INTERNATIONAL INDIAN SCHOOL JEDDAH, SAUDI ARABIA

SUBJECT: INFORMATION PRACTICES

A Project report on:

AIRLINE MANAGEMENT SYSTEM

[2021-22]

NAME: JANA JALALUDDIN

CLASS: 12A3

ROLL.NO:- 38

GR NO.: 34517

ACKNOWLEDGEMENT

We undertook this Project work as the part of XII-Informatics Practices course. We tried to apply the best of knowledge and experience, gained during the study and classwork experience. However, developing software system is generally a quite complex and time-consuming process. It requires systematic study, insight vision and professional approach during the design and development. Moreover, the developer always feels the need, the help and good wishes of the people near you, who have considerable experience and idea.

We would like to extend our sincere thanks and gratitude to our teacher **Bushra Zaidi**. We are very much thankful to our teacher for giving valuable time and moral support to develop this software.

Jana Jalaluddin

12A3

CERTIFICATE

This is to certify that project entitled Airline Management System is a bonafide work done by **Jana Jalaluddin** of class **12A3** session 2021-2022 on partial fulfillment of CBSE’s AISSCE Examination 2022 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidates.

guidelines.

..	**…..**
Signature of Internal examiner	**Signature of Teacher**

CONTENTS

INTRODUCTION

A management information system (MIS) is a computerized database of financial information organized and programmed in such a way that it produces regular reports which can be obtained from the system easily.

This software project is developed to automate the functionalities of Airlines. The purpose of the software project is to develop the Management Information System (MIS) to automate the record of the time and date of arrival and departure of different airlines with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database.

This software is simple in design and working, and can be used by those who have no training.

SYSTEM ANALYSIS

HARDWARE REQUIREMENTS:

→ Processor: Pentium IV or better
→ RAM: 512MB or more
→ Hard Disk: 100 GB
→ Monitor: any

SOTWARE REQUIREMENTS:

→ Operating System: Windows XP, Windows 7, etc
→ Language: Spyder (python 3.7) or better
→ Front end: Spyder(python 3.7) or above
→ Back end: MySQL server 5.0 or above

SOFTWARE USED:

→ Operating System: Microsoft Windows 10
→ Front end: Python (3.8.2)
→ Back end: MySQL(8.0.26)

HARDWARE USED:

→ Monitor: Laptop
→ Processor: intel CORE i7 processor
→RAM: 930 GB

SYNOPSIS

Applications Used:

1. MySQL:

MySQL is an open-source relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The software has a multi-thread server which, support several backs ends, client programs and libraries, wide range application and administrative tools. The software also allows an embedded multi-thread library that can be linked with an application to get a standalone product that is small, fast and easy to manage.

