VITYARTHI PROJECT: HOTEL MANAGEMENT SYSTEM

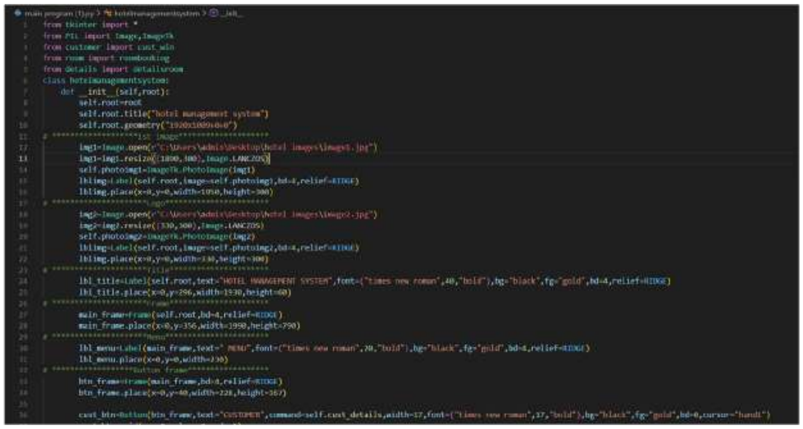
By
PRISHA ARORA
(25BAI10290)

VIT Bhopal University
Kothrikalan -466114,
Madhya Pradesh,
India

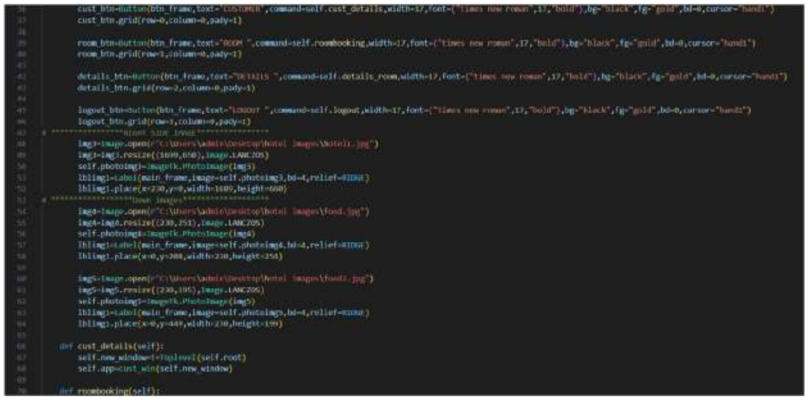
Algorithm:

1. Import Libraries and Modules: Import necessary Python libraries such as Tkinter for GUI, PIL for image handling, and custom modules for customer, room, and details functionalities.
2. Define Main Class that initializes the main window using Tkinter.
3. Initialize GUI Window to set the window's title and geometry (size).
4. Load and display several images (such as header, logo, menu background) using PIL and ImageTk.
5. Create a main frame to organize the GUI layout. Add a title label for the application.
Create Menu and Buttons and add a sidebar/menu section with labeled buttons for different operations such as customer: opens customer details window; room: opens room booking window; details: opens room details window; logout: closes the application. Place each button in a dedicated frame.
6. Display Images and Layout: It used to load images for various visual sections of the application (top banner, sidebar, main area).
7. Define Button Commands: connect each menu button to a method that launches the corresponding functionality window using Toplevel windows.
8. Run Main Loop
9. Instantiate the hotel management system class.
10. Start the main event loop for the application.

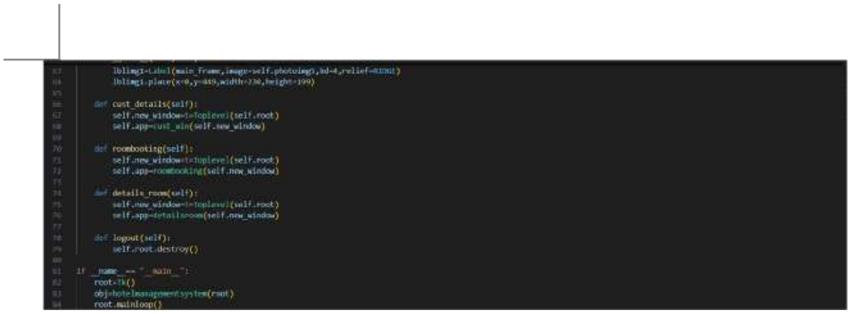
Main Program:



```
from tkinter import *
from PIL import Image,ImageTk
from customer import cust_win
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("1920x1080+0+0")
# ********************1st image********************
        img1=Image.open(r"C:\Users\admin\Desktop\hotel images\image1.jpg")
        img1=img1.resize((1800,300),Image.LANCZOS)
        self.photoimg1=ImageTk.PhotoImage(img1)
        lblimg=Label(self.root,image=self.photoimg1,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=1950,height=300)
# ********************logo********************
        img2=Image.open(r"C:\Users\admin\Desktop\hotel images\image2.jpg")
        img2=img2.resize((330,300),Image.LANCZOS)
        self.photoimg2=ImageTk.PhotoImage(img2)
        lblimg=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=330,height=300)
# ********************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=296,width=1930,height=60)
# ********************frame********************
        main_frame=Frame(self.root,bd=4,relief=RIDGE)
        main_frame.place(x=0,y=356,width=1990,height=790)
# ********************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=40,width=228,height=167)

        cust_btn=Button(btn_frame,text="CUSTOMER",command=self.cust_details,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand1")
```

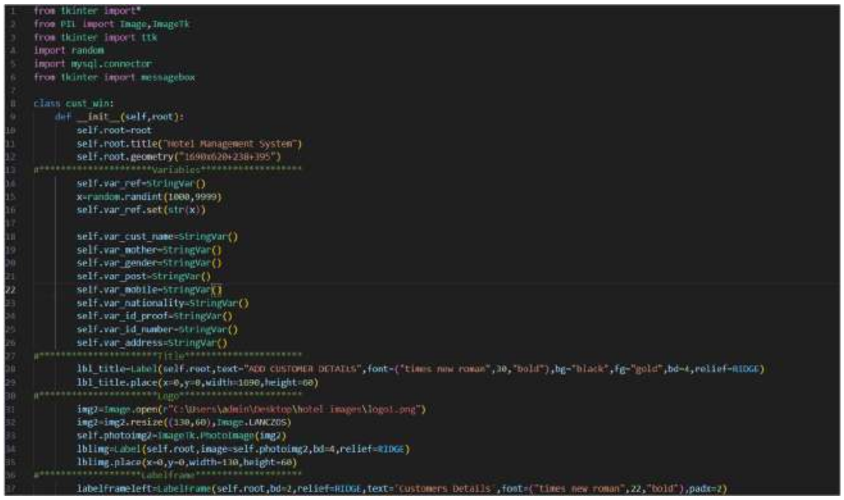


```
        cust_btn=Button(btn_frame,text="CUSTOMER",command=self.cust_details,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand1")
        cust_btn.grid(row=0,column=0,pady=1)

        room_btn=Button(btn_frame,text="ROOM ",command=self.roombooking,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand1")
        room_btn.grid(row=1,column=0,pady=1)

        details_btn=Button(btn_frame,text="DETAILS ",command=self.details_room,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand1")
        details_btn.grid(row=2,column=0,pady=1)

        logout_btn=Button(btn_frame,text="LOGOUT ",command=self.logout,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand1")
        logout_btn.grid(row=3,column=0,pady=1)
# ****************RIGHT SIDE IMAGE****************
        img3=Image.open(r"C:\Users\admin\Desktop\hotel images\hotel1.jpg")
        img3=img3.resize((1699,650),Image.LANCZOS)
        self.photoimg3=ImageTk.PhotoImage(img3)
        lblimg1=Label(main_frame,image=self.photoimg3,bd=4,relief=RIDGE)
        lblimg1.place(x=230,y=0,width=1689,height=660)
# ****************Down images****************
        img4=Image.open(r"C:\Users\admin\Desktop\hotel images\food.jpg")
        img4=img4.resize((230,251),Image.LANCZOS)
        self.photoimg4=ImageTk.PhotoImage(img4)
        lblimg1=Label(main_frame,image=self.photoimg4,bd=4,relief=RIDGE)
        lblimg1.place(x=0,y=208,width=230,height=251)

        img5=Image.open(r"C:\Users\admin\Desktop\hotel images\food1.jpg")
        img5=img5.resize((230,195),Image.LANCZOS)
        self.photoimg5=ImageTk.PhotoImage(img5)
        lblimg1=Label(main_frame,image=self.photoimg5,bd=4,relief=RIDGE)
        lblimg1.place(x=0,y=449,width=230,height=199)

    def cust_details(self):
        self.new_window=Toplevel(self.root)
        self.app=cust_win(self.new_window)

    def roombooking(self):
```

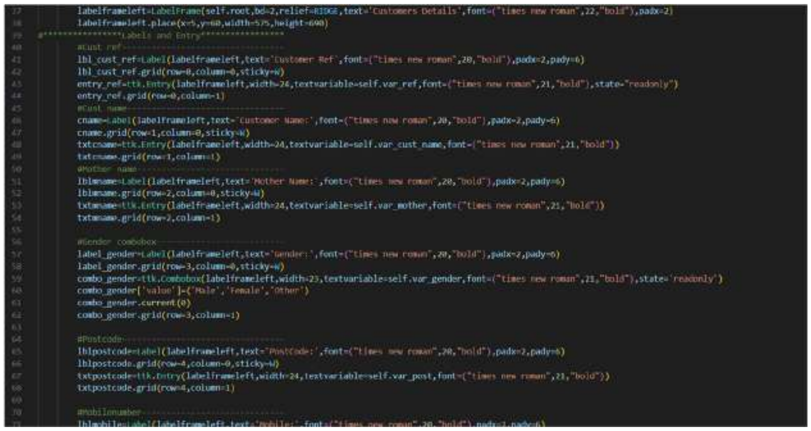


```
        lblimg1=Label(main_frame,image=self.photoimg3,bd=4,relief=RIDGE)
        lblimg1.place(x=0,y=449,width=230,height=199)

    def cust_details(self):
        self.new_window=Toplevel(self.root)
        self.app=cust_win(self.new_window)

    def roombooking(self):
        self.new_window=Toplevel(self.root)
        self.app=roombooking(self.new_window)

    def details_room(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()
```

