**Advanced Programming**

**Lab-1**

**Lab Record**

*Submitted by:*

Manan Jain

211B173

Submitted To: Dr. Mahesh Kumar



2022-2023

**Department of Computer Science & Engineering**

**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA**

**Index Page**

|  |  |
| --- | --- |
| Lab 9 | 3-6 |
| Lab 10/11 | 7-23 |

**LAB 9(GUI)**

1. Write a python script to show a root graphical window.

from tkinter import \*

root = Tk()

root.mainloop()

2. Write a python script to add hello world on the root window of GUI.

from tkinter import \*

root = Tk()

Label(root,text=’Hello World’).pack()

root.mainloop()

4. Write a python script to add a button on the root window, name the button as GO.

from tkinter import \*

root = Tk()

Button(frame, text=’GO’).pack()

root.mainloop()

5. Write a python script to add event to the button added in previous problem and add string “Welcome....” to button event to the root window

from tkinter import \*

root = Tk()

def fun():

Label(root,text=’welcome’).pack()

Button(frame, text=’GO’,command=fun).pack()

root.mainloop()

6. Write a python script to add text box in GUI.

from tkinter import \*

root = Tk()

my\_text= Text (root, width=60, height=20)

my\_text.pack(pady=20)

root.mainloop()

7. Write a python script to read First Name and Last Name using text boxes, Wish the user with the first name “ ....name.... Welcome to Python”

from tkinter import \*

root = Tk()

def fun():

Label(root,text=’welcome’+a.get()).pack()

a=Entry(root)

a.pack()

Button(root,text=’ok’,command=fun).pack()

root.mainloop()

8. Write a python script to read two numbers and print their Sum/ Diff/ Multiplication/ Div/ Remainder on screen, using button for each operation.

from tkinter import \*

root=Tk()

A=Label(root,text='Enter Value of A:-')

A.pack()

a=Entry(root)

a.pack()

B=Label(root,text='Enter Value of B:-')

B.pack()

b=Entry(root)

b.pack()

def plus():

n1=float(a.get())

n2=float(b.get())

s=n1+n2

Label(root,text='{} + {} = {}'.format(n1,n2,s)).pack()

def minus():

n1=float(a.get())

n2=float(b.get())

s=n1-n2

Label(root,text='{} - {} = {}'.format(n1,n2,s)).pack()

def multi():

n1=float(a.get())

n2=float(b.get())

s=n1\*n2

Label(root,text='{} \* {} = {}'.format(n1,n2,s)).pack()

def div():

n1=float(a.get())

n2=float(b.get())

s=n1/n2

Label(root,text='{} / {} = {}'.format(n1,n2,s)).pack()

def modulo():

n1=float(a.get())

n2=float(b.get())

s=n1%n2

Label(root,text='{} % {} = {}'.format(n1,n2,s)).pack()

s=Button(root,text=' + ',command=plus)

d=Button(root,text=' - ',command=minus)

m=Button(root,text=' \* ',command=multi)

di=Button(root,text=' / ',command=div)

mod=Button(root,text=' % ',command=modulo)

s.pack()

d.pack()

m.pack()

di.pack()

mod.pack()

root.title('First Calculator')

root.mainloop()

9. Write a python script that creates a GUI with a single button. When the button is pressed it should create a second button. When that button is pressed, it should create a label that says, “Nice job!”. What happens if you press the buttons more than once?

from tkinter import \*

root = Tk()

def fun():

Button(root,text=’ok’,command=fun1).pack()

def fun1():

Label(root,text=Nice Job).pack()

Button(root,text=’ok’,command=fun).pack()

root.mainloop()

10. Write a python script to computer simple interest on the root window.

from tkinter import \*

root=Tk()

pri=Label(root,text='Enter Principle Amount($):-')

pri.pack()

p=Entry(root)

p.pack()

roi=Label(root,text='Enter Rate of Interest(%):-')

roi.pack()

r=Entry(root)

r.pack()

time=Label(root,text='Enter Time(Years):-')

time.pack()

t=Entry(root)

t.pack()

def SI():

root.bell()

a=float(p.get())

b=float(r.get())

c=float(t.get())

si=a\*b\*c/100

ans=Label(root,text='Simple Interest is ${}'.format(si))

ans.pack()

calculate=Button(root,text='Calculate',command=SI)

calculate.pack()

root.title('Simple Interest Calculator')

root.mainloop()

**LAB 10/11(Tkinter)**

1. See the video clip of the project available in project folder on the server.

2. Create first GUI window containing image, project title, details of the student etc. Use the image available in the project folder.

