**HOTEL MANAGEMENT SYSTEM**

**Login Page Code:**

from tkinter import\*

from tkinter import ttk

from PIL import Image,ImageTk

from tkinter import messagebox

import mysql.connector

from hotel import HotelManagementSystem

from customer import Customer

from room import RoomBooking

from details import DetailsRoom

def main():

    win=Tk()

    app=Login\_Window(win)

    win.mainloop()

class Login\_Window:

    def \_\_init\_\_(self,root):

        self.root=root

        self.root.title("Login")

        self.root.geometry("1550x800+0+0")

        self.bg=ImageTk.PhotoImage(file="C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\bgphoto.jpg")

        lbl\_bg=Label(self.root,image=self.bg)

        lbl\_bg.place(x=0,y=0,relwidth=1,relheight=1)

        frame=Frame(self.root,bg="black")

        frame.place(x=610,y=170,width=340,height=450)

        img1=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img1=img1.resize((100,100),Image.BICUBIC)

        self.photoimage1=ImageTk.PhotoImage(img1)

        lblimg1=Label(image=self.photoimage1,bg="black",borderwidth=0)

        lblimg1.place(x=730,y=175,width=100,height=100)

        get\_str=Label(frame,text="Get Started",font=("times new roman",20,"bold"),fg="white",bg="black")

        get\_str.place(x=95,y=100)

        #label

        username\_lbl=Label(frame,text="Username",font=("times new roman",15,"bold"),fg="white",bg="black")

        username\_lbl.place(x=70,y=155)

        self.txtuser=ttk.Entry(frame,font=("times new roman",15,"bold"))

        self.txtuser.place(x=40,y=180,width=270)

        password\_lbl=Label(frame,text="Password",font=("times new roman",15,"bold"),fg="white",bg="black")

        password\_lbl.place(x=70,y=225)

        self.txtpass=ttk.Entry(frame,show="\*",font=("times new roman",15,"bold"))

        self.txtpass.place(x=40,y=250,width=270)

        #Icon Images

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((25,25),Image.BICUBIC)

        self.photoimage2=ImageTk.PhotoImage(img2)

        lblimg1=Label(image=self.photoimage2,bg="black",borderwidth=0)

        lblimg1.place(x=650,y=323,width=25,height=25)

        img3=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img3=img3.resize((25,25),Image.BICUBIC)

        self.photoimage3=ImageTk.PhotoImage(img3)

        lblimg1=Label(image=self.photoimage3,bg="black",borderwidth=0)

        lblimg1.place(x=650,y=395,width=25,height=25)

        #Login Button

        loginbtn=Button(frame,command=self.login,text="Login",font=("times new roman",15,"bold"),bd=3,relief=RIDGE,fg="white",bg="red",activeforeground="white",activebackground="red")

        loginbtn.place(x=110,y=300,width=120,height=35)

        #Register Button

        registerbtn=Button(frame,text="New User Register",command=self.register\_window,font=("times new roman",10,"bold"),borderwidth=0,fg="white",bg="black",activeforeground="white",activebackground="black")

        registerbtn.place(x=15,y=350,width=160)

        #Forgot Password Button

        forgetbtn=Button(frame,text="Forget Password",command=self.forgot\_password\_window,font=("times new roman",10,"bold"),borderwidth=0,fg="white",bg="black",activeforeground="white",activebackground="black")

        forgetbtn.place(x=10,y=370,width=160)

    def register\_window(self):

        self.new\_window=Toplevel(self.root)

        self.app=Register(self.new\_window)

    def login(self):

        if self.txtuser.get()=="" or self.txtpass.get()=="":

            messagebox.showerror("Error","All fields required to fill")

        elif self.txtuser.get()=="tanisha" and self.txtpass.get()=="tan15":

            messagebox.showinfo("Success","Welcome to Hotel Management System")

        else:

            conn=mysql.connector.connect(host="localhost",user="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            my\_cursor.execute("select \* from register where email=%s and password=%s",(

                                                                                       self.txtuser.get(),

                                                                                       self.txtpass.get()

                                                                                       ))

            row=my\_cursor.fetchone()

            if row==None:

                messagebox.showerror("Error","Invalid Username and Password")

            else:

                open\_main=messagebox.askyesno("YesNo","Access only admin")

                if open\_main>0:

                    self.new\_window=Toplevel(self.root)

                    self.app=HotelManagementSystem(self.new\_window)

                else:

                    if not open\_main:

                        return

            conn.commit()

            conn.close()

    #reset password

    def reset\_pass(self):

        if self.combo\_security\_q.get()=="Select":

            messagebox.showerror("Error","Select the security question",parent=self.root2)

        elif self.txt\_security.get()=="":

            messagebox.showerror("Error","Please enter the answer",parent=self.root2)

        elif self.txt\_newpass.get()=="":

            messagebox.showerror("Error","Please enter the new password",parent=self.root2)

        else:

            conn=mysql.connector.connect(host="localhost",user="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query=("select \* from register where email=%s and securityQ=%s and securityA=%s")

            value=(self.txtuser.get(),self.combo\_security\_q.get(),self.txt\_security.get(),)

            my\_cursor.execute(query,value)

            row=my\_cursor.fetchone()

            if row==None:

                messagebox.showerror("Error","Please enter the correct answer",parent=self.root2)

            else:

                query=("update register set password=%s where email=%s")

                value=(self.txt\_newpass.get(),self.txtuser.get())

                my\_cursor.execute(query,value)

                conn.commit()

                conn.close()

                messagebox.showinfo("Info","Your password has been reset.Please login with new password",parent=self.root2)

                self.root2.destroy()

    #forget password window

    def forgot\_password\_window(self):

        if self.txtuser.get()=="":

            messagebox.showerror("Error","Please enter the email to reset password")

        else:

            conn=mysql.connector.connect(host="localhost",user="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query=("select \* from register where email=%s")

            value=(self.txtuser.get(),)

            my\_cursor.execute(query,value)

            row=my\_cursor.fetchone()

            if row==None:

                messagebox.showerror("Error","Please enter the valid username")

            else:

                conn.close()

                self.root2=Toplevel()

                self.root2.title("Forgot Password")

                self.root2.geometry("340x450+610+170")

                l=Label(self.root2,text="Forget Password",font=("times new roman",20,"bold"),fg="red",bg="white")

                l.place(x=0,y=10,relwidth=1)

                security\_q=Label(self.root2,text="Select Security Question",font=("times new roman",15,"bold"),bg="white",fg="black")

                security\_q.place(x=50,y=80)

                self.combo\_security\_q=ttk.Combobox(self.root2,font=("times new roman",15,"bold"),state="readonly")

                self.combo\_security\_q["values"]=("Select","Your Birth Place","Your Pet Name","Your BestFriend Name")

                self.combo\_security\_q.place(x=50,y=110,width=250)

                self.combo\_security\_q.current(0)

                security\_a=Label(self.root2,text="Security Answer",font=("times new roman",15,"bold"),bg="white",fg="black")

                security\_a.place(x=50,y=150)

                self.txt\_security=ttk.Entry(self.root2,font=("times new roman",15))

                self.txt\_security.place(x=50,y=180,width=250)

                new\_password=Label(self.root2,text="New Password",font=("times new roman",15,"bold"),bg="white",fg="black")

                new\_password.place(x=50,y=220)

                self.txt\_newpass=ttk.Entry(self.root2,font=("times new roman",15))

                self.txt\_newpass.place(x=50,y=250,width=250)

                btn=Button(self.root2,text="Reset",command=self.reset\_pass,font=("times new roman",15,"bold"),fg="white",bg="green")

                btn.place(x=100,y=290)

class Register:

    def \_\_init\_\_(self,root):

        self.root=root

        self.root.title("Register")

        self.root.geometry("1600x900+0+0")

        #variables

        self.var\_fname=StringVar()

        self.var\_lname=StringVar()

        self.var\_contact=StringVar()

        self.var\_email=StringVar()

        self.var\_securityq=StringVar()

        self.var\_securitya=StringVar()

        self.var\_password=StringVar()

        self.var\_confirmpassword=StringVar()

        #Bg image

        self.bg=ImageTk.PhotoImage(file="C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\bgphoto.jpg")

        bg\_lbl=Label(self.root,image=self.bg)

        bg\_lbl.place(x=0,y=0,relwidth=1,relheight=1)

        #left image

        self.bg1=ImageTk.PhotoImage(file="C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\coverphoto.jpg")

        left\_lbl=Label(self.root,image=self.bg1)

        left\_lbl.place(x=50,y=100,width=470,height=550)

        #main frame

        frame=Frame(self.root,bg="white")

        frame.place(x=520,y=100,width=800,height=550)

        register\_lbl=Label(frame,text="REGISTER HERE",font=("times new roman",20,"bold"),fg="darkgreen",bg="white")

        register\_lbl.place(x=20,y=20)

        #Labels and Entries

        #row 1

        fname=Label(frame,text="First Name",font=("times new roman",15,"bold"),bg="white")

        fname.place(x=50,y=100)

        self.fname\_entry=ttk.Entry(frame,textvariable=self.var\_fname,font=("times new roman",15,"bold"))

        self.fname\_entry.place(x=50,y=130,width=250)

        l\_name=Label(frame,text="Last Name",font=("times new roman",15,"bold"),bg="white",fg="black")

        l\_name.place(x=370,y=100)

        self.txt\_lname=ttk.Entry(frame,textvariable=self.var\_lname,font=("times new roman",15))

        self.txt\_lname.place(x=370,y=130,width=250)

