CS Project

AIRLINE RESERVATION SYSTEM

Done by Pranav Karthikeyan Aiyyer
Class: 12-K
Submitted to : Mr Pius Mathew

Introduction

The airline reservation system project is based on most essential features of Python and MySQL. The project is a combination of use of python programming with RDBMS structure of MySQL tables. This system/software will be used by a travel agent to do the following tasks such as booking a ticket for a passenger, updating the passenger type (Economy/Business) and date of travel of the passenger, cancelling the booking of the customer, ticket printing, getting the passenger reports, storing the customer's details and other booking related tasks and operations. The front end of the project python and back end is based on MySQL. This program includes different features of python such as loops, strings, tuples, concatenations, indexing, nested loops, jump statements etc. For the MySQL part, a lot of DDL commands and complex DML commands are used. The overall program will be in a menu-driven format.

Index

Certificate

This is to certify that Pranav Karthikeyan Aiyyer, a student of XII-K has successfully completed the research on the below mentioned Computer Science project under the guidance of Mr. Pius Mathew during year 2021-2022. The Project is a part of the internal assessment of CBSE term 2 examination

Signature of the CS HOD: **Signature of CS teacher:**

Acknowledgement

I would like to express my special thanks of gratitude to my teacher Mr. Pius Mathew, who gave me the golden opportunity to do this wonderful software development project on "Airline Reservation system". I would like to thank my teammates - Muhammad Zaid Musaddique Muckba, Adithya Rajashekara Das and Ryan Joswin Cutinho for their contribution. I learnt a lot of new things after doing this project.

Pranav Karthikeyan Aiyyer

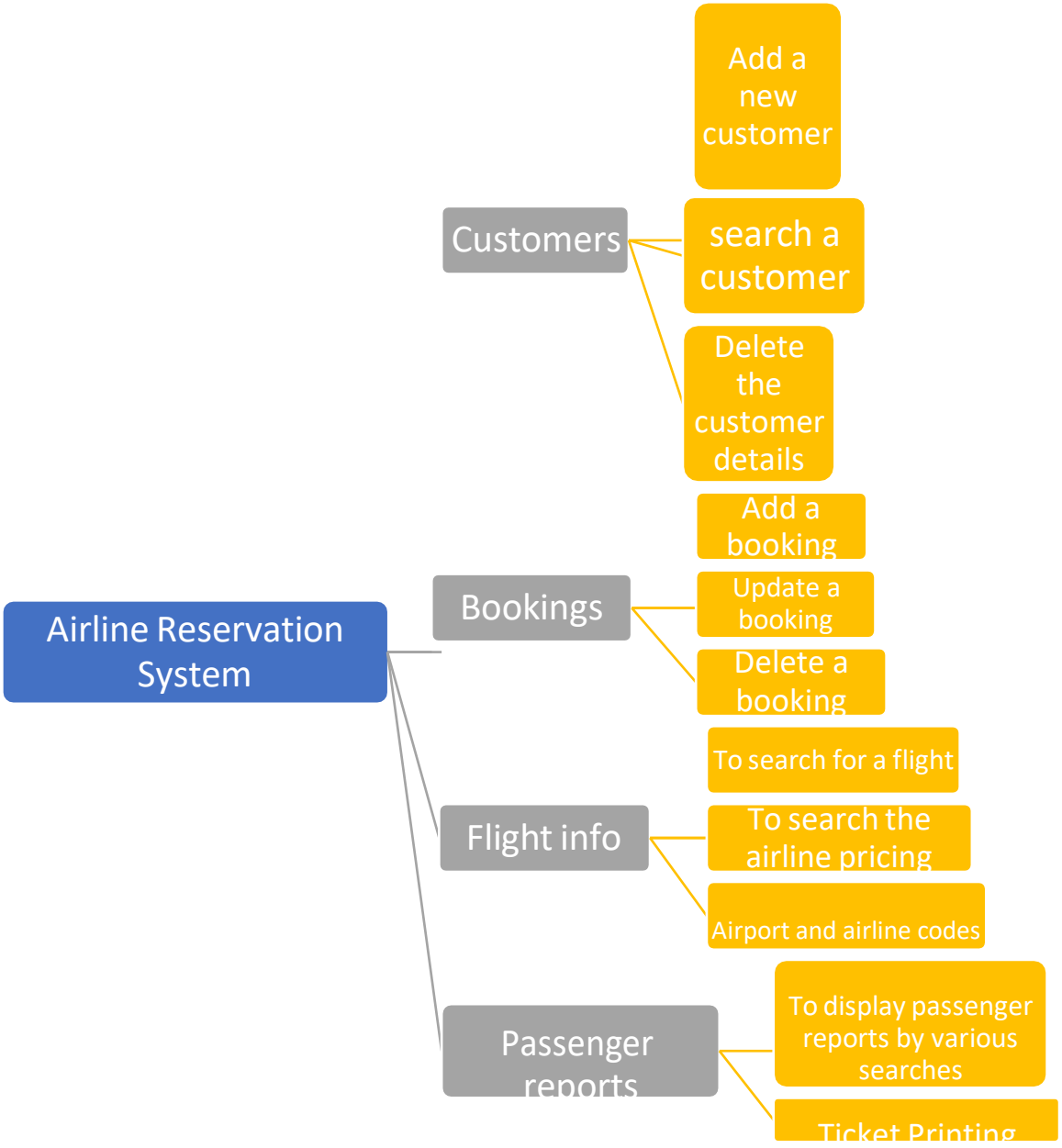
XII-K Computer Science

Gr no. 34497

System Analysis

With the help of well-designed functions, the admin can manage the tasks related to the booking of the passenger, storing the airline information, updating the booking details, cancelling the booking details, searching the passenger for a particular PNR number, ticket printing, storing customer details etc. The database has 8 tables. First 2 tables are to store the airline codes and airport codes. The next 2 tables are to store the flight details and the flight pricing of the airlines. The 5th table is to store the customer's personal details such as his/her address, email address, contact number, Name etc. Therefore, the primary key in this fifth table will be a unique customer ID. The last 3 tables are there to store the passenger's booking information and flight booking information such as ticket number, Agency and airline PNR number, flight number and departure time and departure date, date of booking of the passenger, departure city and arrival city of the passenger.

