PROJECT REPORT ON

AYUTAN FLIGHT BOOKING SYSTEM

BY :

AYUSH PADHY

STUDENT, VIT VELLORE

rocking_ayush04 ayushpadhy1309@gmail.com +91 9327450873

TABLE OF CONTENTS [T O C]

AYUTAN FLIGHT BOOKING SYSTEM

INTRODUCTION

Ayutan Flight Booking System is a python based project. We have developed A GUI based Flight Booking system using Python 3.8, MYSQL and CSV Files.

Our flight booking system provided the user a platform to easily book flight. It is designed with a User Friendly GUI so that the user can book the flights easily.

We have used the following Modules in our project:

- Tkinter
- Time
- Random
- Mysql.connector
- Datetime
- Calender
- CSV

OBJECTIVES OF THE PROJECT

The objective of this project is to:

- Let the user select and book a flight as per his preference.
- Providing the user a platform to easily book flight.
- Give Various choices of flights for the user to choose from.
- Let user book flights for more than 1 person.
- Option of going back.
- Informs the user if any wrong detail entered.

USE OF DIFFERENT MODULES IN THE PROJECT:

The project is primarly based on GUI Interface on pyhton.

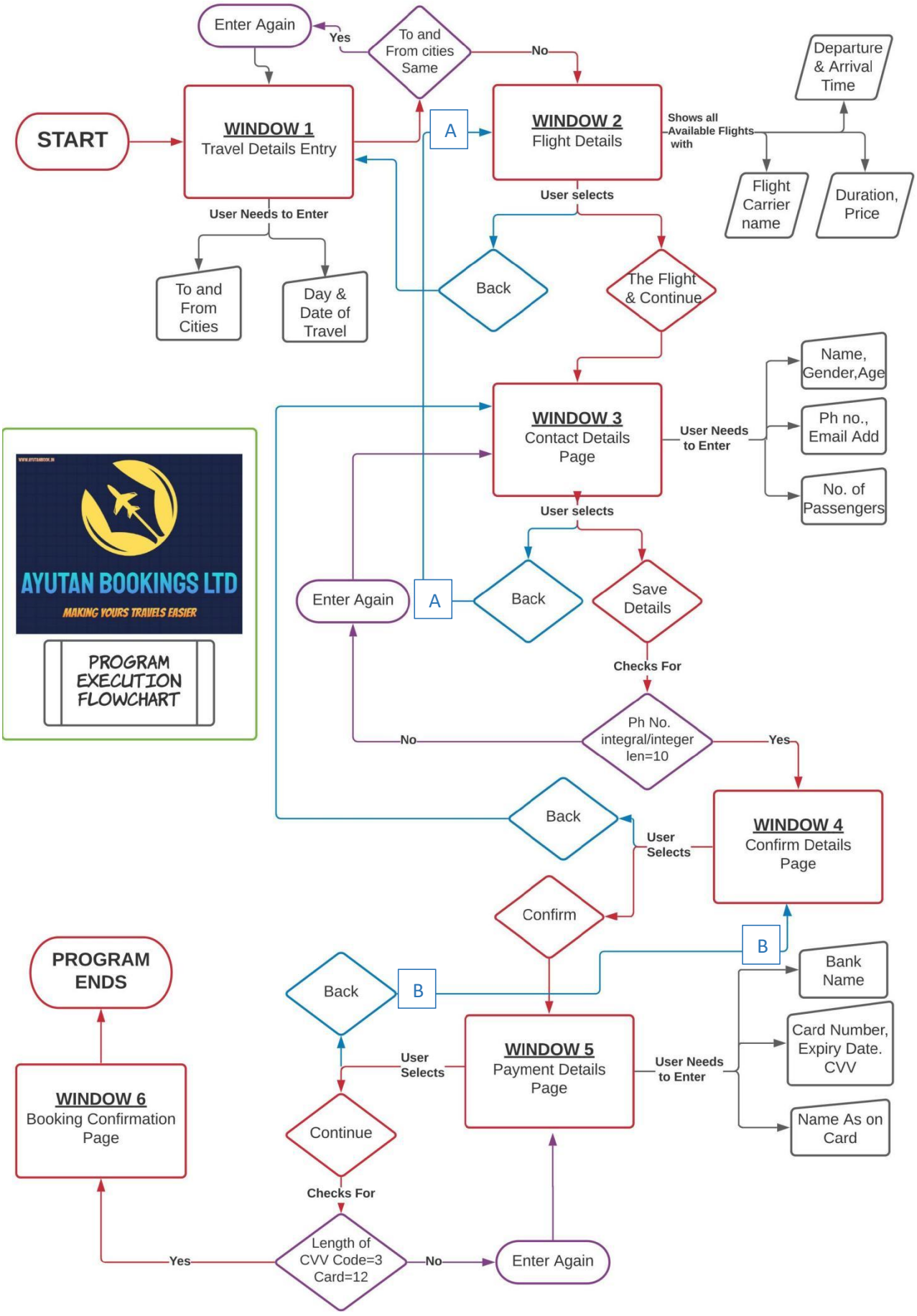
- **Tkinter Module** has been used to create a GUI interface for the project to make the program more user friendly.
- **DateTime & Calender Module** has been used to get the Date and Day Details of the time the user wants to travel.
- A Huge Database of Flight Details has been created using **MYSQL connectivity with Python**.
- **Random module** has been used to showcase 6-10 flights between the selected departure and arrival location by the user from the Database.
- **Time Module** has been used to facilitate the progress bar use.
- **CSV Module** has been used to create an excel sheet & input value into excel sheet of the all important details of the booking.

MORE ABOUT THE PROJECT

This project has many functions which make it interactive and more realistic to flight reservation sites like makemytrip. The program does the following functions:

- Arrival and Destination location selector along with date.
- Displaying numerous flights for the user to select from for his journey.
- Contact details entry page
- Confirm Details Page before proceeding forward to payments.
- Accepting Payments through various leading banks of India through a secure and fast process.
- If user feel the entered any wrong information, they can always go back to the previous page.
- Checks for any error in input of information and displays error if any abnormalities detected.
- Gives a detailed Excel Sheet about every booking made from the AYUTAN Bookings program.

