

# AIRPORT MANAGEMENT SYSTEM

KONICAA SHARMA

XII-D

2020-21

Board Roll No. :26631915

# INDEX

<b>S.No.</b>	<b>TOPIC</b>	<b>PAGE No.</b>	<b>DATE</b>
1.	Acknowledgement	1	1/11/2020
2.	Introduction	2	1/11/2020
3.	Application	3	1/11/2020
4.	Certificate	4	2/11/2020
5.	Hardware and software requirements	5	2/11/2020
6.	Python as Front End	6	2/11/2020
7.	MySQL as back end	7	3/11/2020
8.	E.R. Diagram	9	3/11/2020
9.	Flowchart	10	6/11/2020
10	Python Modules and Built-in functions	11	7/11/2020
11.	UDF's and Glossary	12	7/11/2020
12.	Screenshots of MySQL Queries	14	7/11/2020
13.	Source Code	17	8/11/2020
14.	Output Screenshots	44	8/11/2020
15.	Bibliography	63	12/11/2020

# ACKNOWLEDGEMENT

Any accomplishment requires the efforts of many people and this work is no different. I am happy to take this opportunity to express my heartfelt gratitude towards all those who made the successful completion of this project possible.

I would like to thank **Ms. Bhawna Sachdev** (C.S. Teacher, Holy Child School), for her purposeful guidance and constant encouragement which made this seemingly difficult task a wonderful learning experience.

I am grateful to **Mr. Shoji Chacko** (Hardware Expert, Holy Child School), who helped me in getting the required information and contributed to the successful completion of the project.

Last but not the least, the project has been a great learning experience and I am indebted to **Holy Child Sr. Sec. School** for providing me with an opportunity to do the same.

# INTRODUCTION

It is a term project entitled 'AIRPORT MANAGEMENT SYSTEM'. This project is useful for the airport department as well as the passengers to keep a track of account details. The main objective of the project is to manage the details of flights, passengers, reservation of flights, date, fare. The project is totally built at the administrative end and thus only the administrator is guaranteed the access.

This project also provides convenience to the client by easily booking and cancelling the seats as per the requirement. The purpose of the project is to build an application program to reduce the manual work for management of data. In this, the concept of connectivity of python with RDBMS (here it is MySQL) is used.

# APPLICATION

This program is almost 85-90% commercially viable. But some modifications and addition of extra features will make it upto 100% applicable. The 'AIRPORT MANAGEMENT SYSTEM' project is an attempt to stimulate the basic concepts of the airport managing department. The system enables the passengers to do things such as search for airline flights for two travel cities on a specified date, choose a flight based on the details, reservation of flight as well as cancellation of reservation. This project could be used commercially for booking flights, be it domestic or international.

The project has been planned to be having the view of distributed architecture, with centralised storage of database. It takes care of different modules and their associated reports which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

# CERTIFICATE

This is a bona fide project done satisfactorily at **Holy Child Senior Secondary** School by **Konicaa Sharma, Arpita Oberoi and Shriya Pandey** in fulfilment of the same.

This report or a similar project on the topic '**AIRPORT MANAGEMENT SYSTEM**' has not been submitted for any other examination and does not include any part of any other course and undergone by the candidate.

(Teacher's Signature)

---

Date : \_\_/\_\_/\_\_

# HARDWARE AND SOFTWARE REQUIREMENTS

## HARDWARE REQUIREMENTS:

While developing the project the used hardware are:

- Processor Intel(R) Core(TM)  
i5-4310U CPU @2Ghz
- Installed Memory (RAM) 4.00GB
- System Type 64-bit operating system,  
x64-based processor
- Keyboard and mouse

## SOFTWARE REQUIREMENTS:

- Windows OS
- Python IDLE as frontend Development Environment
- MySQL as backend server
- Microsoft Word for documentation

# PYTHON AS FRONT END

Usually, the data inputted by the user along with the generated output is displayed but not stored because all program execution takes place inside the ram which is a temporary memory. Thus, when application executed for the second time, it requires new set of inputs from the user, this limitation can be overcome by sending the output generated and saving the input fetched from the user in database created at the backend of the application. This is termed as Front End Interface of the application. Here python is used as a front end where all the data for creating database is taken.

Several reasons to use python for programming database applications are as follows:

- Programming in python is arguably more efficient and faster as compared to other languages
- Python is famous for its portability
- It is platform-independent
- Supports SQL cursors
- Supports Relational Database Systems
- Python database APIs are compatible with various databases, so it is very easy to migrate and port database application interfaces
- In many programming languages, the application developer needs to take care of the open and closed connections of the database to avoid further exceptions and errors. In Python, these connections are taken care of.



# **MYSQL AS BACK END**

While working with an application, it is required to save data permanently on some secondary storage device, which is usually the hard disk, so that stored data could be used for future reference, modification, deletion, retrieval. An application usually stores a lot of data in the form of database which is directly accessible to the user. This database is used by the application to give suitable response to the user. This database is called Back End Database. Here MySQL is used as back end where all the data taken from front end is stored.

MySQL is a free to use, open source database that facilitates effective management of database by connecting them to a software. It is a stable, reliable and powerful solution with advanced features including:

## **1. DATA SECURITY**

MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

## **2. POWERFUL PROCESSING CAPABILITIES**

MySQL is a powerful, easy, compatible and fast Relational Database Management System. It can handle complicated corporate applications and processing requirements.

## **3. HIGH PERFORMANCE**

MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

## **4. CLIENT-SERVER ARCHITECTURE**

MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

## **5. RELIABLE**

SQL provides a high level of well defined set of commands that provides the desirable results without any ambiguity.