System Design



Source Code

```
# Module bookings

import mysql.connector

from tabulate import tabulate

import datetime


''' The function below does the booking for a customer. There are 3 query statements in the function along with 3

records because the MySQl tables named 'booking_details','booking_flight_details' and 'customer_bookings' are

interrelated to each other by a unique Agency PNR of the customer, so the records get appended in the those 3 tables

simultaneously in a single function.'''

def customer_booking():


mydb=mysql.connector.connect(host="localhost",user="root",password="786Mm786",database="airline_reservation_system")

   mycursor=mydb.cursor()

   mycursor.execute("select max(AGNCY_PNR) from booking_details;")

   result=mycursor.fetchall()# this code helps us to get the idea about which was the last agency pnr number

   for k in result:

      for r in k:

         Sum=int(r)+1

   AGNCY_PNR=str(Sum)

   P_title=input("Enter customer title:")
```

```
P_Fname=input("Enter First Name:")
P_Mname=input("Enter the Middle name:")
P_Lname=input("Enter Last Name:")
P_Gender=input("Enter Gender:")


mycursor.execute("select max(P_TKTno) from booking_details;")
result=mycursor.fetchall()# the code gives the last ticket number of the passenger
for k in result:
    for r in k:
        Sum=int(r)+1
P_TKTno=str(Sum)
Book_Status="A"
if P_Mname=='':
    P_name=P_Fname+' '+P_Lname
else:
    P_name=P_Fname+' '+P_Mname+' '+P_Lname# in the table customers, the name is in single column but in the table booking_details it is in 3 seperate columns
                        # therefore the line merges the 3 columns into one so thet the customer Id from the customers table can be
                        # given to the travel agent for booking.


mycursor.execute("select cust_Id from customers where cust_name like '%{}%'".format(P_name))
```

```
rt=mycursor.fetchall()
for k in rt:
    for r in k:
        print()
        print()
        print("The required customer ID is:",r)
CUST_ID=input("Enter customer ID:")
```

```
mycursor.execute("select max(AL_PNR) from customer_bookings;")
result=mycursor.fetchall()# this code helps us to get the idea about which was the last Airline pnr number
for k in result:
    for r in k:
        Sum=int(r)+1
AL_PNR=str(Sum)
```

```
    Book_Dt=datetime.date.today()
    flgt_dep_city=input("Enter departure city:")
    flgt_Arr_city=input("Enter arrival city:")
    mycursor.execute("select flight_no from flight_details where dep_city='{}' and arr_city='{}'".format(flgt_dep_city, flgt_Arr_city))
    nateejas=mycursor.fetchall()
    for k in nateejas:
        for r in k:
            print()
            print()
            print("The available flights are", r, end=' ')
    print()
    print()
    print() # helps the travel agent to get the required flight No for a particular departure and arrival city


    Flight_no=input("Enter the flight number")
    mycursor.execute("select * from airline_flight_pricing where dep_city='{}' and Arr_city='{}'".format(flgt_dep_city,flgt_Arr_city))
    result=mycursor.fetchall()
    print("The available flight pricing details for your destination are as follows")
    print(tabulate(result,headers=['Al No','Al Name','Dep City','Arr City','ADT Eco','ADT First','ADT bsns','CHD Eco','CHD First','CHD bsns','Crncy','ICAO_code'],tablefmt='fancy_grid'))
    print()
    print()#gives the pricing details for a particular departure and arrival city
```

```
    P_type=input("Enter passenger type:")
    TKT_price=input("Enter ticket price:")
    mycursor.execute("select Flight_no,OP_day,dep_time from flight_details
where dep_city='{}' and Arr_city='{}' and
flight_no='{}'".format(flgt_dep_city,flgt_Arr_city,Flight_no))
    results=mycursor.fetchall()
    print("The following is the operating day and the departure time for the
required flight")
    print(tabulate(results,headers=["Flight no","Day","Departure Time"],
tablefmt="fancy_grid"))
    print()
    print()


# gives the operating day and departure time for a particular flight number


    flgt_dep_date=input("Enter departure date (YYYY-MM-DD):")
    flgt_Arr_date=input("Enter arrival date (YYYY-MM-DD):")
    dep_time=input("Enter the time of departure of the flight")
    flgt_status="A"
    rec1=(CUST_ID,AGNCY_PNR,AL_PNR,Book_Dt)
```

```
   query1="insert into
customer_bookings(CUST_ID,AGNCY_PNR,AL_PNR,Book_Dt)values(%s,%s,%s,%s
);"
   mycursor.execute(query1,rec1)
   mydb.commit()



rec2=(AGNCY_PNR,P_title,P_Fname,P_Mname,P_Lname,P_Gender,P_type,TKT_
price,P_TKTno,Book_Status)
   query2="insert into
booking_details(AGNCY_PNR,P_title,P_Fname,P_Mname,P_Lname,P_Gender,P_
type,TKT_price,P_TKTno,Book_Status)values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s);
"
   mycursor.execute(query2,rec2)
   mydb.commit()



rec3=(AGNCY_PNR,AL_PNR,flgt_dep_date,flgt_dep_city,flgt_Arr_date,flgt_Arr_c
ity,flgt_status,Flight_no,dep_time)
   query3="insert into
booking_flight_details(AGNCY_PNR,AL_PNR,flgt_dep_date,flgt_dep_city,flgt_Ar
r_date,flgt_Arr_city,flgt_status,Flight_no,dep_time)values(%s,%s,%s,%s,%s,%s,
%s,%s,%s,%s);"
   mycursor.execute(query3,rec3)
   mydb.commit()
   print()
   print()
```

```
    print("Booking in process",end=' ')
    for k in range(30000000):
        pass
    print(".",end=' ')
    for k in range(30000000):
        pass
    print(".", end=' ')
    for k in range(30000000):
        pass
    print(".", end=' ')
    for k in range(30000000):
        pass
    print(".", end=' ')
    for k in range(30000000):
        pass
    print(".", end=' ')
    for k in range(100000000):
        pass
    print()
    print()
    print("The booking was successful")

# function to update the ticket price of a passenger
def Update_TicketPrice():
```

```
   mydb=mysql.connector.connect(host='localhost', user ='root',
password='786Mm786', database="airline_reservation_system")

   mycursor=mydb.cursor()

   AGNCY_PNR=input("Enter the PNR No of the customer")

   mycursor.execute("Select * from booking_flight_details where
AGNCY_PNR='{}'".format(AGNCY_PNR))

   result=mycursor.fetchall()

   for k in result:

      dep_city=k[3]

      arr_city=k[5]

      flight_no=k[7]


   mycursor.execute("select * from airline_flight_pricing where dep_city='{}' and
Arr_city='{}'".format(dep_city,arr_city))

   results=mycursor.fetchall()

   print("The following is the pricing information")

   print(tabulate(results,headers=['Al No','Al Name','Dep City','Arr City','ADT
Eco','ADT First','ADT bsns','CHD Eco','CHD First','CHD
bsns','Crncy','ICAO_code'],tablefmt='fancy_grid'))

   print()

   print()

   print("The customer's flight number is:",flight_no)

   P_type=input("Enter the new passenger type")

   TKT_price=int(input("Enter the new ticket price"))

   mycursor.execute("Update booking_details set TKT_price={} where
AGNCY_PNR= '{}' ".format(TKT_price, AGNCY_PNR))
```

```
    mydb.commit()
    mycursor.execute("Update booking_details set P_type='{}' where
AGNCY_PNR= '{}' ".format(P_type, AGNCY_PNR))
    print("The ticket price and the passenger type has been updated")
    mydb.commit()



# function to update the travel date of the passenger
def Update_TravelDate():
    mydb=mysql.connector.connect(host='localhost', user ='root',
password='786Mm786', database="airline_reservation_system")
    mycursor=mydb.cursor()
    AGNCY_PNR=input("Enter the PNR No of the customer")
    mycursor.execute("Select * from booking_flight_details where
AGNCY_PNR='{}'".format(AGNCY_PNR))
    result=mycursor.fetchall()
    print("The following is the booking whose travel dates are going to be
updated:")
    print(tabulate(result,headers=["AGNCY_PNR","AL_PNR","Dep date","Dep
City","Arrival date","Arrival city","STatus", "Flight No", "Dep Time"],
tablefmt='fancy_grid'))
    for k in result:
        flight_no=k[7]
```

```
    mycursor.execute("Select OP_Day from flight_details where
flight_no='{}'".format(flight_no))
    results=mycursor.fetchall()
    for k in results:
        for r in k:
            print("the operating days of this flight are..", r)
    print()
    print()
    Departure_date=input("Enter the new departure date")
    Arrival_date=input("Enter the new arrival date")
    mycursor.execute("Update booking_flight_details set flgt_dep_date ='{}'
where AGNCY_PNR='{}'".format(Departure_date,AGNCY_PNR))
    mydb.commit()
    mycursor.execute("Update booking_flight_details set flgt_Arr_date ='{}'
where AGNCY_PNR='{}'".format(Arrival_date,AGNCY_PNR))
    mydb.commit()