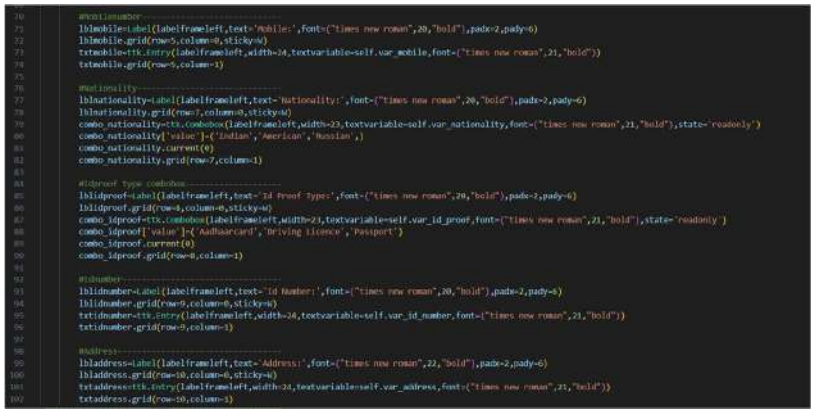
Customer Details Program:



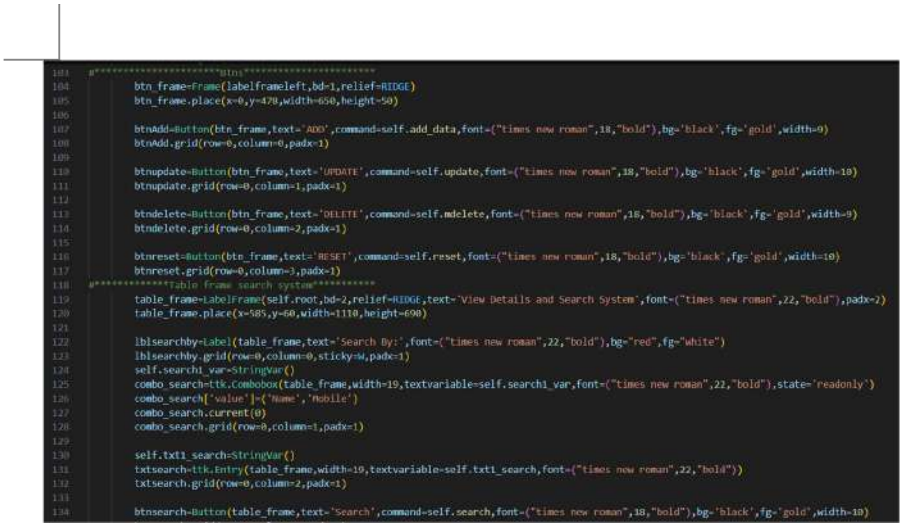
```
from tkinter import*
from PIL import Image,ImageTk
from tkinter import ttk
import random
import mysql.connector
from tkinter import messagebox

class cust_win:
    def __init__(self,root):
        self.root=root
        self.root.title("Hotel Management System")
        self.root.geometry("1690x620+238+395")
#*********************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_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",30,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
        lbl_title.place(x=0,y=0,width=1090,height=60)
#***********************Logo************************
        img2=Image.open(r"C:\Users\admin\Desktop\hotel images\logo1.png")
        img2=img2.resize((130,60),Image.LANCZOS)
        self.photoimg2=ImageTk.PhotoImage(img2)
        lblimg=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=130,height=60)
#*******************labelframe*********************
        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text='Customers Details',font=("times new roman",22,"bold"),padx=2)
```



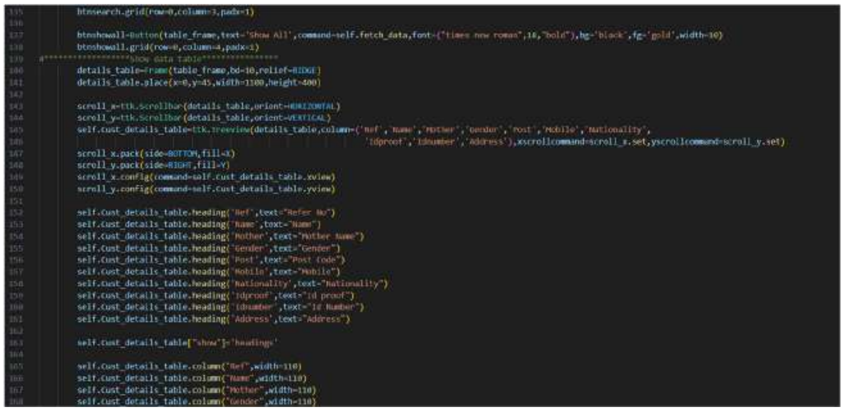
```
        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text='Customers Details',font=("times new roman",22,"bold"),padx=2)
        labelframeleft.place(x=5,y=60,width=575,height=690)
 #****************Labels and Entry*****************
        #Cust ref--------------------------------
        lbl_cust_ref=Label(labelframeleft,text='Customer Ref',font=("times new roman",20,"bold"),padx=2,pady=6)
        lbl_cust_ref.grid(row=0,column=0,sticky=W)
        entry_ref=ttk.Entry(labelframeleft,width=24,textvariable=self.var_ref,font=("times new roman",21,"bold"),state="readonly")
        entry_ref.grid(row=0,column=1)
        #Cust name-------------------------------
        cname=Label(labelframeleft,text='Customer Name:',font=("times new roman",20,"bold"),padx=2,pady=6)
        cname.grid(row=1,column=0,sticky=W)
        txtcname=ttk.Entry(labelframeleft,width=24,textvariable=self.var_cust_name,font=("times new roman",21,"bold"))
        txtcname.grid(row=1,column=1)
        #Mother name-----------------------------
        lblmname=Label(labelframeleft,text='Mother Name:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblmname.grid(row=2,column=0,sticky=W)
        txtmname=ttk.Entry(labelframeleft,width=24,textvariable=self.var_mother,font=("times new roman",21,"bold"))
        txtmname.grid(row=2,column=1)

        #Gender combobox-------------------------
        label_gender=Label(labelframeleft,text='Gender:',font=("times new roman",20,"bold"),padx=2,pady=6)
        label_gender.grid(row=3,column=0,sticky=W)
        combo_gender=ttk.Combobox(labelframeleft,width=23,textvariable=self.var_gender,font=("times new roman",21,"bold"),state='readonly')
        combo_gender['value']=('Male','Female','Other')
        combo_gender.current(0)
        combo_gender.grid(row=3,column=1)

        #Postcode--------------------------------
        lblpostcode=Label(labelframeleft,text='PostCode:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblpostcode.grid(row=4,column=0,sticky=W)
        txtpostcode=ttk.Entry(labelframeleft,width=24,textvariable=self.var_post,font=("times new roman",21,"bold"))
        txtpostcode.grid(row=4,column=1)

        #Mobilenumber----------------------------
        lblmobile=Label(labelframeleft,text='Mobile:',font=("times new roman",20,"bold"),padx=2,pady=6)
```



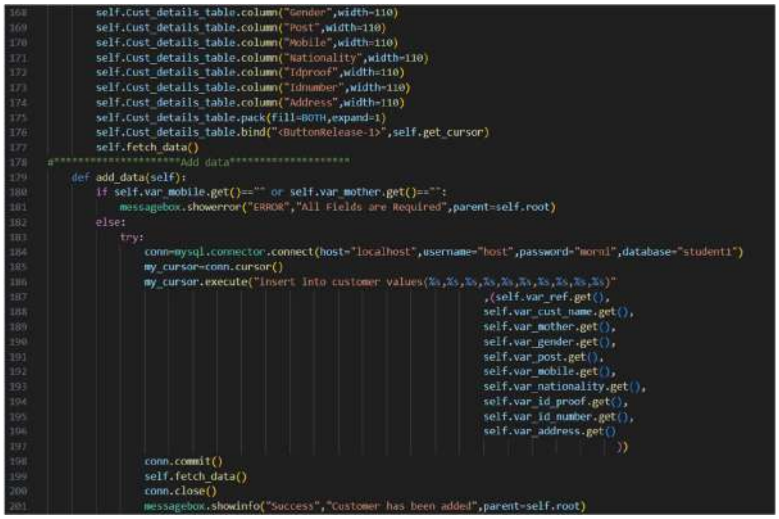
```
        #Mobilenumber----------------------------
        lblmobile=Label(labelframeleft,text='Mobile:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblmobile.grid(row=5,column=0,sticky=W)
        txtmobile=ttk.Entry(labelframeleft,width=24,textvariable=self.var_mobile,font=("times new roman",21,"bold"))
        txtmobile.grid(row=5,column=1)

        #Nationality-----------------------------
        lblnationality=Label(labelframeleft,text='Nationality:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblnationality.grid(row=7,column=0,sticky=W)
        combo_nationality=ttk.Combobox(labelframeleft,width=23,textvariable=self.var_nationality,font=("times new roman",21,"bold"),state='readonly')
        combo_nationality['value']=('Indian','American','Russian',)
        combo_nationality.current(0)
        combo_nationality.grid(row=7,column=1)

        #Idproof type combobox-------------------
        lblidproof=Label(labelframeleft,text='Id Proof Type:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblidproof.grid(row=8,column=0,sticky=W)
        combo_idproof=ttk.Combobox(labelframeleft,width=23,textvariable=self.var_id_proof,font=("times new roman",21,"bold"),state='readonly')
        combo_idproof['value']=('Aadhaarcard','Driving Licence','Passport')
        combo_idproof.current(0)
        combo_idproof.grid(row=8,column=1)

        #Idnumber--------------------------------
        lblidnumber=Label(labelframeleft,text='Id Number:',font=("times new roman",20,"bold"),padx=2,pady=6)
        lblidnumber.grid(row=9,column=0,sticky=W)
        txtidnumber=ttk.Entry(labelframeleft,width=24,textvariable=self.var_id_number,font=("times new roman",21,"bold"))
        txtidnumber.grid(row=9,column=1)

        #Address---------------------------------
        lbladdress=Label(labelframeleft,text='Address:',font=("times new roman",22,"bold"),padx=2,pady=6)
        lbladdress.grid(row=10,column=0,sticky=W)
        txtaddress=ttk.Entry(labelframeleft,width=24,textvariable=self.var_address,font=("times new roman",21,"bold"))
        txtaddress.grid(row=10,column=1)
```



