        loc=request.POST['search']
        dests=pghotels.objects.filter(location=loc)
        return render(request, "pglist.html", {'dests': dests})

    dests=pghotels.objects.all()

    return render(request, "pglist.html", {'dests': dests})

```

Figure 15: pg list function

```

def pglists(request):

    if request.method=='POST':
        ht=request.POST['hotels']
        return render(request, "pay.html", {'hotel': ht})

    dests=pghotels.objects.all()
    return render(request, "pglist.html", {'dests': dests})

```

Figure 16: pg lists function

```

def map(request):
    # TODO: move this token to Django settings from an environment variable
    # found in the Mapbox account settings and getting started instructions
    # see https://www.mapbox.com/account/ under the "Access tokens" section
    mapbox_access_token = 'pk.eyJ1Ijoic3JpZGhhcjAwMDciLC3hIjoiY2s3b21ra3NAMDVyNzNmbXNsZGzsY2htdiJ9.QGZpr-aeWNBms5QBmdHEUA'
    return render(request, 'map.html',
                 { 'mapbox_access_token': mapbox_access_token })

```

Figure 17: map function

```

def pay(request):
    if request.method=='POST':
        fname=request.POST['fname']
        lname=request.POST['lname']
        email=request.POST['email']
        tel=request.POST['pno']
        rtype=request.POST['radio']
        guest=request.POST['optradio']
        adate=request.POST['arrival']
        time=request.POST['time']
        ddate=request.POST['departure']
        gender=request.POST['gender']
        selhot=request.POST['hotelname']

        #x=userdetails.objects.filter(user_id=(request.user).id)

        details=userdetailsxx.objects.create(user_id=(request.user).id,firstname=fname,lastname=lname,email=email,pno=tel,room=rtype,guest=guest,adate=adate,time=time,ddate=ddate,gender=gender)
        details.save()
        return render(request,"razorpay.html")
    else:
        return render(request, "pay.html")

```

Figure 18: pay function

4.2.3 Database

The database we are using is PostgreSQL which is integrated with the Django Framework and the database of the project consists of user data and the PG Hotels data. The program draws from both of these at different stages of the program repeatedly, and also stores data in the appropriate tables.

The username and password of a user are stored in a user table. This file is read from at the registration stage to ensure that a particular username is not already taken, and is written to when a user registers successfully. When a user tries to log in by entering their username and password, the program reads from the user table to verify that the username and password match. Thus, this inbuilt table in Django enables user authentication. The remaining booking and hotel information are stored in different tables.

Data Output Explain Messages Notifications							
	id [PK] integer	password character varying (128)	last_login timestamp with time zone	is_superuser boolean	username character varying (150)	first_name character varying (30)	
1	19	pbkdf2_sha256\$180000\$F8Q...	2020-04-01 16:51:39.672857+05:...	false	sridhar444	sridhar	
2	18	pbkdf2_sha256\$180000\$QlaY...	2020-04-02 10:27:17.403036+05:...	false	shivansh	sridhar	
3	4	pbkdf2_sha256\$180000\$qNe...	2020-03-30 16:13:34.00402+05:30	false	hello123	abcd	
4	20	pbkdf2_sha256\$180000\$9jK...	2020-04-02 20:22:12.565776+05:...	false	sridhar12345	sridhar	
5	21	pbkdf2_sha256\$180000\$7uTl...	2020-04-02 21:03:25.901874+05:...	false	shivan12345	shivan	
6	2	pbkdf2_sha256\$180000\$rgCj...	2020-04-07 15:15:32.181053+05:...	false	sridhar0007	sridhar	
7	22	pbkdf2_sha256\$180000\$plQ1...	2020-04-07 16:10:18.39397+05:30	false	raju123	sridhar	Activate Window
8	23	pbkdf2_sha256\$180000\$yhg...	2020-04-07 16:23:37.690466+05:...	false	ramu123	sridhar	Go to Settings

Figure 19: User table Part-1

last_name character varying (150)	email character varying (254)	is_staff boolean	is_active boolean	date_joined timestamp with time zone
kandi	sridharkandi007@gmail.com	false	true	2020-04-01 16:51:28.759988+05:...
kandi	kandisridhar123@gmail.com	false	true	2020-04-01 16:47:21.843441+05:...
wxyz	kandisridhar123@gmail.com	false	true	2020-03-30 16:13:22.635111+05:...
kandi	kandisridhar123@gmail.com	false	true	2020-04-02 20:19:21.632934+05:...
kandi	kandisridhar123@gmail.com	false	true	2020-04-02 21:00:19.191155+05:...
kandi	sridharkandi007@gmail.com	false	true	2020-03-11 23:56:46.388587+05:...
kandi	kandisridhar123@gmail.com	false	true	2020-04-07 16:09:43.258576+05:...
kandi	kandisridhar123@gmail.com	false	true	2020-04-07 16:21:14.289656+05:...

Figure 20: User table Part-2

Data Output Explain Messages Notifications							
	id [PK] integer	name character varying (100)	img character varying (100)	price integer	offer boolean	location text	
1	1	Marriot	pics/marriot.jpg	500	true	gandipet	
2	2	Mango suites	pics/mango_suites.jpg	450	false	gachibowli	
3	3	Novotel	pics/novotel.jpg	750	false	manikonda	
4	4	Radission	pics/radission.jpg	550	false	gandipet	
5	5	Taj Hotel	pics/taj.jpg	800	true	abids	
6	6	Taj Krishna	pics/tajkrishna.jpg	800	true	alwal	
7	7	sheraton	pics/sheraton.jpg	950	true	hayathnagar	
8	8	siesta	pics/siesta.jpg	900	true	madhapur	

Figure 21: Hotels table

Data Output Explain Messages Notifications						
	user_id [PK] integer	firstname character varying (100)	lastname character varying (100)	email character varying (254)	pno integer	room character varying (100)
3	4	sridhar	kandi	kandisridhar123@gmail.com	2345678	on
4	17	shivan	k	kandisridhar123@gmail.com	12345	on
5	18	sridhar	kandi	kandisridhar123@gmail.com	1234	checked
6	19	sridhar	kandi	Sri.sridhar900@gmail.com	12345	Single
7	20	sridhar	kandi	kandisridhar123@gmail.com	1234566	Double
8	21	sridhar	kandi	kandisridhar123@gmail.com	12345	Double
9	23	sridhar	kandi	kandisridhar123@gmail.com	12345	Double

Figure 22: User Details table Part-1

guest character varying (100)	adate character varying (100)	time character varying (100)	ddate character varying (100)	gender character varying (100)	selhotel character varying (100)
on	2020-03-30	15:33	2020-03-30	on	Taj Hotel
on	2020-04-01	16:38	2020-04-01	on	Mango suites
checked	2020-04-01	16:48	2020-04-01	checked	Taj Hotel
2(Two)-4(Four)	2020-04-01	16:52	2020-04-01	Male	Taj Hotel
2(Two)-4(Four)	2020-04-02	20:24	2020-04-03	Male	Taj Krishna
2(Two)-4(Four)	2020-04-02	21:06	2020-04-03	Male	sheraton
2(Two)-4(Four)	2020-04-07	16:25	2020-04-07	Male	Activate Windows

Figure 23: User Details table Part-2

Data Output Explain Messages Notifications										
	id [PK] integer	action_time timestamp with time zone	object_id text	object_repr character varying (200)	action_flag smallint	change_message text	content_type_id integer	user_id integer		
1	1	2020-03-11 23:59:39.002339+05:...	1	pghotels object (1)	1	[{"added": {}}]	1	1		
2	2	2020-03-12 00:11:10.979177+05:...	2	pghotels object (2)	1	[{"added": {}}]	1	1		
3	3	2020-03-12 00:11:32.510199+05:...	3	pghotels object (3)	1	[{"added": {}}]	1	1		
4	4	2020-03-12 00:11:56.530646+05:...	4	pghotels object (4)	1	[{"added": {}}]	1	1		
5	5	2020-03-12 13:40:58.156398+05:...	5	pghotels object (5)	1	[{"added": {}}]	1	1		
6	6	2020-03-20 20:23:16.42669+05:30	1	pghotels object (1)	2	[{"changed": {"fields": ["..."]}]	1	1		
7	7	2020-03-28 18:38:06.174701+05:...	2	pghotels object (2)	2	[{"changed": {"fields": ["..."]}]			Activate Windows	
8	8	2020-03-28 18:38:20.702096+05:...	3	pghotels object (3)	2	[{"changed": {"fields": ["..."]}]			Go to Settings to activate Windows	
9	9	2020-03-29 10:39:20.642205+05:...	4	pghotels object (4)	2	[{"changed": {"fields": ["..."]}]	1	1		

Figure 24: Django-admin-log table

4.3 Algorithms/ Pseudo Code

Sample App(Main App)

Sample-urls.py

```
from django.contrib import admin
```

```

from django.urls import path,include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path("",include('sapp.urls')),
    path('admin/', admin.site.urls),
    path('reg/',include('reg.urls')),
    path('list/',include('list.urls')),
    path('payment/',include('payment.urls')),
]
urlpatterns
=urlpatterns+static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)

```

Settings.py