    print("The travel dates have been updated")



# function to delete the booking of a passenger
def Delete_booking():
    mydb=mysql.connector.connect(host='localhost', user ='root',
password='786Mm786', database="airline_reservation_system")
```

```
    mycursor=mydb.cursor()


    AGNCY_PNR=input("Enter the Agency PNR number of the passenger whose
booking is to be cancelled")
    mycursor.execute("Select * from booking_details where
AGNCY_PNR='{}'".format(AGNCY_PNR))
    result=mycursor.fetchall()
    print("The following booking will be deleted:")

print(tabulate(result,headers=['AGNCY_PNR','P_Title','P_Fname','P_Mname','P_
Lname','P_Gender','P_Type','TKT_price','P_TKTno','Book_Status'],tablefmt='fanc
y_grid'))
    warning=input("Are you sure that you want to delete the booking for this
customer?  (y/n)")
    if warning=='y':
        mycursor.execute("Delete from customer_bookings where
AGNCY_PNR='{}'".format(AGNCY_PNR))
        mydb.commit()
        mycursor.execute("Delete from booking_flight_details where
AGNCY_PNR='{}'".format(AGNCY_PNR))
        mydb.commit()
        mycursor.execute("Delete from booking_details where
AGNCY_PNR='{}'".format(AGNCY_PNR))
        mydb.commit()
        print()
        print()
        print("The booking was deleted")
```

```
    elif warning=='n':
        print()
        print()
        print("The booking was retained")
    else:
        print()
        print()
        print("Wrong option entered.... failed to delete the booking")
```

```
# Module flight_info
import mysql.connector
from tabulate import tabulate


# function for displaying the details of flights for a paticular departure city and city of arrival
def flight_details():
```

```
mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   Dep_City=input("Enter the city of departure")
   Arr_City=input("Enter the city of arrival")
   mycursor.execute("select * from flight_details where Dep_city='{}' and
Arr_city='{}'".format(Dep_City,Arr_City))
   result=mycursor.fetchall()
   print(tabulate(result,headers=['Al Code','Al Name','flight no','OP day','Dep
city','Dep time','Arr city','Arr time','flight
hrs','ICAO_code'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)

# function for displaying the details of all the flights
def flight_details_of_all():

mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   mycursor.execute("select * from flight_details")
   result=mycursor.fetchall()
   print(tabulate(result,headers=['Al Code','Al Name','flight no','OP day','Dep
city','Dep time','Arr city','Arr time','flight
hrs','ICAO_code'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)
```

```
# function for displaying the airline flight pricing for a paticular departure city and city of arrival

def airline_flight_pricing():


mydb=mysql.connector.connect(host="localhost",user='root',password='786Mm786',database="airline_reservation_system")

   mycursor=mydb.cursor()

   Dep_City=input("Enter the city of departure")

   Arr_City=input("Enter the city of arrival")

   mycursor.execute("select * from airline_flight_pricing where Dep_city='{}' and Arr_City='{}'".format(Dep_City,Arr_City))

   result=mycursor.fetchall()

   print(tabulate(result,headers=['Al No','Al Name','Dep City','Arr City','ADT Eco','ADT First','ADT bsns','CHD Eco','CHD First','CHD bsns','Crncy','ICAO_code'],tablefmt='fancy_grid'))

   print(mycursor.rowcount)


# function for displaying the airline flight pricing of all the flights

def airline_flight_pricing_of_all():


mydb=mysql.connector.connect(host="localhost",user='root',password='786Mm786',database="airline_reservation_system")

   mycursor=mydb.cursor()

   mycursor.execute("select * from airline_flight_pricing")

   result=mycursor.fetchall()
```

```
   print(tabulate(result,headers=['Al No','Al Name','Dep City','Arr City','ADT
Eco','ADT First','ADT bsns','CHD Eco','CHD First','CHD
bsns','Crncy','ICAO_code'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)


# function for displaying the airline codes of all the airlines
def airline_codes():

mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   com='select * from airline_codes;'
   mycursor.execute(com)
   result=mycursor.fetchall()
   print(tabulate(result,headers=['Airline code','Airline Name','ICAO
code'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)


# function for displaying the airport codes of all the airports
def airport_codes():

mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   com='select * from airport_codes'
   mycursor.execute(com)
```

```
    result=mycursor.fetchall()
    print(tabulate(result,headers=["APcode","APname"],tablefmt='fancy_grid'))
    print(mycursor.rowcount)
```

```
# Module customers
import mysql.connector
from tabulate import tabulate
```

```
# function for appending the details of the customer
def customer_append():

mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
    mycursor=mydb.cursor()
    Cust_ID=input("Enter the Customer ID")
    Cust_Name=input("Enter the customer's name")
    Cust_add=input("Enter the customer's address")
    Cust_City=input("Enter the customer's City")
```

```
    Cust_Country=input("Enter the customer's country")
    Cust_Phone=input("Enter the customer's contact number")
    Cust_email=input("Enter the customer's email address")

rec=(Cust_ID,Cust_Name,Cust_add,Cust_City,Cust_Country,Cust_Phone,Cust_email)
    query="insert into customers(Cust_ID,Cust_Name,Cust_add,Cust_City,Cust_Country,Cust_Phone,Cust_email) values(%s,%s,%s,%s,%s,%s,%s);"
    mycursor.execute(query,rec)
    mydb.commit()
    print()
    print()
    print("The customer details has been added successfully in the table")



# function for displaying the details of all the customer
def customer_detailsofall():

mydb=mysql.connector.connect(host="localhost",user='root',password='786Mm786',database="airline_reservation_system")
    mycursor=mydb.cursor()
    com='select * from customers;'
    mycursor.execute(com)
    result=mycursor.fetchall()
```

```
print(tabulate(result,headers=['Cust_ID','Cust_Name','Cust_Add','Cust_City','Cust_Country','Cust_Phone','Cust_email'],tablefmt='fancy_grid'))
   print(mycursor.rowcount,'rows affected')


# function for displaying all the details of a customer for a particular customer ID
def customer_details():

mydb=mysql.connector.connect(host="localhost",user='root',password='786Mm786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   Cust_Name=input("Enter the Customer's name")
   mycursor.execute("select * from customers where Cust_name Like'%{}%'".format(Cust_Name))
   result=mycursor.fetchall()

print(tabulate(result,headers=['Cust_ID','Cust_Name','Cust_Add','Cust_City','Cust_Country','Cust_Phone','Cust_email'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)


# function for deleting the details of the customer
def customer_delete():
```

```
mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   Cust_ID=input("Enter the ID of the customer")
   query="delete from customers where Cust_ID=%s"
   records=(Cust_ID,)
   mycursor.execute("select * from customers where
CUST_ID='{}'".format(Cust_ID))
   result=mycursor.fetchall()
   print("This is the customer whose details you want to delete:")

print(tabulate(result,headers=['Cust_ID','Cust_Name','Cust_Add','Cust_City','Cus
t_Country','Cust_Phone','Cust_email'],tablefmt='fancy_grid'))
   print()
   print()
   warning=input("Are you sure that you want to delete the details of this
customer?  (y/n):")
   if warning=='y':
      mycursor.execute(query,records)
      mydb.commit()
      print("The customer details has been deleted")
   elif warning=='n':
      print()
      print("The customer details were retained")
```

```
# Module passenger_reports

import mysql.connector

from tabulate import tabulate



# function to display the reports of all the customers

def reports_of_all_customers():


mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")

   mycursor=mydb.cursor()

   com='select * from booking_details natural join booking_flight_details order
by AL_PNR;'

   mycursor.execute(com)

   result=mycursor.fetchall()


print(tabulate(result,headers=['AGNCY_PNR','P_Title','P_Fname','P_Mname','P_
Lname','P_Gender','P_Type','TKT_price','P_TKTno','Book_Status','AL_PNR','Flgt_
Dep_Date','Flgt_dep_city','Flgt_Arr_Date','Flgt_Arr_city','Flgt_status','Flight
no','Dep Time'],tablefmt='fancy_grid'))

   print(mycursor.rowcount,"rows affected")
```

```
# function to print the ticket of a passenger
def ticket_printing():