        #row2

        contact=Label(frame,text="Contact No.",font=("times new roman",15,"bold"),bg="white",fg="black")

        contact.place(x=50,y=170)

        self.txt\_contact=ttk.Entry(frame,textvariable=self.var\_contact,font=("times new roman",15))

        self.txt\_contact.place(x=50,y=200,width=250)

        email=Label(frame,text="Email",font=("times new roman",15,"bold"),bg="white",fg="black")

        email.place(x=370,y=170)

        self.txt\_email=ttk.Entry(frame,textvariable=self.var\_email,font=("times new roman",15))

        self.txt\_email.place(x=370,y=200,width=250)

        #row3

        security\_q=Label(frame,text="Select Security Question",font=("times new roman",15,"bold"),bg="white",fg="black")

        security\_q.place(x=50,y=240)

        self.combo\_security\_q=ttk.Combobox(frame,textvariable=self.var\_securityq,font=("times new roman",15,"bold"),state="readonly")

        self.combo\_security\_q["values"]=("Select","Your Birth Place","Your Pet Name","Your BestFriend Name")

        self.combo\_security\_q.place(x=50,y=270,width=250)

        self.combo\_security\_q.current(0)

        security\_a=Label(frame,text="Security Answer",font=("times new roman",15,"bold"),bg="white",fg="black")

        security\_a.place(x=370,y=240)

        self.txt\_security=ttk.Entry(frame,textvariable=self.var\_securitya,font=("times new roman",15))

        self.txt\_security.place(x=370,y=270,width=250)

        #row4

        password=Label(frame,text="Password",font=("times new roman",15,"bold"),bg="white",fg="black")

        password.place(x=50,y=310)

        self.txt\_password=ttk.Entry(frame,textvariable=self.var\_password,font=("times new roman",15))

        self.txt\_password.place(x=50,y=340,width=250)

        confirm\_password=Label(frame,text="Confirm Password",font=("times new roman",15,"bold"),bg="white",fg="black")

        confirm\_password.place(x=370,y=310)

        self.txt\_confirm\_password=ttk.Entry(frame,textvariable=self.var\_confirmpassword,font=("times new roman",15))

        self.txt\_confirm\_password.place(x=370,y=340,width=250)

        #Check Button

        self.var\_check=IntVar()

        self.checkbtn=Checkbutton(frame,variable=self.var\_check,text="I Agree the Terms and Conditions",font=("times new roman",12,"bold"),onvalue=1,offvalue=0)

        self.checkbtn.place(x=50,y=380)

        #buttons

        b1=Button(frame,command=self.register\_data,text="Register",bd=3,relief=RIDGE,cursor="hand2",font=("times new roman",15,"bold"),fg="white",bg="blue",activebackground="blue")

        b1.place(x=10,y=470,width=200)

        b1=Button(frame,command=self.return\_login,text="Login",bd=3,relief=RIDGE,cursor="hand2",font=("times new roman",15,"bold"),fg="white",bg="green",activebackground="green")

        b1.place(x=330,y=470,width=200)

    #function declaration

    def register\_data(self):

        if self.var\_fname.get()=="" or self.var\_email.get()=="" or self.var\_securityq.get()=="Select":

            messagebox.showerror("Error","All fields required")

        elif self.var\_password.get()!=self.var\_confirmpassword.get():

            messagebox.showerror("Error","Password and Confirm Password must be same")

        elif self.var\_check.get()==0:

            messagebox.showerror("Error","Please agree our terms and condition")

        else:

            conn=mysql.connector.connect(host="localhost",user="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query=("select \* from register where email=%s")

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

            my\_cursor.execute(query,value)

            row=my\_cursor.fetchone()

            if row!=None:

                messagebox.showerror("Error","User already exist,please try another email")

            else:

                my\_cursor.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s)",(

                                                                                       self.var\_fname.get(),

                                                                                       self.var\_lname.get(),

                                                                                       self.var\_contact.get(),

                                                                                       self.var\_email.get(),

                                                                                       self.var\_securityq.get(),

                                                                                       self.var\_securitya.get(),

                                                                                       self.var\_password.get()

                                                                                         ))

            conn.commit()

            conn.close()

            messagebox.showinfo("Success","Register Successfull")

    def return\_login(self):

        self.root.destroy()

class HotelManagementSystem:

    def \_\_init\_\_(self,root):

        self.root=root

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

        self.root.geometry("1550x800+0+0")

        # =======================cover photo========================================

        img1=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\coverphoto.jpg")

        img1=img1.resize((1550,140),Image.BICUBIC)

        self.photoimg1=ImageTk.PhotoImage(img1)

        lblimg=Label(self.root,image=self.photoimg1,bd=4,relief=RIDGE)

        lblimg.place(x=0,y=0,width=1550,height=140)

        # =====================logo============================================

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((230,140),Image.BICUBIC)

        self.photoimg2=ImageTk.PhotoImage(img2)

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

        lblimg.place(x=0,y=0,width=230,height=140)

        # ====================title==================================================

        lbl\_title=Label(self.root,text="HOTEL MANAGEMENT SYSTEM",font=("times new roman",40,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=140,width=1550,height=50)

        # =================main frame==================================================

        main\_frame=Frame(self.root,bd=4,relief=RIDGE)

        main\_frame.place(x=0,y=190,width=1550,height=620)

        # ====================menu=================================================

        lbl\_menu=Label(main\_frame,text="MENU",font=("times new roman",20,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_menu.place(x=0,y=0,width=230)

        # ===================button frame============================================

        btn\_frame=Frame(main\_frame,bd=4,relief=RIDGE)

        btn\_frame.place(x=0,y=35,width=228,height=190)

        # ====================buttons==================================================

        cust\_btn=Button(btn\_frame,text="CUSTOMER",command=self.cust\_details,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        cust\_btn.grid(row=0,column=0,pady=1)

        room\_btn=Button(btn\_frame,text="ROOM",command=self.room\_booking,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        room\_btn.grid(row=1,column=0,pady=1)

        details\_btn=Button(btn\_frame,text="DETAILS",command=self.room\_details,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        details\_btn.grid(row=2,column=0,pady=1)

        report\_btn=Button(btn\_frame,text="REPORT",width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        report\_btn.grid(row=3,column=0,pady=1)

        logout\_btn=Button(btn\_frame,text="LOGOUT",command=self.logout,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        logout\_btn.grid(row=4,column=0,pady=1)

        # ========================right side image======================================

        img3=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\bgphoto.jpg")

        img3=img3.resize((1310,590),Image.BICUBIC)

        self.photoimg3=ImageTk.PhotoImage(img3)

        lblimg1=Label(main\_frame,image=self.photoimg3,bd=4,relief=RIDGE)

        lblimg1.place(x=225,y=0,width=1310,height=590)

        # ===========================down images=======================================

        img4=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\fillerphoto1.jpg")

        img4=img4.resize((230,210),Image.BICUBIC)

        self.photoimg4=ImageTk.PhotoImage(img4)

        lblimg1=Label(main\_frame,image=self.photoimg4,bd=4,relief=RIDGE)

        lblimg1.place(x=0,y=225,width=230,height=210)

        img5=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\fillerphoto2.jpg")

        img5=img5.resize((230,190),Image.BICUBIC)

        self.photoimg5=ImageTk.PhotoImage(img5)

        lblimg1=Label(main\_frame,image=self.photoimg5,bd=4,relief=RIDGE)

        lblimg1.place(x=0,y=420,width=230,height=190)

    def cust\_details(self):

        self.new\_window=Toplevel(self.root)

        self.app=Customer(self.new\_window)

    def room\_booking(self):

        self.new\_window=Toplevel(self.root)

        self.app=RoomBooking(self.new\_window)

    def room\_details(self):

        self.new\_window=Toplevel(self.root)

        self.app=DetailsRoom(self.new\_window)

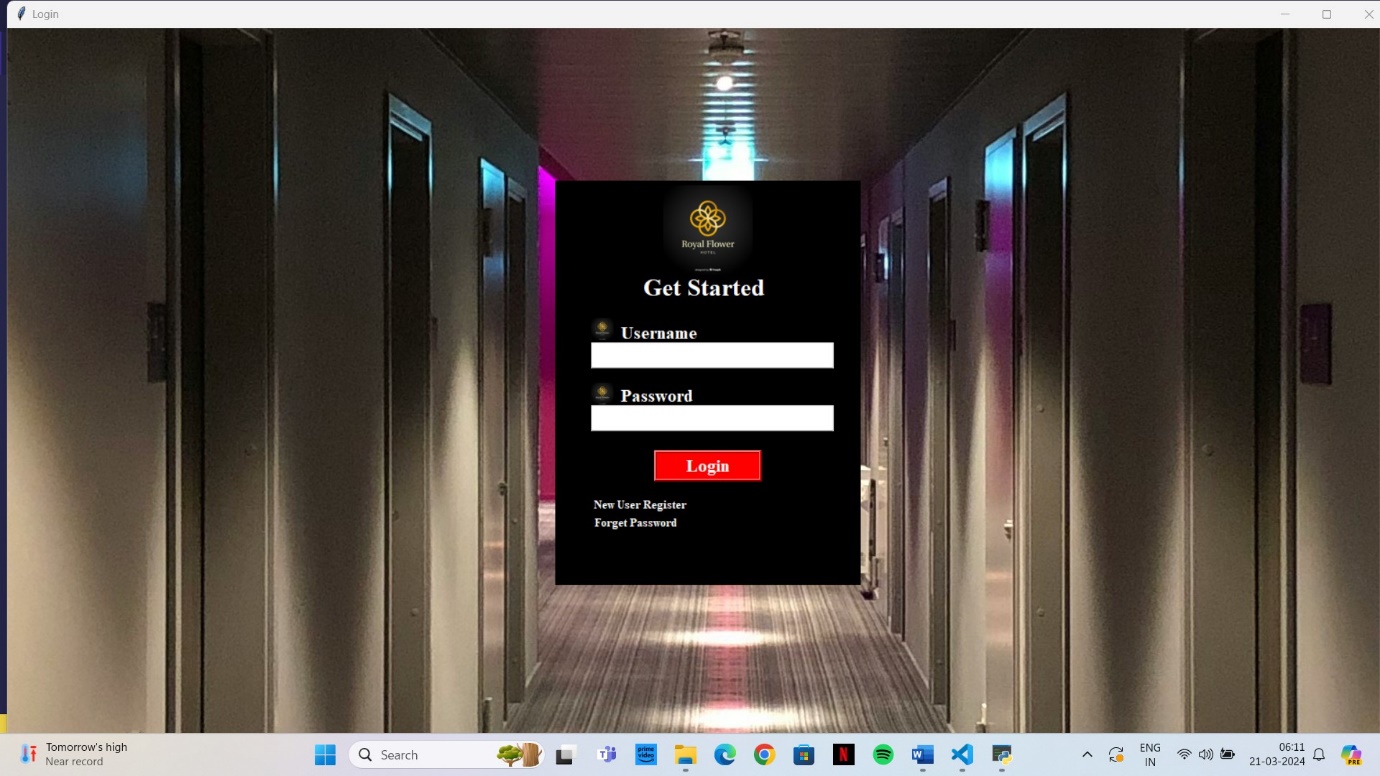
    def logout(self):

