from os import stat

from tkinter import\*

from PIL import Image, ImageTk *#pip install pillow*

from tkinter import ttk

import random

from time import strftime

from datetime import datetime

import mysql.connector

from tkinter import messagebox

class Roombooking:

    def \_\_init\_\_(self,root):

        self.root=root

        self.root.title("Hotel Management System")

        self.root.geometry("1121x452+234+243")

*################ VARIABLES #############*

        self.var\_contact = StringVar()

        self.var\_check\_in = StringVar()

        self.var\_check\_out = StringVar()

        self.var\_room\_type = StringVar()

        self.var\_room\_available = StringVar()

        self.var\_meal = StringVar()

        self.var\_no\_of\_days = StringVar()

        self.var\_paid\_tax = StringVar()

        self.var\_actual\_total = StringVar()

        self.var\_total = StringVar()

*################ Title ##################*

        lbl\_title = Label(self.root,text="ROOMBOOKING DETAILS",font=("times new roman",18,"bold"),bg="blue",fg="white",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=0,width=1121,height=35)

*################## Logo ######################*

        img2 = Image.open("clg-logo.jpeg")

        img2 = img2.resize((100,35),Image.ANTIALIAS)

        self.photoimg2=ImageTk.PhotoImage(img2)

        lblimg = Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)

        lblimg.place(x=0,y=0,width=100,height=35)

*################ LABEL FRAME ###################*

        lblFrameLeft=LabelFrame(self.root,bd=2,relief=RIDGE,text="ROOMBOOKING DETAILS",font=("times new roman",12,"bold"),padx=2)

        lblFrameLeft.place(x=5,y=40,width=425,height=410)

*############### LABELS & ENTRIES ##############*

*############## CUSTOMER CONTACT. #################*

        lbl\_cust\_contact= Label(lblFrameLeft,text="STUDENT CONTACT :",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_cust\_contact.grid(row=0,column=0,sticky=W)

        entry\_contact = ttk.Entry(lblFrameLeft,textvariable= self.var\_contact,width=20,font=("arial",11,"bold"),state=DISABLED)

        entry\_contact.grid(row=0,column=1,sticky=W)

*############# FEATCH DATA BUTTON ##########*

        btn\_featch\_data = Button(lblFrameLeft,command=self.fetch\_contact,text="FETCH DATA",font=("arial",8,"bold"),bg="blue",fg="white",width=10)

        btn\_featch\_data.place(x=333,y=4)

*############# CHECK IN DATE ################*

        lblcheck\_in\_date = Label(lblFrameLeft,text="CHECK IN DATE :",font=("arial",10,"bold"),padx=2,pady=6)

        lblcheck\_in\_date.grid(row=1,column=0,sticky=W)

        txtcheck\_in\_date=ttk.Entry(lblFrameLeft,textvariable=self.var\_check\_in,width=29,font=("arial",11,"bold"))

        txtcheck\_in\_date.grid(row=1,column=1)

*############ CHECK OUT DATE ##############*

        lblcheck\_out\_date = Label(lblFrameLeft,text="CHECK OUT DATE :",font=("arial",10,"bold"),padx=2,pady=6)

        lblcheck\_out\_date.grid(row=2,column=0,sticky=W)

        txtcheck\_out\_date=ttk.Entry(lblFrameLeft,textvariable=self.var\_check\_out,width=29,font=("arial",11,"bold"))

        txtcheck\_out\_date.grid(row=2,column=1)

*############ ROOM TYPE ###############*

        lblroom\_type=Label(lblFrameLeft,text="ROOM TYPE :",font=("arial",10,"bold"),padx=2,pady=6)

        lblroom\_type.grid(row=3,column=0,sticky=W)

        conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

        my\_cursor = conn.cursor()

        my\_cursor.execute("Select ROOM\_TYPE from room")

        data\_rows = my\_cursor.fetchall()

        combo\_room\_type = ttk.Combobox(lblFrameLeft,textvariable=self.var\_room\_type,font=("arial",10,"bold"),width=31,stat="readonly")

        combo\_room\_type["value"]=data\_rows

        combo\_room\_type.current(0)

        combo\_room\_type.grid(row=3,column=1)

