from tkinter import \*

from tkinter import messagebox

import sqlite3

conn = sqlite3.Connection("Database.db")

c = conn.cursor()

c.execute(

    "CREATE Table IF NOT EXISTS BUS(OPERATOR text,BUS\_TYPE text,D\_FROM text,A\_TO text,DATE text,DEP\_TIME text,ARR\_TIME text,FARE integer,SEATS integer)")

conn.commit()

splash\_root = Tk()

splash\_root.geometry("1000x1000")

splash\_label = Label(splash\_root, text="Python and DBS Project", font="Times 50 bold", relief="raised")

splash\_label.pack()

img1 = PhotoImage(file="Python.png")

splash\_image = Label(splash\_root, image=img1).pack()

splash\_label = Label(splash\_root, text="Samarth Dubey", font="Times 30 bold")

splash\_label.pack()

splash\_label = Label(splash\_root, text="191B304", font="Times 30 bold")

splash\_label.pack()

splash\_label = Label(splash\_root, text="B9", font="Times 30 bold")

splash\_label.pack()

def splashDestroy():

    splash\_root.destroy()

splash\_root.after(200,splashDestroy)

splash\_root.mainloop()

root = Tk()

root.geometry("1000x1000")

root\_label = Label(root, text="BUS BOOKING SERVICE", fg="blue",font="Times 50 bold", relief="raised")

root\_label.grid(columnspan=3,padx=100,pady=30)

img = PhotoImage(file="Bus.png")

root\_image = Label(root, image=img).grid(columnspan=3,padx=100,pady=100)

def add\_bus():

    add\_root = Tk()

    global img2

    add\_root.geometry("1000x1000")

    # img2 = PhotoImage(file="bus.png")

    # Label(add\_root, image=img2).grid()

    add\_root\_label = Label(add\_root, text="BUS BOOKING SERVICE", fg="blue",font="Times 50 bold", relief="raised")

    add\_root\_label.grid(padx=100,pady=30,columnspan=2)

    def detail():

        l1=Label(add\_root,text="Operator ID").grid()

        e1=Entry(add\_root)

        e1.grid(row=9,column=1)

        V=StringVar(add\_root)

        V.set("All Types")

        choice=['A/C','Non-A/C','A/C Sleeper','Non-A/C Sleeper','All Types']

        l2=Label(add\_root, text="Bus Type").grid()

        e2=OptionMenu(add\_root, V, \*choice)

        e2.grid(row=10,column=1)

        l3=Label(add\_root,text="From Where").grid()

        e3=Entry(add\_root)

        e3.grid(row=11,column=1)

        l4=Label(add\_root,text="To Where").grid()

        e4=Entry(add\_root)

        e4.grid(row=12,column=1)

        l5=Label(add\_root,text="Date").grid()

        e5=Entry(add\_root)

        e5.grid(row=13,column=1)

        l6=Label(add\_root,text="Arrival Time").grid()

        e6=Entry(add\_root)

        e6.grid(row=14,column=1)

        l6=Label(add\_root,text="Departure Time").grid()

        e7=Entry(add\_root)

        e7.grid(row=15,column=1)

        l7=Label(add\_root,text="Fare").grid()

        e8=Entry(add\_root)

        e8.grid(row=16,column=1)

        l7=Label(add\_root,text="Seats").grid()

        e9=Entry(add\_root)

        e9.grid(row=17,column=1)

        def Save():

            conn = sqlite3.connect('Database.db')

            c = conn.cursor()

            values =(e1.get(),V.get(),e3.get(),e4.get(),e5.get(),e6.get(),e7.get(),e8.get(),e9.get())

            c.execute("""INSERT INTO BUS(OPERATOR,BUS\_TYPE,D\_FROM,A\_TO ,DATE,DEP\_TIME ,ARR\_TIME ,FARE ,SEATS)

                          VALUES(?,?,?,?,?,?,?,?,?)""", values)

            values =(e1.get(),V.get(),e3.get(),e4.get(),e5.get(),e6.get(),e7.get(),e8.get(),e9.get())

            conn.commit()

            row = c.fetchall()

            print(row)

            conn.close()

            add\_root.destroy()

            messagebox.showinfo("DATA", "DATA SAVED")

            add\_bus.destroy()

        b2=Button(add\_root,text="Save",command=Save).grid(row=18,column=1)

