**A**

# PROJECT REPORT

On

## TRAVEL ITINERARY PLANNER

Submitted in partial fulfillment of the requirements for the award of

## DIPLOMAIN
## COMPUTER ENGINEERING

**By**


| | |
|---|---|
| **P.THRISHA** | **21001-CS-001** |
| **RAHEEM** | **21001-CS-029** |
| **V.KEERTHANA** | **21001-CS-037** |
| **N.SAI CHARAN** | **21001-CS-040** |
| **SK.JUVERIA** | **21001-CS-041** |

**DEPARTMENT OF COMPUTER ENGINEERING**
**GOVERNMENT POLYTECHNIC MASABTANK**
HYDERABAD-500028
Telangana


**DEPARTMENT OF COMPUTER ENGINEERING**

CERTIFICATE

This is to certify that the Project report entitled "TRAVEL ITINERARY PLANNER" being submitted by

P.THRISHA 21001-CS-001

SK.RAHEEM 21001-CS-029

V.KEERTHANA 21001-CS-037 N.SAI CHARAN 21001-CS-040 SK.JUVERIA 21001-CS-041

In partial fulfillment of the requirements for the award of **Diploma in Computer Engineering**, State Board of Technical Education &Training arecord of bonafide work carried out by them.

The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other Board for the award of diploma.

**Internal Guide**
**(Khaja Habeebuddin)**

**Head of the Department**
**(J. Govardhan Reddy )**

**Submitted for the viva voice examination held on** _____

**Internal Examiner**

**External Examiner**

## ACKNOWLEDGEMENT

It is a matter of pleasure of acknowledgement by indebtedness to our teachers and also our well-wishers who have helped us for completion ofthis project successfully.

We wish to express our sincere gratitude to **Dr N.RAJESHWARI DEVI Ph.D.** Principal for her consistent help and encouragement to complete this project.

We express our deep sense of gratitude to our respected **Dr J.GOVARDHAN REDDY** Sir, In charge Head of the Department of Computer Engineering. Government Polytechnic who helped us in every step of our project work despite of his busy schedules.

We are most thankful to our guide **KHAJA HABEEBUDDIN** Sir, Lecture of Department of Computer Engineering, Government Polytechnic for his valuable guidance and suggestions in analysing and testing throughout the period until the end of our project work.

We want to express our deepest thankfulness to all the individuals and faculty, who helped us a lot in finalizing this project.

Finally, words are not sufficient to express our guidance to all our family and friends for helping us directly or indirectly in completion of the

# INDEX

# ABSTRACT

## Travel Itinerary Planner

The proposed travel itinerary planner is a comprehensive system designed to streamline the travel planning process. The platform features distinct interfaces for clients and employees, ensuring a user-friendly experience. Clients can easily input their travel preferences, access hotel and transport booking options, and utilize a helpdesk for any assistance needed. The system also incorporates a feedback mechanism to gather valuable insights. On the employee side, the interface facilitates efficient management of client requests, hotel and transport bookings, and prompt responses to helpdesk queries. This integrated solution aims to enhance the overall travel planning experience by providing a seamless and user-centric approach.

**Overview:**

The travel itinerary planner is a streamlined solution offering clients and employees userfriendly interfaces for efficient travel planning. Clients can input preferences, access hotel and transport bookings, and utilize a helpdesk, while employees manage requests seamlessly. The system includes a feedback mechanism, enhancing the overall user experience.

**Proposed System**:

Our proposed travel itinerary planner offers a centralized solution with user-friendly interfaces, efficient booking processes, a responsive helpdesk, and structured feedback mechanisms, aiming to optimize the overall travel planning experience.

**Software Requirements:**

1. Python (Django version—5.0.2)
2. Mysql (version-8.1.26)
3. html
4. css
5. javascript

**Hardware Requirements:**

Dual-core or Higher processor, RAM with minimum of 4GB, storage of minimum 512GB SSD for faster access etc..

# SOFTWARE REQUIREMENT SPECIFICATIONS

**User_registration():**

**Input Parameters:**

Username: The chosen username for the user.

Password: the user's chosen password.

Email: The user's email address.

**Output Parameters:**

User registration confirmation.

**Description**: In this function the has to register with their details(username, password, email).

**User_login():**

**Input Parameters:**

Username: The user's username.

Password: The user's Password.

**Output Parameters:**

User authentication status and user details.

**Description**: In this function implements the authenticate user login and return status and user details.

**Create_itinerary():**

**Input Parameters:**

User_id: The unique identifier of the user creating itinerary.

Destination: The destination of the travel itinerary.

**Output Parameters:**

Itinerary creation confirmation.

**Description**: In this function implements new itinerary plan

**view_itineraries():**

**Input Parameters:**

User_id: the unique identifier of the user.

**Output Parameter:**

Created travel itineraries.

**Description**: In this function displays user travel itineraries list.

**View_hotel_bookings():**

**Input Parameters**: None.

**Output Parameter**: None.

**Description**: This function allows to do hotel bookings.

**View_transport_bookings():**

**Input Parameters**: None.

**Output Parameters**: None.

**Description**: This function allows to do transport bookings.

**add_employee(username, password, role):**

**Input Parameters:**

username(str): The username of the new employee.

password (str): The password of the new employee.

role (str): The role of the new employee (e.g., employee, admin).

**Output Parameters**: None

**Description:**

Adds a new employee to the database with the provided username, password, and role.

**authenticate(username, password)` Function:**

**Input Parameters:**

username (str): The username of the employee attempting to log in.

password (str): The password of the employee attempting to log in.

**Output Parameter:**

A dictionary containing user details if authentication is successful, else None.

**Description**:

Authenticates the employee by checking the provided username and password against the Database.

**Book_hotels()**

**Input_Parameters:**

Check-in_date: the desired check_in_date for hotel stay.

Check-out_date: the desired check_out date

No_of_rooms: no.of rooms required

Price: the actual cost of the room for particular day

Location: select the destination of the client booking

**Output Parameters:**list of available hotels with details like name, price, amenities..etc

**Description:**

This function displays details about hotel rooms.

**Online_payment():**

**Input Parameters:**

Type_of_card: payment using either with debit or credit card

**Output Parameters**:Displays to select either credit or debit card and also select the payment method.

**Description:**

This function describes the online payment methods.

**modify_booking():**

**Input Parameters:**

Booking_id: id for the hotel booked.

New_check_in_date: The desired new check_in_date.

New_check_out_date: the desired new check_out_date

**Output Parameters:**

the new check_in and check_out details.

**Description**:

This function displays new modified details.

**Cancel_booking():**

**Input parameters:**

Booking_id: tells about the booking details

**Output parameters:**

Cancellation confirmation details.

**Description**:

This function cancels the hotel bookings done.

**Search_transport():**

**Input Parameters:**

Origin: The starting location for transportation.

Destination: The destination for transportation.

**Output Parameters:**

List of available transportation options with details(flight details, train schedules, bus option).

**Description:**

It is a function which has details about flight, train , bus options.

**Book_transport():**

**Input_Parameters:**

Travel_date: the desired date to travel to destination.

Price: the cost of each ticket according to the destination.

**Output Parameters:**

Booking confirmation details.

**Description:** This function displays details about transportation.

**Online_payment():**

**Input Parameters:**

Type_of_card: payment using either with debit or credit card

**Output Parameters**: Displays to select either credit or debit card and also select the payment method.

**Description**: This function describes the online payment methods.

**modify_booking():**

**Input Parameters:**

Booking_id: id for the selected transportation option booked.

New_Travel_date: The desired new_travel_date.

**Output Parameters:**

the new_travel_date details.

**Description:**This function displays new modified details.

**Cancel_booking():**

**Input parameters:**

Booking_id: tells about the booking details

**Output parameters:**

Cancellation confirmation details.

**Description**:This function cancels the transport bookings done.

**Viewfeedback**()

**Input Parameters:** feedback text

**Output Parameter:** thank you box will be displayed

**Description:** Allows users to submit their feedback and rating through the website. It collects user-provided

feedback text along with a numerical rating for the website's services and stores this data in the website

owner's database.

