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 Detailsroom:

    def \_\_init\_\_(self,root):

        self.root=root

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

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

*################ 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="NEW ROOM ADD",font=("times new roman",12,"bold"),padx=2)

        lblFrameLeft.place(x=5,y=40,width=460,height=350)

*############## FLOOR #################*

        lbl\_contact = Label(lblFrameLeft,text="Contact ",font=("arial",10,"bold"),padx=2,pady=6)

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

        self.var\_contact=StringVar()

        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)

*############## ROOM NUMBER #################*

        lbl\_check\_out= Label(lblFrameLeft,text="Check Out. ",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_check\_out.grid(row=2,column=0,sticky=W)

        self.var\_check\_out=StringVar()

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

        entry\_check\_out.grid(row=2,column=1,sticky=W)

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

        lbl\_room\_TYPE= Label(lblFrameLeft,text="Room Type. ",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_room\_TYPE.grid(row=3,column=0,sticky=W)

        self.var\_room\_type=StringVar()

        entry\_room\_TYPE = ttk.Entry(lblFrameLeft,textvariable=self.var\_room\_type,width=20,font=("arial",11,"bold"))

        entry\_room\_TYPE.grid(row=3,column=1,sticky=W)

*###################################################################################################################*

*############## FLOOR #################*

        lbl\_check\_in= Label(lblFrameLeft,text="Check In ",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_check\_in.grid(row=1,column=0,sticky=W)

        self.var\_check\_in=StringVar()

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

        entry\_check\_in.grid(row=1,column=1,sticky=W)

*############## ROOM NUMBER #################*

        lbl\_room\_no= Label(lblFrameLeft,text="Room No. ",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_room\_no.grid(row=4,column=0,sticky=W)

        self.var\_room\_no=StringVar()

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

        entry\_room\_no.grid(row=4,column=1,sticky=W)

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

*# lbl\_room\_TYPE= Label(lblFrameLeft,text="ROOM TYPE. ",font=("arial",10,"bold"),padx=2,pady=6)*

*# lbl\_room\_TYPE.grid(row=2,column=0,sticky=W)*

*# self.var\_room\_type=StringVar()*

*# entry\_room\_TYPE = ttk.Entry(lblFrameLeft,textvariable=self.var\_room\_type,width=20,font=("arial",11,"bold"))*

*# entry\_room\_TYPE.grid(row=2,column=1,sticky=W)*

*#####################################################################################################################*

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

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

        btn\_frame.place(x=15,y=200,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)

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

        table\_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="Show Room Details",font=("times new roman",12,"bold"),padx=2)

        table\_frame.place(x=500,y=40,width=680,height=350)

*############################3*

        lblsearchby = Label(table\_frame,text="Search By Contact No. :",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"]=("MOBILE")*

*# 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=1,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=2,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=3,padx=5)

*##########################33333333333*

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

        details\_table.place(x=0,y=34,width=674,height=350)

        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"),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="Student Contacts")

*# 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 NO",text="ROOM NO.")*

        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 NO",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\_room\_type.get()=="":

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

        else:

            try:

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

                my\_cursor = conn.cursor()

                my\_cursor.execute("insert into room values(%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\_no.get()))

                conn.commit()

                self.fetch\_data()

                conn.close()

                messagebox.showinfo("Success","New Room added succeccfully!!!",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="nopassw",database="ankur")

        my\_cursor = conn.cursor()

        my\_cursor.execute("Select MOBILE from customer")

        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 search\_data(self):

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

        my\_cursor = conn.cursor()

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

*# 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 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\_no.set(row[4])

    def update(self):

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

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

        else:

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

            my\_cursor = conn.cursor()

            my\_cursor.execute("update room set CONTACT=%s, CHECK\_IN=%s, CHECK\_OUT=%s,  ROOM\_TYPE=%s where ROOM=%s",(self.var\_contact.get(),self.var\_check\_in.get(),self.var\_check\_out.get(),self.var\_room\_type.get(),self.var\_room\_no.get()))

            conn.commit()

            self.fetch\_data()

            conn.close()

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

    def dat\_Delete(self):

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

        if dat\_Delete>0:

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

            my\_cursor = conn.cursor()

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

            value = (self.var\_room\_no.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\_room\_no.set(""),

        self.var\_room\_type.set("")

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

    root=Tk()

    Obj=Detailsroom(root)

    root.mainloop()