mydb=mysql.connector.connect(host="localhost",user="root",password="786M
m786",database="airline_reservation_system")
   mycursor=mydb.cursor()
   P_tktno=input("Enter ticket number:")
   print("
_____
_____
_____")

print("|=============================================================
==========================================================================
==================|")
   print("|                                        ✈✈✈ **** TICKET **** ✈✈✈
|")

print("|=============================================================
==========================================================================
==================|")
   print("|
|")
   mycursor.execute("select
P_title,P_Fname,P_Mname,P_Lname,P_type,P_tktno,Flight_no,AL_PNR,flgt_de
p_date,dep_time,flgt_dep_city,flgt_arr_city from
booking_details,booking_flight_details where
```

```
booking_details.AGNCY_PNR=Booking_flight_details.AGNCY_PNR and
P_tktno='{}'".format(P_tktno))
   result=mycursor.fetchall()

print(tabulate(result,headers=['P_title','P_Fname','P_Mname','P_Lname','P_Typ
e','ticket number','Flight number','PNR
no.','Date','Time','From','To'],tablefmt='fancy_grid'))
   print("|
|")
   print("|
|")
   print("|
|")
   print("|     Gate closes 20 minutes before departure
|")
   print("|     Please adhere to the airport 1 hour before the departure time.
|")

print("|_____
_____
_____|")
   print("|                                        ********  HAVE A PLEASANT JOURNEY
********                                |")

print("|=====================================================
=============================================================
==================|")

print("|_____
```

```
_____|")
   print(mycursor.rowcount)



#function to search the passenger for a particular Agency PNR number
def search_by_PNR():

mydb=mysql.connector.connect(host="localhost",user="root",password="786M
m786",database="airline_reservation_system")
   mycursor=mydb.cursor()
   AGNCY_PNR=input("Enter the Agency PNR number of the passenger")
   mycursor.execute("select * from booking_details natural join
booking_flight_details where
booking_details.AGNCY_PNR='{}'".format(AGNCY_PNR))
   result=mycursor.fetchall()

print(tabulate(result,headers=['AGNCY_PNR','P_Title','P_Fname','P_Mname','P_
Lname','P_Gender','P_Type','TKT_price','P_TKTno','Book_Status','AL_PNR','Flgt_
Dep_Date','Flgt_dep_city','Flgt_Arr_Date','Flgt_Arr_city','Flgt_status','Flight
no','Dep Time'],tablefmt='fancy_grid'))
   print(mycursor.rowcount)



# function to search the passengers travelling from a particular airline
```

```
def reports_of_all_customers_travellingby_a_flight():

mydb=mysql.connector.connect(host="localhost",user='root',password='786M
m786',database="airline_reservation_system")
   mycursor=mydb.cursor()
   flight_no=input("Enter the flight number whose passenger's reports you want
to get")
   com="select * from booking_details natural join booking_flight_details where
flight_no='{}' order by Flgt_dep_date;".format(flight_no)
   mycursor.execute(com)
   result=mycursor.fetchall()

print(tabulate(result,headers=['AGNCY_PNR','P_Title','P_Fname','P_Mname','P_
Lname','P_Gender','P_Type','TKT_price','P_TKTno','Book_Status','AL_PNR','Flgt_
Dep_Date','Flgt_dep_city','Flgt_Arr_Date','Flgt_Arr_city','Flgt_status','Flight
no','Dep Time'],tablefmt='fancy_grid'))
   print(mycursor.rowcount,"rows affected")



# Main menu
import mysql.connector
from tabulate import tabulate
from bookings import *
from customers import *
from passenger_reports import *
from flight_info import *
```

```
import datetime
while True:
   print()
   print()
   print()

print("+============================================================
==========================+")
   print('|                          PARZ TRAVELS                          |')

print("|============================================================
==========================|")
   print("|                  MENU FOR AIRLINE RESERVATION SYSTEM
|")

print("|============================================================
==========================|")
   print("|1.) Customer bookings                                    |")
   print("|2.) Airlines and flight info                             |")
   print("|3.) Customer details and info                            |")
   print("|4.) Passenger reports                                    |")
   print("|5.) EXIT                                                 |")

print("+____________________________________________________________
_____________________________+")
   option=int(input("Enter your option"))
   if option==1:
```

```
    while True:
        print()
        print()
        print()

print("+============================================================
==========================+")
        print('|                        CUSTOMER BOOKINGS                        |')

print("+============================================================
==========================+")
        print("|1.) Booking a new customer                                        |")
        print("+2.) Updating the ticket price of the passenger
+")
        print("|3.) Updating the departure and arrival date of the passenger
|")
        print("+4.) Deleting the customer booking                               +")
        print("|5.) back                                                        |")

print("+____________________________________________________________
_____________________________+")
        option=int(input("Enter your option"))
        if option==1:
           customer_booking()
        elif option==2:
           Update_TicketPrice()
        elif option==3:
```

```
            Update_TravelDate()
        elif option==4:
            Delete_booking()
        elif option==5:
            break
        else:
            print("Your option does not exist")
    elif option==2:
        while True:
            print()
            print()
            print()

print("+=============================================================
==============================================+")
            print('|                          AIRLINES & FLIGHT INFO
|')

print("+=============================================================
==============================================+")
            print("|1.) To display the flight details of all airlines
|")
            print("+2.) To display the flight details of the airlines for a particular
departure city and arrival city        +")
            print("|3.) To display the airline flight pricing of all the airlines
|")
```

```
        print("+4.) To display the airline flight pricing of the airlines for a
particular departure city and arrival city +")
        print("|5.) To display the airline codes
|")
        print("+6.) To display the airport codes
+")
        print("|7.) back                                                    |")

print("+=============================================================
==========================================+")
        option=int(input("Enter your option"))
        if option==1:
           flight_details_of_all()
        elif option==2:
           flight_details()
        elif option==3:
           airline_flight_pricing_of_all()
        elif option==4:
           airline_flight_pricing()
        elif option==5:
           airline_codes()
        elif option==6:
           airport_codes()
        elif option==7:
           break
        else:
```

```
            print("Your option does not exist")
    elif option==3:
        while True:
            print()
            print()
            print()

print("+=============================================================
====================+")
            print('|                    CUSTOMER DETAILS & INFO                    |')

print("+=============================================================
====================+")
            print("|1.) To append the customer details                              |")
            print("+2.) To display the details of all the customers                    +")
            print("|3.) To search the details of a customer by Name
|")
            print("+4.) To delete the details of a customer                           +")
            print("|5.) back                                                  |")

print("+=============================================================
====================+")
            option=int(input("Enter your option"))
            if option==1:
                customer_append()
            elif option==2:
```