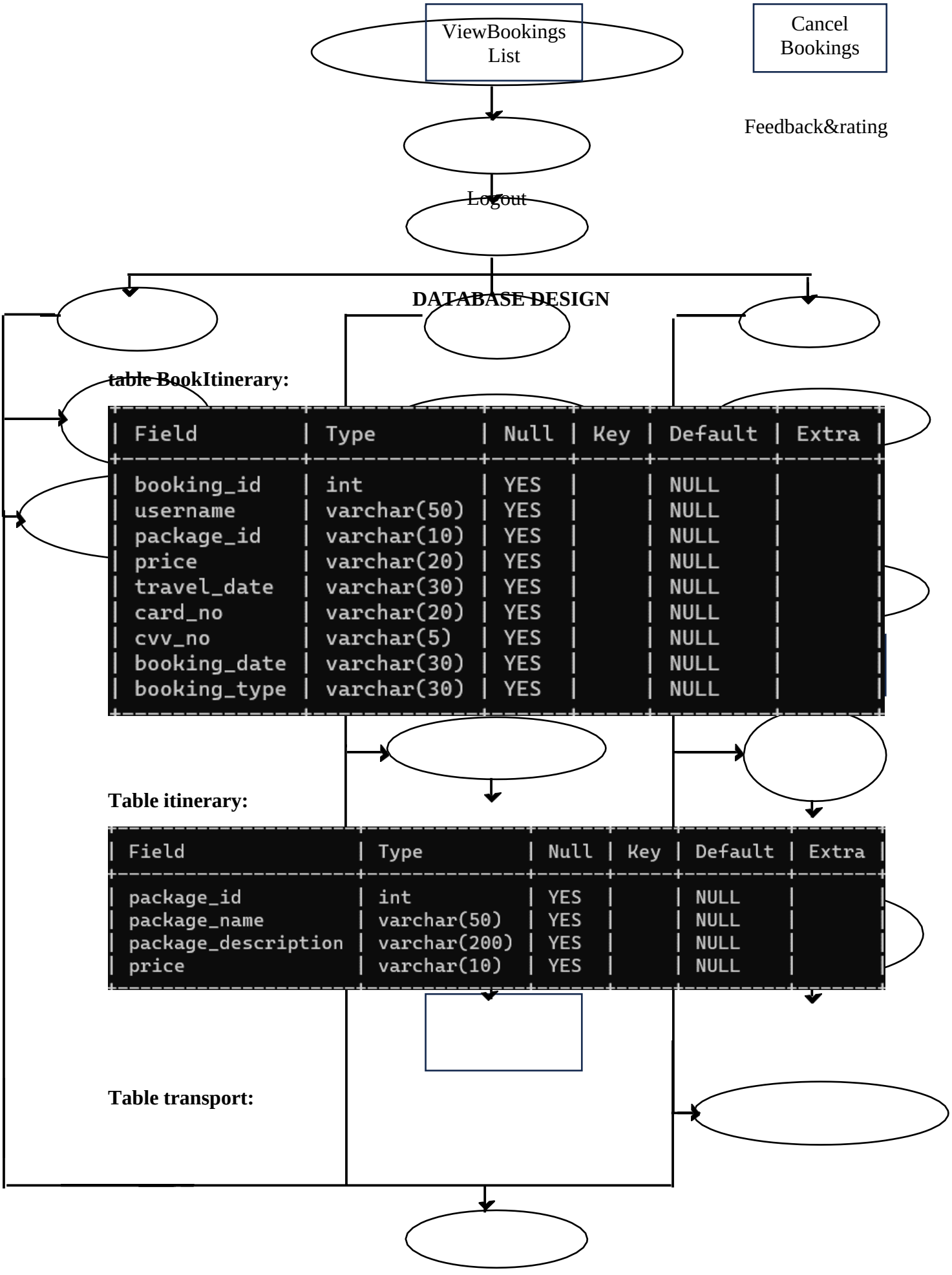
**SYSTEM DESIGN**

Travel Itinerary Planner

Home

Login

| Admin | Employe | User |
| --- | --- | --- |
| Add Employe | Create | Search |
| ViewAuthenticated EmployeeAction | View/modify itinerary | Book itinerary |
| view Employees | Add hotels | Search Hotels |
| | View/modify Hotels | Book Hotels |
| | Add Transport | Search Transpo |
| | View/modify Transport | Book Transport |
| | ViewBookings | Cancel Booking |




**DATABASE DESIGN**

**table BookItinerary:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| booking_id | int | YES | | NULL | |
| username | varchar(50) | YES | | NULL | |
| package_id | varchar(10) | YES | | NULL | |
| price | varchar(20) | YES | | NULL | |
| travel_date | varchar(30) | YES | | NULL | |
| card_no | varchar(20) | YES | | NULL | |
| cvv_no | varchar(5) | YES | | NULL | |
| booking_date | varchar(30) | YES | | NULL | |
| booking_type | varchar(30) | YES | | NULL | |

**Table itinerary:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| package_id | int | YES | | NULL | |
| package_name | varchar(50) | YES | | NULL | |
| package_description | varchar(200) | YES | | NULL | |
| price | varchar(10) | YES | | NULL | |

**Table transport:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| transport_id | int | NO | PRI | NULL | |
| service_name | varchar(50) | YES | | NULL | |
| source_location | varchar(50) | YES | | NULL | |
| destination_location | varchar(150) | YES | | NULL | |
| departure_time | varchar(150) | YES | | NULL | |
| expected_reached_time | varchar(40) | YES | | NULL | |
| price | varchar(40) | YES | | NULL | |

**Table hotel:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| hotel_id | int | YES | | NULL | |
| hotel_name | varchar(50) | YES | | NULL | |
| room_price | varchar(50) | YES | | NULL | |
| location | varchar(100) | YES | | NULL | |
| service_description | varchar(50) | YES | | NULL | |

**Table employee:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| username | varchar(50) | NO | PRI | NULL | |
| password | varchar(50) | YES | | NULL | |
| phone_no | varchar(50) | YES | | NULL | |
| email | varchar(50) | YES | | NULL | |
| address | varchar(50) | YES | | NULL | |

**Table notification:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| employee_name | varchar(50) | YES | | NULL | |
| login_date | varchar(40) | YES | | NULL | |
| login_status | varchar(30) | YES | | NULL | |

**Table user_signup:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| username | varchar(50) | NO | PRI | NULL | |
| password | varchar(50) | YES | | NULL | |
| phone_no | varchar(50) | YES | | NULL | |
| email | varchar(50) | YES | | NULL | |
| address | varchar(50) | YES | | NULL | |

**Table feedback:**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| username | varchar(50) | YES | | NULL | |
| feedback | varchar(120) | YES | | NULL | |
| ratings | varchar(10) | YES | | NULL | |
| feedback_date | varchar(30) | YES | | NULL | |

# TECHNOLOGIES USED

## HTML:

The **HyperText Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It defines the meaning and structure of web content. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

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

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items.

## CSS:

Cascading Style Sheets (CSS) is a language used to illustrate a document's look, style, and format in any markup language. In simple words, it is used to style and organize the layout of Web pages. CSS3 is the latest version of an earlier CSS version, CSS2.

A significant change in CSS3 compared to CSS2 is the introduction of modules. This functionality has the advantage of expediting the finalization and acceptance of specifications, as segments can be finalized and accepted incrementally. Also, this allows the browser to support segments of the specification

## JAVASCRIPT:

JavaScript is a versatile programming language that is primarily used for adding interactivity and dynamic behaviour to web pages. It's a client-side scripting language, meaning it runs in the user's web browser rather than on the server, enabling real-time updates and modifications to web content.

JavaScript is mainly employed for client-side scripting, allowing developers to create

dynamic and interactive web pages. It runs directly in the user's browser, providing a responsive user experience without the need for constant communication with the server

**PYTHON:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**DJANGO:**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites and web applications. It follows the principle of DRY (Don't Repeat Yourself) and emphasizes the convention over configuration paradigm, which means it provides default configurations and structures, reducing the need for developers to make repetitive decisions.

Django's batteries-included approach, extensive documentation, and vibrant community make it a popular choice for building web applications of any size or complexity.

**MYSQL:**

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and storing data in various types of applications.

MySQL is a powerful and reliable database management system that is suitable for a wide range of applications and use cases. Its combination of performance, scalability, and ease of use has made it one of the most popular databases in the world.

