PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

PYTHON PROJECT REPORT 2021/2022

TITLE: HOTEL MANAGEMENT SYSTEM

TEAM MEMBER DETAILS:

PES1UG21CS029 – ADHESH DHEERAJ

PES1UG21EC038 - ANANYA RAVISHANKAR

ABSTRACT:

This is a project on the topic “Hotel Management System”. In this project we have made an attempt to design an interface for an efficient Hotel Management.

By using this software, we can store all the details of a customer in a database using python interface. Here we can:-

1. Store customer details in a database.

2. Store the room booking details.

3. Show the room amenities to the customer.

4. Show the restaurant menu(Buffet).

5. Calculate the total bill including the restaurant bill.

TABLE OF CONTENTS:

INTRODUCTION:

This projects provides for an efficient User Interface to manage a Hotel system. It allows the Hotel administration to view databases of the customers, which include their general details, their preferred date of check-in and check-out etc.

The projects starts off with a display of the Home Screen where the Hotel administer is given several options: The customer data base, The room booking details, The room amenities and the Restaurant Menu.

Once the user clicks on the "Customer" button, he is then displayed the number of customers, their room number, and their general details. One can also edit, delete or add data.

The "Room booking" button allows the user to view the details of the preferred choice of the customer such as his check-in and check-out dates, the number of days the customer wants to stay and the number of meals he prefers. The user is then informed about the total bill after calculation on the choice parameters mentioned above.

The user also has the option to view the "Room Amenities" and the "Restaurant Menu" for breakfast, lunch and dinner.

DESIGN / IMPLEMENTATION:

1. hotel.py (homescreen):

```
from tkinter import*

from PIL import Image,ImageTk


class HotelManagementSystem:

   def __init__(self,root):

       self.root=root

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

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




       ############Top Image##########

       img1=Image.open(r"C:\Users\Dr.N.T.Murali
                      Mohan\Desktop\Hotel project pics\hotel1.jpg")

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

       self.photoimg1=ImageTk.PhotoImage(img1)
```

```
label1=Label(self.root,image=self.photoimg1,bd=4,relief=RIDGE)

label1.place(x=0,y=0,width=1550,height=140)



##########LOGO##########

img2=Image.open(r"C:\Users\Dr.N.T.Murali
        Mohan\Desktop\Hotel project pics\hotel logo.jpg")

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

self.photoimg2=ImageTk.PhotoImage(img2)



label1=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)

label1.place(x=0,y=0,width=230,height=140)



#####Title########

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

lbtitle.place(x=0,y=140,width=1550,height=50)




#####Main frame#####

mframe=Frame(self.root,bd=4,relief=RIDGE)

mframe.place(x=0,y=190,width=1550,height=620)
```

```
#######menu######
menu1=Label(mframe,text="MENU",font=("times new roman",20,"bold"),bg="black",fg="blue",bd=4,relief=RIDGE)
menu1.place(x=0,y=0,width=230)
#####Buttonframe######
btframe=Frame(mframe,bd=4,relief=RIDGE)
btframe.place(x=0,y=35,width=228,height=190)



#####Buttons####

custbt=Button(btframe,text="CUSTOMER",width=22,font=("times new roman",14,"bold"),bg="black",fg="blue",bd=0,relief=RIDGE,cursor="hand2")
custbt.grid(row=0,column=0,pady=1)


roombt=Button(btframe,text="ROOMS",width=22,font=("times new roman",14,"bold"),bg="black",fg="blue",bd=0,relief=RIDGE,cursor="hand2")
roombt.grid(row=1,column=0,pady=1)
```

```
detailsbt=Button(btframe,text="DETAILS",width=22,font=("times new roman",14,"bold"),bg="black",fg="blue",bd=0,relief=RIDGE,cursor="hand2")

detailsbt.grid(row=2,column=0,pady=1)



restbt=Button(btframe,text="RESTAURANT MENU",width=22,font=("times new roman",14,"bold"),bg="black",fg="blue",bd=0,relief=RIDGE,cursor="hand2")



restbt.grid(row=3,column=0,pady=1)


logoutbt=Button(btframe,text="LOGOUT",width=22,font=("times new roman",14,"bold"),bg="black",fg="blue",bd=0,relief=RIDGE,cursor="hand2")

logoutbt.grid(row=4,column=0,pady=1)




   ####main frame img#######

img3=Image.open(r"C:\Users\Dr.N.T.Murali Mohan\Desktop\Hotel project pics\hotel 2.jpg")

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

self.photoimg3=ImageTk.PhotoImage(img3)
```

```
label2=Label(mframe,image=self.photoimg3,bd=4,relief=RIDGE)

label2.place(x=225,y=0,width=1310,height=590)


  #######down images#####


img4=Image.open(r"C:\Users\Dr.N.T.Murali Mohan\Desktop\Hotel project pics\bed.jpg")

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

self.photoimg4=ImageTk.PhotoImage(img4)


label3=Label(mframe,image=self.photoimg4,bd=4,relief=RIDGE)

label3.place(x=0,y=225,width=230,height=210)


img5=Image.open(r"C:\Users\Dr.N.T.Murali Mohan\Desktop\Hotel project pics\restaurant.jpg")

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

self.photoimg5=ImageTk.PhotoImage(img5)


label4=Label(mframe,image=self.photoimg5,bd=4,relief=RIDGE)

label4.place(x=0,y=420,width=230,height=190)
```

```
if __name__=="__main__":

    root=Tk()

    obj=HotelManagementSystem(root)

    root.mainloop()
```