```

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '3_*bc)0*g%r)fdecxdi5@o2-=4lt0&#b)k2p#@nv714fer1^n^'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['192.168.43.113','127.0.0.1']

# Application definition

INSTALLED_APPS = [
    'sapp.apps.SappConfig',

```

```

'payment.apps.PaymentConfig',
'reg.apps.RegConfig',
'list.apps.ListConfig',
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'sample.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

```

        },
        },
    ]
WSGI_APPLICATION = 'sample.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'sampledb',
        'USER': 'postgres',
        'PASSWORD': '910075',
        'HOST': 'localhost'
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
# Internationalization

```

```

# https://docs.djangoproject.com/en/3.0/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR,'static')
]
STATIC_ROOT=os.path.join(BASE_DIR,'assets')
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

EMAIL_BACKEND ='django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
EMAIL_HOST_USER = 'kandisridhar123@gmail.com'
EMAIL_HOST_PASSWORD = '*****'

```

Sapp(HomePage App)

Views.py

```

from django.shortcuts import render
# Create your views here.
def index(request):
    return render(request, "index.html")

```

Urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path("",views.index,name='index')
]
reg(Registration and login App)
views.py
from django.shortcuts import render,redirect
from django.contrib import messages
#from .models import Register
from django.contrib.auth.models import User,auth
from django.conf import settings
from django.core.mail import send_mail
from django.http import HttpResponseRedirect
# Create your views here.

def signin(request):
    if request.method == 'POST':
        username = request.POST['your_name']
        password = request.POST['your_pass']
        user = auth.authenticate(username=username, password=password)
        if user is not None:
            auth.login(request, user)
            return HttpResponseRedirect("pglist")
        else:
            messages.info(request, 'invalid credentials')
            return HttpResponseRedirect('signin')
    else:
        return render(request, 'signin.html')

def register(request):
    if request.method == 'POST':
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']

```

```

username=request.POST['username']
email=request.POST['email']
password1=request.POST['pass']
password2=request.POST['re_pass']

send_mail('PGHOTEL','Hello {}, your registration at PG HOTELS is successful
'.format(first_name),'kandisridhar123@gmail.com',[email],fail_silently=False,)

if password1==password2:
    if User.objects.filter(username=username).exists():
        messages.info(request,'Username Taken')
        return redirect('register')
    else:
        user=User.objects.create_user(username=username,password=password1,email=email,first_n
ame=first_name,last_name=last_name)
        user.save();
        return redirect('signin')
    else:
        messages.info(request,'Password not matching')
        return redirect('register')
    return redirect('/')
else:
    return render(request,'register.html')

def logout(request):
    auth.logout(request)
    return redirect('/')

```

urls.py

```

from django.urls import path
from . import views
urlpatterns = [

```

```
path('register',views.register,name='register'),
path('signin',views.signin,name='signin'),
path('logout',views.logout,name='logout')
]
```

list(to show list of hotels,booking,navigation and payment App)

views.py

```
from django.shortcuts import render
from .models import pghotels
from .models import userdetailsxx
from django.contrib.auth.models import User,auth
# Create your views here.

def pglist(request):
    if request.method=='POST':
        loc=request.POST['search']
        dests=pghotels.objects.filter(location=loc)
        return render(request, "pglist.html", {'dests': dests})

    dests=pghotels.objects.all()
    return render(request, "pglist.html", {'dests': dests})
```

def pglists(request):

```
if request.method=='POST':
    ht=request.POST['hotels']
    return render(request, "pay.html", {'hotel': ht})
dests=pghotels.objects.all()
return render(request, "pglist.html", {'dests': dests})
```

def map(request):

```
# TODO: move this token to Django settings from an environment variable
# found in the Mapbox account settings and getting started instructions
# see https://www.mapbox.com/account/ under the "Access tokens" section
```

```

mapbox_access_token = 'pk.eyJ1Ijoic3JpZGhhcjAwMDciLCJhIjoiY2s3b2lra3N4MDVyNzNmbXNsZGZsY2htdiJ9.QGZpr-aeWNBms5QBmdHEUA'

return render(request, 'map.html',
             { 'mapbox_access_token': mapbox_access_token })

def pay(request):
    if request.method=='POST':
        fname=request.POST['fname']
        lname=request.POST['lname']
        email=request.POST['email']
        tel=request.POST['pno']
        rtype=request.POST['radio']
        guest=request.POST['optradio']
        adate=request.POST['arrival']
        time=request.POST['time']
        ddate=request.POST['departure']
        gender=request.POST['gender']
        selhot=request.POST['hotelname']

        #x=userdetails.objects.filter(user_id=(request.user).id)

        details=userdetailsxx.objects.create(user_id=(request.user).id,firstname=fname,lastname=lname,email=email,pno=tel,room=rtype,guest=guest,adate=adate,time=time,ddate=ddate,gender=gender,selhotel=selhot)
        details.save()
        return render(request,"razorpay.html")
    else:
        return render(request, "pay.html")

```

urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path('pglist',views.pglist,name='pglist'),
    path('pglists',views.pglists,name='pglists'),
    path('map',views.map,name='map'),
    path('pay',views.pay,name='pay'),
]

```

Models.py

```

from django.db import models
from django.contrib.auth.models import User,auth
# Create your models here.

class pghotels(models.Model):
    name=models.CharField(max_length=100)
    img=models.ImageField(upload_to='pics')
    price=models.IntegerField()
    offer=models.BooleanField(default=False)
    location=models.TextField()

class userdetailsxx(models.Model):
    user = models.OneToOneField(User ,on_delete=models.CASCADE, primary_key=True)
    firstname=models.CharField(max_length=100)
    lastname=models.CharField(max_length=100)
    email=models.EmailField()
    pno=models.IntegerField()
    room=models.CharField(max_length=100)
    guest=models.CharField(max_length=100)
    adate=models.CharField(max_length=100)
    time=models.CharField(max_length=100)
    ddate=models.CharField(max_length=100)
    gender=models.CharField(max_length=100)

```

```
selhotel=models.CharField(max_length=100)

def __str__(self):
    return self.user.username
```

admin.py

```
from django.contrib import admin

from .models import pghotels,userdetailsxx

# Register your models here.

admin.site.register(pghotels)

admin.site.register(userdetailsxx)
```

Templates

Index.html

```
{% load static %}

{% static "images" as baseUrl %}

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Travello</title>

    <meta charset="utf-8">

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

    <meta name="description" content="Travello template project">

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

    <link rel="stylesheet" type="text/css" href="{% static 'styles/bootstrap4/bootstrap.min.css' %}">

</head>

<body>

    <div class="super_container">

        <header class="header">

            <div class="container">

                <div class="row">

                    <div class="col">
```

```

<div class="header_content d-flex flex-row align-items-center justify-content-start">
    <div class="header_content_inner d-flex flex-row align-items-end justify-content-start">
        <div class="logo"><a href="index.html">Paying Guest</a></div>
        <nav class="main_nav">
            <ul class="d-flex flex-row align-items-start justify-content-start">
                <li class="active"><a href="index.html">Home</a></li>
                <li><a href="about.html">About us</a></li>
                <li><a href="contact.html">Contact</a></li>
                {% if user.is_authenticated%}
                    <li>Hello, {{user.first_name}} </li>
                {% else %}
                    <li><a href="reg/logout">Logout</a></li>
                    <li><a href="reg/register">Register</a></li>
                    <li><a href="reg/signin">Login</a></li>
                {% endif %}</ul>
            </nav>
            <div class="header_phone ml-auto">Call us: 8639292963</div>
            <div class="hamburger ml-auto">
                <i class="fa fa-bars" aria-hidden="true"></i>
            </div>
            <div class="header_social d-flex flex-row align-items-center justify-content-start">
                <ul class="d-flex flex-row align-items-start justify-content-start">
                    <li><a href="#"><i class="fa fa-pinterest" aria-hidden="true"></i></a></li>
                    <li><a href="#"><i class="fa fa-facebook" aria-hidden="true"></i></a></li>
                    <li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>
                    <li><a href="#"><i class="fa fa-dribbble" aria-hidden="true"></i></a></li>
                    <li><a href="#"><i class="fa fa-behance" aria-hidden="true"></i></a></li>
                    <li><a href="#"><i class="fa fa-linkedin" aria-hidden="true"></i></a></li>
                </ul>
            </div>
        </div>
        <!-- Menu -->
        <div class="menu">