    Heading=Label(add\_root,text="Bus Operator Detail Filling" ,font='Times 30 bold').grid(padx=100,pady=10,columnspan=2)

    l1=Label(add\_root,text="Full Name").grid(row=5)

    e1=Entry(add\_root)

    e1.grid(row=5,column=1)

    l2=Label(add\_root,text="Contact Number").grid()

    e2=Entry(add\_root)

    e2.grid(row=6,column=1)

    l3=Label(add\_root,text="Address").grid()

    e3=Entry(add\_root)

    e3.grid(row=7,column=1)

    def check():

        if(e1.get()=="" or e2.get()=="" or e3.get==""):

            add\_root.destroy()

            messagebox.showinfo("error","Insert values")

        else:

            detail()

    b1=Button(add\_root,text="Add Details",command=check).grid(row=8,column=1)

    add\_root.mainloop()

def search\_bus():

    search\_root = Tk()

    search\_root.geometry("1000x1000")

    search\_root\_label = Label(search\_root, text="BUS BOOKING SERVICE", fg="blue",font="Times 50 bold", relief="raised")

    search\_root\_label.grid(padx=100,pady=30,columnspan=2)

    Heading=Label(search\_root,text="Bus Details" ,font='Times 30 bold').grid(padx=100,pady=10,columnspan=2)

    V=StringVar(search\_root)

    V.set("All Types")

    choice=['A/C','Non-A/C','A/C Sleeper','Non-A/C Sleeper','All Types']

    l1=Label(search\_root, text="Type of Bus").grid(row=5,column=0)

    OptionMenu(search\_root, V, \*choice).grid(row=5,column=1)

    l2 = Label(search\_root, text="From Where").grid(row=6,column=0)

    e1 = Entry(search\_root)

    e1.grid(row=6,column=1)

    l3 = Label(search\_root, text="Where to").grid(row=7,column=0)

    e2= Entry(search\_root)

    e2.grid(row=7,column=1)

    l4= Label(search\_root, text="Date").grid(row=8,column=0)

    e3= Entry(search\_root)

    e3.grid(row=8,column=1)

    def bookingScreen():

            search\_root.destroy()

            bookingScreen\_root = Tk()

            bookingScreen\_root.geometry("1000x1000")

            bookingScreen\_root\_label = Label(bookingScreen\_root, text="BUS BOOKING SERVICE", fg="blue",font="Times 50 bold", relief="raised")

            bookingScreen\_root\_label.grid(padx=100,pady=30,columnspan=11)

            Heading=Label(bookingScreen\_root,text="Booking Screen" ,font='Times 30 bold').grid(padx=100,pady=10,columnspan=11)

            conn = sqlite3.connect('Database.db')

            c = conn.cursor()

            E1=(e1.get())

            E2=(e2.get())

            c.execute("SELECT \* FROM BUS WHERE D\_FROM= ? AND A\_TO = ?  ",(E1,E2))

            conn.commit()

            ro=c.fetchall()

            for i in ro:

                qwerty=Label(op,text=i , bg="yellow")

                qwerty.pack(padx=5,pady=20,side=LEFT).pack(padx=40)

                qwe=Label(op,text='')

                qwe.pack()

            print(ro)

            conn.close()

            l1=Label(bookingScreen\_root, text="Operator").grid(row=5,column=0)

            l2=Label(bookingScreen\_root, text="Type").grid(row=5,column=1)

            l3=Label(bookingScreen\_root, text="From").grid(row=5,column=2)

            l4=Label(bookingScreen\_root, text="To").grid(row=5,column=3)

            l5=Label(bookingScreen\_root, text="Date").grid(row=5,column=4)

            l6=Label(bookingScreen\_root, text="Dept Time").grid(row=5,column=5)

            l7=Label(bookingScreen\_root, text="Arr Time").grid(row=5,column=6)

            l8=Label(bookingScreen\_root, text="Fare").grid(row=5,column=7)

            l9=Label(bookingScreen\_root, text="Seats Availability").grid(row=5,column=8)

            def quit\_booking():

                    bookingScreen\_root.destroy()

            HomeButton = Button(bookingScreen\_root, text="Home", command=quit\_booking).grid(columnspan=11)

            bookingScreen\_root.mainloop()

    def check():

        if(e1.get()=="" or e2.get()=="" or e3.get==""):

            search\_root.destroy()

            messagebox.showinfo("error","Insert values")

        else:

            bookingScreen()

    Button1 = Button(search\_root, text="Find Buses", command=check).grid(column=1)

    def quit\_search():

        search\_root.destroy()

    HomeButton = Button(search\_root, text="Home",fg="red", command=quit\_search,padx=10,pady=10).grid(columnspan=2,padx=20,pady=20)

    search\_root.mainloop()

add\_button = Button(root, text="Add Bus", width=15, height=5, command=add\_bus).grid(column=0,row=3)

search\_button = Button(root, text="Search Bus", width=15, height=5, command=search\_bus).grid(column=2,row=3)

root.mainloop()