## CODING

**VIEWS.PY**

```
from django.shortcuts import render
import pymysql
from datetime import datetime
from django.template import RequestContext
from django.contrib import messages
import pymysql
from django.http import HttpResponse
from django.core.files.storage import FileSystemStorage
import os


global username


def CancelBookingAction(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "delete from BookItinerary where booking_id='"+pid+"'"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
```

```
            data = "Selected Booking Deleted : "+pid
            context= {'data':data}
            return render(request,'UserScreen.html', context)
        else:
            data = "Error in cancelling booking"
            context= {'data':data}
            return render(request,'UserScreen.html', context)


def CancelBookings(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Booking ID</font></th>'
        output+='<th><font size=3 color=black>Username</font></th>'
        output+='<th><font size=3 color=black>Package ID</font></th>'
        output+='<th><font size=3 color=black>Price</th>'
        output+='<th><font size=3 color=black>Travel Date/Hotel Occupying Date</th>'
        output+='<th><font size=3 color=black>Card No</th>'
        output+='<th><font size=3 color=black>CVV No</th>'
        output+='<th><font size=3 color=black>Booking Date</th>'
        output+='<th><font size=3 color=black>Booking Type</th>'
        output+='<th><font size=3 color=black>Cancel Booking</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
```

```
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from BookItinerary")
            lists = result.fetchall()
            for ls in lists:
                if ls[1] == username:
                    output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[5]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[6]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[7]+'</font></td>'
                    output+='<td><font size=3 color=black>'+ls[8]+'</font></td>'
                    output+='<td><a href=\'CancelBookingAction?pid='+str(ls[0])+'\'><font
size=3 color=black>Click Here to Cancel</font></a></td></tr>'
        context= {'data':output}
        return render(request,'UserScreen.html', context)


def  ViewBookings(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
```

```
output+='<tr><th><font size=3 color=black>Booking ID</font></th>'

output+='<th><font size=3 color=black>Username</font></th>'

output+='<th><font size=3 color=black>Package ID</font></th>'

output+='<th><font size=3 color=black>Price</th>'

output+='<th><font size=3 color=black>Travel Date/Hotel Occupying Date</th>'

output+='<th><font size=3 color=black>Card No</th>'

output+='<th><font size=3 color=black>CVV No</th>'

output+='<th><font size=3 color=black>Booking Date</th>'

output+='<th><font size=3 color=black>Booking Type</th></tr>'

mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')

with mysqlConnect:

    result = mysqlConnect.cursor()

    result.execute("select * from BookItinerary")

    lists = result.fetchall()

    for ls in lists:

        output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[5]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[6]+'</font></td>'

        output+='<td><font size=3 color=black>'+ls[7]+'</font></td>'
```

```
            output+='<td><font size=3 color=black>'+ls[8]+'</font></td>'
        context= {'data':output}
        return render(request,'EmployeeScreen.html', context)


def  ViewFeedback(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Username</font></th>'
        output+='<th><font size=3 color=black>Feedback</font></th>'
        output+='<th><font size=3 color=black>Ratings</font></th>'
        output+='<th><font size=3 color=black>Feedback Date</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from feedback")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[3]+'</font></td></tr>'
        context= {'data':output}
        return render(request,'EmployeeScreen.html', context)
```

```
def Feedbacks(request):
    if request.method == 'GET':
        return render(request,'Feedbacks.html', {})


def FeedbacksAction(request):
    if request.method == 'POST':
        global username
        feedback = request.POST.get('t1', False)
        rating = request.POST.get('t2', False)
        today = str(datetime.now())
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "INSERT INTO feedback(username,feedback,ratings,feedback_date)
VALUES('"+str(username)+"','"+feedback+"','"+rating+"','"+today+"')"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Your feedback Accepted! Our employees will work on it"
            context= {'data':data}
            return render(request,'UserScreen.html', context)
        else:
            data = "Error in adding your feedback details"
            context= {'data':data}
```

```
            return render(request,'UserScreen.html', context)


def BookHotelAction(request):
    if request.method == 'POST':
        global username
        pid = request.POST.get('t1', False)
        price = request.POST.get('t2', False)
        traveldate = request.POST.get('t3', False)
        card = request.POST.get('t4', False)
        cvv = request.POST.get('t5', False)
        arr = traveldate.split("-")
        traveldate = arr[2]+"-"+arr[1]+"-"+arr[0]
        today = str(datetime.now())
        bid = 0
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select max(booking_id) from BookItinerary")
            lists = result.fetchall()
            for ls in lists:
                bid = ls[0]
        if bid is not None:
            bid += 1
        else:
```

```
        bid = 1
    dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
    dbcursor = dbconnection.cursor()
    qry = "INSERT INTO BookItinerary(booking_id,username,package_id,price,travel_date,card_no,cvv_no,booking_date,booking_type) VALUES('"+str(bid)+"','"+username+"','"+pid+"','"+price+"','"+traveldate+"','"+card+"','"+cvv+"','"+today+"','Hotel')"
    dbcursor.execute(qry)
    dbconnection.commit()
    if dbcursor.rowcount == 1:
        data = "Hotel booking details added with ID : "+str(pid)
        context= {'data':data}
        return render(request,'UserScreen.html', context)
    else:
        data = "Error in adding Hotel booking details"
        context= {'data':data}
        return render(request,'UserScreen.html', context)


def BookHotel(request):
  if request.method == 'GET':
    pid = request.GET['pid']
    price  = request.GET['price']
    output='<tr><td><font size="3" color="black">Hotel ID</td><td><input type="text" name="t1" size="25" value="'+pid+'" readonly/></td></tr>'
```

```
        output += '<tr><td><font size="3" color="black">Price</td><td><input type="text" name="t2" size="25" value="'+price+'" readonly/></td></tr>'
        context= {'data1':output}
        return render(request,'BookHotel.html', context)


def SearchHotels(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center border-collapse=collapse>'
        output+='<tr><th><font size=3 color=black>Hotel ID</font></th>'
        output+='<th><font size=3 color=black>Hotel Name</font></th>'
        output+='<th><font size=3 color=black>Room Price</font></th>'
        output+='<th><font size=3 color=black>Hotel Location</th>'
        output+='<th><font size=3 color=black>Description</th>'
        output+='<th><font size=3 color=black>Book Hotel</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from hotel")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
```

```
            output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'
            output+='<td><a
href=\'BookHotel?pid='+str(ls[0])+'&price='+ls[2]+'\'><fontsize=3
color=black>Click Here to Book</font></a></td></tr>'
        context= {'data':output}
        return render(request,'UserScreen.html', context)


def BookTransportAction(request):
    if request.method == 'POST':
        global username
        pid = request.POST.get('t1', False)
        price = request.POST.get('t2', False)
        traveldate = request.POST.get('t3', False)
        card = request.POST.get('t4', False)
        cvv = request.POST.get('t5', False)
        arr = traveldate.split("-")
        traveldate = arr[2]+"-"+arr[1]+"-"+arr[0]
        today = str(datetime.now())
        bid = 0
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select max(booking_id) from BookItinerary")
```

```
        lists = result.fetchall()
        for ls in lists:
            bid = ls[0]
    if bid is not None:
        bid += 1
    else:
        bid = 1
    dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
    dbcursor = dbconnection.cursor()
    qry = "INSERT INTO BookItinerary(booking_id,username,package_id,price,travel_date,card_no,cvv_no,booking_date,booking_type) VALUES('"+str(bid)+"','"+username+"','"+pid+"','"+price+"','"+traveldate+"','"+card+"','"+cvv+"','"+today+"','Transport')"
    dbcursor.execute(qry)
    dbconnection.commit()
    if dbcursor.rowcount == 1:
        data = "Transport booking details added with ID : "+str(pid)
        context= {'data':data}
        return render(request,'UserScreen.html', context)
    else:
        data = "Error in adding Transport booking details"
        context= {'data':data}
        return render(request,'UserScreen.html', context)
```

```
def  BookTransport(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        price  = request.GET['price']
        output='<tr><td><fontsize="3" color="black">Transport ID</td><td><input type="text" name="t1" size="25" value="'+pid+'" readonly/></td></tr>'
        output += '<tr><td><font size="3" color="black">Price</td><td><input type="text" name="t2" size="25" value="'+price+'" readonly/></td></tr>'
        context= {'data1':output}
        return render(request,'BookTransport.html', context)


def SearchTransport(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Transport ID</font></th>'
        output+='<th><font size=3 color=black>Transport Name</font></th>'
        output+='<th><font size=3 color=black>Source Location</font></th>'
        output+='<th><font size=3 color=black>Destination Location</th>'
        output+='<th><font size=3 color=black>Departure Time</th>'
        output+='<th><font size=3 color=black>Expected Arrival Time</th>'
        output+='<th><font size=3 color=black>Price</th>'
        output+='<th><font size=3 color=black>Book Transport</th></tr>'
        mysqlConnect =  pymysql.connect(host='127.0.0.1',port  =  3306,user  =  'root',
```

```
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from transport")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[5]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[6]+'</font></td>'
                output+='<td><a
href=\'BookTransport?pid='+str(ls[0])+'&price='+str(ls[6])+'\'><fontsize=3
color=black>Click Here to Book Transport</font></a></td></tr>'
        context= {'data':output}
        return render(request,'UserScreen.html', context)


def BookItineraryAction(request):
    if request.method == 'POST':
        global username
        pid = request.POST.get('t1', False)
        price = request.POST.get('t2', False)
        traveldate = request.POST.get('t3', False)
```

```
        card = request.POST.get('t4', False)
        cvv = request.POST.get('t5', False)
        arr = traveldate.split("-")
        traveldate = arr[2]+"-"+arr[1]+"-"+arr[0]
        today = str(datetime.now())
        bid = 0
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select max(booking_id) from BookItinerary")
            lists = result.fetchall()
            for ls in lists:
                bid = ls[0]
        if bid is not None:
            bid += 1
        else:
            bid = 1
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "INSERT INTO
BookItinerary(booking_id,username,package_id,price,travel_date,card_no,cvv_no,b
ooking_date,booking_type)
VALUES('"+str(bid)+"','"+username+"','"+pid+"','"+price+"','"+traveldate+"','"+card
+"','"+cvv+"','"+today+"','Itinerary')"
```

```
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Itinerary booking details added with ID : "+str(pid)
            context= {'data':data}
            return render(request,'UserScreen.html', context)
        else:
            data = "Error in adding Itinerary booking details"
            context= {'data':data}
            return render(request,'UserScreen.html', context)


def BookItinerary(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        price  = request.GET['price']
        output = '<tr><td><font size="3" color="black">Package ID</td><td><input type="text" name="t1" size="25" value="'+pid+'" readonly/></td></tr>'
        output += '<tr><td><font size="3" color="black">Price</td><td><input type="text" name="t2" size="25" value="'+price+'" readonly/></td></tr>'
        context= {'data1':output}
        return render(request,'BookItinerary.html', context)


def  SearchItinerary(request):
    if request.method == 'GET':
```

```
    global username
    output = '<table border=1 align=center>'
    output+='<tr><th><font size=3 color=black>Itinerary ID</font></th>'
    output+='<th><font size=3 color=black>Itinerary Package Name</font></th>'
    output+='<th><font size=3 color=black>Package Description</font></th>'
    output+='<th><font size=3 color=black>Price</th>'
    output+='<th><font size=3 color=black>Book Itinerary</th></tr>'
    mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
    with mysqlConnect:
        result = mysqlConnect.cursor()
        result.execute("select * from Itinerary")
        lists = result.fetchall()
        for ls in lists:
            output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
            output+='<td><a
href=\'BookItinerary?pid='+str(ls[0])+'&price='+ls[3]+'\'><font size=3
color=black>Click Here to Book</font></a></td></tr>'
    context= {'data':output}
    return render(request,'UserScreen.html', context)

#=========================================================
```