```

```

<div class="menu_header d-flex flex-row align-items-center justify-content-start">
  <div class="menu_logo"><a href="index.html">Travello</a></div>
  <div class="menu_close_container ml-auto">
    <div class="menu_close">
      </div>
    </div>
  </div>
<div class="menu_content"><ul>
  <li class="active"><a href="index.html">Home</a></li>
  <li><a href="about.html">About us</a></li>
  <li><a href="contact.html">Contact</a></li>
    {%- if user.is_authenticated %}
    <li>Hello, {{user.first_name}} </li>
    <li><a href="reg/logout">Logout</a>
      {%- else %}
    <li><a href="reg/register">Register</a></li>
    <li><a href="reg/signin">Login</a></li>
      {%- endif %}
    </ul></div>
<div class="menu_social">
  <div class="menu_phone ml-auto">Call us: 00-56 445 678 33</div>
  <ul class="d-flex flex-row align-items-start justify-content-start">
    <li><a href="#"><i class="fa fa-pinterest" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-facebook" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-dribbble" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-behance" aria-hidden="true"></i></a></li>
    <li><a href="#"><i class="fa fa-linkedin" aria-hidden="true"></i></a></li>
  </ul></div>
<!-- Home -->
<div class="home">
  <!-- Home Slider -->
  <div class="home_slider_container">

```

```
<div class="owl-carousel owl-theme home_slider">

<!-- Slide -->
<div class="owl-item">
<div class="background_image"
style="background-image:url({% static 'images/home_slider.jpg' %})"></div>
<div class="home_slider_content_container">
<div class="container">
<div class="row">
<div class="col">
<div class="home_slider_content">
<div class="home_title">
<h2>Let us take you away</h2>
</div>
<!-- Slide -->
<div class="owl-item">
<div class="background_image"
style="background-image:url({% static 'images/home_slider.jpg' %})"></div>
<div class="home_slider_content_container">
<div class="container">
<div class="row">
<div class="col">
<div class="home_slider_content">
<div class="home_title">
<h2>Let us take you away</h2>
</div>
<!-- Slide -->
<div class="owl-item">
<div class="background_image"
style="background-image:url({% static 'images/home_slider.jpg' %})"></div>
<div class="home_slider_content_container">
```



```

<div class="home_search_container">
<div class="home_search_title">Search for your trip</div>
<div class="home_search_content">
<form action="#" class="home_search_form" id="home_search_form">
<div
class="d-flex    flex-lg-row    flex-column    align-items-start    justify-content-lg-between
justify-content-start">
<input type="text" class="search_input search_input_1" placeholder="City"
required="required">
<input type="text" class="search_input search_input_2" placeholder="Departure"
required="required">
<input type="text" class="search_input search_input_3" placeholder="Arrival"
required="required">
<input type="text" class="search_input search_input_4" placeholder="Budget"
required="required">
<button class="home_search_button">search</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- Intro -->

```

```

<div class="intro">
<div class="intro_background" style="background-image:url({% static 'images/beach.svg'
%})"></div>
<div class="container">
<div class="row">

```

```
<div class="col">
<div class="intro_container">
<div class="row">

<!-- Intro Item -->
<div class="col-lg-4 intro_col">
<div class="intro_item d-flex flex-row align-items-end justify-content-start">
<div class="intro_icon"></div>
<div class="intro_content">
<div class="intro_title">Top Destinations</div>
<div class="intro_subtitle">
<p>Nulla pretium tincidunt felis, nec.</p>
</div>
</div>
</div>
</div>

<!-- Intro Item -->
<div class="col-lg-4 intro_col">
<div class="intro_item d-flex flex-row align-items-end justify-content-start">
<div class="intro_icon"></div>
<div class="intro_content">
<div class="intro_title">The Best Prices</div>
<div class="intro_subtitle">
<p>Sollicitudin mauris lobortis in.</p>
</div>
</div>
</div>
</div>

<!-- Intro Item -->
```

```
<div class="col-lg-4 intro_col">
<div class="intro_item d-flex flex-row align-items-end justify-content-start">
<div class="intro_icon">
</div>
<div class="intro_content">
<div class="intro_title">Amazing Services</div>
<div class="intro_subtitle">
<p>Nulla pretium tincidunt felis, nec.</p>
</div>
</div>
</div>
</div>

</div>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- Destinations -->

<div class="destinations" id="destinations">
<div class="container">
<div class="row">
<div class="col text-center">
<div class="section_subtitle">simply amazing places</div>
<div class="section_title">
<h2>Popular Destinations</h2>
</div>
</div>
</div>
</div>
```

```

<div class="row destinations_row">
<div class="col">
<div class="destinations_container item_grid">

{%- for dest in dests %}

<!-- Destination -->
<div class="destination_item">
<div class="destination_image">


{%- if dest.offer %}

<div class="spec_offer text-center"><a href="#">Special Offer</a></div>
{%- endif %}

</div>

<div class="destination_content">
<div class="destination_title"><a href="destinations.html">{{ dest.name }}</a>
</div>

<div class="destination_subtitle">
<p>{{ dest.desc }}</p>
</div>

<div class="destination_price">From ${{{ dest.price }}}</div>
</div>
</div>

{%- endfor %}

</div>
</div>
</div>
</div>
</div>

```

```

<!-- Testimonials -->

<div class="testimonials" id="testimonials">
<div class="parallax_background parallax-window" data-parallax="scroll"
data-image-src="{% static 'images/testimonials.jpg' %}" data-speed="0.8"></div>
<div class="container">
<div class="row">
<div class="col text-center">
<div class="section_subtitle">simply amazing places</div>
<div class="section_title">
<h2>Testimonials</h2>
</div>
</div>
</div>
<div class="row testimonials_row">
<div class="col">

<!-- Testimonials Slider -->
<div class="testimonials_slider_container">
<div class="owl-carousel owl-theme testimonials_slider">

</div>
</div>
<!-- News Post -->
<div
class="news_post d-flex flex-md-row flex-column align-items-start justify-content-start">
<div class="news_post_image"></div>
<div class="news_post_content">
<div class="news_post_date d-flex flex-row align-items-end justify-content-start">
<div>02</div>
<div>june</div>
</div>

```

```

<div class="news_post_title"><a href="#">Best tips to travel light</a>
</div>

<div class="news_post_category">
<ul>
<li><a href="#">lifestyle & travel</a></li>
</ul>
</div>

<div class="news_post_text">
<p>Pellentesque sit amet elementum ccumsan sit amet mattis eget,  

tristique at  

leo. Vivamus massa. Tempor massa et laoreet.</p>
</div>
</div>
</div>

```

```

<!-- News Post -->
<div
class="news_post d-flex flex-md-row flex-column align-items-start justify-content-start">
<div class="news_post_image"></div>
<div class="news_post_content">
<div class="news_post_date d-flex flex-row align-items-end justify-content-start">
<div>01</div>
<div>june</div>
</div>
<div class="news_post_title"><a href="#">Best tips to travel light</a>
</div>
<div class="news_post_category">
<ul>
<li><a href="#">lifestyle & travel</a></li>
</ul>
</div>
<div class="news_post_text">

```

```
<p>Tempor massa et laoreet malesuada. Pellentesque sit amet  
elementum ccumsan  
sit amet mattis eget, tristique at leo.</p>  
</div>  
</div>  
<div  
class="news_post d-flex flex-md-row flex-column align-items-start justify-content-start">  
<div class="news_post_image"></div>  
<div class="news_post_content">  
<div class="news_post_date d-flex flex-row align-items-end justify-content-start">  
<div>29</div>  
<div>may</div>  
</div>  
<div class="news_post_title"><a href="#">Best tips to travel light</a>  
</div>  
<div class="news_post_category">  
<ul>  
<li><a href="#">lifestyle & travel</a></li>  
</ul>  
</div>  
<div class="news_post_text">  
<p>Vivamus massa. Tempor massa et laoreet malesuada. Aliquam nulla  
nisl, accumsan  
sit amet mattis.</p>  
</div>  
</div>  
</div>  
<!-- News Sidebar -->  
<div class="col-4">  
<div class="travello">  
<div class="background_image"  
style="background-image:url({% static 'images/travello.jpg' %})"></div>
```

```
<div class="travello_content">
<div class="travello_content_inner">
<div></div>
<div></div>
</div>
</div>

<div class="travello_container">
<a href="#">
<div class="d-flex flex-column align-items-center justify-content-end">
<span class="travello_title">Get a 20% Discount</span>
<span class="travello_subtitle">Buy Your Vacation Online Now</span>
</div>
</a>
</div>
</div>
</div>

<footer class="footer">
<div class="parallax_background parallax-window" data-parallax="scroll" data-image-src="{% static 'images/footer_1.jpg' %}" data-speed="0.8">
</div>
<div class="container">
<div class="row">
<div class="col">
<div class="newsletter">
<div class="newsletter_title_container text-center">
<div class="newsletter_title">Subscribe to our newsletter to
get the latest
trends &
news</div>
<div class="newsletter_subtitle">Join our database NOW!
</div>
</div>
```

```

<div class="newsletter_form_container">
<form action="#">
  class="newsletter_form" d-flex flex-md-row flex-column align-items-start
  justify-content-between"
  id="newsletter_form">
    <div
      class="d-flex flex-md-row flex-column align-items-start justify-content-between">
      <div><input type="text"
        class="newsletter_input newsletter_input_name"
        id="newsletter_input_name" placeholder="Name"
        required="required">
        <div class="input_border"></div>
      </div>
      <div><input type="email"
        class="newsletter_input newsletter_input_email"
        id="newsletter_input_email" placeholder="Your e-mail"
        required="required">
        <div class="input_border"></div>
      </div>
    </div>
    <div><button class="newsletter_button">subscribe</button>
    </div>
  </form>
  <div class="row footer_contact_row">
    <div class="col-xl-10 offset-xl-1">
      <div class="row">
        <!-- Footer Contact Item -->
        <div class="col-xl-4 footer_contact_col">
          <div
            class="footer_contact_item d-flex flex-column align-items-center justify-content-start
            text-center">
            <div class="footer_contact_icon">
</div>

<div class="footer_contact_title">give us a call</div>
<div class="footer_contact_list">
<ul>
<li>Office Landline: +44 5567 32 664 567</li>
<li>Mobile: +44 5567 89 3322 332</li>
</ul>
</div>

<!-- Footer Contact Item -->
<div class="col-xl-4 footer_contact_col">
<div
class="footer_contact_item d-flex flex-column align-items-center justify-content-start
text-center">
<div class="footer_contact_icon">
</div>
<div class="footer_contact_title">come & drop by</div>
<div class="footer_contact_list">
<ul style="max-width:190px">
<li>4124 Barnes Street, Sanford, FL 32771</li>
</ul>
</div>
<div class="col-xl-4 footer_contact_col">
<div
class="footer_contact_item d-flex flex-column align-items-center justify-content-start
text-center">
<div class="footer_contact_icon">
</div>
<div class="footer_contact_title">send us a message
</div>