2. Python:

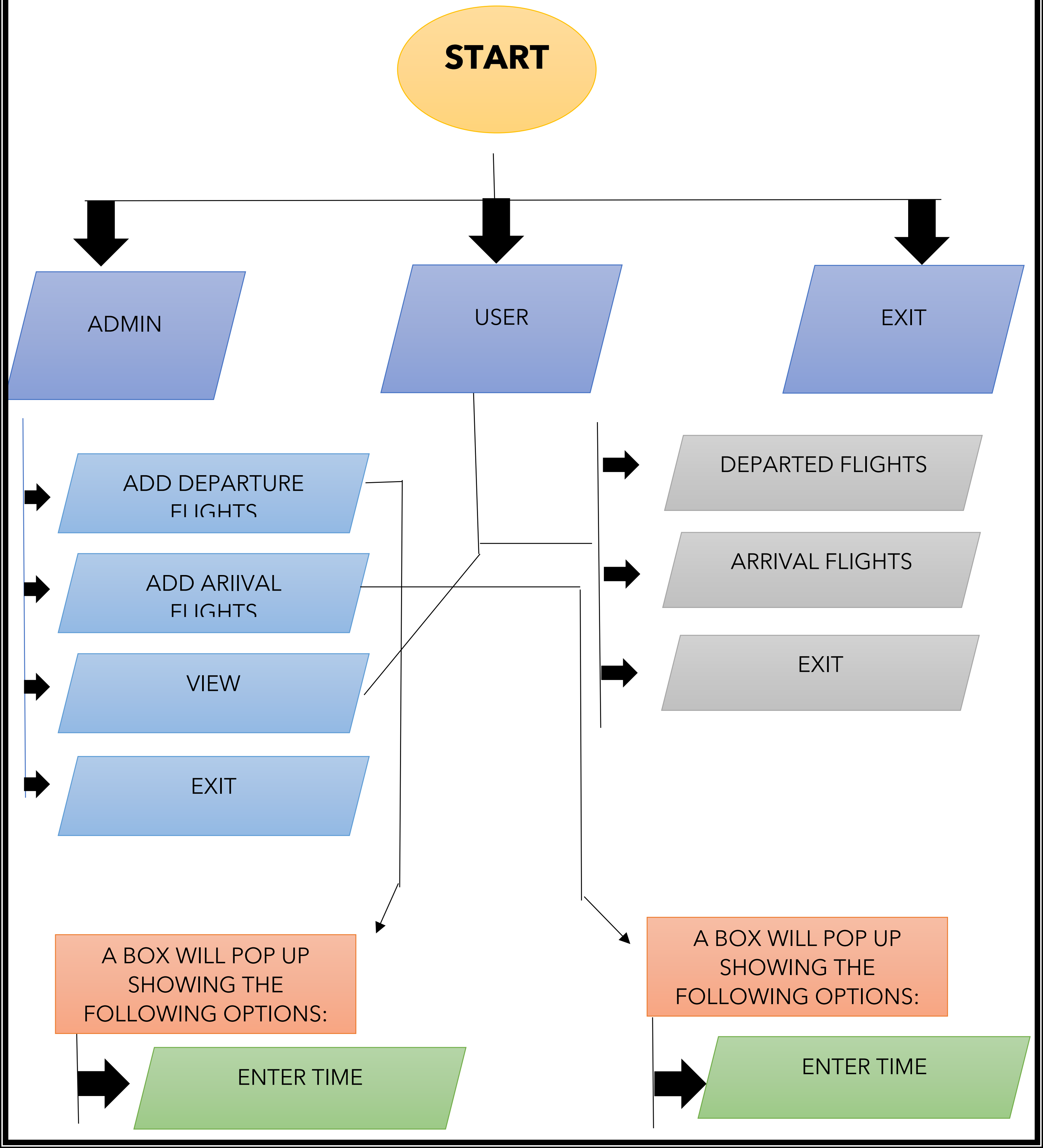
Python is an interpreted high-level general-purpose programming language. During coding and design of the software project, Python Pandas DataFrame, a powerful frontend tool is used for User Interface (GUI) based integrated platform and coding simplicity. It is a front-end that cares about an application's part that users interact with. The main aim of front-end management is to get the strategic perspective right. Using Application program or front-end, we can store, retrieve and manage all information in a proper way.