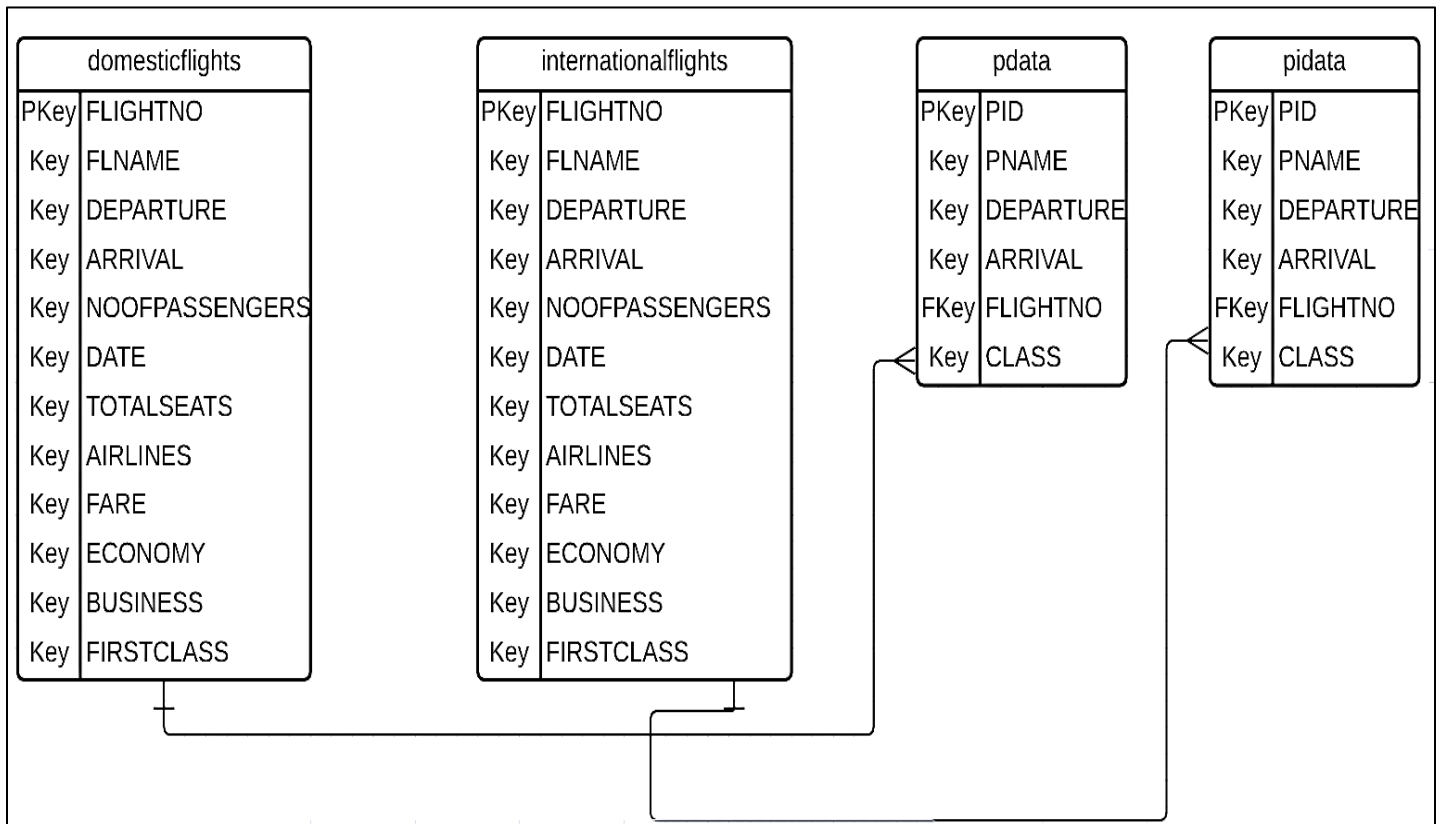
## **6. AUTHORIZATION**

MySQL provides DDL commands to check for user authentication and authorization by restricting access to relations and views

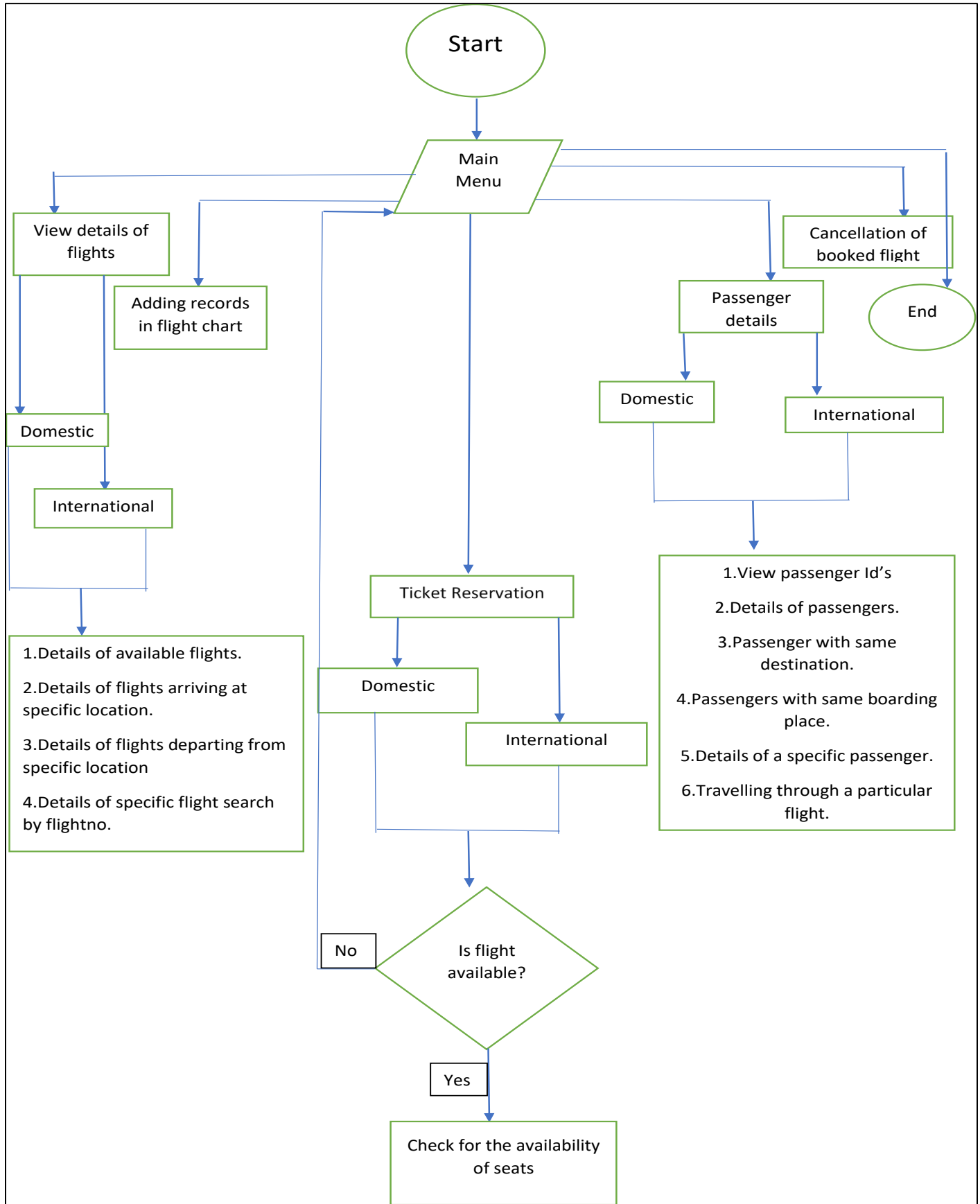
## **7. MULTI-PLATFORM SUPPORT**

MySQL supports several different platforms like UNIX, LINUX, Mac OS X and Microsoft Windows.

# E.R. DIAGRAM



# FLOWCHART



# PYTHON MODULES AND BUILT-IN FUNCTIONS

## **MODULES:**

MySQL.connector module:

Used to connect MySQL with IDLE.

## **BUILT-IN FUNCTIONS:**

1.connect():The connect() statement establishes a connection to the MySQL server and returns a MySQL connection object

2.execute():Used to execute SQL queries from Python

3.commit():It sends a COMMIT statement to the MySQL server,committing the current transaction

4.fetchone():The fetchone() statement is used to fetch the next row of query result set.

5.fetchall():It fetches all the rows in a result set.