2. customer.py(customer details):

```
from tkinter import*

from tkinter.ttk import Combobox, Treeview

from PIL import Image,ImageTk

from tkinter import messagebox

import random

import mysql.connector

class CustWin:

    def __init__(self,root):

        self.root=root

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

        self.root.geometry("1295x550+230+220")



        #####Variables#####
```

```
 self.var_ref=StringVar()

 x=random.randint(100,999)

 self.var_ref.set(str(x))



 self.var_cust_name=StringVar()

 self.var_gender=StringVar()

 self.var_mobile=StringVar()

 self.var_email=StringVar()

 self.var_address=StringVar()

 self.var_idproof=StringVar()

 self.var_idno=StringVar()

 self.var_nat=StringVar()



 #######Title########

 lbtitle=Label(self.root,text="ENTER CUSTOMER
 DETAILS",font=("times new
 roman",18,"bold"),bg="black",fg="blue",bd=4,relief=RIDGE)

 lbtitle.place(x=0,y=0,width=1295,height=50)



#############LOGO##########

 img2=Image.open(r"C:\Users\Dr.N.T.Murali
 Mohan\Desktop\Hotel project pics\hotel logo.jpg")

 img2=img2.resize((100,40),Image.ANTIALIAS)
 self.photoimg2=ImageTk.PhotoImage(img2)


 label1=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)

 label1.place(x=5,y=2,width=100,height=40)





 #######labelframe#########

 lbl=LabelFrame(self.root,bd=2,relief=RIDGE,text="Customer
 Details",font=("arial",12,"bold"),padx=2,pady=6)

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



 ######Labels and entry#######


 ##Custref

 lbl_cust_ref=Label(lbl,text="Customer Reference
 No.",font=("arial",12,"bold"),padx=2,pady=6)

 lbl_cust_ref.grid(row=0,column=0,sticky=W)


entry_ref=Entry(lbl,font=("arial",13,"bold"),textvariable=self.var_r
ef,width=22,state="readonly")

 entry_ref.grid(row=0,column=1)
```

```
  ####Customer name######

  lbl_cust_name=Label(lbl,text="Customer
  Name",font=("arial",12,"bold"),padx=2,pady=6)

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



txtname=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=s
elf.var_cust_name)

  txtname.grid(row=1,column=1)


  #Gender combobox


lbl_cust_gen=Label(lbl,text="Gender",font=("arial",12,"bold"),pad
x=2,pady=6)

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



combo_gen=Combobox(lbl,font=("arial",12,"bold"),width=20,state=
"readonly",textvariable=self.var_gender)

  combo_gen["value"]=("Male","Female","Other")

  combo_gen.current(0)

  combo_gen.grid(row=2,column=1)
```

```
 #Mobile no.

     lbl_cust_mob=Label(lbl,text="Mobile
     No.",font=("arial",12,"bold"),padx=2,pady=6)

     lbl_cust_mob.grid(row=3,column=0,sticky=W)



   entry_mob=Entry(lbl,font=("arial",13,"bold"),width=22,textvariabl
   e=self.var_mobile)

     entry_mob.grid(row=3,column=1)


     #Email id

     lbl_cust_email=Label(lbl,text="Email
ID",font=("arial",12,"bold"),padx=2,pady=6)

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



entry_email=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=
self.var_email)

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


     #Nationality
```

```
lbl_cust_nat=Label(lbl,text="Nationality",font=("arial",12,"bold"),padx=2,pady=6)

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


combo_nat=Combobox(lbl,font=("arial",12,"bold"),width=20,state="readonly",textvariable=self.var_nat)

        combo_nat["value"]=("Indian","American","British")

        combo_nat.current(0)

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




        #Address


lbl_cust_add=Label(lbl,text="Address",font=("arial",12,"bold"),padx=2,pady=6)

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




entry_add=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=self.var_address)

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



        #ID type
        lbl_cust_idtype=Label(lbl,text="ID Type:",font=("arial",12,"bold"),padx=2,pady=6)

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




combo_id=Combobox(lbl,font=("arial",12,"bold"),width=20,state="readonly",textvariable=self.var_idproof)

        combo_id["value"]=("Aadhar Card","Driving License","Passport")

        combo_id.current(0)

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



        #ID number

        lbl_cust_idnum=Label(lbl,text="ID Number",font=("arial",12,"bold"),padx=2,pady=6)

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



entry_idnum=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=self.var_idno)

        entry_idnum.grid(row=8,column=1)
```

```
   ########btns#########

    btn_frame=Frame(lbl,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="blue",width=10)

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


btndel=Button(btn_frame,text="DELETE",command=self.mDelete,fon
t=("arial",11,"bold"),bg="black",fg="blue",width=10)

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



btnreset=Button(btn_frame,text="RESET",command=self.reset,font=(
"arial",11,"bold"),bg="black",fg="blue",width=10)

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


    ######TableFrame######



Table_Frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="View
details",font=("arial",12,"bold"),padx=2,pady=6)

    Table_Frame.place(x=435,y=50,width=860,height=490)
```

```
        #Show data table

        details_table=Frame(Table_Frame,bd=2,relief=RIDGE)

        details_table.place(x=0,y=50,width=860,height=350)



        scroll_x=Scrollbar(details_table,orient=HORIZONTAL)

        scroll_y=Scrollbar(details_table,orient=VERTICAL)




    self.Cust_details_table=Treeview(details_table,column=("Ref","Name"
    ,"Gender","Mobile no.","Email ID","Nationality","Address","ID
    proof","ID
    no."),xscrollcommand=scroll_x.set,yscrollcommand=scroll_y.set)



        scroll_x.pack(side=BOTTOM,fill=X)

        scroll_y.pack(side=RIGHT,fill=X)



        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="Reference No.")

        self.Cust_details_table.heading("Name",text="Name")

        self.Cust_details_table.heading("Gender",text="Gender")

        self.Cust_details_table.heading("Mobile no.",text="Mobile No.")
```

```
self.Cust_details_table.heading("Email ID",text="Email ID")

self.Cust_details_table.heading("Nationality",text="Nationality")

self.Cust_details_table.heading("Address",text="Address")

self.Cust_details_table.heading("ID proof",text="ID proof")

self.Cust_details_table.heading("ID no.",text="ID No.")



self.Cust_details_table["show"]="headings"

self.Cust_details_table.pack(fill=BOTH,expand=1)



self.Cust_details_table.column("Ref",width=100)

self.Cust_details_table.column("Name",width=100)

self.Cust_details_table.column("Gender",width=100)

self.Cust_details_table.column("Mobile no.",width=100)

self.Cust_details_table.column("Email ID",width=100)

self.Cust_details_table.column("Nationality",width=100)

self.Cust_details_table.column("Address",width=100)

self.Cust_details_table.column("ID proof",width=100)

self.Cust_details_table.column("ID no.",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_cust_name.get()=="" :

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

    else:

      try:


conn=mysql.connector.connect(host="localhost",user="root",password
="1234",database="management",auth_plugin="mysql_native_passwo
rd")

        my_cursor=conn.cursor()

        my_cursor.execute("insert into customer
values(%s,%s,%s,%s,%s,%s,%s,%s,%s)",(

                                                self.var_ref.get(),

                                                self.var_cust_name.get(),

                                                self.var_gender.get(),

                                                self.var_mobile.get(),

                                                self.var_email.get(),

                                                self.var_nat.get(),

                                                self.var_address.get(),

                                                self.var_idproof.get(),

                                                self.var_idno.get()



          ))
```

```
                conn.commit()

                self.fetch_data()

                conn.close()

                messagebox.showinfo("Success","customer has been
added",parent=self.root)

            except Exception as es:

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




  def fetch_data(self):


conn=mysql.connector.connect(host="localhost",user="root",password
="1234",database="management",auth_plugin="mysql_native_passwo
rd")

      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_gender.set(row[2]),

        self.var_mobile.set(row[3]),

        self.var_email.set(row[4]),

        self.var_nat.set(row[5]),

        self.var_address.set(row[6]),

        self.var_idproof.set(row[7]),

        self.var_idno.set(row[8])


    def mDelete(self):

        mDelete=messagebox.askyesno("Hotel Management System","Do
you want to delete this customer?",parent=self.root)

        if mDelete>0:
```

```
conn=mysql.connector.connect(host="localhost",user="root",password
="1234",database="management",auth_plugin="mysql_native_passwo
rd")
            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()



    def reset(self):
        self.var_cust_name.set("")
        self.var_mobile.set("")
        self.var_email.set("")
        self.var_address.set("")
        self.var_idno.set("")


        x=random.randint(100,999)
```

```
self.var_ref.set(str(x))

if __name__=="__main__":

    root=Tk()

    obj=CustWin(root)

    root.mainloop()
```