        self.root.destroy()

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

    main()

**Login Page Output:**



**Register Page Code:**

from tkinter import\*

from tkinter import ttk

from PIL import Image,ImageTk

from tkinter import messagebox

import mysql.connector

class Register:

    def \_\_init\_\_(self,root):

        self.root=root

        self.root.title("Register")

        self.root.geometry("1600x900+0+0")

        #variables

        self.var\_fname=StringVar()

        self.var\_lname=StringVar()

        self.var\_contact=StringVar()

        self.var\_email=StringVar()

        self.var\_securityq=StringVar()

        self.var\_securitya=StringVar()

        self.var\_password=StringVar()

        self.var\_confirmpassword=StringVar()

        #Bg image

        self.bg=ImageTk.PhotoImage(file="C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\bgphoto.jpg")

        bg\_lbl=Label(self.root,image=self.bg)

        bg\_lbl.place(x=0,y=0,relwidth=1,relheight=1)

        #left image

        self.bg1=ImageTk.PhotoImage(file="C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\coverphoto.jpg")

        left\_lbl=Label(self.root,image=self.bg1)

        left\_lbl.place(x=50,y=100,width=470,height=550)

        #main frame

        frame=Frame(self.root,bg="white")

        frame.place(x=520,y=100,width=800,height=550)

        register\_lbl=Label(frame,text="REGISTER HERE",font=("times new roman",20,"bold"),fg="darkgreen",bg="white")

        register\_lbl.place(x=20,y=20)

        #Labels and Entries

        #row 1

        fname=Label(frame,text="First Name",font=("times new roman",15,"bold"),bg="white")

        fname.place(x=50,y=100)

        self.fname\_entry=ttk.Entry(frame,textvariable=self.var\_fname,font=("times new roman",15,"bold"))

        self.fname\_entry.place(x=50,y=130,width=250)

        l\_name=Label(frame,text="Last Name",font=("times new roman",15,"bold"),bg="white",fg="black")

        l\_name.place(x=370,y=100)

        self.txt\_lname=ttk.Entry(frame,textvariable=self.var\_lname,font=("times new roman",15))

        self.txt\_lname.place(x=370,y=130,width=250)

        #row2

        contact=Label(frame,text="Contact No.",font=("times new roman",15,"bold"),bg="white",fg="black")

        contact.place(x=50,y=170)

        self.txt\_contact=ttk.Entry(frame,textvariable=self.var\_contact,font=("times new roman",15))

        self.txt\_contact.place(x=50,y=200,width=250)

        email=Label(frame,text="Email",font=("times new roman",15,"bold"),bg="white",fg="black")

        email.place(x=370,y=170)

        self.txt\_email=ttk.Entry(frame,textvariable=self.var\_email,font=("times new roman",15))

        self.txt\_email.place(x=370,y=200,width=250)

        #row3

        security\_q=Label(frame,text="Select Security Question",font=("times new roman",15,"bold"),bg="white",fg="black")

        security\_q.place(x=50,y=240)

        self.combo\_security\_q=ttk.Combobox(frame,textvariable=self.var\_securityq,font=("times new roman",15,"bold"),state="readonly")

        self.combo\_security\_q["values"]=("Select","Your Birth Place","Your Pet Name","Your BestFriend Name")

        self.combo\_security\_q.place(x=50,y=270,width=250)

        self.combo\_security\_q.current(0)

        security\_a=Label(frame,text="Security Answer",font=("times new roman",15,"bold"),bg="white",fg="black")

        security\_a.place(x=370,y=240)

        self.txt\_security=ttk.Entry(frame,textvariable=self.var\_securitya,font=("times new roman",15))

        self.txt\_security.place(x=370,y=270,width=250)

        #row4

        password=Label(frame,text="Password",font=("times new roman",15,"bold"),bg="white",fg="black")

        password.place(x=50,y=310)

        self.txt\_password=ttk.Entry(frame,textvariable=self.var\_password,font=("times new roman",15))

        self.txt\_password.place(x=50,y=340,width=250)

        confirm\_password=Label(frame,text="Confirm Password",font=("times new roman",15,"bold"),bg="white",fg="black")

        confirm\_password.place(x=370,y=310)

        self.txt\_confirm\_password=ttk.Entry(frame,textvariable=self.var\_confirmpassword,font=("times new roman",15))

        self.txt\_confirm\_password.place(x=370,y=340,width=250)

        #Check Button

        self.var\_check=IntVar()

        self.checkbtn=Checkbutton(frame,variable=self.var\_check,text="I Agree the Terms and Conditions",font=("times new roman",12,"bold"),onvalue=1,offvalue=0)

        self.checkbtn.place(x=50,y=380)

        #buttons

        b1=Button(frame,command=self.register\_data,text="Register",bd=3,relief=RIDGE,cursor="hand2",font=("times new roman",15,"bold"),fg="white",bg="blue",activebackground="blue")

        b1.place(x=10,y=470,width=200)

        b1=Button(frame,text="Login",bd=3,relief=RIDGE,cursor="hand2",font=("times new roman",15,"bold"),fg="white",bg="green",activebackground="green")

        b1.place(x=330,y=470,width=200)

    #function declaration

    def register\_data(self):

        if self.var\_fname.get()=="" or self.var\_email.get()=="" or self.var\_securityq.get()=="Select":

            messagebox.showerror("Error","All fields required")

        elif self.var\_password.get()!=self.var\_confirmpassword.get():

            messagebox.showerror("Error","Password and Confirm Password must be same")

        elif self.var\_check.get()==0:

            messagebox.showerror("Error","Please agree our terms and condition")

        else:

            conn=mysql.connector.connect(host="localhost",user="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query=("select \* from register where email=%s")

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

            my\_cursor.execute(query,value)

            row=my\_cursor.fetchone()

            if row!=None:

                messagebox.showerror("Error","User already exist,please try another email")

            else:

                my\_cursor.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s)",(

                                                                                       self.var\_fname.get(),

                                                                                       self.var\_lname.get(),

                                                                                       self.var\_contact.get(),

                                                                                       self.var\_email.get(),

                                                                                       self.var\_securityq.get(),

                                                                                       self.var\_securitya.get(),

                                                                                       self.var\_password.get()

                                                                                         ))

            conn.commit()

            conn.close()

            messagebox.showinfo("Success","Register Successfull")