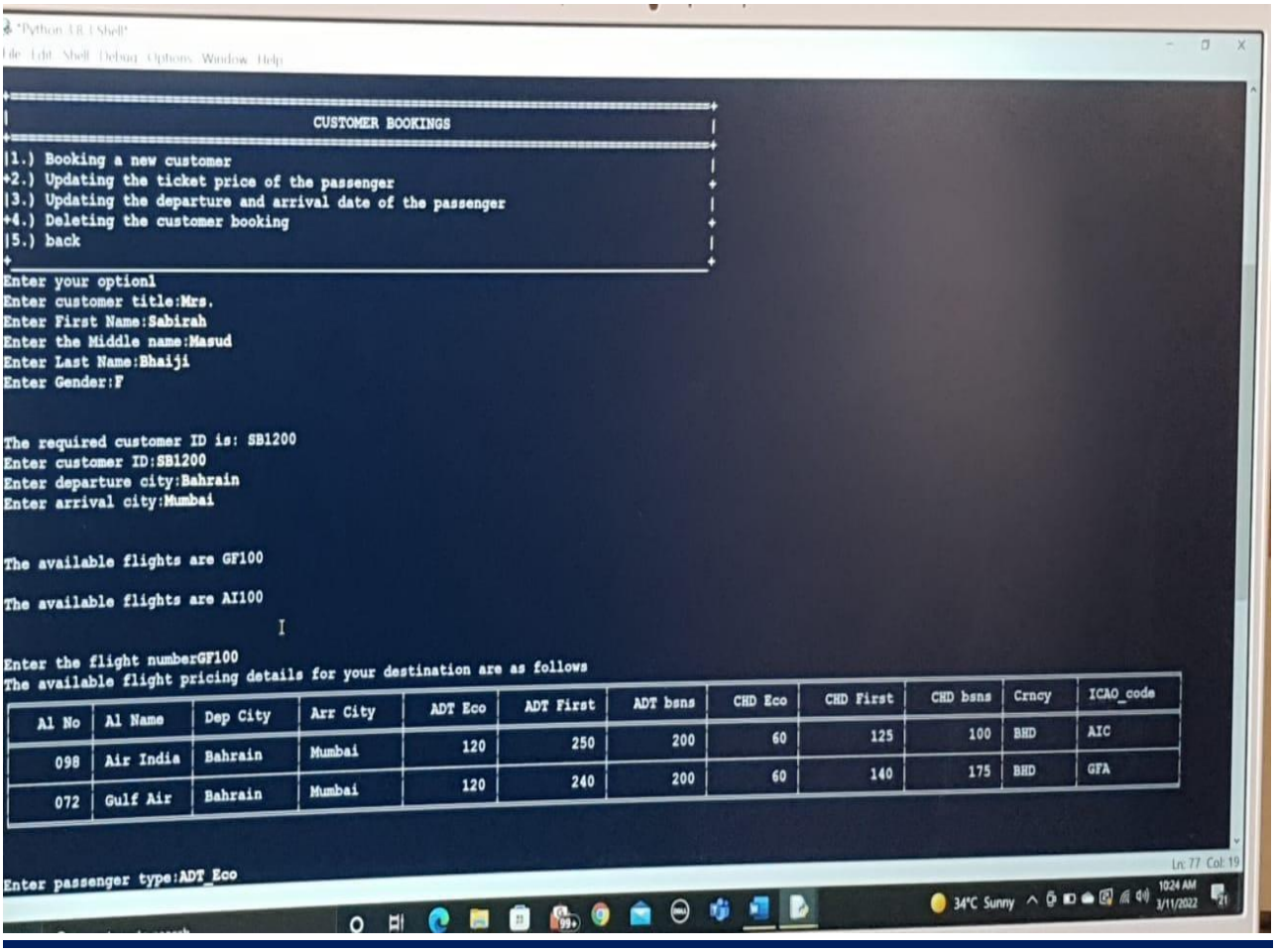
```
            customer_detailsofall()
        elif option==3:
            customer_details()
        elif option==4:
            customer_delete()
        elif option==5:
            break
        else:
            print("Your option does not exist")
    elif option==4:
        while True:
            print()
            print()
            print()

print("+=============================================================
============================+")
            print('|                        PASSENGER REPORTS                          |')

print("+=============================================================
============================+")
            print("|1.) To display the reports of all passengers
|")
            print("+2.) To display the details of the passenger for a particular agency
PNR number        +")
```

```
        print("|3.) To print the ticket of the passenger
|")
        print("|4.) To display the reports of all the passengers of a particular
flight            |")
        print("+5.) back                                                    +")

print("+============================================================
============================+")
        option=int(input("Enter your option"))
        if option==1:
           reports_of_all_customers()
        elif option==2:
           search_by_PNR()
        elif option==3:
           ticket_printing()
        elif option==4:
           reports_of_all_customers_travellingby_a_flight()
        elif option==5:
           break
        else:
           print("Your option does not exist")
   elif option==5:
      print()
      print()
      print()
      print("Thank You for using our software :)")
```

```
    break
else:
    print("Your option does not exist")
```

