NAME : SANJAY M

USN : 1NH18CS170

SEM/SEC : 5C

COURSE CODE : 20CSE59

COURSE : MINI PROJECT USING PYTHON

CHAPTER 1

INTRODUCTION

1.1 COURSE OBJECTIVE

The objective of the course is to provide Basic knowledge of Python and Database Management Systems. The focus of the course is to provides an introduction to programming, I/O, and visualization using the Python programming language and Database Management Systems. Python language is used for programming in mostly for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming. The Database Management Systems, provides an introduction to the management of database systems. This course focus on the understanding of the fundamentals of relational systems including data models, database architectures, and database manipulations. This course also provides an understanding of new developments and concepts. The course uses a problem-based approach for learning.

1.2 PROBLEM STATEMENT

- This Project on Hotel services is a general software developed (using Python) to simplify hotel operations by automating them.
- It covers the major aspects of hotel services and management.
- It should perform the following basic operations of a hotel like Hotel Booking, Provide Hotel Rooms Info, Room Service, Billing and Record-Keeping.
- It should contain the database connected at the backend for the documentation and Record-Keeping.
- There must be a front end that contains the GUI programming for the easy usage of the application.

1.3 EXPECTED OUTCOMES

- The application should increase efficiency of managing the rooms booking.

- It should provide the searching facilities based on various factors, such as booking , Customer, payment services.
- It tracks all the information of the hotel room and payment etc.
- Manage the information of all aspects of the hotel. Show the information and description of the booking and customers details. Editing, adding and updating of records.
- Proper resource management of the booking data.

CHAPTER 2

REQUIREMENT SPECIFICATIONS

2.1 HARDWARE REQUIREMENTS

- **RAM** : 512 MB or more
- **PROCESSOR** : Processor with single core or more cores
- **HARD DISC SPACE**
- **INPUT AND OUTPUT DEVICE**

