**A**

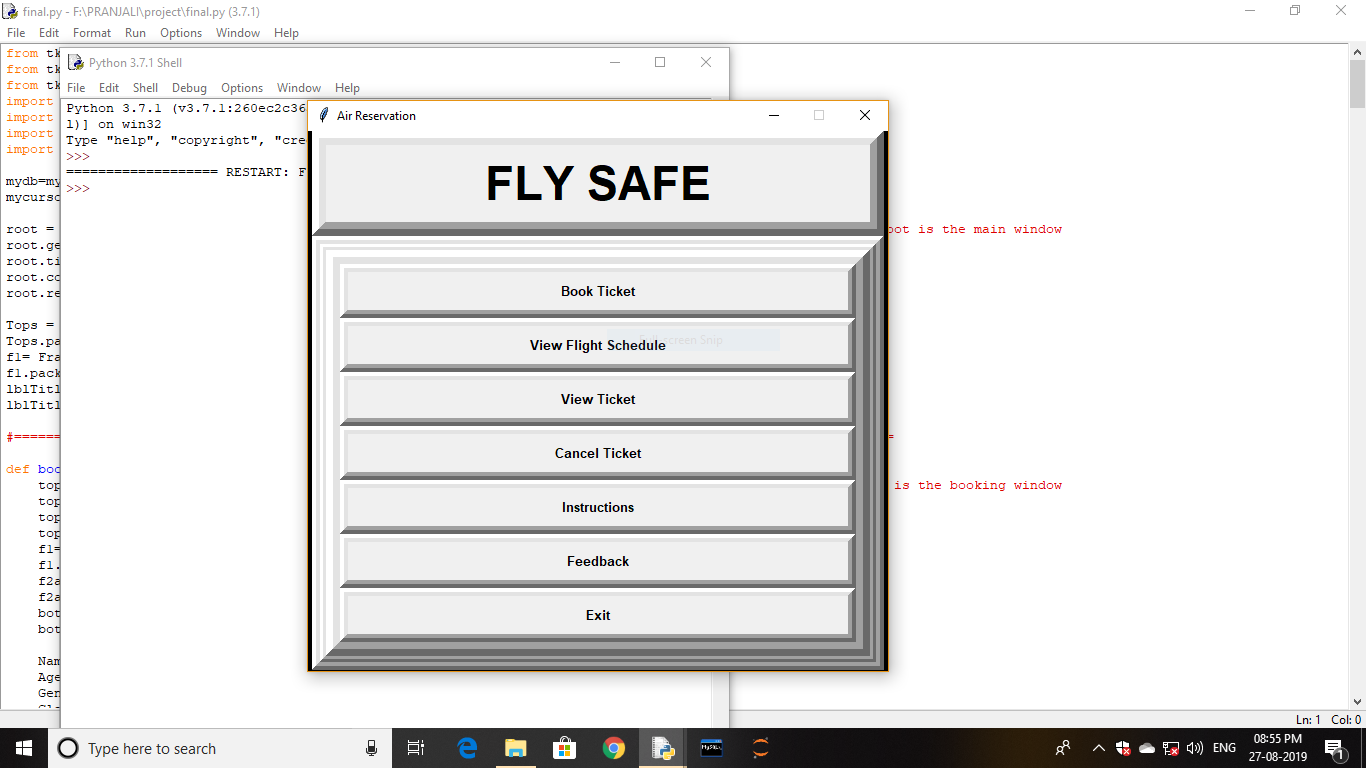
**PROJECT REPORT**

**OF**

**INFORMATICS PRACTICES**

**ON**

**AIR RESERVATION**

****

**Submitted By -**

Pranjali Kothari

XII(Science)

(Year 2019-20)

Roll no. -

**CERTIFICATE**

This is to certify that Miss Pranjali Kothari of Class XII(Science) of Army Public School Devlali has successfully completed the project in Informatics Practices as a part of her curriculum academic year 2019-20.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner’s Principal’s External Examiner’s**

**Signature Signature Signature**

**ACKNOWLEDGEMENT**

I would like to give my sincere gratitude to our Principal Mrs. Medha Gode for giving me an opportunity to do this project.

I express my sincere thanks to our Informatics Practices teacher Mrs. Archana S. Patil who guided me throughout the completion of this project. This would have not been possible without the cooperation and direction provided by her.

**Pranjali Kothari**

**XII (Science)**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr no.** | **Title** | | **Page no.** |
| 1. | Description of Project | | 5 |
| 2. | Technical Specification | | 6 |
|  | 2.1 | Hardware Requirement |
| 2.2 | Software Requirement |
| 3. | Use Case Diagram | | 7 |
| 4. | Screen Layouts with Coding | | 8 |
|  | 4.1 | Connectivity Statements and Imports Used in Project | 9 |
| 4.2 | Screenshot of Database and Tables used | 10 |
| 4.3 | Python Coding along with Screenshots | 12 |
| 5. | Bibliography | | 47 |
| 5.1 | Reference Books |
| 5.2 | Reference Websites |

**Description of Project**

User characteristics

The users of system are categorised as follows:

* Administrator – should be able to update and have access to system maintenance function.
* Client – should be able to satisfy his queries and take test practice.

The system has the following features:

The system is an integrated system that will provide centralized access to the Air Reservation Project.

The system will reduce all type of paper work which were used redundantly.

The system will have good data management, data reliability and accuracy. This will also reduce inaccurate and non-availability of information on time.

**Technical Specifications**

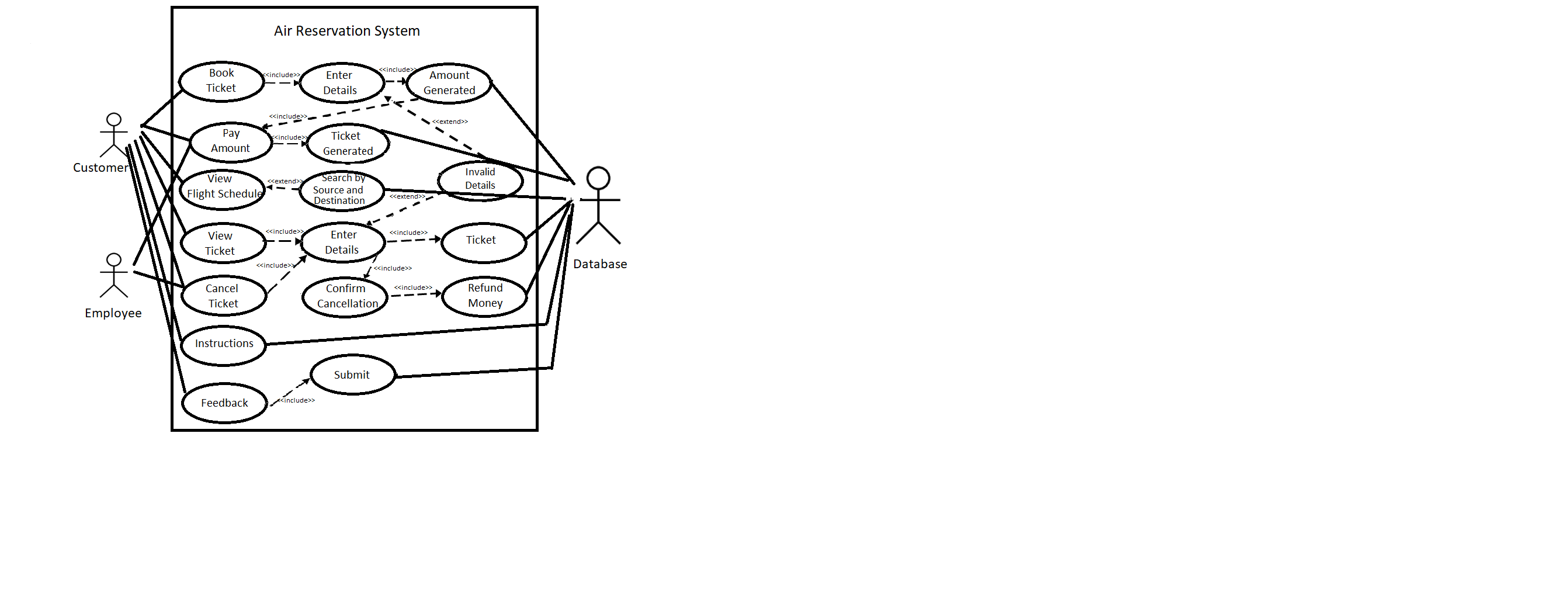
Hardware Requirement (Minimum)

* Processor Intel R Core i3
* CPU E2180@ 2.00 GHz
* Monitor – color CRT/TFT 15.5” Monitor
* System Type – 32 bits or 64 bits
* RAM – 2GB
* Hard disk – 128 GB DDR3

Software Requirement (Minimum)

* Python IDLE 3.6
* MySQL Server 5.1
* MS-Word 2007 or higher
* Windows Professional(Service pack-1) or higher

**Use Case diagram**



***SCREEN LAYOUTS WITH CODING***

**Connectivity Statements and Imports Used in Project**

**Imports:**

from tkinter import \*

from tkinter import Tk, StringVar, ttk

from tkinter import messagebox

import mysql.connector

import random

import datetime

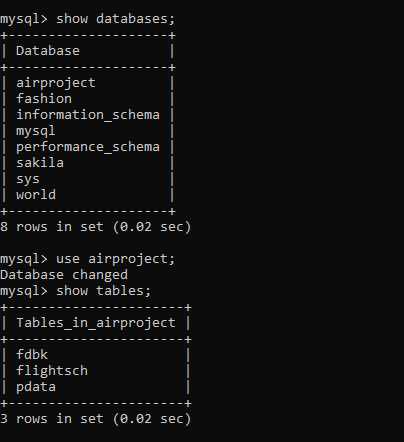
**Connectivity statements:**

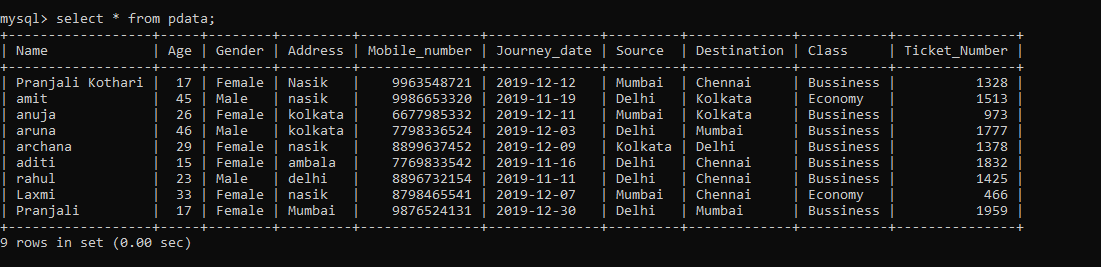
mydb=mysql.connector.connect(host='localhost',user='root',passwd='0305',database='airproject')

