

THE INDIAN SCHOOL BAHRAIN

Computer Science Project

TRAVELS AND TOURISM



Submitted by
JOEL JUBY | XII-K

CERTIFICATE

This is to certify that the project titled Travels and Tourism has been completed by Joel Juby of class XII-K and submitted to his Computer Science teacher Mr. Pius Mathew according to the rules and norms set by the Central Board of Secondary Education (CBSE) for the academic year 2020-2021.

Signature of
Internal Examiner

Signature of
External Examiner

TABLE OF CONTENTS

1. Introduction
2. System analysis
3. System design
4. Source code
5. Sample output
6. Acknowledgement
7. Conclusion

INTRODUCTION

The Travel and Tourism project is divided into two, Consumer section, and the Employee section.

The Consumer section is further divided into sections for booking, and cancellation. The consumer may browse through a selection of countries provided, and read through the recommended cities and tourist attractions before making their final booking. The consumer will proceed to get an estimate for cost of the trip, which is calculated based on location, duration of stay, number of travellers, and the number of rooms required. This cost can be displayed in Bahraini Dinar, US Dollars, Indian Rupees, or the local currency. The consumer is then asked for their details, upon which a consumer ID is assigned to them, the consumer status is set to "ACTIVE", and a booking is made.

This program utilizes both Python and MySQL.

About Python

Python is an object-oriented programming language that is designed in C. By nature, it is a high-level programming language that allows for the creation of both simple as well as complex operations.

Features of Python

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

- Easy to code
- High-level language
- Extensive array of library
- Support for other languages

Easy to Code

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn.

High-Level Language

Python has been designed to be a high-level programming language, which means that when you code in Python you don't need to be aware of the coding structure, architecture as well as memory management.

Extensive Array of Library

Out of the box, Python comes inbuilt with a large number of libraries that can be imported at any instance and be used in a specific program. The presence of libraries also makes sure that you don't need to write all the code yourself and can import the same from those that already exist in the libraries.

Support for Other Languages

Being coded in C, Python by default supports the execution of code written in other programming languages such as Java, C, and C#, thus making it one of the versatile in the industry.

MySQL

SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

SYSTEM ANALYSIS

The Travel and Tourism project is divided into two, Consumer section, and the Employee section.

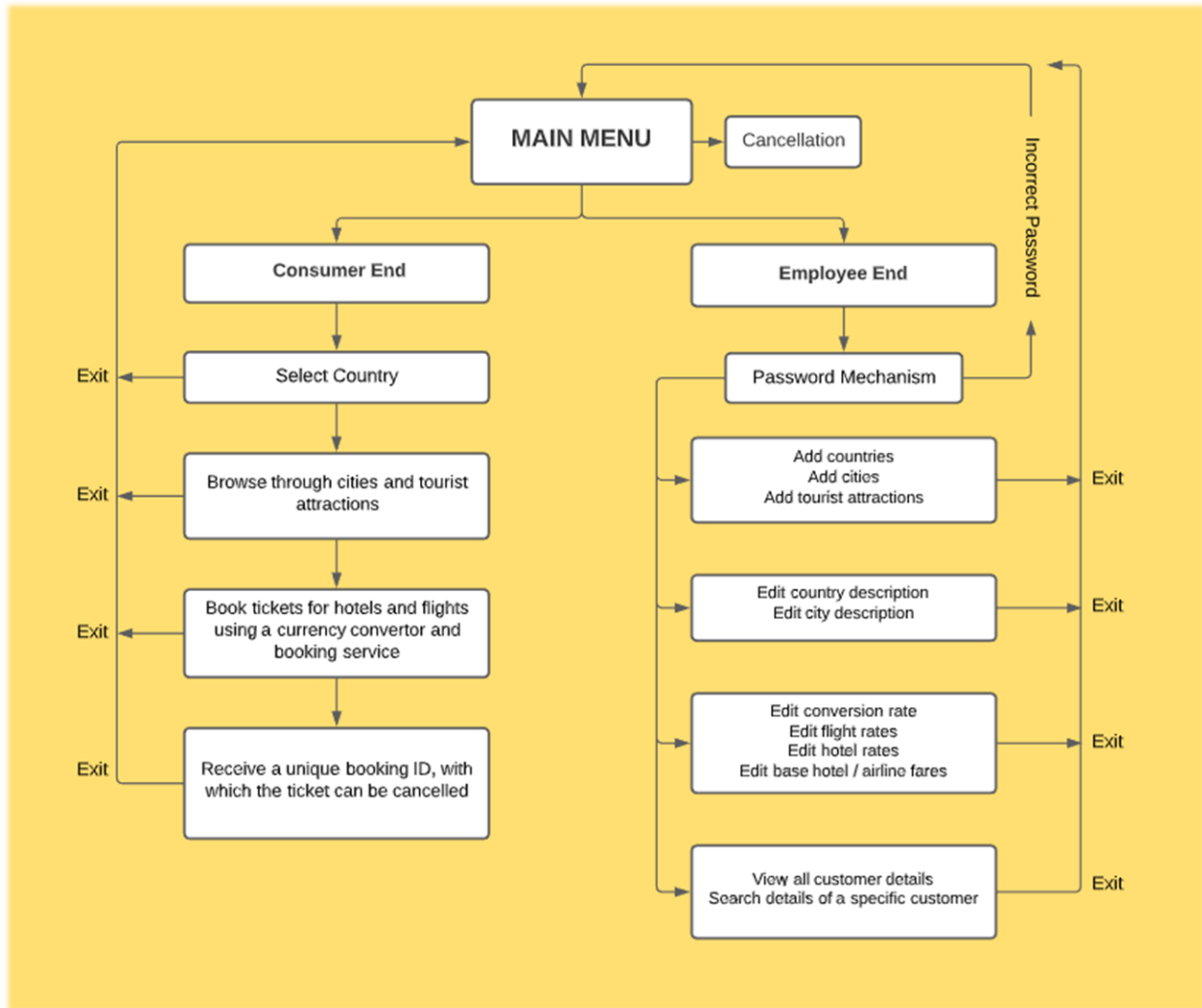
The consumer may cancel their booking at any time. To cancel the booking, the consumer will need to enter their consumer ID, upon which they will receive a refund, and the consumer status is set to "CANCELLED".