*############ AVAILABLE ROOM ##########*

        lblroom\_available = Label(lblFrameLeft,text="ROOM AVAILABLE :",font=("arial",10,"bold"),padx=2,pady=6)

        lblroom\_available.grid(row=4,column=0,sticky=W)

*#txtroom\_available=ttk.Entry(lblFrameLeft,textvariable=self.var\_room\_available,width=29,font=("arial",11,"bold"))*

*#txtroom\_available.grid(row=4,column=1)*

        conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

        my\_cursor = conn.cursor()

        my\_cursor.execute("Select ROOM from room")

        rows = my\_cursor.fetchall()

        combo\_room\_no = ttk.Combobox(lblFrameLeft,textvariable=self.var\_room\_available,font=("arial",10,"bold"),width=31,stat="readonly")

        combo\_room\_no["value"]=rows

        combo\_room\_no.current(0)

        combo\_room\_no.grid(row=4,column=1)

*# ############ MEAL ###############*

*# lblroom\_meal = Label(lblFrameLeft,text="MEAL :",font=("arial",10,"bold"),padx=2,pady=6)*

*# lblroom\_meal.grid(row=5,column=0,sticky=W)*

*# txtroom\_meal=ttk.Entry(lblFrameLeft,textvariable=self.var\_meal,width=29,font=("arial",11,"bold"))*

*# txtroom\_meal.grid(row=5,column=1)*

*# ########### NUMBER OF DAYS ################*

*# lblno\_of\_days = Label(lblFrameLeft,text="NO. OF DAYS :",font=("arial",10,"bold"),padx=2,pady=6)*

*# lblno\_of\_days.grid(row=6,column=0,sticky=W)*

*# txtno\_of\_days=ttk.Entry(lblFrameLeft,textvariable=self.var\_no\_of\_days,width=29,font=("arial",11,"bold"))*

*# txtno\_of\_days.grid(row=6,column=1)*

*########## PAID TAX ###############*

        lblno\_of\_days = Label(lblFrameLeft,text="PAID TAX :",font=("arial",10,"bold"),padx=2,pady=6)

        lblno\_of\_days.grid(row=7,column=0,sticky=W)

        txtno\_of\_days=ttk.Entry(lblFrameLeft,textvariable=self.var\_paid\_tax,width=29,font=("arial",11,"bold"))

        txtno\_of\_days.grid(row=7,column=1)

*########### SUB TOTAL ################*

        lblno\_of\_days = Label(lblFrameLeft,text="SUB TOTAL :",font=("arial",10,"bold"),padx=2,pady=6)

        lblno\_of\_days.grid(row=8,column=0,sticky=W)

        txtno\_of\_days=ttk.Entry(lblFrameLeft,textvariable=self.var\_actual\_total,width=29,font=("arial",11,"bold"))

        txtno\_of\_days.grid(row=8,column=1)

*########## TOTAL COST ################*

        lbLid\_number = Label(lblFrameLeft,text="TOTAL COST :",font=("arial",10,"bold"),padx=2,pady=6)

        lbLid\_number.grid(row=9,column=0,sticky=W)

        txtid\_number = ttk.Entry(lblFrameLeft,textvariable=self.var\_total,width=29,font=("arial",11,"bold"))

        txtid\_number.grid(row=9,column=1)

*############# BILL BUTTON ##########*

        btn\_bill\_button = Button(lblFrameLeft,text="BILL",command=self.total,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_bill\_button.grid(row=10,column=0,padx=5,sticky=W)

*############## BUTTONS ##############*

        btn\_frame = Label(lblFrameLeft,bd=2,relief=RIDGE)

        btn\_frame.place(x=0,y=350,width=412,height=32)

        btn\_add = Button(btn\_frame,text="ADD",command=self.add\_data,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_add.grid(row=0,column=0,padx=5)

        btn\_update = Button(btn\_frame,text="UPDATE",command=self.update,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_update.grid(row=0,column=1,padx=5)

        btn\_delete = Button(btn\_frame,text="DELETE",command=self.dat\_Delete,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_delete.grid(row=0,column=2,padx=5)

        btn\_reset = Button(btn\_frame,text="RESET",command=self.reset,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_reset.grid(row=0,column=3,padx=5)