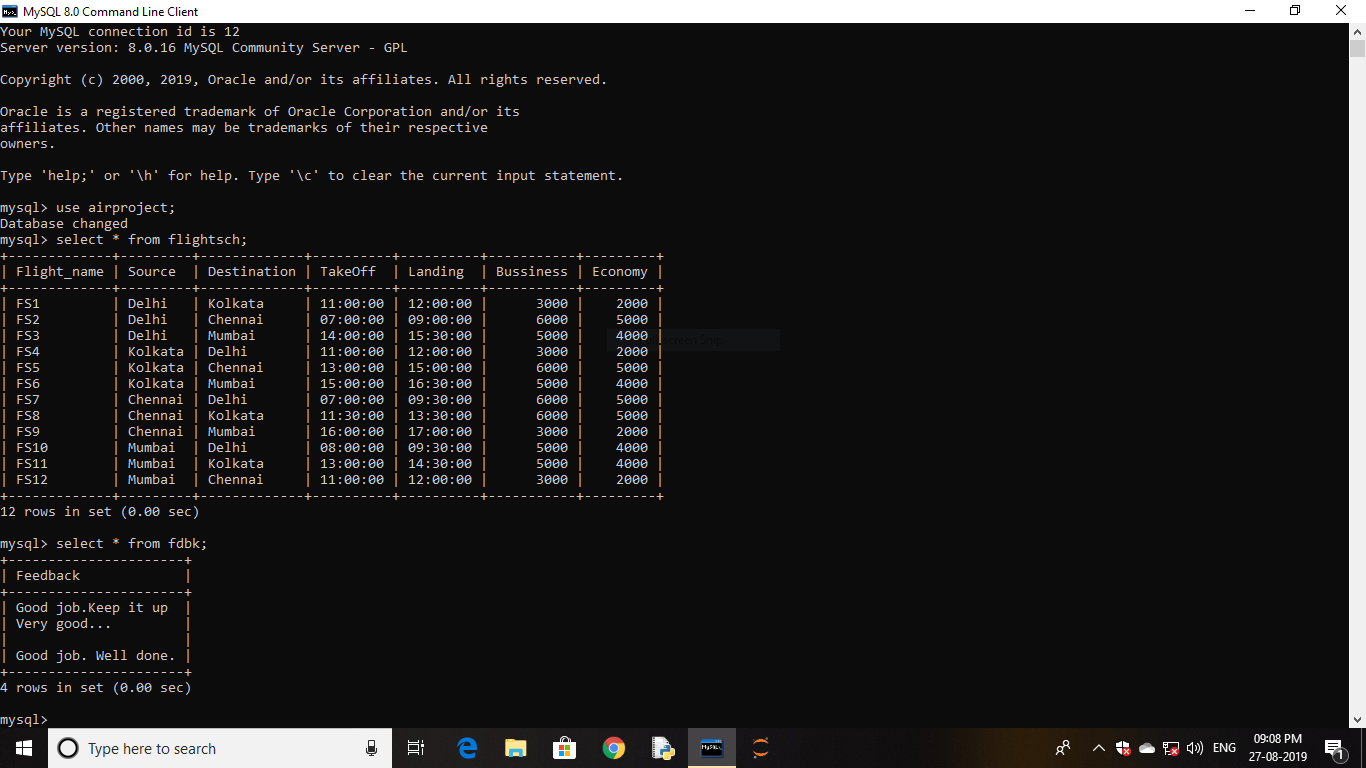
mycursor = mydb.cursor()

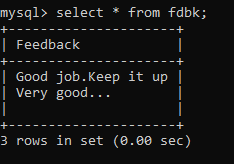
***SCREENSHOT OF DATABASE AND***

***TABLES USED***

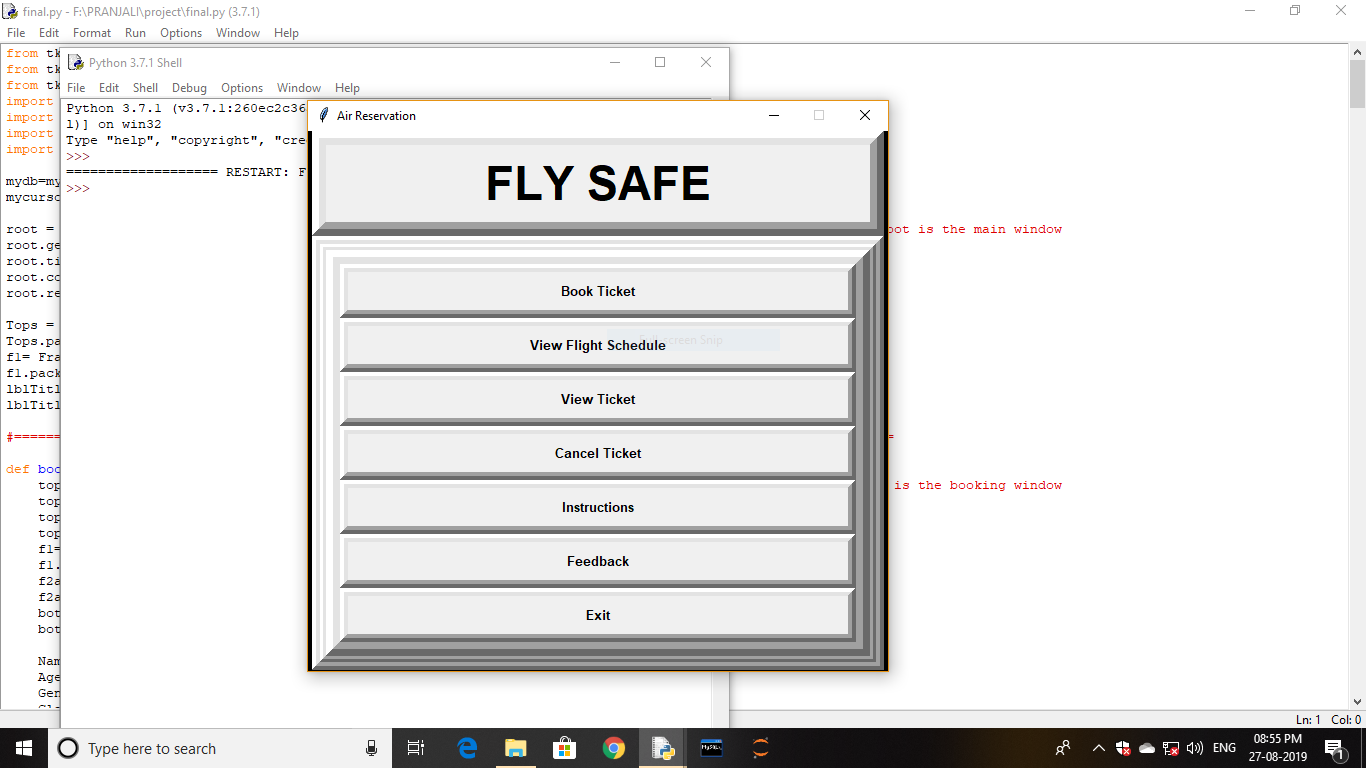








**Python Coding Along With Screenshots**

****

from tkinter import \*

from tkinter import Tk, StringVar, ttk

from tkinter import messagebox

import mysql.connector

import random

import time;

import datetime

mydb=mysql.connector.connect(host='localhost',user='root',passwd='0305',database='airproject')

mycursor=mydb.cursor()

root = Tk() #root is the main window

root.geometry("580x540+300+100")

root.title("Air Reservation")

root.configure(background='black')

root.resizable(0,0)

Tops = Frame(root, width =1000, height =600, bd=14, relief='raise')

Tops.pack(side=TOP)

f1= Frame(root, width =1000,height=500, bd=8, relief='raise')

f1.pack(side='top')

lblTitle=Label(Tops,font=('arial',35,'bold'),text='FLY SAFE', bd=10, width=18, justify='center')

lblTitle.grid(row=0,column=0)

**#==================================Functions===================================**

**Book Ticket**

def booking():

top=Toplevel() #top is the booking window

top.title('Booking window')

top.geometry('1010x500+200+100')

top.resizable(0,0)

f1= Frame(top, width =700,height=350, bd=8, relief='raise')

f1.pack(side=TOP)

f2a = Frame(f1, width =700,height=350, bd=6, relief='raise')

f2a.pack(side=LEFT)

bottomLeft1 = Frame(f2a, width=450, height=450, bd=14,relief='raise')

bottomLeft1.pack(side=RIGHT)

Name = StringVar()

Age = int()

Gender = StringVar()

Class = StringVar()

Address = str()

Year = int()

Month = int()

Date = int()

MobileNumber = int()

Source = StringVar()

Destination = StringVar()

tno = int()

Name = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Name', bd=16, anchor='w')

Name.grid(row=0,column=0)

Name = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Name', bd=10, width=10,

bg='#ffffff', justify='right')

Name.grid(row=0,column=1)

Age = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Age', bd=16,anchor='w')

Age.grid(row=2,column=0)

Age = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Age', bd=10, width=10,

bg='#ffffff', justify='right')

Age.grid(row=2,column=1)

lblGender = Label(bottomLeft1,font=('arial',20,'bold'), text='Gender', bd=4,anchor='e')

lblGender.grid(row=0,column=2)

cboGender =ttk.Combobox(bottomLeft1,textvariable=Gender, state='readonly',\

font=('arial',20,'bold'), width=10)

cboGender['value']=('','Male','Female','Others')

cboGender.current(0)

cboGender.grid(row=0,column=3)

lblClass = Label(bottomLeft1,font=('arial',20,'bold'), text='Class', bd=4,anchor='w')

lblClass.grid(row=2,column=2)

cboClass =ttk.Combobox(bottomLeft1,textvariable=Class, state='readonly',

font=('arial',20,'bold'), width=10)

cboClass['value']=('','Economy','Bussiness')

cboClass.current(0)

cboClass.grid(row=2,column=3)

lblSource = Label(bottomLeft1,font=('arial',20,'bold'), text='Source', bd=4,anchor='w')

lblSource.grid(row=3,column=2)

cboSource =ttk.Combobox(bottomLeft1,textvariable=Source, state='readonly',

font=('arial',20,'bold'), width=10)

cboSource['value']=('','Delhi','Mumbai','Kolkata','Chennai')

cboSource.current(0)

cboSource.grid(row=3,column=3)

lblDestination = Label(bottomLeft1,font=('arial',20,'bold'), text='Destination', bd=4,anchor='w')

lblDestination.grid(row=4,column=2)

cboDestination =ttk.Combobox(bottomLeft1,textvariable=Destination, state='readonly',

font=('arial',20,'bold'), width=10)

cboDestination['value']=('','Delhi','Mumbai','Kolkata','Chennai')

cboDestination.current(0)

cboDestination.grid(row=4,column=3)

Address = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Address', bd=16,anchor='w')

Address.grid(row=3,column=0)

Address = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Address', bd=10,

width=10, bg='#ffffff', justify='right')

Address.grid(row=3,column=1)

MobileNumber = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Mobile Number',

bd=16,anchor='w')

MobileNumber.grid(row=4,column=0)

MobileNumber = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='MobileNumber',

bd=10, width=10, bg='#ffffff', justify='right')

MobileNumber.grid(row=4,column=1)

Year = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Journey: Year', bd=8,anchor='w')

Year.grid(row=6,column=0)

Year = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Year', bd=10, width=8,

bg='#ffffff', justify='right')

Year.grid(row=6,column=1)

Month = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Month', bd=8,anchor='w')

Month.grid(row=6,column=2)

Month = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Month', bd=10, width=8,

bg='#ffffff', justify='right')

Month.grid(row=6,column=3)