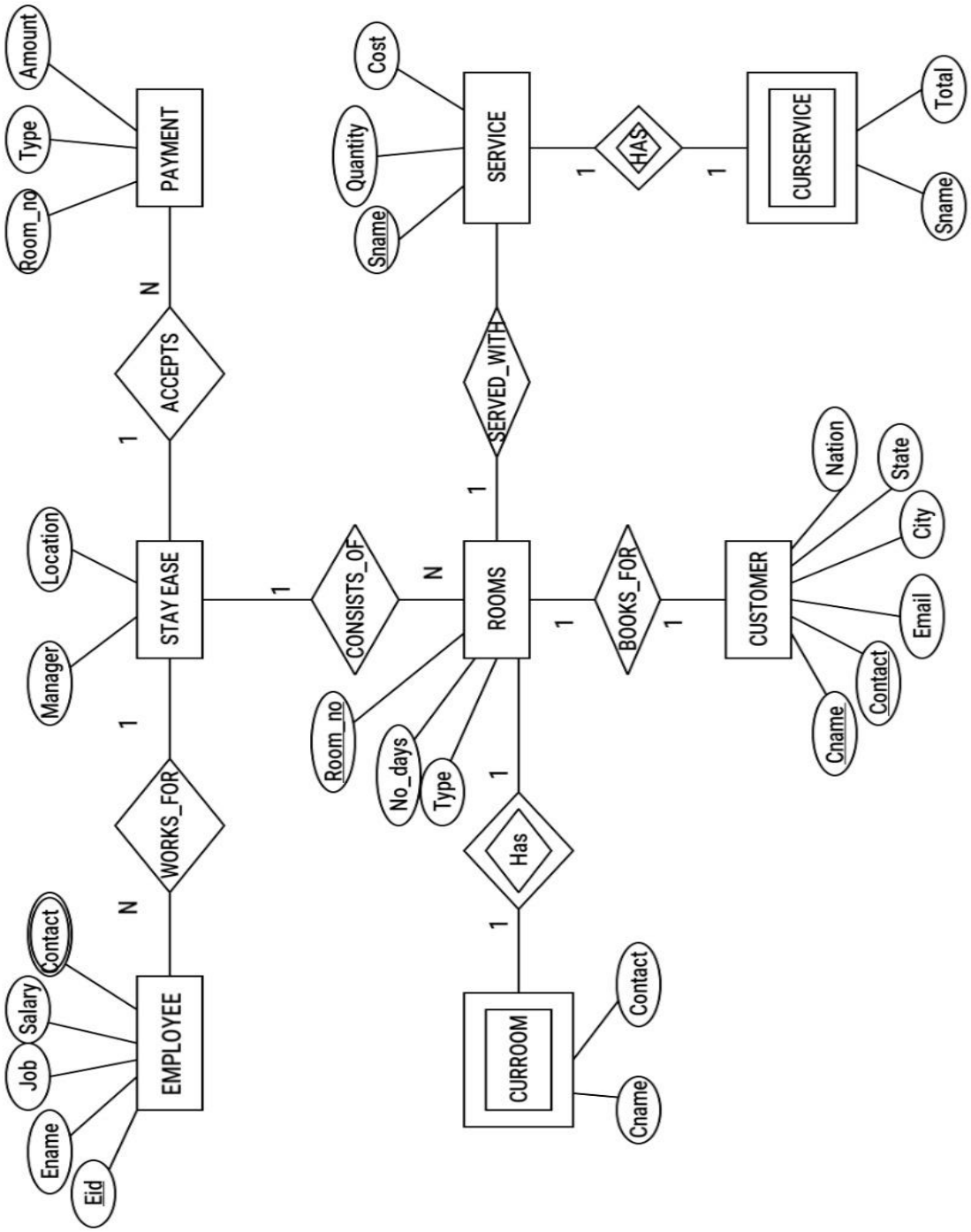
2.2 SOFTWARE REQUIREMENTS

- **COMPILERS** : IDLE python 3.8.compiler
- **OPERATING SYSTEM :** Windows 10 or any other OS
- **PROGRAMMING LANGUAGE** : Python scripting language
- **DBMS :** MYSQL
- **GUI PROGRAMMING**