3. room.py(Room Booking):

```
from tkinter import*

from tkinter.ttk import Combobox, Treeview

from PIL import Image,ImageTk

from tkinter import messagebox

import random

import mysql.connector

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("1295x550+230+220")
        #######Title########
        lbtitle=Label(self.root,text="ROOM BOOKING DETAILS",font=("times new roman",18,"bold"),bg="black",fg="blue",bd=4,relief=RIDGE)
        lbtitle.place(x=0,y=0,width=1295,height=50)
        ########LOGO##########
        img2=Image.open(r"C:\Users\Dr.N.T.Murali Mohan\Desktop\Hotel project pics\hotel logo.jpg")
        img2=img2.resize((100,40),Image.ANTIALIAS)
        self.photoimg2=ImageTk.PhotoImage(img2)

        label1=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        label1.place(x=5,y=2,width=100,height=40)

        ##Variables
        self.var_roomno=StringVar()
        x=random.randint(100,999)
        self.var_roomno.set(str(x))

        self.var_contact=StringVar()
        self.var_checkin=StringVar()
        self.var_checkout=StringVar()
        self.var_roomtype=StringVar()
```

```
 self.var_noofdays=StringVar()

    self.var_meal=StringVar()

    self.var_bill=StringVar()

    #######labelframe#########

    lbl=LabelFrame(self.root,bd=2,relief=RIDGE,text="Room booking Details",font=("arial",12,"bold"),padx=2,pady=6)

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



    ##Custcontact

    lbl_cust_contact=Label(lbl,text="Customer Contact no.",font=("arial",12,"bold"),padx=2,pady=6)

    lbl_cust_contact.grid(row=0,column=0,sticky=W)



entry_contact=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=self.var_contact)

    entry_contact.grid(row=0,column=1)



    ##Check in

    check_in_date=Label(lbl,text="Check-In Date",font=("arial",12,"bold"),padx=2,pady=6)

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

```
        entry_checkin=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=self.var_checkin)
        entry_checkin.grid(row=1,column=1)


        ##Check out
        check_out_date=Label(lbl,text="Check-Out Date",font=("arial",12,"bold"),padx=2,pady=6)
        check_out_date.grid(row=2,column=0,sticky=W)


        entry_checkout=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=self.var_checkout)
        entry_checkout.grid(row=2,column=1)


        ##Room type
        lbl_roomtype=Label(lbl,text="Room Type",font=("arial",12,"bold"),padx=2,pady=6)
        lbl_roomtype.grid(row=3,column=0,sticky=W)


        combo_roomtype=Combobox(lbl,font=("arial",12,"bold"),width=20,state="readonly",textvariable=self.var_roomtype)
        combo_roomtype["value"]=("Standard","Standard AC","Deluxe","Deluxe AC")
        combo_roomtype.current(0)
        combo_roomtype.grid(row=3,column=1)
        ##Meal

lbl_meal=Label(lbl,text="Meal",font=("arial",12,"bold"),padx=2,pady=6)
        lbl_meal.grid(row=4,column=0,sticky=W)



combo_meal=Combobox(lbl,font=("arial",12,"bold"),width=20,state="readonly",textvariable=self.var_meal)
        combo_meal["value"]=("Breakfast","Lunch","Dinner","All")
        combo_meal.current(0)
        combo_meal.grid(row=4,column=1)



        ##No. of days
        lbl_noofdays=Label(lbl,text="No. of days",font=("arial",12,"bold"),padx=2,pady=6)
        lbl_noofdays.grid(row=5,column=0,sticky=W)
```

```
entry_ref=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=sel
f.var_noofdays)

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

    ###Available room



    lbl_avr=Label(lbl,text="Room
no.",font=("arial",12,"bold"),padx=2,pady=6,)