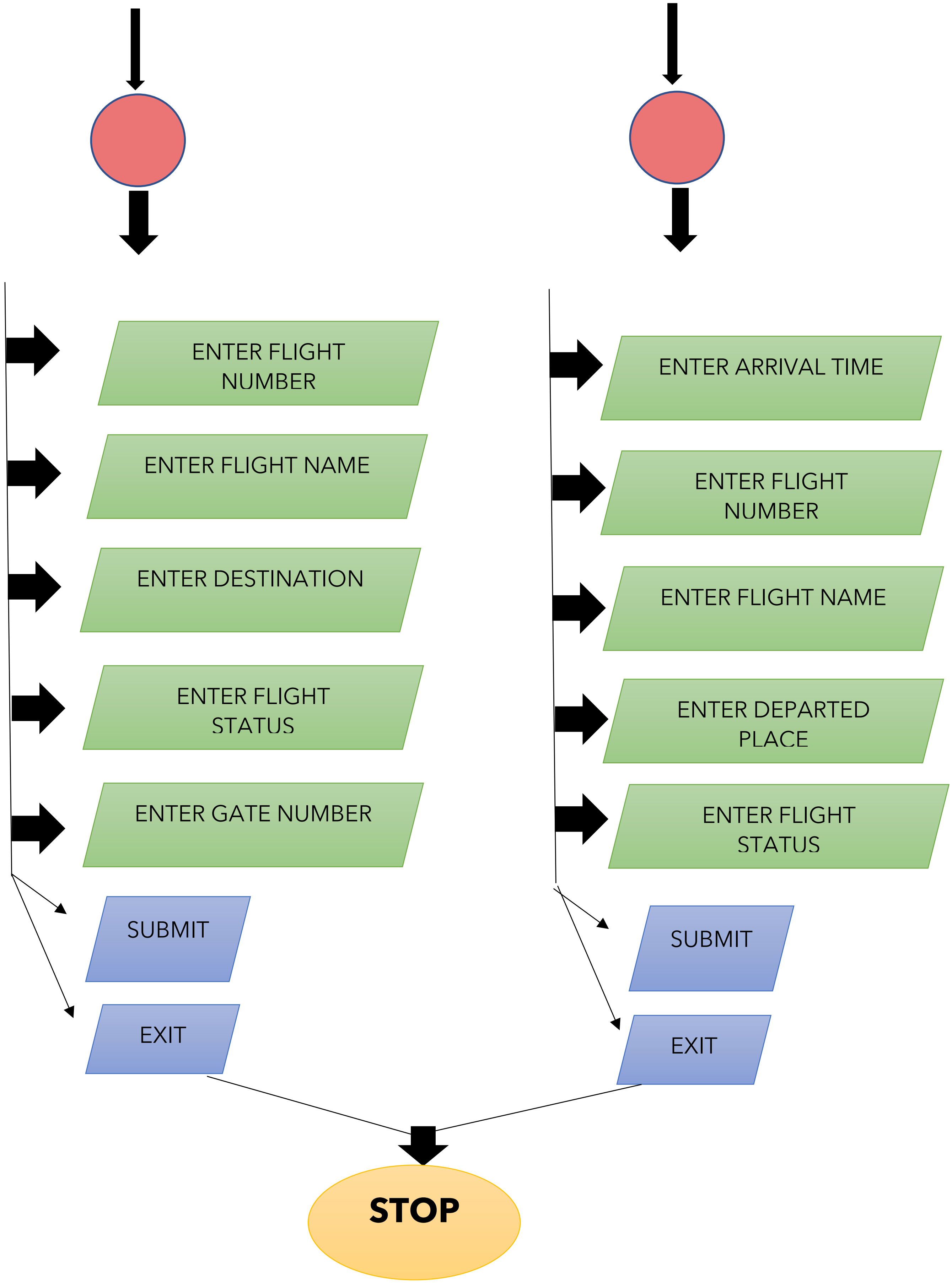
3.Tkinter:

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. It is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

The aim of the software is to enable the user to retrieve and update the information from centralized database designed with MySQL. This software aims to provide graphical and user-friendly interface to interact with a centralized database and maintain all the records of Airline Management System and generate the required reports and information when required.

FLOW CHART


START
ADMIN
USER
EXIT
ADD DEPARTURE FLIGHTS
ADD ARIIVAL FLIGHTS
VIEW
EXIT
DEPARTED FLIGHTS
ARRIVAL FLIGHTS
EXIT
A BOX WILL POP UP SHOWING THE FOLLOWING OPTIONS:
ENTER TIME
A BOX WILL POP UP SHOWING THE FOLLOWING OPTIONS:
ENTER TIME


ENTER FLIGHT NUMBER
ENTER FLIGHT NAME
ENTER DESTINATION
ENTER FLIGHT STATUS
ENTER GATE NUMBER
SUBMIT
EXIT
ENTER ARRIVAL TIME
ENTER FLIGHT NUMBER
ENTER FLIGHT NAME
ENTER DEPARTED PLACE
ENTER FLIGHT STATUS
SUBMIT
EXIT
STOP

SOURCE CODE

```
from tkinter import *
from PIL import ImageTk,Image
import tkinter.messagebox as MessageBox
import mysql.connector as mys
from tkinter import ttk
import tkinter as tk

db = mys.connect(host='localhost', user='root', passwd='jane', database="Airliance")

cur = db.cursor()

cur.execute("drop database Airliance")
cur.execute("Create Database Airliance")
cur.execute("Use Airliance")
qry1 = ('''Create table Departure
        (Scheduled_time Varchar(20),
        Flight_no Varchar(10),
        Flight_name Varchar(20),
        Destination Varchar(20),
        Status Varchar(20),
        Gate_no Varchar(5))''')
cur.execute(qry1)

qry2 = ('''Create table Arival
        (Scheduled_Time Varchar(20),
        Arival_time Varchar(20),
        Flight_no Varchar(10),
        Flight_name Varchar(20),
        Departed_place Varchar(20),
        Status Varchar(30))
        ''' )
```

```
cur.execute(qry2)

root = Tk()
root.geometry("1894x769")
root.title('Air Tower')

def insert():
   global e1
   global e2
   global e3
   global e4
   global e5
   global e6

   Scheduled_time= e1.get();
   Flight_no= e2.get();
   Flight_name= e3.get();
   Destination= e4.get();
   Status= e5.get();
   Gate_no= e6.get();

   if(Scheduled_time=="" or Flight_no=="" or Flight_name=="" or Destination=="" or Status=="" or Gate_no==""):
      MessageBox.showinfo("Insert Status","All Fields are required")
   else:
      db = mys.connect(host='localhost', user='root', passwd='jane', database="Airliance")
      cur = db.cursor()
      sql=("Insert into Departure (Scheduled_time, Flight_no, Flight_name, Destination, Status, Gate_no) VALUES(%s, %s, %s, %s, %s, %s)")
      val=(Scheduled_time, Flight_no, Flight_name, Destination, Status, Gate_no)
      cur.execute(sql,val)
      db.commit()
      MessageBox.showinfo("Insert Status","Inserted Successfully")


def Dadd():
```