```

```

<div class="footer_contact_list">
<ul>
<li>youremail@gmail.com</li>
<li>Office@yourbusinessname.com</li>
</ul>
</div>

<div class="col text-center">
Copyright &copy;
<script>document.write(new Date().getFullYear());</script> All rights
reserved | This template
is made
with <i class="fa fa-heart-o" aria-hidden="true"></i> by <a href="https://colorlib.com"
target="_blank">Colorlib</a>
</footer>
</div>

<script src="{% static 'js/jquery-3.2.1.min.js' %}">
</script>

<script src="{% static 'styles/bootstrap4/popper.js' %}"></script>
<script src="{% static 'styles/bootstrap4/bootstrap.min.js' %}"></script>
<script src="{% static 'plugins/OwlCarousel2-2.2.1/owl.carousel.js' %}"></script>
<script src="{% static 'plugins/Isotope/isotope.pkgd.min.js' %}"></script>
<script src="{% static 'plugins/scrollTo/jquery.scrollTo.min.js' %}"></script>
<script src="{% static 'plugins/easing/easing.js' %}"></script>
<script src="{% static 'plugins/parallax-js-master/parallax.min.js' %}"></script>
<script src="{% static 'js/custom.js' %}"></script>
</body>
</html>

```

Register.html

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Sign Up Form by Colorlib</title>
        <link rel="stylesheet" href="%static%fonts/material-icon/css/material-design-iconic-font.min.css" %}>
</head>
<body>
    <section class="signup">
        <div class="container">
            <div class="signup-content">
                <div class="signup-form">
                    <h2 class="form-title">Sign up</h2>
                    <form action='register' method='POST' class="register-form" id="register-form">
                        {% csrf_token %}
                    <div class="form-group">
                        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
                        <input type="text" name="first_name" id="name" placeholder="First Name"/>
                    </div>
                    <div class="form-group">
                        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
                        <input type="text" name="last_name" id="name" placeholder="Last Name"/>
                    </div>
                    <div class="form-group">
                        <label for="name"><i class="zmdi zmdi-account material-icons-name"></i></label>
                        <input type="text" name="username" id="name" placeholder="User Name"/>
                    </div>
                    <div class="form-group">
                        <label for="email"><i class="zmdi zmdi-email"></i></label>
                        <input type="email" name="email" id="email" placeholder="Your Email"/>
                    </div>
                </div>
            </div>
        </div>
    </section>

```

```

</div>
<div class="form-group">
<label for="pass"><i class="zmdi zmdi-lock"></i></label>
<input type="password" name="pass" id="pass" placeholder="Password"/>
</div>
<div class="form-group">
<label for="re-pass"><i class="zmdi zmdi-lock-outline"></i></label>
<input type="password" name="re_pass" id="re_pass" placeholder="Repeat your password"/>
</div>
<div class="form-group">
<input type="checkbox" name="agree-term" id="agree-term" class="agree-term" />
<label for="agree-term" class="label-agree-term"><span><span></span></span>I agree all statements in <a href="#" class="term-service">Terms of service</a></label>
</div>
<div class="form-group form-button">
<input type="submit" name="signup" id="signup" class="form-submit" value="Register"/>
</div>
<center>
<div>
{%
for message in messages%}
<h3><font color="red">{{message}}</font></h3>
{%
endfor%}
</div>
</center>
</form>
</div>
<div class="signup-image">
<figure></figure>

```

```

<a href="" class="signup-image-link">I am already member</a>
</div>
</div>
</section>
<script src="vendor/jquery/jquery.min.js"></script>
<script src="js/main.js"></script>
</body>
</html>

```

Signin.html

```

<!DOCTYPE html>
{%- load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Sign Up Form</title>
        <link rel="stylesheet" href="{% static "fonts/material-icon/css/material-design-iconic-font.min.css" %}">
</head>
<body>
    <form action='signin' method = "post">
        {% csrf_token %}
        <div class="main">
            <!-- Sing in Form -->
            <section class="sign-in">
                <div class="container">
                    <div class="signin-content">
                        <div class="signin-image">
                            <figure></figure>

```

```

<a href="signup.html" class="signup-image-link"><b>Create an
account</b></a>
</div>

<div class="signin-form">
<h2 class="form-title">Sign in</h2>
<form method="POST" class="register-form" id="login-form">
<div class="form-group">
<label for="your_name"><i class="zmdi zmdi-account
material-icons-name"></i></label>
<input type="text" name="your_name" id="your_name"
placeholder="Your Name"/>
</div>
<div class="form-group">
<label for="your_pass"><i class="zmdi zmdi-lock"></i></label>
<input type="password" name="your_pass" id="your_pass"
placeholder="Password"/>
</div>
<div class="form-group">
<input type="checkbox" name="remember-me" id="remember-me"
class="agree-term" />
<label for="remember-me"
class="label-agree-term"><span><span></span></span>Remember me</label>
</div>
<div class="form-group form-button">
<input type="submit" name="signin" id="signin" class="form-submit"
value="Log in"/>
</div>
<center>
<div>
{%-for message in messages%}
<h3><font color="red">{ {message} }</font></h3>

```

```

{%-endfor%}
</div>
</center>
</form>
<div class="social-login">
<span class="social-label">Or login with</span>
<ul class="socials">
<li><a href="#"><i class="display-flex-center zmdi zmdi-facebook"></i></a></li>
<li><a href="#"><i class="display-flex-center zmdi zmdi-twitter"></i></a></li>
<li><a href="#"><i class="display-flex-center zmdi zmdi-google"></i></a></li>
</ul>
</div>
</div>
</div>
</div>
</section>
<script src="vendor/jquery/jquery.min.js"></script>
<script src="js/main.js"></script>
</form>
</body>
</html>

```

Pglist.html

```

{%- load static %}

{%- static "images" as baseUrl %}

<!DOCTYPE html>

<html lang="en">
<head>
<title>Travello</title>
<meta charset="utf-8">

```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="description" content="Travello template project">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="%static
'styles/bootstrap4/bootstrap.min.css' %}>
</head>
<body>
    <div class="dp">      <font color="#17202A" ><b>Hello,
    {{user.first_name}}<b></font></div>
    <div class="tp">      <font color="#00FF00" ><b><a
    href='..//reg/logout'>Logout</a></b></font></div>
        <font color="#17202A" ><b><script>
document.write(Date())
</script></b></font>
        <div class="container">z
        <form action='pglist' method='POST'>
            {%- csrf_token %}
            <div class="input-group">
                <input type="text" class="form-control" placeholder="Search" name="search">
                <div class="input-group-btn">
                    <button class="btn btn-default" type="submit" value="Search" >Search
                </button>
            </div>
        </div>
        </form>
        <div class="destinations" id="destinations">
            <div class="container">
                <div class="row destinations_row">
                    <div class="col">
                        <div class="destinations_container item_grid">
                            <form action="pglists" method="POST" id="myform">
                                {%-csrf_token%}

```

```

{%- for dest in dests %}

<!-- Destination -->

<div class="destination item">
<div class="destination_image">

{%- if dest.offer %}

<div class="spec_offer text-center"><a href="#">Special Offer</a></div>

{%- endif %}

</div>

<div class="destination_content">
<div class="destination_title">
<label ><b> <a href="javascript: void(0)" onclick="document.getElementById('myform').submit();">{{dest.name}}</a></b></label>
<input type="hidden" name="hotels" value="{{dest.name}}">
</div>

<div class="destination_subtitle">
<p>{{dest.location}}</p>
<label><b><a href="map">Directions</a></b></label>
</div>

<div class="destination_price">From ${{dest.price}}</div>
</div>
</div>

{%- endfor %}

</form>
</div>

<script src="{{ static 'plugins/parallax-js-master/parallax.min.js' }}></script>
<script src="{{ static 'js/custom.js' }}></script>
</body>
</html>