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




entry_avr=Entry(lbl,font=("arial",13,"bold"),width=22,state="readonl
y",textvariable=self.var_roomno)

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



    ##Bill

    lbl_total=Label(lbl,text="Total
Bill",font=("arial",12,"bold"),padx=2,pady=6)

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




entry_total=Entry(lbl,font=("arial",13,"bold"),width=22,textvariable=s
elf.var_bill)

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

    ##Billlbtn###
```

```
btnbill=Button(lbl,text="BILL",command=self.total,font=("arial",11,"
bold"),bg="black",fg="blue",width=10)

    btnbill.grid(row=8,column=0,padx=1)


    ########btns#########


    btn_frame=Frame(lbl,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="blue",width=10)

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



btndel=Button(btn_frame,text="DELETE",command=self.mDelete,fon
t=("arial",11,"bold"),bg="black",fg="blue",width=10)

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



btnreset=Button(btn_frame,text="RESET",command=self.reset,font=(
"arial",11,"bold"),bg="black",fg="blue",width=10)

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

```
        ####Rightsideimage

        img4=Image.open(r"C:\Users\Dr.N.T.Murali Mohan\Desktop\Hotel project pics\bed.jpg")

        img4=img4.resize((520,300),Image.ANTIALIAS)

        self.photoimg4=ImageTk.PhotoImage(img4)


        label3=Label(self.root,image=self.photoimg4,bd=0,relief=RIDGE)

        label3.place(x=600,y=55,width=520,height=200)



         ######TableFrame######


Table_Frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="View details",font=("arial",12,"bold"),padx=2,pady=6)

        Table_Frame.place(x=435,y=280,width=860,height=260)





        #Show data table

        details_table=Frame(Table_Frame,bd=2,relief=RIDGE)

        details_table.place(x=0,y=50,width=860,height=180)


        scroll_x=Scrollbar(details_table,orient=HORIZONTAL)
        scroll_y=Scrollbar(details_table,orient=VERTICAL)



self.room_table=Treeview(details_table,column=("Contact","Check-
in","Check-out","Room type","available
room","meal","No.ofdays"),xscrollcommand=scroll_x.set,yscrollcomm
and=scroll_y.set)


    scroll_x.pack(side=BOTTOM,fill=X)

    scroll_y.pack(side=RIGHT,fill=X)


    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("Check-in",text="Check-In")

    self.room_table.heading("Check-out",text="Check-Out")

    self.room_table.heading("Room type",text="Room Type")

    self.room_table.heading("available room",text="Available room")


    self.room_table.heading("meal",text="Meal")


    self.room_table.heading("No.ofdays",text="No. of days”)
    self.room_table["show"]="headings"

    self.room_table.pack(fill=BOTH,expand=1)


    self.room_table.column("Contact",width=100)

    self.room_table.column("Check-in",width=100)

    self.room_table.column("Check-out",width=100)

    self.room_table.column("Room type",width=100)

    self.room_table.column("available room",width=100)


    self.room_table.column("meal",width=100)


    self.room_table.column("No.ofdays",width=100)




    self.room_table.pack(fill=BOTH,expand=1)

    self.room_table.bind("<ButtonRelease-1>",self.get_cursor)


    self.fetch_data()
```

```
    #Add data

    def add_data(self):

        if self.var_contact.get()=="" or self.var_checkin.get()=="" :

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

        else:

            try:


conn=mysql.connector.connect(host="localhost",user="root",password="1234",database="management",auth_plugin="mysql_native_password")

                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_roomno.get(),

                    self.var_meal.get(),

                    self.var_noofdays.get()

                                                                              ))

                conn.commit()
                self.fetch_data()

                conn.close()

                messagebox.showinfo("Success","customer has been
added",parent=self.root)

            except Exception as es:

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





    #Fetch data

    def fetch_data(self):


conn=mysql.connector.connect(host="localhost",user="root",password
="1234",database="management",auth_plugin="mysql_native_passwo
rd")

        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()



    #get cursor

    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_roomno.set(row[4]),

        self.var_meal.set(row[5]),

        self.var_noofdays.set(row[6])




    def mDelete(self):

        mDelete=messagebox.askyesno("Hotel Management System","Do
you want to delete this customer?",parent=self.root)

        if mDelete>0:
```

```
conn=mysql.connector.connect(host="localhost",user="root",password
="1234",database="management",auth_plugin="mysql_native_passwo
rd")
         my_cursor=conn.cursor()
         query="delete from room where Contact=%s"
         value=(self.var_contact.get(),)
         my_cursor.execute(query,value)
      else:
         if not mDelete:
            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_meal.set("")
```

```
        self.var_noofdays.set("")



    x=random.randint(100,999)

        self.var_roomno.set(str(x))

        self.var_bill.set("")



    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()=="Standard"):

            q1=float(200)

            q2=float(1000)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)
            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)

        elif (self.var_meal.get()=="Breakfast" and
self.var_roomtype.get()=="Standard AC"):

            q1=float(200)

            q2=float(1500)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)


        elif (self.var_meal.get()=="Breakfast" and
self.var_roomtype.get()=="Deluxe"):

            q1=float(200)

            q2=float(2000)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)
        elif (self.var_meal.get()=="Breakfast" and
self.var_roomtype.get()=="Deluxe AC"):
            q1=float(200)
            q2=float(2500)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            TT="Rs."+str("%.2f"%((q5)))
            self.var_bill.set(TT)


        elif (self.var_meal.get()=="Lunch" and
self.var_roomtype.get()=="Standard"):
            q1=float(600)
            q2=float(1000)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            TT="Rs."+str("%.2f"%((q5)))
            self.var_bill.set(TT)


        elif (self.var_meal.get()=="Lunch" and
self.var_roomtype.get()=="Standard AC"):
            q1=float(600)
            q2=float(1500)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)



        elif (self.var_meal.get()=="Lunch" and
self.var_roomtype.get()=="Deluxe"):

            q1=float(600)

            q2=float(2000)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)



        elif (self.var_meal.get()=="Lunch" and
self.var_roomtype.get()=="Deluxe AC"):

            q1=float(600)

            q2=float(2500)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)
```

```
      q5=float(q3*q4)
      TT="Rs."+str("%.2f"%((q5)))
      self.var_bill.set(TT)



    elif (self.var_meal.get()=="Dinner" and
self.var_roomtype.get()=="Standard"):
      q1=float(700)
      q2=float(1000)
      q3=float(self.var_noofdays.get())
      q4=float(q1+q2)
      q5=float(q3*q4)
      TT="Rs."+str("%.2f"%((q5)))
      self.var_bill.set(TT)


    elif (self.var_meal.get()=="Dinner" and
self.var_roomtype.get()=="Standard AC"):
      q1=float(700)
      q2=float(1500)
      q3=float(self.var_noofdays.get())
      q4=float(q1+q2)
      q5=float(q3*q4)
      TT="Rs."+str("%.2f"%((q5)))
```

```
            self.var_bill.set(TT)



        elif (self.var_meal.get()=="Dinner" and
self.var_roomtype.get()=="Deluxe"):
    q1=float(700)
            q2=float(2000)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            TT="Rs."+str("%.2f"%((q5)))
            self.var_bill.set(TT)

        elif (self.var_meal.get()=="Dinner" and
self.var_roomtype.get()=="Deluxe AC"):
            q1=float(700)
            q2=float(2500)
            q3=float(self.var_noofdays.get())
            q4=float(q1+q2)
            q5=float(q3*q4)
            TT="Rs."+str("%.2f"%((q5)))
            self.var_bill.set(TT)
        elif (self.var_meal.get()=="All" and
self.var_roomtype.get()=="Standard"):

            q1=float(1500)

            q2=float(1000)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)


        elif (self.var_meal.get()=="All" and
self.var_roomtype.get()=="Standard AC"):

            q1=float(1500)

            q2=float(1500)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)


        elif (self.var_meal.get()=="All" and
self.var_roomtype.get()=="Deluxe"):
            q1=float(1500)

            q2=float(2000)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

     self.var_bill.set(TT)


        elif (self.var_meal.get()=="All" and
self.var_roomtype.get()=="Deluxe AC"):

            q1=float(1500)

            q2=float(2500)

            q3=float(self.var_noofdays.get())

            q4=float(q1+q2)

            q5=float(q3*q4)

            TT="Rs."+str("%.2f"%((q5)))

            self.var_bill.set(TT)
```

```
if __name__=="__main__":

    root=Tk()

    obj=RoomBooking(root)

    root.mainloop()
```