A password is required to access the Employee section, set to "1234" for this program. After the user has gained access to the Employee section, they may edit the existing info, or input new entries as required.

There are a large number of functions for the employee section of the program. Firstly, insertcountry, which is used to insert a country into the main table. It comes with functions such as insertcity and inserttourism, each of which are their own independent functions that can be used by the employee.

Functions such as edithotelmult and editflightmult, however, are specifically used outside of the other functions, as they are used to update values that already exist for a certain country.

SYSTEM DESIGN



SQL TABLES

```
mysql> use travel_and_tourism;
```

```
Database changed
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_travel_and_tourism |
+-----+
| conversion                    |
| countrycity                   |
| countrytourism                |
| customerinfo                  |
| fares                          |
| flights                       |
| hotels                        |
| maintable                     |
+-----+
```

```
8 rows in set (0.01 sec)
```

```
mysql> desc conversion;
```

Field	Type	Null	Key	Default	Extra
COUNTRY	varchar(20)	YES		NULL	
RATEBHD	decimal(5,2)	YES		NULL	
RATEUSD	decimal(5,2)	YES		NULL	
RATEINR	decimal(5,2)	YES		NULL	

```
4 rows in set (0.01 sec)
```

```
mysql> desc countrycity;
```

Field	Type	Null	Key	Default	Extra
COUNTRY	varchar(20)	YES		NULL	
CITY	varchar(20)	YES		NULL	
DESCRIPTION	varchar(2000)	YES		NULL	

```
3 rows in set (0.01 sec)
```

```
mysql> desc countrytourism;
```

Field	Type	Null	Key	Default	Extra
COUNTRY	varchar(20)	YES		NULL	
PLACES	varchar(200)	YES		NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> desc customerinfo;
```

Field	Type	Null	Key	Default	Extra
MOBILE	varchar(20)	YES		NULL	
EMAIL	varchar(60)	YES		NULL	
ID	varchar(8)	YES		NULL	
NAME	varchar(30)	YES		NULL	
STATUS	varchar(15)	YES		NULL	

```
5 rows in set (0.02 sec)
```

```
mysql> desc farees;
```

```
ERROR 1146 (42S02): Table 'travel_and_tourism.farees' doesn't exist
```

```
mysql> desc fares;
```

Field	Type	Null	Key	Default	Extra
COUNTRY	varchar(20)	YES		NULL	
AIRFARE	int	YES		NULL	
HOTELFARE	int	YES		NULL	
CURRENCYNAME	varchar(20)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> desc flights;
```