*############## RIGHT SIDE IMAGE ################*

        img3 = Image.open("room1.jpg")

        img3 = img3.resize((365,211),Image.ANTIALIAS)

        self.photoimg3=ImageTk.PhotoImage(img3)

        lblimg1 = Label(self.root,image=self.photoimg3,bd=4,relief=RIDGE)

        lblimg1.place(x=750,y=38,width=365,height=211)

*############## TABLE FRAME SEARCH ###############*

        table\_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="SEARCH AND VIEW DETAILS",font=("times new roman",12,"bold"),padx=2)

        table\_frame.place(x=435,y=248,width=680,height=201)

        lblsearchby = Label(table\_frame,text="SEARCH BY :",font=("arial",10,"bold"),bg="red",fg="white")

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

        self.search\_var = StringVar()

        combo\_search=ttk.Combobox(table\_frame,textvariable=self.search\_var,font=("arial",10,"bold"),width=12,state="readonly")

        combo\_search["value"]=("CONTACT","ROOM")

        combo\_search.current(0)

        combo\_search.grid(row=0,column=1,padx=4)

        self.txt\_search = StringVar()

        entry\_search = ttk.Entry(table\_frame,textvariable=self.txt\_search,width=29,font=("arial",11,"bold"))

        entry\_search.grid(row=0,column=2,padx=4)

        btn\_search = Button(table\_frame,text="SEARCH",command=self.search\_data,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_search.grid(row=0,column=3,padx=5)

        btn\_showall = Button(table\_frame,text="SHOW ALL!!",command=self.fetch\_data,font=("arial",10,"bold"),bg="blue",fg="white",width=10)

        btn\_showall.grid(row=0,column=4,padx=5)

*################ SHOW DATA TABLE ##############*

        details\_table = Label(table\_frame,bd=2,relief=RIDGE)

        details\_table.place(x=0,y=34,width=674,height=148)

        Scroll\_x=ttk.Scrollbar(details\_table,orient=HORIZONTAL)

        Scroll\_y=ttk.Scrollbar(details\_table,orient=VERTICAL)

        self.room\_table=ttk.Treeview(details\_table,columns=("CONTACT","CHECK\_IN","CHECK\_OUT","ROOM\_TYPE","ROOM\_AVAILABLE","MEAL","NO\_OF\_DAYS"),xscrollcommand=Scroll\_x.set,yscrollcommand=Scroll\_y.set)

        Scroll\_x.pack(side=BOTTOM,fill=X)

        Scroll\_y.pack(side=RIGHT,fill=Y)

        Scroll\_x.config(command=self.room\_table.xview)

        Scroll\_y.config(command=self.room\_table.yview)

        self.room\_table.heading("CONTACT",text="CONTACT")

        self.room\_table.heading("CHECK\_IN",text="CHECK IN")

        self.room\_table.heading("CHECK\_OUT",text="CHECK OUT")

        self.room\_table.heading("ROOM\_TYPE",text="ROOM TYPE")

        self.room\_table.heading("ROOM\_AVAILABLE",text="ROOM AVAILABLE")

        self.room\_table.heading("MEAL",text="MEAL")

        self.room\_table.heading("NO\_OF\_DAYS",text="NO. OF DAYS")

        self.room\_table["show"]="headings"

        self.room\_table.column("CONTACT",width=100)

        self.room\_table.column("CHECK\_IN",width=100)

        self.room\_table.column("CHECK\_OUT",width=100)

        self.room\_table.column("ROOM\_TYPE",width=100)

        self.room\_table.column("ROOM\_AVAILABLE",width=100)

        self.room\_table.column("MEAL",width=100)

        self.room\_table.column("NO\_OF\_DAYS",width=100)

        self.room\_table.pack(fill=BOTH,expand=1)

        self.room\_table.bind("<ButtonRelease-1>",self.get\_cursor)

        self.fetch\_data()

    def add\_data(self):

        if self.var\_contact.get()=="" or self.var\_check\_in.get()=="":

            messagebox.showerror("Error","ALL FIELDS ARE REQUIRED!!!",parent=self.root)

        else:

            try:

                conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

                my\_cursor = conn.cursor()