Sample Output

#Booking



#Booking

```
Enter passenger type:ADT_Eco
Enter ticket price:120
The following is the operating day and the departure time for the required flight
```

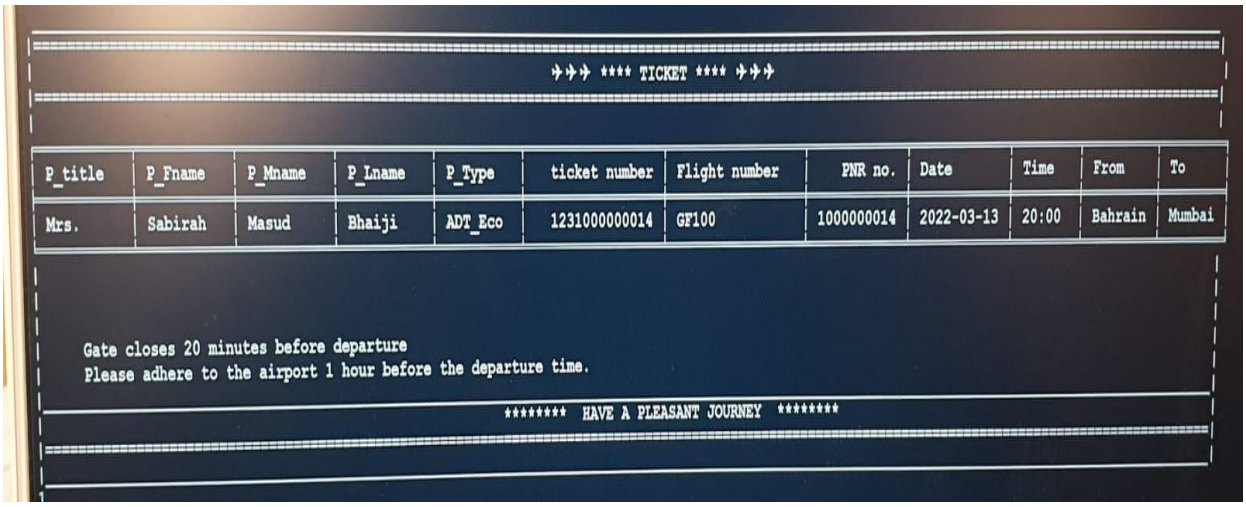
Flight no	Day	Departure Time
GF100	Sunday	20:00

```
Enter departure date (YYYY-MM-DD):2022-03-13
Enter arrival date (YYYY-MM-DD):2022-03-13
Enter the time of departure of the flight20:00

Booking in process . . . . .

The booking was successful
```

#Customer details

Cust_ID	Cust_Name	Cust_Add	Cust_City	Cust_Country	Cust_Phone	Cust_email
A4290	Adithya Rajashekara Das	Roadno-819 blockno-908	West Riffa	Bahrain	36235569	adithya1029das@gmail.com
AA4444	Atif Aslam	Roadno-128 blockno-969	Karachi	Pakistan	12567989	Atif@gmail.com
AI4290	Aiden Panvalkar	Roadno-822 blockno-900	East Riffa	Bahrain	36545579	aiden@gmail.com
AN4290	Anushid	Roadno-867 blockno-420	Muharraq	Bahrain	36995579	anushid@gmail.com
AS4290	Amit Shibu	Roadno-827 blockno-920	Muharraq	Bahrain	39095579	amit@gmail.com
AW4290	Ashwin Prem	Roadno-857 blockno-490	Manama	Bahrain	31995579	ashwin@gmail.com
BP0000	Brad Pitt	Roadno-121 blockno-977	San Francisco	USA	13566989	Pitt@gmail.com
BS2345	Ben Stokes	Roadno-645 blockno-977	Manchester	United Kingdom	13577991	Stokes@gmail.com
CC4290	Clement Cammillus	Roadno-897 blockno-590	Manama	Bahrain	11995579	clement@gmail.com
CN4646	Christopher Nolan	Roadno-129 blockno-964	London	United Kingdom	12567980	Christynol@gmail.com
CR7290	Cristiano Ronaldo	Roadno-212 Blockno-673	Porto	Portugal	21377201	cr7@gmail.com
DP4290	Dikpaal Prakashbhai Patel	Roadno-777 blockno-990	Sitra	Bahrain	22995579	dikpaal@gmail.com
EB4290	Elwin Babu	Roadno-444 blockno-123	Sitra	Bahrain	27995579	elwin@gmail.com
EM2345	Eoin Morgan	Roadno-635 blockno-977	Manchester	United Kingdom	13677991	Mor@gmail.com
GS4290	Gurpreet Singh	Roadno-333 blockno-223	Zayed Town	Bahrain	24395579	gurpreet@gmail.com
HB4221	Hafsah Mohsin Bhaiji	Roadno-123 blockno-245	Mumbai	India	34210099	hafsah@gmail.com
JB4646	James Bond	Roadno-129 blockno-969	London	United Kingdom	12567981	james@gmail.com
JR2345	Joe Root	Roadno-345 blockno-977	London	United Kingdom	13566991	Root@gmail.com
KS4290	Karthika Suresh	Roadno-273 blockno-163	Zayed Town	Bahrain	20295579	karthika@gmail.com