# UDF's AND GLOSSARY

## UDF's

### **1.flight\_details()-**

For viewing the details of flights such as

- To View Airlines
- To View the details of all available Flights (Domestic as well as International)
- To View the details of Flights arriving at a specific location
- To View the details of Flights departing from a specific location
- To View the details of a specific flight

### **2.add()-**

- To add new flights (Domestic as well as International)  
in the head office records

### **3.booking ()-**

- For booking of flights
  1. **pdata()**-reservations for domestic flights
  2. **pidata()**-reservations for international flights

### **4.p\_details()-**

For viewing the details of passengers such as

- To View Passenger id's of all passengers
- To View the details of all passengers travelling through Domestic as well as International Flights
- To View the details of passengers arriving at a specific location
- To View the details of passengers boarding from a specific location
- To View the details of a specific passenger
- To View the details of passengers travelling through a particular flight

### **5.cancel()-**

**Used for cancellation of the reservations**

1. **cancel\_Dticket()**-for domestic airlines reservation cancellation
2. **cancel\_Iticket()**-for international airlines reservation cancellation

## **GLOSSARY**

**1.flight\_details**-details of flight

used as the function header for viewing the details of domestic as well as international flights

**2.ch**-choice

to accept user's choice

**3.des**-destination

used to accept boarding as well as arrival locations from the user

**4.f**-passenger id

used to accept the passenger id of the passenger whose details are to be viewed

**5.ftype**-choice to be entered by the user in order to enter the flight records

**6.flightno**-flight number

to accept flight number from the user

**7.flname**-flight name

to accept flight name from the user in case of domestic as well as international flights

**8.noofpassengers**-number of passengers

used to find out the number of passengers travelling through a particular flight

**9.pname**-passenger's name

used to accept passenger's name from the user

**10.pid**-passenger's id

used to uniquely identify a passenger

**11.s**-fare of flight

to display the flight fare after flight reservation is completed

**12.CLAS**-class of flight (economy, business, first)

Used to accept user's choice according to the type of class chosen

**13.cancel\_Dticket**-for cancellation of domestic flights

**14.cancel\_Iticket**- for cancellation of international flights

# SCREENSHOTS OF MySQL RELATED QUERIES

```
mysql> use airlines;
Database changed
mysql> Create table Domesticflights(FLIGHTNO int(4) Primary Key,
->                                FLNAME varchar(15) ,
->                                DEPARTURE varchar(15),
->                                ARRIVAL varchar(15),
->                                NOOFPASSENGERS int(3),
->                                date date,
->                                TOTALSEATS int(11),
->                                airlines varchar(20),
->                                fare int(11),
->                                economy int(11),
->                                business int(11),
->                                firstclass int(11));
Query OK, 0 rows affected (0.37 sec)
```

```
mysql> Create table internationalflights(FLIGHTNO char(4) Primary Key,
->                                FLNAME varchar(20) ,
->                                DEPARTURE varchar(30),
->                                ARRIVAL varchar(30),
->                                NOOFPASSENGERS int(3),
->                                date date,
->                                TOTALSEATS int(11),
->                                airlines varchar(30),
->                                fare int(11),
->                                economy int(11),
->                                business int(11),
->                                firstclass int(11));
Query OK, 0 rows affected (0.19 sec)
```



```
mysql> Create table pdata(pid int(11) Primary Key,
->                             pname varchar(30) ,
->                             DEPARTURE varchar(20),
->                             ARRIVAL varchar(20),
->                             FLIGHTNO int(11),
->                             CLASS varchar(20));
Query OK, 0 rows affected (0.20 sec)
```

```
mysql> Create table pidata(pid int(11) Primary Key,
->                             pname varchar(30) ,
->                             DEPARTURE varchar(30),
->                             ARRIVAL varchar(30),
->                             FLIGHTNO int(11),
->                             CLASS varchar(15));
Query OK, 0 rows affected (0.27 sec)
```

```
mysql> use airlines;
Database changed
mysql> desc domesticflights;
```

Field	Type	Null	Key	Default	Extra
FLIGHTNO	int(4)	NO		NULL	
FLNAME	varchar(15)	YES		NULL	
DEPARTURE	varchar(15)	YES		NULL	
ARRIVAL	varchar(15)	YES		NULL	
NOOFPASSENGERS	int(3)	YES		NULL	
date	date	YES		NULL	
TOTALSEATS	int(11)	YES		NULL	
airlines	varchar(20)	YES		NULL	
fare	int(11)	YES		NULL	
economy	int(11)	YES		NULL	
business	int(11)	YES		NULL	
firstclass	int(11)	YES		NULL	

12 rows in set (0.02 sec)

```
mysql> desc internationalflights;
```

Field	Type	Null	Key	Default	Extra
FLIGHTNO	char(4)	YES		NULL	
FLNAME	varchar(20)	YES		NULL	
DEPARTURE	varchar(30)	YES		NULL	
ARRIVAL	varchar(30)	YES		NULL	
NOOFPASSENGERS	int(11)	YES		NULL	
DATE	date	YES		NULL	
TOTALSEATS	int(11)	YES		NULL	
AIRLINES	varchar(30)	YES		NULL	
FARE	int(11)	YES		NULL	
economy	int(11)	YES		NULL	
business	int(11)	YES		NULL	
firstclass	int(11)	YES		NULL	

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

```
mysql> desc pdata;
```

Field	Type	Null	Key	Default	Extra
pid	int(11)	YES		NULL	
pname	varchar(30)	YES		NULL	
departure	varchar(20)	YES		NULL	
arrival	varchar(20)	YES		NULL	
FLIGHTNO	int(11)	YES		NULL	
CLASS	varchar(20)	YES		NULL	

```
6 rows in set (0.17 sec)
```

```
mysql> desc pidata;
```

Field	Type	Null	Key	Default	Extra
PID	int(11)	YES		NULL	
PNAME	varchar(30)	YES		NULL	
DEPARTURE	varchar(30)	YES		NULL	
ARRIVAL	varchar(30)	YES		NULL	
FLIGHTNO	int(11)	YES		NULL	
CLASS	varchar(15)	YES		NULL	

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

# SOURCE CODE

# #AIRPORT MANAGEMENT SYSTEM

## #FLIGHT DETAILS

```
import mysql.connector
```

```
mydb=mysql.connector.connect(host="localhost",user="root",passwd="admin",database="airlines")
```

```
mycursor=mydb.cursor()
```

```
print("*****WELCOME TO AIRPORT MANAGEMENT SYSTEM*****")
```

```
def flight_details():
```

```
print("***** FLIGHT DETAILS *****")
```

while True:

```
print("1.Domestic Flights")
```

```
print("2.International Flights")
```

```
print("3.Go back to the Main Menu")
```

```
choice=int(input("Enter flight type whose details you want to view(1/2)="))
```

```
print("=====")
```

```
if choice==1:
```

while True:

```
print("1.View Airlines")
```

```
print("2.View the details of all available Domestic Flights")
```

```
print("3.View the details of Flights arriving at a specific location")
```

```
print("4.View the details of Flights departing from a specific location")
```

```
print("5.View the details of a specific flight")
```

```
print("6.Go Back to the main menu")
```

```
ch=int(input("Enter your choice(1-6)="))
```

[illegible]

```
if ch==1:
```

```
mycursor.execute("select distinct(airlines) from domesticflights")
```

```
for x in mycursor:
```

[illegible]

try:

```
sql="SELECT * FROM domesticflights WHERE DEPARTURE like '{ }'".format(des)
mycursor.execute(sql)
data=mycursor.fetchall()
```

```
if data==[]:
```

```
print("Oops!!!No flight with this location is available at the moment")
```

```
print("Contact Head Office for more information.")
```

else:

```
print('FL_NO',' FLIGHT_NAME',' ','BOARDING',' ARRIVAL','\t','S_FILL',
      '\t','DATE','\t\t','T_SEATS','AIRLINES','\t','FARE')
```

```
for i in data:
```

```
print(i[0], ' ', i[1], '\t', i[2], '\t ', i[3], '\t', i[4], '\t', i[5], '\t',
      i[6], '\t', i[7], '\t', i[8])
```

except Exception as e:

```
print(e)
```

[illegible]

```
elif ch==5:
```

```
f=int(input("Enter the flight number whose details you want to view:"))
```

try:

```
sql="SELECT * FROM domesticflights WHERE FLIGHTNO like {}".format(f)
mycursor.execute(sql)
data=mycursor.fetchone()
```

```
if data==None:
```

```
print("The given flight number does not exist")
```

else:

```
print('FL_NO','FLIGHT_NAME',' ','BOARDING',' ARRIVAL','\t','S_FILL',
      '\t','DATE','\t','T_SEATS','AIRLINES','\t','FARE')
print(data[0],' ',data[1],'\t',data[2],'\t ',data[3],'\t',data[4],'\t',
      data[5],'\t',data[6],'\t',data[7],'\t',data[8])
```