Field	Type	Null	Key	Default	Extra
SLNO	varchar(2)	YES		NULL	
FLIGHTNAME	varchar(20)	YES		NULL	
ECONMULT	decimal(2,1)	YES		NULL	
BUSINESSMULT	decimal(2,1)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> desc hotels;
```

Field	Type	Null	Key	Default	Extra
SLNO	varchar(2)	YES		NULL	
NAME	varchar(20)	YES		NULL	
MULTIPLIER	decimal(2,1)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> desc maintable;
```

Field	Type	Null	Key	Default	Extra
COUNTRY	varchar(20)	YES		NULL	
CONTINENT	varchar(20)	YES		NULL	
DESCRIPTION	varchar(2000)	YES		NULL	

```
3 rows in set (0.00 sec)
```

SOURCE CODE

```
import mysql.connector
mydb=mysql.connector.connect(host="localhost",user="root",passwd="1234",database="TRAVEL_AND_TOURISM")
mycursor=mydb.cursor()
import random
from tabulate import tabulate

n=None
m=None
totalhotel=None
totalair=None
countryname=None

#CONSUMER FUNCTIONS#

def deco():
    #use to create line of decor
    print("><"*78)

def mainmenu():
    #use to return back to menu
    deco()
    print("\n")
    global n
    x="***WELCOME TO AZURE TRAVEL AND TOURISM INC.***"
    y='ENTER 1 FOR BOOKING'
    t='ENTER 2 FOR CANCELLATION'
    z='ENTER 3 IF YOU ARE AN EMPLOYEE'
    k="ENTER 0 IN ORDER TO EXIT"
    print(x.center(158),"\n",y.center(158),"\n",t.center(156),"\n",z.center(155),"\n\n",k.center(156),end="\n\n\n")
    deco()
    n=int(input("> Kindly enter your choice :"))

def maintable():
    #use to display screen 1 for consumer
    global m
    com="select country,continent from MAINTABLE"
    mycursor.execute(com)
    lists=mycursor.fetchall()
    deco()
    print(tabulate(lists,headers=["COUNTRY","CONTINENT"],tablefmt='fancy_grid'))
    deco()
    m=input("> Kindly enter name of country you are interested in :")

def CONSUMERSCREEN1():
```

```

#displays the main consumer screen
global m
m=m.capitalize()
print("\n",m.center(154),"\n\n")
com="select description from MAINTABLE where country='{}' ".format(m)
mycursor.execute(com)
lists=mycursor.fetchall()
print("==>",lists[0][0],"\n")
print("## Cities you are recommended to visit during your stay in {} include ## ".format(m),"\n")
com="select city,description from COUNTRYCITY where country='{}' ".format(m)
mycursor.execute(com)
lists=mycursor.fetchall()
for tuples in lists:
    print("<>",tuples[0],"\n","->",tuples[1],"\n")

print("## Popular tourist attractions in {} include ## ".format(m),"\n")
com="select places from COUNTRYTOURISM where country='{}' ".format(m)
mycursor.execute(com)
lists=mycursor.fetchall()
for tuples in lists:
    print("<>",tuples[0])
print("\n")
deco()

```

```

def calculatecost():
    #used in total cost calculation
    global x
    com="select airfare,hotelfare,currencyname from FARES where country='{}' ".format(m)
    mycursor.execute(com)
    lists=mycursor.fetchall()
    air=lists[0][0]
    hotel=lists[0][1]
    currency=lists[0][2]
    return air,hotel,currency

```

```

def flights():
    #used to display available flights
    global y
    global totalair
    fares=calculatecost()
    currency=fares[2]
    airfare=fares[0]
    com="select slno,flightname,econmult,businessmult from flights"
    mycursor.execute(com)
    lists=mycursor.fetchall()
    deco()
    k=0
    print("SL.NO", "\t\t", "AIRLINE", "\t\t", "ECONOMY CLASS", "\t\t", "BUSINESS CLASS")
    print("+"*80)
    for tuples in lists:

```

```

        print(">",tuples[0],"\t\t",tuples[1],"\t\t",airfare*tuples[2], " ",currency,"\t\t",airfare*tuples[3], " ",currency)
        k+=1
    deco()
    print("\n")
    n=input("Enter serial number of the airline of your choice :")
    print("\n")
    a=input("Enter class preference (E/B) :")
    print("\n")
    deco()
    a=a.upper()
    for tuples in lists:
        if tuples[0]==n:
            if a=="E":
                mult=tuples[2]
            else:
                mult=tuples[3]
            totalair=airfare*mult*y

def hotelsandflights():
    #used for display of both hotels and flights
    global m
    global x
    global y
    global z
    global totalhotel
    global totalair
    fares=calculatecost()
    hotelfare=fares[1]
    currency=fares[2]
    com="select slno,name,multiplier from hotels"
    mycursor.execute(com)
    lists=mycursor.fetchall()
    com="select city from countrycity where country='{ }'".format(m)
    mycursor.execute(com)
    lists2=mycursor.fetchall()
    L=[]
    for tuples in lists2:
        L.append(tuples[0])
    k=0
    print("SL.NO","\t\t","HOTEL","\t\t\t","CITY","\t\t\t","PRICE/NIGHT")
    print("+"*80)
    for tuples in lists:
        print(">",tuples[0],"\t\t",tuples[1],"\t\t",L[k],"\t\t",hotelfare*tuples[2], " ",currency)
        k+=1
    deco()
    print("\n")
    n=input("> Enter serial number of the hotel of your choice :")
    for tuples in lists:
        if tuples[0]==n:
            mult=tuples[2]

```

```

        break
    totalhotel=hotelfare*mult*z*x
    print("\n")
    flights()
    print("\n")
    print("> Total flight cost (for all travellers) is :",totalair,currency,"\n")
    print("> Total hotel cost (for all travellers) is :",totalhotel,currency,"\n")
    print("> Total cost of trip for all travellers is estimated at :",totalair+totalhotel,currency)
    print("\n")

```

```

def convertor():
    #used for currency conversion
    global m
    global a
    com="select ratebhd,rateusd from CONVERSION where country='{ }'.format(m)
    mycursor.execute(com)
    lists=mycursor.fetchall()
    tuples=lists[0]
    if a==1:
        currency="BHD"
        rate=tuples[0]
    elif a==2:
        currency="USD"
        rate=tuples[1]
    return rate,currency

```

```

def booking():
    #booking mechanism
    global mob
    email=input("> Enter email address :")
    print("\n")
    method=input("> Enter method of payment [CREDIT CARD/DEBIT CARD] :")
    print("\n")
    deco()
    print("\n")
    print(">>WELCOME TO EZPAY SECURE PAYMENT GATEWAY<<".center(158))
    cred=int(input("> ENTER CARD NUMBER :"))
    print("\n")
    date=input("> ENTER DATE OF EXPIRY [MM/YYYY] :")
    print("\n")
    name=input("> ENTER NAME OF CARDHOLDER :")
    print("\n")
    deco()
    print(">>EZPAY SECURE PAYMENT GATEWAY<<".center(158))
    print("> Card Number : ",cred,"\n")
    print("> Expiry Date : ",date,"\n")
    print("> Name of card holder : ",name,"\n")
    global totalsir
    global totalhotel
    global currency

```

```

global x
global y
global z
global r

cost=(totalair*r)+(totalhotel*r)
print("> AMOUNT TO BE PAID : ",cost,currency,"\n")
n=input("> Press Y to confirm payment :")
print("\n")
deco()
print("\n")
print("> CONGRATULATIONS !!!! You have successfully completed the booking of your trip !!","\n")
ran="2002"+str(random.randint(1000,2000))
print("> Your booking ID is : ",ran, "\n")
print("> *DO NOT SHARE THE ABOVE BOOKING ID WITH ANYONE. KINDLY SAVE IT TO AVAIL OUR OTHER SERVICES*", "\n")
print("> Your booking confirmation shall be sent to your e-mail address within 24 hours", "\n")
print("> For more information/queries, kindly contact us at team2@gmail.com")
print("\n")
print("> You may also contact us at 38726382/36478283")

print("\n\n\n", "THANK YOU FOR USING OUR SERVICES, WE LOOK FORWARD TO HELPING YOU PLAN YOUR DREAM VACATION ONCE AGAIN".center(155))
print("\n")
com="insert into CUSTOMERINFO values('{}','{}','{}','{}','{}').format(mob,email,ran,name,"ACTIVE")
mycursor.execute(com)
mydb.commit()