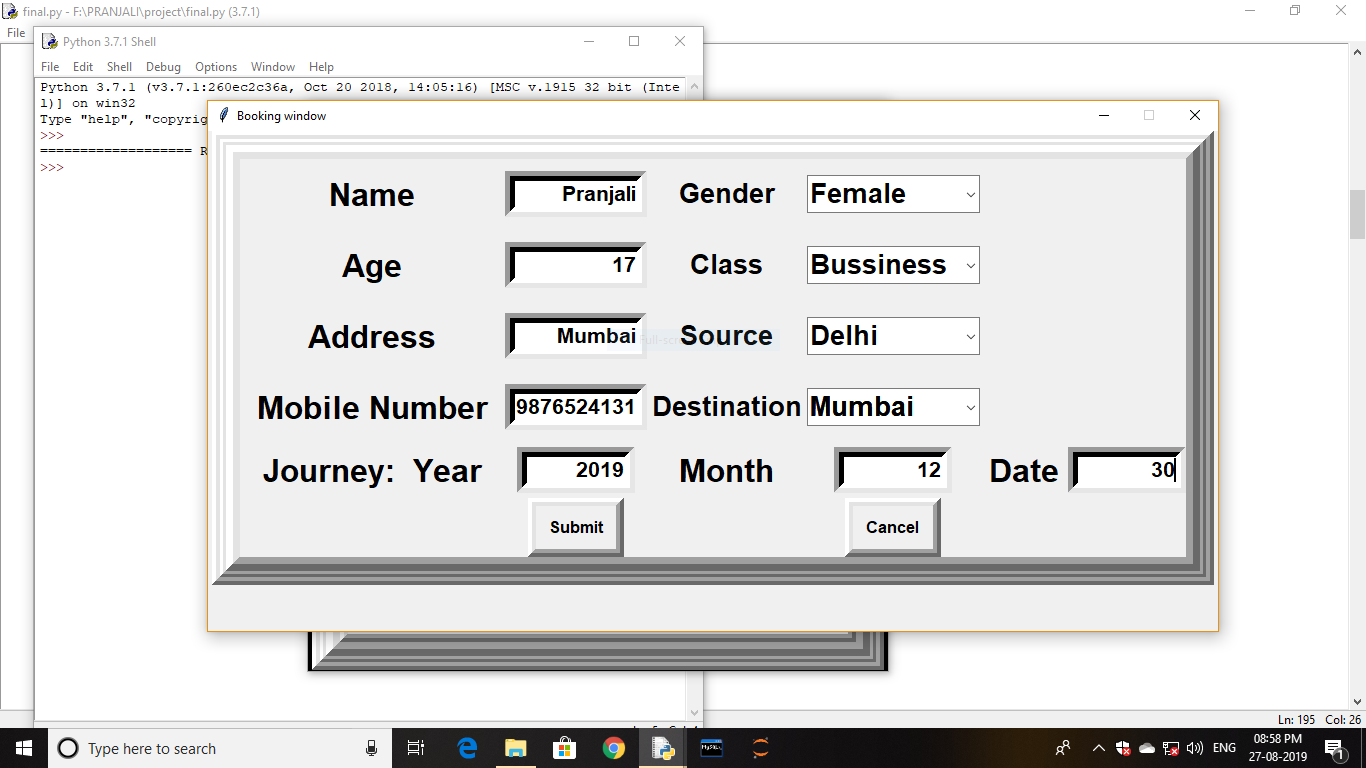
Date = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Date', bd=8,anchor='w')

Date.grid(row=6,column=4)

Date = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Date', bd=10, width=8,

bg='#ffffff', justify='right')

Date.grid(row=6,column=5)



def submit():

top2=Toplevel()

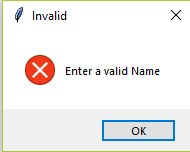
top2.title('Ticket Amount')

top2.geometry('330x170+500+250')

top2.resizable(0,0)

f1= Frame(top2, width =40,height=20, bd=8, relief='raise')

f1.pack()

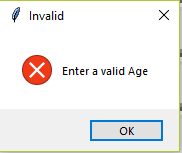
 name=Name.get()

if ((name.isalpha()==False) or (len(name)>35) ):

messagebox.showerror('Invalid','Enter a valid Name')

top.destroy()

return

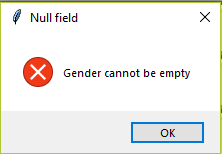
 age=Age.get()

if ((age.isdigit()==False) or (len(age)>3)):

messagebox.showerror('Invalid','Enter a valid Age')

top.destroy()

return

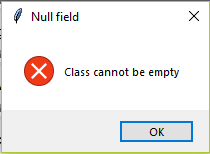
 gender=Gender.get()

if gender=="":

messagebox.showerror('Null field','Gender cannot be empty')

top.destroy()

return

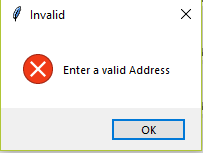
 clas=Class.get()

if clas=="":

messagebox.showerror('Null field','Class cannot be empty')

top.destroy()

return

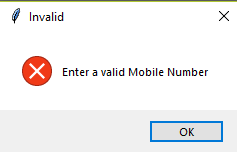
 address=Address.get()

if ((address.isalnum()==False) or (len(address)>50)):

messagebox.showerror('Invalid','Enter a valid Address')

top.destroy()

return

 mob=MobileNumber.get()

if ((mob.isdigit()==False) or (len(mob)>10)):

messagebox.showerror('Invalid','Enter a valid Mobile

Number’)

top.destroy()

return

a=str(Year.get())

p=int(a)

b=str(Month.get())

q=int(b)

c=str(Date.get())

r=int(c)

jr\_date=datetime.date(p,q,r)

if(jr\_date<=datetime.date.today()):

messagebox.showerror('Error','Enter a valid date')

top.destroy()

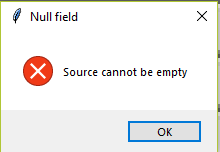
return

elif (jr\_date==""):

messagebox.showerror('Null field','Journey Date cannot be empty')

top.destroy()

return

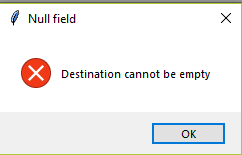
 source=Source.get()

if source=="":

messagebox.showerror('Null field','Source cannot be empty')

top.destroy()

return

 dest=dest.get()

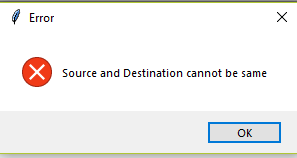
if Destination=="":

messagebox.showerror('Null field','Destination cannot be

empty')

top.destroy()

return

 if (source==dest):

messagebox.showerror('Error','Source and Destination

cannot be same')

top.destroy()

return

tno=random.randint(200,2000)

am=StringVar()

sql=mycursor.ecexute("select "+clas+" from flightsch where Source = '"+source+"' and

Destination = '"+dest+"';")

am=mycursor.fetchone()

for i in am:

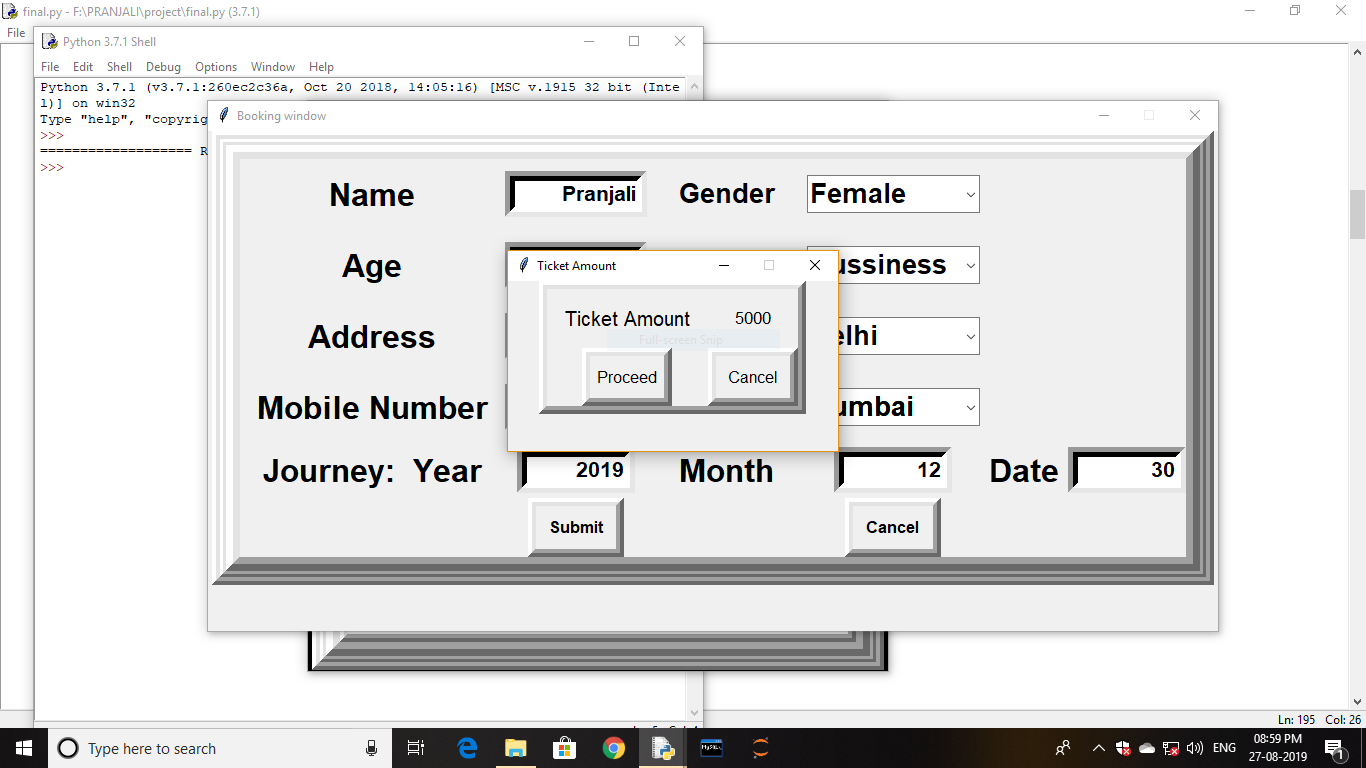
str1=i

lbl1 = Label(f1, font=('arial', 15), text="Ticket Amount", bd=17, anchor='center')

lbl1.grid(row=0,column=0)

lbl2 = Label(f1, font=('arial', 13), text=str1, bd=17, anchor='center')

lbl2.grid(row=0,column=2)



def proceed():

top2.destroy()

top.destroy()

top1=Toplevel()

top1.title('Ticket Number')

top1.geometry('675x490+445+150')

top1.resizable(0,0)

f1= Frame(top1, width =40,height=20, bd=8, relief='raise')

f1.pack()

f2a = Frame(f1, width =40,height=20, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=100, bd=14,relief='raise')

bottomLeft1.pack()

sql=mycursor.execute("insert into pdata(Name,Age,Gender,Class,Address,Mobile\_Number,\

Journey\_date,Source,Destination,Ticket\_Number) values(‘{}’,{},’{}’,’{}’,’{}’,{},’{}’,’{}’,’{}’,{})” \

.format(name,age,gender,clas,address,mob,jr\_date,source,dest,tno))

mydb.commit()

lbl = Label(bottomLeft1, font=('arial', 20, 'bold'), text="Ticket Booked", bd=17)

lbl.grid(row=0,column=1)

lbl1 = Label(bottomLeft1, font=('arial', 20, 'bold'), text="Ticket number", bd=15, anchor='nw')

lbl1.grid(row=2,column=0)

lbl2 = Label(bottomLeft1, font=('arial', 20, 'bold'), text=tno, bd=13, anchor='nw')

lbl2.grid(row=2,column=2)

sql1=mycursor.execute("select Flight\_name from flightsch where Source='{}' and

Destination='{}'".format(source,dest))

fname=mycursor.fetchone()

lbl3 = Label(bottomLeft1, font=('arial', 20, 'bold'), text="Flight Name", bd=15, anchor='nw')

lbl3.grid(row=4,column=0)

lbl4 = Label(bottomLeft1, font=('arial', 20, 'bold'), text=fname, bd=13, anchor='nw')

lbl4.grid(row=4,column=2)

sql2=mycursor.execute("select TakeOff from flightsch where Source='{}' and

Destination='{}'".format(source,dest))

tt=mycursor.fetchone()

lbl5 = Label(bottomLeft1, font=('arial', 20, 'bold'), text="Take Off", bd=15, anchor='nw')

lbl5.grid(row=5,column=0)

lbl6 = Label(bottomLeft1, font=('arial', 20, 'bold'), text=tt, bd=13, anchor='nw')

lbl6.grid(row=5,column=2)

sql3=mycursor.execute("select Landing from flightsch where Source='{}' and

Destination='{}'".format(source,dest))

tl=mycursor.fetchone()