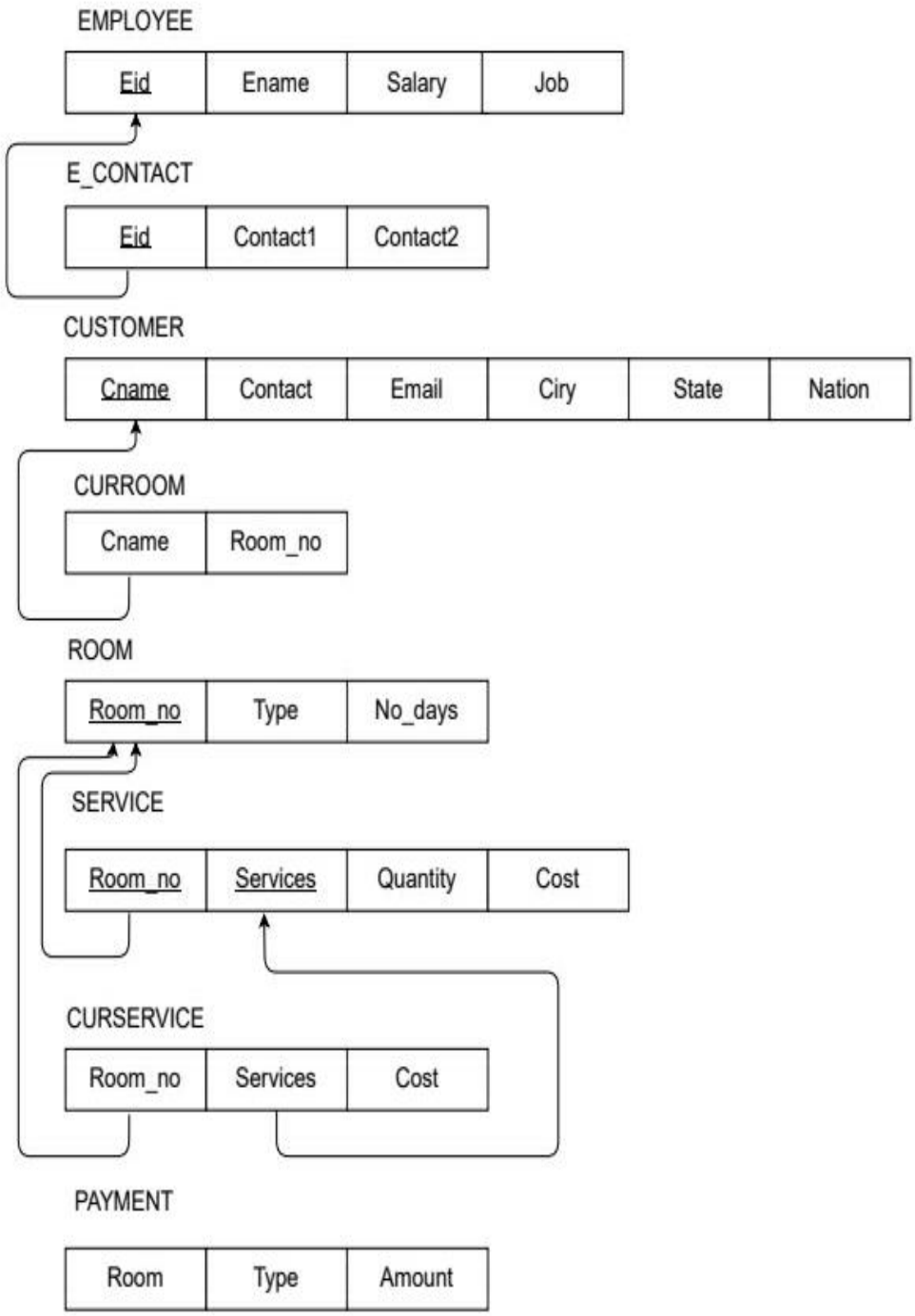
CHAPTER 3

DATABASE / ALGORITHM DESIGN

ER DIAGRAM



RELATIONAL / SCHEMA DIAGRAM



CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 FUNTIONALITY

MAIN PAGE

```
from subprocess import call
from tkinter import *
root = Tk(className=" MINI PROJECT ")
root.geometry('1920x1080')
def click_customer():
   call(["python", "customer.py"])
def click_checkout():
   call(["python", "checkout.py"])
def click_cusdetails():
   call(["python", "cusdetails.py"])
def click_vacancy():
   call(["python", "vacancy.py"])
def click_service():
   call(["python", "service.py"])
def click_payment():
   call(["python", "payment.py"])
def click_employee():
   call(['python', 'employee.py'])


menu_bar = Menu(root)
root.config(menu=menu_bar)
home_menu = Menu(menu_bar)
menu_bar.add_cascade(label="Home", menu=home_menu)
home_menu.add_command(label="Vacancy", command=click_vacancy)
home_menu.add_separator()
home_menu.add_command(label="Exit", command=root.quit)

about_menu = Menu(menu_bar)
menu_bar.add_cascade(label="About", menu=about_menu)
about_menu.add_separator()
about_menu.add_command(label="Customer details", command=click_cusdetails)
about_menu.add_separator()
about_menu.add_command(label="Employee details", command=click_employee)

tlb = Label(root, text="--------- STAY EASE - HOTEL SERVICES ---------", height=2,
font=('Orbitron', 26), bg="#ff80bf")
tlb.pack(fill=X)
```

```
welcome_label = Label(root, text="WELCOME", bg="blue", fg="white", font=('Orbitron', 20))
welcome_label.pack(fill=X)

blankspace = Label(root, text="\n")
blankspace.pack()
image1 = PhotoImage(file="logo.png")
label_for_image = Label(root, image=image1)
label_for_image.pack()

cin_button = Button(root, text="Check In", bg='#000000', fg='white', font=('Orbitron', 20,
'bold'), width=30,command=click_customer)
cin_button.pack(pady=10)
cot_button = Button(root, text="Check Out", bg='#1a1a1a', fg='white', font=('Orbitron', 20,
'bold'), width=30,command=click_checkout)
cot_button.pack(pady=10)
se_button = Button(root, text="Services", bg='#404040', fg='white', font=('Orbitron', 20,
'bold'), width=30,command=click_service)
se_button.pack(pady=10)
py_button = Button(root, text="Payment", bg='#666666', fg='white', font=('Orbitron', 20,
'bold'), width=30,command=click_payment)
py_button.pack(pady=10)
exit_button = Button(root, text="Exit", bg="#ff0000", fg="white", width=30,
command=root.quit,font=('Orbitron', 20, 'bold'))
exit_button.pack(pady=10)
root.mainloop()
```