==========user functions above

```
def ModifyHotelAction(request):
    if request.method == 'POST':
        pid = request.POST.get('t1', False)
        name = request.POST.get('t2', False)
        price = request.POST.get('t3', False)
        location = request.POST.get('t4', False)
        desc = request.POST.get('t5', False)

        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "update hotel set hotel_name='"+name+"', room_price='"+price+"', location='"+location+"', service_description='"+desc+"' where hotel_id='"+pid+"'"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Hotel details modified with ID : "+str(pid)
            context= {'data':data}
            return render(request,'Hotels.html', context)
        else:
            data = "Error in modifying Hotel details"
            context= {'data':data}
            return render(request,'Hotels.html', context)
```

```
def ModifyHotel(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from hotel where hotel_id='"+pid+"'")
            lists = result.fetchall()
            for ls in lists:
                name = ls[1]
                price = ls[2]
                location = ls[3]
                desc = ls[4]
                break
        output = '<tr><td><font size="3" color="black">Hotel ID</td><td><input type="text" name="t1" size="25" value="'+pid+'" readonly/></td></tr>'
        output += '<tr><td><font size="3" color="black">Hotel Name</td><td><input type="text" name="t2" size="25" value="'+name+'"/></td></tr>'
        output += '<tr><td><font size="3" color="black">Price</td><td><input type="text" name="t3" size="50" value="'+price+'"/> </td></tr>'
        output += '<tr><td><font size="3" color="black">Hotel Location</td><td><input type="text" name="t4" size="15" value="'+location+'"/> </td></tr>'
```

```
        output += '<tr><td><font size="3" color="black">Description</td><td><input type="text" name="t5" size="15" value="'+desc+'"/> </td></tr>'
        context= {'data1':output}
        return render(request,'ModifyHotel.html', context)


def ViewHotel(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Hotel ID</font></th>'
        output+='<th><font size=3 color=black>Hotel Name</font></th>'
        output+='<th><font size=3 color=black>Room Price</font></th>'
        output+='<th><font size=3 color=black>Hotel Location</th>'
        output+='<th><font size=3 color=black>Description</th>'
        output+='<th><font size=3 color=black>Modify</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from hotel")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
```

```
            output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'
            output+='<td><a     href=\'ModifyHotel?pid='+str(ls[0])+'\'><font     size=3 color=black>Click Here to Modify</font></a></td></tr>'
        context= {'data':output}
        return render(request,'Hotels.html', context)

def  AddHotelAction(request):
    if request.method == 'POST':
        name = request.POST.get('t1', False)
        price = request.POST.get('t2', False)
        location = request.POST.get('t3', False)
        desc = request.POST.get('t4', False)

        pid = 0
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select max(hotel_id) from hotel")
            lists = result.fetchall()
            for ls in lists:
                pid = ls[0]
        if pid is not None:
```

```
            pid += 1
        else:
            pid = 1
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "INSERT INTO hotel(hotel_id,hotel_name,room_price,location,service_description) VALUES('"+str(pid)+"','"+name+"','"+price+"','"+location+"','"+desc+"')"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Hotel details added with ID : "+str(pid)
            context= {'data':data}
            return render(request,'AddHotel.html', context)
        else:
            data = "Error in adding hotel details"
            context= {'data':data}
            return render(request,'AddHotel.html', context)


def AddHotel(request):
    if request.method == 'GET':
        return render(request,'AddHotel.html', {})


def Hotels(request):
```

```
    if request.method == 'GET':
        return render(request,'Hotels.html', {})


def ModifyTransportAction(request):
    if request.method == 'POST':
        pid = request.POST.get('t1', False)
        name = request.POST.get('t2', False)
        source = request.POST.get('t3', False)
        dest = request.POST.get('t4', False)
        depart = request.POST.get('t5', False)
        arrival = request.POST.get('t6', False)
        price = request.POST.get('t7', False)
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "update transport set service_name='"+name+"',
source_location='"+source+"', destination_location='"+dest+"',
departure_time='"+depart+"', expected_reached_time='"+arrival+"',
price='"+price+"' where transport_id='"+pid+"'"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Transport details modified with ID : "+str(pid)
            context= {'data':data}
            return render(request,'Transport.html', context)
```

```
        else:
            data = "Error in modifying Transport details"
            context= {'data':data}
            return render(request,'Transport.html', context)


def ModifyTransport(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from transport where transport_id='"+pid+"'")
            lists = result.fetchall()
            for ls in lists:
                name = ls[1]
                source = ls[2]
                dest = ls[3]
                depart = ls[4]
                arrival = ls[5]
                price = ls[6]
                break
        output='<tr><td><fontsize="3"
color="black">Transport ID</td><td><input        type="text"        name="t1"
size="25" value="'+pid+'" readonly/></td></tr>'
```

```
        output += '<tr><td><font size="3" color="black">Transport Name</td><td><input type="text" name="t2" size="25" value="'+name+'"/></td></tr>'

        output += '<tr><td><font size="3" color="black">Source Location</td><td><input type="text" name="t3" size="50" value="'+source+'"/> </td></tr>'

        output += '<tr><td><font size="3" color="black">Destination Location</td><td><input type="text" name="t4" size="15" value="'+dest+'"/> </td></tr>'

        output += '<tr><td><font size="3" color="black">Departure Time</td><td><input type="text" name="t5" size="15" value="'+depart+'"/> </td></tr>'

        output += '<tr><td><font size="3" color="black">Arrival Time</td><td><input type="text" name="t6" size="15" value="'+arrival+'"/> </td></tr>'

        output += '<tr><td><font size="3" color="black">Price</td><td><input type="text" name="t7" size="15" value="'+price+'"/> </td></tr>'

        context= {'data1':output}

        return render(request,'ModifyTransport.html', context)


def  ViewTransport(request):

    if request.method == 'GET':

        global username

        output = '<table border=1 align=center>'

        output+='<tr><th><font size=3 color=black>Transport ID</font></th>'

        output+='<th><font size=3 color=black>Transport Name</font></th>'

        output+='<th><font size=3 color=black>Source Location</font></th>'
```

```
        output+='<th><font size=3 color=black>Destination Location</th>'
        output+='<th><font size=3 color=black>Departure Time</th>'
        output+='<th><font size=3 color=black>Expected Arrival Time</th>'
        output+='<th><font size=3 color=black>Price</th>'
        output+='<th><font size=3 color=black>Modify</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from transport")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[4]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[5]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[6]+'</font></td>'
                output+='<td><a href=\'ModifyTransport?pid='+str(ls[0])+'\'><font size=3 color=black>Click Here to Modify</font></a></td></tr>'
        context= {'data':output}
        return render(request,'Transport.html', context)


def AddTransportAction(request):
```

```
    if request.method == 'POST':
        name = request.POST.get('t1', False)
        source = request.POST.get('t2', False)
        dest = request.POST.get('t3', False)
        depart = request.POST.get('t4', False)
        arrival = request.POST.get('t5', False)
        price = request.POST.get('t6', False)
        pid = 0
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select max(transport_id) from transport")
            lists = result.fetchall()
            for ls in lists:
                pid = ls[0]
        if pid is not None:
            pid += 1
        else:
            pid = 1
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "INSERT INTO
transport(transport_id,service_name,source_location,destination_location,departure_
```

```
time,expected_reached_time,price)
VALUES('"+str(pid)+"','"+name+"','"+source+"','"+dest+"','"+depart+"','"+arrival+"'
,'"+price+"')"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Transport details added with ID : "+str(pid)
            context= {'data':data}
            return render(request,'AddTransport.html', context)
        else:
            data = "Error in adding Transport details"
            context= {'data':data}
            return render(request,'AddTransport.html', context)


def AddTransport(request):
    if request.method == 'GET':
        return render(request,'AddTransport.html', {})


def Transport(request):
    if request.method == 'GET':
        return render(request,'Transport.html', {})


def ModifyItineraryAction(request):
    if request.method == 'POST':
        pid = request.POST.get('t1', False)
```

```
        name = request.POST.get('t2', False)
        desc = request.POST.get('t3', False)
        price = request.POST.get('t4', False)
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "update Itinerary set package_name='"+name+"', package_description='"+desc+"',price='"+price+"' where package_id='"+pid+"'"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Itinerary details modified with ID : "+str(pid)
            context= {'data':data}
            return render(request,'Itinerary.html', context)
        else:
            data = "Error in modifying Itinerary details"
            context= {'data':data}
            return render(request,'Itinerary.html', context)

def ModifyItinerary(request):
    if request.method == 'GET':
        pid = request.GET['pid']
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
```

```
        result = mysqlConnect.cursor()
        result.execute("select * from Itinerary where package_id='"+pid+"'")
        lists = result.fetchall()
        for ls in lists:
            name = ls[1]
            desc = ls[2]
            price = ls[3]
            break
    output = '<tr><td><font size="3" color="black">Itinerary ID</td><td><input type="text" name="t1" size="25" value="'+pid+'" readonly/></td></tr>'
    output += '<tr><td><font size="3" color="black">Itinerary Name</td><td><input type="text" name="t2" size="25" value="'+name+'"/></td></tr>'
    output += '<tr><td><font size="3" color="black">Attraction Description</td><td><input type="text" name="t3" size="50" value="'+desc+'"/> </td></tr>'
    output += '<tr><td><font size="3" color="black">Package Price</td><td><input type="text" name="t4" size="15" value="'+price+'"/> </td></tr>'
    context= {'data1':output}
    return render(request,'ModifyItinerary.html', context)


def ViewItinerary(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
```

```
        output+='<tr><th><font size=3 color=black>Itinerary ID</font></th>'
        output+='<th><font size=3 color=black>Itinerary Package Name</font></th>'
        output+='<th><font size=3 color=black>Package Description</font></th>'
        output+='<th><font size=3 color=black>Price</th>'
        output+='<th><font size=3 color=black>Modify</th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from Itinerary")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
                output+='<td><a href=\'ModifyItinerary?pid='+str(ls[0])+'\'><font size=3 color=black>Click Here to Modify</font></a></td></tr>'
        context= {'data':output}
        return render(request,'Itinerary.html', context)


def CreateItineraryAction(request):
    if request.method == 'POST':
        name = request.POST.get('t1', False)
        desc = request.POST.get('t2', False)
```

```
    price = request.POST.get('t3', False)
    pid = 0
    mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
    with mysqlConnect:
        result = mysqlConnect.cursor()
        result.execute("select max(package_id) from Itinerary")
        lists = result.fetchall()
        for ls in lists:
            pid = ls[0]
    if pid is not None:
        pid += 1
    else:
        pid = 1
    dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
    dbcursor = dbconnection.cursor()
    qry = "INSERT INTO
Itinerary(package_id,package_name,package_description,price)
VALUES('"+str(pid)+"','"+name+"','"+desc+"','"+price+"')"
    dbcursor.execute(qry)
    dbconnection.commit()
    if dbcursor.rowcount == 1:
        data = "Itinerary details added with ID : "+str(pid)
        context= {'data':data}
```

```
            return render(request,'CreateItinerary.html', context)
        else:
            data = "Error in adding Itinerary details"
            context= {'data':data}
            return render(request,'CreateItinerary.html', context)



def  CreateItinerary(request):
    if request.method == 'GET':
        return render(request,'CreateItinerary.html', {})

def Itinerary(request):
    if request.method == 'GET':
        return render(request,'Itinerary.html', {})

def ViewAuthEmployee(request):
    if request.method == 'GET':
        return render(request,'ViewAuthEmployee.html', {})

def ViewAuthEmployeeAction(request):
    if request.method == 'POST':
        from_date = request.POST.get('t1', False)
        to_date = request.POST.get('t2', False)
        arr = from_date.split("-")
```

```
        if int(arr[0]) <= 9:
            arr[0] = "0"+arr[0]
        if int(arr[1]) <= 9:
            arr[1] = "0"+arr[1]
        from_date = arr[2]+"-"+arr[1]+"-"+arr[0]
        arr = to_date.split("-")
        if int(arr[0]) <= 9:
            arr[0] = "0"+arr[0]
        if int(arr[1]) <= 9:
            arr[1] = "0"+arr[1]
        to_date = arr[2]+"-"+arr[1]+"-"+arr[0]
        print(from_date+" "+to_date)
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Employee Name</font></th>'
        output+='<th><font size=3 color=black>Login Date</font></th>'
        output+='<th><font size=3 color=black>Login Status</font></th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from notification where login_date between
'"+from_date+"' and '"+to_date+"'")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
```

```
            output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[2]+'</font></td></tr>'
        context= {'data':output}
        return render(request,'AdminScreen.html', context)


def ViewEmployee(request):
    if request.method == 'GET':
        global username
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Employee Name</font></th>'
        output+='<th><font size=3 color=black>Password</font></th>'
        output+='<th><font size=3 color=black>Contact No</font></th>'
        output+='<th><font size=3 color=black>Email ID</font></th>'
        output+='<th><font size=3 color=black>Address</font></th></tr>'
        mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        with mysqlConnect:
            result = mysqlConnect.cursor()
            result.execute("select * from employee")
            lists = result.fetchall()
            for ls in lists:
                output+='<tr><td><font size=3 color=black>'+str(ls[0])+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[1]+'</font></td>'
                output+='<td><font size=3 color=black>'+ls[2]+'</font></td>'
```

```
            output+='<td><font size=3 color=black>'+ls[3]+'</font></td>'
            output+='<td><font size=3 color=black>'+ls[4]+'</font></td></tr>'
        context= {'data':output}
        return render(request,'AdminScreen.html', context)


def AddEmployeeAction(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        email = request.POST.get('t4', False)
        address = request.POST.get('t5', False)
        page = None
        dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
        dbcursor = dbconnection.cursor()
        qry = "INSERT INTO employee(username,password,phone_no,email,address)
VALUES('"+str(username)+"','"+password+"','"+contact+"','"+email+"','"+address+"
')"
        dbcursor.execute(qry)
        dbconnection.commit()
        if dbcursor.rowcount == 1:
            data = "Employee details added"
            context= {'data':data}
            return render(request,'AddEmployee.html', context)
```

```
        else:
            data = "Error in adding employee details"
            context= {'data':data}
            return render(request,'AddEmployee.html', context)

def EmployeeScreen(request):
    if request.method == 'GET':
        return render(request,'EmployeeScreen.html', {})

def AddEmployee(request):
    if request.method == 'GET':
        return render(request,'AddEmployee.html', {})

def index(request):
    if request.method == 'GET':
        return render(request,'index.html', {})

def UserLogin(request):
    if request.method == 'GET':
       return render(request, 'UserLogin.html', {})

def Register(request):
    if request.method == 'GET':
       return render(request, 'Register.html', {})
```

```
def AdminLogin(request):
    if request.method == 'GET':
        return render(request, 'AdminLogin.html', {})


def EmployeeLogin(request):
    if request.method == 'GET':
        return render(request, 'EmployeeLogin.html', {})


def FAQ(request):
    if request.method == 'GET':
        return render(request, 'FAQ.html', {})


def isUserExists(username):
    is_user_exists = False
    global details
    mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
    with mysqlConnect:
        result = mysqlConnect.cursor()
        result.execute("select * from user_signup where username='"+username+"'")
        lists = result.fetchall()
        for ls in lists:
            is_user_exists = True
    return is_user_exists
```

```
def RegisterAction(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        email = request.POST.get('t4', False)
        address = request.POST.get('t5', False)
        record = isUserExists(username)
        page = None
        if record == False:
            dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
            dbcursor = dbconnection.cursor()
            qry = "INSERT INTO user_signup(username,password,phone_no,email,address) VALUES('"+str(username)+"','"+password+"','"+contact+"','"+email+"','"+address+"')"
            dbcursor.execute(qry)
            dbconnection.commit()
            if dbcursor.rowcount == 1:
                data = "Signup Done! You can login now"
                context= {'data':data}
                return render(request,'Register.html', context)
            else:
```

```
            data = "Error in signup process"
            context= {'data':data}
            return render(request,'Register.html', context)
        else:
            data = "Given "+username+" already exists"
            context= {'data':data}
            return render(request,'Register.html', context)


def checkUser(uname, password, utype):
    global username
    msg = "Invalid Login Details"
    mysqlConnect = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'keeru@0506', database = 'Travel',charset='utf8')
    with mysqlConnect:
        result = mysqlConnect.cursor()
        result.execute("select * from "+utype+" where username='"+uname+"' and password='"+password+"'")
        lists = result.fetchall()
        for ls in lists:
            msg = "success"
            username = uname
            break
    return msg
```

```
def UserLoginAction(request):
    if request.method == 'POST':
        global username
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        msg = checkUser(username, password, "user_signup")
        if msg == "success":
            context= {'data':"Welcome "+username}
            return render(request,'UserScreen.html', context)
        else:
            context= {'data':msg}
            return render(request,'UserLogin.html', context)


def EmployeeLoginAction(request):
    if request.method == 'POST':
        global username
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        msg = checkUser(username, password, "employee")
        if msg == "success":
            today = str(datetime.now())
            dbconnection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'keeru@0506', database = 'Travel',charset='utf8')
            dbcursor = dbconnection.cursor()
            qry = "INSERT INTO notification(employee_name,login_date,login_status)
```

```
VALUES('"+str(username)+"','"+today+"','Successful')"
            dbcursor.execute(qry)
            dbconnection.commit()
            context= {'data':"Welcome "+username}
            return render(request,'EmployeeScreen.html', context)
        else:
            context= {'data':msg}
            return render(request,'EmployeeLogin.html', context)



def AdminLoginAction(request):
    if request.method == 'POST':
        global username
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'admin' and password == 'admin':
            context= {'data':"Welcome "+username}
            return render(request,'AdminScreen.html', context)
        else:
            context= {'data':msg}
            return render(request,'AdminLogin.html', context)
```