FLOW CHART


Enter Again
Yes
To and From cities Same
No
START
WINDOW 1
Travel Details Entry
A
WINDOW 2
Flight Details
Shows all Available Flights with
Departure & Arrival Time
Flight Carrier name
Duration, Price
User Needs to Enter
To and From Cities
Day & Date of Travel
User selects
Back
The Flight & Continue
WINDOW 3
Contact Details Page
User Needs to Enter
Name, Gender,Age
Ph no., Email Add
No. of Passengers
User selects
Enter Again
A
Back
Save Details
Checks For
Ph No. integral/integer len=10
No
Yes
Back
User Selects
WINDOW 4
Confirm Details Page
Confirm
B
Bank Name
Back
B
WINDOW 5
Payment Details Page
User Selects
User Needs to Enter
Card Number, Expiry Date. CVV
Name As on Card
Continue
Checks For
PROGRAM ENDS
WINDOW 6
Booking Confirmation Page
Yes
Length of CVV Code=3 Card=12
No
Enter Again
AYUTAN BOOKINGS LTD
MAKING YOURS TRAVELS EASIER
PROGRAM EXECUTION FLOWCHART

SOURCE CODE

```
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import time
import random
import mysql.connector
import csv
import datetime
import calendar
import os
import sys
try:
    import docx
except:
    os.system("pip install python_docx")
    import docx


mydb = mysql.connector.connect(host="localhost",user="root", passwd="ayush123")


mycursor = mydb.cursor()


def First_Page():




    root=Tk()
    root.title("Enter Details")
    root.geometry("1000x600")


    def findDay(date):
        born = datetime.datetime.strptime(date, '%d %m %Y').weekday()
        return (calendar.day_name[born])
```

```
    def Submit():
        global From
        From=fromenter.get()
        global To
        To=toenter.get()
        l1=[]
        for x in range (0,31):
             l1.append(x)
        if int(dd.get())==0 or int(dd.get()) not in l1 or mm.get() not in Months or int(yy.get())
not in [2020,2021,2023,2022,2024]:

            l=Label(root, text="INVALID DATE")
            l.grid(row=4, column=1)
            return
        global Date
        Date= str(dd.get()) + " / " + str((Months.index(mm.get()))+1) + " / "+ str(yy.get())
        global day
        day=findDay(str(dd.get()) + " " + str((Months.index(mm.get()))+1) + " "+ str(yy.get()))

        if To==From or To not in locations or From not in locations:
           l=Label(root, text="INVALID LOCATIONS")
           l.grid(row=4, column=1)
           return
        else:


            root.destroy()


            wait = Tk()
            wait.title("Searching.......")
            wait.geometry("1000x100")


            Label(wait,text="Searching out the Best Flights for your Travel...",font="Bold 20").pa
ck()
```

```
            progress = ttk.Progressbar(wait, orient = HORIZONTAL,length = 100, mode = 'determinate
')


            progress.place(relx =0.5, rely=0.5, anchor=CENTER, relheight=0.25, relwidth=0.85)

            import time

            progress['value'] = 20

            wait.update()

            time.sleep(0.75)


            progress['value'] = 40

            wait.update()

            time.sleep(0.75)


            progress['value'] = 60

            wait.update()

            time.sleep(0.75)


            progress['value'] = 80

            wait.update()

            time.sleep(0.75)

            progress['value'] = 100


            wait.destroy()


            wait.mainloop()




            Flight_Details()

        # Next=Button(root, text="Search Flights", bd='5', command=lambda: [root.destroy(),Flight_
Details()])

        # Next.grid(row=7,column=1)
```

```
    from_=Label(root,text= "From: " , font="50")

    from_.grid(row= 0, column= 0, sticky= W, pady=2)

    n = StringVar()

    fromenter=ttk.Combobox(root, width = 70, textvariable = n)

    fromenter['values'] = ('Ahmedabad','New Delhi', 'Mumbai', 'Vadodara', 'Bangaluru', 'Hyderabad',
 'Bhubaneswar',

                           'Chennai','Kolkata')

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




    to_=Label(root, text="To: ", font="50")

    to_.grid(row= 1, column= 0, sticky= W, pady=2)

    n = StringVar()

    toenter= ttk.Combobox(root, width = 70, textvariable = n)

    toenter['values'] = ('Ahmedabad','New Delhi', 'Mumbai', 'Vadodara', 'Bangaluru', 'Hyderabad',
'Bhubaneswar',

                           'Chennai','Kolkata')

    locations=['Ahmedabad','New Delhi', 'Mumbai', 'Vadodara', 'Bangaluru', 'Hyderabad', 'Bhubanesw
ar',

                           'Chennai','Kolkata']

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









    Date_=Label(root, text="Date of Journey: ", font="50")

    Date_.grid(row= 2, column= 0, sticky= W, pady=2)

    dd=Spinbox(root, from_= 0, to = 31, width=5)

    dd.grid(row=2, column=1, sticky=W)

    n = StringVar()


    mm=ttk.Combobox(root, width = 27, textvariable = n)

    Months=[' January',

                           ' February',

                           ' March',

                           ' April',
```

```
                            ' May',
                            ' June',
                            ' July',
                            ' August',
                            ' September',
                            ' October',
                            ' November',
                            ' December']
mm['values'] = (' January',
                            ' February',
                            ' March',
                            ' April',
                            ' May',
                            ' June',
                            ' July',
                            ' August',
                            ' September',
                            ' October',
                            ' November',
                            ' December')



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



yy=Spinbox(root, from_= 2021, to = 2023, width=10)
yy.grid(row=2, column=1, sticky=E)



submit=Button(root, text="Save Details", command=Submit)
```

```
    submit.grid(row=3,column=1)









    root.mainloop()



def Flight_Details():



    def back():
        root1.destroy()
        First_Page()
    def Submit1():
        v=V.get()
        global selected
        selected=DispList[v]


        root1.destroy()


        Details_Page()


    root1=Tk()
    root1.title("Select FLight")
    root1.geometry('2000x2000')







    try:
        mycursor.execute("USE AYUTANBOOK")
    except:
```

```
        mycursor.execute("CREATE DATABASE AYUTANBOOK")


        mycursor.execute("USE AYUTANBOOK")


        mycursor.execute("CREATE TABLE FlightDetails(Flight char(20),Departure_Time char(10),Arriv
al_Time char(10),Duration char (20),Flight_Type char(80), Price Numeric(10))")


        sql = "INSERT INTO FlightDetails (Flight, Departure_Time,Arrival_Time, Duration, Flight_Ty
pe, Price) VALUES (%s, %s,%s,%s, %s,%s)"

        val=[('Indigo-6E365','9:40 am','11.40 am','2hrs','Non-Stop',4500),

              ('SpiceJet-SJ56','6:00 am','7:50 am','1 hrs 50mins','Non-Stop',4300),

              ('Vistara-VTAQ8','7:40 am','10:00 am','2 hrs 20mins','Non-Stop',4500),

              ('AirIndia-AI789','11.30 am','1:20 pm','1 hrs 50mins','Non-Stop',5600),

              ('GoAir-GA557','11:45 am','1:45 pm','2hrs','Non-Stop',4600),

              ('Indigo-6E890','3:40 pm','5:40 pm','2 hrs','Non-Stop',4900),

              ('AirAsia-AA48K','4:20 pm','6.00 pm','1 hrs 40mins', 'Non-Stop',4900),

              ('Indigo-6E156','4:50 pm','7:00 pm','2 hrs 10mins','Non-Stop',5100),

              ('SpiceJet-SJ90','6:00 pm','8:30 pm','2 hrs 30mins','Non-Stop',4100),

              ('Vistara-VTAZ7','7:00 pm','8:40 pm','1 hrs 40mins','Non-Stop',4900),

              ('GoAir-GA698','7.45 pm','9:50 pm','2 hrs 5mins','Non-Stop',4300),

              ('AirIndia-AI340','8:50 pm','10.35pm','1 hrs 45mins','Non-Stop',5400),

              ('Indigo-6E691','10:00 pm','11:50 pm', '1 hrs 50mins','Non-Stop',4400),

              ('AirAsia-AA78J','8:30 am','12:30 pm','4hrs','1 hr 30 mins Stop at Bhopal', 3900),