4.2 MODULE 2 FUNTIONALITY

VACANCY

```
from tkinter import *
import mysql.connector
root = Tk(className=" MINI PROJECT ")
root.geometry('1000x650')
heading_label = Label(root, text="---------  ALL VACANCIES  ---------", font=('Orbitron', 25),
bg="black", fg="white")
heading_label.pack(fill=X)
top_frame = Frame(root)
top_frame.pack()

o_label = Label(top_frame, text='OCCUPIED', fg='red', font=('Orbitron', 25))
u_label = Label(top_frame, text='UN-OCCUPIED', fg='green', font=('Orbitron', 25))
o_label.grid(row=0, column=0)
u_label.grid(row=0, column=1)

text_o = Text(top_frame, bd=5, fg="red", width=30, bg='#b3ffe6', font=('Teko SemiBold',
```

```
20))
text_o.grid(row=1, column=0)
text_u = Text(top_frame, bd=5, fg="green", width=30, bg='#b3ffe6', font=('Teko SemiBold',
20))
text_u.grid(row=1, column=1)
rooms = [100, 101, 102, and so on ]
mydb = mysql.connector.connect(host='localhost', user='root', password='password',
database='miniproject')
cur = mydb.cursor()
cur.execute('SELECT room_no from curroom')
result = cur.fetchall()
occupied_rooms = []
for i in result:
   a = list(i)
   occupied_rooms.append(a[0])
c1 = 0
c2 = 0
for j in rooms:
   if j in occupied_rooms:
      text_o.insert(INSERT, str(j))
      text_o.insert(INSERT, "    ")
      c1 = c1 + 1
      if c1 == 5:
         text_o.insert(INSERT, "\n")
         c1 = 0
   else:
      text_u.insert(INSERT, str(j))
      text_u.insert(INSERT, "    ")
      c2 = c2 + 1
      if c2 == 5:
         text_u.insert(INSERT, "\n")
         c2 = 0
root.mainloop()
```