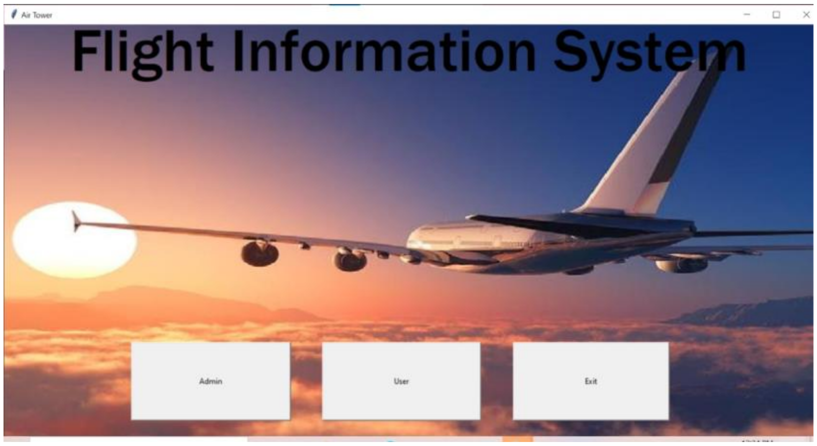
```
Ne=Toplevel(root)

Ne.title("Departure")

Ne.geometry("400x250")


global e1

global e2

global e3

global e4

global e5

global e6


Scheduled_time=Label (Ne, text="Enter Time (Eg:Hr:min)", font=('bold', 10))

Flight_no = Label (Ne, text= "Enter Flight Number", font=('bold', 10))

Flight_name = Label (Ne, text= "Enter Flight Name", font=('bold', 10))

Destination = Label (Ne, text= "Enter Destination", font=('bold', 10))

Status = Label (Ne, text= "Enter Flight Status", font=('bold', 10))

Gate_no = Label (Ne, text= "Enter Gate Number", font=('bold', 10))



Scheduled_time.place(x=18, y=10)

Flight_no.place(x=18,y=40)

Flight_name.place(x=18,y=70)

Destination.place(x=18,y=100)

Status.place(x=18,y=130)

Gate_no.place(x=18,y=160)


e1=Entry(Ne)

e1.place(x=190,y=10)

e2=Entry(Ne)

e2.place(x=190, y=40)

e3=Entry(Ne)

e3.place(x=190,y=70)

e4=Entry(Ne)

e4.place(x=190,y=100)
```