```

```

def cancellation():
    #used to cancel a booking
    global cancel
    com="update CUSTOMERINFO set status='CANCELLED' where ID={}".format(cancel)
    mycursor.execute(com)
    mydb.commit()

```

EMPLOYEE FUNCTIONS

```

def rateupdate():
    #used to change conversion rate
    print("\n")
    country=input("> Enter name of country whose conversion rates you would like to edit : ")
    print("\n")
    country=country.upper()
    L=['ratebhd','rateusd','rateinr']
    for i in range(3):
        t=L[i]
        n=input("> Would you like to edit {} (Y/N)".format(t))
        n.upper()
        if n in "Yy":
            print("\n")
            x=float(input("> Enter new conversion rate : "))
            print("\n")

```



```

com="update conversion set {}={} where country='{}'".format(t,x,country)
mycursor.execute(com)
mydb.commit()

```

```

def editflightmult():
    #used to edit flight multipliers
    mycursor.execute("select SLNO,FLIGHTNAME from flights")
    result=mycursor.fetchall()
    print(tabulate(result,headers=['SLNO','FLIGHTNAME'],tablefmt='fancy_grid'))
    print("\n")
    a=input("> Enter the name of the flight whose multiplier is to be edited :")
    mycursor.execute("select * from flights where FLIGHTNAME='{}'".format(a))
    result=mycursor.fetchall( )
    print(tabulate(result,headers=['SLNO','FLIGHTNAME','ECONMULT','BUSINESSMULT'],tablefmt='fancy_grid'))
    print("\n")
    M=input('> Are you sure you wish to edit this flight ? (Y/N) : ')
    print("\n")
    if M in "Yy":
        N=input(">Which multiplier do you wish to edit ? (E/B) :")
        if N in "Ee":
            print("\n")
            X=input("> Enter new multiplier : ")
            print("\n")
            query="update flights set BUSINESSMULT='{}' where FLIGHTNAME='{}'".format(X,a)
            mycursor.execute(query)
            mydb.commit( )
            mycursor.execute("select * from flights where FLIGHTNAME='{}'".format(a))
            result=mycursor.fetchall( )
            print(tabulate(result,headers=['SLNO','FLIGHTNAME','ECONMULT','BUSINESSMULT'],tablefmt='fancy_grid'))
        elif N in "Bb":
            print("\n")
            Y=input("> Enter new multiplier :")
            query="update flights set ECONMULT='{}' where FLIGHTNAME='{}'".format(Y,a)
            mycursor.execute(query)
            mydb.commit()
            mycursor.execute("select * from flights where FLIGHTNAME='{}'".format(a))
            result=mycursor.fetchall( )
            print(tabulate(result,headers=['SLNO','FLIGHTNAME','ECONMULT','BUSINESSMULT'],tablefmt='fancy_grid'))
        elif M in 'Nn':
            print('> You may try again later')

```

```

def edithotelmult():
    # used to edit hotel multipliers
    mycursor.execute("select SLNO,NAME from HOTELS")
    result=mycursor.fetchall( )
    print(tabulate(result,headers=['SLNO','HOTELNAME'],tablefmt="fancy_grid"))
    print("\n")
    b=input("> Enter the name of the hotel whose multiplier is to be edited :")
    mycursor.execute("select * from hotels where NAME='{}'".format(b))
    result=mycursor.fetchall( )

```

```

print("\n")
print(tabulate(result,headers=['SLNO','HOTELNAME','MULTIPLIER'],tablefmt='fancy_grid'))
X=input('> Are you sure you wish to edit this hotel ? (Y/N) : ')
print("\n")
if X in 'Yy':
    print("\n")
    Y=input('>Enter new multiplier :')
    print("\n")
    query="update hotels set MULTIPLIER='{}' where NAME='{}'".format(Y,b)
    mycursor.execute(query)
    mydb.commit()
    mycursor.execute("select * from hotels where NAME='{}'".format(b))
    result=mycursor.fetchall()
    print(tabulate(result,headers=['SLNO','HOTELNAME','MULTIPLIER'],tablefmt='fancy_grid'))
elif X in 'Nn':
    print('>You may try again later')