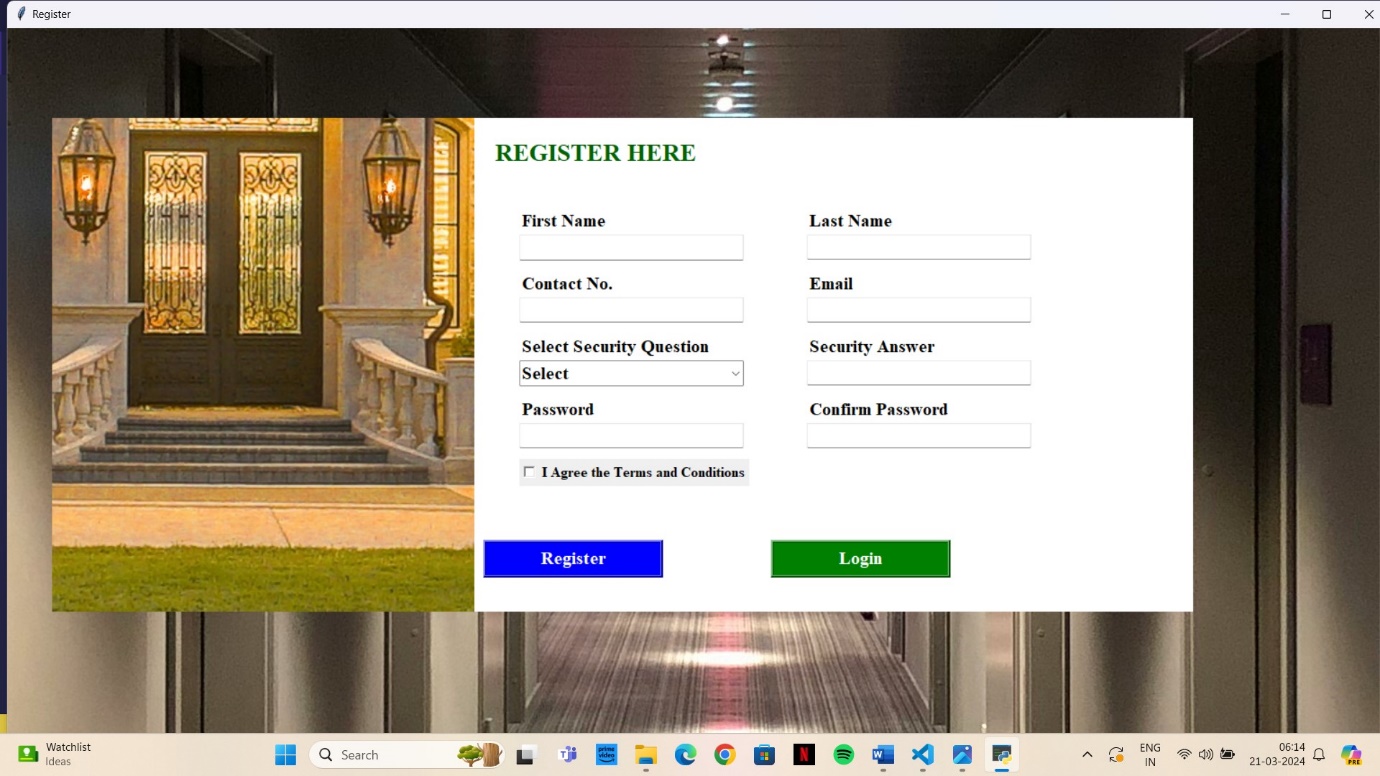
if \_\_name\_\_ == "\_\_main\_\_":

    root=Tk()

    app=Register(root)

    root.mainloop()

**Register Page Output:**



**Home Page Code:**

from tkinter import\*

from PIL import Image,ImageTk

from customer import Customer

from room import RoomBooking

from details import DetailsRoom

class HotelManagementSystem:

    def \_\_init\_\_(self,root):

        self.root=root

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

        self.root.geometry("1550x800+0+0")

        # =======================cover photo========================================

        img1=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\coverphoto.jpg")

        img1=img1.resize((1550,140),Image.BICUBIC)

        self.photoimg1=ImageTk.PhotoImage(img1)

        lblimg=Label(self.root,image=self.photoimg1,bd=4,relief=RIDGE)

        lblimg.place(x=0,y=0,width=1550,height=140)

        # =====================logo============================================

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((230,140),Image.BICUBIC)

        self.photoimg2=ImageTk.PhotoImage(img2)

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

        lblimg.place(x=0,y=0,width=230,height=140)

        # ====================title==================================================

        lbl\_title=Label(self.root,text="HOTEL MANAGEMENT SYSTEM",font=("times new roman",40,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=140,width=1550,height=50)

        # =================main frame==================================================

        main\_frame=Frame(self.root,bd=4,relief=RIDGE)

        main\_frame.place(x=0,y=190,width=1550,height=620)

        # ====================menu=================================================

        lbl\_menu=Label(main\_frame,text="MENU",font=("times new roman",20,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_menu.place(x=0,y=0,width=230)

        # ===================button frame============================================

        btn\_frame=Frame(main\_frame,bd=4,relief=RIDGE)

        btn\_frame.place(x=0,y=35,width=228,height=190)

        # ====================buttons==================================================

        cust\_btn=Button(btn\_frame,text="CUSTOMER",command=self.cust\_details,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        cust\_btn.grid(row=0,column=0,pady=1)

        room\_btn=Button(btn\_frame,text="ROOM",command=self.room\_booking,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        room\_btn.grid(row=1,column=0,pady=1)

        details\_btn=Button(btn\_frame,text="DETAILS",command=self.room\_details,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        details\_btn.grid(row=2,column=0,pady=1)

        report\_btn=Button(btn\_frame,text="REPORT",width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        report\_btn.grid(row=3,column=0,pady=1)

        logout\_btn=Button(btn\_frame,text="LOGOUT",command=self.logout,width=22,font=("times new roman",14,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")

        logout\_btn.grid(row=4,column=0,pady=1)

        # ========================right side image======================================

        img3=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\bgphoto.jpg")

        img3=img3.resize((1310,590),Image.BICUBIC)

        self.photoimg3=ImageTk.PhotoImage(img3)

        lblimg1=Label(main\_frame,image=self.photoimg3,bd=4,relief=RIDGE)

        lblimg1.place(x=225,y=0,width=1310,height=590)

        # ===========================down images=======================================

        img4=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\fillerphoto1.jpg")

        img4=img4.resize((230,210),Image.BICUBIC)

        self.photoimg4=ImageTk.PhotoImage(img4)

        lblimg1=Label(main\_frame,image=self.photoimg4,bd=4,relief=RIDGE)

        lblimg1.place(x=0,y=225,width=230,height=210)

        img5=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\fillerphoto2.jpg")

        img5=img5.resize((230,190),Image.BICUBIC)

        self.photoimg5=ImageTk.PhotoImage(img5)

        lblimg1=Label(main\_frame,image=self.photoimg5,bd=4,relief=RIDGE)

        lblimg1.place(x=0,y=420,width=230,height=190)

    def cust\_details(self):

        self.new\_window=Toplevel(self.root)

        self.app=Customer(self.new\_window)

    def room\_booking(self):

        self.new\_window=Toplevel(self.root)

        self.app=RoomBooking(self.new\_window)

    def room\_details(self):

        self.new\_window=Toplevel(self.root)

        self.app=DetailsRoom(self.new\_window)

    def logout(self):

        self.root.destroy()

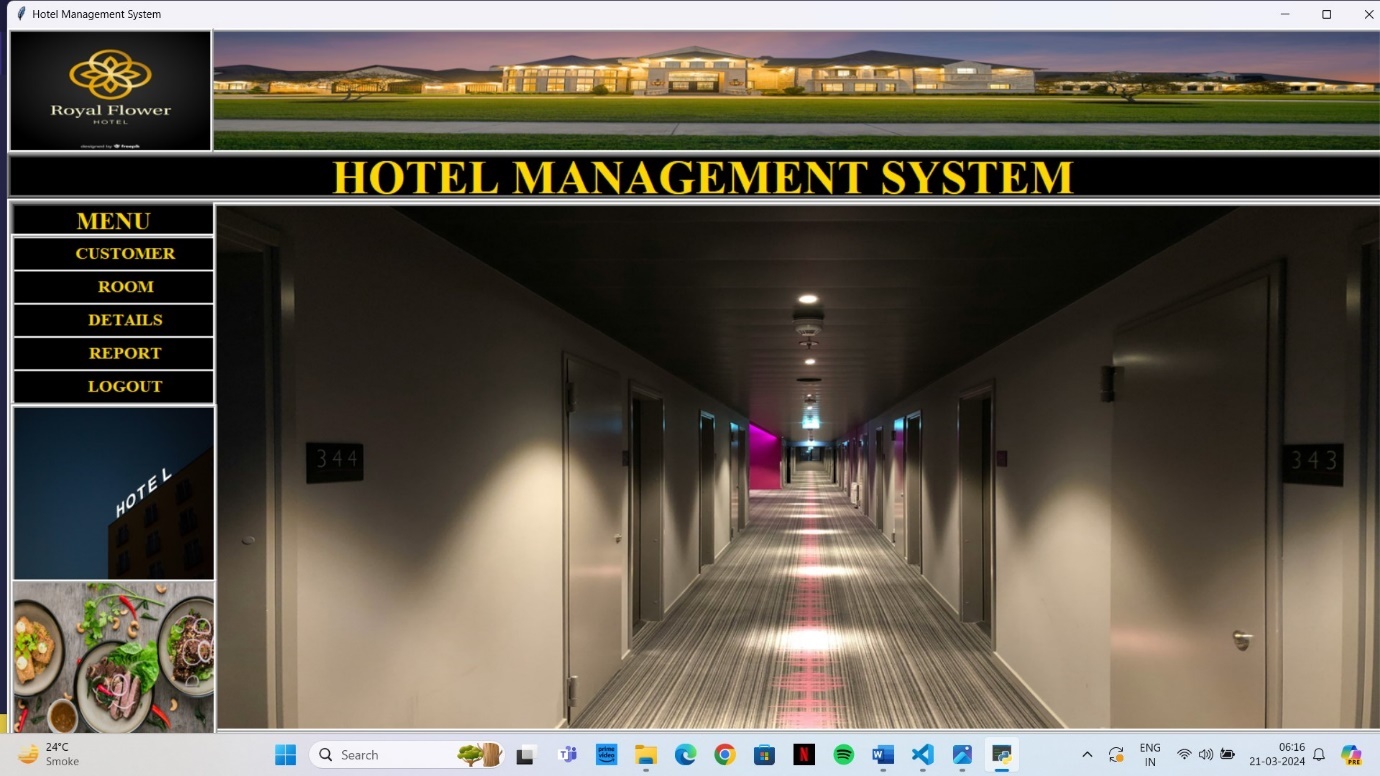
if \_\_name\_\_ == "\_\_main\_\_":

    root=Tk()

    obj=HotelManagementSystem(root)

    root.mainloop()

**Home Page Output:**



**Customer Page Code:**

from tkinter import\*

from PIL import Image,ImageTk

from tkinter import ttk

import random

from tkinter import messagebox

import mysql.connector

class Customer:

    def \_\_init\_\_(self,root):

        self.root=root

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

        self.root.geometry("1285x550+230+220")