4.3 MODULE 3 FUNTIONALITY

SERVICE

```
from tkinter import *
from tkinter import messagebox
from subprocess import call
import mysql.connector
root = Tk()
root.title("MINI PROJECT")
```

```
root.geometry("1000x750")
#funtion call

#variable declaration
def click_submit():
    #get variables
   if room_no == 0 and v1 == 0 and v2 == 0 and v3 == 0 and v4 == 0 and v5 == 0 and v6 == 0
and v7 == 0 and n1 == 0 and n2 == 0 and n3 == 0 and n4 == 0 and n5 == 0 and n6 == 0 and n7
== 0:
       messagebox.showwarning("Warning", "Incomplete Data Entry")
   else:

       #mysql connect

       avail = 0
       for i in res:
          a = list(i)
          avail = a[0]
       if avail == 1:
          s1 = "LAUNDRY"
          s2 = "RESTAURANT"
          t1 = n1+n2+n3
          t2 = n4+n5+n6+n7
          c1 = (n1*300+n2*300+n3*150)
          c2 = (n4*200+n5*200+n6*200+n7*100)
          cur.execute('insert into service' '(room_no,services,quantity,cost)'
                'values(%s, %s, %s, %s)', (room_no, s1, t1, c1))
          cur.execute('insert into service' '(room_no,services,quantity,cost)'
                'values(%s,%s,%s,%s)',
                (room_no, s2, t2, c2))
          cur.execute('insert into curservice' '(room_no,cost)''values(%s,%s)', (room_no,
c1+c2))
          messagebox.showinfo("DETAILS STORED", "SERVICE DATA STORED")
          root.destroy()
       else:
          messagebox.showinfo("Room", "ROOM NOT OCCUPIED")

heading_label = Label(root, text="---------  SERVICE DATA  ----------", font=('Orbitron', 25),
bg="black",fg="white")
heading_label.pack(fill=X)
top_frame = Frame(root)
top_frame.pack()
rn_label = Label(top_frame, text="Room Number : ", font=('Orbitron', 20))
rn_entry = Entry(top_frame, textvar=Room_no, bd=5, bg="#ccefff", fg='blue', width=5,
font=('Arial', 15))
rn_label.grid(row=0, column=1, padx=15, pady=10, sticky=E)
rn_entry.grid(row=0, column=2, ipady=5, ipadx=60, sticky=W)
```

```
ld_label = Label(top_frame, text="  LAUNDRY  ", font=('Orbitron', 20))
ld_label.grid(row=1, column=2, padx=15, pady=10, sticky=E)
Cb1 = Checkbutton(top_frame, text="MEN'S WEAR", variable=var1, onvalue=1, offvalue=0)
Cb1.grid(row=2, column=1, sticky=W)
Cb2 = Checkbutton(top_frame, text="WOMEN'S WEAR", variable=var2, onvalue=1,
offvalue=0)
Cb2.grid(row=3, column=1, sticky=W)
Cb3 = Checkbutton(top_frame, text="KID'S WEAR", variable=var3, onvalue=1, offvalue=0)
Cb3.grid(row=4, column=1, sticky=W)

qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qb = Spinbox(top_frame, textvar=num1, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=2, column=2, padx=15, pady=10, sticky=E)
qb.grid(row=2, column=3, ipady=5, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qb = Spinbox(top_frame, textvar=num2, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=3, column=2, padx=15, pady=10, sticky=E)
qb.grid(row=3, column=3, ipady=5, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qb = Spinbox(top_frame, textvar=num3, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=4, column=2, padx=15, pady=10, sticky=E)
qb.grid(row=4, column=3, ipady=5, sticky=W)

rl = Label(top_frame, text="  RESTUARENT ", font=('Orbitron', 20))
rl.grid(row=5, column=2, padx=15, pady=10, sticky=E)
Cb1 = Checkbutton(top_frame, text="BREAK FAST", variable=var4, onvalue=1, offvalue=0)
Cb1.grid(row=6, column=1, sticky=W)
Cb2 = Checkbutton(top_frame, text="LUNCH", variable=var5, onvalue=1, offvalue=0)
Cb2.grid(row=7, column=1, sticky=W)
Cb3 = Checkbutton(top_frame, text="DINNER", variable=var6, onvalue=1, offvalue=0)
Cb3.grid(row=8, column=1, sticky=W)
Cb4 = Checkbutton(top_frame, text="SNACKS AND DRINKS", variable=var7, onvalue=1,
offvalue=0)
Cb4.grid(row=9, column=1, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qn = Spinbox(top_frame, textvar=num4, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=6, column=2, padx=15, pady=10, sticky=E)
qn.grid(row=6, column=3, ipady=5, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qn = Spinbox(top_frame, textvar=num5, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
```

```
qu.grid(row=7, column=2, padx=15, pady=10, sticky=E)
qn.grid(row=7, column=3, ipady=5, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qn = Spinbox(top_frame, textvar=num6, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=8, column=2, padx=15, pady=10, sticky=E)
qn.grid(row=8, column=3, ipady=5, sticky=W)
qu = Label(top_frame, text="QUANTITY : ", font=('Orbitron', 10))
qn = Spinbox(top_frame, textvar=num7, bg="#ccefff", fg='blue', from_=0, to=30, width=5,
bd=5,font=('Orbitron', 15))
qu.grid(row=9, column=2, padx=15, pady=10, sticky=E)
qn.grid(row=9, column=3, ipady=5, sticky=W)

submit_button = Button(root, text="SUBMIT", width=15, bg="#269900", fg='Black',
font=('ARIAL BLACK', 20), relief=RAISED,command=click_submit)
submit_button.place(relx=0.5, rely=0.95, anchor=S)
mainloop()
```