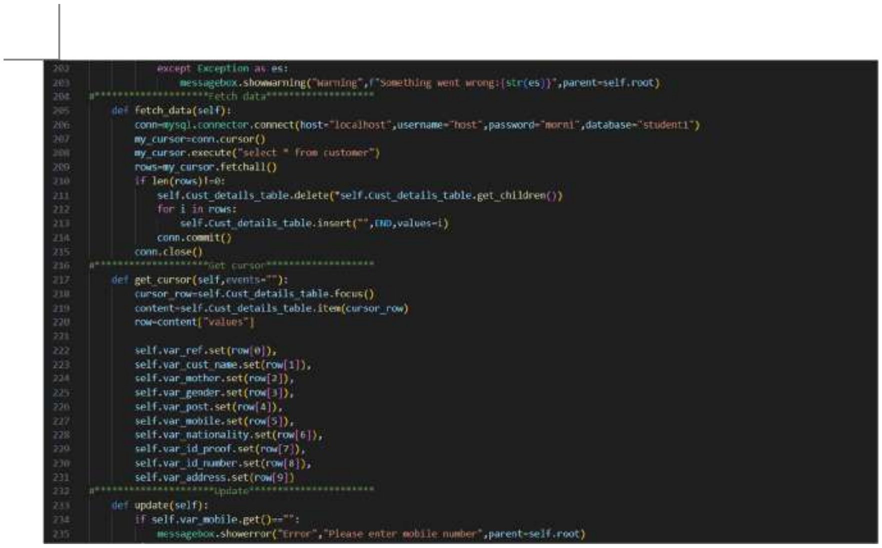
```
    #**********************Btns***********************
        btn_frame=Frame(labelframeleft,bd=1,relief=RIDGE)
        btn_frame.place(x=0,y=470,width=650,height=50)

        btnAdd=Button(btn_frame,text='ADD',command=self.add_data,font=("times new roman",18,"bold"),bg='black',fg='gold',width=9)
        btnAdd.grid(row=0,column=0,padx=1)

        btnupdate=Button(btn_frame,text='UPDATE',command=self.update,font=("times new roman",18,"bold"),bg='black',fg='gold',width=10)
        btnupdate.grid(row=0,column=1,padx=1)

        btndelete=Button(btn_frame,text='DELETE',command=self.mdelete,font=("times new roman",18,"bold"),bg='black',fg='gold',width=9)
        btndelete.grid(row=0,column=2,padx=1)

        btnreset=Button(btn_frame,text='RESET',command=self.reset,font=("times new roman",18,"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",22,"bold"),padx=2)
        table_frame.place(x=585,y=60,width=1110,height=690)

        lblsearchby=Label(table_frame,text='Search By:',font=("times new roman",22,"bold"),bg="red",fg="white")
        lblsearchby.grid(row=0,column=0,sticky=W,padx=1)
        self.search1_var=StringVar()
        combo_search=ttk.Combobox(table_frame,width=19,textvariable=self.search1_var,font=("times new roman",22,"bold"),state='readonly')
        combo_search['value']=('Name','Mobile')
        combo_search.current(0)
        combo_search.grid(row=0,column=1,padx=1)

        self.txt1_search=StringVar()
        txtsearch=ttk.Entry(table_frame,width=19,textvariable=self.txt1_search,font=("times new roman",22,"bold"))
        txtsearch.grid(row=0,column=2,padx=1)

        btnsearch=Button(table_frame,text='Search',command=self.search,font=("times new roman",18,"bold"),bg='black',fg='gold',width=10)
```



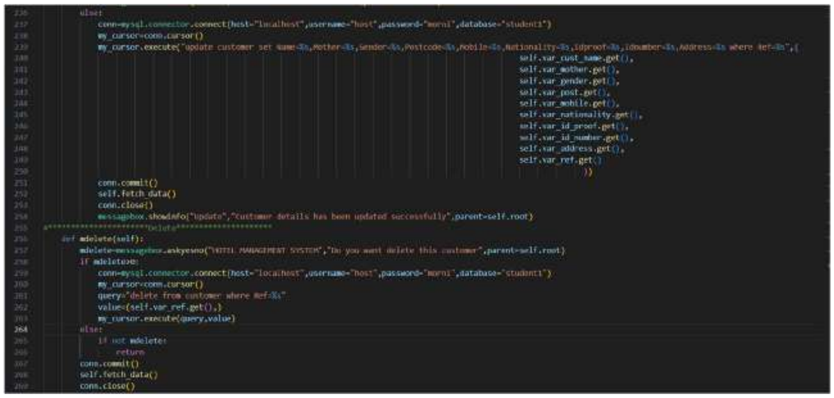
```
        btnsearch.grid(row=0,column=3,padx=1)

        btnshowall=Button(table_frame,text='Show All',command=self.fetch_data,font=("times new roman",18,"bold"),bg='black',fg='gold',width=10)
        btnshowall.grid(row=0,column=4,padx=1)
    #*****************Show data table*****************
        details_table=Frame(table_frame,bd=10,relief=RIDGE)
        details_table.place(x=0,y=45,width=1100,height=400)

        scroll_x=ttk.Scrollbar(details_table,orient=HORIZONTAL)
        scroll_y=ttk.Scrollbar(details_table,orient=VERTICAL)
        self.cust_details_table=ttk.Treeview(details_table,column=('Ref','Name','Mother','Gender','Post','Mobile','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="Post Code")
        self.Cust_details_table.heading('Mobile',text="Mobile")
        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=110)
        self.Cust_details_table.column("Name",width=110)
        self.Cust_details_table.column("Mother",width=110)
        self.Cust_details_table.column("Gender",width=110)
```



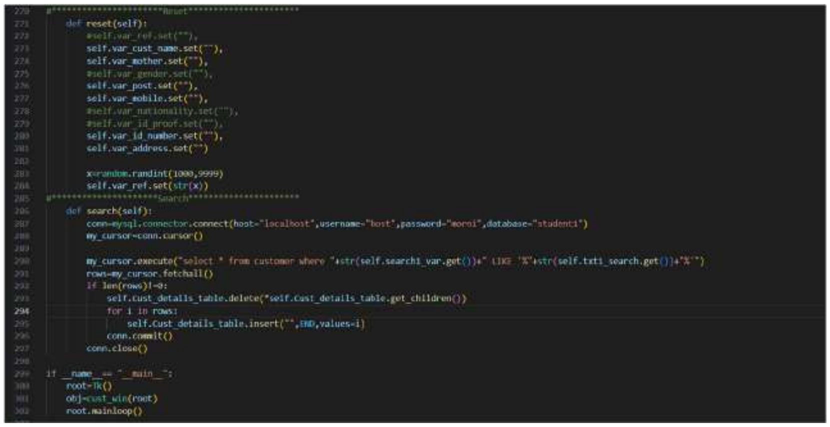
```
        self.Cust_details_table.column("Gender",width=110)
        self.Cust_details_table.column("Post",width=110)
        self.Cust_details_table.column("Mobile",width=110)
        self.Cust_details_table.column("Nationality",width=110)
        self.Cust_details_table.column("Idproof",width=110)
        self.Cust_details_table.column("Idnumber",width=110)
        self.Cust_details_table.column("Address",width=110)
        self.Cust_details_table.pack(fill=BOTH,expand=1)
        self.Cust_details_table.bind("<ButtonRelease-1>",self.get_cursor)
        self.fetch_data()
    #********************Add 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="host",password="mornl",database="student1")
                my_cursor=conn.cursor()
                my_cursor.execute("insert into customer values(%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_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)
#********************Fetch data*******************
    def fetch_data(self):
        conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
        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()
#********************Get cursor*******************
    def get_cursor(self,events=""):
        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_nationality.set(row[6]),
        self.var_id_proof.set(row[7]),
        self.var_id_number.set(row[8]),
        self.var_address.set(row[9])
#********************Update**********************
    def update(self):
        if self.var_mobile.get()=="":
            messagebox.showerror("Error","Please enter mobile number",parent=self.root)
```

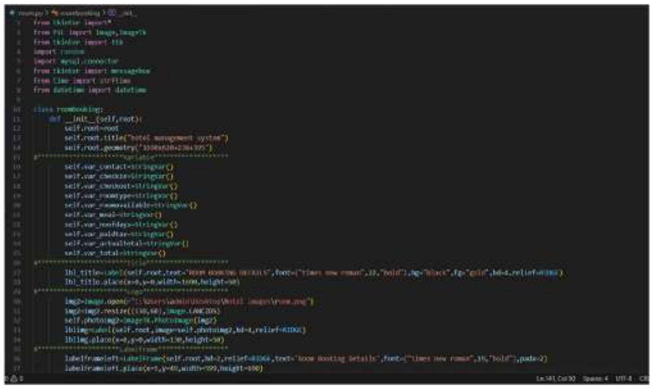


```
        else:
            conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
            my_cursor=conn.cursor()
            my_cursor.execute("update customer set Name=%s,Mother=%s,Gender=%s,Postcode=%s,Mobile=%s,Nationality=%s,Idproof=%s,Idnumber=%s,Address=%s 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_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)
#********************Delete*******************
    def mdelete(self):
        mdelete=messagebox.askyesno("HOTEL MANAGEMENT SYSTEM","Do you want delete this customer",parent=self.root)
        if mdelete>0:
            conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
            my_cursor=conn.cursor()
            query="delete from customer where Ref=%s"
            value=(self.var_ref.get(),)
            my_cursor.execute(query,value)
        else:
            if not mdelete:
                return
        conn.commit()
        self.fetch_data()
        conn.close()
```