4. roominfo.py(Room amenities):

```
from tkinter import*

from PIL import Image,ImageTk


class RoomInfo:

    def __init__(self,root):

        self.root=root

        self.root.title("Hotel management system")

        self.root.geometry("1295x550+230+220")


         #######Title########

        lbtitle=Label(self.root,text="ROOM AMENITIES",font=("times
new roman",18,"bold"),bg="black",fg="blue",bd=4,relief=RIDGE)

        lbtitle.place(x=0,y=0,width=1295,height=50)
```

```
          ########LOGO##########

          img2=Image.open(r"C:\Users\Dr.N.T.Murali
Mohan\Desktop\Hotel project pics\hotel logo.jpg")

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

     self.photoimg2=ImageTk.PhotoImage(img2)



          label1=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)

          label1.place(x=5,y=2,width=100,height=40)



          ###Room amenities

          ####main frame img#######

          img6=Image.open(r"C:\Users\Dr.N.T.Murali
Mohan\Desktop\Hotel project pics\room amenities.png")

          img6=img6.resize((1310,590),Image.ANTIALIAS)

          self.photoimg6=ImageTk.PhotoImage(img6)



          label2=Label(self.root,image=self.photoimg6,bd=4,relief=RIDGE)

          label2.place(x=0,y=50,width=1310,height=590)
```

```
if __name__=="__main__":

    root=Tk()

    obj=RoomInfo(root)

    root.mainloop()
```

5. restmenu.py(Restaurant menu):

```
from tkinter import*

from PIL import Image,ImageTk


class RestMenu:

    def __init__(self,root):

        self.root=root

        self.root.title("Hotel management system")

        self.root.geometry("1295x550+230+220")


         #######Title########

    lbtitle=Label(self.root,text="RESTAURANT MENU",font=("times
new roman",18,"bold"),bg="black",fg="blue",bd=4,relief=RIDGE)

        lbtitle.place(x=0,y=0,width=1295,height=50)
```