4.4 MODULE 4 FUNTIONALITY

PAYMENT

```
from tkinter import *
from subprocess import call
import mysql.connector
from tkinter import messagebox
root = Tk(className=" MINI PROJECT ")
root.geometry('1000x750')
#funtion call
Tp = StringVar()
Room_no = IntVar()
Amount = IntVar()
def click_submit():
    room_no = Room_no.get()
    amount = Amount.get()
    tp = Tp.get()
    if room_no == 0 or amount == 0 or tp == '':
        messagebox.showwarning("Warning", "Incomplete Data Entry")
    else:
        #mysql connect
        avail = 0
        for i in res:
            a = list(i)
            avail = a[0]
        if avail == 1:
```

```
        cur.execute('SELECT type,no_days from room where room_no=%s', (room_no,))
        rdetail = cur.fetchall()
        cur.execute('select sum(cost) from curservice where room_no=%s', (room_no,))
        total = cur.fetchall()
        rbill = []
        t = []
        for i in total:
            t = list(i)
        for j in rdetail:
            rbill = list(j)
        if rbill[1] == 1:
            am = rbill[1] * 2000
        else:
            am = rbill[1] * 1500
        tl = am+t[0]
        if tl == amount:
            cur.execute('insert into payment'
                        '(room_no,type,amount)'
                        'values(%s,%s,%s)', (room_no, tp, amount))
            cur.execute('delete from curroom where room_no=%s', (room_no,))
            cur.execute('delete from curservice where room_no=%s', (room_no,))
            mydb.commit()
            messagebox.showinfo("PAYMENT", "PAYMENT DONE")
            root.destroy()
        else:
            messagebox.showinfo("PAYMENT", "AMOUNT MISMATCH")
    else:
        messagebox.showinfo("PAYMENT", "Room NOT Occupied")

heading_label = Label(root, text="---------  PAYMENT  ---------", font=('Orbitron', 25),
bg="black", fg="white")
heading_label.pack(fill=X)
black_space = Label(root, text="\n\n")
black_space.pack()
top_frame = Frame(root)
top_frame.pack()

rn_label = Label(top_frame, text="ROOM NO : ", font=('Orbitron', 20))
rn_entry = Entry(top_frame, textvar=Room_no, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
rn_label.grid(row=1, column=0, padx=15, pady=10, sticky=E)
rn_entry.grid(row=1, column=1, ipady=5, ipadx=60, sticky=W)
am_label = Label(top_frame, text=" AMOUNT : ", font=('Orbitron', 20))
am_entry = Entry(top_frame, bd=5, textvar=Amount, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
```

```
am_label.grid(row=2, column=0, padx=15, pady=10, sticky=E)
am_entry.grid(row=2, column=1, ipady=5, ipadx=60)
t_label = Label(top_frame, text=" TYPE : ", font=('Orbitron', 20))
t_entry = Entry(top_frame, bd=5, textvar=Tp, bg="#ccefff", fg='blue', width=20, font=('Arial',
15))
t_label.grid(row=3, column=0, padx=15, pady=10, sticky=E)
t_entry.grid(row=3, column=1, ipady=5, ipadx=60)
c_label = Label(top_frame, text=" * CASH, CARDS OR UPI ", font=('Orbitron', 10))
c_label.grid(row=4, column=1, padx=15, pady=10, sticky=E)

submit_button = Button(top_frame, text="SUBMIT", width=15, bg="#269900", fg='Black',
font=('ARIAL BLACK', 20), relief=RAISED,command=click_submit)
submit_button.2(row=5, column=1, padx=15, pady=10, sticky=E)
root.mainloop()
```