              ('Indigo-6E410','11:30 am','4:50 pm','5 hrs 20mins','1 hr 50 mins Stop at Raipur', 4
100),

              ('GoAir-GA689','1:30 pm','5:20 pm','3 hrs 50mins','1 hr Stop a Pune', 4200)]


        mycursor.executemany(sql,val)


        mydb.commit()


    number=random.randint(7,10)



    mycursor.execute("SELECT * FROM FlightDetails")
```

```
filedata = mycursor.fetchall()


L=[]
for x in filedata:
    L.append(x)
Len=len(L)


DispList=[]
m=[]
for i in range(Len):
    m.append(i)


for i in range(number):
    a=random.choice(m)
    m.remove(a)
    DispList.append(L[a])



row_no=number
column_no= len(DispList[0])


V=IntVar()
frame1=Frame(root1, height=130)
frame1.pack(side= TOP, fill=X)
frame2=Frame(root1)
frame2.pack(side= TOP, fill=X)
frame3=Frame(root1)
frame3.pack(side= TOP, fill=X)



l1=Label(frame1, text=From+"  \u2192", font="30")
l1.place(x=5,y=10)
l2=Label(frame1, text=Date,font="30")
l2.place(x=5, y=70)
```

```
    l3=Label(frame1, text=To,font="30")
    l3.place(x=160,y=10)
    l4=Label(frame1, text=day,font="30")
    l4.place(x=150,y=70)



    for row in range(row_no):
        for xyz in range(column_no+1):
            frame2.columnconfigure(xyz, weight=1)
        if row==0:
                label1 = Label(frame2, text="FLIGHT",font=('Arial',14), borderwidth=2,relief="soli
d")
                label1.grid(row=row, column=0, sticky="nsew", padx=1, pady=1)
                label2 = Label(frame2, text="DEPARTURE_TIME",font=('Arial',14), borderwidth=1,reli
ef="solid")
                label2.grid(row=row, column=1, sticky="nsew", padx=1, pady=1)
                label3 = Label(frame2, text="ARRIVAL_TIME",font=('Arial',14), borderwidth=1,relief
="solid")
                label3.grid(row=row, column=2, sticky="nsew", padx=1, pady=1)
                label4 = Label(frame2, text="DURATION",font=('Arial',14), borderwidth=1, relief="s
olid")
                label4.grid(row=row, column=3, sticky="nsew", padx=1, pady=1)
                label5 = Label(frame2, text="FLIGHT_TYPE",font=('Arial',14), borderwidth=1,relief=
"solid")
                label5.grid(row=row, column=4, sticky="nsew", padx=1, pady=1)
                label6 = Label(frame2, text="PRICE",font=('Arial',14), borderwidth=1,relief="solid
")
                label6.grid(row=row, column=5, sticky="nsew", padx=1, pady=1)
                label7 = Label(frame2, text="SELECT",font=('Arial',14), borderwidth=1,relief="soli
d")
                label7.grid(row=row, column=6, sticky="nsew", padx=1, pady=1)



        else:
            for column in range(column_no):
                disp=DispList[row][column]
                label=Label(frame2,text=disp, borderwidth=1,relief="solid")
                label.grid(row=row,column=column,sticky="nsew",padx=1,pady=1)
```

```
            radioButton=Radiobutton(frame2,text="Book Now", variable=V,value=row, borderwidth=1,re
lief="solid")
            radioButton.grid(row=row,column=6,sticky="nsew")



    submit1=Button(frame3, text="Submit", bd='5', command=Submit1)
    submit1.pack(anchor= CENTER)
    Back_button=Button(frame3,text="Back", command=back, bd='5')
    Back_button.pack(anchor=CENTER)

    root1.mainloop()



def Details_Page():
    root=Tk()
    root.title("Enter Details")
    root.geometry("1000x600")

    def back():
        root.destroy()
        Flight_Details()

    def Submit2():

        try:
            global age
            age=age_var.get()
            global phone
            phone=phone_var.get()
        except:
            try:
                age=age_var.get()
            except:
                l=Label(root, text="INVALID AGE")
                l.grid(row=7, column=1)
```

```
            try:
                phone=phone_var.get()
            except:
                l=Label(root, text="INVALID PHONE NUMBER")
                l.grid(row=8, column=1)
        else:
            if len(str(phone))!=10:
                l=Label(root, text="INVALID PHONE NUMBER")
                l.grid(row=8, column=1)
            else:
                global name
                name=name_var.get()
                global gender
                gender=n.get()
                global email
                email=email_var.get()
                global no
                no=no_box.get()
                root.destroy()
                CONFIRMATION_PAGE()


name_var=StringVar()
name_label =Label(root, text = 'Name: ',font="50")
name_entry =Entry(root, textvariable = name_var, width=50)
name_label.grid(row=0,column=0,sticky=W, pady=2)
name_entry.grid(row=0,column=1,sticky=W, pady=2)


gender_=Label(root, text="Gender: ",font="50")
gender_.grid(row= 1, column= 0, sticky= W, pady=2)
n = StringVar()
genderenter= ttk.Combobox(root, width =10, textvariable = n)
genderenter['values'] = ("Male","Female")
genderenter.grid(row= 1, column= 1,sticky=W, pady=2)
```

```
    age_var=IntVar()
    age_label =Label(root, text = 'Age: ',font="50")
    age_entry =Entry(root, textvariable = age_var,width=5)
    age_label.grid(row=2,column=0,sticky=W, pady=2)
    age_entry.grid(row=2,column=1,sticky=W, pady=2)

    phone_var=IntVar()
    phone_label =Label(root, text = 'Phone Number: ',font="50")
    phone_entry =Entry(root, textvariable = phone_var)
    phone_label2 =Label(root, text = '+91')
    phone_label.grid(row=3,column=0,sticky=W, pady=2)
    phone_label2.grid(row=3,column=0,sticky=E, pady=2)
    phone_entry.grid(row=3,column=1,sticky=W, pady=2)

    email_var=StringVar()
    email_label =Label(root, text = 'Email Add: ',font="50")
    email_entry =Entry(root, textvariable = email_var, width=50)
    email_label.grid(row=4,column=0,sticky=W, pady=2)
    email_entry.grid(row=4,column=1,sticky=W, pady=2)

    no_label=Label(root, text="No. of Passengers:     ",font="50")
    no_label.grid(row=5,column=0,sticky=W, pady=2)
    no_box=Spinbox(root, from_= 1, to = 10, width=10)
    no_box.grid(row=5, column=1,sticky=W, pady=2)

    submit2=Button(root, text="Save Details", command=Submit2)
    submit2.grid(row=6,column=1,sticky=W, padx=5)

    Back_button=Button(root,text="Back", command=back)
    Back_button.grid(row=6,column=0,sticky=E)

    root.mainloop()

def CONFIRMATION_PAGE():
```

```
    def Continue():
        root3.destroy()
        wait2 = Tk()
        wait2.title("Directing")
        wait2.geometry("1000x100")

        Label(wait2,text="Directing To Secure Payment Site......",font="Bold 20").pack()

        progress = ttk.Progressbar(wait2, orient = HORIZONTAL,length = 100, mode = 'determinate')

        progress.place(relx =0.5, rely=0.5, anchor=CENTER, relheight=0.25, relwidth=0.85)

        import time
        progress['value'] = 20
        wait2.update()
        time.sleep(0.75)

        progress['value'] = 40
        wait2.update()
        time.sleep(0.75)

        progress['value'] = 60
        wait2.update()
        time.sleep(0.75)

        progress['value'] = 80
        wait2.update()
        time.sleep(0.75)
        progress['value'] = 100

        wait2.destroy()

        wait2.mainloop()

        Payment_page()
```

```
def back():
    root3.destroy()
    Details_Page()


root3=Tk()
root3.title("Confirm Details")
root3.geometry('500x500')


l1=Label(root3,text="Boarding Location: ")
l1.grid(row=0,column=0, sticky=W,padx=2)
l2=Label(root3,text="Final Destination : ")
l2.grid(row=1,column=0, sticky=W,padx=2)
l3=Label(root3,text="Date of Travel: ")
l3.grid(row=2,column=0, sticky=W,padx=2)
l4=Label(root3,text="Day of Travel: ")
l4.grid(row=3,column=0, sticky=W,padx=2)
l5=Label(root3,text="Departure Time: ")
l5.grid(row=4,column=0, sticky=W,padx=2)
l6=Label(root3,text="Arrival Time: ")
l6.grid(row=5,column=0, sticky=W,padx=2)
l7=Label(root3,text="Duration: ")
l7.grid(row=6,column=0, sticky=W,padx=2)
l8=Label(root3,text="Flight Name-Id: ")
l8.grid(row=7,column=0, sticky=W,padx=2)
l9=Label(root3,text="Flight type: ")
l9.grid(row=8,column=0, sticky=W,padx=2)
l10=Label(root3,text="Name: ")
l10.grid(row=9,column=0, sticky=W,padx=2)
l11=Label(root3,text="Age: ")