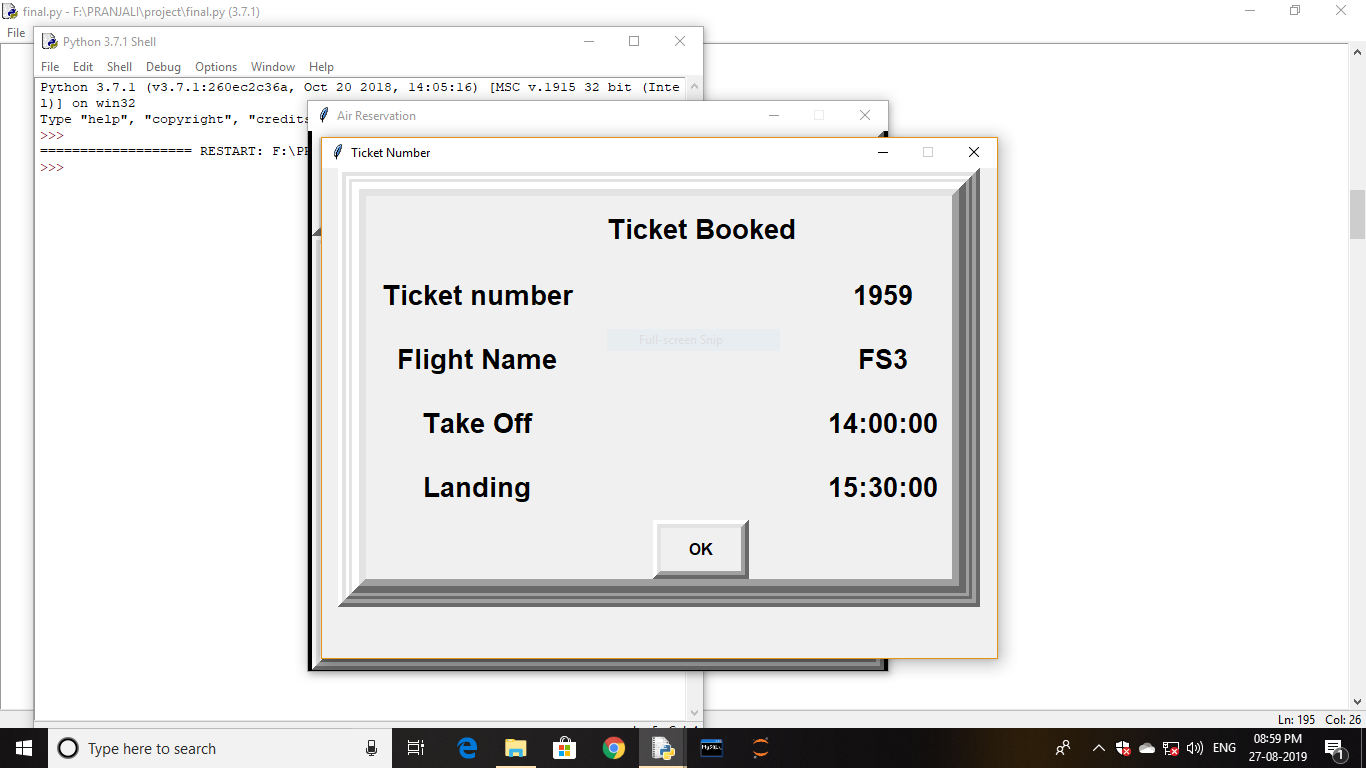
lbl7 = Label(bottomLeft1, font=('arial', 20, 'bold'), text="Landing", bd=15, anchor='nw')

lbl7.grid(row=6,column=0)

lbl8 = Label(bottomLeft1, font=('arial', 20, 'bold'), text=tl, bd=13, anchor='nw')

lbl8.grid(row=6,column=2)

top.destroy()



def ca():

top1.destroy()

return

ok = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="OK", width=6, command=ca).grid(row=8,column=1)

def Cancel():

top2.destroy()

top.destroy()

proceed = Button(f1, padx=8, pady=8, bd=8, fg='black', font=('arial',12),

text="Proceed", width=6, command=proceed).grid(row=2,column=0,)

can = Button(f1, padx=8, pady=8, bd=8, fg='black', font=('arial',12),

text="Cancel", width=6, command=Cancel).grid(row=2,column=2,sticky='w')

def cancel():

top.destroy()

return

submit = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Submit", width=6, command=submit).grid(row=10,column=1)

cancel = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Cancel", width=6, command=cancel).grid(row=10,column=3)

**View Flight Schedule**

def schedule():

sch=Toplevel() #sch is the view schedule window

sch.title('Schedule window')

sch.geometry('400x250+400+200')

sch.resizable(0,0)

f1= Frame(sch, width =700,height=350, bd=8, relief='raise')

f1.pack(side=TOP)

f2a = Frame(f1, width =700,height=350, bd=6, relief='raise')

f2a.pack(side=LEFT)

bottomLeft1 = Frame(f2a, width=450, height=450, bd=14,relief='raise')

bottomLeft1.pack(side=LEFT)

Source=StringVar()

Destination=StringVar()

lblSource = Label(bottomLeft1,font=('arial',24,'bold'), text='Source', bd=4)

lblSource.grid(row=4,column=0, sticky=W)

cboSource =ttk.Combobox(bottomLeft1,textvariable=Source, state='readonly',

font=('arial',16,'bold'), width=8)

cboSource['value']=('','Mumbai','Kolkata','Chennai','Delhi')

cboSource.current(0)

cboSource.grid(row=4,column=1)

lblDestination = Label(bottomLeft1,font=('arial',24,'bold'), text='Destination', bd=4)

lblDestination.grid(row=6,column=0, sticky=W)

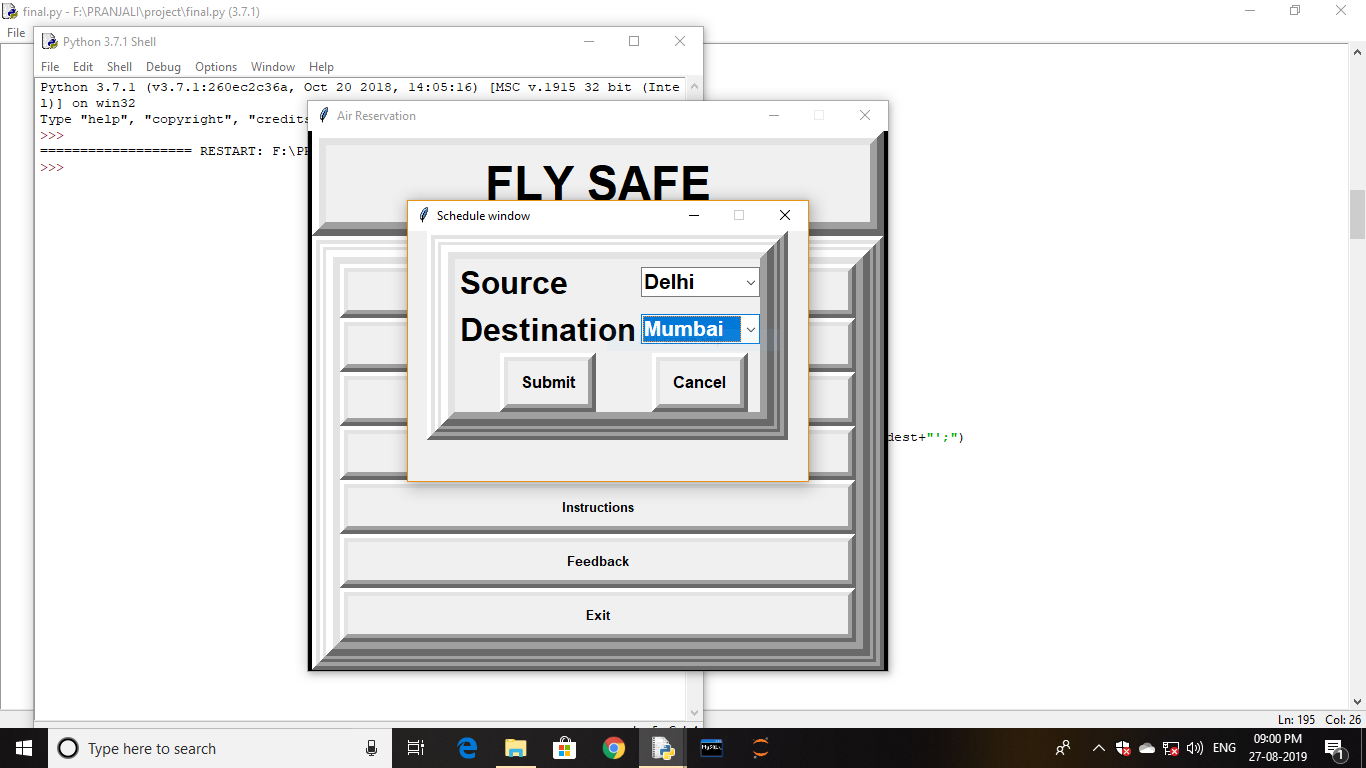
cboDestination =ttk.Combobox(bottomLeft1,textvariable=Destination, state='readonly',

font=('arial',16,'bold'), width=8)

cboDestination['value']=('','Mumbai','Kolkata','Chennai','Delhi')

cboDestination.current(0)

cboDestination.grid(row=6,column=1)



def submit():

sch1=Toplevel() # sch1 is the flight schedule window

sch1.title('Flight Details')

sch1.geometry('650x640+280+40')

sch1.resizable(0,0)

f1= Frame(sch1, width =40,height=20, bd=8, relief='raise')

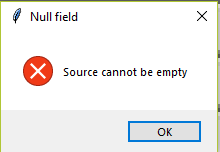
f1.pack()

f2a = Frame(f1, width =40,height=20, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=100, bd=14,relief='raise')

bottomLeft1.pack()

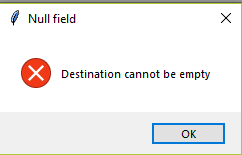
 source=Source.get()

if source=="":

messagebox.showerror('Null field','Source cannot be empty')

sch1.destroy()

return

 destination=Destination.get()

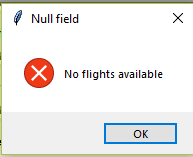
if destination=="":

messagebox.showerror('Null field','Destination cannot be

empty')

sch1.destroy()

return

 if (source==destination):

messagebox.showerror('Null field','No flights available')

sch1.destroy()

sch.destroy()

return

sql1=mycursor.execute("select Flight\_Name from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

a=mycursor.fetchone()

sql2=mycursor.execute("select Source from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

b=mycursor.fetchone()

sql3=mycursor.execute("select Destination from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

c=mycursor.fetchone()

sql4=mycursor.execute("select TakeOff from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

d=mycursor.fetchone()

sql5=mycursor.execute("select Landing from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

e=mycursor.fetchone()

sql6=mycursor.execute("select Bussiness from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

f=mycursor.fetchone()

sql7=mycursor.execute("select Economy from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

g=mycursor.fetchone()

fn = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Flight Name', bd=16, anchor='w')

fn.grid(row=1,column=0)

fn = Label(bottomLeft1, font=('arial', 18), text=a, bd=16, anchor='w')

fn.grid(row=1,column=2)

so = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Source', bd=16, anchor='w')

so.grid(row=2,column=0)

so = Label(bottomLeft1, font=('arial', 18), text=b, bd=16, anchor='w')

so.grid(row=2,column=2)

de = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Destination', bd=16, anchor='w')

de.grid(row=3,column=0)

de = Label(bottomLeft1, font=('arial', 18), text=c, bd=16, anchor='w')

de.grid(row=3,column=2)

tt = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Take Off', bd=16, anchor='w')

tt.grid(row=4,column=0)

tt = Label(bottomLeft1, font=('arial', 18), text=d, bd=16, anchor='w')

tt.grid(row=4,column=2)

la = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Landing', bd=16, anchor='w')

la.grid(row=5,column=0)

la = Label(bottomLeft1, font=('arial', 18), text=e, bd=16, anchor='w')

la.grid(row=5,column=2)

bu = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Bussiness Class', bd=16, anchor='w')

bu.grid(row=6,column=0)

bu = Label(bottomLeft1, font=('arial', 18), text=f, bd=16, anchor='w')

bu.grid(row=6,column=2)