```

Map.html

```

{%- load static %}

<!DOCTYPE html>

```

```

<html>
<head>
<meta charset="utf-8" />
<title>Locate the user</title>
<meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
<script src="https://api.mapbox.com/mapbox-gl-js/v1.8.1/mapbox-gl.js"></script>
<link href="https://api.mapbox.com/mapbox-gl-js/v1.8.1/mapbox-gl.css" rel="stylesheet" />
<meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />
<script src="https://api.mapbox.com/mapbox-gl-js/v1.8.1/mapbox-gl.js"></script>
<link href="https://api.mapbox.com/mapbox-gl-js/v1.8.1/mapbox-gl.css" rel="stylesheet" />
<style>
body { margin: 0; padding: 0; }

#map { position: absolute; top: 0; bottom: 0; width: 100%; }

}
</style>
</head>
<body>
<script
src="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.0.2/mapbox-gl-directions.js"></script>
<link
rel="stylesheet"
href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.0.2/mapbox-gl-directions.css"
type="text/css"/><div id="map"></div>
<script>
    mapboxgl.accessToken = '{{ mapbox_access_token }}';
    mapboxgl.accessToken
    =
'pk.eyJ1Ijoic3JpZGhhcjAwMDciLCJhIjoiY2s3b2lra3N4MDVyNzNmbXNsZGZsY2htdiJ9.Q
GZpr-aeWNBms5QBmdHEUA';

    var map = new mapboxgl.Map({
        container: 'map',

```

```

        style: 'mapbox://styles/mapbox/streets-v11',
        center: [78.46667, 17.36667],
        zoom: 9
    });

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.3211, 17.3848])
        .setHTML('<h1><b>My hotel</b></h1>')
        .addTo(map);

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.4, 17.3])
        .setHTML('<h1><b>Taj Banjara</b></h1>')
        .addTo(map);

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.3169, 17.380799])
        .setHTML('<h1><b>Marriot</b></h1>')
        .addTo(map);

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.548487, 17.097637])
        .setHTML('<h1><b>Mango Suites</b></h1>')
        .addTo(map);

    var popup = new mapboxgl.Popup({ closeOnClick: false })
        .setLngLat([78.5, 17.45])
        .setHTML('<h1><b>Raddission</b></h1>')
        .addTo(map);

// Add geolocate control to the map.

    map.addControl(
        new mapboxgl.GeolocateControl({
            positionOptions: {
                enableHighAccuracy: true
            },
            trackUserLocation: true
        })
    );

```

```

);
map.addControl(new mapboxgl.NavigationControl());
    map.addControl(
new MapboxDirections({
accessToken: mapboxgl.accessToken
}),
'top-left'
);
    map.addControl(
new MapboxGeocoder({
accessToken: mapboxgl.accessToken,
mapboxgl: mapboxgl
})
);
var nullIsland = new mapboxgl.MercatorCoordinate(77,17, 0);
var popup = new mapboxgl.Popup({ closeOnClick: false })
.setLngLat([-96, 37.8])
.setHTML('<h1>Hello World!</h1>')
.addTo(map);
</script>
</body>
</html>

```

Pay.html

```

{%- load static %}

<html>
<head>
</head>
<body style="background-image:url('{{ static 'images/test3.jpg' }})">
<div class="view" >
<div class="container">
<div class="jumbotron" >

```

```

<form class="form-horizontal" action="pay" method="POST" >
{%- csrf_token %}
<div class="form-group">
<label class="control-label col-sm-2" for="email"><b>First_Name:</b></label>
<div class="col-sm-10">
    <input type="text" class="form-control" placeholder="Enter First_Name"
name="fname">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2" for="pwd"><b>Last_Name:</b></label>
<div class="col-sm-10">
    <input type="text" class="form-control" placeholder="Enter Last_Name"
name="lname">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2" for="pwd"><b>Email_Id:</b></label>
<div class="col-sm-10">
    <input type="email" class="form-control" id="email" placeholder="Enter Email_Id"
name="email">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2" for="pwd"><b>Phone_Number:</b></label>
<div class="col-sm-10">
    <input type="tel" class="form-control" placeholder="Enter Phone Number"
name="pno">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2" for="pwd"><b>Room_Type:</b></label>

```

```

<div class="col-sm-10">
  <div class="radio">
    <label><input type="radio" name="radio" value='Single'>Single</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="radio" value='Double'>Double</label>
  </div>
  <div class="radio ">
    <label><input type="radio" name="radio" value='Triple'>Triple</label>
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="pwd" ><b>Guests:</b></label>
  <div class="col-sm-10">
    <div class="radio">
      <label><input type="radio" name="optradio" value='1(one)'>1(one)</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="optradio" value='2(Two)'>2(Two)</label>
    </div>
    <div class="radio ">
      <label><input type="radio" name="optradio" value='2(Two)-4(Four)'>2(Two)-4(Four)</label>
    </div>
  </div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="pwd"><b>Arrival_Date:</b></label>
  <div class="col-sm-10">
    <input type="date" id="birthday" class="form-control" name="arrival">
  </div>

```

```

</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Time:</b></label>
    <div class="col-sm-10">
        <input type="time" id="appt" class="form-control" name="time">
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Departure_Date:</b></label>
    <div class="col-sm-10">
        <input type="date" id="birthday" class="form-control" name="departure">
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" for="pwd"><b>Gender:</b></label>
    <div class="col-sm-10">
        <div class="radio">
            <label><input type="radio" name="gender" value='Male'>Male</label>
        </div>
        <div class="radio">
            <label><input type="radio" name="gender" value='Female'>Female</label>
        </div>
        <div class="radio ">
            <label><input type="radio" name="gender" value='Others'>Others</label>
        </div>
    </div>
</div>
<div class="form-group">
    <label class="control-label col-sm-2" ><b>Selected_Hotel:</b></label>
    <div class="col-sm-10">
        <input type="text" class="form-control" id="" name="hotelname" value="{{ hotel }}>

```

```

    </div>
</div>
<div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
        <div class="checkbox">
            <label><input type="checkbox" name="remember"> Remember me</label>
        </div>
    </div>
</div>
<div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
        <center>      <button type="submit" class="btn btn-success
btn-lg">Submit</button></center>
    </div>
</div>
</form>
</div>
</div>
</div>
</body>
</html>

```

Razorpay.html

```

<div class="razorpay-embed-btn" data-url="https://rzp.io/l/ny716WS" data-text="Pay Now"
data-color="#528FF0" data-size="large">
<script>
(function(){
    var d=document; var x=!d.getElementById('razorpay-embed-btn-js')
    if(x){ var s=d.createElement('script'); s.defer=!0;s.id='razorpay-embed-btn-js';
s.src='https://cdn.razorpay.com/static/embed_btn/bundle.js';d.body.appendChild(s); } else {var
rzp=window['__rzp__'];
rzp && rzp.init && rzp.init(){}});

```

```
</script>
```

```
</div>
```

4.4 Data Set description

A data set (or dataset) is a collection of data. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

The database of the project consists of Project data. The program draws from both of these at different stages of the program repeatedly, and also stores data in the appropriate file.

The username and password of a user are stored in a .dat file. This file is read from at the registration stage to ensure that a particular username is not already taken, and is written to when a user registers successfully. The username and password are stored in string form in the file. When a user tries to login by entering their username and password, the program reads from this file to verify that the username and password match. Thus, this file enables user authentication.

The locations are stored in the views of the python and it is stored in a .py file and from there we retrieve the data.

4.5 Testing Process

The test cases were designed keeping in mind where the program is liable to fail and throw errors or give erroneous outputs. The following cases are some of the important scenarios to be kept in mind.

4.5.1 Registration with an existing username

If the user tries to register with a username that already belongs to another user, the following message appears:

Sign up

First Name

Last Name

sridhar

Your Email

.....
[I am already member](#)

Repeat your password

I agree all statements in [Terms of service](#)



Username Taken

**Username
Already taken**

4.5.2 Wrong login credentials

If the username and password don't match, or no such username has been registered, the following message appears:



[Create an account](#)

Sign in

sridharabc

.....