[illegible]

[illegible]

```
for i in data:
```

```
print(i[0], ' ', i[1], '\t', i[2], '\t ', i[3], '\t', i[4], '\t', i[5], '\t',  
      i[6], '\t', i[7], '\t', i[8])
```

except Exception as e:

```
print(e)
```

[illegible]

```
elif c==5:
```

```
f=int(input("Enter the flight number whose details you want to view:"))
```

try:

```
sql="SELECT * FROM internationalflights WHERE FLIGHTNO like { }".format(f)
```

```
mycursor.execute(sql)
```

```
data=mycursor.fetchone()
```

```
if data==None:
```

```
print("The given flight number does not exist")
```

else:

```
print('FL_NO','FLIGHT_NAME',' ', 'BOARDING',' ARRIVAL','\t','S_FILL',
      '\t','DATE','\t\t','T_SEATS','AIRLINES','\t','FARE')
```

```
print(data[0],',',data[1],'\t',data[2],'\t ',data[3],'\t',data[4],'\t',
      data[5],'\t',data[6],'\t',data[7],'\t',data[8])
```

except Exception as e:

```
print(e)
```

[illegible]

```
elif c==6:
```

return

```
elif choice==3:
```

return

```
def add():
```

```
print("*****ADDING RECORDS IN THE FLIGHT CHART*****")
```



```
print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

print("Where do you want to enter the records?")

print("1.Enter records in DomesticFlights")

print("2.Enter records in InternationalFlights")

print("3.Go back to the Main Menu")

ftype=int(input("Enter your choice="))

print("=====")

if ftype==1:

    try:

        mydb=mysql.connector.connect(host="localhost",user="root",password="admin",database="airlines")

        mycursor=mydb.cursor()

        while True:

            flightno=int(input("Enter flight number:"))

            flname=input("Enter flight name:")

            departure=input("Enter departure city:")

            arrival=input("Enter arrival city:")

            noofpassengers=0

            date=input('Enter the date of departure:')

            totalseats=300

            airlines=input("Enter name of the airlines:")

            fare=int(input("Enter flight fare:"))

            economy=0;business=0;firstclass=0

            sql="insert into domesticflights(flightno,flname,departure,arrival,\

noofpassengers,date,totalseats,airlines,fare,economy,business,firstclass)values({},{'','{}','{}',{},{},'{}','{}',{},{},{{}},{})".format(flightno,flname,departure,arrival,noofpassengers,date,totalseats,airlines,fare,economy,business,firstclass)

            mycursor.execute(sql)

            print("\n!!!Successfully inserted the record!!!")

            ans=input(" Do you want to enter more(y/n)?")

            if ans in 'Nn':
```

```

        break

mydb.commit()

except Exception as e:

    print(e)

elif ftype==2:

    try:

        mydb=mysql.connector.connect(host="localhost",user="root",password="admin",database="airlines")

        mycursor=mydb.cursor()

        while True:

            flightno=int(input("Enter flight number:"))

            flname=input("Enter flight name:")

            departure=input("Enter boarding country:")

            arrival=input("Enter destination:")

            noofpassengers=0

            date=input('Enter date of departure:')

            totalseats=300

            airlines=input("Enter name of the airlines:")

            fare=int(input("Enter flight fare:"))

            economy=0;business=0;firstclass=0

            sql="insert into internationalflights(flightno,flname,departure,\

            arrival,noofpassengers,date,totalseats,airlines,fare,economy,business,\

            firstclass)values({}, '{}', '{}', {}, {}, '{}', '{}', {}, {}, \

            {})".format(flightno,flname,departure,arrival,noofpassengers,date,\

                        totalseats,airlines,fare,economy,business,firstclass)

            mycursor.execute(sql)

            print("!!!Successfully inserted the record!!!")

            ans=input("You want to enter more(y/n)?")

            if ans in 'nN':

                break

        mydb.commit()

```

```

except Exception as e:
    print(e)
elif ftype==3:
    print("TERMINATING")
    return
else:
    print("WRONG CHOICE")

def booking():
    def pdata():

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

    try:
        mydb=mysql.connector.connect(host='localhost',user='root',passwd='admin',database='airlines')
        mycursor=mydb.cursor()
        pname=input("Enter your name:")
        arrival=input("Enter your Destination:")
        sql="SELECT * FROM domesticflights WHERE arrival like '{}'.format(arrival)
        mycursor.execute(sql)
        data=mycursor.fetchall()
        mydb.commit()
        if data==[]:
            print("SORRY!!!..NO FLIGHT AVAILABLE FOR",arrival,". CONTACT HEAD OFFICE")
        else:

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

        print("The available flights are:")

        sql="SELECT FLIGHTNO,FLNAME,DEPARTURE FROM DOMESTICFLIGHTS WHERE
ARRIVAL='{}'.format(arrival)

```

```

print()
print("FLIGHT_NO,FLIGHT_NAME,DEPARTURE")
mycursor.execute(sql)
for i in mycursor:
    print(i)
a=int(input("Do you want to confirm the booking ? Enter 1 for yes:"))
if a==1:
    print()
    flightno=int(input("Enter flight number="))
    print()
    data="SELECT * FROM DOMESTICFLIGHTS WHERE flightno={ }".format(flightno)
    mycursor.execute(data)
    b=mycursor.fetchall()
    for i in b:
        if i[4]>=i[6]:
            print("SORRY!!! SEATS FOR THIS FLIGHT ARE ALREADY BOOKED.LOOK FOR
SOME ANOTHER FLIGHT.")
            break
        else:
            pid=int(input("Enter passenger id:"))
            departure=input("Enter boarding place=")
            print ("please choose a class type\n")
            s=0
            print ("FOLLOWING ROOMS ARE AVAILABLE:-")
            print ("1. type First class---->rs 8000 PN\("-")
            print ("2. type Business class---->rs 6000 PN\("-")
            print ("3. type Economy class---->rs 4000 PN\("-")
            flag=0
            while flag!=1:
                x=int(input("Enter Your Choice:"))
                if(x==1):

```

```

        if i[11]<60:
            print ("you have opted First class")
            s=8000
            sql="update domesticflights set firstclass=firstclass+1 where
flightno={ }".format(flightno)
            mycursor.execute(sql)
            flag=1
        else:
            print("SEATS OF FIRST CLASS ARE ALREADY FILLED!!!LOOK FOR SOME
ANOTHER SEAT")
            flag=0
    elif(x==2):
        if i[10]<120:
            print ("you have opted Business class")
            s=6000
            sql="update domesticflights set business=business+1 where
flightno={ }".format(flightno)
            mycursor.execute(sql)
            flag=1
        else:
            print("SEATS OF BUSINESS CLASS ARE ALREADY FILLED!!!LOOK FOR
SOME ANOTHER SEAT")
            flag=0

    elif(x==3):
        s=0
        if i[9]<120:
            print ("you have opted Economy class")
            s=4000
            sql="update domesticflights set economy=economy+1 where
flightno={ }".format(flightno)
            mycursor.execute(sql)