3. Create second GUI window as shown in video clip.

4. Create third GUI window shown in video clip.

5. Create fourth GUI window shown in video clip, use home image on the button.

6. Create GUI windows for shown video clip (Project) adding all the required components including radio buttons, and drop-down menu wherever needed.

7. Add Popup functionality to validate all the input fields.

8. Display appropriate messages through Popup to give relevant information / error to the user.

9. Pops must cover all inputs validations i.e. input through radio buttons, dropdown menu, and entries.

from tkinter import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

#root.geometry('{}x{}'.format(w,h))

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=3,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,padx=w//3)

name=Label(root,text='Name: Manan Jain',fg='Blue2',font='Arial 16 bold',pady=70)

er=Label(root,text='Er: 211B173',fg='Blue2',font='Arial 16 bold')

mob=Label(root,text='Mobile: 8871113041',fg='Blue2',font='Arial 16 bold',pady=70)

sub=Label(root,text='Submitted to : Dr. Mahesh Kumar',bg='light blue',fg='Red',font='Arial 24 bold')

prj=Label(root,text='Project Based Learning',fg='Red',font='Arial 14')

name.grid(row=2,column=0,padx=w//3)

er.grid(row=3,column=0,padx=w//3)

mob.grid(row=4,column=0,padx=w//3)

sub.grid(row=5,column=0,padx=w//3)

prj.grid(row=6,column=0,padx=w//3)

def fun(e=0):

    root.destroy()

    import BuyAdd

root.bind('<KeyPress>',fun)

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=3,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=3,padx=w//3)

def nextseat():

    root.destroy()

    import bookwindow

def nextbooked():

    root.destroy()

    import checkbooking

def nextbus():

    root.destroy()

    import Newadd

seat=Button(root,text='Seat Booking',bg='light green',font='Arial 20 bold',command=nextseat)

booked=Button(root,text='Check Booked Seat',bg='green3',font='Arial 20 bold',command=nextbooked)

buses=Button(root,text='Add Bus Details',bg='dark green',font='Arial 20 bold',command=nextbus)

admin=Label(root,text='For Admin Only',fg='Red',font='TimesNewRoman 14 bold')

seat.grid(row=2,column=0,pady=60)

booked.grid(row=2,column=1,pady=60)

buses.grid(row=2,column=2,pady=60)

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

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=12,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=12,padx=w//3)

detailtxt=Label(root,text='Enter Journey Details',fg='green4',bg='light green',font='Arial 16 bold')

detailtxt.grid(row=2,column=0,columnspan=12,padx=w//3,pady=20)

frame1=Frame(root)

frame1.grid(row=3,column=0,columnspan=12)

frame2=Frame(root)

frame2.grid(row=8,column=0,columnspan=20,pady=20)

import sqlite3

con=sqlite3.connect('pythonbus.db')

cur=con.cursor()

bus\_select=IntVar()

tof=Entry(frame1)

fromf=Entry(frame1)

datef=Entry(frame1)

def confirmamt():

    if(checkerrordetails()==False):

        cur.execute("""select fare,name from bus,operator where busid={} and bus\_opid=opid""".format(bus\_select.get(),bus\_select.get()))

        price=cur.fetchall()

        amt=price[0][0]

        opname=price[0][1]

        with open(".//mobile.txt","r+") as f:

            f.write("{}".format(int(mobilef.get())))

            print(f.read())

        ans=askquestion('Fare Confirmation','Total amount to be paid is Rs {}'.format(amt\*int(seatsf.get())))

        if ans=='yes':

            cur.execute("""select DATE('now')""")