```

```

def insertcity(identifier):
    #used to add city to countrycity table
    if identifier==1:
        global countryname
    else:
        print("\n")
        countryname=input("> Enter name of country : ")
    print("\n")
    ncity=int(input("> Enter number of cities to be added : "))
    for i in range(ncity):
        print("\n")
        m=input('> Enter name of city to be added : ')
        print("\n")
        n=input('> Enter description of city : ')
        print("\n")
        com="insert into COUNTRYCITY values ('{}','{}','{}'".format(countryname,m,n)
        mycursor.execute(com)
        mydb.commit()

```

```

def inserttourism(identifier):
    #used to add a tourist place to countrytourism table
    if identifier==1:
        global countryname
    else:
        print("\n")
        countryname=input("> Enter name of country where attraction is located : ")
    print("\n")
    ntourism=int(input("> Enter number of tourist attractions to be added : "))
    for i in range(ntourism):
        print("\n")
        t=input("Enter name of tourist attraction to be added : ")
        print("\n")
        com="insert into COUNTRYTOURISM values ('{}','{}'".format(countryname,t)

```

```
mycursor.execute(com)
mydb.commit()
```

```
def insertfares():
```

```
#only used in insertcountry to add new fares
global countryname
print("\n")
a=int(input("> Enter average airfare : "))
print("\n")
h=int(input("> Enter average hotelfare : "))
print("\n")
curry=input("Enter currency name : ")
print("\n")
com="insert into FARES values ('{}','{}','{}','{}').format(countryname,a,h,curry)
mycursor.execute(com)
mydb.commit()
```

```
def updatefares():
```

```
print("\n")
c=input("> Enter name of country where base hotel/airfares are to be edited : ")
print("\n")
h=int(input("> Enter new hotel fare : "))
print("\n")
a=int(input("> Enter new air fare : "))
print("\n")
com="update FARES set AIRFARE={}, HOTELFARE={} where country={}".format(c,a,h)
mycursor.execute(com)
mydb.commit()
```

```
def insertcountry():
```

```
#used to add new country
global countryname
print("\n")
countryname=input('> Enter name of country to be inserted : ')
print("\n")
s=input('> Enter name of continent : ')
print("\n")
n=input('> Enter description of country : ')
print("\n")
com="insert into MAINTABLE values ('{}','{}','{}').format(countryname,s,n)
mycursor.execute(com)
mydb.commit()
i=float(input("> Enter conversion rate to INR : "))
b=float(input("> Enter conversion rate to BHD : "))
d=float(input("> Enter conversion rate to USD : "))
newcom="insert into CONVERSION values ('{}','{}','{}','{}').format(countryname,b,d,i)
mycursor.execute(newcom)
mydb.commit()
```

```

insertcity(1)
inserttourism(1)
insertfares()

def updatecitydesc():
    #used to edit description of city
    print("\n")
    m=input('> Enter name of city to be edited : ')
    print("\n")
    s=input('> Enter new description : ')
    print("\n")
    com="update COUNTRYCITY set description='{}' where city='{}'.format(s,m)
    mycursor.execute(com)
    mydb.commit()

def updatecountrydesc():
    print("\n")
    c=input('> Enter name of country to be edited : ')
    print("\n")
    s=input('> Enter new description : ')
    print("\n")
    com="update MAINTABLE set description='{}' where country='{}'.format(s,c)
    mycursor.execute(com)
    mydb.commit()

def displaycustomers():
    com="select * from customerinfo"
    mycursor.execute(com)
    lists=mycursor.fetchall()
    print("\n")
    print(tabulate(lists,headers=["MOBILE","EMAIL","ID","NAME","STATUS"],tablefmt='fancy_grid'))

def customersearch():
    print("\n")
    Id=int(input("Enter booking ID of required customer : "))
    print("\n")
    com="select * from customerinfo where ID='{}'.format(Id)
    mycursor.execute(com)
    lists=mycursor.fetchall()
    print(tabulate(lists,headers=["MOBILE","EMAIL","ID","NAME","STATUS"],tablefmt='fancy_grid'))

# START OF PROGRAM #
while True:
    mainmenu()
    if n==0:
        #END PROGRAM