        #variables

        self.var\_ref=StringVar()

        x=random.randint(1000,9999)

        self.var\_ref.set(str(x))

        self.var\_cust\_name=StringVar()

        self.var\_mother=StringVar()

        self.var\_gender=StringVar()

        self.var\_post=StringVar()

        self.var\_mobile=StringVar()

        self.var\_email=StringVar()

        self.var\_nationality=StringVar()

        self.var\_id\_proof=StringVar()

        self.var\_id\_number=StringVar()

        self.var\_address=StringVar()

         # ====================title==================================================

        lbl\_title=Label(self.root,text="ADD CUSTOMER DETAILS",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=0,width=1295,height=50)

         # =====================logo============================================

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((100,40),Image.BICUBIC)

        self.photoimg2=ImageTk.PhotoImage(img2)

        lblimg=Label(self.root,image=self.photoimg2,bd=0,relief=RIDGE)

        lblimg.place(x=5,y=2,width=100,height=40)

        #label frame

        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text="Customer Details",padx=2,font=("times new roman",12,"bold"))

        labelframeleft.place(x=5,y=50,width=425,height=490)

        #labels and entries

        #Customer Reference

        lbl\_cust\_ref=Label(labelframeleft,text="Customer Reference",font=("arial",12,"bold"),padx=2,pady=6)

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

        entry\_ref=ttk.Entry(labelframeleft,width=27,textvariable=self.var\_ref,font=("times new roman",13,"bold"),state="readonly")

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

        #Customer Name

        cname=Label(labelframeleft,font=("arial",12,"bold"),text="Customer Name:",padx=2,pady=6)

        cname.grid(row=1,column=0,sticky=W)

        txtcname=ttk.Entry(labelframeleft,textvariable=self.var\_cust\_name,font=("arial",12,"bold"),width=27)

        txtcname.grid(row=1,column=1)

        #Mother Name

        lblmname=Label(labelframeleft,font=("arial",12,"bold"),text="Mother Name:",padx=2,pady=6)

        lblmname.grid(row=2,column=0,sticky=W)

        txtmname=ttk.Entry(labelframeleft,textvariable=self.var\_mother,font=("arial",12,"bold"),width=27)

        txtmname.grid(row=2,column=1)

        #Gender Combobox

        label\_gender=Label(labelframeleft,font=("arial",12,"bold"),text="Gender:",padx=2,pady=6)

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

        combo\_gender=ttk.Combobox(labelframeleft,textvariable=self.var\_gender,font=("arial",12,"bold"),width=25,state="readonly")

        combo\_gender["value"]=("Male","Female","Other")

        combo\_gender.current(0)

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

        #Post Code

        lblPostCode=Label(labelframeleft,font=("arial",12,"bold"),text="PostCode:",padx=2,pady=6)

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

        txtPostCode=ttk.Entry(labelframeleft,textvariable=self.var\_post,font=("arial",12,"bold"),width=27)

        txtPostCode.grid(row=4,column=1)

        #Mobile Number

        lblMobile=Label(labelframeleft,font=("arial",12,"bold"),text="Mobile:",padx=2,pady=6)

        lblMobile.grid(row=5,column=0,sticky=W)

        txtMobile=ttk.Entry(labelframeleft,textvariable=self.var\_mobile,font=("arial",12,"bold"),width=27)

        txtMobile.grid(row=5,column=1)

        #Email

        lblEmail=Label(labelframeleft,font=("arial",12,"bold"),text="Email:",padx=2,pady=6)

        lblEmail.grid(row=6,column=0,sticky=W)

        txtEmail=ttk.Entry(labelframeleft,textvariable=self.var\_email,font=("arial",12,"bold"),width=27)

        txtEmail.grid(row=6,column=1)

        #Nationality

        lblNationality=Label(labelframeleft,font=("arial",12,"bold"),text="Nationality:",padx=2,pady=6)

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

        combo\_Nationality=ttk.Combobox(labelframeleft,textvariable=self.var\_nationality,font=("arial",12,"bold"),width=25,state="readonly")

        combo\_Nationality["value"]=("Indian","American","Japanese")

        combo\_Nationality.current(0)

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

        #Id-proof type combobox

        lblIdProof=Label(labelframeleft,font=("arial",12,"bold"),text="Id Proof Type:",padx=2,pady=6)

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

        combo\_Id=ttk.Combobox(labelframeleft,textvariable=self.var\_id\_proof,font=("arial",12,"bold"),width=25,state="readonly")

        combo\_Id["value"]=("AadharCard","DrivingLicense","Passport")

        combo\_Id.current(0)

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

        #Id Number

        lblIdNumber=Label(labelframeleft,font=("arial",12,"bold"),text="Id Number:",padx=2,pady=6)

        lblIdNumber.grid(row=9,column=0,sticky=W)

        txtIdNumber=ttk.Entry(labelframeleft,textvariable=self.var\_id\_number,font=("arial",13,"bold"),width=27)

        txtIdNumber.grid(row=9,column=1)

        #Address

        lblAddress=Label(labelframeleft,font=("arial",12,"bold"),text="Address:",padx=2,pady=6)

        lblAddress.grid(row=10,column=0,sticky=W)

        txtAddress=ttk.Entry(labelframeleft,textvariable=self.var\_address,font=("arial",13,"bold"),width=27)

        txtAddress.grid(row=10,column=1)

        #Buttons

        btn\_frame=Frame(labelframeleft,bd=2,relief=RIDGE)

        btn\_frame.place(x=0,y=400,width=412,height=40)

        btnAdd=Button(btn\_frame,text="Add",command=self.add\_data,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnAdd.grid(row=0,column=0,padx=1)

        btnUpdate=Button(btn\_frame,text="Update",command=self.update,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnUpdate.grid(row=0,column=1,padx=1)

        btnDelete=Button(btn\_frame,text="Delete",command=self.delete,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnDelete.grid(row=0,column=2,padx=1)

        btnReset=Button(btn\_frame,text="Reset",command=self.reset,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnReset.grid(row=0,column=3,padx=1)

        #Table Frame Search System

        Table\_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="View Details and Search System",font=("times new roman",12,"bold"))

        Table\_frame.place(x=435,y=50,width=860,height=490)

        lblSearchBy=Label(Table\_frame,font=("arial",12,"bold"),text="Search By:",bg="red",fg="white")

        lblSearchBy.grid(row=0,column=0,sticky=W,padx=2)

        self.search\_var=StringVar()

        combo\_Search=ttk.Combobox(Table\_frame,textvariable=self.search\_var,font=("arial",12,"bold"),width=24,state="readonly")

        combo\_Search["value"]=("Mobile","Ref")

        combo\_Search.current(0)

        combo\_Search.grid(row=0,column=1,padx=2)

        self.txt\_search=StringVar()

        txtSearch=ttk.Entry(Table\_frame,textvariable=self.txt\_search,font=("arial",13,"bold"),width=24)

        txtSearch.grid(row=0,column=2,padx=2)

        btnSearch=Button(Table\_frame,text="Search",command=self.search,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnSearch.grid(row=0,column=3,padx=1)

        btnShowAll=Button(Table\_frame,text="Show All",command=self.fetch\_data,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnShowAll.grid(row=0,column=4,padx=1)

        #Show Data Table

        details\_table=Frame(Table\_frame,bd=2,relief=RIDGE)

        details\_table.place(x=0,y=50,width=840,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","name","mother","gender","post","mobile",

                                                                    "email","nationality","idproof",

                                                                    "idnumber","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",text="Refer No.")

        self.Cust\_Details\_Table.heading("name",text="Name")

        self.Cust\_Details\_Table.heading("mother",text="Mother Name")

        self.Cust\_Details\_Table.heading("gender",text="Gender")

        self.Cust\_Details\_Table.heading("post",text="PostCode")

        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("idproof",text="Id Proof")

        self.Cust\_Details\_Table.heading("idnumber",text="Id Number")

        self.Cust\_Details\_Table.heading("address",text="Address")

        self.Cust\_Details\_Table["show"]="headings"

        self.Cust\_Details\_Table.column("ref",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("idproof",width=100)

        self.Cust\_Details\_Table.column("idnumber",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\_mobile.get()=="" or self.var\_mother.get()=="":

            messagebox.showerror("Error","All fields are required",parent=self.root)

        else:

            try:

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                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\_mother.get(),

                                                                                               self.var\_gender.get(),

                                                                                               self.var\_post.get(),

                                                                                               self.var\_mobile.get(),

                                                                                               self.var\_email.get(),

                                                                                               self.var\_nationality.get(),

                                                                                               self.var\_id\_proof.get(),

                                                                                               self.var\_id\_number.get(),

                                                                                               self.var\_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"Something went wrong:{str(es)}",parent=self.root)

    def fetch\_data(self):

        conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

        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\_mother.set(row[2]),

        self.var\_gender.set(row[3]),

        self.var\_post.set(row[4]),

        self.var\_mobile.set(row[5]),

        self.var\_email.set(row[6]),

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

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

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

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

    def update(self):

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

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

        else:

            conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            my\_cursor.execute("update customer set Name=%s,Mother=%s,Gender=%s,PostCode=%s,Mobile=%s,Email=%s,Nationality=%s,IdProof=%s,IdNumber=%s,Address=%s where Ref=%s",(