#Ticket



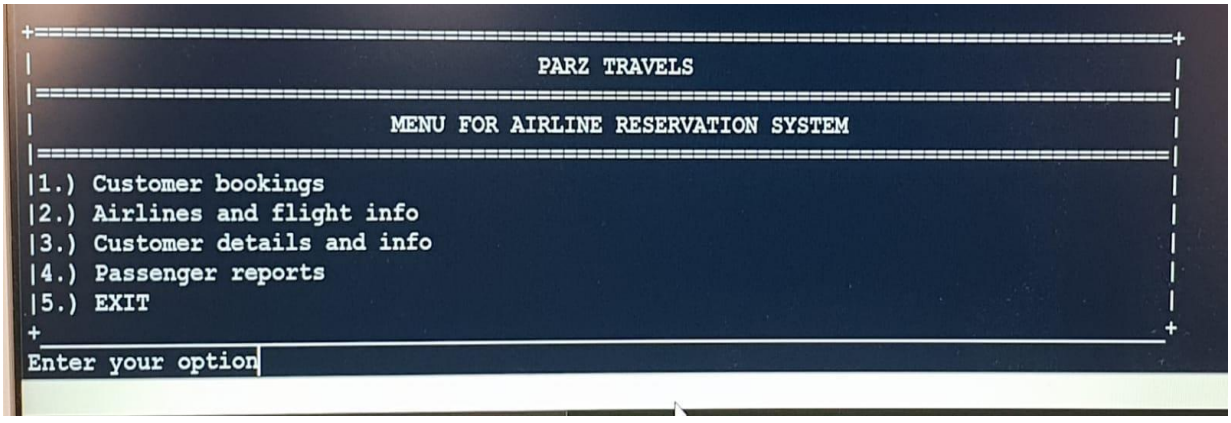
✈✈✈ **** TICKET **** ✈✈✈

P_title	P_Fname	P_Mname	P_Lname	P_Type	ticket number	Flight number	PNR no.	Date	Time	From	To
Mrs.	Sabirah	Masud	Bhaiji	ADT_Eco	1231000000014	GF100	1000000014	2022-03-13	20:00	Bahrain	Mumbai

Gate closes 20 minutes before departure
Please adhere to the airport 1 hour before the departure time.

******* HAVE A PLEASANT JOURNEY *******

#Main Menu



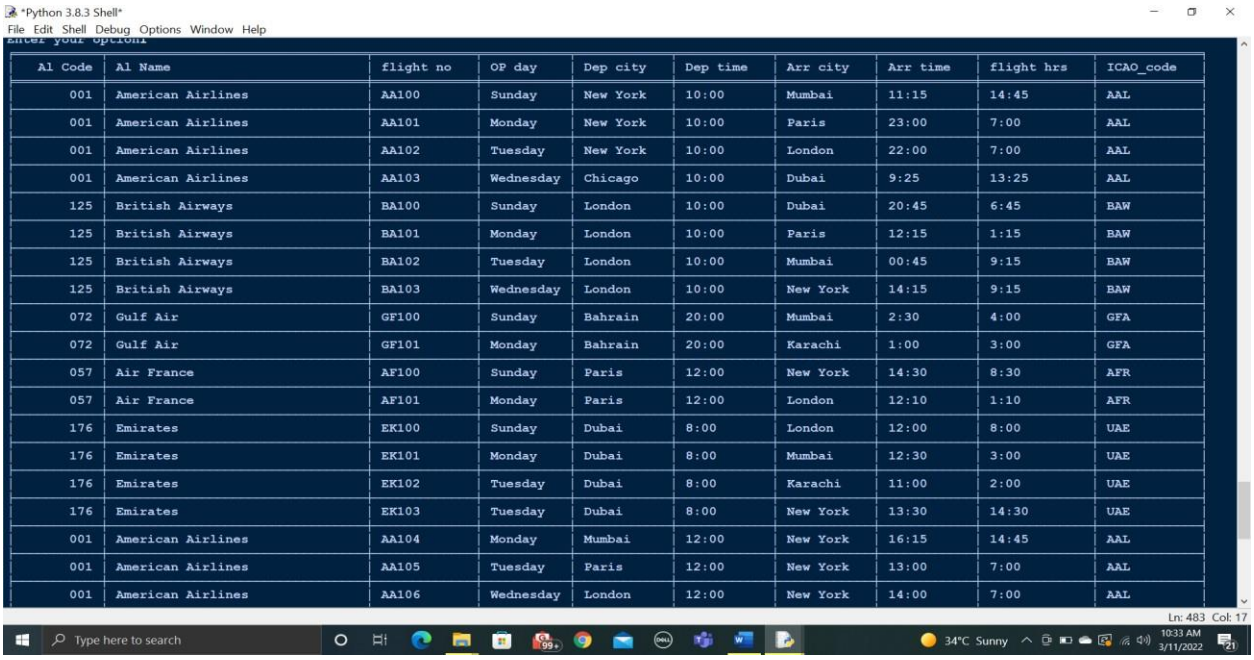
PARZ TRAVELS

MENU FOR AIRLINE RESERVATION SYSTEM

1.) Customer bookings
2.) Airlines and flight info
3.) Customer details and info
4.) Passenger reports
5.) EXIT