```

```

break
if n==1:
    #CONSUMER BOOKING
    maintable()
    deco()
    CONSUMERSCREEN1()
    print("\n")
    print("> In order to return to main menu, kindly type 0 in the following field ::")
    print("\n")
    x=int(input("> To plan your itinerary, kindly enter the duration of your stay (in days) :"))
    print("\n")
    if x==0:
        continue
    y=int(input("> Enter number of travellers :"))
    print("\n")
    z=int(input("> Enter number of hotel rooms required :"))
    print("\n")
    deco()
    hotelsandflights()
    deco()
    print("\n")
    print("> In order to return to main menu, kindly type 0 in the following field ::")
    print("\n")
    a=int(input("> Enter 1 to convert into BHD / 2 to convert into USD :"))
    print("\n")
    if a==0:
        continue
    deco()
    conversion=convertor()
    r=conversion[0]
    currency=conversion[1]
    print("\n")
    print("> Total flight cost (for all travellers) is :",totalair*r,currency,"\n")
    print("> Total hotel cost (for all travellers) is :",totalhotel*r,currency,"\n")
    print("> Total cost of trip for all travellers is estimated at :",(totalair*r)+(totalhotel*r),currency)
    print("\n")
    deco()
    print("\n")
    print("> In order to return to main menu, kindly type 0 in the following field ::")
    print("\n")
    mob=int(input("> In order to book your trip, kindly enter your mobile number :"))
    print("\n")
    if mob==0:
        continue
    booking()
    continue
if n==2:
    #CANCELLATION
    print("\n")
    cancel=input("> In order to cancel your trip, kindly enter your booking ID :")

```

```

cancellation()
deco()
print("\n")
print("> Your booking has been cancelled successfully","\n")
print("> The refund shall be made directly to your account ", "\n")
print("> To convey any grievances regarding our services, kindly contact us at team2complaints@gmail.com")
print("\n")
print("> You may also contact us at 38726382/36478283")
print("\n")
if n==3:
    #EMPLOYEE PROGRAM
    #PASSWORD = '1234'
    deco()
    j=0
    while True:
        passwd=input("ENTER PASSWORD : ")
        if passwd!="1234":
            print('Incorrect')
            k=int(input("Enter '1' to retry, enter '2' to return to main menu : "))
            if k==1:
                deco()
                continue
            if k==2:
                j=1
                break
        j=0
        break
    if j==1:
        continue
    #MENU FOR EMPLOYEES
    while True:
        deco()
        print("\n")
        exitvariable=0
        print("> Press '1' to add a new country <","\n")
        print("> Press '2' to add a new city <","\n")
        print("> Press '3' to add new tourist attractions <","\n")
        print("> Press '4' to modify a country description <","\n")
        print("> Press '5' to modify a city description <","\n")
        print("> Press '6' to edit conversion rate <","\n")
        print("> Press '7' to edit a flight multiplier <","\n")
        print("> Press '8' to edit a hotel multiplier","\n")
        print("> Press '9' to edit hotel/airline fares <","\n")
        print("> Press '10' to view all customer details <","\n")
        print("> Press '11' to view details of a single customer <","\n")
        print("> Press '12' if you wish to return to the main menu <","\n")
        print("\n")
        deco()
        choice=int(input("> Kindly enter your choice : "))
        deco()

```

```
print("\n")

if choice==1:
    insertcountry()
elif choice==2:
    insertcity(0)
elif choice==3:
    inserttourism(0)
elif choice==4:
    updatecountrydesc()
elif choice==5:
    updatecitydesc()
elif choice==6:
    rateupdate()
elif choice==7:
    editflightmult()
elif choice==8:
    edithotelmult()
elif choice==9:
    updatefares()
elif choice==10:
    displaycustomers()
elif choice==11:
    customersearch()
elif choice==12:
    exitvariable=1
else:
    print("> Invalid choice, please try again ")
if exitvariable==1:
    break
continue
```


ACKNOWLEDGEMENT

I would like to express my gratitude towards my computer science teacher Mr. Pius Mathew for guiding me immensely throughout the course of this project. His mentorship and advice has been largely responsible for the successful completion of this project.

I would also like to thank my teammates for their contributions during this project.

CONCLUSION

This project demonstrates the remarkable ability of python to handle large amounts of data efficiently, and with ease for both the programmer, and the consumer. A python connection with MySQL can be used to handle extensive databases for a variety of purposes such as travel and tourism as has been shown here.