Remember me

[Log in](#)

**Invalid
Credentials**

invalid credentials

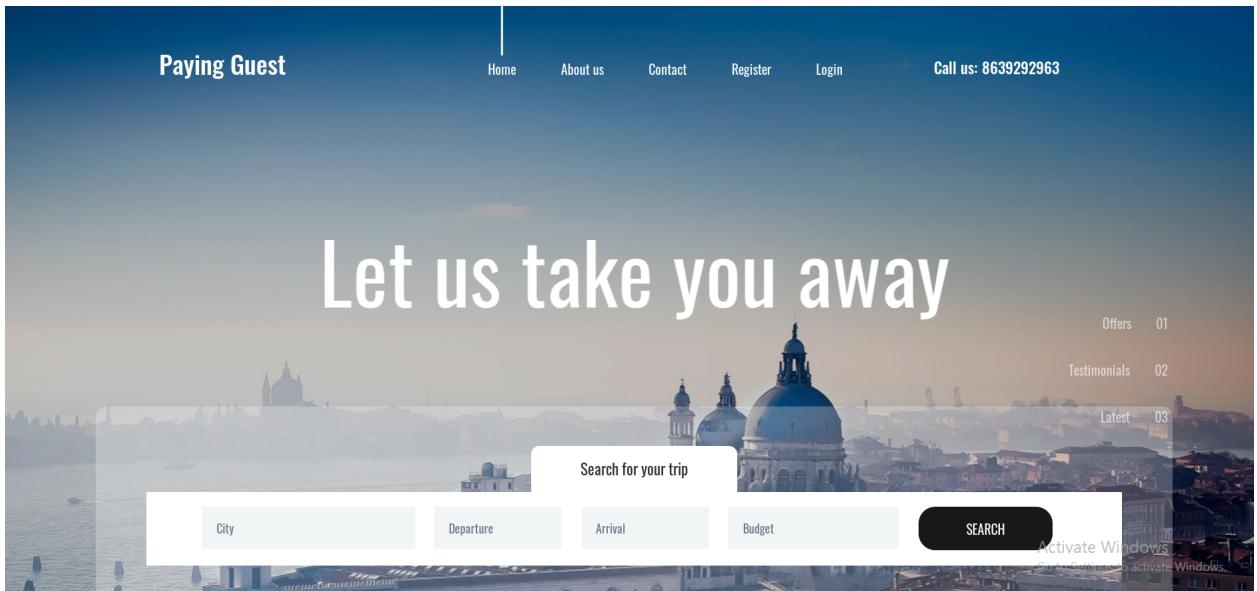
Or login with



The program was validated by running a test case with some correct answers and wrong answers, as well as both wrong and correct login credentials. Care was taken to include all possible exceptional cases and other problematic scenarios in the input.

5. RESULTS / OUTPUTS AND DISCUSSIONS

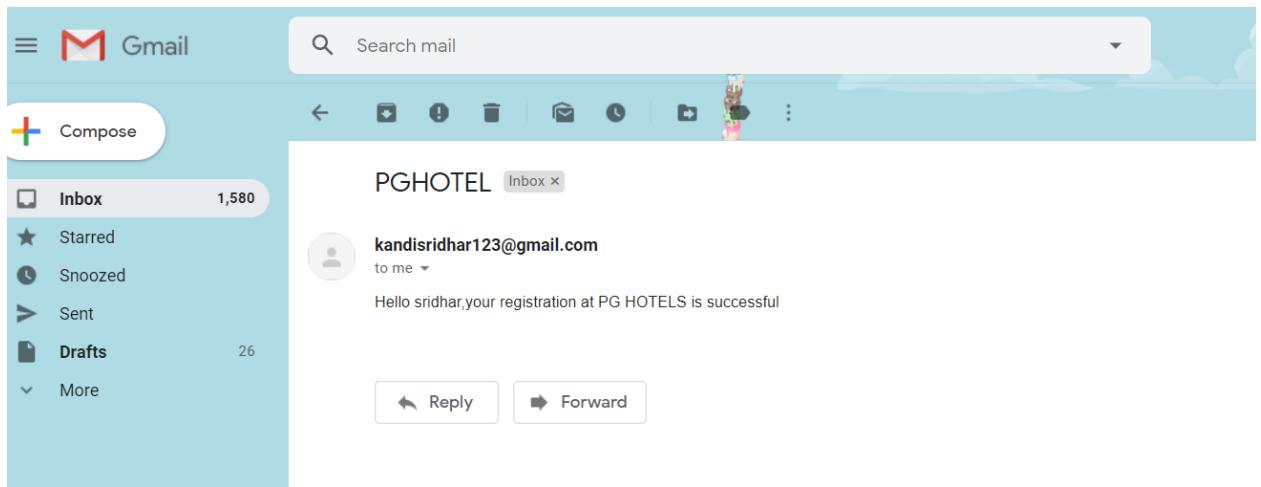
The Home page looks as follows:



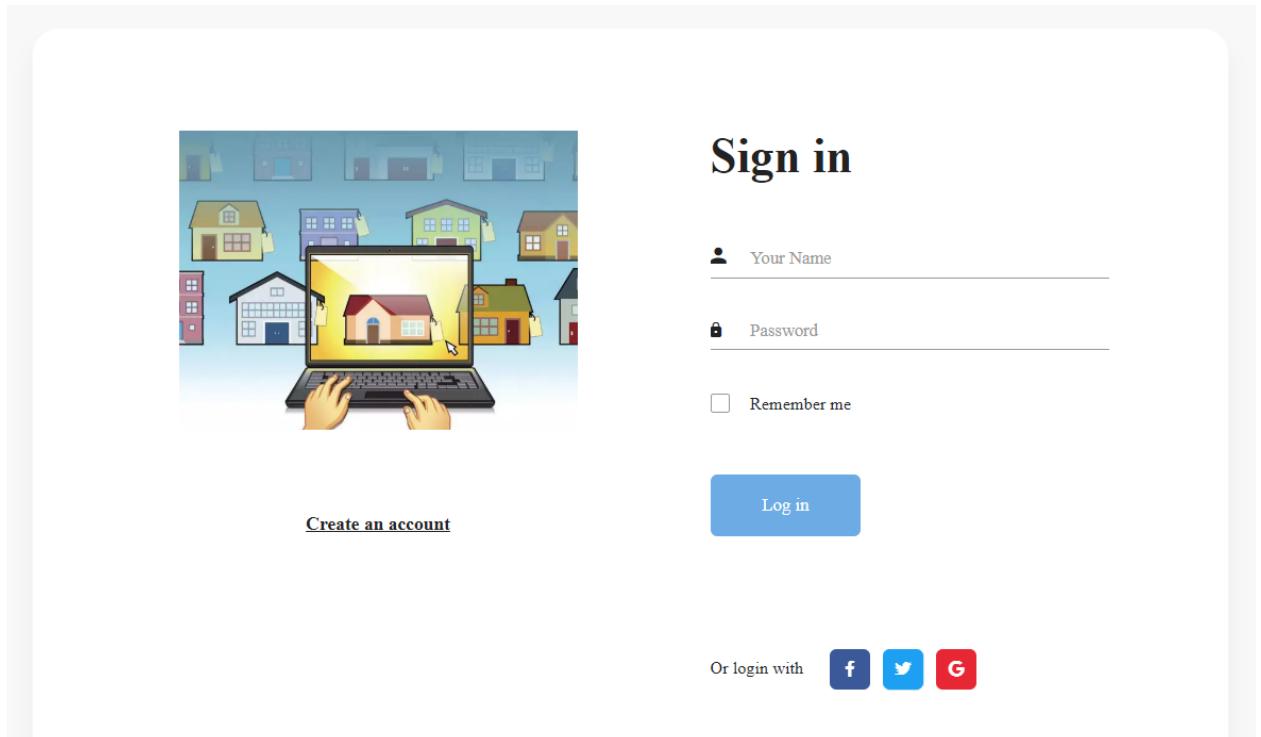
On clicking register, the screen changes to:

A screenshot of the "Sign up" registration form. On the left, there are input fields for "First Name" (with placeholder "sridhar"), "Last Name" (empty), "Your Email" (empty), "Password" (empty), and "Repeat your password" (empty). Below these is a checkbox for agreeing to terms of service. On the right, there is a large green circular logo containing a stylized house with a person walking out of it, and the text "Paying Guest" below it. At the bottom right of the form area, there is a link "I am already member". At the bottom center is a blue "Register" button.

On successful registration, the user will receive a mail.



Enter your credentials to log in.



If login is successful, it shows a list of hotels:

Search

Search



Marriot

gandipet

[Directions](#)

From \$500



Mango suites

gachibowli

[Directions](#)

From \$450



Novotel

manikonda

[Directions](#)

From \$750

Activate Windows
Go to Settings to activate Windows.



Taj Hotel

abids

[Directions](#)

From \$800



Radission

gandipet

[Directions](#)

From \$550

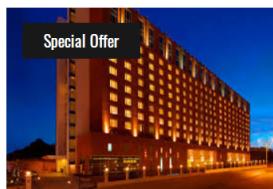


Taj Krishna

alwal

[Directions](#)

From \$800



sheraton

hayathnagar

[Directions](#)

From \$650



siesta

madhapur

[Directions](#)

Activate Windo
Go to Settings to acti

Search bar to search for hotels in different locations.

Z



Marriot

gandipet

[Directions](#)

From \$500



Radission

gandipet

[Directions](#)

From \$550

Activate Wind
Go to Settings to ac

Django administration to add hotels from the admin side.