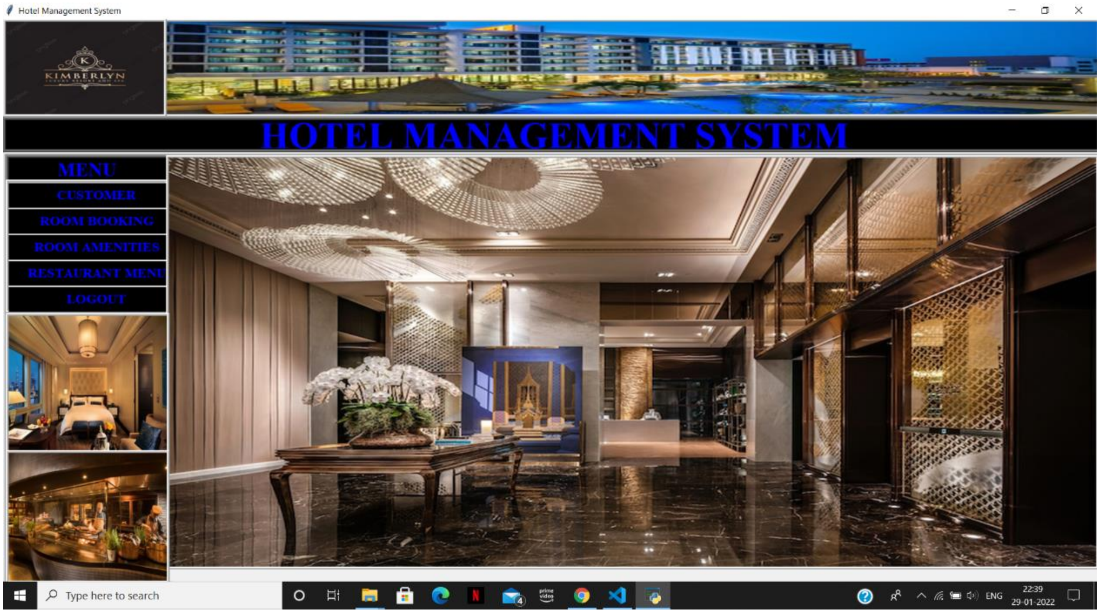
```
########LOGO##########

img2=Image.open(r"C:\Users\Dr.N.T.Murali
Mohan\Desktop\Hotel project pics\hotel logo.jpg")

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

self.photoimg2=ImageTk.PhotoImage(img2)



label1=Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)

label1.place(x=5,y=2,width=100,height=40)



###Restaurant menu

####main frame img#######

img6=Image.open(r"C:\Users\Dr.N.T.Murali
Mohan\Desktop\Hotel project pics\menu.png")

img6=img6.resize((1310,590),Image.ANTIALIAS)

self.photoimg6=ImageTk.PhotoImage(img6)



label2=Label(self.root,image=self.photoimg6,bd=4,relief=RIDGE)

label2.place(x=0,y=50,width=1310,height=590)
```

```
if __name__=="__main__":

    root=Tk()

    obj=RestMenu(root)

    root.mainloop()
```

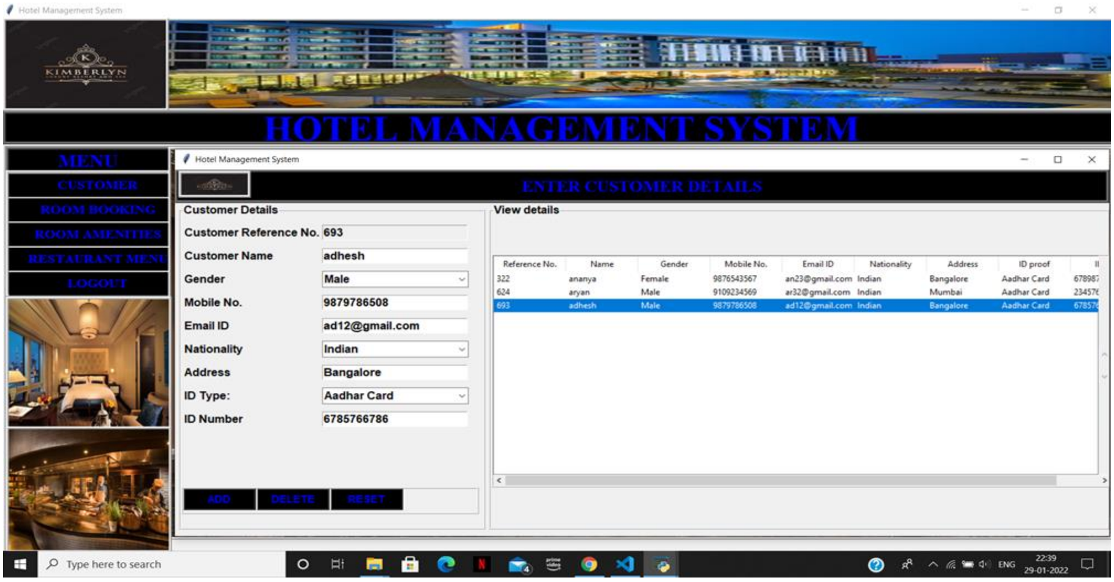
TESTING:

RESULT AND ANALYSIS:

1. **Homescreen:** Displays the homescreen



2. **Customer details entry:** Stores customer details



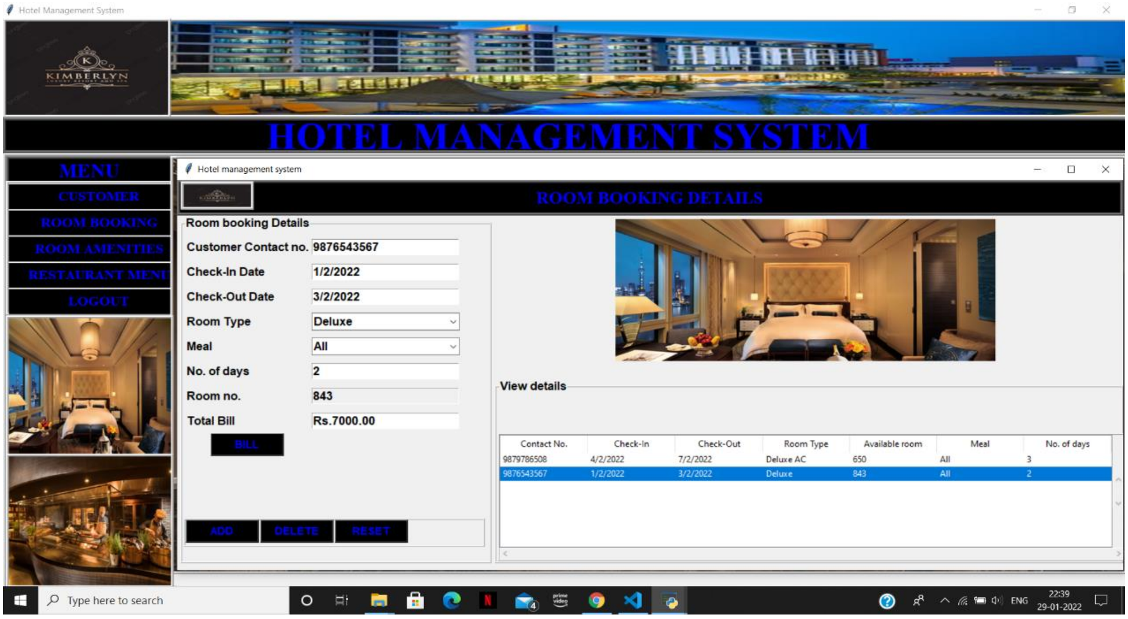
Customer Details

Customer Reference No.	693
Customer Name	adhesh
Gender	Male
Mobile No.	9879786508
Email ID	ad12@gmail.com
Nationality	Indian
Address	Bangalore
ID Type:	Aadhar Card
ID Number	6785766786