l11.grid(row=10,column=0, sticky=W,padx=2)
l12=Label(root3,text="Phone No.: ")
l12.grid(row=11,column=0, sticky=W,padx=2)
l13=Label(root3,text="Email ID: ")
```

```
l13.grid(row=12,column=0, sticky=W,padx=2)

l14=Label(root3,text="Number of Passengers: ")

l14.grid(row=13,column=0, sticky=W,padx=2)

l15=Label(root3,text="Amount to be paid: ")

l15.grid(row=14,column=0, sticky=W,padx=2)


w1=Label(root3,text=From)

w1.grid(row=0,column=1, sticky=W,padx=2)

w2=Label(root3,text=To)

w2.grid(row=1,column=1, sticky=W,padx=2)

w3=Label(root3,text=Date)

w3.grid(row=2,column=1, sticky=W,padx=2)

w4=Label(root3,text=day)

w4.grid(row=3,column=1, sticky=W,padx=2)

w5=Label(root3,text=selected[1])

w5.grid(row=4,column=1, sticky=W,padx=2)

w6=Label(root3,text=selected[2])

w6.grid(row=5,column=1, sticky=W,padx=2)

w7=Label(root3,text=selected[3])

w7.grid(row=6,column=1, sticky=W,padx=2)

w8=Label(root3,text=selected[0])

w8.grid(row=7,column=1, sticky=W,padx=2)

w9=Label(root3,text=selected[4])

w9.grid(row=8,column=1, sticky=W,padx=2)

w10=Label(root3,text=name)

w10.grid(row=9,column=1, sticky=W,padx=2)

w11=Label(root3,text=str(age))

w11.grid(row=10,column=1, sticky=W,padx=2)

w12=Label(root3,text=str(phone))

w12.grid(row=11,column=1, sticky=W,padx=2)

w13=Label(root3,text=email)

w13.grid(row=12,column=1, sticky=W,padx=2)

w14=Label(root3,text=str(no))

w14.grid(row=13,column=1, sticky=W,padx=2)
```

```
    w15=Label(root3,text=str(int(selected[5])*int(no)))
    w15.grid(row=14,column=1, sticky=W,padx=2)

    Continue_button=Button(root3,text="Confirm", command=Continue)
    Continue_button.grid(row=15,column=1,sticky=W,padx=5)
    Back_button=Button(root3,text="Back", command=back)
    Back_button.grid(row=15,column=0,sticky=E)
    root3.mainloop()



def Payment_page():

    def bac():
        root4.destroy()
        CONFIRMATION_PAGE()
    def Sub():
        sel_bank=bank.get()
        if sel_bank not in bank_list:
            l=Label(root4, text=" INVALID BANK")
            l.grid(row=6, column=1)
            return
        F_name=n1.get()
        L_name=n2.get()
        expmonth=exp_mm.get()
        l1=[]
        for x in range(1,13):
            l1.append(x)
        try:
            expm=int(expmonth)
        except:
            l=Label(root4, text="INVALID MONTH")
            l.grid(row=6, column=1)
            return
```

```
        if int(expmonth) not in l1 :
            l=Label(root4, text="INVALID MONTH")
            l.grid(row=6, column=1)
            return
        l2=[]
        for x in range(2021,2041):
            l2.append(x)
        expyear=exp_yy.get()
        try:
            expy=int(expyear)
        except:
            l=Label(root4, text=" INVALID YEAR")
            l.grid(row=6, column=1)
            return
        if int(expyear) not in l2:
            l=Label(root4, text=" INVALID YEAR")
            l.grid(row=6, column=1)
            return
        try:
            cvvno=cvv.get()
        except:
            l=Label(root4, text="   INVALID CVV   ")
            l.grid(row=6, column=1)
        else:
            if len(cvvno)!=3:
                l=Label(root4, text="   INVALID CVV   ")
                l.grid(row=6, column=1)
                return
            root4.destroy()

            wait2 = Tk()
            wait2.title("Confirming Your Payment")
            wait2.geometry("1000x180")
```

```
            Label(wait2,text="Processing Payment......",font="Bold 20").pack()


            progress = ttk.Progressbar(wait2, orient = HORIZONTAL,length = 100, mode = 'determinat
e')


            progress.place(relx =0.5, rely=0.5, anchor=CENTER, relheight=0.15, relwidth=0.85)

            import time

            progress['value'] = 20

            wait2.update()

            time.sleep(0.75)


            progress['value'] = 40

            wait2.update()

            time.sleep(0.75)


            progress['value'] = 60

            wait2.update()

            time.sleep(0.75)


            progress['value'] = 80

            wait2.update()

            time.sleep(0.75)

            progress['value'] = 100


            wait2.destroy()


            wait2.mainloop()


            wait3 = Tk()

            wait3.title("Confirming Your Booking")

            wait3.geometry("1000x180")


            Label(wait3,text='''Payment Successful

Directing Back to Merchant Site......''',font="Bold 20").pack()
```

```
            progress = ttk.Progressbar(wait3, orient = HORIZONTAL,length = 100, mode = 'determinat
e')


            progress.place(relx =0.5, rely=0.5, anchor=CENTER, relheight=0.15, relwidth=0.85)

            import time

            progress['value'] = 20

            wait3.update()

            time.sleep(0.75)


            progress['value'] = 40

            wait3.update()

            time.sleep(0.75)


            progress['value'] = 60

            wait3.update()

            time.sleep(0.75)


            progress['value'] = 80

            wait3.update()

            time.sleep(0.75)

            progress['value'] = 100


            wait3.destroy()


            wait3.mainloop()


            length=0

            try:

                with open('.\\booking_record.csv','r',newline='') as file:

                    r=csv.reader(file)

                    for i in r:

                        length+=1

            except:
```

```
                pass
            with open('.\\booking_record.csv','a+',newline='') as file:
                w=csv.writer(file)
                if length==0:
                    w.writerow(['Name',"Age",'Phone_no','Email_ad',"Flight_Name","Boarding_Locatio
n",
"Destination","Dep_Date","Dep_Day","Dep_Time","Arrival_Time","Flight_Type","No_of_Passengers",
"Amount_paid"])

                w.writerow([name,age,phone,email,selected[0],From,To,Date,day,selected[1],
selected[2],selected[4],no,int(selected[5])*int(no)])



            Final_Page()


    root4=Tk()
    root4.title("Make Payment")
    root4.geometry('500x500')



    bank_n=Label(root4, text="Bank Name").grid(row=0,column=0,sticky=W,pady=2)
    n1_n=Label(root4, text="First Name").grid(row=1,column=0,sticky=W,pady=2)
    n2_n=Label(root4, text="Last Name").grid(row=2,column=0,sticky=W,pady=2)
    date_n=Label(root4, text="Expiry Month/Year(MM/YYYY)").grid(row=3,column=0,sticky=W,pady=2)
    cvv_n=Label(root4, text="CVV").grid(row=4,column=0,sticky=W,pady=2)

    bank_var=StringVar()
    bank=ttk.Combobox(root4, width = 27, textvariable = bank_var, text='Bank Name')
    bank['values']=['State Bank of India(SBI)','Kotak Bank','HDFC bank',
        'ICICI Bank','Bank of Baroda(BOB)','Indusland Bank','Yes Bank']
    bank_list=['State Bank of India(SBI)','Kotak Bank','HDFC bank',
        'ICICI Bank','Bank of Baroda(BOB)','Indusland Bank', 'Yes Bank']
    bank.grid(row=0,column=1,sticky=W,pady=2)

    n1_var=StringVar()
```

```
    n1=Entry(root4, textvariable= n1_var, width=50)
    n1.grid(row=1,column=1, sticky=W,pady=2)
    n2_var=StringVar()
    n2=Entry(root4, textvariable= n2_var, width=50)
    n2.grid(row=2,column=1, sticky=W,pady=2)

    exp_mm=Spinbox(root4, from_=0, to = 12, width=10)
    exp_mm.grid(row=3, column=1, sticky=W,pady=2)
    exp_yy=Spinbox(root4, from_=2021, to = 2040, width=10)
    exp_yy.grid(row=3, column=1, sticky=E, pady=2)

    cvv_var=IntVar()
    cvv=Entry(root4, textvariable=cvv_var, width=50)
    cvv.grid(row=4,column=1,sticky=W, pady=2)

    s_button=Button(root4,text="Submit", command=Sub)
    s_button.grid(row=5,column=1,sticky=W,padx=5,pady=2)
    b_button=Button(root4,text="Back", command=bac)
    b_button.grid(row=5,column=0,sticky=E)

    root4.mainloop()

def Home_page():
    def Book():
        ws.destroy()
        First_Page()
    def Exit():
        ws.destroy()
        sys.exit()
    ws = Tk()
    ws.title('Welcome')
    ws.geometry('1920x1080')
    bg =PhotoImage(file = ".\\Photo.pgm", master=ws)
    label1 = Label( ws, image = bg)
```