## SYSTEM TEST CASES

### WHITE BOX TESTING:

White box testing is also known as structural testing or code-based testing, and it is used to test the software's internal logic, flow, and structure. The tester creates test cases to examine the code paths and logic flows to ensure they meet the specified requirements. Input: Requirements, Functional specifications, design documents, source code.

| Test case no | Description | Test Data | Expected output | Actual output | Status |
|---|---|---|---|---|---|
| 1 | Verify response when a admin clicks on admin login | Admin gesture clicks on admin login | Admin page rendered successful | Admin page rendered successful | Pass |
| 2 | Verify response when aemployee clicks on Employee Login | Employee gesture Clicks on EmployeeLogin | Employee page renderedsuccessful | Employee page rendered successful | Pass |
| 3 | Verify response when a userclicks on User login | User gesture Clicks on user login | User login page renderedsuccessful | User login page rendered successful | Pass |
| 4 | Verify response when a userclicks on sign up | User gesture Clicks on sign up | Sign up page renderedsuccessful | Sign up page rendered successful | Pass |
| 5 | Verify response when a admin clicks on addemployees | Admin gestureClicks on add employees | Add employees pagerendered successful | Add employees pagerendered successful | Pass |
| 6 | Verify response when admin clicks on view authenticatedemployees | Admin gesture Clicks on view authenticated employees | Authenticated Pagerendered successful | Authenticated Pagerendered successful | Pass |
| 7 | verify response when employee clicks on add transport | emplyoee gesture clickson add transport | add transport page rendered successful | add transport page rendered successful | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 8 | verify response when employee clicks on addhotels | employee gesture clickson add hotels | add hotels page rendered successful | add hotels page rendered successful | Pass |
| 9 | verify response when employee clicks on create itinerary package | employee gesture clickson create itinerary package | create package pagerendered successful | create package pagerendered successful | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 10 | verify response when employee clicks viewbookings | employee gesture clickson view bookings | view bookings pagerendered successful | view bookings pagerendered successful | Pass |
| 11 | verify response when employee clicks on feedback and ratings | employee gesture clickson feedback and ratings | feedback page renderedsuccessful | feedback page rendered successful | Pass |
| 12 | verify response when user clicks on search itinerarypackage | user gesture clicks onsearch itinerary package | serach itinerary pagerendered successful | search itinerary page rendered successful | Pass |
| 13 | verify response when userclicks on search transport | user gesture clicks onsearch transport | transport page renderedsuccessful | transport page rendered successful | Pass |
| 14 | verify response when user clicks on serach hotels | user gesture clicks on serach hotels | hotels page rendered successful | hotels page rendered successful | Pass |
| 15 | verify response when userclicks on cancel bookings | user gesture clicks oncancel bookings | cancel bookings pagerendered successful | cancel bookings page rendered successful | Pass |