            date=cur.fetchall()

            curdate=date[0][0]

            dated=dater()

            cur.execute("""insert into bookinghistory (pname,mobile,age,seats,from\_station,to\_station,date\_booked,gender,boarding\_date,fare,op\_name)values("{}",{},{},{},"{}","{}","{}","{}","{}",{},"{}")""".format(namef.get(),mobilef.get(),agef.get(),seatsf.get(),fromf.get(),tof.get(),curdate,sex\_mf.get(),dated,amt,opname))

            cur.execute("""update runs set seat\_available=seat\_available-{} where runs\_busid={}""".format(int(seatsf.get()),bus\_select.get()))

            con.commit()

            root.destroy()

            import ticketbooked

        else:

            return

def dater():

    olddate=datef.get()

    newdate=olddate[6:]+'-'+olddate[3:5]+'-'+olddate[:2]

    return newdate

def showbuses():

    if(checkerrortofromdate()==False):

        frame=Frame(root)

        frame.grid(row=5,column=0,columnspan=12)

        dated=dater()

        cur.execute("""select op.name,b.bus\_type,r.seat\_available,b.capacity,b.fare,b.bus\_opid,st.stid as start\_st,ed.stid as end\_st from operator as op,bus as b,route as st,route as ed,runs as r where r.runs\_date='{}' and st.station\_name="{}" and ed.station\_name="{}" and st.stid< ed.stid and st.rid=ed.rid and b.bus\_rid=st.rid and b.bus\_opid=op.opid and r.runs\_busid=b.busid""".format(dated,fromf.get(),tof.get()))

        res=cur.fetchall()

        buses\_count=len(res)

        select\_bus=Label(frame,text='Select Bus',fg='green3',font='Arial 14 bold')

        op=Label(frame,text='Operator',fg='green3',font='Arial 14')

        bus\_type=Label(frame,text='Bus Type',fg='green3',font='Arial 14')

        avail=Label(frame,text='Available/Capacity',fg='green3',font='Arial 14')

        fare=Label(frame,text='Fare',fg='green3',font='Arial 14')

        bookproceed=Button(frame,text='Proceed to book',bg='light green',font='Arial 14',command=proceedtobook)

        if buses\_count==0:

            showerror('No BUS','NO BUSES FOUND')

            return

        i=0

        for i in range(0,buses\_count):

            Radiobutton(frame,text='Bus'+str(i+1),variable=bus\_select, value=res[i][5]).grid(row=5+i,column=1)

            Label(frame,text=res[i][0],font='Arial 12',fg='blue').grid(row=5+i,column=2)

            Label(frame,text=res[i][1],font='Arial 12',fg='blue').grid(row=5+i,column=3)

            Label(frame,text=str(res[i][2])+'/'+str(res[i][3]),font='Arial 12',fg='blue').grid(row=5+i,column=4)

            Label(frame,text=res[i][4],font='Arial 12',fg='blue').grid(row=5+i,column=5)

        select\_bus.grid(row=4,column=1,padx=5)

        op.grid(row=4,column=2,padx=5)

        bus\_type.grid(row=4,column=3,padx=5)

        avail.grid(row=4,column=4,padx=5)

        fare.grid(row=4,column=5,padx=5)

        bookproceed.grid(row=6+i,column=7)

filldetail=Label(frame2,text='Fill Passenger Details to book the bus ticket',bg='light blue',fg='Red',font='Arial 24 bold')

name=Label(frame2,text='Name',font='Arial 14 bold')

sex=Label(frame2,text='Gender',font='Arial 14 bold')

sex\_mf=StringVar()

sex\_mf.set('--M/F--')

option=['Male','Female','Other']

s\_menu=OptionMenu(frame2,sex\_mf,\*option)

seats=Label(frame2,text='No of Seats',font='Arial 14 bold')

mobile=Label(frame2,text='Mobile No',font='Arial 14 bold')

age=Label(frame2,text='Age',font='Arial 14 bold')

namef=Entry(frame2)

seatsf=Entry(frame2)

mobilef=Entry(frame2)

agef=Entry(frame2)

def proceedtobook():