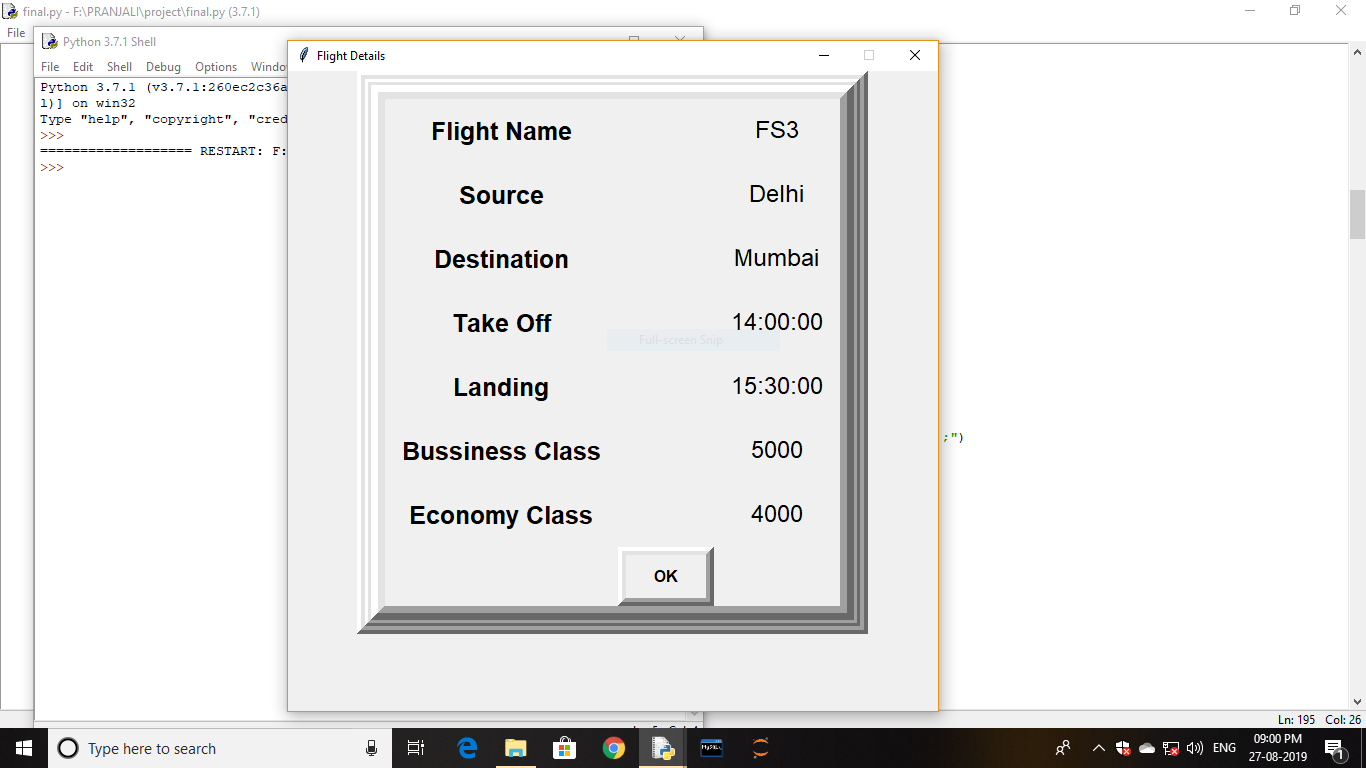
ec = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Economy Class', bd=16, anchor='w')

ec.grid(row=7,column=0)

ec = Label(bottomLeft1, font=('arial', 18), text=g, bd=16, anchor='w')

ec.grid(row=7,column=2)

sch.destroy()



def ca():

sch1.destroy()

return

ok = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="OK", width=6, command=ca).grid(row=8,column=1)

def cancel():

sch.destroy()

return

submit = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Submit", width=6, command=submit).grid(row=8,column=0)

cancel = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Cancel", width=6, command=cancel).grid(row=8,column=1)

**View Ticket**

def details():

top=Toplevel() #top is the view ticket details window

top.title('Details window')

top.geometry('550x360+300+100')

top.resizable(0,0)

f1= Frame(top, width =700,height=350, bd=8, relief='raise')

f1.pack(side=TOP)

f2a = Frame(f1, width =700,height=350, bd=6, relief='raise')

f2a.pack(side=LEFT)

bottomLeft1 = Frame(f2a, width=450, height=450, bd=14,relief='raise')

bottomLeft1.pack(side=LEFT)

Name = StringVar()

tno = int()

Source = StringVar()

Destination = StringVar()

Name = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Name', bd=16, anchor='w')

Name.grid(row=0,column=0)

Name = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Name', bd=10, width=16,

bg='#ffffff', justify='right')

Name.grid(row=0,column=1)

tno = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Ticket Number', bd=16, anchor='w')

tno.grid(row=2,column=0)

tno = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='tno', bd=10, width=16, bg='#ffffff',

justify='right')

tno.grid(row=2,column=1)

lblSource = Label(bottomLeft1,font=('arial',24,'bold'), text='Source', bd=4)

lblSource.grid(row=4,column=0)

cboSource =ttk.Combobox(bottomLeft1,textvariable=Source, state='readonly',

font=('arial',16,'bold'), width=9)

cboSource['value']=('','Mumbai','Kolkata','Chennai','Delhi')

cboSource.current(0)

cboSource.grid(row=4,column=1)

lblDestination = Label(bottomLeft1,font=('arial',24,'bold'), text='Destination', bd=4)

lblDestination.grid(row=6,column=0)

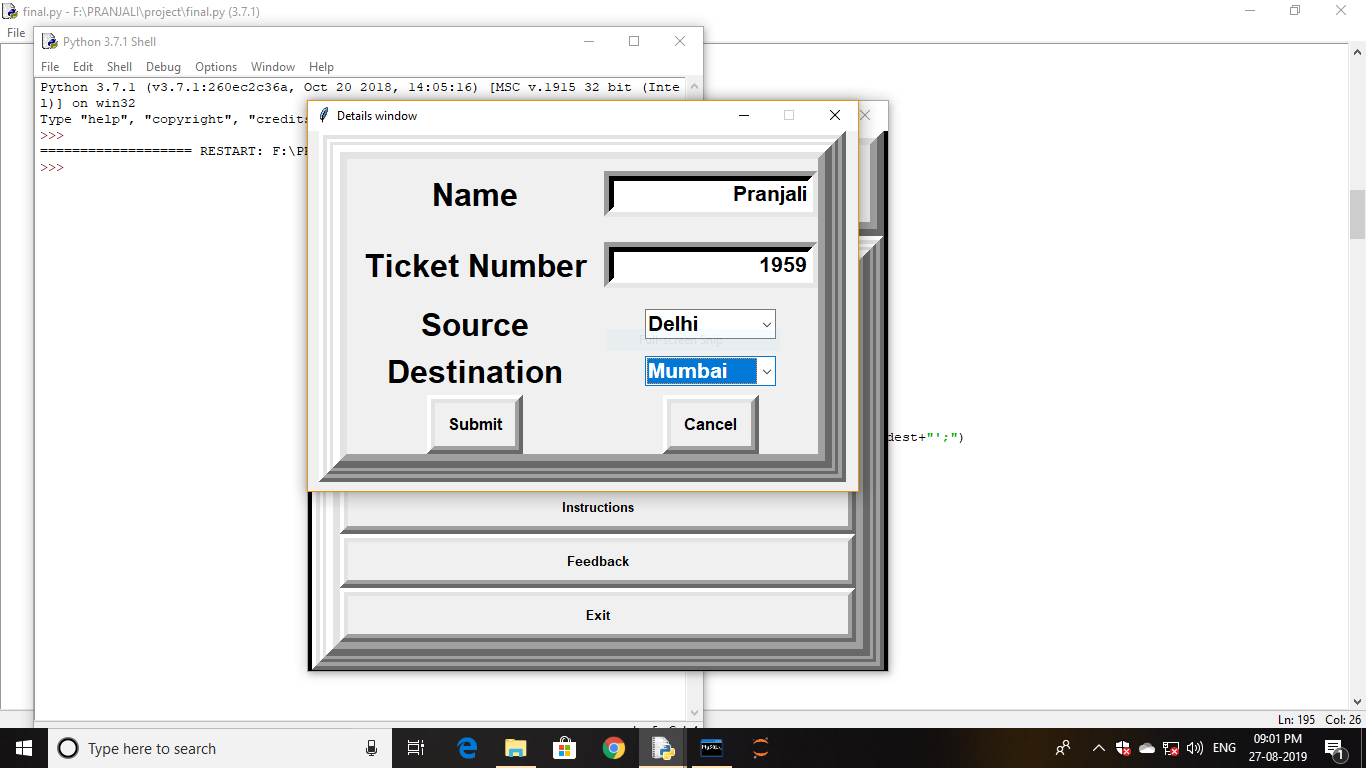
cboDestination =ttk.Combobox(bottomLeft1,textvariable=Destination, state='readonly',

font=('arial',16,'bold'), width=9)

cboDestination['value']=('','Mumbai','Kolkata','Chennai','Delhi')

cboDestination.current(0)

cboDestination.grid(row=6,column=1)



def submit():

top1=Toplevel() # top1 is the view ticket window

top1.title('Ticket Details')

top1.geometry('650x695+290+0')

top1.resizable(0,0)

f1= Frame(top1, width =40,height=20, bd=8, relief='raise')

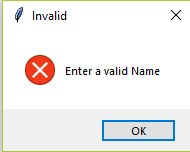
f1.pack()

f2a = Frame(f1, width =40,height=20, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=100, bd=14,relief='raise')

bottomLeft1.pack()

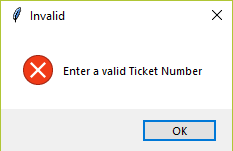
 name=Name.get()

if (name.isalpha()==False):

messagebox.showerror('Invalid','Enter a valid Name')

top1.destroy()

return

 Tno=tno.get()

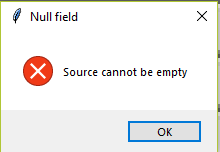
if (Tno.isdigit()==False):

messagebox.showerror('Invalid','Enter a valid Ticket

Number')

top1.destroy()

return

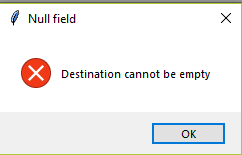
 source=Source.get()

if source=="":

messagebox.showerror('Null field','Source cannot be empty')

top1.destroy()

return

 destination=Destination.get()

if destination=="":

messagebox.showerror('Null field','Destination cannot be

empty')

top1.destroy()

return

sql1=mycursor.execute("select Name from pdata where Name='{}'and

Ticket\_Number={}".format(name,Tno))

a=mycursor.fetchone()

sql2=mycursor.execute("select Flight\_Name from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

b=mycursor.fetchone()

sql3=mycursor.execute("select TakeOff from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

c=mycursor.fetchone()

sql4=mycursor.execute("select Landing from flightsch where Source='{}'and

Destination='{}'".format(source,destination))

d=mycursor.fetchone()

sql5=mycursor.execute("select Class from pdata where Name='{}'and

Ticket\_Number={}".format(name,Tno))

e=mycursor.fetchone()

sql6=mycursor.execute("select Source from pdata where Name='{}'and

Ticket\_Number={}".format(name,Tno))

f=mycursor.fetchone()

sql7=mycursor.execute("select Destination from pdata where Name='{}'and

Ticket\_Number={}".format(name,Tno))

g=mycursor.fetchone()

sql8=mycursor.execute("select Journey\_date from pdata where Name='{}'and

Ticket\_Number={}".format(name,Tno))

h=mycursor.fetchone()

wish = Label(bottomLeft1, font=('arial', 20,'bold'), text='Happy Journey', bd=16, anchor='center')

wish.grid(row=0,column=1)

na = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Name', bd=16, anchor='w')

na.grid(row=1,column=0)

na = Label(bottomLeft1, font=('arial', 18), text=a, bd=16, anchor='w')

na.grid(row=1,column=2)

fn = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Flight Name', bd=16, anchor='w')

fn.grid(row=2,column=0)

fn = Label(bottomLeft1, font=('arial', 18), text=b, bd=16, anchor='w')

fn.grid(row=2,column=2)

to = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Take Off', bd=16, anchor='w')

to.grid(row=3,column=0)

to = Label(bottomLeft1, font=('arial', 18), text=c, bd=16, anchor='w')

to.grid(row=3,column=2)

la = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Landing', bd=16, anchor='w')

la.grid(row=4,column=0)

la = Label(bottomLeft1, font=('arial', 18), text=d, bd=16, anchor='w')

la.grid(row=4,column=2)

cl = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Class', bd=16, anchor='w')

cl.grid(row=5,column=0)

cl = Label(bottomLeft1, font=('arial', 18), text=e, bd=16, anchor='w')

cl.grid(row=5,column=2)

so = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Source', bd=16, anchor='w')

so.grid(row=6,column=0)

so = Label(bottomLeft1, font=('arial', 18), text=f, bd=16, anchor='w')

so.grid(row=6,column=2)

de = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Destination', bd=16, anchor='w')

de.grid(row=7,column=0)

de = Label(bottomLeft1, font=('arial', 18), text=g, bd=16, anchor='w')

de.grid(row=7,column=2)