**BLACK BOX TESTING:**

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

| Test case no | Description | Test Data | Expected output | Actual output | Status |
|---|---|---|---|---|---|
| 1 | Verify response adminenters username and password | username: admin password: ***in | welcome admin | welcome admin | Pass |
| 2 | verify response employeeenters username and password | username: manoj password: ****j | Welcome manoj | Welcome manoj | Pass |
| 3 | verify response employee enters invaild username or password | username: manj password: ****j | invalid login details | invalid login details | Pass |
| 4 | verify response admin login and add employee | employee name :keerthanaPassword: keeru contact No:8743634678 email id: keeru01@gmail.com address: hyderabad | employee details added | employee details added | Pass |
| 5 | verify response employee login and creates itinerary package | Itinerary name: keralaPackage price:70000 | Itinearary details added | Itinerary details added | Pass |
| 6 | verify response employeelogin and add transport details | Transport name: flight Source Location: hyderabadDestination Location: kerala Departure time: 10:00am Arrival time: 2:00pm | Transport details added | transport details added | Pass |
| 7 | verify response employee login and add hotels details | Hotel name: marriot beachhouse room price:2000 Hotel Location: kerala | Hotels details added | hotels details added | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 8 | verify response user loginand search itinerary | Click here to book the itinerary and add travellingdate and card no cvv no and click on submit | Itinerary booking detailsadded with ID: | Itinerary booking detailsadded with ID: | Pass |
| 9 | verify response user loginand search Transport | Click here to book the Transport and travellingdate and card no cvv no and click on submit | Transport booking details added with ID: | Transport booking details added with ID: | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 10 | verify response user loginand search hotels | Click here to book hotels and Travelling date and card no cvv no and click on submit | Hotel booking details added with ID: | Hotel booking details added with ID: | Pass |
| 11 | verify response user loginand cancel booking | click here to cancel | Selected Booking Deleted: ID | Selected Booking Deleted:5 | Pass |
| 12 | verify response user loginand Feedback and Ratings | Feedback: good Rating:4 and click on submit | Your feedback Accepted! our employees will work onit | Your feedback Accepted! our employees will work onit | Pass |

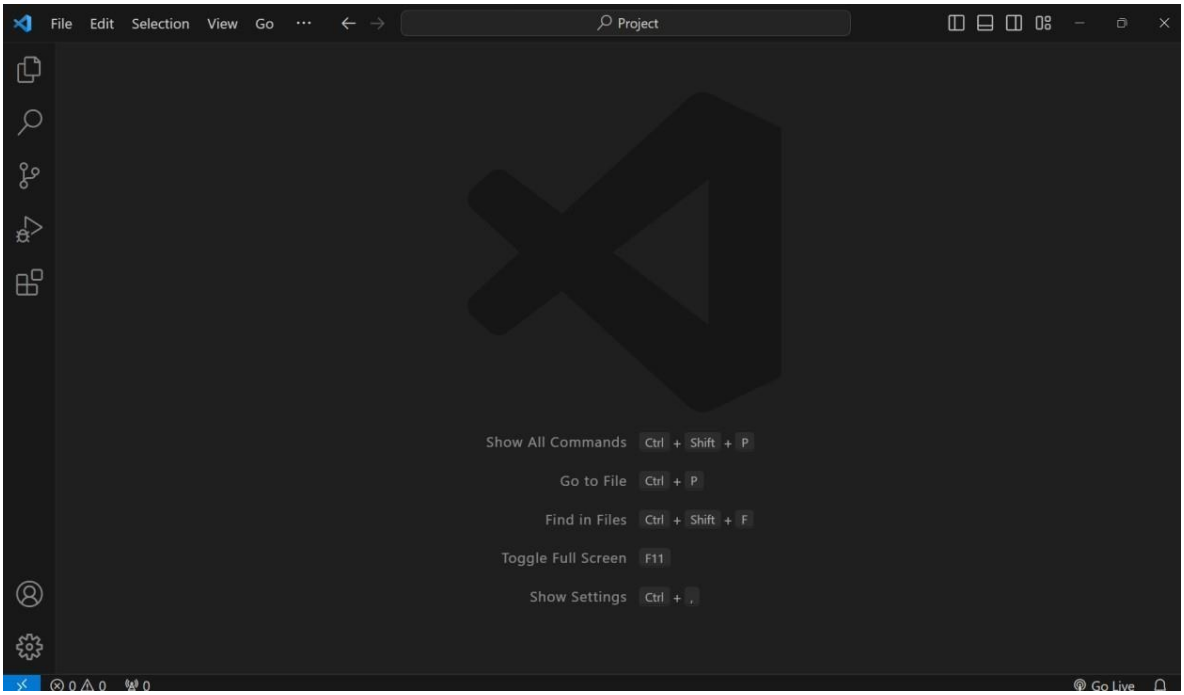
# HELP FILE

## VISUAL STUDIO CODE:

Visual Studio is an **Integrated Development Environment(IDE)** developed by Microsoft to develop Desktop applications, GUI(Graphical User Interface), console, web applications, mobile applications, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

## Install VSCODE:

Download the vscode from official visual studio code



**DJANGO:**

Django is a Python-based web framework that allows you to quickly create web applications without all of the installation or dependency problems that you normally will find with other frameworks. Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. The latest official version is 2.2.5.

**Install Python:** Django is a Python web framework, so you need to have Python installed on your system. You can download the latest version of Python from the official website: Python Downloads.

**Install Django using pip:** Django can be installed using the pip package manager. Open a command prompt or terminal window and run the following command:

**pip install Django**


**Verify Django version:**

```
python -m django –version
```

**DJANGO-MYSQL:**

You should have MySQL installed and be able to log in with appropriate privileges

**Install Mysql using pip:** Open terminal and run the following command:

**Pip install mysqlclient**

Go to settings.py here inside the databases variable configure MYSQL Database values

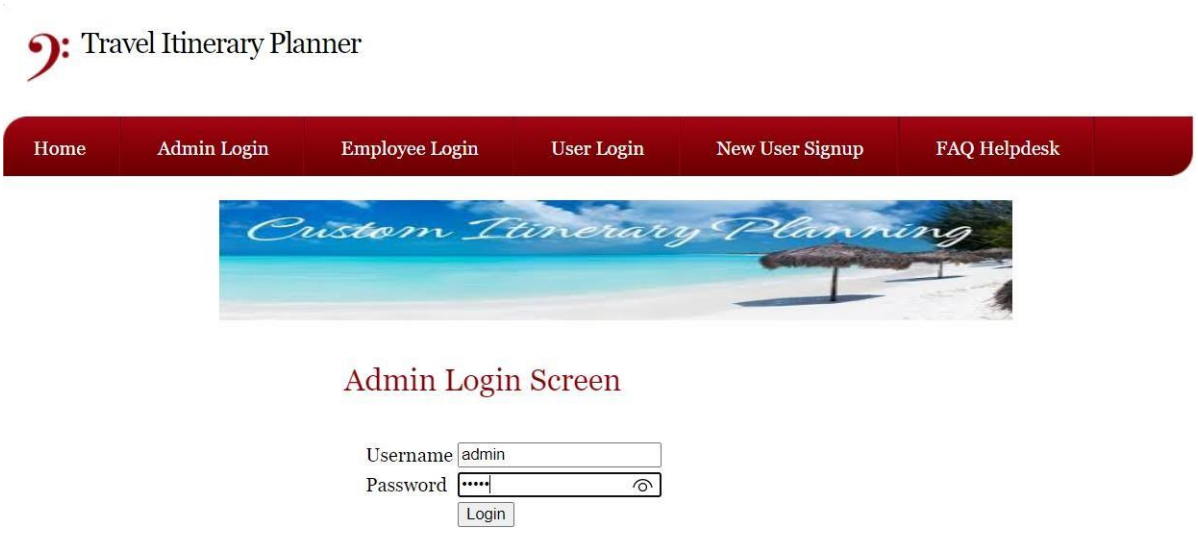
```
DATABASES = {
```

```
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'Travel',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'USER': 'root',
        'PASSWORD': 'keeru@0506',
            'OPTIONS': {
          'autocommit': True,
        },
    }
}
```