View details

Reference No.	Name	Gender	Mobile No.	Email ID	Nationality	Address	ID proof	II
322	ananya	Female	9876543567	an23@gmail.com	Indian	Bangalore	Aadhar Card	678987
624	aryan	Male	9109234569	ar32@gmail.com	Indian	Mumbai	Aadhar Card	234576
693	adhesh	Male	9879786508	ad12@gmail.com	Indian	Bangalore	Aadhar Card	678576

3. Room booking: Stores room booking details



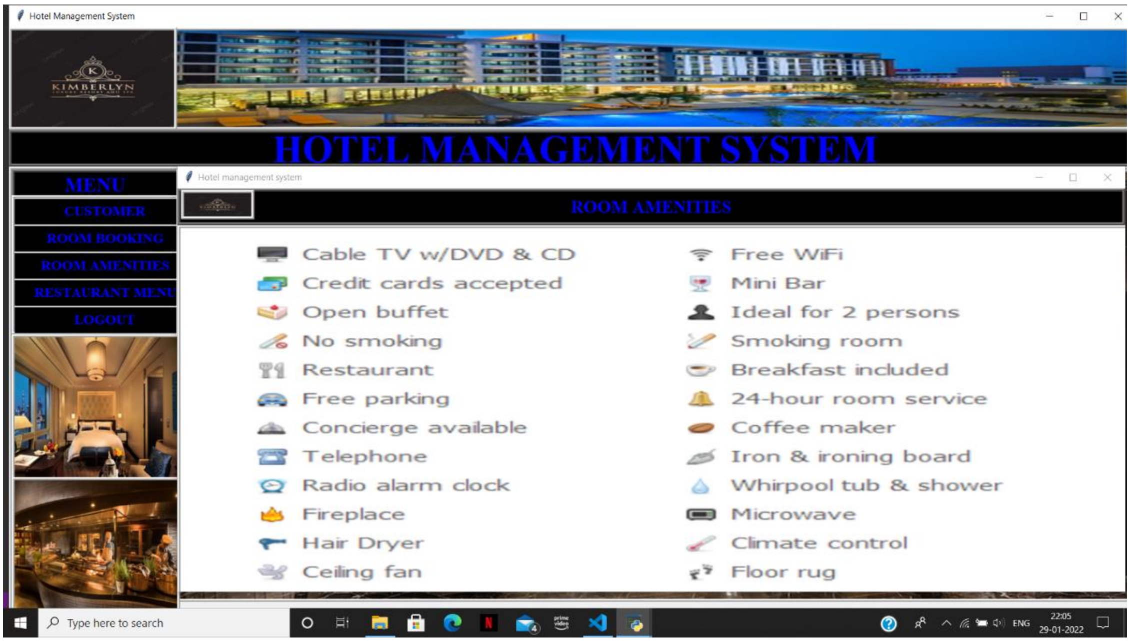
Room booking Details

Customer Contact no.	9876543567
Check-In Date	1/2/2022
Check-Out Date	3/2/2022
Room Type	Deluxe
Meal	All
No. of days	2
Room no.	843
Total Bill	Rs.7000.00

View details

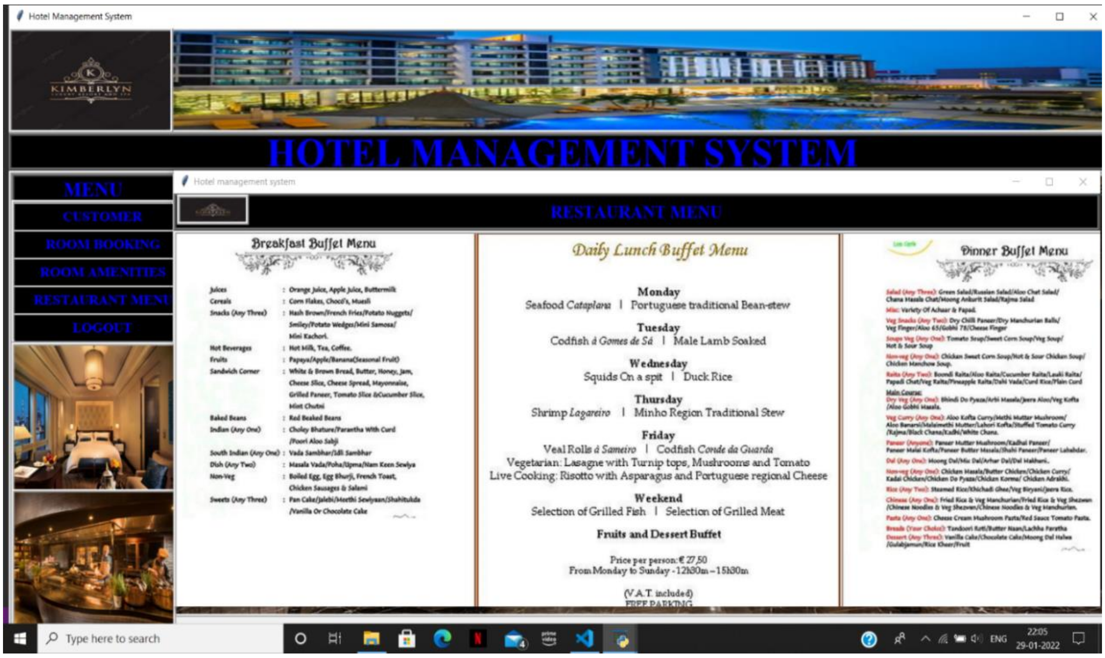
Contact No.	Check-In	Check-Out	Room Type	Available room	Meal	No. of days
9879786508	4/2/2022	7/2/2022	Deluxe AC	650	All	3
9876543567	1/2/2022	3/2/2022	Deluxe	843	All	2