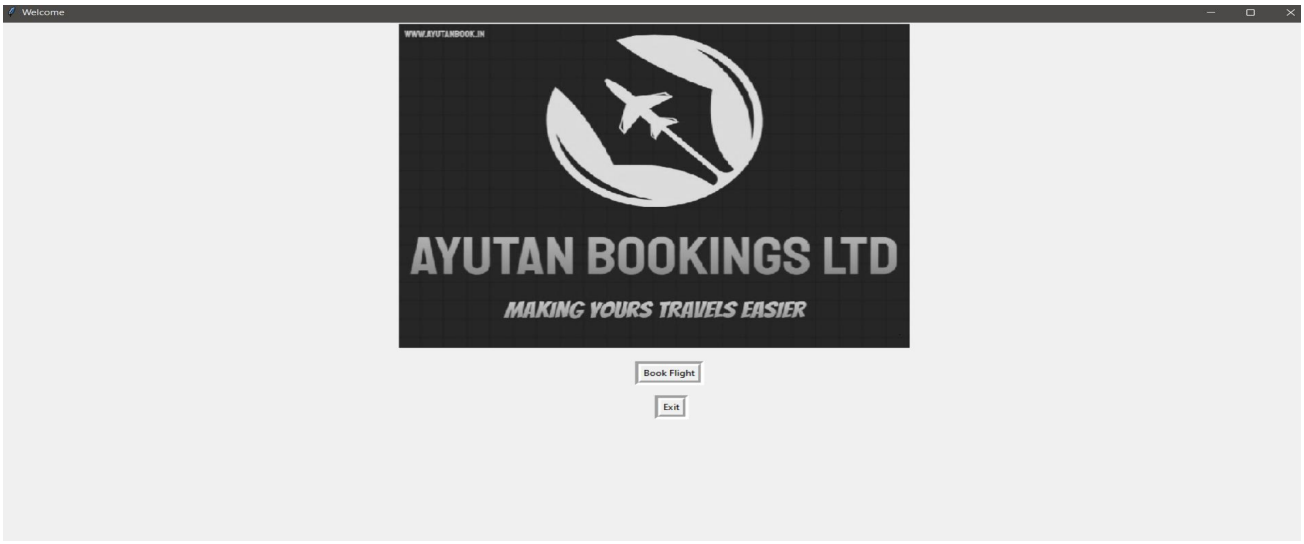
```
    label1.image=bg
    label1.place(relx = 0.5, anchor=N)
    book=Button(ws, text="Book Flight", command=Book,borderwidth = 7,relief="groove")
    book.place(relx=0.485,rely=0.6)
    b2=Button(ws, text="Exit", command=Exit,borderwidth = 7,relief="groove")
    b2.place(relx=0.5,rely=0.66)

    ws.mainloop()

def Final_Page():

    page=Tk()
    page.title("Booking Complete")
    page.geometry("500x700")

    def b1():
        page.destroy()
        Home_page()

    # def Print():
    #     doc1=docx.Document()
    #     doc1.add_heading

    frame1=Frame(page, height=150)
    frame1.pack(side= TOP, fill=X)
    frame2=Frame(page)
    frame2.pack(side= TOP, fill=X)
    frame3=Frame(page,height=200)
    frame3.pack(side= TOP, fill=X)

    label1=Label(frame1, text="Booking Confirmed!!!",font="Bold 20",borderwidth = 7,relief="groove
")
    label1.place(relx=0.5,rely=0.5, anchor=CENTER)
```

```
    l1=Label(frame2,text="Boarding Location: ")
    l1.grid(row=0,column=0, sticky=W,padx=2)
    l2=Label(frame2,text="Final Destination : ")
    l2.grid(row=1,column=0, sticky=W,padx=2)
    l3=Label(frame2,text="Date of Travel: ")
    l3.grid(row=2,column=0, sticky=W,padx=2)
    l4=Label(frame2,text="Day of Travel: ")
    l4.grid(row=3,column=0, sticky=W,padx=2)
    l5=Label(frame2,text="Departure Time: ")
    l5.grid(row=4,column=0, sticky=W,padx=2)
    l6=Label(frame2,text="Arrival Time: ")
    l6.grid(row=5,column=0, sticky=W,padx=2)
    l8=Label(frame2,text="Flight Name-Id: ")
    l8.grid(row=6,column=0, sticky=W,padx=2)

    w1=Label(frame2,text=From)
    w1.grid(row=0,column=1, sticky=W,padx=2)
    w2=Label(frame2,text=To)
    w2.grid(row=1,column=1, sticky=W,padx=2)
    w3=Label(frame2,text=Date)
    w3.grid(row=2,column=1, sticky=W,padx=2)
    w4=Label(frame2,text=day)
    w4.grid(row=3,column=1, sticky=W,padx=2)
    w5=Label(frame2,text=selected[1])
    w5.grid(row=4,column=1, sticky=W,padx=2)
    w6=Label(frame2,text=selected[2])
    w6.grid(row=5,column=1, sticky=W,padx=2)
    w8=Label(frame2,text=selected[0])
    w8.grid(row=6,column=1, sticky=W,padx=2)

    label2=Label(frame3, text='''Thanks For Using
AYUTAN Booking System''',font="Bold 20",borderwidth = 7,relief="ridge")
    label2.place(relx=0.5,rely=0.5, anchor=CENTER)
```