Django administration

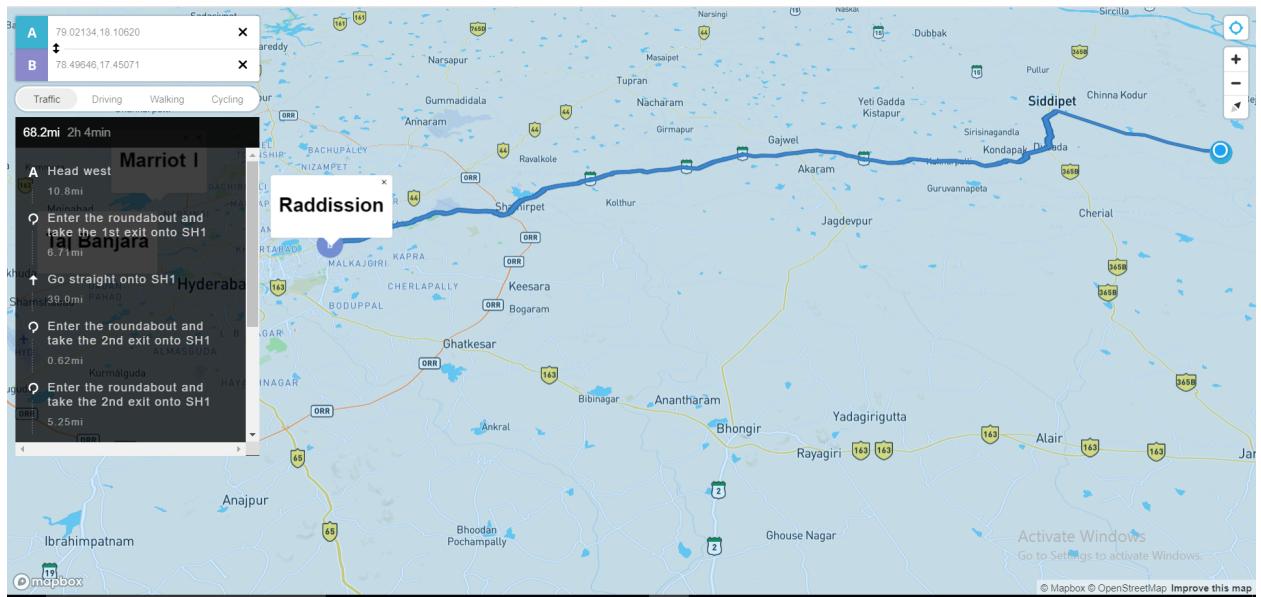
WELCOME, SRIDHAR. [VIEW SITE / CHANGE PASSWORD / LOG OUT](#)

Home › List › Pghotels › Add pghotels

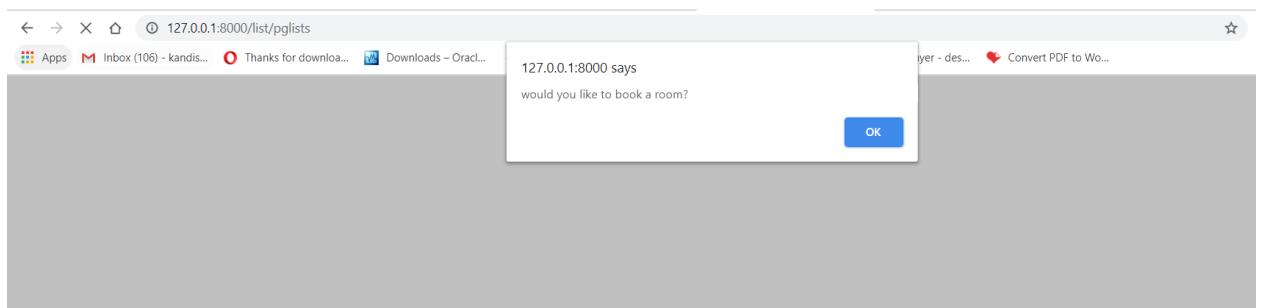
Add pghotels

Name:	<input type="text"/>
Img:	<input type="file"/> Choose File No file chosen
Price:	<input type="text"/>
<input type="checkbox"/> Offer	
Location:	<input type="text"/>

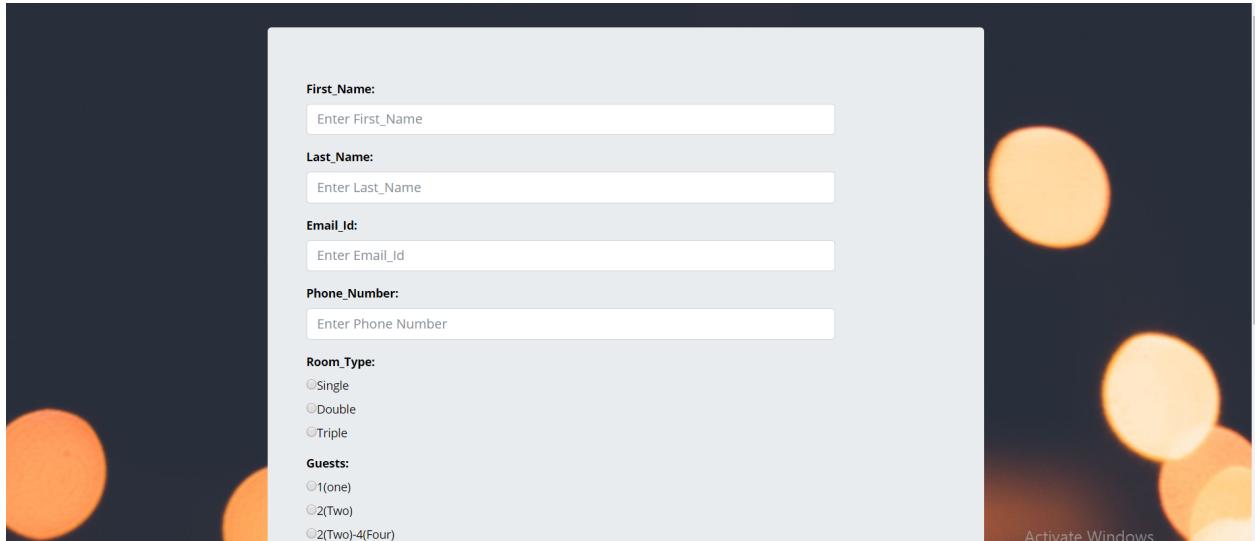
Map to navigate from source location to the destination hotel.



Ask the user, would he/she likes to book a room.



Ask the user to enter his/her personal details.



First_Name:

Last_Name:

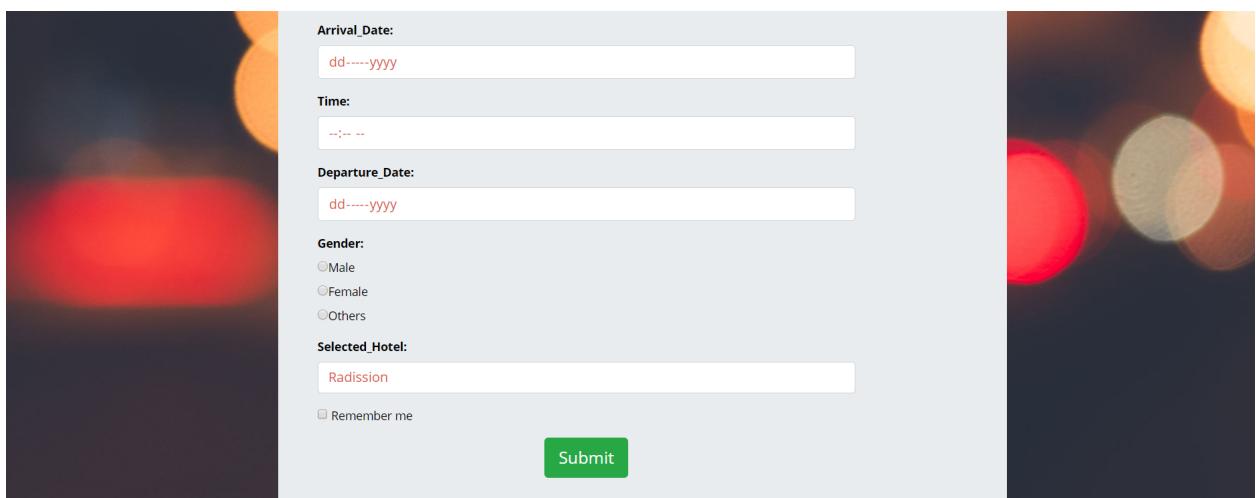
Email_Id:

Phone_Number:

Room_Type:
 Single
 Double
 Triple

Guests:
 1(one)
 2(Two)
 2(Two)-4(Four)

Activate Windows



Arrival_Date:

Time:

Departure_Date:

Gender:
 Male
 Female
 Others

Selected_Hotel:

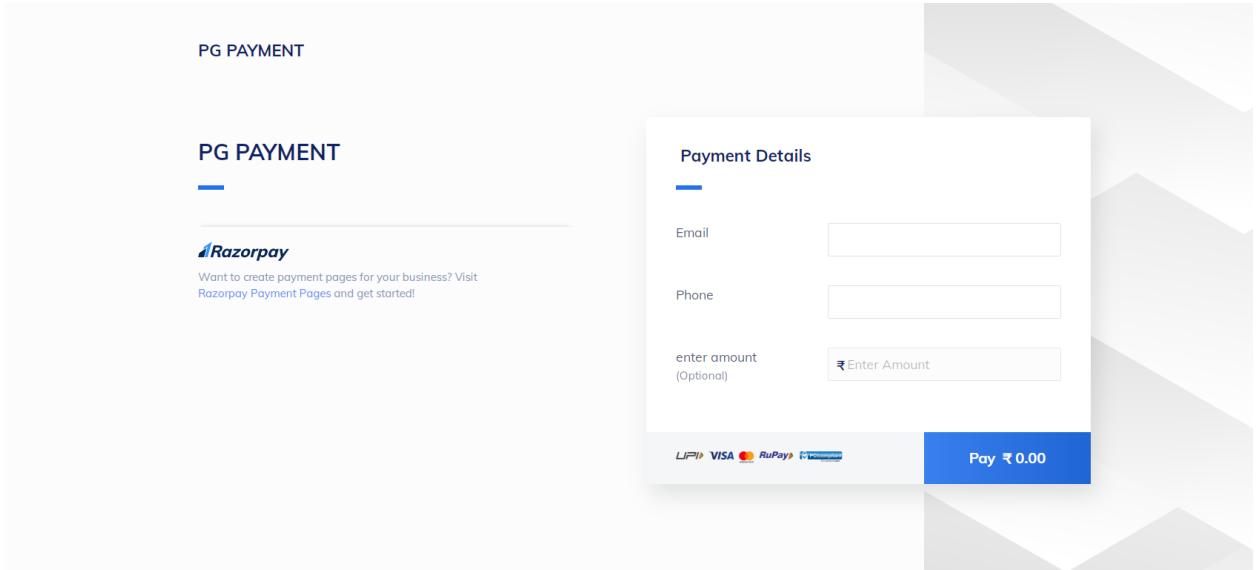
Remember me

Submit

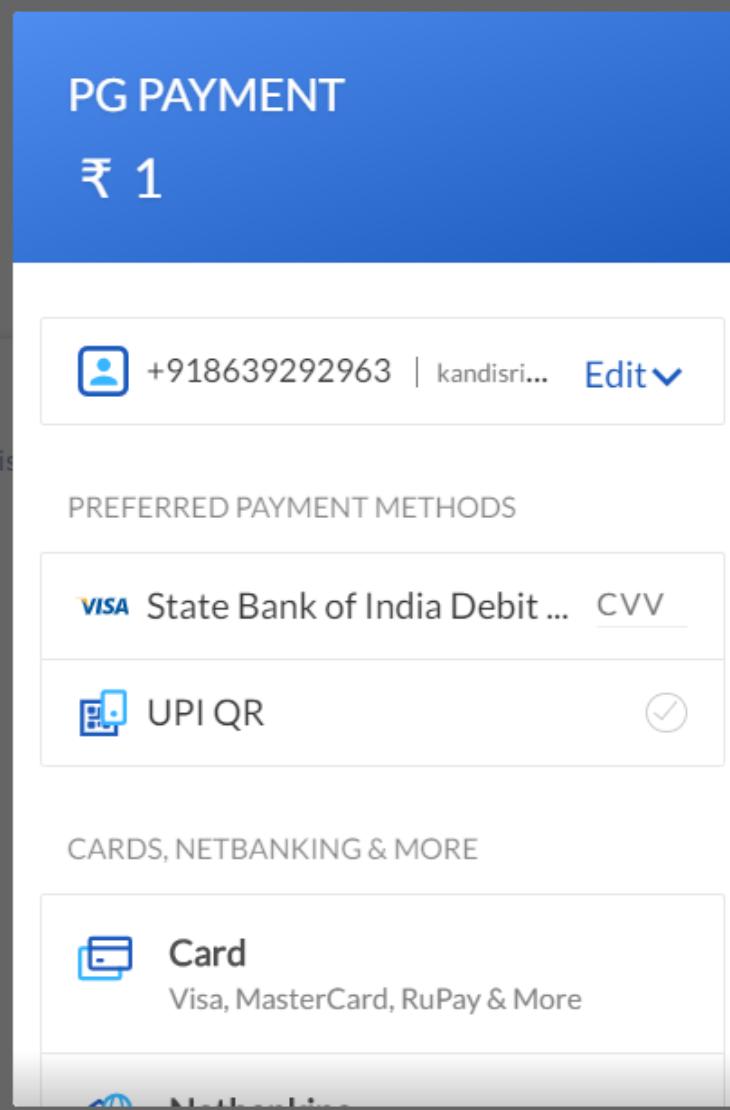
Click on Pay Now to proceed to the Payment Gateway.

Pay Now

Powered by 



Different payment methods.





Merchant : PG PAYMENT
Transaction Amount : ₹ 1.00
SBI Debit Card : 4591xxxx xxxx xx52

Authenticate Payment

OTP sent to your mobile number ending **99xx7xxx95**

Enter One Time Password (OTP)

Make Payment

[Resend OTP](#)

[Click here to abort transaction and go back to merchant site](#)



PCI DSS Certified

Important - To activate your SBI debit card for online transactions (if you are a new SBI customer or haven't transacted online after June 01, 2015), please send an sms **SWON<space>ECOM<space>XXXX** to **09223966666** (where XXXX is the last 4 digits of your SBI Debit Card) from your registered mobile number.

If payment is successful, it shows the confirmation message.

PG PAYMENT

PG PAYMENT



Want to create payment pages for your business? Visit [Razorpay Payment Pages](#) and get started!



₹ 1.00

Payment received successfully

A confirmation email has been mailed to you
Payment ID: pay_EgKVyQ1SAze2rw

Email showing the successful payment.

The screenshot shows a Gmail inbox with 1,579 messages. A single email is selected, titled "Payment successful for PG PAYMENT". The email is from "Team Razorpay <no-reply@razorpay.com> to me" at 5:07 PM (2 minutes ago). The message body includes a payment summary and detailed transaction information. At the bottom, there is a yellow box with a COVID-19 prevention message and links to settings.

Payment Summary:

- Powered by **PG PAYMENT**
- ₹1.00**
- Paid Successfully**

Transaction Details:

Payment Id	pay_EgKVyQ1SAze2rw
Paid On	19th Apr, 2020
Method	XXXX-XXXX-XXXX-5652 Card
Mobile Number	+918639292963

COVID-19 Prevention Message:

Protect yourself and your loved ones from Coronavirus disease (COVID-19) by following [these instructions](#) from WHO

Activation Links:

- Activate W
- Go to Settings

CHAPTER-6

CONCLUSION

6.1 Conclusion :

Web applications abridge the gap between the user and the hosts . This integrates basic amenities for the users in one platform. It further extends the feature of advertising the host's accommodation facilities over the website application. All in all this application will turn out to be a boon for all the students by providing them with a portable all-in-one application. There are many conclusive features on the website which suggests there can be further development and an outlook can be created for a business perspective using various hosting platforms. The integration of these platforms can make a subtle environment where a user can incur fewer data and also save time. The portable and easy approach of the website will lead to the future development of the website. This website culminates all the day-to-day issues which can be taken up as an idea for business perception. In the long run, this website can serve as a potential business supplement. Since the communication between the client and the host is subjunctive and very wide open this may lead to data transparency. This application not only serves the issues but also looks after the choices or preferences of the clients using this application in a new location. From the managerial point of view, it is rightly said: "Time is Money". Finally, to wrap the content of the report, it suggests a method to reduce the time and data of an individual which serves as an asset to the being.

6.1.1 Limitations:

- This system is based mainly on the PG booking it doesn't contain any other essential things that any other day to day tasks of a person. It may not provide results on the things you liked in the order according to the recent searches. It cannot show you anything except the PG directions in the

maps, maps are used only to track your location and PG location only. It will show you all the important intermediate stops but not everything.

- It is limited to only Hyderabad city, it will be integrated further.
- Users cannot book a PG without registering.
- PG owner validation is not done in this project.

6.2 FUTURE ENHANCEMENTS:

- Online chat with multiple website users.
- Implement google translator that can be used for communication in different languages.
- Integrate the analysis models to web applications.
- Make a model predict search patterns based on previous data.
- Make the data cleaner and get better and accurate results at the end

REFERENCES:

- <https://docs.djangoproject.com/en/3.0/>
- <https://www.tutorialspoint.com/django/>
- <https://www.postgresqltutorial.com/>
- <https://www.w3schools.com/css/default.asp>
- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://www.w3schools.com/js/default.asp>
- <https://docs.python.org/3/tutorial/>
- <https://www.youtube.com/watch?v=SIyxjRJ8VNY&list=PLsyeobzWxl7r2ukVgTqIQcl-1T0C2mzau>
- <https://getbootstrap.com/>
- <https://www.mapbox.com/>
- <https://razorpay.com/>