                                                                                                                                                                                self.var\_cust\_name.get(),

                                                                                                                                                                                self.var\_mother.get(),

                                                                                                                                                                                self.var\_gender.get(),

                                                                                                                                                                                self.var\_post.get(),

                                                                                                                                                                                self.var\_mobile.get(),

                                                                                                                                                                                self.var\_email.get(),

                                                                                                                                                                                self.var\_nationality.get(),

                                                                                                                                                                                self.var\_id\_proof.get(),

                                                                                                                                                                                self.var\_id\_number.get(),

                                                                                                                                                                                self.var\_address.get(),

                                                                                                                                                                                self.var\_ref.get()

                                                                                                                                                                               ))

            conn.commit()

            self.fetch\_data()

            conn.close()

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

    def delete(self):

        delete=messagebox.askyesno("YesNo","Do you want to delete this customer",parent=self.root)

        if delete>0:

            conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query="delete from customer where Ref=%s"

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

            my\_cursor.execute(query,value)

        else:

            if not delete:

                return

        conn.commit()

        self.fetch\_data()

        conn.close()

    def reset(self):

        # self.var\_ref.set(""),

        self.var\_cust\_name.set(""),

        self.var\_mother.set(""),

        # self.var\_gender.set(""),

        self.var\_post.set(""),

        self.var\_mobile.set(""),

        self.var\_email.set(""),

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

        # self.var\_id\_proof.set(""),

        self.var\_id\_number.set(""),

        self.var\_address.set("")

        x=random.randint(1000,9999)

        self.var\_ref.set(str(x))

    def search(self):

         conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

         my\_cursor=conn.cursor()

         my\_cursor.execute("SELECT \* FROM customer WHERE " + str(self.search\_var.get()) + " LIKE '%" + str(self.txt\_search.get()) + "%'")

         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()

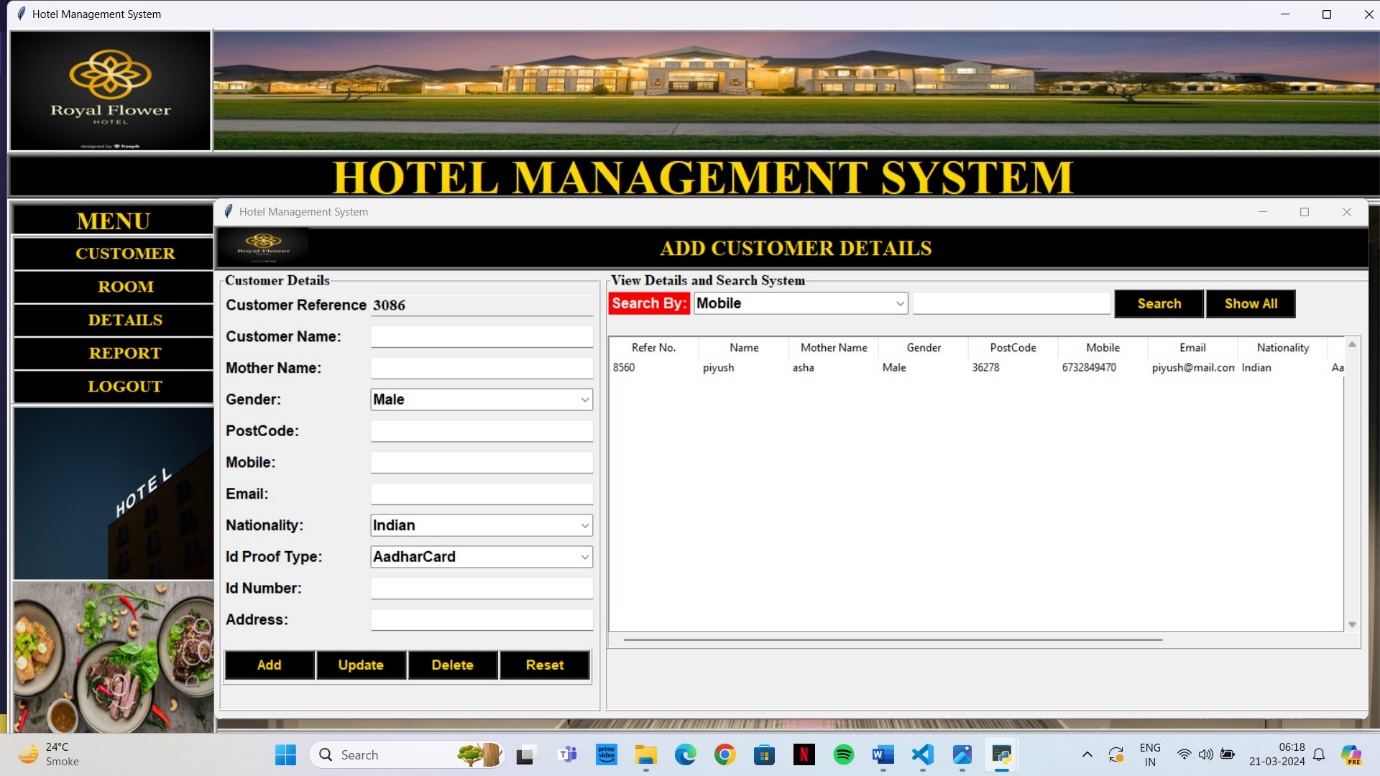
if \_\_name\_\_ == "\_\_main\_\_":

    root=Tk()

    obj=Customer(root)

    root.mainloop()

**Customer Page Output:**



**Room Booking Page Code:**

from tkinter import\*

from PIL import Image,ImageTk

from tkinter import ttk

import random

from time import strftime

from datetime import datetime

from tkinter import messagebox

import mysql.connector

class RoomBooking:

    def \_\_init\_\_(self,root):

        self.root=root

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

        self.root.geometry("1285x550+230+220")

        #variables

        self.var\_contact=StringVar()

        self.var\_checkin=StringVar()

        self.var\_checkout=StringVar()

        self.var\_roomtype=StringVar()

        self.var\_roomavailable=StringVar()

        self.var\_meal=StringVar()

        self.var\_noofdays=StringVar()

        self.var\_paidtax=StringVar()

        self.var\_actualtotal=StringVar()

        self.var\_total=StringVar()

         # ====================title==================================================

        lbl\_title=Label(self.root,text="ROOM BOOKING",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=0,width=1295,height=50)

        # =====================logo============================================

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((100,40),Image.BICUBIC)

        self.photoimg2=ImageTk.PhotoImage(img2)

        lblimg=Label(self.root,image=self.photoimg2,bd=0,relief=RIDGE)

        lblimg.place(x=5,y=2,width=100,height=40)

        #label frame

        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text="Room Booking",padx=2,font=("times new roman",12,"bold"))

        labelframeleft.place(x=5,y=50,width=425,height=490)

        #labels and entries

        #Customer Contact

        lbl\_cust\_contact=Label(labelframeleft,text="Customer Contact",font=("arial",12,"bold"),padx=2,pady=6)

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

        entry\_contact=ttk.Entry(labelframeleft,textvariable=self.var\_contact,width=20,font=("times new roman",13,"bold"))

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

        #Fetch Data Button

        btnFetchData=Button(labelframeleft,command=self.fetch\_contact,text="Fetch Data",font=("arial",8,"bold"),bg="black",fg="gold",width=8)

        btnFetchData.place(x=347,y=4)

        #labels and entries

        #Check-in Date

        check\_in\_date=Label(labelframeleft,text="Check-in Date",font=("arial",12,"bold"),padx=2,pady=6)

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

        txtcheck\_in\_date=ttk.Entry(labelframeleft,textvariable=self.var\_checkin,width=27,font=("times new roman",13,"bold"))

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

        #Check-out Date

        lbl\_check\_out=Label(labelframeleft,text="Check-out Date",font=("arial",12,"bold"),padx=2,pady=6)

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

        txt\_check\_out=ttk.Entry(labelframeleft,textvariable=self.var\_checkout,width=27,font=("times new roman",13,"bold"))

        txt\_check\_out.grid(row=2,column=1)

        #Room Type

        label\_RoomType=Label(labelframeleft,text="Room Type",font=("arial",12,"bold"),padx=2,pady=6)

        label\_RoomType.grid(row=3,column=0,sticky=W)

        conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

        my\_cursor=conn.cursor()

        my\_cursor.execute("select RoomType from details")

        room\_types=my\_cursor.fetchall()

        combo\_RoomType=ttk.Combobox(labelframeleft,textvariable=self.var\_roomtype,font=("arial",12,"bold"),width=25,state="readonly")

        combo\_RoomType["values"]=room\_types

        combo\_RoomType.current(0)

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

        #Available Room

        lblRoomAvailable=Label(labelframeleft,text="Available Room",font=("arial",12,"bold"),padx=2,pady=6)

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

        # txtRoomAvailable=ttk.Entry(labelframeleft,textvariable=self.var\_roomavailable,width=27,font=("times new roman",13,"bold"))

        # txtRoomAvailable.grid(row=4,column=1)

        conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

        my\_cursor=conn.cursor()

        my\_cursor.execute("select `RoomNo.` from details")

        rows=my\_cursor.fetchall()

        combo\_RoomNo=ttk.Combobox(labelframeleft,textvariable=self.var\_roomavailable,font=("arial",12,"bold"),width=25,state="readonly")

        combo\_RoomNo["value"]=rows

        combo\_RoomNo.current(0)

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

         #Meal

        lblMeal=Label(labelframeleft,text="Meal",font=("arial",12,"bold"),padx=2,pady=6)

        lblMeal.grid(row=5,column=0,sticky=W)

        txtMeal=ttk.Entry(labelframeleft,textvariable=self.var\_meal,width=27,font=("times new roman",13,"bold"))

        txtMeal.grid(row=5,column=1)

