from asyncore import read

from faulthandler import disable

from sre\_parse import State

from tkinter import\*

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

from tkinter import ttk

import random

import mysql.connector

from tkinter import messagebox

from setuptools import Command

class Cust\_Win:

    def \_\_init\_\_(self,root):

        self.root=root

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

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

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

        self.var\_ref = StringVar()

        x = random.randint(1000,9999)

        self.var\_ref.set(str(x))

        self.var\_cust\_name = StringVar()

        self.var\_cust\_mother = StringVar()

        self.var\_cust\_gender = StringVar()

        self.var\_cust\_post = StringVar()

        self.var\_cust\_mobile = StringVar()

        self.var\_cust\_email = StringVar()

        self.var\_cust\_nationality = StringVar()

        self.var\_cust\_id\_proff = StringVar()

        self.var\_cust\_id\_number = StringVar()

        self.var\_cust\_address = StringVar()

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

        lbl\_title = Label(self.root,text="ADD STUDENT 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="STUDENT DETAILS",font=("times new roman",12,"bold"),padx=2)

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

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

*############## REFRENCE NO. #################*

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

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

        entry\_ref = ttk.Entry(lblFrameLeft,textvariable=self.var\_ref,width=29,font=("arial",11,"bold"),state=DISABLED)

        entry\_ref.grid(row=0,column=1)

*############## STUDENT NAME ##############*

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

        lbl\_cust\_name.grid(row=1,column=0,sticky=W)

        entry\_name = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_name,width=29,font=("arial",11,"bold"))

        entry\_name.grid(row=1,column=1)

*############# MOTHERS NAME ###############*

        lbl\_cust\_mname = Label(lblFrameLeft,text="MOTHER NAME :",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_cust\_mname.grid(row=2,column=0,sticky=W)

        entry\_mname = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_mother,width=29,font=("arial",11,"bold"))

        entry\_mname.grid(row=2,column=1)

*############ GENDER BOX ###############*

        lbl\_gender = Label(lblFrameLeft,font=("arial",10,"bold"),text="GENDER :",padx=2,pady=6)

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

        combo\_gender=ttk.Combobox(lblFrameLeft,textvariable=self.var\_cust\_gender,font=("arial",10,"bold"),width=31,state="readonly")

        combo\_gender["value"]=("MALE","FEMALE","OTHERS")

        combo\_gender.current(0)

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

*########### POST CODE ###############*

        lbl\_cust\_post = Label(lblFrameLeft,text="POST CODE :",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_cust\_post.grid(row=4,column=0,sticky=W)

        entry\_post = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_post,width=29,font=("arial",11,"bold"))

        entry\_post.grid(row=4,column=1)

*########## MOBILE NUMBER ##############*

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

        lbl\_cust\_mob.grid(row=5,column=0,sticky=W)

        entry\_mob = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_mobile,width=29,font=("arial",11,"bold"))

        entry\_mob.grid(row=5,column=1)

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

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

        lbl\_cust\_email.grid(row=6,column=0,sticky=W)

        entry\_email = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_email,width=29,font=("arial",11,"bold"))

        entry\_email.grid(row=6,column=1)

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

        lblNationality = Label(lblFrameLeft,font=("arial",10,"bold"),text="NATIONALITY :",padx=2,pady=6)

        lblNationality.grid(row=7,column=0,sticky=W)

        combo\_nationality=ttk.Combobox(lblFrameLeft,textvariable=self.var\_cust\_nationality,font=("arial",10,"bold"),width=31,state="readonly")

        combo\_nationality["value"]=("INDIAN","AMERICAN","BRITISH")

        combo\_nationality.current(0)

        combo\_nationality.grid(row=7,column=1)

*############ IDPROFF TYPE #############*

        lblIdproff = Label(lblFrameLeft,font=("arial",10,"bold"),text="ID PROFF TYPE :",padx=2,pady=6)

        lblIdproff.grid(row=8,column=0,sticky=W)

        combo\_idproff=ttk.Combobox(lblFrameLeft,textvariable=self.var\_cust\_id\_proff,font=("arial",10,"bold"),width=31,state="readonly")

        combo\_idproff["value"]=("AADHAR CARD","PASSPORT","DRIVING LICIENCE","PAN CARD")

        combo\_idproff.current(0)

        combo\_idproff.grid(row=8,column=1)

*############ ID NUMBER #################*

        lbl\_cust\_idno = Label(lblFrameLeft,text="ID NUMBER :",font=("arial",10,"bold"),padx=2,pady=6)

        lbl\_cust\_idno.grid(row=9,column=0,sticky=W)

        entry\_idno = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_id\_number,width=29,font=("arial",11,"bold"))

        entry\_idno.grid(row=9,column=1)

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

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

        lbl\_cust\_addr.grid(row=10,column=0,sticky=W)

        entry\_addr = ttk.Entry(lblFrameLeft,textvariable=self.var\_cust\_address,width=29,font=("arial",11,"bold"))

        entry\_addr.grid(row=10,column=1)

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

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

        btn\_frame.place(x=0,y=352,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.data\_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="SEARCH AND VIEW DETAILS",font=("times new roman",12,"bold"),padx=2)

        table\_frame.place(x=435,y=40,width=680,height=410)

        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"]=("MOBILE","REF\_NO")

        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=350)

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

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

        self.Cust\_details\_table=ttk.Treeview(details\_table,columns=("REF\_NO","NAME","MOTHER","GENDER","POST","MOBILE","EMAIL","NATIONALITY","ID\_PROFF","ID\_NO","ADDRESS"),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.Cust\_details\_table.xview)

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

        self.Cust\_details\_table.heading("REF\_NO",text="REFER NO.")