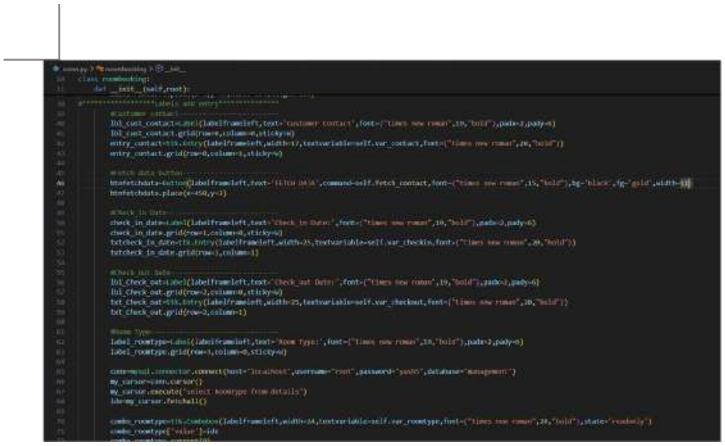


```
#********************Reset*********************
    def reset(self):
        #self.var_ref.set(""),
        self.var_cust_name.set(""),
        self.var_mother.set(""),
        #self.var_gender.set(""),
        self.var_post.set(""),
        self.var_mobile.set(""),
        #self.var_nationality.set(""),
        #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))
#********************Search*********************
    def search(self):
        conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
        my_cursor=conn.cursor()

        my_cursor.execute("select * from customer where "+str(self.search1_var.get())+" LIKE '%"+str(self.txt1_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=cust_win(root)
    root.mainloop()
```

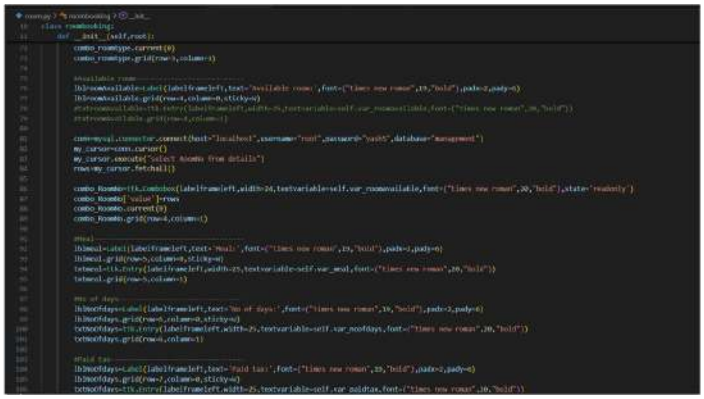
Room Program:



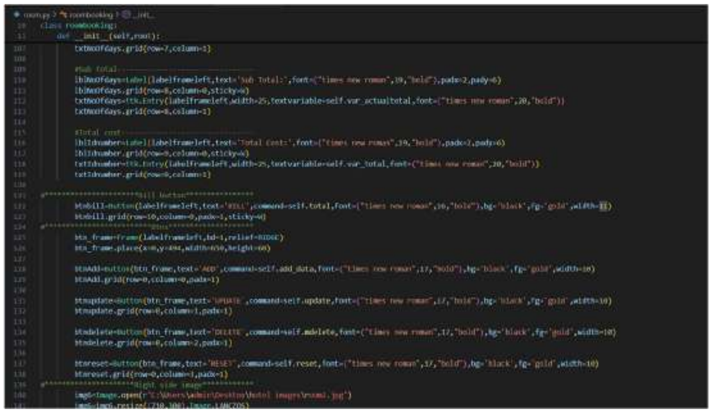
```
from tkinter import*
from PIL import Image,ImageTk
from tkinter import ttk
import random
import mysql.connector
from tkinter import messagebox
from time import strftime
from datetime import datetime

class roombooking:
    def __init__(self,root):
        self.root=root
        self.root.title("hotel management system")
        self.root.geometry("1300x520+230+335")
#**********************variable*****************
        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 DETAILS",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
        lbl_title.place(x=0,y=0,width=1300,height=50)
#**********************logo*********************
        img2=Image.open(r"C:\Users\admin\Desktop\hotel images\room.png")
        img2=img2.resize((130,60),Image.LANCZOS)
        self.photoimg2=ImageTk.PhotoImage(img2)
        lblimg=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=130,height=50)
#**********************labelframe****************
        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text="Room Booking Details",font=("times new roman",15,"bold"),padx=2)
        labelframeleft.place(x=5,y=49,width=599,height=490)
```



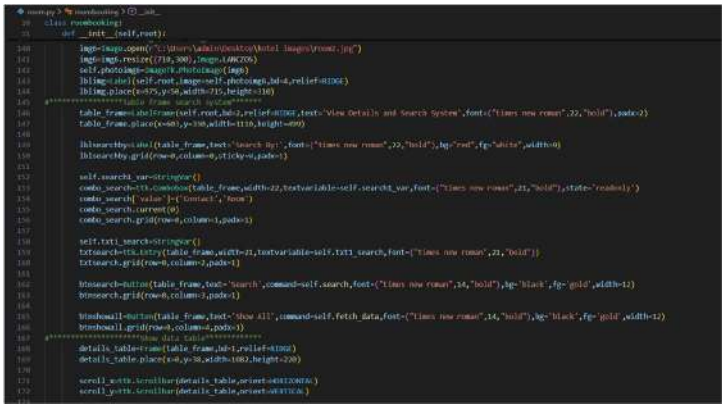
```
class roombooking:
    def __init__(self,root):
#******************Labels and entry****************
        #Customer contact--------------------------
        lbl_cust_contact=Label(labelframeleft,text='Customer contact',font=("times new roman",19,"bold"),padx=2,pady=6)
        lbl_cust_contact.grid(row=0,column=0,sticky=W)
        entry_contact=ttk.Entry(labelframeleft,width=17,textvariable=self.var_contact,font=("times new roman",20,"bold"))
        entry_contact.grid(row=0,column=1,sticky=W)

        #Fetch data button---------------------
        btnfetchdata=Button(labelframeleft,text='FETCH DATA',command=self.fetch_contact,font=("times new roman",15,"bold"),bg='black',fg='gold',width=11)
        btnfetchdata.place(x=450,y=3)

        #Check_in Date-------------------------
        check_in_date=Label(labelframeleft,text='Check_in Date:',font=("times new roman",19,"bold"),padx=2,pady=6)
        check_in_date.grid(row=1,column=0,sticky=W)
        txtcheck_in_date=ttk.Entry(labelframeleft,width=25,textvariable=self.var_checkin,font=("times new roman",20,"bold"))
        txtcheck_in_date.grid(row=1,column=1)

        #Check out Date------------------------
        lbl_Check_out=Label(labelframeleft,text='Check_out Date:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lbl_Check_out.grid(row=2,column=0,sticky=W)
        txt_Check_out=ttk.Entry(labelframeleft,width=25,textvariable=self.var_checkout,font=("times new roman",20,"bold"))
        txt_Check_out.grid(row=2,column=1)

        #Room Type-----------------------------
        label_roomtype=Label(labelframeleft,text='Room Type:',font=("times new roman",19,"bold"),padx=2,pady=6)
        label_roomtype.grid(row=3,column=0,sticky=W)

        conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
        my_cursor=conn.cursor()
        my_cursor.execute("select RoomType from details")
        ide=my_cursor.fetchall()

        combo_roomtype=ttk.Combobox(labelframeleft,width=24,textvariable=self.var_roomtype,font=("times new roman",20,"bold"),state='readonly')
        combo_roomtype['value']=ide
```