## OUTPUT SCREENS

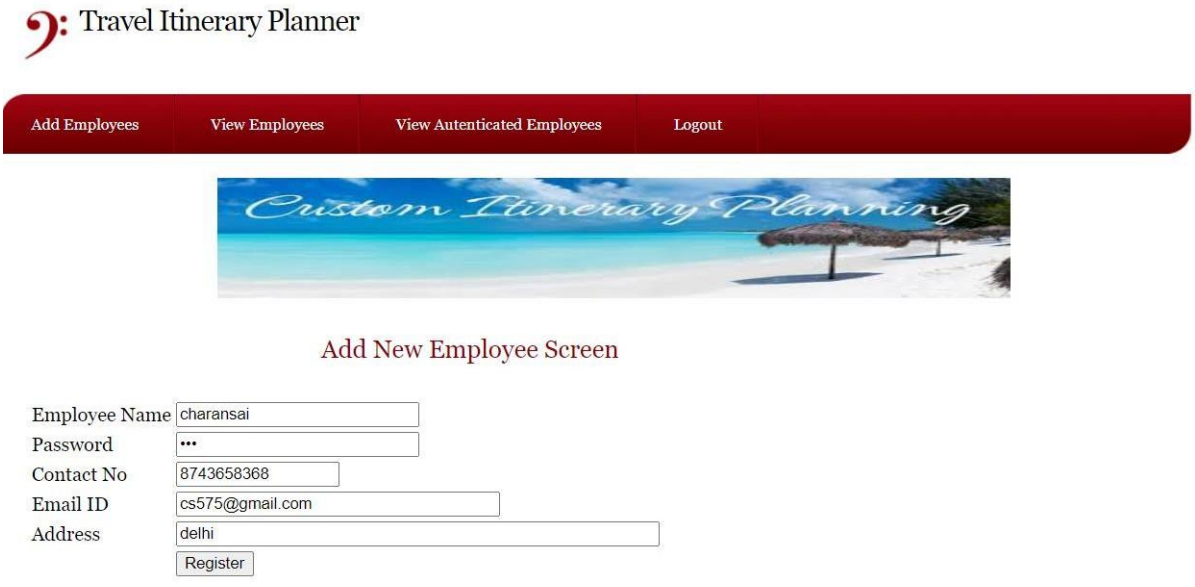
### Welcome Page:


### Admin Login Screen:



## Admin welcome page:


## Add Employees page:



## View Employees page:

Add Employees | View Employees | View Autenticated Employees | Logout


| Employee Name | Password | Contact No | Email ID | Address |
|---|---|---|---|---|
| charansai | sai | 8743658368 | cs575@gmail.com | delhi |
| juveria | 1234 | 1234569879 | juveria@gmail.com | khammam |
| keerthana | keeru | 8743634678 | keeru01@gmail.com | hyderabad |
| manoj | manoj | 9675467231 | manoj123@gmail.com | delhi |
| shiva | shiva | 1234567890 | shiva123@gmail.com | hyd |
| shivaa | shiva | 9966499110 | shiva123@gmail.com | hyd |

## View Authenticated page:

Travel Itinerary Planner

Add Employees | View Employees | View Autenticated Employees | Logout


View Authenticated Employee Screen

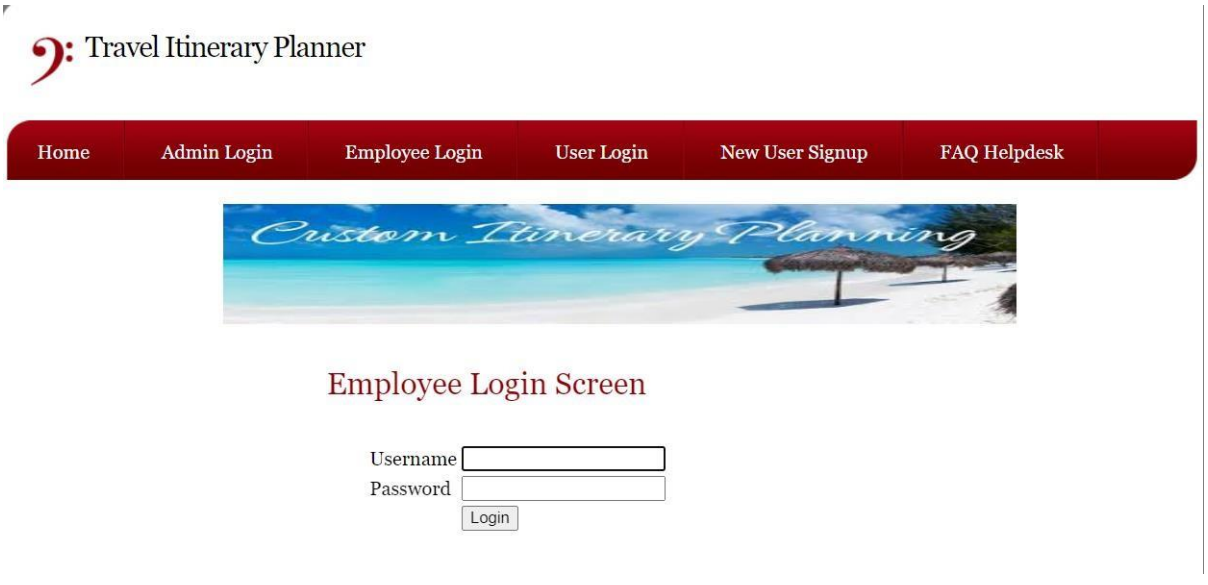
**Frome Date**

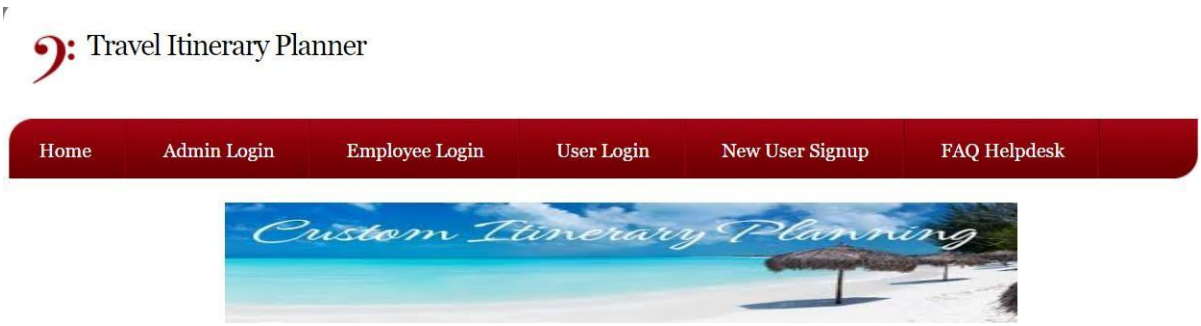
**To Date**

Submit


| Employee Name | Login Date | Login Status |
|---|---|---|
| shivaa | 2024-02-04 09:10:06.525306 | Successful |
| shivaa | 2024-02-04 09:18:21.905063 | Successful |
| manoj | 2024-02-04 09:53:30.147127 | Successful |
| manoj | 2024-02-04 10:00:57.103972 | Successful |
| manoj | 2024-02-04 12:14:10.360131 | Successful |
| manoj | 2024-02-04 16:40:20.630292 | Successful |
| manoj | 2024-02-04 16:45:07.180118 | Successful |
| manoj | 2024-02-05 11:50:25.655059 | Successful |
| manoj | 2024-02-05 12:14:22.334664 | Successful |
| manoj | 2024-02-05 12:22:04.249329 | Successful |
| manoj | 2024-02-05 14:21:09.846410 | Successful |
| manoj | 2024-02-05 14:24:23.199435 | Successful |
| manoj | 2024-02-05 14:43:28.969912 | Successful |

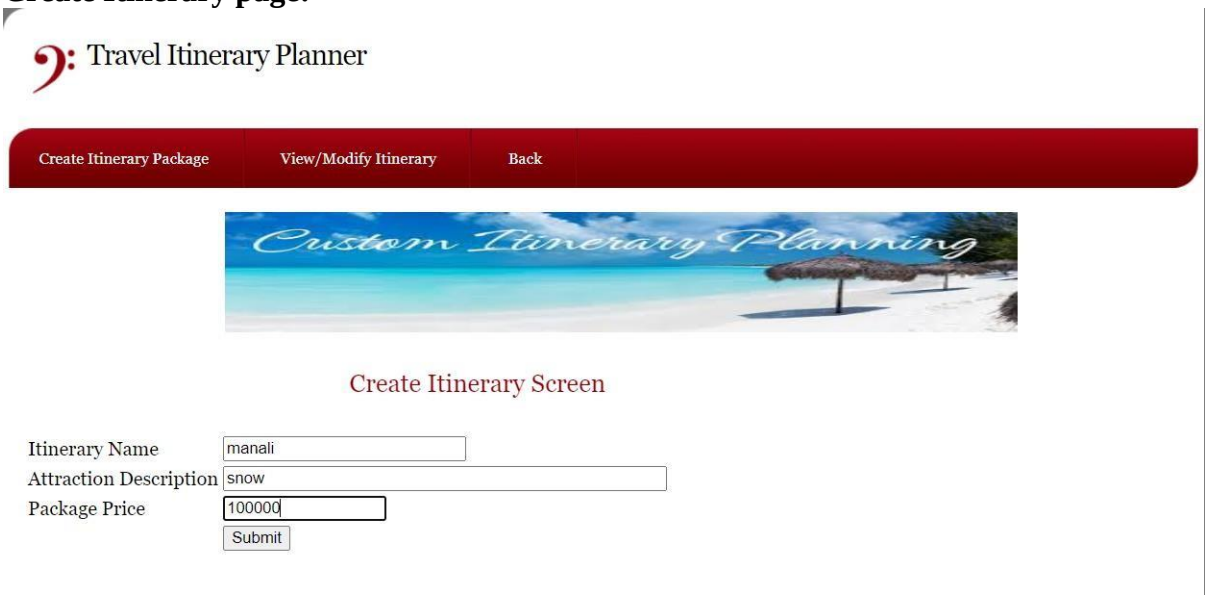
## Employee Login page:



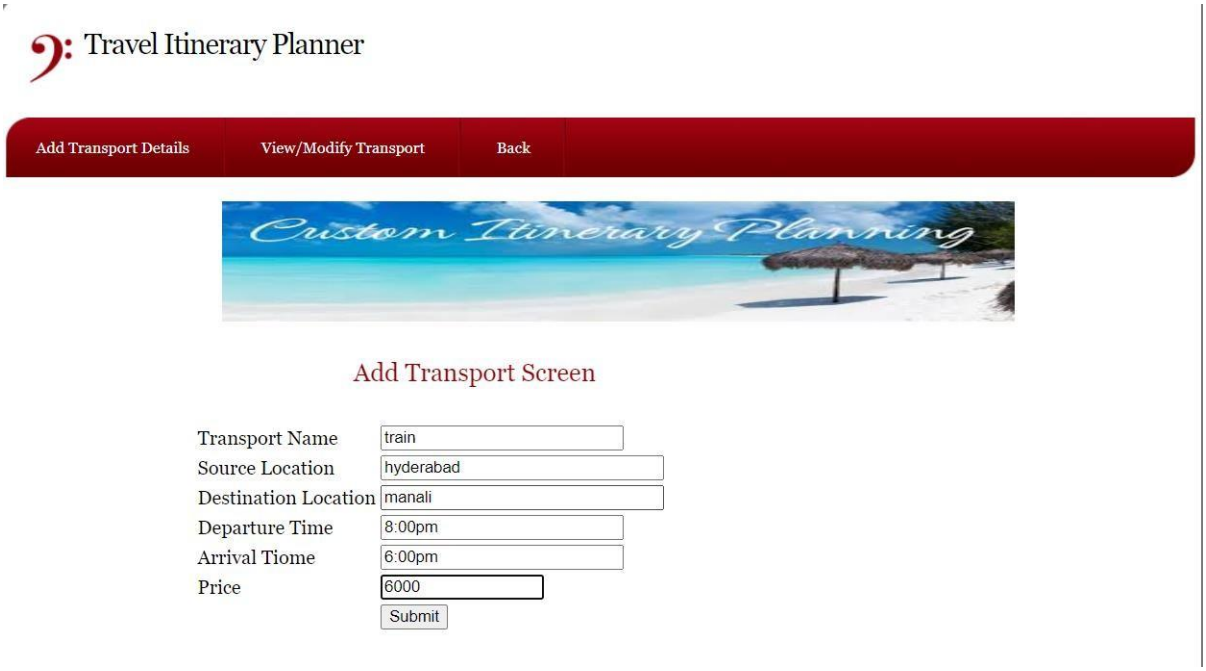



## Employee Home page:

## Create Itinerary page:




## Add Transport page:



## Add Hotels page:

Travel Itinerary Planner

Add Hotels | View/Modify Hotels | Back


Add Hotel Screen

| | |
|---|---|
| Hotel Name | SG View |
| Room Price | 3000 |
| Hotel Location | manali |
| Service Description | neat and clean rooms |

Submit

## View bookings page:

Create Itinerary Package | Add Transport | Add Hotels | View Bookings | Feedback & Ratings | Logout

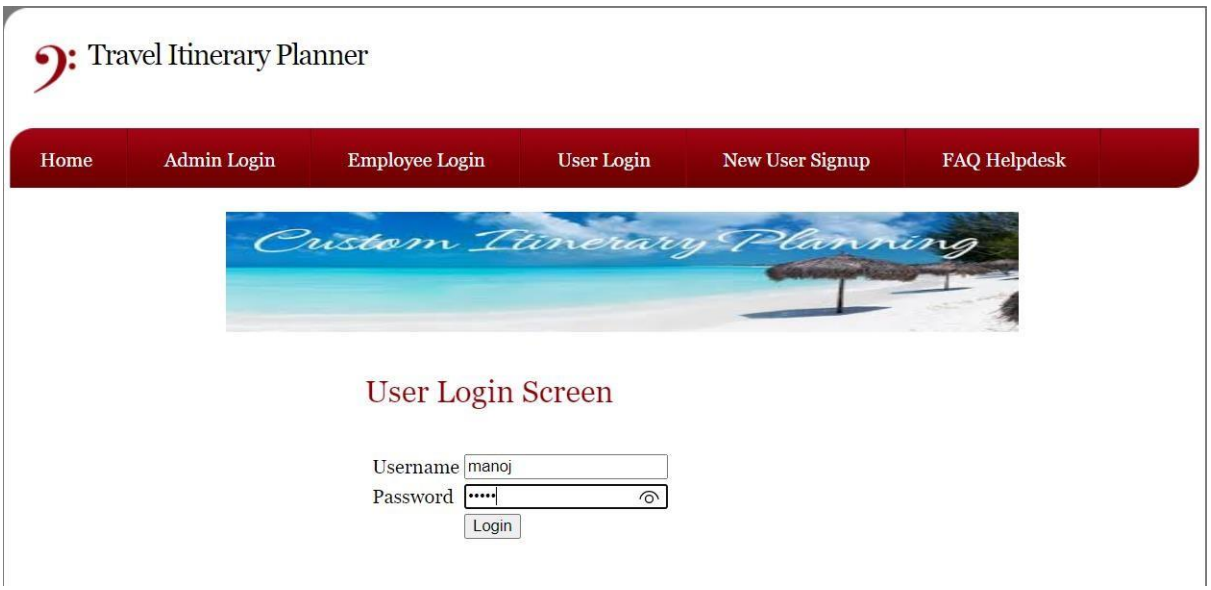

| Booking ID | Username | Package ID | Price | Travel Date/Hotel Occupying Date | Card No | CVV No | Booking Date | Booking Type |
|---|---|---|---|---|---|---|---|---|
| 2 | vineel | 1 | 2000 | 2024-2-15 | 1 | 1 | 2024-02-04 09:15:56.740036 | Transport |
| 3 | vineel | 1 | 100000 | 2024-2-15 | 1 | 1 | 2024-02-04 09:16:13.351232 | Hotel |
| 4 | manoj | 1 | 2000 | 2024-2-2 | 2324 | 9090 | 2024-02-04 09:59:12.487467 | Transport |

## Feedback and ratings page:


| Username | Feedback | Ratings | Feedback Date |
|---|---|---|---|
| vineel | b | 1 | 2024-02-04 09:16:53.921794 |
| manoj | okk | 2 | 2024-02-04 16:59:48.787169 |
| manoj | good | 4 | 2024-02-21 13:45:08.575230 |

---

## User login page:



## User Home page:

Travel Itinerary Planner

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


Welcome manoj

## Search Itinerary page:

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


| Itinerary ID | Itinerary Package Name | Package Description | Price | Book Itinerary |
|---|---|---|---|---|
| 1 | hyd-delhi | hyd-madyapradesh-delhi | 100000 | Click Here to Book |
| 2 | kerala | nature | 8000 | Click Here to Book |
| 3 | goa | beaches | 30000 | Click Here to Book |
| 4 | delhi | tajmahal | 10000 | Click Here to Book |
| 5 | kerala | nature | 70000 | Click Here to Book |
| 6 | manali | snow | 100000 | Click Here to Book |

## Search Transport page:

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


| Transport ID | Transport Name | Source Location | Destination Location | Departure Time | Expected Arrival Time | Price | Book Transport |
|---|---|---|---|---|---|---|---|
| 1 | bike | hyd | delhi | 4:00 am | 3:00am | 2000 | Click Here to Book Transport |
| 2 | flight | hyd | mumbai | 10:00 | 7:00pm | 20000 | Click Here to Book Transport |
| 3 | buses | hyd | kerala | 2:00am | 10:00pm | 1000 | Click Here to Book Transport |
| 4 | flight | hyderabad | kerala | 10:00am | 2:00pm | 5000 | Click Here to Book Transport |
| | | | | | | | Click Here |

## Search Hotels page:

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


| Hotel ID | Hotel Name | Room Price | Hotel Location | Description | Book Hotel |
|---|---|---|---|---|---|
| 1 | taj hotel mumbai | 100000 | mumbai | tata ka hotel | Click Here to Book |
| 2 | noida | 100 | kolkata | 2325 | Click Here to Book |
| 3 | novotel | 10000 | kerala | travelling | Click Here to Book |

## Cancel booking page:

Travel Itinerary Planner

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


| Booking ID | Username | Package ID | Price | Travel Date/Hotel Occupying Date | Card No | CVV No | Booking Date | Booking Type | Cancel Booking |
|---|---|---|---|---|---|---|---|---|---|
| 4 | manoj | 1 | 2000 | 2024-2-2 | 2324 | 9090 | 2024-02-04 09:59:12.487467 | Transport | Click Here to Cancel |

## Feedback and ratings page:

Travel Itinerary Planner

Search Itinerary Package | Search Transport | Search Hotels | Cancel Booking | Feedback & Ratings | Logout


Feedback Screen

Feedback
Ratings 1
Submit

**UserSignup page:**

Travel Itinerary Planner

Home | Admin Login | Employee Login | User Login | New User Signup | FAQ Helpdesk


## New User Signup Screen

Username
Password
Contact No
Email ID
Address
Register

## CONCLUSION

The travel itinerary planner project has successfully achieved its objectives of creating a user-friendly platform for organizing and optimizing travel plans. Through thorough research, efficient design, and implementation of features such as destination recommendations, transportation options, accommodation suggestions, and activity scheduling, we have developed a comprehensive tool that meets the needs of travelers seeking convenience and efficiency in trip planning

## REFERENCES

Information for this project is gathered from Google and our syllabus book of WEBDESIGNING, and from the following websites.


https://www.geeksforgeeks.org/

https://www.w3schools.com/

https://chat.openai.com/