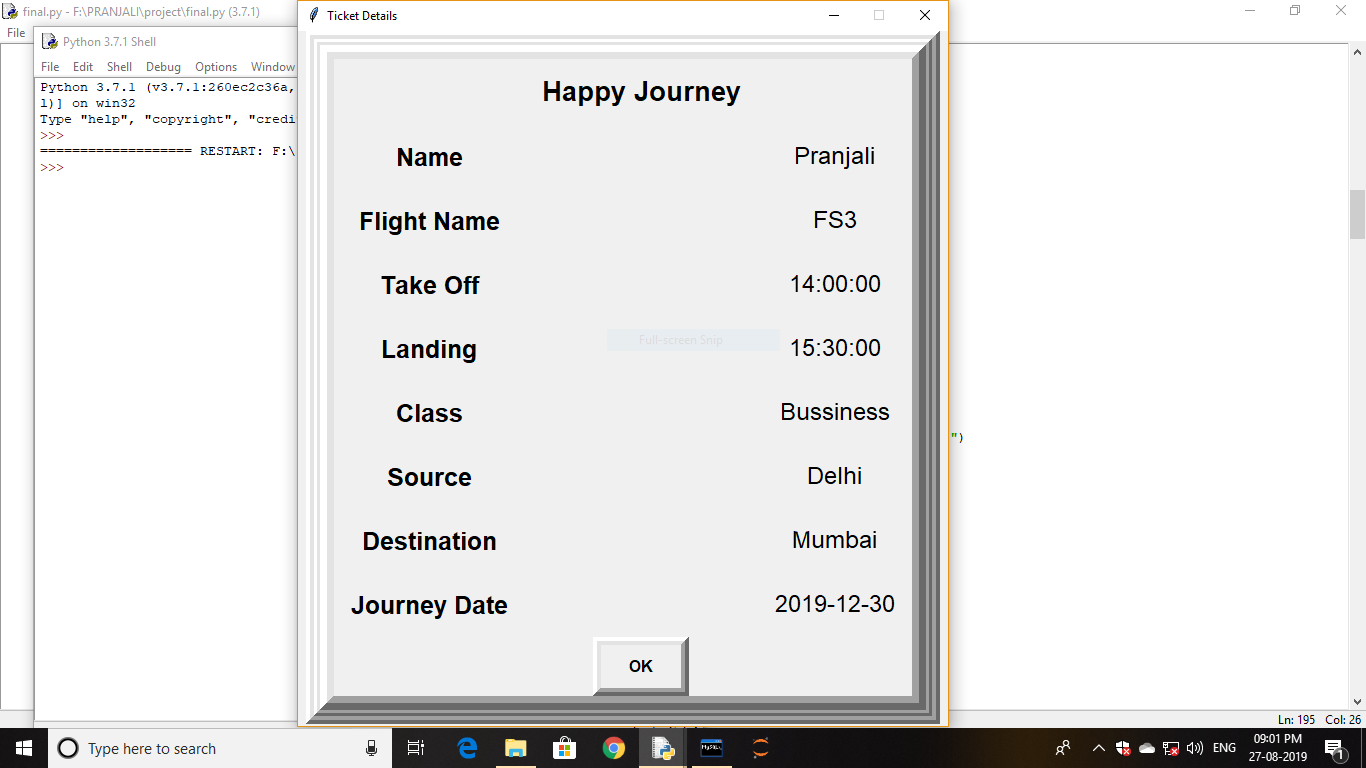
jd = Label(bottomLeft1, font=('arial', 19, 'bold'), text='Journey Date', bd=16, anchor='w')

jd.grid(row=8,column=0)

jd = Label(bottomLeft1, font=('arial', 18), text=h, bd=16, anchor='w')

jd.grid(row=8,column=2)

top.destroy()



def ca():

top1.destroy()

return

ok = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="OK", width=6, command=ca).grid(row=9,column=1)

def cancel():

top.destroy()

return

submit = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Submit", width=6, command=submit).grid(row=8,column=0)

cancel = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Cancel", width=6, command=cancel).grid(row=8,column=1)

**Cancel Ticket**

def cancel():

canc=Toplevel() #canc is the cancelation detail window

canc.title('Cancel window')

canc.geometry('600x420+300+150')

canc.resizable(0,0)

f1= Frame(canc, width =700,height=350, bd=8, relief='raise')

f1.pack(side=TOP)

f2a = Frame(f1, width =700,height=350, bd=6, relief='raise')

f2a.pack(side=LEFT)

bottomLeft1 = Frame(f2a, width=450, height=450, bd=14,relief='raise')

bottomLeft1.pack(side=LEFT)

Name = StringVar()

tno = int()

Class = StringVar()

Source = StringVar()

Destination = StringVar()

Name = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Name', bd=16, anchor='w')

Name.grid(row=0,column=0)

Name = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='Name', bd=10, width=18,

bg='#ffffff', justify='right')

Name.grid(row=0,column=1)

tno = Label(bottomLeft1, font=('arial', 24, 'bold'), text='Ticket Number', bd=16, anchor='w')

tno.grid(row=1,column=0)

tno = Entry(bottomLeft1, font=('arial', 16, 'bold'), textvariable='tno', bd=10, width=18, bg='#ffffff',

justify='right')

tno.grid(row=1,column=1)

lblClass = Label(bottomLeft1,font=('arial',20,'bold'), text='Class', bd=4,anchor='w')

lblClass.grid(row=2,column=0)

cboClass =ttk.Combobox(bottomLeft1,textvariable=Class, state='readonly', font=('arial',20,'bold'),

width=10)

cboClass['value']=('','Economy','Bussiness')

cboClass.current(0)

cboClass.grid(row=2,column=1)

lblSource = Label(bottomLeft1,font=('arial',20,'bold'), text='Source', bd=4,anchor='w')

lblSource.grid(row=3,column=0)

cboSource =ttk.Combobox(bottomLeft1,textvariable=Source, state='readonly',

font=('arial',20,'bold'), width=10)

cboSource['value']=('','Delhi','Mumbai','Kolkata','Chennai')

cboSource.current(0)

cboSource.grid(row=3,column=1)

lblDestination = Label(bottomLeft1,font=('arial',20,'bold'), text='Destination', bd=4,anchor='w')

lblDestination.grid(row=4,column=0)

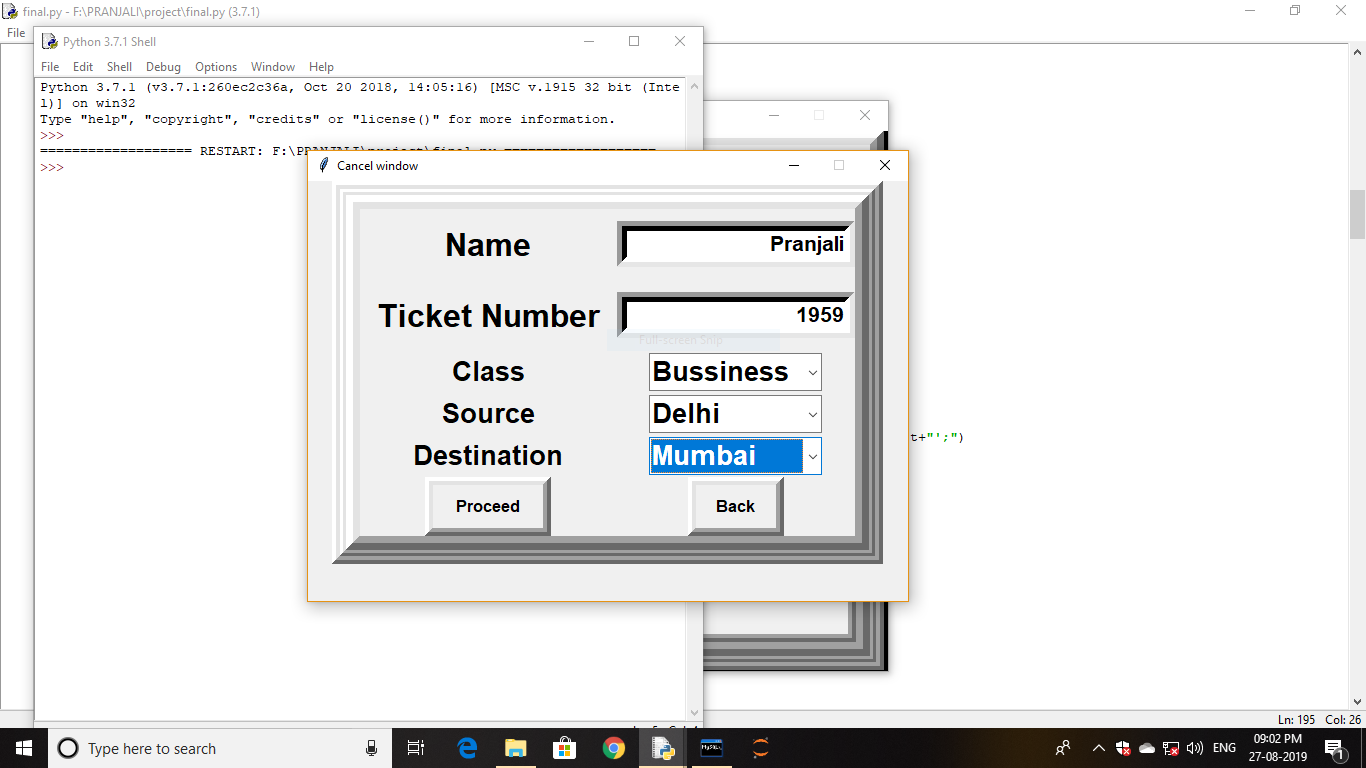
cboDestination =ttk.Combobox(bottomLeft1,textvariable=Destination, state='readonly',

font=('arial',20,'bold'), width=10)

cboDestination['value']=('','Delhi','Mumbai','Kolkata','Chennai')

cboDestination.current(0)

cboDestination.grid(row=4,column=1)



def ct():

canc2=Toplevel() #canc1 is the cancelation window

canc2.title('Cancel')

canc2.geometry('420x180+430+230')

canc2.resizable(0,0)

f1= Frame(canc2, width =40,height=20, bd=8, relief='raise')

f1.pack()

name=Name.get()

if (name==''):

messagebox.showerror('Null field','Name cannot be empty')

canc1.destroy()

return

Tno=tno.get()

if (Tno.isdigit()==False):

messagebox.showerror('Null field','Ticket Number cannot be empty')

canc1.destroy()

return

clas=Class.get()

if (clas==''):

messagebox.showerror('Null field','Class cannot be empty')

canc1.destroy()

return

source=Source.get()

if (source==''):

messagebox.showerror('Null field','Source cannot be empty')

canc1.destroy()

return

dest=Destination.get()

if (dest==''):

messagebox.showerror('Null field','Destination cannot be empty')

canc1.destroy()

return

sql1=mycursor.execute("select "+clas+" from flightsch where Source = '"+source+"' and