        #No. Of Days

        lblNoOfDays=Label(labelframeleft,text="No. Of Days",font=("arial",12,"bold"),padx=2,pady=6)

        lblNoOfDays.grid(row=6,column=0,sticky=W)

        txtNoOfDays=ttk.Entry(labelframeleft,textvariable=self.var\_noofdays,width=27,font=("times new roman",13,"bold"))

        txtNoOfDays.grid(row=6,column=1)

        #Paid Tax

        lblPaidTax=Label(labelframeleft,text="Paid Tax",font=("arial",12,"bold"),padx=2,pady=6)

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

        txtPaidTax=ttk.Entry(labelframeleft,textvariable=self.var\_paidtax,width=27,font=("times new roman",13,"bold"))

        txtPaidTax.grid(row=7,column=1)

        #Sub Total

        lblSubTotal=Label(labelframeleft,text="Sub Total",font=("arial",12,"bold"),padx=2,pady=6)

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

        txtSubTotal=ttk.Entry(labelframeleft,textvariable=self.var\_actualtotal,width=27,font=("times new roman",13,"bold"))

        txtSubTotal.grid(row=8,column=1)

        #Total Cost

        lblTotalCost=Label(labelframeleft,text="Total Cost",font=("arial",12,"bold"),padx=2,pady=6)

        lblTotalCost.grid(row=9,column=0,sticky=W)

        txtTotalCost=ttk.Entry(labelframeleft,textvariable=self.var\_total,width=27,font=("times new roman",13,"bold"))

        txtTotalCost.grid(row=9,column=1)

        #Bill Button

        btnBill=Button(labelframeleft,text="Bill",command=self.total,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnBill.grid(row=10,column=0,padx=1,sticky=W)

         #Buttons

        btn\_frame=Frame(labelframeleft,bd=2,relief=RIDGE)

        btn\_frame.place(x=0,y=400,width=412,height=40)

        btnAdd=Button(btn\_frame,text="Add",command=self.add\_data,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnAdd.grid(row=0,column=0,padx=1)

        btnUpdate=Button(btn\_frame,text="Update",command=self.update,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnUpdate.grid(row=0,column=1,padx=1)

        btnDelete=Button(btn\_frame,text="Delete",command=self.delete,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnDelete.grid(row=0,column=2,padx=1)

        btnReset=Button(btn\_frame,text="Reset",command=self.reset,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnReset.grid(row=0,column=3,padx=1)

        #Right Side Image

        img3=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\fillerphoto1.jpg")

        img3=img3.resize((520,300),Image.BICUBIC)

        self.photoimg3=ImageTk.PhotoImage(img3)

        lblimg=Label(self.root,image=self.photoimg3,bd=0,relief=RIDGE)

        lblimg.place(x=760,y=55,width=520,height=300)

         #Table Frame Search System

        Table\_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="View Details and Search System",font=("times new roman",12,"bold"))

        Table\_frame.place(x=435,y=280,width=860,height=260)

        lblSearchBy=Label(Table\_frame,font=("arial",12,"bold"),text="Search By:",bg="red",fg="white")

        lblSearchBy.grid(row=0,column=0,sticky=W,padx=2)

        self.search\_var=StringVar()

        combo\_Search=ttk.Combobox(Table\_frame,textvariable=self.search\_var,font=("arial",12,"bold"),width=24,state="readonly")

        combo\_Search["value"]=("Contact","Room")

        combo\_Search.current(0)

        combo\_Search.grid(row=0,column=1,padx=2)

        self.txt\_search=StringVar()

        txtSearch=ttk.Entry(Table\_frame,textvariable=self.txt\_search,font=("arial",13,"bold"),width=24)

        txtSearch.grid(row=0,column=2,padx=2)

        btnSearch=Button(Table\_frame,text="Search",command=self.search,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnSearch.grid(row=0,column=3,padx=1)

        btnShowAll=Button(Table\_frame,text="Show All",command=self.fetch\_data,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnShowAll.grid(row=0,column=4,padx=1)

         #Show Data Table

        details\_table=Frame(Table\_frame,bd=2,relief=RIDGE)

        details\_table.place(x=0,y=50,width=840,height=180)

        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","checkin","checkout","roomtype","roomavailable","meal","noofdays",),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 No.")

        self.room\_table.heading("checkin",text="Check-In Date")

        self.room\_table.heading("checkout",text="Check-Out Date")

        self.room\_table.heading("roomtype",text="Room Type")

        self.room\_table.heading("roomavailable",text="Room No.")

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

        self.room\_table.heading("noofdays",text="No. Of Days")

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

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

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

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

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

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

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

        self.room\_table.column("noofdays",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\_checkin.get()=="":

            messagebox.showerror("Error","All fields are required",parent=self.root)

        else:

            try:

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                my\_cursor=conn.cursor()

                my\_cursor.execute("insert into room values(%s,%s,%s,%s,%s,%s,%s)",(

                                                                                                    self.var\_contact.get(),

                                                                                                    self.var\_checkin.get(),

                                                                                                    self.var\_checkout.get(),

                                                                                                    self.var\_roomtype.get(),

                                                                                                    self.var\_roomavailable.get(),

                                                                                                    self.var\_meal.get(),

                                                                                                    self.var\_noofdays.get()

                                                                                               ))

                conn.commit()

                self.fetch\_data()

                conn.close()

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

            except Exception as es:

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

    #Fetch Data

    def fetch\_data(self):

        conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

        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\_checkin.set(row[1])

        self.var\_checkout.set(row[2])

        self.var\_roomtype.set(row[3])

        self.var\_roomavailable.set(row[4])

        self.var\_meal.set(row[5])

        self.var\_noofdays.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(host="localhost",username="root",password="root",database="hotel\_management\_system")

            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\_checkin.get(),

                                                                                                                                                    self.var\_checkout.get(),

                                                                                                                                                    self.var\_roomtype.get(),

                                                                                                                                                    self.var\_roomavailable.get(),

                                                                                                                                                    self.var\_meal.get(),

                                                                                                                                                    self.var\_noofdays.get(),

                                                                                                                                                    self.var\_contact.get()

                                                                                                                                                  ))

            conn.commit()

            self.fetch\_data()

            conn.close()

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

    def delete(self):

        delete=messagebox.askyesno("YesNo","Do you want to delete this room",parent=self.root)

        if delete>0:

            conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

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

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

            my\_cursor.execute(query,value)

        else:

            if not delete:

                return

        conn.commit()

        self.fetch\_data()

        conn.close()

    def reset(self):

        self.var\_contact.set("")

        self.var\_checkin.set("")

        self.var\_checkout.set("")

        self.var\_roomtype.set("")

        self.var\_roomavailable.set("")

        self.var\_meal.set("")

        self.var\_noofdays.set("")

        self.var\_paidtax.set("")

        self.var\_actualtotal.set("")

        self.var\_total.set("")

    # All Data Fetch

    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="root",database="hotel\_management\_system")

            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","This Number Not Found",parent=self.root)

            else:

                conn.commit()

                conn.close()

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

                showDataFrame.place(x=450,y=55,width=300,height=180)

                lblName=Label(showDataFrame,text="Name:",font=("arial",12,"bold"))

                lblName.place(x=0,y=0)

                lbl=Label(showDataFrame,text=row,font=("arial",12,"bold"))

                lbl.place(x=90,y=0)

               # Gender

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                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",12,"bold"))

                lblGender.place(x=0,y=30)

                lbl1=Label(showDataFrame,text=row,font=("arial",12,"bold"))

                lbl1.place(x=90,y=30)

                #Email

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                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()

                lblEmail=Label(showDataFrame,text="Email:",font=("arial",12,"bold"))

                lblEmail.place(x=0,y=60)

                lbl2=Label(showDataFrame,text=row,font=("arial",12,"bold"))

                lbl2.place(x=90,y=60)

                #Nationality

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                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()

                lblNationality=Label(showDataFrame,text="Nationality:",font=("arial",12,"bold"))

                lblNationality.place(x=0,y=90)

                lbl3=Label(showDataFrame,text=row,font=("arial",12,"bold"))

                lbl3.place(x=90,y=90)

                #Address

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                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()

                lblAddress=Label(showDataFrame,text="Address:",font=("arial",12,"bold"))

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

                lbl4=Label(showDataFrame,text=row,font=("arial",12,"bold"))

                lbl4.place(x=90,y=120)

    #Search System

    def search(self):

         conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

         my\_cursor=conn.cursor()

         my\_cursor.execute("SELECT \* FROM room WHERE " + str(self.search\_var.get()) + " LIKE '%" + str(self.txt\_search.get()) + "%'")

         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 total(self):

        inDate=self.var\_checkin.get()

        outDate=self.var\_checkout.get()

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

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

        self.var\_noofdays.set(abs(outDate-inDate).days)

        if (self.var\_meal.get()=="Breakfast" and self.var\_roomtype.get()=="Luxury"):

            q1=float(300)

            q2=float(3000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Lunch" and self.var\_roomtype.get()=="Luxury"):

            q1=float(500)

            q2=float(3000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Dinner" and self.var\_roomtype.get()=="Luxury"):

            q1=float(700)

            q2=float(3000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Breakfast" and self.var\_roomtype.get()=="Single"):

            q1=float(300)

            q2=float(1000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Lunch" and self.var\_roomtype.get()=="Single"):

            q1=float(500)

            q2=float(1000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Dinner" and self.var\_roomtype.get()=="Single"):

            q1=float(700)

            q2=float(1000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Breakfast" and self.var\_roomtype.get()=="Double"):

            q1=float(300)

            q2=float(2000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Lunch" and self.var\_roomtype.get()=="Double"):

            q1=float(500)

            q2=float(2000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

            self.var\_total.set(Total\_T)

        elif (self.var\_meal.get()=="Dinner" and self.var\_roomtype.get()=="Double"):

            q1=float(700)

            q2=float(2000)

            q3=float(self.var\_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3+q4)