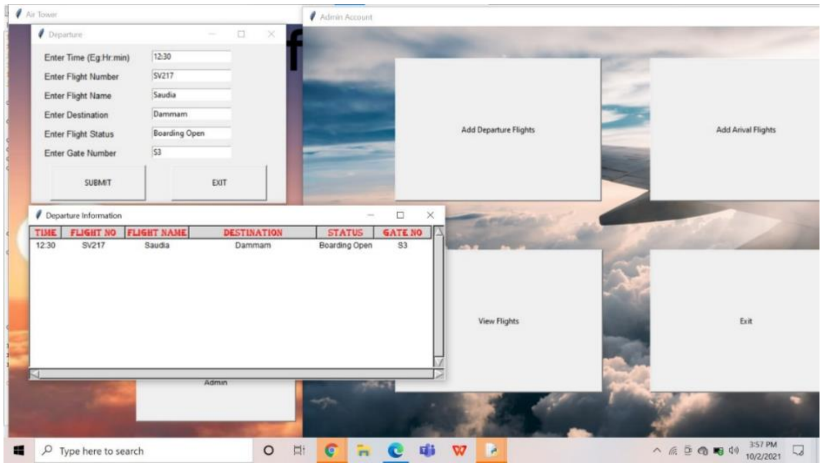
```
    e5=Entry(Ne)
    e5.place(x=190,y=130)
    e6=Entry(Ne)
    e6.place(x=190,y=160)

    Button(Ne, text='SUBMIT',command=insert, height=3, width=20).place(x=30, y=190)
    Button(Ne, text='EXIT',command=Ne.destroy, height=3, width=20).place(x=220,y=190)

    Ne.mainloop()

def Ainsert():
    global a1
    global a2
    global a3
    global a4
    global a5
    global a6

    Scheduled_time= a1.get();
    Arival_time=a2.get();
    Flight_no= a3.get();
    Flight_name= a4.get();
    Departed_place= a5.get();
    Status= a6.get();

    if(Scheduled_time=="" or Arival_time=="" or Flight_no=="" or Flight_name=="" or Departed_place=="" or Status==""):
        MessageBox.showinfo("Insert Status","All Fields are required")
    else:
        db = mys.connect(host='localhost', user='root', passwd='jane', database="Airliance")
        cur = db.cursor()
        sql=("Insert into Arival (Scheduled_time, Arival_time, Flight_no, Flight_name, Departed_place, Status) VALUES(%s, %s, %s, %s, %s, 
%s)")
        val=(Scheduled_time, Arival_time, Flight_no, Flight_name, Departed_place, Status)
        cur.execute(sql,val)
        db.commit()
        MessageBox.showinfo("Insert Status","Inserted Successfully");

def Aadd():
    NO=Toplevel(root)
    NO.title("Arival")
    NO.geometry("400x250")
    global a1
    global a2
    global a3
    global a4
    global a5
    global a6

    Scheduled_time=Label (NO, text="Enter Time (HR:MIN)", font=('bold', 10))
    Arival_time= Label (NO, text="Enter Arival Time (HR:MIN)", font=('bold', 10))
    Flight_no = Label (NO, text= "Enter Flight Number", font=('bold', 10))
    Flight_name = Label (NO, text= "Enter Flight Name", font=('bold', 10))
    Departed_place = Label (NO, text= "Enter Daparted Place", font=('bold', 10))
    Status = Label (NO, text= "Enter Flight Status", font=('bold', 10))

    Scheduled_time.place(x=18, y=10)
    Arival_time.place(x=18,y=40)
    Flight_no.place(x=18,y=70)
    Flight_name.place(x=18,y=100)
    Departed_place.place(x=18,y=130)
    Status.place(x=18,y=160)

    a1=Entry(NO)
    a1.place(x=190,y=10)
    a2=Entry(NO)
    a2.place(x=190, y=40)
    a3=Entry(NO)
    a3.place(x=190,y=70)
    a4=Entry(NO)
    a4.place(x=190,y=100)
    a5=Entry(NO)
    a5.place(x=190,y=130)
    a6=Entry(NO)
    a6.place(x=190,y=160)

    Button(NO, text='SUBMIT',command=Ainsert, height=3, width=20).place(x=30, y=190)
    Button(NO, text='EXIT',command=NO.destroy, height=3, width=20).place(x=220,y=190)

    NO.mainloop()

def ad():
    NW = Toplevel(root)
    NW.geometry("1000x667")
    NW.title("Admin Account")
    bg2 = PhotoImage(file="C:/Users/lenovo/Desktop/GUI/JANA2PIC.png")
    bg2_label = Label(NW, image=bg2)
    bg2_label.place(x=0, y=0, relwidth=1, relheight=1)
    B11 = Button(NW, text="Add Departure Flights", command=Dadd, padx=100, pady=100)
    B11.place(x=145, y=50)
    B12 = Button(NW, text="Add Arival Flights", command=Aadd, padx=100, pady=100)
    B12.place(x=545, y=50)
    B22 = Button(NW, text="View Flights", command=Vi, padx=127, pady=100)
    B22.place(x=145, y=350)
    B23 = Button(NW, text="Exit", command=lambda: NW.destroy(), padx=137, pady=100)
    B23.place(x=545, y=350)

    NW.mainloop()

def AVi():
    NA = Toplevel(root)
    NA.title("Arival Information")
    db = mys.connect(host='localhost', user='root', passwd='jane', database="Airliance")
```

```
cur = db.cursor()
cur.execute("Select * from Arival")

Atree = ttk.Treeview(NA)
Atree['show']='headings'

s = ttk.Style(NA)
s.theme_use("classic")
s.configure(".", font=('Helvetica',9))
s.configure("Treeview.Heading", foreground='Blue', font=('Algerian',11))

Atree["columns"]=("Scheduled_time", "Arival_time", "Flight_no","Flight_name","Departed_Place", "Status")

Atree.column("Scheduled_time",width=50, minwidth=50,  anchor=tk.CENTER)
Atree.column("Arival_time", width=100, minwidth=100, anchor=tk.CENTER)
Atree.column("Flight_no", width=100, minwidth=100, anchor=tk.CENTER)