    if bus\_select.get()==0:

        showerror('SELECT ERROR','NO BUS SELECTED')

    else:

        bookseat=Button(frame2,text='Book Seat',bg='light green',font='Arial 14',command=confirmamt)

        dated=dater()

        cur.execute("""select b.fare,b.bus\_opid,st.stid as start\_st,ed.stid as end\_st from bus as b,route as st,route as ed,runs as r where r.runs\_date='{}' and st.station\_name="{}" and ed.station\_name="{}" and st.stid< ed.stid and st.rid=ed.rid and b.busid={}""".format(dated,fromf.get(),tof.get(),bus\_select.get()))

        bus\_detail=cur.fetchall()

        filldetail.grid(row=7,column=0,columnspan=20,pady=15)

        name.grid(row=8,column=1)

        namef.grid(row=8,column=2)

        sex.grid(row=8,column=3)

        s\_menu.grid(row=8,column=4)

        seats.grid(row=8,column=5)

        seatsf.grid(row=8,column=6)

        mobile.grid(row=8,column=7)

        mobilef.grid(row=8,column=8)

        age.grid(row=8,column=9)

        agef.grid(row=8,column=10)

        bookseat.grid(row=8,column=11)

def checkerrortofromdate():

    if fromf.get()=='':

        showerror('Error','Source Empty')

        return True

    elif tof.get()=='':

        showerror('Error','Destination Empty')

        return True

    elif datef.get()=='':

        showerror('Error','Date Empty')

        return True

    else:

        return False

def checkerrordetails():

    if namef.get()=='':

        showerror('Error','Name Empty')

        return True

    elif sex\_mf=='':

        showerror('Error','No Gender Selected')

        return True

    elif mobilef.get()=='':

        showerror('Error','No Mobile No Entered')

        return True

    elif agef.get()=='':

        showerror('Error','No Age Entered')

        return True

    elif seatsf.get()=='':

        showerror('Error','No Seats Entered')

        return True

    else:

        return False

def takehome():

    root.destroy()

    import BuyAdd

to=Label(frame1,text='To',font='Arial 12')

From=Label(frame1,text='From',font='Arial 12')

date=Label(frame1,text='Journey Date',font='Arial 12')

show=Button(frame1,text='Show Bus',command=showbuses,bg='SeaGreen1',font='Arial 16 bold')

home=Button(frame1,image=img1,bg='light green',command=takehome)

to.grid(row=3,column=1,sticky=E,padx=10)

tof.grid(row=3,column=2,sticky=W,padx=10)

From.grid(row=3,column=3,sticky=E,padx=10)

fromf.grid(row=3,column=4,sticky=W,padx=10)

date.grid(row=3,column=5,sticky=E,padx=10)

datef.grid(row=3,column=6,sticky=W,padx=10)

show.grid(row=3,column=7,padx=10)

home.grid(row=3,column=8,padx=10)

root.title('Booking Window')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

import sqlite3

con=sqlite3.Connection('pythonbus.db')

cur=con.cursor()

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=8,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=8,padx=w//3)

t2=Label(root,text='Check Your Booking',bg='green3',fg='dark green',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=8,padx=w//3,pady=20)

mob=Label(root,text='Enter your mobile number',font='Arial 14')

mobf=Entry(root)

frame1=Frame(root,relief='groove',bd=5)

frame1.grid(row=4,column=0,columnspan=8,rowspan=10,padx=w//3)

def checkbook():

    cur.execute("""select \* from bookinghistory where mobile={}""".format(int(mobf.get())))

    res=cur.fetchall()

    print(res)

    if len(res)!=0:

        pname='Passengers:'+res[0][0]

        mobile='Phone:'+str(int(res[0][1]))

        age='Age:'+str(int(res[0][2]))

        seats='No of Seats:'+str(res[0][3])

        from\_st='Boarding Point:'+str(res[0][4])

        to\_st='Destination Point:'+str(res[0][5])

        date\_booked='Booked On:'+str(res[0][6])

        ref\_number='Booking Ref.'+str(res[0][7])

        sex='Gender:'+str(res[0][8])

        date\_bus='Travel On:'+str(res[0][9])

        fare='Fare Rs. :'+str(res[0][10])+'\*'

        op\_name='Bus Detail:'+str(res[0][11])

        term='\*Total amount of Rs'+str(res[0][3]\*res[0][10])+'/- to be paid at the time of boarding the bus'