4. Room amenities: Displays room amenities



- Cable TV w/DVD & CD
- Credit cards accepted
- Open buffet
- No smoking
- Restaurant
- Free parking
- Concierge available
- Telephone
- Radio alarm clock
- Fireplace
- Hair Dryer
- Ceiling fan
- Free WiFi
- Mini Bar
- Ideal for 2 persons
- Smoking room
- Breakfast included
- 24-hour room service
- Coffee maker
- Iron & ironing board
- Whirpool tub & shower
- Microwave
- Climate control
- Floor rug

5. <u>Restaurant Menu:</u> Displays restaurant menu(Buffet)



Breakfast Buffet Menu

Juices	: Orange Juice, Apple Juice, Buttermilk
Cereals	: Corn Flakes, Choco's, Muesli
Snacks (Any Three)	: Hash Brown/French Fries/Potato Nuggets/ Smiley/Potato Wedges/Mini Samosa/ Mini Kachori.
Hot Beverages	: Hot Milk, Tea, Coffee.
Fruits	: Papaya/Apple/Banana(Seasonal Fruit)
Sandwich Corner	: White & Brown Bread, Butter, Honey, Jam, Cheese Slice, Cheese Spread, Mayonnaise, Grilled Paneer, Tomato Slice &Cucumber Slice, Mint Chutni
Baked Beans	: Red Beaked Beans
Indian (Any One)	: Choley Bhature/Parantha With Curd /Poori Aloo Sabji
South Indian (Any One)	: Vada Sambhar/Idli Sambhar
Dish (Any Two)	: Masala Vada/Poha/Upma/Nam Keen Sewiya
Non-Veg	: Boiled Egg, Egg Bhurji, French Toast, Chicken Sausages & Salami
Sweets (Any Three)	: Pan Cake/Jalebi/Meethi Sewiyaan/Shahitukda /Vanilla Or Chocolate Cake

Daily Lunch Buffet Menu

Monday
Seafood *Cataplana* | Portuguese traditional Bean-stew

Tuesday
Codfish *à Gomes de Sá* | Male Lamb Soaked

Wednesday
Squids On a spit | Duck Rice

Thursday
Shrimp *Lagareiro* | Minho Region Traditional Stew

Friday
Veal Rolls *à Sameiro* | Codfish *Conde da Guarda*
Vegetarian: Lasagne with Turnip tops, Mushrooms and Tomato
Live Cooking: Risotto with Asparagus and Portuguese regional Cheese

Weekend
Selection of Grilled Fish | Selection of Grilled Meat

Fruits and Dessert Buffet

Price per person: € 27,50
From Monday to Sunday - 12h30m – 15h30m

(V.A.T. included)
FREE PARKING

Dinner Buffet Menu

Salad (Any Three): Green Salad/Russian Salad/Aloo Chat Salad/ Chana Masala Chat/Moong Ankurit Salad/Rajma Salad

Misc: Variety Of Achaar & Papad.

Veg Snacks (Any Two): Dry Chilli Paneer/Dry Manchurian Balls/ Veg Finger/Aloo 65/Gobhi 78/Cheese Finger

Soups Veg (Any One): Tomato Soup/Sweet Corn Soup/Veg Soup/ Hot & Sour Soup

Non-veg (Any One): Chicken Sweet Corn Soup/Hot & Sour Chicken Soup/ Chicken Manchow Soup.

Raita (Any Two): Boondi Raita/Aloo Raita/Cucumber Raita/Lauki Raita/ Papadi Chat/Veg Raita/Pineapple Raita/Dahi Vada/Curd Rice/Plain Curd

<u>Main Course:</u>
Dry Veg (Any One): Bhindi Do Pyaza/Arbi Masala/Jeera Aloo/Veg Kofta /Aloo Gobhi Masala.

Veg Curry (Any One): Aloo Kofta Curry/Methi Muttar Mushroom/ Aloo Banarsi/Malaimethi Mutter/Lahori Kofta/Stuffed Tomato Curry /Rajma/Black Chana/Kadhi/White Chana.

Paneer (Anyone): Paneer Muttar Mushroom/Kadhai Paneer/ Paneer Malai Kofta/Paneer Butter Masala/Shahi Paneer/Paneer Lababdar.

Dal (Any One): Moong Dal/Mix Dal/Arhar Dal/Dal Makhani.

Non-veg (Any One): Chicken Masala/Butter Chicken/Chicken Curry/ Kadai Chicken/Chicken Do Pyaza/Chicken Korma/ Chicken Adraki.

Rice (Any Two): Steamed Rice/Khichadi Ghee/Veg Biryani/Jeera Rice.

Chinese (Any One): Fried Rice & Veg Manchurian/Fried Rice & Veg Shezwan /Chinese Noodles & Veg Shezwan/Chinese Noodles & Veg Manchurian.

Pasta (Any One): Cheese Cream Mushroom Pasta/Red Sauce Tomato Pasta.

Breads (Your Choice): Tandoori Roti/Butter Naan/Lachha Paratha

Dessert (Any Three): Vanilla Cake/Chocolate Cake/Moong Dal Halwa /Gulabjamun/Rice Kheer/Fruit

CONCLUSION AND FUTURE ENHANCEMENT:

This project uses different tools in Python to create an User Interface for the Hotel to be managed. It helps keep in track of the customers, their credentials and stores them in a database using MySQL.

Future enhancement:

We believe that the project could have been more efficient if the GUI created had been an interactive platform for both the customer and the Hotel staff , where the customer is allowed to book rooms at the Hotel

We believe that the database could have had a search button so that the work of the User is made easy by just searching a key-word.

REFERENCES:

1. https://www.stackoverflow.com/
2. https://www.geeksforgeeks.org/
3. https://www.javatpoint.com/
4. Google Images.