Atree.column("Flight_name", width=100, minwidth=100, anchor=tk.CENTER)
Atree.column("Departed_Place", width=200, minwidth=200, anchor=tk.CENTER)
Atree.column("Status", width=100, minwidth=100, anchor=tk.CENTER)



Atree.heading("Scheduled_time", text="Time", anchor=tk.CENTER)
Atree.heading("Arival_time", text="Arival Time" ,anchor=tk.CENTER)
Atree.heading("Flight_no", text="Flight No", anchor=tk.CENTER)
Atree.heading("Flight_name", text="Flight Name", anchor=tk.CENTER)
Atree.heading("Departed_Place", text="Place of Departure", anchor=tk.CENTER)
Atree.heading("Status", text="Status", anchor=tk.CENTER)

i = 0
for ro in cur:
   Atree.insert('', i,text="",values=(ro[0],ro[1],ro[2],ro[3],ro[4],ro[5]))
   i=i + 1

hsb=ttk.Scrollbar(NA, orient="vertical")
```

```
    hsb.configure(command=Atree.yview)
    Atree.configure(yscrollcommand=hsb.set)
    hsb.pack(fill=Y, side= RIGHT)

    hsb=ttk.Scrollbar(NA, orient="horizontal")


    hsb.configure(command=Atree.xview)
    Atree.configure(xscrollcommand=hsb.set)
    hsb.pack(fill=X, side= BOTTOM)

    Atree.pack()

def DVi():
    ND = Toplevel(root)
    ND.title("Departure Information")
    db = mys.connect(host='localhost', user='root', passwd='jane', database="Airliance")
    cur = db.cursor()
    cur.execute("Select * from Departure")

    Dtree = ttk.Treeview(ND)
    Dtree['show']='headings'

    s = ttk.Style(ND)
    s.theme_use("classic")
    s.configure(".", font=('Helvetica',9))
    s.configure("Treeview.Heading", foreground='red', font=('Algerian',11))

    Dtree["columns"]=("Scheduled_time", "Flight_no","Flight_name","Destination", "Status","Gate_no")

    Dtree.column("Scheduled_time",width=50, minwidth=50, anchor=tk.CENTER)
    Dtree.column("Flight_no", width=100, minwidth=100, anchor=tk.CENTER)
    Dtree.column("Flight_name", width=100, minwidth=100, anchor=tk.CENTER)
    Dtree.column("Destination", width=200, minwidth=200, anchor=tk.CENTER)
    Dtree.column("Status", width=90, minwidth=90, anchor=tk.CENTER)
    Dtree.column("Gate_no", width=90, minwidth=90, anchor=tk.CENTER)

    Dtree.heading("Scheduled_time", text="Time", anchor=tk.CENTER)
    Dtree.heading("Flight_no", text="Flight No", anchor=tk.CENTER)
    Dtree.heading("Flight_name", text="Flight Name", anchor=tk.CENTER)
    Dtree.heading("Destination", text="Destination", anchor=tk.CENTER)
    Dtree.heading("Status", text="Status", anchor=tk.CENTER)
    Dtree.heading("Gate_no", text="Gate No" ,anchor=tk.CENTER)

    i = 0
    for ro in cur:
        Dtree.insert('', i,text="",values=(ro[0],ro[1],ro[2],ro[3],ro[4],ro[5]))
        i=i + 1

    hsb=ttk.Scrollbar(ND, orient="horizontal")

    hsb.configure(command=Dtree.xview)
    Dtree.configure(xscrollcommand=hsb.set)
    hsb.pack(fill=X, side= BOTTOM)

    hsb=ttk.Scrollbar(ND, orient="vertical")

    hsb.configure(command=Dtree.yview)
    Dtree.configure(yscrollcommand=hsb.set)
    hsb.pack(fill=Y, side= RIGHT)

    Dtree.pack()

def Vi():
    NView = Toplevel(root)
    NView.geometry("960x639")
    bg3 = PhotoImage(file="C:/Users/lenovo/Desktop/GUI/jana3pi.png")
    bg3_label = Label(NView, image=bg3)
    bg3_label.place(x=0, y=0, relwidth=1, relheight=1)

    DB= Button (NView, text="Departure Flights",command=DVi, padx=100, pady=80)
    DB.place(x=70, y=400)
    DA= Button (NView, text="Arival Flights",command=AVi, padx=100, pady=80)
    DA.place (x=371, y=400)
    B3 = Button(NView, text="Exit", command=lambda: NView.destroy() , padx=120, pady=80)
    B3.place(x=650, y=400)

    NView.mainloop()

bg = PhotoImage(file="C:/Users/lenovo/Desktop/GUI/jan.png")
bg_label = Label(root, image=bg)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)

B1 = Button(root, text="Admin", command=ad, padx=103, pady=50)
B1.place(x=200, y=500)

B2 = Button(root, text="User", command=Vi, padx=109, pady=50)
B2.place(x=500, y=500)

B3 = Button(root, text="Exit", command=lambda: root.destroy() , padx=109, pady=50)
B3.place(x=800, y=500)

root.mainloop()
```