```

```

        flag=1
    else:
        print("SEATS OF ECONOMY CLASS ARE ALREADY FILLED!!!.LOOK FOR
SOME ANOTHER SEAT")
        flag=0
        print ("AMOUNT PAYABLE =",s,"\n")
        CLAS=input("Enter class=")
        sql="INSERT INTO
pdata(pid,pname,departure,arrival,flightno,class)values({},{},{},{},{},{})".format(pid,pname,departure,arri
val,flightno,CLAS)
        mycursor.execute(sql)
        mydb.commit()
        print("Your PassengerId is:",pid)
        print("*****TICKET RESERVATION SUCCESSFULLY
COMPLETED*****")
        sql1="UPDATE DOMESTICFLIGHTS SET NOOFPASSENGERS=NOOFPASSENGERS+1
WHERE FLIGHTNO={ }".format(flightno)
        mycursor.execute(sql1)
        mydb.commit()
    except Exception as e:
        print(e)

def pidata():

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

    try:
        mydb=mysql.connector.connect(host='localhost',user='root',passwd='admin',database='airlines')
        mycursor=mydb.cursor()
        pname=input("Enter your name:")
        arrival=input("Enter your Destination:")
        sql="SELECT * FROM internationalflights WHERE arrival like '{ }'".format(arrival)
        mycursor.execute(sql)

```

```

data=mycursor.fetchall()

mydb.commit()

if data==[]:

    print("SORRY!!!..NO FLIGHT AVAILABLE FOR",arrival,". CONTACT HEAD OFFICE")

else:

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

    print("The available flights are:")

    sql="SELECT FLIGHTNO,FLNAME,DEPARTURE FROM INTERNATIONALFLIGHTS WHERE
ARRIVAL='{ }'".format(arrival)

    print()

    print("FLIGHT_NO,FLIGHT_NAME,DEPARTURE")

    mycursor.execute(sql)

    for i in mycursor:

        print(i)

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

a=int(input("Do you want to confirm the booking.Enter 1 for yes:"))

if a==1:

    print()

    pid=int(input("Enter passenger id:"))

    flightno=int(input("Enter flight number="))

    departure=input("Enter boarding country=")

    print()

    data="SELECT * FROM INTERNATIONALFLIGHTS WHERE flightno={ }".format(flightno)

    mycursor.execute(data)

    b=mycursor.fetchall()

    for i in b:

        if i[4]>=i[6]:

            print("SORRY!!! SEATS FOR THIS FLIGHT ARE ALREADY BOOKED.LOOK FOR
SOME ANOTHER FLIGHT.")

```

```

        break
    else:
        print ("please choose a class type\n")
        s=0
        print ("FOLLOWING ROOMS ARE AVAILABLE:-")
        print ("1. type First class---->rs 80000 PN\-" )
        print ("2. type Business class---->rs 60000 PN\-" )
        print ("3. type Economy class---->rs 40000 PN\-" )

        flag=0
        while flag!=1:
            x=int(input("Enter Your Choice:"))
            if(x==1):
                if i[11]<60:
                    print ("you have opted First class")
                    s=80000
                    sql="update internationalflights set firstclass=firstclass+1 where
flightno={ }".format(flightno)
                    mycursor.execute(sql)
                    flag=1
                else:
                    print("SEATS OF FIRST CLASS ARE ALREADY FILLED!!!.LOOK FOR SOME
ANOTHER SEAT")
                    flag=0
            elif(x==2):
                if i[10]<120:
                    print ("you have opted Business class")
                    s=60000
                    sql="update internationalflights set business=business+1 where
flightno={ }".format(flightno)
                    mycursor.execute(sql)
                    flag=1

```



```

        else:

            print("SEATS OF BUSINESS CLASS ARE ALREADY FILLED!!!LOOK FOR
SOME ANOTHER SEAT")

            flag=0

        elif(x==3):

            s=0

            if i[9]<120:

                print ("you have opted Economy class")

                s=40000

                sql="update internationalflights set economy=economy+1 where
flightno={ }".format(flightno)

                mycursor.execute(sql)

                flag=1

            else:

                print("SEATS OF ECONOMY CLASS ARE ALREADY FILLED!!!LOOK FOR
SOME ANOTHER SEAT")

                flag=0

                print ("AMOUNT PAYABLE =",s,"\n")

                CLAS=input("Enter class=")

                sql="INSERT INTO
pIdata(pid,pname,departure,arrival,flightno,class)values({},{},{},{},{},{})".format(pid,pname,departure,arr
ival,flightno,CLAS)

                mycursor.execute(sql)

                mydb.commit()

                print("Your PassengerId is:",pid)

                print("*****TICKET RESERVATION SUCCESSFULLY
COMPLETED*****")

                sql1="UPDATE INTERNATIONALFLIGHTS SET
NOOFPASSENGERS=NOOFPASSENGERS+1 WHERE FLIGHTNO={ }".format(flightno)

                mycursor.execute(sql1)

                mydb.commit()

    except Exception as e:

```

```
print(e)
```

```
print("***** WELCOME TO AIRLINES RESERVATION SYSTEM  
*****")
```

```
print("\n1.RESERVATION FOR DOMESTIC AIRLINES")
```

```
print("2.RESERVATION FOR INTERNATIONAL AIRLINES")
```

```
print("3.EXIT")
```

```
ch=int(input("Enter your choice(1-3):"))
```

```
if ch==1:
```

```
    pdata()
```

```
elif ch==2:
```

```
    pidata()
```

```
elif ch==3:
```

```
    print("TERMINATING")
```

```
else:
```

```
    print("WRONG CHOICE")
```

```
mydb.commit()
```

```
def p_details():
```

```
    print("***** PASSENGER DETAILS *****")
```

```
    while True:
```

```
        print("1.View records of passengers travelling through Domestic Flights")
```

```
        print("2.View records of passengers travelling through International Flights")
```

```
        print("3.Go back to the Main Menu")
```

```
        choice=int(input("Enter your choice(1/3)="))
```

```
        print("=====")
```

```
        if choice==1:
```

```
            while True:
```

# KONICAA SHARMA

```
for i in data:

    print(i[0],i[1],'\t',i[2],'\t','\t',i[3],'\t',i[4],'\t',i[5])

except Exception as e:

    print(e)

print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

elif ch==4:

    des=input("Enter boarding place:")

    try:

        sql="SELECT * FROM pdata WHERE DEPARTURE like '{ }'.format(des)

        mycursor.execute(sql)

        data=mycursor.fetchall()

        mydb.commit()

        if data==[]:

            print("No passenger with this destination is mentioned in the office records")

        else:

            print("P_ID","NAME","\t","DEPARTURE","\t","ARRIVAL","\t","F_NO","\t","CLASS")

            for i in data:

                print(i[0],i[1],'\t',i[2],'\t','\t',i[3],'\t',i[4],'\t',i[5])

    except Exception as e:

        print(e)

        print("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

elif ch==5:

    f=int(input("Enter the passenger id of the passenger whose details you want to view:"))

    try:

        sql="SELECT * FROM pdata WHERE pid like { }".format(f)

        mycursor.execute(sql)

        data=mycursor.fetchone()

        mydb.commit()

        if data==None:

            print("The given passenger id does not exist in the office records")

        else:
```