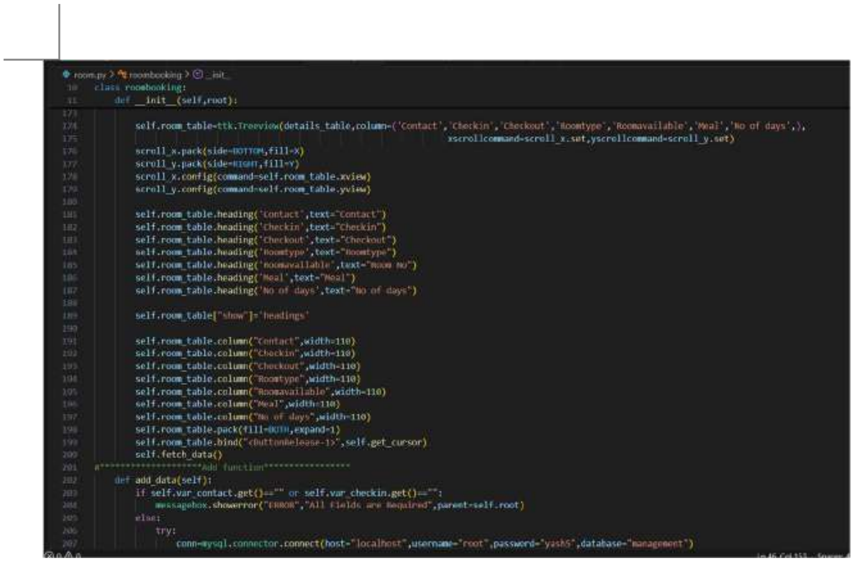
```
class roombooking:
    def __init__(self,root):
        combo_roomtype.current(0)
        combo_roomtype.grid(row=3,column=1)

        #Available rooms-----------------------
        lblroomAvailable=Label(labelframeleft,text='Available rooms:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblroomAvailable.grid(row=4,column=0,sticky=W)
        #txtroomavailable=ttk.Entry(labelframeleft,width=25,textvariable=self.var_roomavailable,font=("times new roman",20,"bold"))
        #txtroomavailable.grid(row=4,column=1)

        conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
        my_cursor=conn.cursor()
        my_cursor.execute("select RoomNo from details")
        rows=my_cursor.fetchall()

        combo_RoomNo=ttk.Combobox(labelframeleft,width=24,textvariable=self.var_roomavailable,font=("times new roman",20,"bold"),state='readonly')
        combo_RoomNo['value']=rows
        combo_RoomNo.current(0)
        combo_RoomNo.grid(row=4,column=1)

        #Meal----------------------------------
        lblmeal=Label(labelframeleft,text='Meal:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblmeal.grid(row=5,column=0,sticky=W)
        txtmeal=ttk.Entry(labelframeleft,width=25,textvariable=self.var_meal,font=("times new roman",20,"bold"))
        txtmeal.grid(row=5,column=1)

        #No of days----------------------------
        lblNoOfdays=Label(labelframeleft,text='No of days:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblNoOfdays.grid(row=6,column=0,sticky=W)
        txtNoOfdays=ttk.Entry(labelframeleft,width=25,textvariable=self.var_noofdays,font=("times new roman",20,"bold"))
        txtNoOfdays.grid(row=6,column=1)

        #Paid tax------------------------------
        lblNoOfdays=Label(labelframeleft,text='Paid tax:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblNoOfdays.grid(row=7,column=0,sticky=W)
        txtNoOfdays=ttk.Entry(labelframeleft,width=25,textvariable=self.var_paidtax,font=("times new roman",20,"bold"))
```



```
class roombooking:
    def __init__(self,root):
        txtNoOfdays.grid(row=7,column=1)

        #Sub Total-----------------------------
        lblNoOfdays=Label(labelframeleft,text='Sub Total:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblNoOfdays.grid(row=8,column=0,sticky=W)
        txtNoOfdays=ttk.Entry(labelframeleft,width=25,textvariable=self.var_actualtotal,font=("times new roman",20,"bold"))
        txtNoOfdays.grid(row=8,column=1)

        #Total cost----------------------------
        lblIdnumber=Label(labelframeleft,text='Total Cost:',font=("times new roman",19,"bold"),padx=2,pady=6)
        lblIdnumber.grid(row=9,column=0,sticky=W)
        txtIdnumber=ttk.Entry(labelframeleft,width=25,textvariable=self.var_total,font=("times new roman",20,"bold"))
        txtIdnumber.grid(row=9,column=1)

#**********************Bill button***************
        btnbill=Button(labelframeleft,text='BILL',command=self.total,font=("times new roman",16,"bold"),bg='black',fg='gold',width=11)
        btnbill.grid(row=10,column=0,padx=1,sticky=W)
#*************************Btns*******************
        btn_frame=Frame(labelframeleft,bd=1,relief=RIDGE)
        btn_frame.place(x=0,y=494,width=650,height=40)

        btnAdd=Button(btn_frame,text='ADD',command=self.add_data,font=("times new roman",17,"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=("times new roman",17,"bold"),bg='black',fg='gold',width=10)
        btnupdate.grid(row=0,column=1,padx=1)

        btndelete=Button(btn_frame,text='DELETE',command=self.mdelete,font=("times new roman",17,"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=("times new roman",17,"bold"),bg='black',fg='gold',width=10)
        btnreset.grid(row=0,column=3,padx=1)
#**********************Right side image***********
        img6=Image.open(r"C:\Users\admin\Desktop\hotel images\room2.jpg")
```

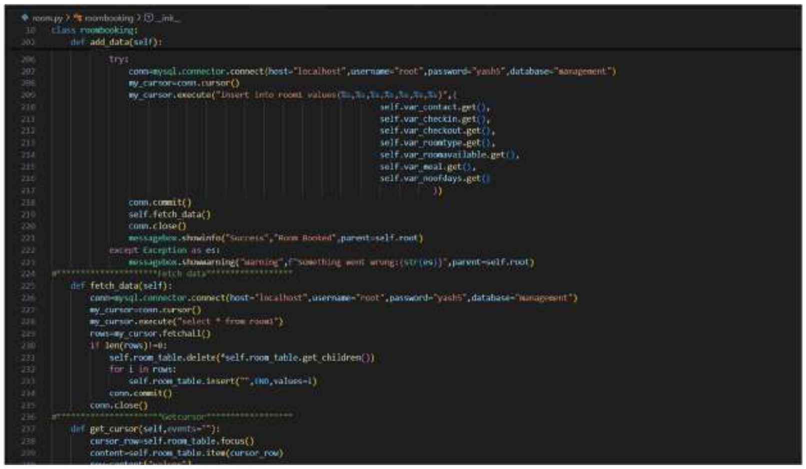


```
class roombooking:
    def __init__(self,root):
        img6=Image.open(r"C:\Users\admin\Desktop\hotel images\room2.jpg")
        img6=img6.resize((710,300),Image.LANCZOS)
        self.photoimg6=ImageTk.PhotoImage(img6)
        lblimg=Label(self.root,image=self.photoimg6,bd=4,relief=RIDGE)
        lblimg.place(x=975,y=50,width=715,height=310)
#******************table frame search system*******
        table_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text='View Details and Search System',font=("times new roman",22,"bold"),padx=2)
        table_frame.place(x=603,y=390,width=1110,height=499)

        lblsearchby=Label(table_frame,text='Search By:',font=("times new roman",22,"bold"),bg="red",fg="white",width=9)
        lblsearchby.grid(row=0,column=0,sticky=W,padx=1)

        self.search1_var=StringVar()
        combo_search=ttk.Combobox(table_frame,width=22,textvariable=self.search1_var,font=("times new roman",21,"bold"),state='readonly')
        combo_search['value']=('Contact','Room')
        combo_search.current(0)
        combo_search.grid(row=0,column=1,padx=1)

        self.txt1_search=StringVar()
        txtsearch=ttk.Entry(table_frame,width=21,textvariable=self.txt1_search,font=("times new roman",21,"bold"))
        txtsearch.grid(row=0,column=2,padx=1)

        btnsearch=Button(table_frame,text='Search',command=self.search,font=("times new roman",14,"bold"),bg='black',fg='gold',width=12)
        btnsearch.grid(row=0,column=3,padx=1)

        btnshowall=Button(table_frame,text='Show All',command=self.fetch_data,font=("times new roman",14,"bold"),bg='black',fg='gold',width=12)
        btnshowall.grid(row=0,column=4,padx=1)
#*********************Show data Table************
        details_table=Frame(table_frame,bd=1,relief=RIDGE)
        details_table.place(x=0,y=38,width=1082,height=220)

        scroll_x=ttk.Scrollbar(details_table,orient=HORIZONTAL)
        scroll_y=ttk.Scrollbar(details_table,orient=VERTICAL)
```



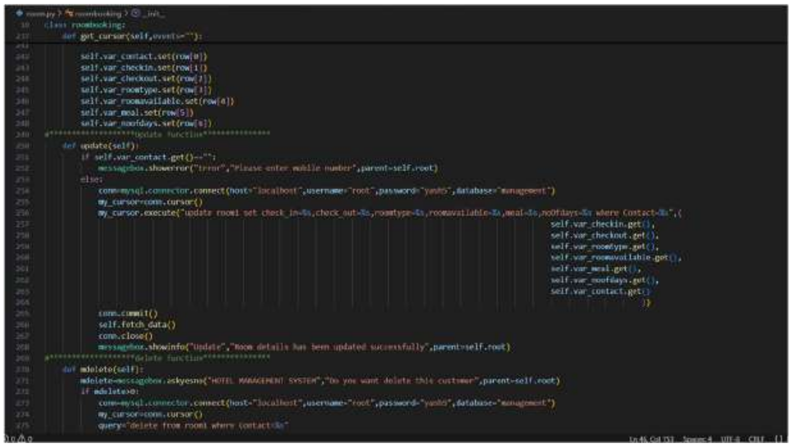
room.py > roombooking > __init__

```
class roombooking:
    def __init__(self,root):

        self.room_table=ttk.Treeview(details_table,column=('Contact','Checkin','Checkout','Roomtype','Roomavailable','Meal','No of days',),
                                                            xscrollcommand=scroll_x.set,yscrollcommand=scroll_y.set)
        scroll_x.pack(side=BOTTOM,fill=X)
        scroll_y.pack(side=RIGHT,fill=Y)
        scroll_x.config(command=self.room_table.xview)
        scroll_y.config(command=self.room_table.yview)

        self.room_table.heading('Contact',text="Contact")
        self.room_table.heading('Checkin',text="Checkin")
        self.room_table.heading('Checkout',text="Checkout")
        self.room_table.heading('Roomtype',text="Roomtype")
        self.room_table.heading('Roomavailable',text="Room No")
        self.room_table.heading('Meal',text="Meal")
        self.room_table.heading('No of days',text="No of days")

        self.room_table["show"]='headings'

        self.room_table.column("Contact",width=110)
        self.room_table.column("Checkin",width=110)
        self.room_table.column("Checkout",width=110)
        self.room_table.column("Roomtype",width=110)
        self.room_table.column("Roomavailable",width=110)
        self.room_table.column("Meal",width=110)
        self.room_table.column("No of days",width=110)
        self.room_table.pack(fill=BOTH,expand=1)
        self.room_table.bind("<ButtonRelease-1>",self.get_cursor)
        self.fetch_data()
#*******************Add function****************
    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="yash5",database="management")
```



room.py > roombooking > __init__

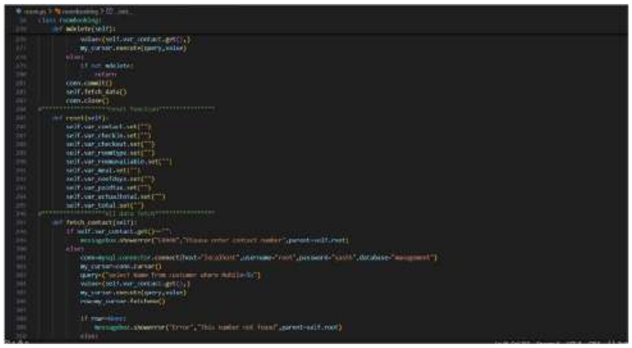
```
class roombooking:
    def add_data(self):
            try:
                conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
                my_cursor=conn.cursor()
                my_cursor.execute("insert into room1 values(%s,%s,%s,%s,%s,%s,%s)",(
                                                                        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="yash5",database="management")
        my_cursor=conn.cursor()
        my_cursor.execute("select * from room1")
        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()
#*********************Getcursor*****************
    def get_cursor(self,events=""):
        cursor_row=self.room_table.focus()
        content=self.room_table.item(cursor_row)
```