SAMPLE OUTPUT

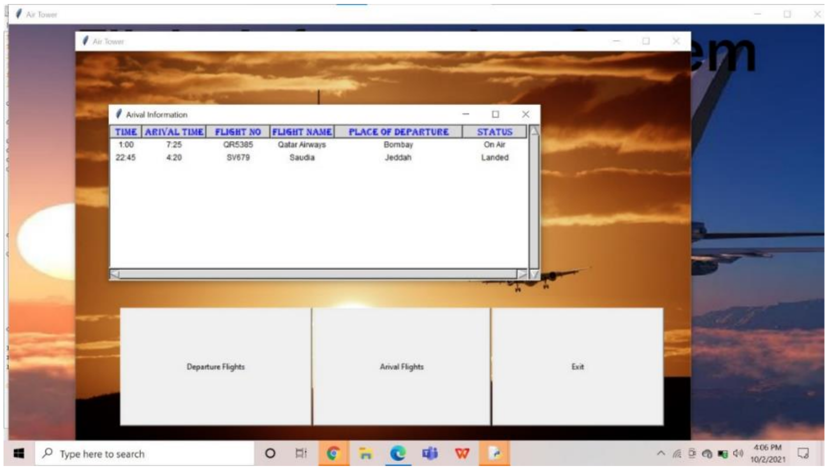




Enter Time (Eg:Hr:min) 12:30
Enter Flight Number SV217
Enter Flight Name Saudia
Enter Destination Dammam
Enter Flight Status Boarding Open
Enter Gate Number S3

Departure Information

TIME	FLIGHT NO	FLIGHT NAME	DESTINATION	STATUS	GATE NO
12:30	SV217	Saudia	Dammam	Boarding Open	S3



Arival Information

TIME	ARIVAL TIME	FLIGHT NO	FLIGHT NAME	PLACE OF DEPARTURE	STATUS
1:00	7:25	QR5385	Qatar Airways	Bombay	On Air
22:45	4:20	SV679	Saudia	Jeddah	Landed

PROBLEMS AND SOLUTIONS

- **ERRORS ENCOUNTERED:**
- Syntax errors when typing the program
- Errors in installing the tkinter module, MySQL connector
- Errors while using MySQL connector during program and successfully getting desired results in MySQL database

- **HOW THEY WERE RECTIFIED:**
- By correcting the Syntax.
- By importing the module's frontend and backend
- By successful installation of the connector and checking the connectivity using python.

SUGGESTED ENHANCEMENTS

- The program can be made more detailed and complex by adding more functionalities like searching for a specific record, deleting all records at once, etc.
- The range of arrival and departure details can be broadened, i.e, duration, fee, etc. can be included in the program admin
- The overall tkinter interface can be made more attractive and professional.

Bibligraphy

BOOKS

a) NCERT text-book class-12
b) Sumita Arora class-11
c) Sumita Arora class-12

Others:-

a)stackflow

b)codemy

c)youtube