                my\_cursor.execute("insert into room values(%s,%s,%s,%s,%s,%s,%s)",(self.var\_contact.get(),self.var\_check\_in.get(),self.var\_check\_out.get(),self.var\_room\_type.get(),self.var\_room\_available.get(),self.var\_meal.get(),

        self.var\_no\_of\_days.get()))

                conn.commit()

                self.fetch\_data()

                conn.close()

                messagebox.showinfo("Success","Room booked!!!",parent=self.root)

            except Exception as es:

                messagebox.showwarning("Warning",f"Some thing went wrong:{str(es)}",parent=self.root)

    def fetch\_data(self):

        conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

        my\_cursor = conn.cursor()

        my\_cursor.execute("Select \* from room")

        rows = my\_cursor.fetchall()

        if len(rows)!= 0:

            self.room\_table.delete(\*self.room\_table.get\_children())

            for i in rows:

                self.room\_table.insert("",END,values=i)

            conn.commit()

        conn.close()

    def get\_cursor(self,event=""):

        cursor\_row = self.room\_table.focus()

        content = self.room\_table.item(cursor\_row)

        row = content["values"]

        self.var\_contact.set(row[0]),

        self.var\_check\_in.set(row[1]),

        self.var\_check\_out.set(row[2]),

        self.var\_room\_type.set(row[3]),

        self.var\_room\_available.set(row[4]),

        self.var\_meal.set(row[5]),

        self.var\_no\_of\_days.set(row[6])

    def update(self):

        if self.var\_contact.get()=="":

            messagebox.showerror("Error","Please enter mobile number",parent=self.root)

        else:

            conn = mysql.connector.connect(password=" ",database="ankur")

            my\_cursor = conn.cursor()

            my\_cursor.execute("update room set CHECK\_IN=%s,CHECK\_OUT=%s,ROOM\_TYPE=%s,ROOM\_AVAILABLE=%s,MEAL=%s,NO\_OF\_DAYS=%s where CONTACT=%s",(self.var\_check\_in.get(),self.var\_check\_out.get(),self.var\_room\_type.get(),self.var\_room\_available.get(),self.var\_meal.get(),self.var\_no\_of\_days.get(),self.var\_contact.get()))

            conn.commit()

            self.fetch\_data()

            conn.close()

            messagebox.showinfo("Update","Room details has been updated sucessfully",parent=self.root)

    def dat\_Delete(self):

        dat\_Delete = messagebox.askyesno("Hotel Management System,","Do you want to remove this customers room",parent=self.root)

        if dat\_Delete>0:

            conn = mysql.connector.connect(password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query = "delete from room where CONTACT=%s"

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

        else:

            if not dat\_Delete:

                return

        conn.commit()

        self.fetch\_data()

        conn.close()

    def reset(self):

        self.var\_contact.set(""),

        self.var\_check\_in.set(""),

        self.var\_check\_out.set(""),

        self.var\_room\_type.set(""),

        self.var\_room\_available.set(""),

        self.var\_meal.set(""),

        self.var\_no\_of\_days.set("")

        self.var\_paid\_tax.set("")

        self.var\_actual\_total.set("")

        self.var\_total.set("")

    def fetch\_contact(self):

        if self.var\_contact.get()=="":

            messagebox.showerror("Error","Please enter contact number",parent=self.root)

        else:

            conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query=("select NAME from customer where MOBILE=%s")

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

            row = my\_cursor.fetchone()

        if row==None:

            messagebox.showerror("Error","Number not found",parent=self.root)

        else:

            conn.commit()

            conn.close()

            showdataframe=Frame(self.root,bd=4,relief=RIDGE,padx=2)

            showdataframe.place(x=445,y=55,width=300,height=180)

*######## NAME ##############*

            lblname=Label(showdataframe,text="NAME:",font=("arial",10,"bold"))

            lblname.place(x=0,y=0)

            lbl\_data\_name = Label(showdataframe,text=row,font=("arial",10,"bold"))

            lbl\_data\_name.place(x=110,y=0)

*########### GENDER ###########*

            conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query=("select GENDER from customer where MOBILE=%s")

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

            row = my\_cursor.fetchone()

            lblgender=Label(showdataframe,text="GENDER:",font=("arial",10,"bold"))

            lblgender.place(x=0,y=30)

            lbl\_data\_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

            lbl\_data\_gender.place(x=110,y=30)