room.py > roombooking > __init__

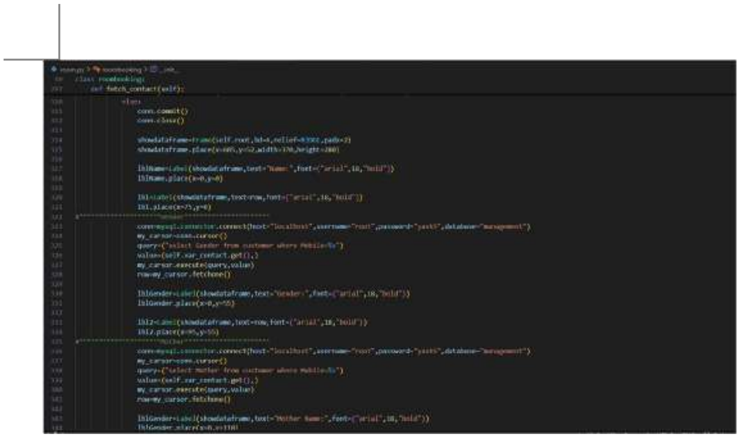
```
class roombooking:
    def get_cursor(self,events=""):
        self.var_contact.set(row[0])
        self.var_checkin.set(row[1])
        self.var_checkout.set(row[2])
        self.var_roomtype.set(row[3])
        self.var_roomavailable.set(row[4])
        self.var_meal.set(row[5])
        self.var_noofdays.set(row[6])
#******************Update function**************
    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="yash5",database="management")
            my_cursor=conn.cursor()
            my_cursor.execute("update room1 set check_in=%s,check_out=%s,roomtype=%s,roomavailable=%s,meal=%s,noOfdays=%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)
#******************Delete function**************
    def mdelete(self):
        mdelete=messagebox.askyesno("HOTEL MANAGEMENT SYSTEM","Do you want delete this customer",parent=self.root)
        if mdelete>0:
            conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
            my_cursor=conn.cursor()
            query="delete from room1 where Contact=%s"
```

Ln 46, Col 153   Spaces: 4   UTF-8   CRLF

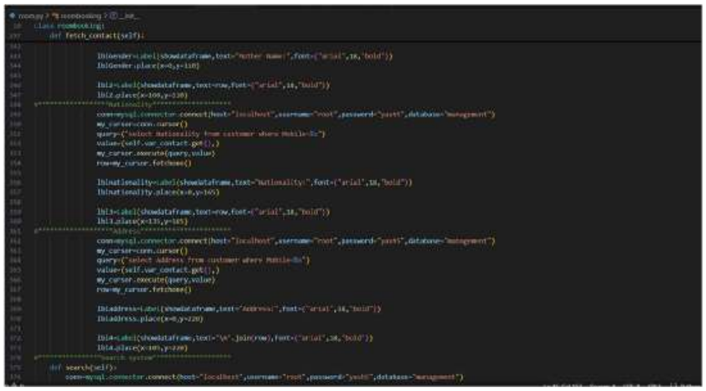


room.py > roombooking > __init__

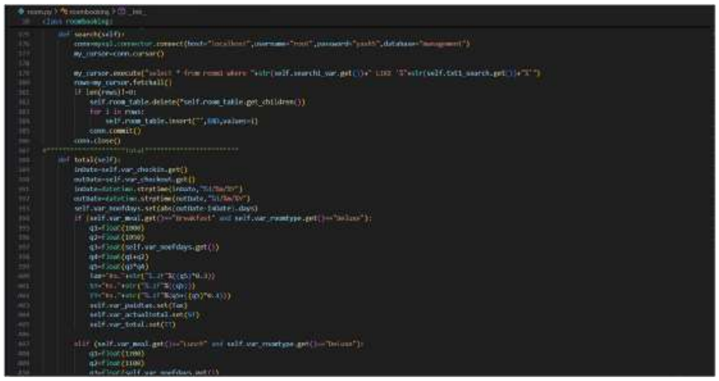
```
class roombooking:
    def mdelete(self):
            value=(self.var_contact.get(),)
            my_cursor.execute(query,value)
        else:
            if not mdelete:
                return
        conn.commit()
        self.fetch_data()
        conn.close()
#********************Reset function**************
    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="yash5",database="management")
            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:
```



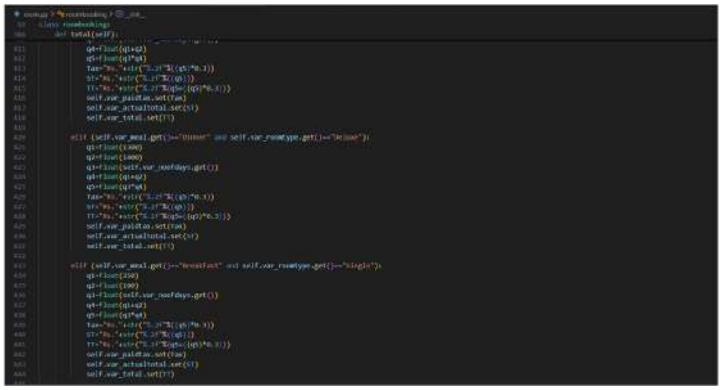
```
class roombooking:
    def fetch_contact(self):
                else:
                    conn.commit()
                    conn.close()

                    showdataframe=Frame(self.root,bd=4,relief=RIDGE,padx=2)
                    showdataframe.place(x=805,y=52,width=370,height=280)

                    lblName=Label(showdataframe,text="Name:",font=("arial",18,"bold"))
                    lblName.place(x=0,y=0)

                    lbl=Label(showdataframe,text=row,font=("arial",18,"bold"))
                    lbl.place(x=75,y=0)
    #*********************Gender**********************
                    conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
                    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",18,"bold"))
                    lblGender.place(x=0,y=55)

                    lbl2=Label(showdataframe,text=row,font=("arial",18,"bold"))
                    lbl2.place(x=95,y=55)
    #*********************Mother**********************
                    conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
                    my_cursor=conn.cursor()
                    query=("select Mother from customer where Mobile=%s")
                    value=(self.var_contact.get(),)
                    my_cursor.execute(query,value)
                    row=my_cursor.fetchone()

                    lblGender=Label(showdataframe,text="Mother Name:",font=("arial",18,"bold"))
                    lblGender.place(x=0,y=110)
```



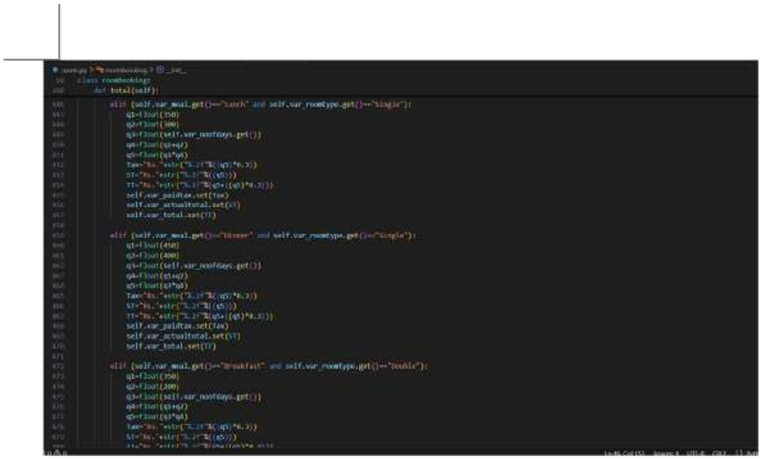
```
class roombooking:
    def fetch_contact(self):

                    lblGender=Label(showdataframe,text="Mother Name:",font=("arial",18,"bold"))
                    lblGender.place(x=0,y=110)

                    lbl2=Label(showdataframe,text=row,font=("arial",18,"bold"))
                    lbl2.place(x=160,y=110)
    #*******************Nationality********************
                    conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
                    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",18,"bold"))
                    lblnationality.place(x=0,y=165)

                    lbl3=Label(showdataframe,text=row,font=("arial",18,"bold"))
                    lbl3.place(x=135,y=165)
    #*******************Address********************
                    conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
                    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",18,"bold"))
                    lbladdress.place(x=0,y=220)

                    lbl4=Label(showdataframe,text="\n".join(row),font=("arial",18,"bold"))
                    lbl4.place(x=105,y=220)
    #*******************search system********************
        def search(self):
            conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
```



```
class roombooking:
        def search(self):
            conn=mysql.connector.connect(host="localhost",username="root",password="yash5",database="management")
            my_cursor=conn.cursor()

            my_cursor.execute("select * from room1 where "+str(self.search1_var.get())+" LIKE '%"+str(self.txt1_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()
    #*********************total***********************
        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()=="Deluxe"):
                q1=float(1000)
                q2=float(1050)
                q3=float(self.var_noofdays.get())
                q4=float(q1+q2)
                q5=float(q3*q4)
                Tax="Rs."+str("%.2f"%((q5)*0.3))
                ST="Rs."+str("%.2f"%((q5)))
                TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
                self.var_paidtax.set(Tax)
                self.var_actualtotal.set(ST)
                self.var_total.set(TT)

            elif (self.var_meal.get()=="Lunch" and self.var_roomtype.get()=="Deluxe"):
                q1=float(1000)
                q2=float(1100)
                q3=float(self.var_noofdays.get())
```