[illegible]

```
elif c==5:

    f=int(input("Enter the passenger id of the passenger whose details you want to view:"))

    try:

        sql="SELECT * FROM pidata WHERE pid like {}".format(f)

        mycursor.execute(sql)

        data=mycursor.fetchone()

        mydb.commit()

        if data==None:

            print("The given passenger id does not exist in the office records")

        else:

            print("P_ID","NAME","\t","DEPARTURE","\t","ARRIVAL","\t","F_NO","\t","CLASS")
```





# KONICAA SHARMA

[illegible]

```

mydb=mysql.connector.connect(host='localhost',user='root',passwd='admin',database='airlines')
mycursor=mydb.cursor()
pid=int(input("Enter passenger id whose details is to be deleted="))
sql="delete from pidata where pid=%s"
sid=(pid,)
no=int(input("Enter the flight number for confirmation:"))
mycursor.execute(sql,sid)
mydb.commit()
sql="update internationalflights set noofpassengers=noofpassengers-1\
    where flightno={ }".format(no)
mycursor.execute(sql)
print("1.FIRSTCLASS \t 2.BUSINESS \t 3.ECONOMY")
CLASS=int(input("Choose your registered class(1-3)>>"))
if CLASS==1:
    sql1="update internationalflights set firstclass=firstclass-1 \
        where flightno={ }".format(no)
elif CLASS==2:
    sql1="update internationalflights set business=business-1\
        where flightno={ }".format(no)
elif CLASS==3:
    sql1="update internationalflights set economy=economy-1\
        where flightno={ }".format(no)
mycursor.execute(sql1)
mydb.commit()
print("Reservation for passenger with passenger id",pid,"has been cancelled")
except Exception as e:
    print(e)
mydb.commit()

```

while True:

```

print("***** FLIGHT RESERVATION CANCELLATION
*****")

print("1.DOMESTIC AIRLINES RESERVATION CANCELLATION")
print("\n2.INTERNATIONAL AIRLINES RESERVATION CANCELLATION ")
print("\n3.Go back to the Main Menu")
ch=int(input("Enter from where do you want to delete the records?:"))
print("=====")
if ch==1:
    cancel_Dticket()
elif ch==2:
    cancel_Iticket()
elif ch==3:
    return
else:
    print("WRONG CHOICE")

##### MAIN PROGRAM #####

while True:

print("=====")
)
    print("1.VIEW DETAILS OF AVAILABLE FLIGHTS")
    print("2.ADD RECORDS IN THE FLIGHT CHART")
    print("3.TICKET RESERVATION")
    print("4.VIEW PASSENGER DETAILS")
    print("5.CANCELLATION OF A BOOKED FLIGHT")
    print("6.EXIT\n")

print("=====")
)
    ch=int(input("Enter your choice(1-6)="))

```

```
print("=====
")
if ch==1:
    flight_details()
elif ch==2:
    add()
elif ch==3:
    booking()
elif ch==4:
    p_details()
elif ch==5:
    cancel()
elif ch==6:
    print("***** THANK YOU *****")
    break
else:
    print("WRONG CHOICE")
```

# OUTPUT SCREENSHOTS

```
>>>
===== RESTART: C:\Users\Hp\Desktop\MODIFIED(2) CS PROJECT.py =====
*****WELCOME TO AIRPORT MANAGEMENT SYSTEM*****
=====
1.VIEW DETAILS OF AVAILABLE FLIGHTS
2.ADD RECORDS IN THE FLIGHT CHART
3.TICKET RESERVATION
4.VIEW PASSENGER DETAILS
5.CANCELLATION OF A BOOKED FLIGHT
6.EXIT

=====
Enter your choice(1-6)=1
=====
***** FLIGHT DETAILS *****
1.Domestic Flights
2.International Flights
3.Go back to the Main Menu
Enter flight type whose details you want to view(1/2)=1
=====
1.View Airlines
2.View the details of all available Domestic Flights
3.View the details of Flights arriving at a specific location
4.View the details of Flights departing from a specific location
5.View the details of a specific flight
6.Go Back to the main menu
Enter your choice(1-6)=1
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
('SPICEJET',)
('VISTARA',)
('LUFTHANSA',)
('SAHARA',)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Enter your choice(1-6)=2

[illegible]

Enter your choice(1-6)=3

[illegible]

Enter your choice(1-6)=3

```

Ooops!!!No flight with this location is available at the moment

```

[illegible]

[illegible][illegible]



```
1.View Airlines
2.View the details of all available Domestic Flights
3.View the details of Flights arriving at a specific location
4.View the details of Flights departing from a specific location
5.View the details of a specific flight
6.Go Back to the main menu
```

Enter your choice(1-6)=5

Enter the flight number whose details you want to view:10001

The given flight number does not exist

```
1.View Airlines
2.View the details of all available Domestic Flights
3.View the details of Flights arriving at a specific location
4.View the details of Flights departing from a specific location
5.View the details of a specific flight
6.Go Back to the main menu
```

Enter your choice(1-6)=6

```
=====
1.VIEW DETAILS OF AVAILABLE FLIGHTS
2.ADD RECORDS IN THE FLIGHT CHART
3.TICKET RESERVATION
4.VIEW PASSENGER DETAILS
5.CANCELLATION OF A BOOKED FLIGHT
6.EXIT
```

Enter your choice(1-6)=1

\*\*\*\*\* FLIGHT DETAILS \*\*\*\*\*

```
1.Domestic Flights
2.International Flights
3.Go back to the Main Menu
```

Enter flight type whose details you want to view(1/2)=2

[illegible][illegible][illegible][illegible][illegible][illegible]





Enter your choice(1-6)=2

Where do you want to enter the records?

Enter your choice=1

```
!!!Successfully inserted the record!!!
```

Do you want to enter more(y/n)?N

Where do you want to enter the records?

Enter your choice=2

```
!!!Successfully inserted the record!!!
```

```
You want to enter more(y/n)?N
```



```

1.VIEW DETAILS OF AVAILABLE FLIGHTS
2.ADD RECORDS IN THE FLIGHT CHART
3.TICKET RESERVATION
4.VIEW PASSENGER DETAILS
5.CANCELLATION OF A BOOKED FLIGHT
6.EXIT

```

```

=====
Enter your choice(1-6)=1
=====

```

```

***** FLIGHT DETAILS *****

```

```

1.Domestic Flights
2.International Flights
3.Go back to the Main Menu

```

```

Enter flight type whose details you want to view(1/2)=1
=====

```

```

1.View Airlines
2.View the details of all available Domestic Flights
3.View the details of Flights arriving at a specific location
4.View the details of Flights departing from a specific location
5.View the details of a specific flight
6.Go Back to the main menu

```

```

Enter your choice(1-6)=5

```

```

Enter the flight number whose details you want to view:1033

```

FL_NO	FLIGHT_NAME	BOARDING	ARRIVAL	S_FILL	DATE	T_SEATS	AIRLINES	FARE
1033	SPICEJET FL078	SURAT	MUMBAI	0	2021-09-01	300	SPICEJET	9000

```

Where do you want to enter the records?

```

```

1.Enter records in DomesticFlights
2.Enter records in InternationalFlights
3.Go back to the Main Menu

```

```

Enter your choice=2
=====

```

```

Enter flight number:8003
Enter flight name:VISTARA_FL001
Enter boarding country:SYRIA
Enter destination:INDIA
Enter date of departure:'2021-10-12'
Enter name of the airlines:VISTARA
Enter flight fare:88000

```

```

!!!Successfully inserted the record!!!

```

```

You want to enter more(y/n)?N

```

```

Where do you want to enter the records?

```

```

1.Enter records in DomesticFlights
2.Enter records in InternationalFlights
3.Go back to the Main Menu

```