        self.Cust\_details\_table.heading("NAME",text="NAME")

        self.Cust\_details\_table.heading("MOTHER",text="MOTHER")

        self.Cust\_details\_table.heading("GENDER",text="GENDER")

        self.Cust\_details\_table.heading("POST",text="POST")

        self.Cust\_details\_table.heading("MOBILE",text="MOBILE")

        self.Cust\_details\_table.heading("EMAIL",text="EMAIL")

        self.Cust\_details\_table.heading("NATIONALITY",text="NATIONALITY")

        self.Cust\_details\_table.heading("ID\_PROFF",text="ID PROFF.")

        self.Cust\_details\_table.heading("ID\_NO",text="ID NO.")

        self.Cust\_details\_table.heading("ADDRESS",text="ADDRESS")

        self.Cust\_details\_table["show"]="headings"

        self.Cust\_details\_table.column("REF\_NO",width=100)

        self.Cust\_details\_table.column("NAME",width=100)

        self.Cust\_details\_table.column("MOTHER",width=100)

        self.Cust\_details\_table.column("GENDER",width=100)

        self.Cust\_details\_table.column("POST",width=100)

        self.Cust\_details\_table.column("MOBILE",width=100)

        self.Cust\_details\_table.column("EMAIL",width=100)

        self.Cust\_details\_table.column("NATIONALITY",width=100)

        self.Cust\_details\_table.column("ID\_PROFF",width=100)

        self.Cust\_details\_table.column("ID\_NO",width=100)

        self.Cust\_details\_table.column("ADDRESS",width=100)

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

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

        self.fetch\_data()

    def add\_data(self):

        if self.var\_cust\_mobile.get()=="" or self.var\_cust\_mother.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 customer values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(self.var\_ref.get(),self.var\_cust\_name.get(),self.var\_cust\_mother.get(),self.var\_cust\_gender.get(),self.var\_cust\_post.get(),self.var\_cust\_mobile.get(),self.var\_cust\_email.get(),self.var\_cust\_nationality.get(),self.var\_cust\_id\_proff.get(),self.var\_cust\_id\_number.get(),self.var\_cust\_address.get()))

                conn.commit()

                self.fetch\_data()

                conn.close()

                messagebox.showinfo("Success","Customer has been added",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 \* from customer")

        rows = my\_cursor.fetchall()

        if len(rows)!= 0:

            self.Cust\_details\_table.delete(\*self.Cust\_details\_table.get\_children())

            for i in rows:

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

            conn.commit()

        conn.close()

    def get\_cursor(self,event=""):

        cursor\_row = self.Cust\_details\_table.focus()

        content = self.Cust\_details\_table.item(cursor\_row)

        row = content["values"]

        self.var\_ref.set(row[0]),

        self.var\_cust\_name.set(row[1]),

        self.var\_cust\_mother.set(row[2]),

        self.var\_cust\_gender.set(row[3]),

        self.var\_cust\_post.set(row[4]),

        self.var\_cust\_mobile.set(row[5]),

        self.var\_cust\_email.set(row[6]),

        self.var\_cust\_nationality.set(row[7]),

        self.var\_cust\_id\_proff.set(row[8]),

        self.var\_cust\_id\_number.set(row[9]),

        self.var\_cust\_address.set(row[10])

    def update(self):

        if self.var\_cust\_mobile.get()=="":

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

        else:

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

            my\_cursor = conn.cursor()

            my\_cursor.execute("update customer set NAME=%s,MOTHER=%s,GENDER=%s,POST=%s,MOBILE=%s,EMAIL=%s,NATIONALITY=%s,ID\_PROFF=%s,ID\_NO=%s,ADDRESS=%s where REF\_NO=%s",(self.var\_cust\_name.get(),self.var\_cust\_mother.get(),self.var\_cust\_gender.get(),self.var\_cust\_post.get(),self.var\_cust\_mobile.get(),self.var\_cust\_email.get(),self.var\_cust\_nationality.get(),self.var\_cust\_id\_proff.get(),self.var\_cust\_id\_number.get(),self.var\_cust\_address.get(),self.var\_ref.get()))

            conn.commit()

            self.fetch\_data()

            conn.close()

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

    def dat\_Delete(self):

        dat\_Delete = messagebox.askyesno("Hotel Management System,","Do you want to delete this customer",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 customer where REF\_NO=%s"

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

            my\_cursor.execute(query,value)

*# messagebox.showinfo("Update","Customer details has been deleted sucessfully",parent=self.root)*

        else:

            if not dat\_Delete:

                return

        conn.commit()

        self.fetch\_data()

        conn.close()

    def data\_reset(self):

*#self.var\_ref.set(""),*

        self.var\_cust\_name.set(""),

        self.var\_cust\_mother.set(""),

*#self.var\_cust\_gender.set(""),*

        self.var\_cust\_post.set(""),

        self.var\_cust\_mobile.set(""),

        self.var\_cust\_email.set(""),

*#self.var\_cust\_nationality.set(""),*

*#self.var\_cust\_id\_proff.set(""),*

        self.var\_cust\_id\_number.set(""),

        self.var\_cust\_address.set("")

        x = random.randint(1000,9999)

        self.var\_ref.set(str(x))

    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 "+str(self.search\_var.get())+" LIKE '%"+str(self.txt\_search.get())+"%'")

        rows\_featch = my\_cursor.fetchall()

        if len(rows\_featch)!=0:

            self.Cust\_details\_table.delete(\*self.Cust\_details\_table.get\_children())

            for i in rows\_featch:

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

            conn.commit()

        conn.close()

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

    root=Tk()

    Obj=Cust\_Win(root)

    root.mainloop()