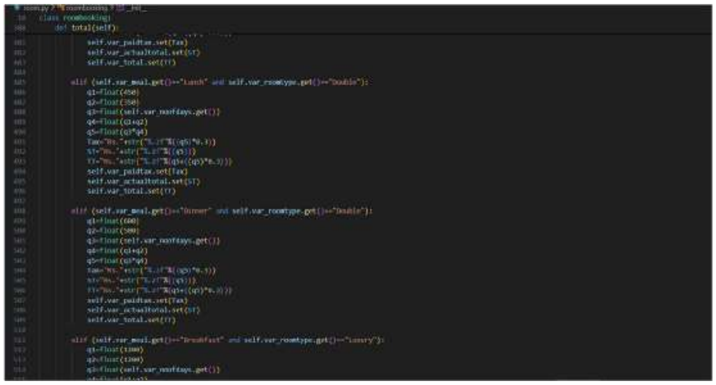


```
class roombooking:
        def total(self):
                q4=float(q1+q2)
                q5=float(q3*q4)
                Tax="Rs."+str("%.2f"%((q5)*0.3))
                ST="Rs."+str("%.2f"%((q5)))
                TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
                self.var_paidtax.set(Tax)
                self.var_actualtotal.set(ST)
                self.var_total.set(TT)

            elif (self.var_meal.get()=="Dinner" and self.var_roomtype.get()=="Deluxe"):
                q1=float(1300)
                q2=float(1400)
                q3=float(self.var_noofdays.get())
                q4=float(q1+q2)
                q5=float(q3*q4)
                Tax="Rs."+str("%.2f"%((q5)*0.3))
                ST="Rs."+str("%.2f"%((q5)))
                TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
                self.var_paidtax.set(Tax)
                self.var_actualtotal.set(ST)
                self.var_total.set(TT)

            elif (self.var_meal.get()=="Breakfast" and self.var_roomtype.get()=="Single"):
                q1=float(250)
                q2=float(100)
                q3=float(self.var_noofdays.get())
                q4=float(q1+q2)
                q5=float(q3*q4)
                Tax="Rs."+str("%.2f"%((q5)*0.3))
                ST="Rs."+str("%.2f"%((q5)))
                TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
                self.var_paidtax.set(Tax)
                self.var_actualtotal.set(ST)
                self.var_total.set(TT)
```



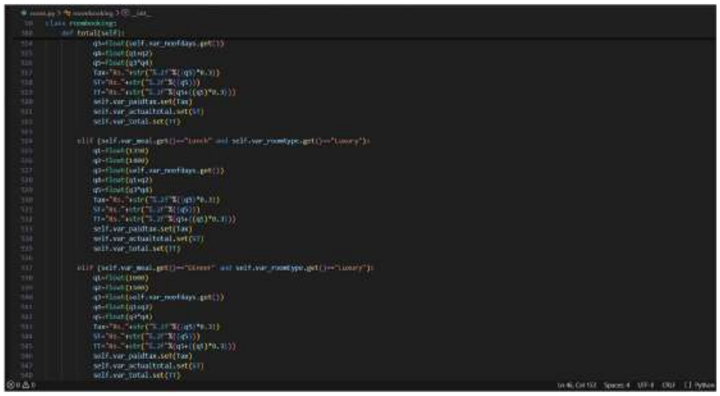
```
class roombooking:
    def total(self):
        elif (self.var_meal.get()=="Lunch" and self.var_roomtype.get()=="Single"):
            q1=float(350)
            q2=float(300)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Dinner" and self.var_roomtype.get()=="Single"):
            q1=float(450)
            q2=float(400)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Breakfast" and self.var_roomtype.get()=="Double"):
            q1=float(350)
            q2=float(200)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
```

Ln 46, Col 153 Spaces: 4 UTF-8 CRLF {} Python



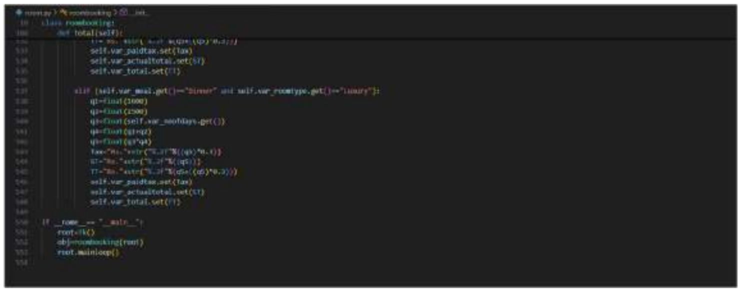
room.py > roombooking > __init__

```
class roombooking:
    def total(self):
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Lunch" and self.var_roomtype.get()=="Double"):
            q1=float(450)
            q2=float(350)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Dinner" and self.var_roomtype.get()=="Double"):
            q1=float(600)
            q2=float(500)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Breakfast" and self.var_roomtype.get()=="Luxury"):
            q1=float(1200)
            q2=float(1200)
            q3=float(self.var_noofdays.get())
```



```
class roombooking:
    def total(self):
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Lunch" and self.var_roomtype.get()=="Luxury"):
            q1=float(1300)
            q2=float(1000)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Dinner" and self.var_roomtype.get()=="Luxury"):
            q1=float(1600)
            q2=float(1500)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)
```

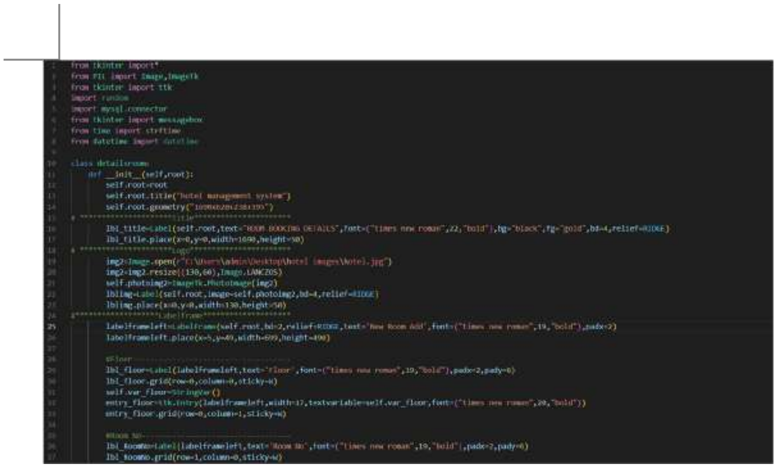
Ln 46, Col 153 Spaces: 4 UTF-8 CRLF {} Python



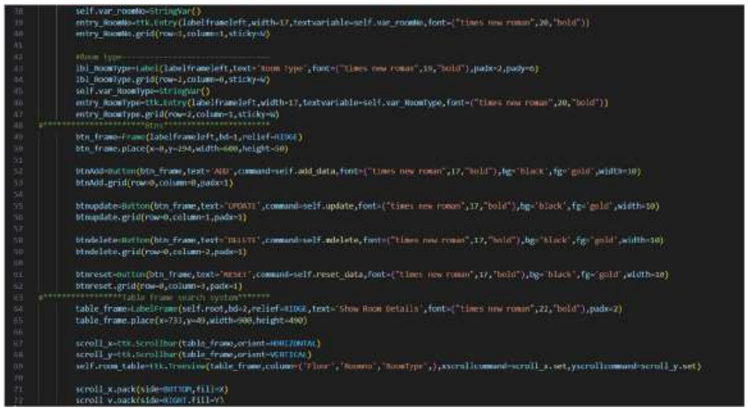
room.py > roombooking > __init__

```
class roombooking:
    def total(self):
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

        elif (self.var_meal.get()=="Dinner" and self.var_roomtype.get()=="Luxury"):
            q1=float(1600)
            q2=float(1500)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            Tax="Rs."+str("%.2f"%((q5)*0.3))
            ST="Rs."+str("%.2f"%((q5)))
            TT="Rs."+str("%.2f"%(q5+((q5)*0.3)))
            self.var_paidtax.set(Tax)
            self.var_actualtotal.set(ST)
            self.var_total.set(TT)

if __name__ == "__main__":
    root=Tk()
    obj=roombooking(root)
    root.mainloop()
```

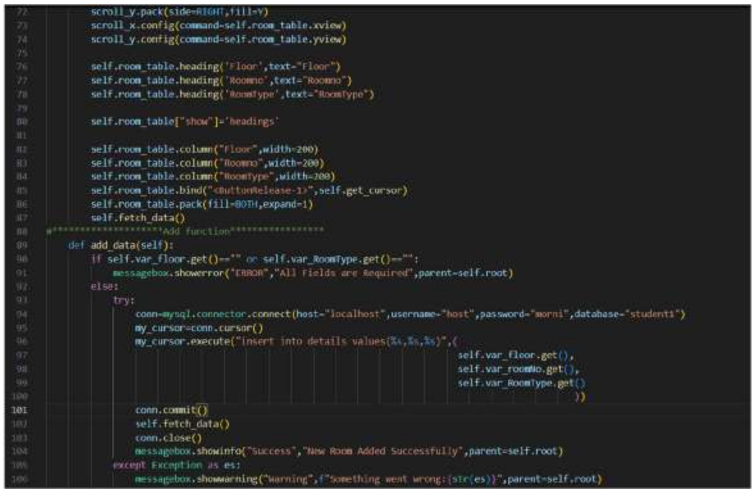
Details:



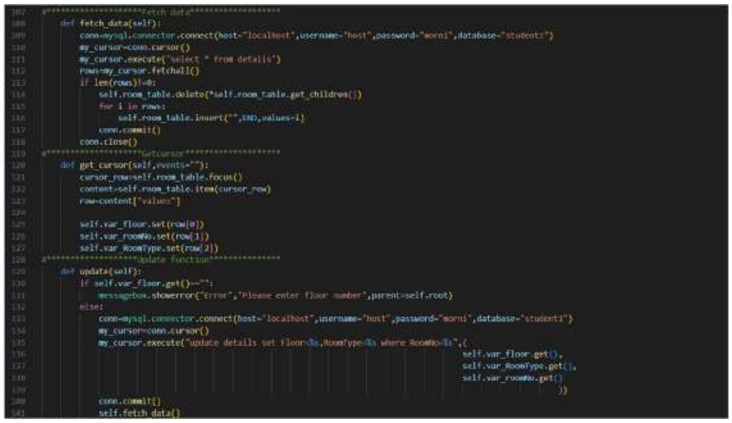
```
from tkinter import*
from PIL import Image,ImageTk
from tkinter import ttk
import random
import mysql.connector
from tkinter import messagebox
from time import strftime
from datetime import datetime

class detailsroom:
    def __init__(self,root):
        self.root=root
        self.root.title("hotel management system")
        self.root.geometry("1090x620+230+195")
# ************************title***********************
        lbl_title=Label(self.root,text="ROOM BOOKING DETAILS",font=("times new roman",22,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
        lbl_title.place(x=0,y=0,width=1090,height=50)
# ************************logo************************
        img2=Image.open(r"C:\Users\admin\Desktop\hotel images\hotel.jpg")
        img2=img2.resize((130,60),Image.LANCZOS)
        self.photoimg2=ImageTk.PhotoImage(img2)
        lblimg=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=130,height=50)
#*********************labelframe*********************
        labelframeleft=LabelFrame(self.root,bd=2,relief=RIDGE,text="New Room Add",font=("times new roman",19,"bold"),padx=2)
        labelframeleft.place(x=5,y=49,width=699,height=490)

        #Floor-------------------------------------
        lbl_floor=Label(labelframeleft,text="Floor",font=("times new roman",19,"bold"),padx=2,pady=6)
        lbl_floor.grid(row=0,column=0,sticky=W)
        self.var_floor=StringVar()
        entry_floor=ttk.Entry(labelframeleft,width=17,textvariable=self.var_floor,font=("times new roman",20,"bold"))
        entry_floor.grid(row=0,column=1,sticky=W)

        #Room No-----------------------------------
        lbl_RoomNo=Label(labelframeleft,text="Room No",font=("times new roman",19,"bold"),padx=2,pady=6)
        lbl_RoomNo.grid(row=1,column=0,sticky=W)
```



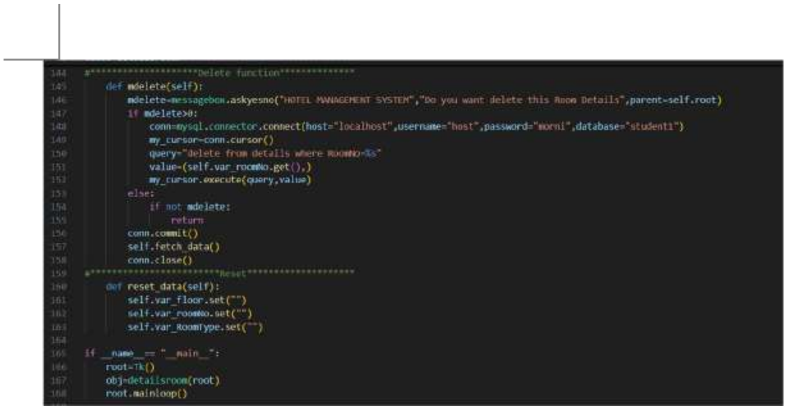
```
        self.var_roomNo=StringVar()
        entry_RoomNo=ttk.Entry(labelframeleft,width=17,textvariable=self.var_roomNo,font=("times new roman",20,"bold"))
        entry_RoomNo.grid(row=1,column=1,sticky=W)

        #Room Type---------------------------------
        lbl_RoomType=Label(labelframeleft,text="Room Type",font=("times new roman",19,"bold"),padx=2,pady=6)
        lbl_RoomType.grid(row=2,column=0,sticky=W)
        self.var_RoomType=StringVar()
        entry_RoomType=ttk.Entry(labelframeleft,width=17,textvariable=self.var_RoomType,font=("times new roman",20,"bold"))
        entry_RoomType.grid(row=2,column=1,sticky=W)
#**********************btns***********************
        btn_frame=Frame(labelframeleft,bd=1,relief=RIDGE)
        btn_frame.place(x=0,y=294,width=600,height=50)

        btnAdd=Button(btn_frame,text='ADD',command=self.add_data,font=("times new roman",17,"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=("times new roman",17,"bold"),bg='black',fg='gold',width=10)
        btnupdate.grid(row=0,column=1,padx=1)

        btndelete=Button(btn_frame,text='DELETE',command=self.mdelete,font=("times new roman",17,"bold"),bg='black',fg='gold',width=10)
        btndelete.grid(row=0,column=2,padx=1)

        btnreset=Button(btn_frame,text='RESET',command=self.reset_data,font=("times new roman",17,"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",22,"bold"),padx=2)
        table_frame.place(x=733,y=49,width=900,height=490)

        scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)
        scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)
        self.room_table=ttk.Treeview(table_frame,column=('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_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="Roomno")
        self.room_table.heading('RoomType',text="RoomType")

        self.room_table["show"]='headings'

        self.room_table.column("Floor",width=200)
        self.room_table.column("Roomno",width=200)
        self.room_table.column("RoomType",width=200)
        self.room_table.bind("<ButtonRelease-1>",self.get_cursor)
        self.room_table.pack(fill=BOTH,expand=1)
        self.fetch_data()
#*********************Add function*****************
    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="host",password="morni",database="student1")
                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)
```



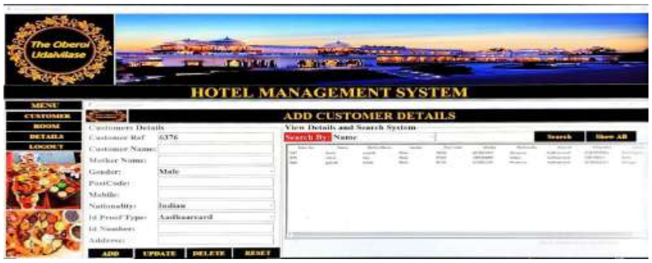
```
#********************Fetch data********************
    def fetch_data(self):
        conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
        my_cursor=conn.cursor()
        my_cursor.execute("select * from details")
        rows=my_cursor.fetchall()
        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()
#********************Getcursor*********************
    def get_cursor(self,events=""):
        cursor_row=self.room_table.focus()
        content=self.room_table.item(cursor_row)
        row=content["values"]

        self.var_floor.set(row[0])
        self.var_roomNo.set(row[1])
        self.var_RoomType.set(row[2])
#********************update function***************
    def update(self):
        if self.var_floor.get()=="":
            messagebox.showerror("Error","Please enter floor number",parent=self.root)
        else:
            conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
            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()
```



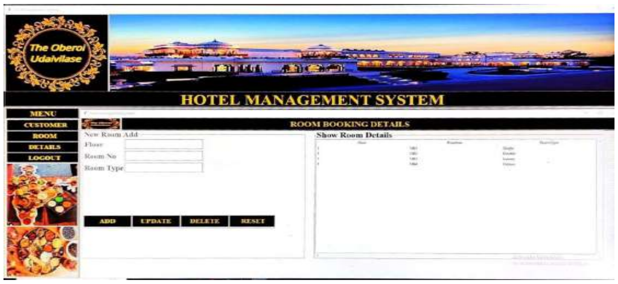
```
#********************Delete function*************
    def mdelete(self):
        mdelete=messagebox.askyesno("HOTEL MANAGEMENT SYSTEM","Do you want delete this Room Details",parent=self.root)
        if mdelete>0:
            conn=mysql.connector.connect(host="localhost",username="host",password="morni",database="student1")
            my_cursor=conn.cursor()
            query="delete from details where RoomNo=%s"
            value=(self.var_roomNo.get(),)
            my_cursor.execute(query,value)
        else:
            if not mdelete:
                return
        conn.commit()
        self.fetch_data()
        conn.close()
#************************Reset*******************
    def reset_data(self):
        self.var_floor.set("")
        self.var_roomNo.set("")
        self.var_RoomType.set("")

if __name__ == "__main__":
    root=Tk()
    obj=detailsroom(root)
    root.mainloop()
```

System Architecture:

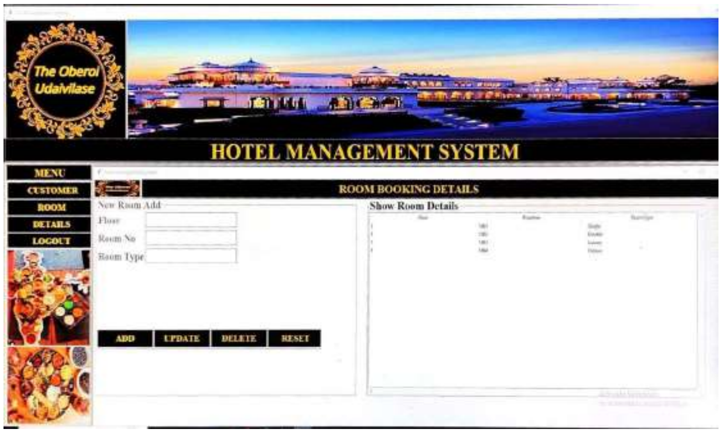
1. Presentation Layer (User Interface) built using Tkinter (Python’s standard GUI library
2. Functional Module: Customer, Room- Booking, Room- Detail
3. Databases: MySQL



Customer Ref: 6376
Customer Name:
Mother Name:
Gender: Male
PostCode:
Mobile:
Nationality: Indian
Id Proof Type: Aadhaarcard
Id Number:
Address:
Search By: Name



The Oberoi Udaivilase
HOTEL MANAGEMENT SYSTEM
MENU
CUSTOMER
ROOM
DETAILS
LOGOUT
ROOM BOOKING DETAILS
New Room Add
Floor
Room No
Room Type
Show Room Details
ADD
UPDATE
DELETE
RESET



The Oberoi Udaivilase
HOTEL MANAGEMENT SYSTEM
MENU
CUSTOMER
ROOM
DETAILS
LOGOUT
ROOM BOOKING DETAILS
New Room Add
Floor
Room No
Room Type
Show Room Details
ADD
UPDATE
DELETE
RESET