Destination = '"+dest+"';")

A=mycursor.fetchone()

for x in A:

a=x

lblca = Label(f1, font=('arial', 15), text="Amount paid :", bd=16)

lblca.grid(row=0,column=0)

lblca1 = Label(f1, font=('arial', 15), text=a, bd=16, anchor='nw')

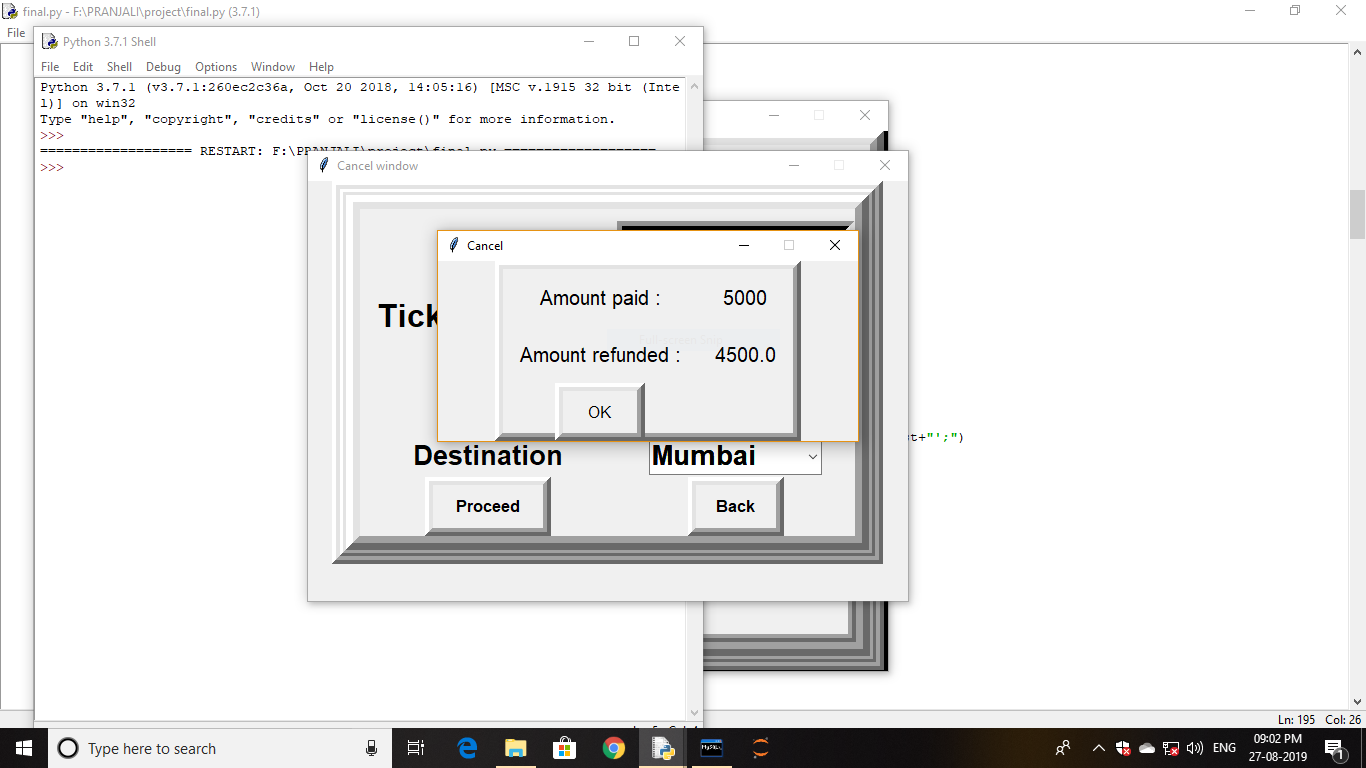
lblca1.grid(row=0,column=2)

lblca2 = Label(f1, font=('arial', 15), text="Amount refunded :", bd=16)

lblca2.grid(row=1,column=0)

lblca3 = Label(f1, font=('arial', 15), text=((90/100)\*a), bd=16)

lblca3.grid(row=1,column=2)



def o1():

canc1=Toplevel() #canc1 is the cancelation window

canc1.title('Cancel')

canc1.geometry('530x220+400+200')

canc1.resizable(0,0)

f1= Frame(canc1, width =40,height=20, bd=8, relief='raise')

f1.pack()

f2a = Frame(f1, width =40,height=20, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=100, bd=14,relief='raise')

bottomLeft1.pack()

sql=mycursor.execute("delete from pdata where Name='{}' and

Ticket\_Number={}".format(name,Tno))

lblcancel = Label(bottomLeft1, font=('arial', 15), text="Your ticket has been cancelled", bd=16,

anchor='nw')

lblcancel.grid(row=0,column=0)

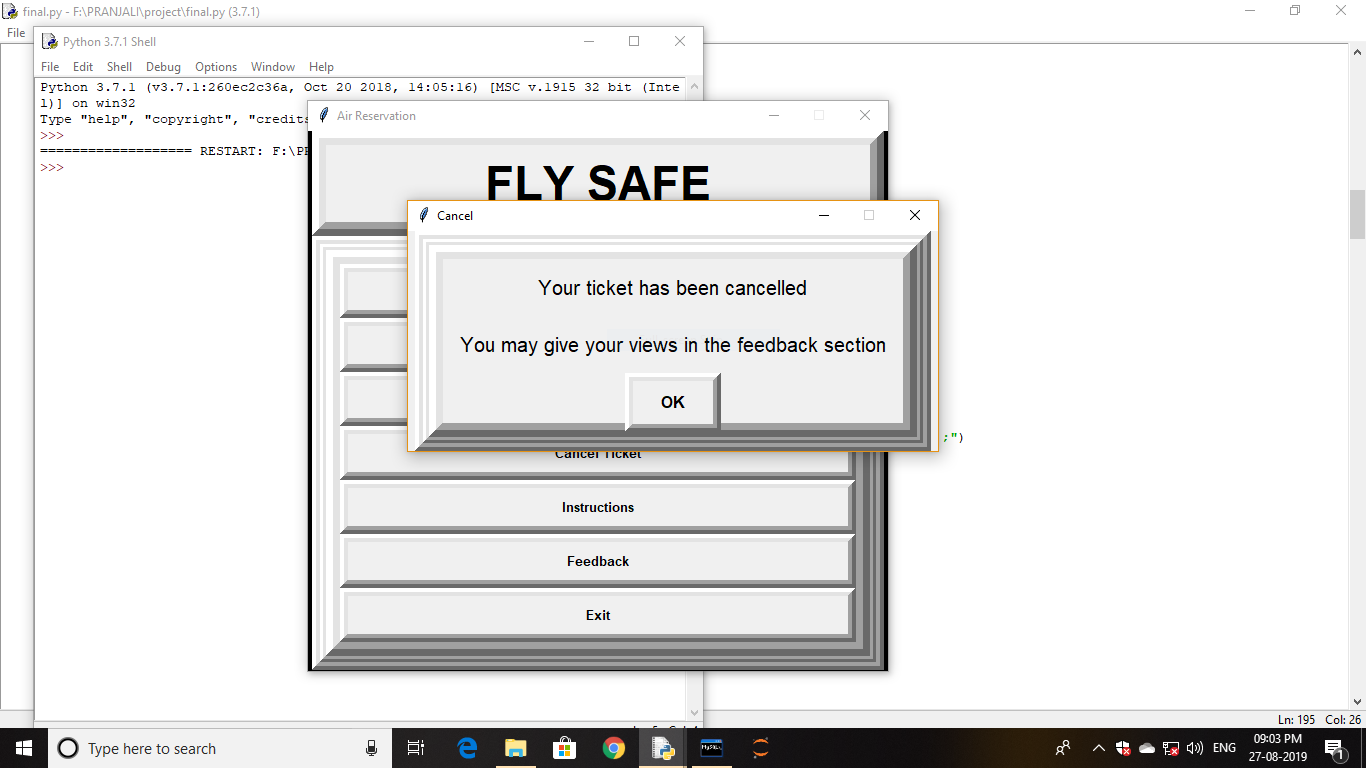
lblcf = Label(bottomLeft1, font=('arial', 15), text="You may give your views in the feedback

section", bd=16, anchor='nw')

lblcf.grid(row=1,column=0)

canc2.destroy()

canc.destroy()



def ca():

canc1.destroy()

return

ok2 = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="OK", width=6, command=ca).grid(row=2)

mydb.commit()

ok1 = Button(f1, padx=8, pady=8, bd=8, fg='black', font=('arial',12),\

text="OK", width=6, command=o1).grid(row=2)

def back():

canc.destroy()

return

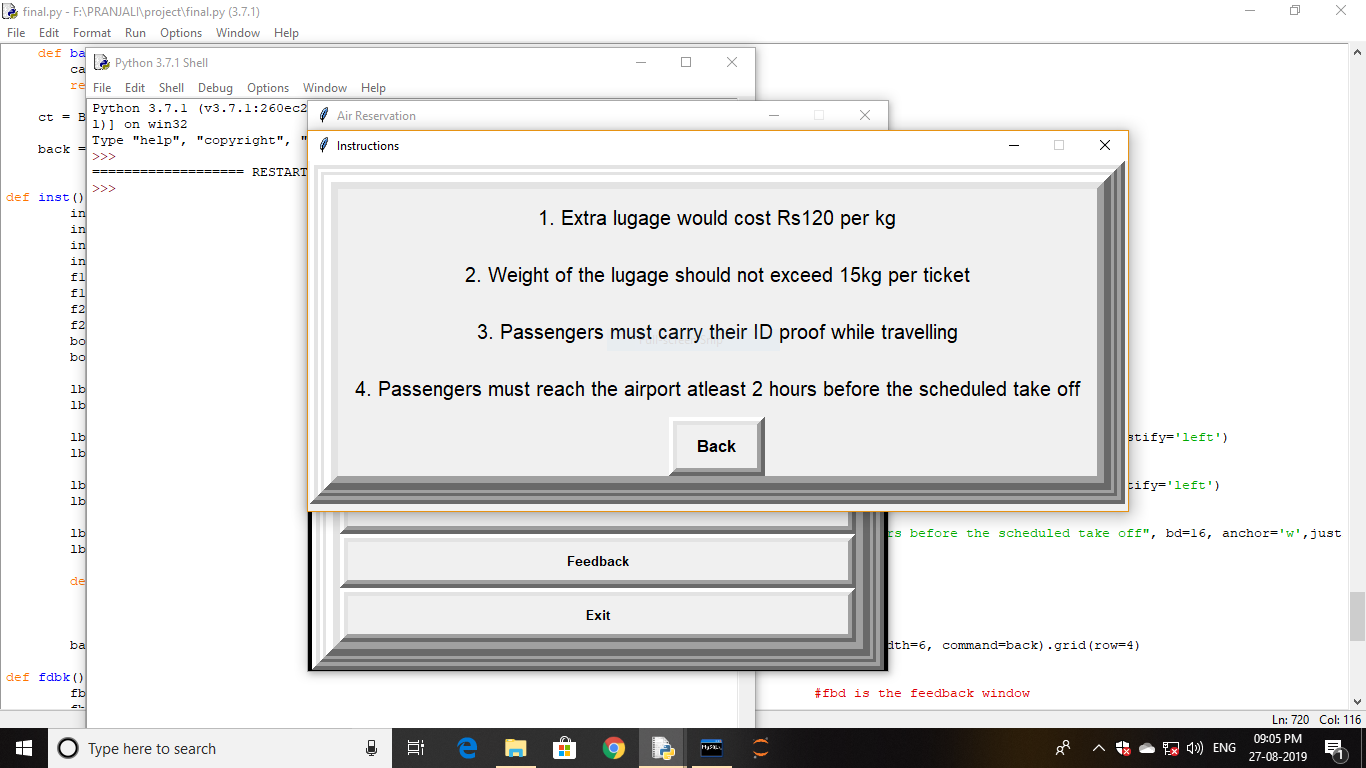
ct = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Proceed", width=9, command=ct).grid(row=5,column=0)

back = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Back", width=6, command=back).grid(row=5,column=1)