        Label(frame1,text=pname,font='Arial 12 bold').grid(row=4,column=1,padx=5)

        Label(frame1,text=mobile,font='Arial 12 bold').grid(row=4,column=2,padx=5)

        Label(frame1,text=age,font='Arial 12 bold').grid(row=5,column=1,padx=5)

        Label(frame1,text=ref\_number,font='Arial 12 bold').grid(row=5,column=2,padx=5)

        Label(frame1,text=from\_st,font='Arial 12 bold').grid(row=6,column=1,padx=5)

        Label(frame1,text=to\_st,font='Arial 12 bold').grid(row=6,column=2,padx=5)

        Label(frame1,text=date\_booked,font='Arial 12 bold').grid(row=7,column=1,padx=5)

        Label(frame1,text=date\_bus,font='Arial 12 bold').grid(row=7,column=2,padx=5)

        Label(frame1,text=sex,font='Arial 12 bold').grid(row=8,column=1,padx=5)

        Label(frame1,text=seats,font='Arial 12 bold').grid(row=8,column=2,padx=5)

        Label(frame1,text=op\_name,font='Arial 12 bold').grid(row=9,column=1,padx=5)

        Label(frame1,text=fare,font='Arial 12 bold').grid(row=9,column=2,padx=5)

        Label(frame1,text=term,font='Arial 8 italic').grid(row=10,column=1,columnspan=2)

    else:

        ch=askyesno('No Booking Record','Do you want to book seat now ?')

        if ch== True:

            root.destroy()

            import bookwindow

        else:

            return

checkb=Button(root,text='Check Booking',font='Arial 14',command=checkbook)

mob.grid(row=3,column=2)  #sticky=W or E

mobf.grid(row=3,column=3,sticky=EW,padx=5)

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

root.title('Check Ticket Window')

root.mainloop()

from tkinter import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=5,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=5,padx=w//3)

t2=Label(root,text='Add New Details to DataBase',bg='seashell2',fg='green3',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=5,padx=w//3,pady=20)

def nextop():

    root.destroy()

    import operator

def nextbus():

    root.destroy()

    import newbus

def nextroute():

    root.destroy()

    import busroute

def nextrun():

    root.destroy()

    import busrunning

op=Button(root,text='New Operator',bg='light green',font='Arial 16',command=nextop)

busnew=Button(root,text='New Bus',bg='orange red',font='Arial 16',command=nextbus)

route=Button(root,text='New Route',bg='steel blue1',font='Arial 16',command=nextroute)

newrun=Button(root,text='New Run',bg='light coral',font='Arial 16',command=nextrun)

op.grid(row=3,column=0,pady=20)

busnew.grid(row=3,column=1,pady=20)

route.grid(row=3,column=2,pady=20)

newrun.grid(row=3,column=3,pady=20)

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

import sqlite3

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=20,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=20,padx=w//3)

t2=Label(root,text='Add Bus Operator Details',bg='seashell2',fg='green3',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=20,padx=w//3,pady=20)

opid=Label(root,text='Operator ID',font='Arial 14')

name=Label(root,text='Name',font='Arial 14')

add=Label(root,text='Address',font='Arial 14')

ph=Label(root,text='Phone',font='Arial 14')

mail=Label(root,text='Email',font='Arial 14')

con=sqlite3.Connection('pythonbus.db')

cur=con.cursor()

opf=Entry(root)

nf=Entry(root)

addf=Entry(root)

phf=Entry(root)

mf=Entry(root)

frame1=Frame(root)

frame1.grid(row=4,column=0,columnspan=13)

def checkblank():

    if opf.get()=='':

        showerror('Operator ID Error','Operator ID Empty')

        return True

    elif nf.get()=='':

        showerror('Name Error','Name Empty')