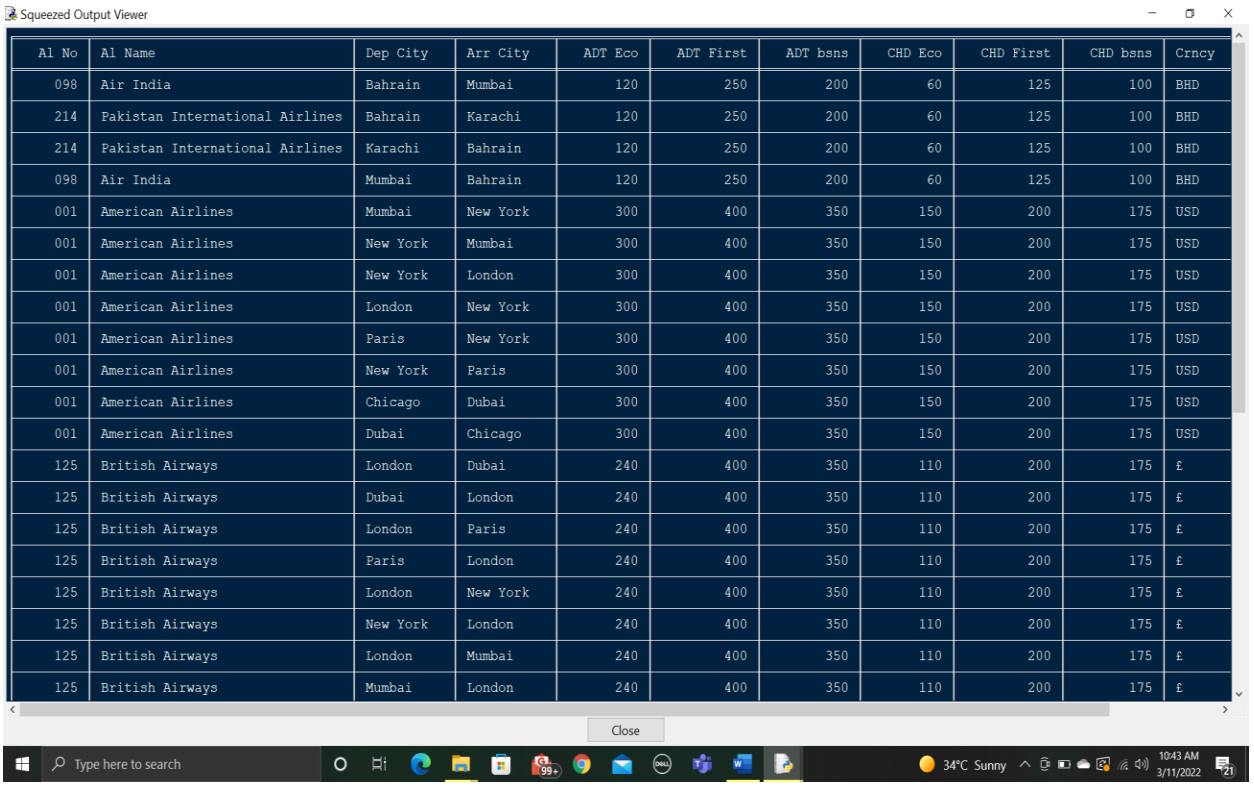
Enter your option

#Airline Information



Al Code	Al Name	flight no	OP day	Dep city	Dep time	Arr city	Arr time	flight hrs	ICAO_code
001	American Airlines	AA100	Sunday	New York	10:00	Mumbai	11:15	14:45	AAL
001	American Airlines	AA101	Monday	New York	10:00	Paris	23:00	7:00	AAL
001	American Airlines	AA102	Tuesday	New York	10:00	London	22:00	7:00	AAL
001	American Airlines	AA103	Wednesday	Chicago	10:00	Dubai	9:25	13:25	AAL
125	British Airways	BA100	Sunday	London	10:00	Dubai	20:45	6:45	BAW
125	British Airways	BA101	Monday	London	10:00	Paris	12:15	1:15	BAW
125	British Airways	BA102	Tuesday	London	10:00	Mumbai	00:45	9:15	BAW
125	British Airways	BA103	Wednesday	London	10:00	New York	14:15	9:15	BAW
072	Gulf Air	GF100	Sunday	Bahrain	20:00	Mumbai	2:30	4:00	GFA
072	Gulf Air	GF101	Monday	Bahrain	20:00	Karachi	1:00	3:00	GFA
057	Air France	AF100	Sunday	Paris	12:00	New York	14:30	8:30	AFR
057	Air France	AF101	Monday	Paris	12:00	London	12:10	1:10	AFR
176	Emirates	EK100	Sunday	Dubai	8:00	London	12:00	8:00	UAE
176	Emirates	EK101	Monday	Dubai	8:00	Mumbai	12:30	3:00	UAE
176	Emirates	EK102	Tuesday	Dubai	8:00	Karachi	11:00	2:00	UAE
176	Emirates	EK103	Tuesday	Dubai	8:00	New York	13:30	14:30	UAE
001	American Airlines	AA104	Monday	Mumbai	12:00	New York	16:15	14:45	AAL
001	American Airlines	AA105	Tuesday	Paris	12:00	New York	13:00	7:00	AAL
001	American Airlines	AA106	Wednesday	London	12:00	New York	14:00	7:00	AAL

#Airline flight pricing



Al No	Al Name	Dep City	Arr City	ADT Eco	ADT First	ADT bsns	CHD Eco	CHD First	CHD bsns	Crncy
098	Air India	Bahrain	Mumbai	120	250	200	60	125	100	BHD
214	Pakistan International Airlines	Bahrain	Karachi	120	250	200	60	125	100	BHD
214	Pakistan International Airlines	Karachi	Bahrain	120	250	200	60	125	100	BHD
098	Air India	Mumbai	Bahrain	120	250	200	60	125	100	BHD
001	American Airlines	Mumbai	New York	300	400	350	150	200	175	USD
001	American Airlines	New York	Mumbai	300	400	350	150	200	175	USD
001	American Airlines	New York	London	300	400	350	150	200	175	USD
001	American Airlines	London	New York	300	400	350	150	200	175	USD
001	American Airlines	Paris	New York	300	400	350	150	200	175	USD
001	American Airlines	New York	Paris	300	400	350	150	200	175	USD
001	American Airlines	Chicago	Dubai	300	400	350	150	200	175	USD
001	American Airlines	Dubai	Chicago	300	400	350	150	200	175	USD
125	British Airways	London	Dubai	240	400	350	110	200	175	£
125	British Airways	Dubai	London	240	400	350	110	200	175	£
125	British Airways	London	Paris	240	400	350	110	200	175	£
125	British Airways	Paris	London	240	400	350	110	200	175	£
125	British Airways	London	New York	240	400	350	110	200	175	£
125	British Airways	New York	London	240	400	350	110	200	175	£
125	British Airways	London	Mumbai	240	400	350	110	200	175	£
125	British Airways	Mumbai	London	240	400	350	110	200	175	£

Conclusion

This project will be very useful for our future IT career, and I would like to thank our school and Mr. Pius Mathew sir for giving us an opportunity to develop a software like this. This project was easily one of the best things we have ever build in our life till now. It will boost one's confidence if he/she is interested in software engineering or software development. The designing of the project will open a creative space also in coding and programming and will thus help those students who are interested in app development or web designing. Logic building was also improved of the students while developing this beautiful project. Python indeed is a very good start for beginners like us since its syntax is easy to grasp and will thus give a lot of confidence to the coding/programming beginners. Finally, I would like to thank all my teammates and my fellow classmates because it was them only who gave me the motivation to improve my software step by step and day by day. I will keep making changes in this software after the final exam and will try to take this project to a commercial level.

THANK YOU!!