**Instructions**



def inst():

ins=Toplevel() #ins is the instruction window

ins.title('Instructions')

ins.geometry('820x350+200+160')

ins.resizable(0,0)

f1= Frame(ins, width =40,height=20, bd=8, relief='raise')

f1.pack()

f2a = Frame(f1, width =40,height=20, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=100, bd=14,relief='raise')

bottomLeft1.pack()

lbl1 = Label(bottomLeft1, font=('arial', 15), text="1. Extra lugage would cost Rs120 per kg",

bd=16, anchor='w')

lbl1.grid(row=0,column=0)

lbl2 = Label(bottomLeft1, font=('arial', 15), text="2. Weight of the lugage should not exceed

15kg per ticket", bd=16, anchor='w',justify='left')

lbl2.grid(row=1,column=0)

lbl3 = Label(bottomLeft1, font=('arial', 15), text="3. Passengers must carry their ID proof while

travelling", bd=16, anchor='w',justify='left')

lbl3.grid(row=2,column=0)

lbl4 = Label(bottomLeft1, font=('arial', 15), text="4. Passengers must reach the airport atleast 2

hours before the scheduled take off", bd=16, anchor='w',justify='left')

lbl4.grid(row=3,column=0)

def back():

ins.destroy()

return

back = Button(bottomLeft1, padx=8, pady=8, bd=8, fg='black', font=('arial',12,'bold'),

text="Back", width=6, command=back).grid(row=4)

**Feedback**

def fdbk():

fbd=Toplevel() #fbd is the feedback window

fbd.title('Feedback')

fbd.geometry('550x250+300+120')

fbd.resizable(0,0)

f1= Frame(fbd, width =140,height=600, bd=8, relief='raise')

f1.pack()

f2a = Frame(f1, width =140,height=600, bd=6, relief='raise')

f2a.pack()

bottomLeft1 = Frame(f2a, width=200, height=2000, bd=14,relief='raise')

bottomLeft1.pack()

fdk=StringVar()

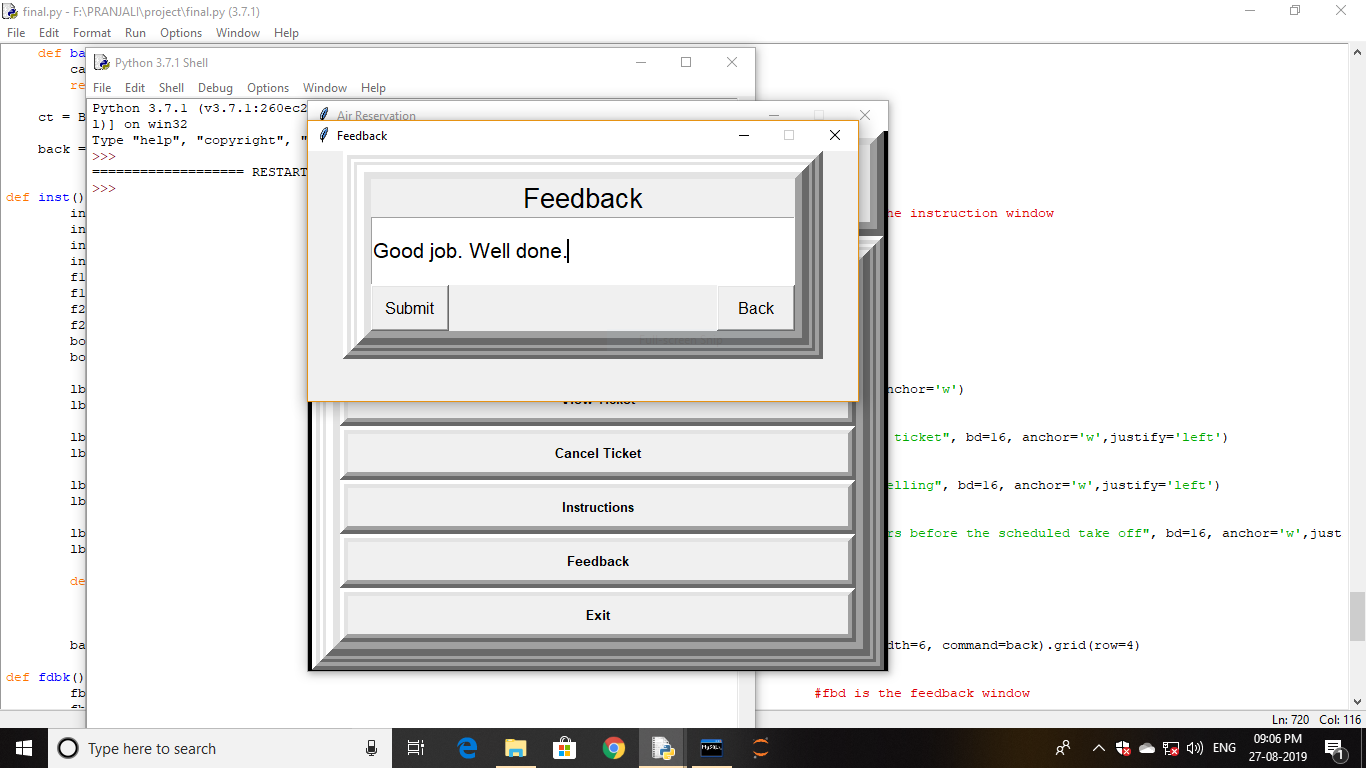
label = Label(bottomLeft1, font=('arial', 20), text='Feedback')

label.grid(row=0,column=1,sticky='s')

fd = StringVar()

fd = Entry(bottomLeft1, font=('arial', 16), textvariable = fdk, width = 35)

fd.grid(row=1,column=1,sticky='s',rowspan=3,ipady=20)



def submit():

fd=fdk.get()

if (fd==""):

fbd.destroy()

sql=mycursor.execute("insert into fdbk(Feedback)values('{}')".format(fd))

mydb.commit()

fbd.destroy()

def back():

fbd.destroy()

return

submit = Button(bottomLeft1, padx=8, pady=8, fg='black', font=('arial',12),text="Submit",

width=6, command=submit).grid(row=5,column=1,sticky='w')

back = Button(bottomLeft1, padx=8, pady=8, fg='black', font=('arial',12), text="Back", width=6,

command=back).grid(row=5,column=1,sticky='e')

**Exit**

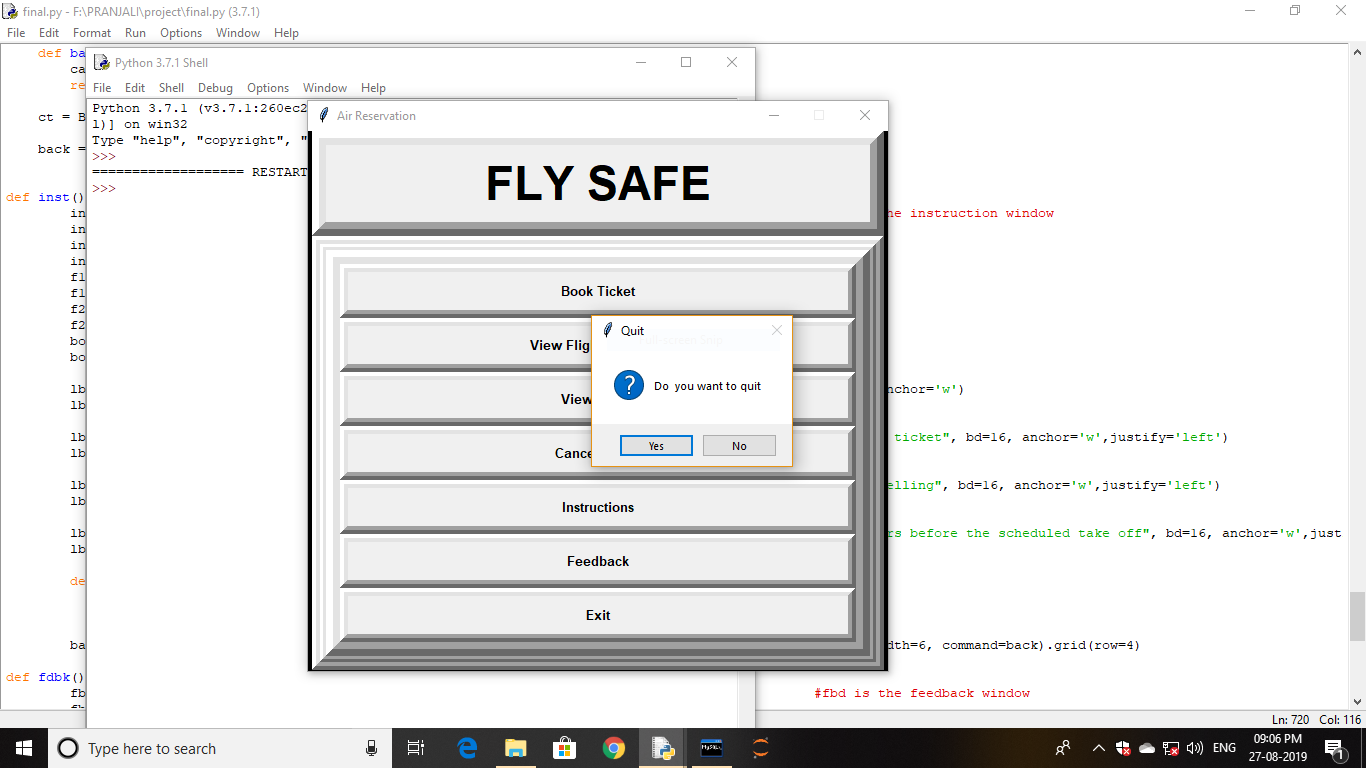
def iExit():

qExit = messagebox.askyesno("Quit","Do you want to quit")

if qExit > 0:

root.destroy()

return



**#==================================Buttons=====================================**

f2a = Frame(f1, width =1000,height=500, bd=6, relief='raise')

f2a.pack(side=BOTTOM)

bottomLeft2 = Frame(f2a, width=1000, height=700, bd=14,relief='raise')

bottomLeft2.pack(side=RIGHT)

bookbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="Book Ticket", width=60, command=booking).grid(row=0,column=0)

schedulebtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="View Flight Schedule", width=60, command=schedule).grid(row=1,column=0)

detailbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="View Ticket", width=60, command=details).grid(row=2,column=0)

cancelbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="Cancel Ticket", width=60, command=cancel).grid(row=3,column=0)

geninstbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="Instructions", width=60, command=inst).grid(row=4,column=0)

fedbkbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="Feedback", width=60, command=fdbk).grid(row=5,column=0)

exitbtn = Button(bottomLeft2, padx=8, pady=8, bd=8, fg='black', font=('arial',10,'bold'),

text="Exit", width=60, command=iExit).grid(row=6,column=0)

**Bibliography**

**Reference Books:**

* Informatics Practices book for class XI
* Informatics Practices book for class XII

**Reference Websites:**

* stackoverflow.com
* [www.pythonlake.com](http://www.pythonlake.com)
* [www.programiz.com](http://www.programiz.com)
* [www.effbot.org](http://www.effbot.org)
* [www.youtube.com](http://www.youtube.com)