        return True

    elif addf.get()=='':

        showerror('Address Error','Address Empty')

        return True

    elif phf.get()=='':

        showerror('Phone Error','Phone Empty')

        return True

    elif mf.get()=='':

        showerror('Mail Error','Email Empty')

        return True

    else:

        return False

def addnew():

    if checkblank()==False:

        cur.execute("""insert into operator (opid,name,phone,address,email)values({},"{}",{},"{}","{}")""".format(int(opf.get()),nf.get(),int(phf.get()),addf.get(),mf.get()))

        con.commit()

        op1=Label(frame1,text='{} {} {} {} {}'.format(opf.get(),nf.get(),addf.get(),phf.get(),mf.get()),font='Arial 12')

        op1.grid(row=4)

        showinfo('Operator Entry Updated','Operator Record updated successfully')

def editnew():

    if checkblank()==False:

        cur=con.cursor()

        cur.execute("""delete from operator where opid={}""".format(int(opf.get())))

        cur.execute("""insert into operator (opid,name,phone,address,email)values({},"{}",{},"{}","{}")""".format(int(opf.get()),nf.get(),int(phf.get()),addf.get(),mf.get()))

        con.commit()

        op1=Label(frame1,text='{} {} {} {} {}'.format(opf.get(),nf.get(),addf.get(),phf.get(),mf.get()),font='Arial 12')

        op1.grid(row=4)

        showinfo('Operator Entry Updated','Operator Record updated successfully')

addb=Button(root,text='Add',bg='SpringGreen2',font='Arial 14',command=addnew)

eb=Button(root,text='Edit',bg='SpringGreen2',font='Arial 14',command=editnew)

def takehome():

    con.close()

    root.destroy()

    import BuyAdd

home=Button(root,image=img1,bg='light green',command=takehome)

opid.grid(row=3,column=1)  #stick=W or E

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

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

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

add.grid(row=3,column=5)

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

ph.grid(row=3,column=7)

phf.grid(row=3,column=8)

mail.grid(row=3,column=9)

mf.grid(row=3,column=10)

addb.grid(row=3,column=11)

eb.grid(row=3,column=12)

home.grid(row=5,column=9)

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=20,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=20,padx=w//3)

t2=Label(root,text='Add Bus Details',bg='seashell2',fg='green3',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=20,padx=w//3,pady=20)

bid=Label(root,text='Bus ID',font='Arial 14')

btype=Label(root,text='Bus Type',font='Arial 14')

cap=Label(root,text='Capacity',font='Arial 14')

fare=Label(root,text='Fare Rs',font='Arial 14')

opid=Label(root,text='Operator ID',font='Arial 14')

rid=Label(root,text='Route ID',font='Arial 14')

bus\_type=StringVar()

bus\_type.set('--select--')

option=['AC 2X2','AC 3X2','Non AC 2X2','Non AC 3X2','AC-Sleeper 2X1','Non-AC Sleeper 2X1']

d\_menu=OptionMenu(root,bus\_type,\*option)

import sqlite3

con=sqlite3.connect('pythonbus.db')

cur=con.cursor()

bidf=Entry(root)

opf=Entry(root)

capf=Entry(root)

faref=Entry(root)

ridf=Entry(root)

def checkblank():

    if bidf.get()=='':

        showerror('Bus ID Error','Bus ID Empty')

        return True

    elif opf.get()=='':

        showerror('Operator Error','ID Empty')

        return True

    elif capf.get()=='':

        showerror('Capacity Error','Capacity Empty')

        return True

    elif faref.get()=='':

        showerror('Fare Error','Fare Empty')

        return True

    elif ridf.get()=='':

        showerror('Route ID Error','RID Empty')

        return True

    elif bus\_type.get()=='--select--':

        showerror('Bus Error','Bus type not defined')

        return True

    else:

        return False

def addnew1():

    if checkblank()==False:

        cur.execute("""select busid from bus where busid={}""".format(int(bidf.get())))

        res=cur.fetchall()

        if len(res)!=0:

            showerror('DB Insertion Error','Record Already Exists')

        else:

            cur.execute("""insert into bus (busid,bus\_type,bus\_opid,capacity,fare,bus\_rid)values({},"{}",{},{},{},{})""".format(int(bidf.get()),bus\_type.get(),int(opf.get()),int(capf.get()),int(faref.get()),int(ridf.get())))

            con.commit()

            op1=Label(root,text='{} {} {} {} {} {}'.format(bidf.get(),bus\_type.get(),capf.get(),faref.get(),opf.get(),ridf.get()),font='Arial 12')

            op1.grid(row=4,columnspan=13)

            showinfo('Bus Entry','Bus Record added')

def takehome():

    root.destroy()

    import BuyAdd

def editnew():

    if checkblank()==False:

        cur.execute("""delete from bus where busid={}""".format(int(bidf.get())))

        cur.execute("""insert into bus (busid,bus\_type,bus\_opid,capacity,fare,bus\_rid)values({},"{}",{},{},{},{})""".format(int(bidf.get()),bus\_type.get(),int(opf.get()),int(capf.get()),int(faref.get()),int(ridf.get())))

        con.commit()

        op1=Label(root,text='{} {} {} {} {} {}'.format(bidf.get(),bus\_type.get(),capf.get(),faref.get(),opf.get(),ridf.get()),font='Arial 12')

        op1.grid(row=4,columnspan=13)

        showinfo('Bus Entry','Bus Record added')

addb=Button(root,text='Add Bus',bg='SpringGreen2',font='Arial 14',command=addnew1)

eb=Button(root,text='Edit Bus',bg='SpringGreen2',font='Arial 14',command=editnew)

home=Button(root,image=img1,bg='light green',command=takehome)

bid.grid(row=3,column=1)  #stick=W or E

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

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

d\_menu.grid(row=3,column=4)

cap.grid(row=3,column=5)

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

fare.grid(row=3,column=7)

faref.grid(row=3,column=8)

opid.grid(row=3,column=9)

opf.grid(row=3,column=10)

rid.grid(row=3,column=11)

ridf.grid(row=3,column=12)

addb.grid(row=5,column=7)

eb.grid(row=5,column=8)

home.grid(row=5,column=9)

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=20,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=20,padx=w//3)

t2=Label(root,text='Add Bus Route Details',bg='seashell2',fg='green3',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=20,padx=w//3,pady=20)

rid=Label(root,text='Route ID',font='Arial 14')

stname=Label(root,text='Station Name',font='Arial 14')

stid=Label(root,text='Station ID',font='Arial 14')

ridf=Entry(root)

stnamef=Entry(root)

stidf=Entry(root)

import sqlite3

con=sqlite3.connect('pythonbus.db')

cur=con.cursor()

def checkblank():

    if ridf.get()=='':

        showerror('Route ID Error','Route ID Empty')

        return True

    elif stnamef.get()=='':

        showerror('Station Error','Station Name Empty')

        return True

    elif stidf.get()=='':

        showerror('Station Error','Station ID Empty')

        return True

    else:

        return False

def takehome():

    con.close()

    root.destroy()

    import BuyAdd

def addnew():

    if checkblank()==False:

        cur.execute("""insert into route (rid,stid,station\_name)values({},{},"{}")""".format(int(ridf.get()),int(stidf.get()),stnamef.get()))

        con.commit()

        op1=Label(root,text='{} {} {}'.format(ridf.get(),stnamef.get(),stidf.get()),font='Arial 12')

        op1.grid(row=4,columnspan=13)

        showinfo('Route Entry Updated','Bus Route Record updated successfully')

def editnew():

    if checkblank()==False:

        cur.execute("""select \* from route where rid={} and stid={}""".format(int(ridf.get()),int(stidf.get())))

        res=cur.fetchall()

        if(len(res)==0):

            showerror('No Route','No Route Found with specified Details to delete')

            return

        else:

            cur.execute("""delete from route where rid={} and stid={}""".format(int(ridf.get()),int(stidf.get())))

            con.commit()

        op1=Label(root,text='{} {} {}'.format(ridf.get(),stnamef.get(),stidf.get()),font='Arial 12')

        op1.grid(row=4,columnspan=13)

        showinfo('Route Entry Updated','Bus Route Record updated successfully')

addb=Button(root,text='Add Route',bg='SpringGreen2',font='Arial 14',command=addnew)

eb=Button(root,text='Delete Route',bg='SpringGreen2',fg='Red',font='Arial 14',command=editnew)

home=Button(root,image=img1,bg='light green',command=takehome)

rid.grid(row=3,column=1)  #stick=W or E

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

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

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

stid.grid(row=3,column=5)