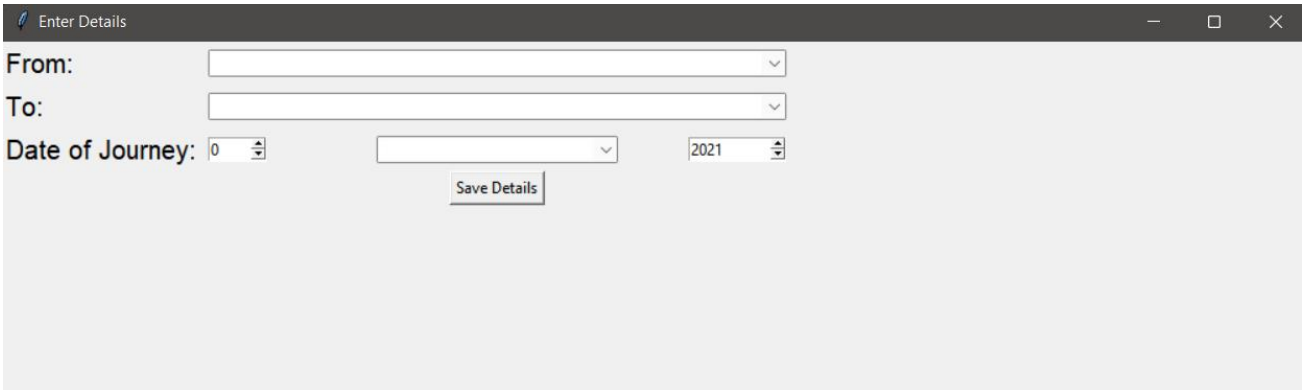
```
    b1=Button(frame3, text="Done", command=b1)
    b1.place(relx=0.5,rely=0.85, anchor=CENTER)
    # b2=Button(frame3, text="Print Ticket", command=Print )
    # b2.place(anchor=CENTRE)

    page.mainloop()

Home_page()
```