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

            Sub\_T="Rs."+str("%.2f"%((q5)))

            Total\_T="Rs."+str("%.2f"%((q5+(q5)\*0.1)))

            self.var\_paidtax.set(Tax)

            self.var\_actualtotal.set(Sub\_T)

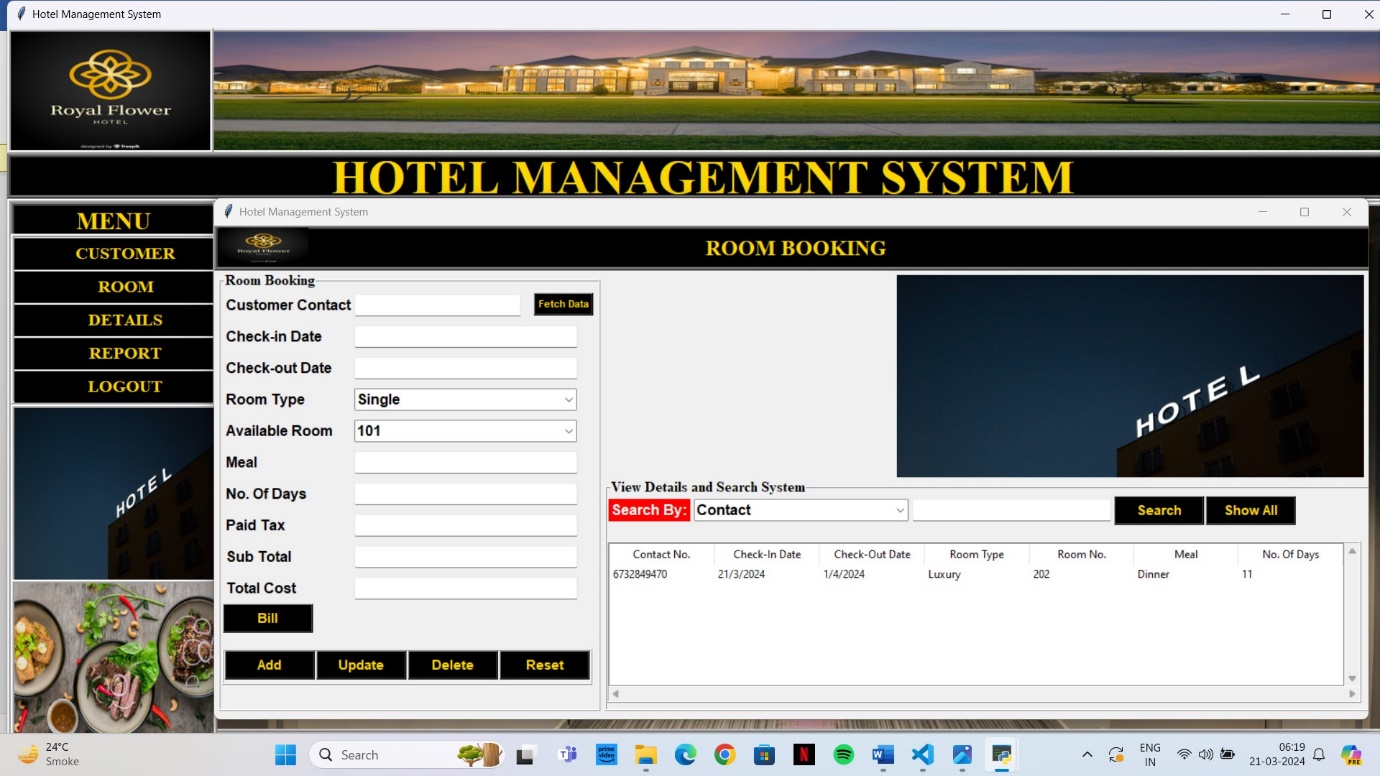
            self.var\_total.set(Total\_T)

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

    root=Tk()

    obj=RoomBooking(root)

    root.mainloop()

**Room Booking Page Output:**

**Room Details Page Code:**

from tkinter import\*

from PIL import Image,ImageTk

from tkinter import ttk

import random

from time import strftime

from datetime import datetime

from tkinter import messagebox

import mysql.connector

class DetailsRoom:

    def \_\_init\_\_(self,root):

        self.root=root

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

        self.root.geometry("1285x550+230+220")

         # ====================title==================================================

        lbl\_title=Label(self.root,text="ROOM BOOKING DETAILS",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl\_title.place(x=0,y=0,width=1295,height=50)

        # =====================logo============================================

        img2=Image.open("C:\\Users\\TANISHA\\OneDrive\\Desktop\\Hotel\_Management\_System\\Images\\logo.jpg")

        img2=img2.resize((100,40),Image.BICUBIC)

        self.photoimg2=ImageTk.PhotoImage(img2)

        lblimg=Label(self.root,image=self.photoimg2,bd=0,relief=RIDGE)

        lblimg.place(x=5,y=2,width=100,height=40)

        #label frame

        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text="New Room Add",padx=2,font=("times new roman",12,"bold"))

        labelframeleft.place(x=5,y=50,width=540,height=350)

        #Floor

        lbl\_floor=Label(labelframeleft,text="Floor",font=("arial",12,"bold"),padx=2,pady=6)

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

        self.var\_floor=StringVar()

        entry\_floor=ttk.Entry(labelframeleft,textvariable=self.var\_floor,width=20,font=("times new roman",13,"bold"))

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

         #Room No.

        lbl\_RoomNo=Label(labelframeleft,text="Room No.",font=("arial",12,"bold"),padx=2,pady=6)

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

        self.var\_roomno=StringVar()

        entry\_RoomNo=ttk.Entry(labelframeleft,textvariable=self.var\_roomno,width=20,font=("times new roman",13,"bold"))

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

         #Room Type

        lbl\_RoomType=Label(labelframeleft,text="Room Type",font=("arial",12,"bold"),padx=2,pady=6)

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

        self.var\_roomtype=StringVar()

        entry\_RoomType=ttk.Entry(labelframeleft,textvariable=self.var\_roomtype,width=20,font=("times new roman",13,"bold"))

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

         #Buttons

        btn\_frame=Frame(labelframeleft,bd=2,relief=RIDGE)

        btn\_frame.place(x=0,y=200,width=412,height=40)

        btnAdd=Button(btn\_frame,text="Add",command=self.add\_data,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnAdd.grid(row=0,column=0,padx=1)

        btnUpdate=Button(btn\_frame,text="Update",command=self.update,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnUpdate.grid(row=0,column=1,padx=1)

        btnDelete=Button(btn\_frame,text="Delete",command=self.delete,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnDelete.grid(row=0,column=2,padx=1)

        btnReset=Button(btn\_frame,text="Reset",command=self.reset,font=("arial",11,"bold"),bg="black",fg="gold",width=10)

        btnReset.grid(row=0,column=3,padx=1)

          #Table Frame Search System

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

        Table\_frame.place(x=600,y=55,width=600,height=350)

        scroll\_x=ttk.Scrollbar(Table\_frame,orient=HORIZONTAL)

        scroll\_y=ttk.Scrollbar(Table\_frame,orient=VERTICAL)

        self.room\_table=ttk.Treeview(Table\_frame,columns=("floor","roomno","roomtype",),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("floor",text="Floor")

        self.room\_table.heading("roomno",text="Room No.")

        self.room\_table.heading("roomtype",text="Room Type")

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

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

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

        self.room\_table.column("roomtype",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\_floor.get()=="" or self.var\_roomtype.get()=="":

            messagebox.showerror("Error","All fields are required",parent=self.root)

        else:

            try:

                conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

                my\_cursor=conn.cursor()

                my\_cursor.execute("insert into details values(%s,%s,%s)",(

                                                                                                    self.var\_floor.get(),

                                                                                                    self.var\_roomno.get(),

                                                                                                    self.var\_roomtype.get(),

                                                                                               ))

                conn.commit()

                self.fetch\_data()

                conn.close()

                messagebox.showinfo("Success","New Room Added Successfully",parent=self.root)

            except Exception as es:

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

    def fetch\_data(self):

        conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

        my\_cursor=conn.cursor()

        my\_cursor.execute("select \* from details")

        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"]

        if row:

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

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

            self.var\_roomtype.set(row[2])

    def update(self):

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

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

        else:

         conn = mysql.connector.connect(host="localhost", username="root", password="root", database="hotel\_management\_system")

         my\_cursor = conn.cursor()

         my\_cursor.execute("update details set Floor=%s, RoomType=%s where `RoomNo.`=%s", (

            self.var\_floor.get(),

            self.var\_roomtype.get(),

            self.var\_roomno.get(),

        ))

        conn.commit()

        self.fetch\_data()

        conn.close()

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

    def delete(self):

        delete=messagebox.askyesno("YesNo","Do you want to delete this room details",parent=self.root)

        if delete>0:

            conn=mysql.connector.connect(host="localhost",username="root",password="root",database="hotel\_management\_system")

            my\_cursor=conn.cursor()

            query="delete from details where `RoomNo.`=%s"

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

            my\_cursor.execute(query,value)

        else:

            if not delete:

                return

        conn.commit()

        self.fetch\_data()

        conn.close()

    def reset(self):

        self.var\_floor.set(""),

        self.var\_roomno.set(""),

        self.var\_roomtype.set("")

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

    root=Tk()

    obj=DetailsRoom(root)

    root.mainloop()

**Room Details Page Output:**