*############ EMAIL ################*

            conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query=("select EMAIL from customer where MOBILE=%s")

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

            row = my\_cursor.fetchone()

            lblgender=Label(showdataframe,text="EMAIL:",font=("arial",10,"bold"))

            lblgender.place(x=0,y=60)

            lbl\_data\_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

            lbl\_data\_gender.place(x=110,y=60)

*######## NATIONALITY ################*

            conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query=("select NATIONALITY from customer where MOBILE=%s")

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

            row = my\_cursor.fetchone()

            lblgender=Label(showdataframe,text="NATIONALITY:",font=("arial",10,"bold"))

            lblgender.place(x=0,y=90)

            lbl\_data\_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

            lbl\_data\_gender.place(x=110,y=90)

*######### ADDRESS ##############*

            conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

            my\_cursor = conn.cursor()

            query=("select ADDRESS from customer where MOBILE=%s")

            value = (self.var\_contact.get(),)

            my\_cursor.execute(query,value)

            row = my\_cursor.fetchone()

            lblgender=Label(showdataframe,text="ADDRESS:",font=("arial",10,"bold"))

            lblgender.place(x=0,y=120)

            lbl\_data\_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

            lbl\_data\_gender.place(x=110,y=120)

*############## SEARCH ####################*

    def search\_data(self):

        conn = mysql.connector.connect(host="localhost",username="root",password=" ",database="ankur")

        my\_cursor = conn.cursor()

        my\_cursor.execute("select \* from room where "+str(self.search\_var.get())+" LIKE '%"+str(self.txt\_search.get())+"%'")

        rows\_featch = my\_cursor.fetchall()

        if len(rows\_featch)!=0:

            self.room\_table.delete(\*self.room\_table.get\_children())

            for i in rows\_featch:

                self.room\_table.insert("",END,values=i)

            conn.commit()

        conn.close()

    def total(self):

        indate = self.var\_check\_in.get()

        outdate = self.var\_check\_out.get()

        indate=datetime.strptime(indate,"%d/%m/%Y")

        outdate=datetime.strptime(outdate,"%d/%m/%Y")

        self.var\_no\_of\_days.set(abs(outdate-indate).days)

        if (self.var\_room\_type.get()=="THREE BED"):

            q1 = float(300)

            q2 = float(700)

            q3 = float(self.var\_no\_of\_days.get())

            q4 = float(q1+q2)

            q5 = float(q3+q4)

            Tax = "RS."+str("%.2f"%((q5)\*0.1))

            ST = "RS."+str("%.2f"%((q5)))

            TT = "RS."+str("%.2f"%(q5+((q5)\*0.1)))

            self.var\_paid\_tax.set(Tax)

            self.var\_actual\_total.set(ST)

            self.var\_total.set(TT)

        elif (self.var\_room\_type.get()=="SINGLE BED"):

            q1 = float(300)

            q2 = float(700)

            q3 = float(self.var\_no\_of\_days.get())

            q4 = float(q1+q2)

            q5 = float(q3+q4)

            Tax = "RS."+str("%.2f"%((q5)\*0.1))

            ST = "RS."+str("%.2f"%((q5)))

            TT = "RS."+str("%.2f"%(q5+((q5)\*0.1)))

            self.var\_paid\_tax.set(Tax)

            self.var\_actual\_total.set(ST)

            self.var\_total.set(TT)

        elif (self.var\_room\_type.get()=="DOUBLE BED"):

            q1 = float(500)

            q2 = float(1000)

            q3 = float(self.var\_no\_of\_days.get())

            q4 = float(q1+q2)

            q5 = float(q3+q4)

            Tax = "RS."+str("%.2f"%((q5)\*0.1))

            ST = "RS."+str("%.2f"%((q5)))

            TT = "RS."+str("%.2f"%(q5+((q5)\*0.1)))

            self.var\_paid\_tax.set(Tax)

            self.var\_actual\_total.set(ST)

            self.var\_total.set(TT)

if \_\_name\_\_ == "\_\_main\_\_":

    root=Tk()

    Obj=Roombooking(root)

    root.mainloop()