```

Enter your choice=3
=====

```

```

TERMINATING

```

- 1.VIEW DETAILS OF AVAILABLE FLIGHTS
- 2.ADD RECORDS IN THE FLIGHT CHART
- 3.TICKET RESERVATION
- 4.VIEW PASSENGER DETAILS
- 5.CANCELLATION OF A BOOKED FLIGHT
- 6.EXIT

Enter your choice(1-6)=1

\*\*\*\*\* FLIGHT DETAILS \*\*\*\*\*

- 1.Domestic Flights
- 2.International Flights
- 3.Go back to the Main Menu

Enter flight type whose details you want to view(1/2)=2

- 1.View Airlines
- 2.View the details of all available International Flights
- 3.View the details of Flights arriving at a specific location
- 4.View the details of Flights departing from a specific location
- 5.View the details of a specific flight
- 6.Go Back to the main menu

Enter your choice(1-6)=5

Enter the flight number whose details you want to view:8003

FL_NO	FLIGHT_NAME	BOARDING	ARRIVAL	S_FILL	DATE	T_SEATS	AIRLINES	FARE
8003	VISTARA_FL001	SYRIA	INDIA	0	2021-10-12	300	VISTARA	88000

- 1.VIEW DETAILS OF AVAILABLE FLIGHTS
- 2.ADD RECORDS IN THE FLIGHT CHART
- 3.TICKET RESERVATION
- 4.VIEW PASSENGER DETAILS
- 5.CANCELLATION OF A BOOKED FLIGHT
- 6.EXIT

Enter your choice(1-6)=3

\*\*\*\*\* WELCOME TO AIRLINES RESERVATION SYSTEM \*\*\*\*\*

- 1.RESERVATION FOR DOMESTIC AIRLINES
- 2.RESERVATION FOR INTERNATIONAL AIRLINES
- 3.EXIT

Enter your choice(1-3):1

Enter your name:DIKSHA  
Enter your Destination:MUMBAI

The available flights are:

FLIGHT_NO	FLIGHT_NAME	DEPARTURE
(1021,	'SPICEJET_FL100',	'DELHI')
(1030,	'LUFTHANSA_FL714',	'DELHI')
(1033,	'SPICEJET_FL078',	'SURAT')

Do you want to confirm the booking ? Enter 1 for yes:1

Enter flight number=1030

SORRY!!! SEATS FOR THIS FLIGHT ARE ALREADY BOOKED.LOOK FOR SOME ANOTHER FLIGHT.

- =====
- 1.VIEW DETAILS OF AVAILABLE FLIGHTS
  - 2.ADD RECORDS IN THE FLIGHT CHART
  - 3.TICKET RESERVATION
  - 4.VIEW PASSENGER DETAILS
  - 5.CANCELLATION OF A BOOKED FLIGHT
  - 6.EXIT

=====

Enter your choice(1-6)=3

=====

\*\*\*\*\* WELCOME TO AIRLINES RESERVATION SYSTEM \*\*\*\*\*

- 1.RESERVATION FOR DOMESTIC AIRLINES
- 2.RESERVATION FOR INTERNATIONAL AIRLINES
- 3.EXIT

Enter your choice(1-3):1

=====

Enter your name:DIKSHA

Enter your Destination:MUMBAI

=====

The available flights are:

FLIGHT\_NO,FLIGHT\_NAME,DEPARTURE  
(1021, 'SPICEJET\_FL100', 'DELHI')  
(1030, 'LUFTHANSA\_FL714', 'DELHI')  
(1033, 'SPICEJET\_FL078', 'SURAT')

Do you want to confirm the booking ? Enter 1 for yes:1

Enter flight number=1033

Enter passenger id:5504

Enter boarding place=SURAT

please choose a class type

FOLLOWING ROOMS ARE AVAILABLE:-

1. type First class---->rs 8000 PN\-
2. type Business class---->rs 6000 PN\-
3. type Economy class---->rs 4000 PN\-

Enter Your Choice:1

you have opted First class

AMOUNT PAYABLE = 8000

Enter class=FIRSTCLASS

Your PassengerId is: 5504

\*\*\*\*\*TICKET RESERVATION SUCCESSFULLY COMPLETED\*\*\*\*\*

- =====
- 1.VIEW DETAILS OF AVAILABLE FLIGHTS
  - 2.ADD RECORDS IN THE FLIGHT CHART
  - 3.TICKET RESERVATION
  - 4.VIEW PASSENGER DETAILS
  - 5.CANCELLATION OF A BOOKED FLIGHT
  - 6.EXIT



```

Enter your choice(1-6)=3
=====
***** WELCOME TO AIRLINES RESERVATION SYSTEM *****

1.RESERVATION FOR DOMESTIC AIRLINES
2.RESERVATION FOR INTERNATIONAL AIRLINES
3.EXIT
Enter your choice(1-3):2
=====
Enter your name:RACHNA
Enter your Destination:ENGLAND
=====
The available flights are:

FLIGHT_NO,FLIGHT_NAME,DEPARTURE
('9001', 'LUFTHANSA_FL092', 'INDIA')
=====
Do you want to confirm the booking.Enter 1 for yes:1

Enter passenger id:6502
Enter flight number=9001
Enter boarding country=INDIA

please choose a class type

FOLLOWING ROOMS ARE AVAILABLE:-
1. type First class---->rs 80000 PN\-
2. type Business class---->rs 60000 PN\-
3. type Economy class---->rs 40000 PN\-
Enter Your Choice:3

```

```

Enter Your Choice:3
SEATS OF ECONOMY CLASS ARE ALREADY FILLED!!!.LOOK FOR SOME ANOTHER SEAT
Enter Your Choice:2
SEATS OF BUSINESS CLASS ARE ALREADY FILLED!!!.LOOK FOR SOME ANOTHER SEAT
Enter Your Choice:1
you have opted First class
AMOUNT PAYABLE = 80000

Enter class=FIRSTCLASS
Your PassengerId is: 6502
*****TICKET RESERVATION SUCCESSFULLY COMPLETED*****
=====

```

Enter your choice (1-6)=4

```
1.View records of passengers travelling through Domestic Flights
2.View records of passengers travelling through International Flights
3.Go back to the Main Menu
Enter your choice(1/3)=1
```

```
Enter your choice(1-7):1
```

(5500, )  
(5501, )  
(5503, )  
(5504, )

```
Enter your choice(1-7):2
```

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
5500	DIVYANSHI	PUNE	CHENNAI	1032	FIRSTCLASS
5501	CHESHTA	DELHI	MUMBAI	1030	ECONOMY
5503	SRINIDHI	DELHI	MUMBAI	1021	FIRSTCLASS
5504	DIKSHA	SURAT	MUMBAI	1033	FIRSTCLASS

```
Enter your choice(1-7):3
```

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
5501	CHESHTA	DELHI	MUMBAI	1030	ECONOMY
5503	SRINIDHI	DELHI	MUMBAI	1021	FIRSTCLASS
5504	DIKSHA	SURAT	MUMBAI	1033	FIRSTCLASS

```

1.View Passenger id's of all passengers
2.View the details of all passengers travelling through Domestic Flights
3.View the details of passengers arriving at a specific location
4.View the details of passengers boarding from a specific location
5.View the details of a specific passenger
6.View the details of passengers travelling through a particular flight
7.Go Back to the main menu

```

Enter your choice(1-7):4

Enter boarding place:delhi

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
5501	CHESHTA	DELHI	MUMBAI	1030	ECONOMY
5503	SRINIDHI	DELHI	MUMBAI	1021	FIRSTCLASS