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

addb.grid(row=3,column=8)

eb.grid(row=3,column=9)

home.grid(row=5,column=7)

root.title('Python Bus Service')

root.mainloop()

from tkinter import \*

from tkinter.messagebox import \*

root=Tk()

w,h=root.winfo\_screenwidth(),root.winfo\_screenheight()

root.geometry('%dx%d+0+0'%(w,h))

img=PhotoImage(file='.\\bus.png')

img1=PhotoImage(file='.\\home.png')

bus=Label(root,image=img)

bus.grid(row=0,column=0,columnspan=20,padx=w//3)

t1=Label(root,text='Online Bus Booking System',bg='light blue',fg='Red',font='Arial 32 bold')

t1.grid(row=1,column=0,columnspan=20,padx=w//3)

t2=Label(root,text='Add Bus Running Details',bg='seashell2',fg='green3',font='Arial 22 bold')

t2.grid(row=2,column=0,columnspan=20,padx=w//3,pady=20)

bid=Label(root,text='Bus ID',font='Arial 14')

rdate=Label(root,text='Running Date',font='Arial 14')

sav=Label(root,text='Seat Available',font='Arial 14')

import sqlite3

con=sqlite3.connect('pythonbus.db')

cur=con.cursor()

bidf=Entry(root)

rdatef=Entry(root)

savf=Entry(root)

def takehome():

    con.close()

    root.destroy()

    import BuyAdd

def checkblank():

    if bidf.get()=='':

        showerror('BUS ID Error','Bus ID Empty')

        return True

    elif rdatef.get()=='':

        showerror('Date Error','Date Error Empty')

        return True

    elif savf.get()=='':

        showerror('Seats Error','Seats Empty')

        return True

    else:

        return False

def dater():

    olddate=rdatef.get()

    newdate=olddate[6:]+'-'+olddate[3:5]+'-'+olddate[:2]

    return newdate

def addnew2():

    if checkblank()==False:

        dated=dater()

        cur.execute("""insert into runs(runs\_busID,runs\_date,seat\_available)values({},'{}',{})""".format(int(bidf.get()),dated,int(savf.get())))

        con.commit()

        op1=Label(root,text='{} {} {}'.format(bidf.get(),rdatef.get(),savf.get()),font='Arial 12')

        op1.grid(row=4,columnspan=13)

        showinfo('Bus Running Updated','Bus Running Record updated successfully')

def editnew():

    if checkblank()==False:

        dated=dater()

        cur.execute("""delete from runs where runs\_busID={}""".format(int(bidf.get())))

        cur.execute("""insert into runs(runs\_busID,runs\_date,seat\_available)values({},'{}',{})""".format(int(bidf.get()),dated,int(savf.get())))

        con.commit()

        op1=Label(root,text='{} {} {}'.format(bidf.get(),rdatef.get(),savf.get()),font='Arial 12')

        op1.grid(row=4,columnspan=13)

        showinfo('Bus Running Updated','Bus Running Record updated successfully')

addb=Button(root,text='Add Run',bg='SpringGreen2',font='Arial 14',command=addnew2)

eb=Button(root,text='Delete Run',bg='SpringGreen2',fg='Red',font='Arial 14',command=editnew)

home=Button(root,image=img1,bg='light green',command=takehome)

bid.grid(row=3,column=1)  #stick=W or E

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

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

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

sav.grid(row=3,column=5)

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

addb.grid(row=3,column=8)

eb.grid(row=3,column=9)

home.grid(row=5,column=8)

root.title('Python Bus Service')

root.mainloop()