OUTPUT

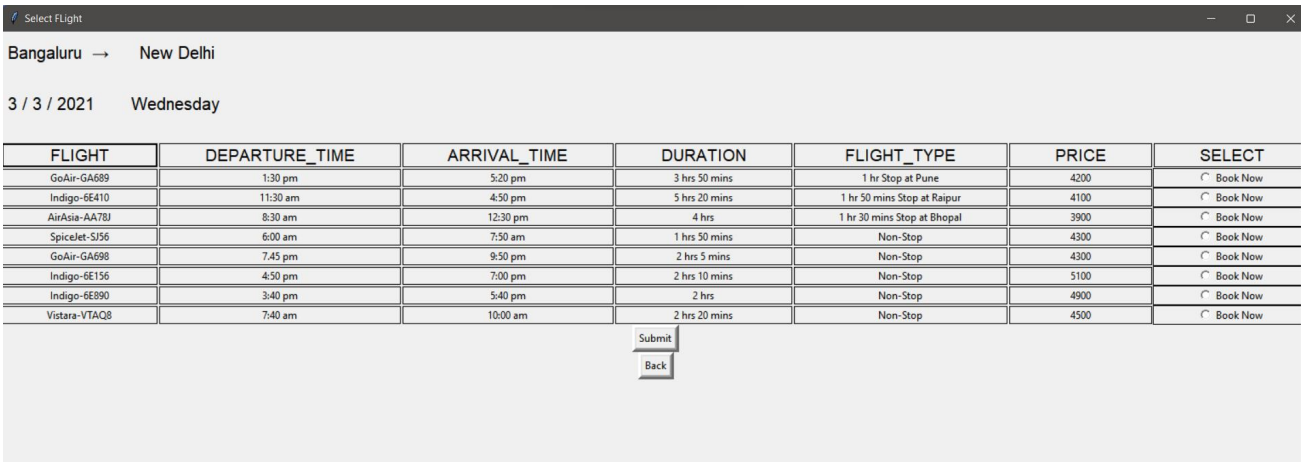
WINDOW 0- Introductory Page



WINDOW 1 - Travel Details Page

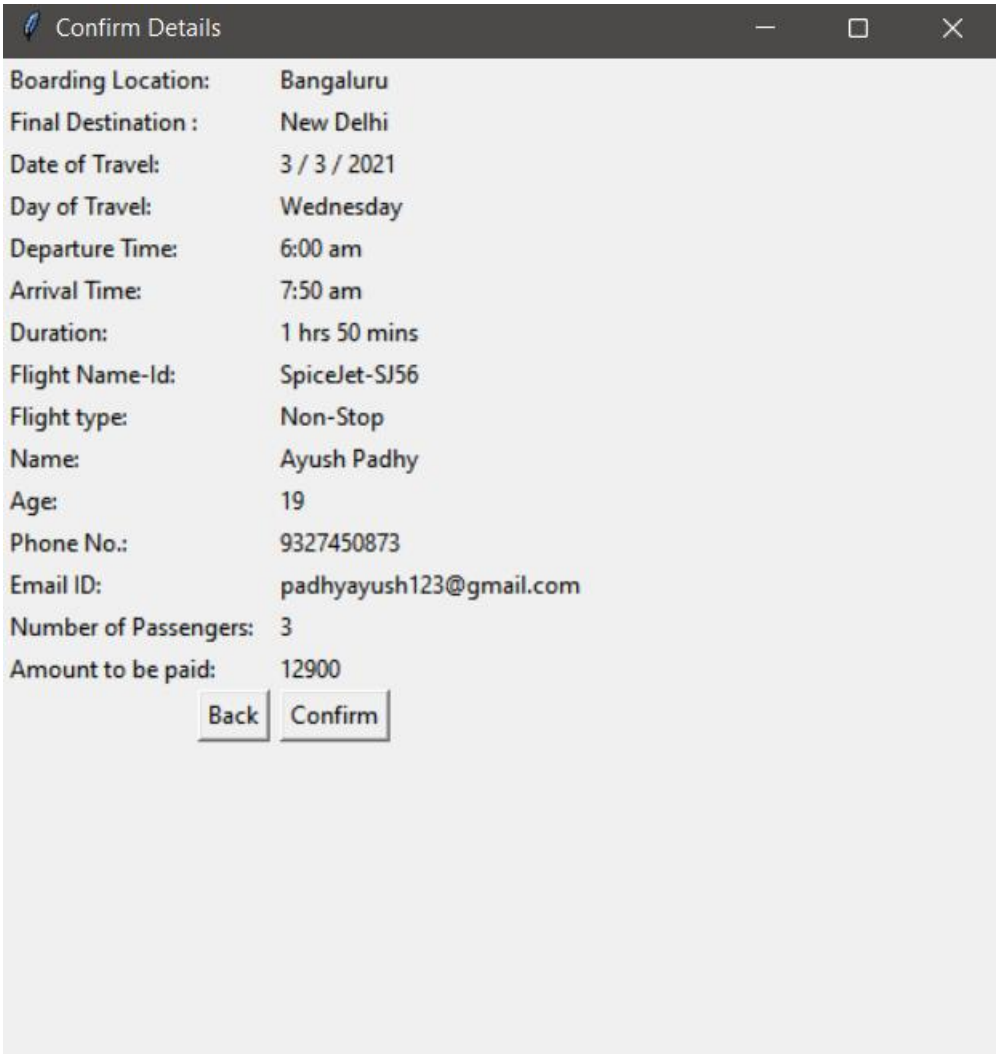


WINDOW 2 - Flight Details

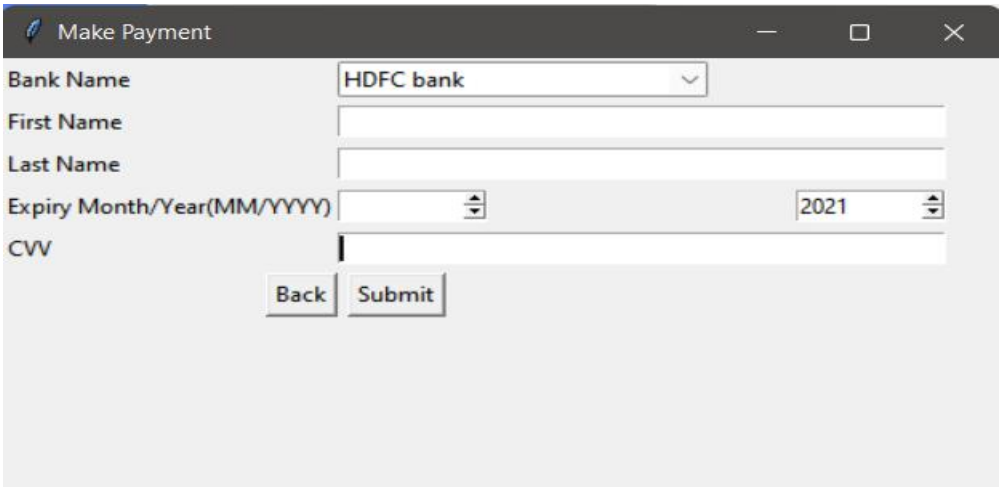


WINDOW 3 - Contact Details Page

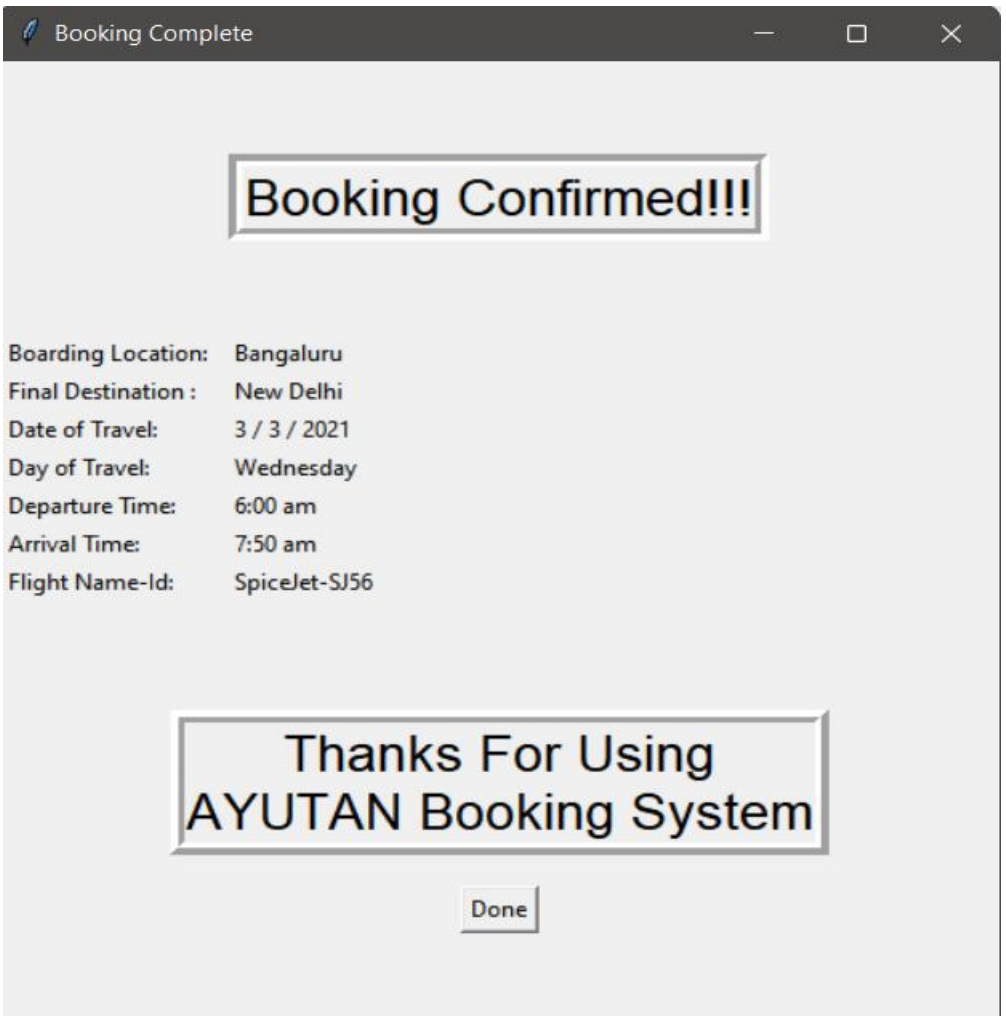
WINDOW 4 - Confirm Details Page



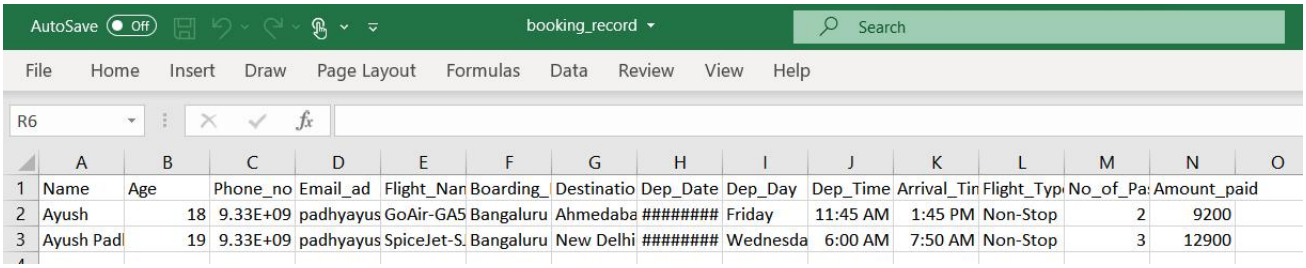
WINDOW 5 - Payment Details Page



WINDOW 6 - Booking Confirm Page



ALL BOOKINGS CSV FILE CREATED AT FILE DIRECTORY



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Age	Phone_no	Email_ad	Flight_Nan	Boarding_	Destinatio	Dep_Date	Dep_Day	Dep_Time	Arrival_Tin	Flight_Typ	No_of_Pa	Amount_paid	
2	Ayush	18	9.33E+09	padhyayus	GoAir-GA5	Bangaluru	Ahmedaba	########	Friday	11:45 AM	1:45 PM	Non-Stop	2	9200	
3	Ayush Pad	19	9.33E+09	padhyayus	SpiceJet-S	Bangaluru	New Delhi	########	Wednesda	6:00 AM	7:50 AM	Non-Stop	3	12900	

HARDWARE AND SOFTWARE REQUIREMENTS

I.OPERATING SYSTEM	:	WINDOWS 10 AND ABOVE
II. PROCESSOR	:	MINIMUM AMD PROCESSER
III. RAM	:	2GB+
IV. Hard disk	:	100 GB OR ABOVE

OTHER SOFTWARE REQUIREMENTS:

- Windows 7 onwards
- Python 3.0 onwards
- MYSQL Latest version

OTHER REQUIREMENTS:

1. GUI & CSV Module should be installed in python.

2. Change the password of your MYSQL to that of your desktop for the MYSQL connectivity in python to function.

3. Keep the image “Photo.pmg” in the same folder as of the program for the program to function.

BIBLIOGRAPHY

- Computer science With Python - Class XI and XII By : SumitaArora
- Website: geeksforgeeks.com