```

1.View Passenger id's of all passengers
2.View the details of all passengers travelling through Domestic Flights
3.View the details of passengers arriving at a specific location
4.View the details of passengers boarding from a specific location
5.View the details of a specific passenger
6.View the details of passengers travelling through a particular flight
7.Go Back to the main menu

```

Enter your choice(1-7):4

Enter boarding place:lucknow

No passenger with this destination is mentioned in the office records

```

1.View Passenger id's of all passengers
2.View the details of all passengers travelling through Domestic Flights
3.View the details of passengers arriving at a specific location

```

```

4.View the details of passengers boarding from a specific location
5.View the details of a specific passenger
6.View the details of passengers travelling through a particular flight
7.Go Back to the main menu

```

Enter your choice(1-7):5

Enter the passenger id of the passenger whose details you want to view:5504

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
5504	DIKSHA	SURAT	MUMBAI	1033	FIRSTCLASS

```

1.View Passenger id's of all passengers
2.View the details of all passengers travelling through Domestic Flights
3.View the details of passengers arriving at a specific location
4.View the details of passengers boarding from a specific location
5.View the details of a specific passenger
6.View the details of passengers travelling through a particular flight
7.Go Back to the main menu

```

Enter your choice(1-7):6

Enter the flight number:1030

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
5501	CHESHTA	DELHI	MUMBAI	1030	ECONOMY

Enter your choice (1-6)=4

\*\*\*\*\* PASSENGER DETAILS \*\*\*\*\*

- 1.View records of passengers travelling through Domestic Flights
- 2.View records of passengers travelling through International Flights
- 3.Go back to the Main Menu

Enter your choice(1/3)=2

- 1.View Passenger id's of all passengers
- 2.View the details of all passengers travelling through International Flights
- 3.View the details of passengers arriving at a specific location
- 4.View the details of passengers boarding from a specific location
- 5.View the details of a specific passenger
- 6.View the details of passengers travelling through a particular flight
- 7.Go Back to the main menu

```
Enter your choice(1-7):1
```

(6500, )

(6501, )

(6502, )

- 1.View Passenger id's of all passengers
- 2.View the details of all passengers travelling through International Flights
- 3.View the details of passengers arriving at a specific location
- 4.View the details of passengers boarding from a specific location
- 5.View the details of a specific passenger
- 6.View the details of passengers travelling through a particular flight
- 7.Go Back to the main menu

Enter your choice(1-7):2

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
6500	AYUSHMAN	INDIA	AUSTRALIA	8011	ECONOMY
6501	KARMANYA	INDIA	ENGLAND	9001	FIRSTCLASS
6502	RACHNA	INDIA	ENGLAND	9001	FIRSTCLASS

- 1.View Passenger id's of all passengers
- 2.View the details of all passengers travelling through International Flights
- 3.View the details of passengers arriving at a specific location
- 4.View the details of passengers boarding from a specific location
- 5.View the details of a specific passenger
- 6.View the details of passengers travelling through a particular flight
- 7.Go Back to the main menu

Enter your choice(1-7):3

```
Enter destination:ENGLAND
P_ID NAME          DEPARTURE      ARRIVAL        F_NO    CLASS
6501 KARMANYA      INDIA          ENGLAND        9001    FIRSTCLASS
6502 RACHNA       INDIA          ENGLAND        9001    FIRSTCLASS
```



Enter your choice(1-7):4

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
6500	AYUSHMAN	INDIA	AUSTRALIA	8011	ECONOMY
6501	KARMANYA	INDIA	ENGLAND	9001	FIRSTCLASS
6502	RACHNA	INDIA	ENGLAND	9001	FIRSTCLASS

Enter your choice(1-7):4

No passenger with this destination is mentioned in the office records

```
Enter your choice(1-7):5
```

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
6502	RACHNA	INDIA	ENGLAND	9001	FIRSTCLASS

```
Enter your choice(1-7):5
```

59

```
=====
Enter your choice(1-6)=4
=====
```

```
***** PASSENGER DETAILS *****
```

- 1.View records of passengers travelling through Domestic Flights
- 2.View records of passengers travelling through International Flights
- 3.Go back to the Main Menu

```
Enter your choice(1/3)=2
=====
```

- 1.View Passenger id's of all passengers
- 2.View the details of all passengers travelling through International Flights
- 3.View the details of passengers arriving at a specific location
- 4.View the details of passengers boarding from a specific location
- 5.View the details of a specific passenger
- 6.View the details of passengers travelling through a particular flight
- 7.Go Back to the main menu

```
Enter your choice(1-7):6
^
```

```
Enter the flight number:9001
```

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
6501	KARMANYA	INDIA	ENGLAND	9001	FIRSTCLASS

```
^
```

- 1.VIEW DETAILS OF AVAILABLE FLIGHTS
- 2.ADD RECORDS IN THE FLIGHT CHART
- 3.TICKET RESERVATION
- 4.VIEW PASSENGER DETAILS
- 5.CANCELLATION OF A BOOKED FLIGHT
- 6.EXIT

```
=====
Enter your choice(1-6)=5
=====
```

```
***** FLIGHT RESERVATION CANCELLATION *****
```

- 1.DOMESTIC AIRLINES RESERVATION CANCELLATION
- 2.INTERNATIONAL AIRLINES RESERVATION CANCELLATION

- 3.Go back to the Main Menu

```
Enter from where do you want to delete the records?:1
=====
```

```
Do you want to view the passenger names along with their details.Enter 1 for yes.2
```

```
Do you want to view the passenger id's of all registered passengers.Enter 1 for yes.2
^
```

```
Enter passenger id whose details is to be deleted=5504
```

```
Enter the flight number for confirmation:1033
```

- 1.FIRSTCLASS
- 2.BUSINESS
- 3.ECONOMY

```
Choose your registered class(1-3)>>:1
```

```
Reservation for passenger with passenger id 5504 has been cancelled
=====
```

```
=====
Enter your choice(1-6)=5
```

1.DOMESTIC AIRLINES RESERVATION CANCELLATION

2.INTERNATIONAL AIRLINES RESERVATION CANCELLATION

```
Enter from where do you want to delete the records?:2
```

P_ID	NAME	DEPARTURE	ARRIVAL	F_NO	CLASS
6500	AYUSHMAN	INDIA	AUSTRALIA	8011	ECONOMY
6501	KARMANYA	INDIA	ENGLAND	9001	FIRSTCLASS
6502	RACHNA	INDIA	ENGLAND	9001	FIRSTCLASS

```
Enter the flight number for confirmation:9001
```

```
Choose your registered class(1-3)>>1
```

```
Reservation for passenger with passenger id 6502 has been cancelled
```

```
=====
Enter your choice(1-6)=4
```

- 1.View records of passengers travelling through Domestic Flights
- 2.View records of passengers travelling through International Flights
- 3.Go back to the Main Menu

Enter your choice(1/3)=2

```

1.View Passenger id's of all passengers
2.View the details of all passengers travelling through International Flights
3.View the details of passengers arriving at a specific location
4.View the details of passengers boarding from a specific location
5.View the details of a specific passenger
6.View the details of passengers travelling through a particular flight
7.Go Back to the main menu
Enter your choice(1-7):5

```

[illegible]



# BIBLIOGRAPHY

1. Python text book Class12 (By Preeti Arora).
2. Google Chrome.
3. Previous year projects.