4.5 MODULE 5 FUNTIONALITY

CHECKOUT

```
from subprocess import call
from tkinter import *
import mysql.connector
from tkinter import messagebox
root = Tk(className=" MINI PROJECT ")
root.geometry('1000x750')

F_name = StringVar()
Room_no = IntVar()

def click_proceed():
   f_name = F_name.get()
   room_no = Room_no.get()
   if f_name == '' or room_no == 0:
      messagebox.showwarning("Warning", "Incomplete Data Entry")
   else:
      text.delete('1.0', END)

           #mysql connect

            avail = 0
      for i in res:
         a = list(i)
         avail = a[0]
      if avail == 1:
         cur.execute('SELECT cname,cont,email from customer where cname=%s', (f_name,))
         cusdetail = cur.fetchall()
         cur.execute('SELECT type,no_days from room where room_no=%s', (room_no,))
```

```
        rdetail = cur.fetchall()
        cur.execute('select sum(cost) from service where room_no=%s', (room_no,))
        total = cur.fetchall()
        mydb.commit()
        fname_entry.delete(0, 'end')
        rn_entry.delete(0, 'end')
        text.insert(INSERT, "\t\t  Checked out --- Continue to the payment\n\n")
        formatting = "----------------------------------------------------------------------------------------" \
             "----------\n"
        text.insert(INSERT, formatting)
        text.insert(INSERT, formatting)
        bill = []
        rbill = []
        for i in cusdetail:
            bill = list(i)
            sbill = "Name :\t " + bill[0] + "\n" + "Contct:\t" + str(bill[1]) + "\n"+"Email :\t" +
bill[2] + "\n"
            text.insert(INSERT, sbill)
            for j in rdetail:
                rbill = list(j)
            s1 = "Room Number :\t " + str(room_no) + "\n"
            s2 = "Room Type :\t " + str(rbill[0]) + "\n"
            s3 = "Number of Days :\t " + str(rbill[1]) + "\n"
            text.insert(INSERT, s1)
            text.insert(INSERT, s2)
            text.insert(INSERT, s3)
            if rbill[1] == 1:
                amount = rbill[1] * 2000
            else:
                amount = rbill[1] * 1500
            t = []
            for j in total:
                t = list(j)
            s4 = "Room Amount To Be Paid : \t" + str(amount)
            s5 = "\nReatuarant and laundry bill:\t" + str(t[0])
            s6 = "\nTotal amount to be paid:\t" + str(amount+t[0])
            text.insert(INSERT, s4)
            text.insert(INSERT, s5)
            text.insert(INSERT, s6)
    else:
        fname_entry.delete(0, 'end')
        rn_entry.delete(0, 'end')
        text.insert(INSERT, "INVALID DATA !!!!!!!\t\t Please Enter Correct Details   !!!!!")

heading_label = Label(root, text="---------  CUSTOMER CHECK OUT  ---------", font=('Orbitron',
```

```
25), bg="black",  fg="white")
heading_label.pack(fill=X)
black_space = Label(root, text="\n\n")
black_space.pack()
top_frame = Frame(root)
top_frame.pack()

fname_label = Label(top_frame, text="Customer Name : ", font=('Orbitron', 20))
fname_entry = Entry(top_frame, textvar=F_name, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
fname_label.grid(row=0, column=0, padx=15, pady=10, sticky=E)
fname_entry.grid(row=0, column=1, pady=10, ipady=5, ipadx=60)

rn_label = Label(top_frame, text="Room Number : ", font=('Orbitron', 20))
rn_entry = Entry(top_frame, textvar=Room_no, bd=5, bg="#ccefff", fg='blue', width=5,
font=('Arial', 15))
rn_label.grid(row=2, column=0, padx=15, pady=10, sticky=E)
rn_entry.grid(row=2, column=1, ipady=5, ipadx=60, sticky=W)

proceed_button = Button(root, text="PROCEED", width=10, bg="#0000b3", fg='White',
font=('ARIAL BLACK', 20),relief=RAISED, command=click_proceed)
proceed_button.place(relx=0.5, rely=0.45, anchor=S)

text = Text(root, bd=5, bg="white", fg='blue', width=200, font=('Arial', 15))
text.place(rely=0.48)
root.mainloop()
```

4.6 MODULE 6 FUNTIONALITY

CUSTOMER

```
from tkinter import *
from subprocess import call
import mysql.connector
from tkinter import messagebox
root = Tk(className=" MINI PROJECT ")
root.geometry('1000x750')
#function call

#variables declaration
def click_submit():
   #get variables
   if f_name == '' or phone == 0 or email == '' or city == '' or state == '' or nation == '' or
room_no == '' or room_type == '' or no_days == '':
      messagebox.showwarning("Warning", "Incomplete Data Entry")
```

```
   else:
     #mysql connect
 avail = 0
      for i in res:
         a = list(i)
         avail = a[0]
      if avail == 0:
         cur.execute('INSERT INTO customer'
                  '(cname, cont, email,City, state, nation) '
                  ' VALUES(%s,%s,%s,%s,%s,%s)', (f_name, phone, email, city, state, nation))
         cur.execute('insert into room'
               '(room_no,type,no_days)'
               'values(%s,%s,%s)', (room_no, room_type, no_days))
         cur.execute('insert into curroom'
                  '(room_no,cname)'
                  'values(%s,%s)', (room_no, f_name))
         mydb.commit()
         messagebox.showinfo("DETALIS STORED", "ROOM ALLOCATION DONE")
         root.destroy()
      else:
         messagebox.showinfo("Room", "Room Already Occupied")
         rn_entry.delete(0, 'end')

heading_label = Label(root, text="---------  CUSTOMER CHECK IN FORM  ---------",
font=('Orbitron', 25), bg="black",fg="white")
heading_label.pack(fill=X)
black_space = Label(root, text="\n\n")
black_space.pack()
top_frame = Frame(root)
top_frame.pack()

fname_label = Label(top_frame, text="First Name : ", font=('Orbitron', 20))
fname_entry = Entry(top_frame, textvar=F_name, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
fname_label.grid(row=0, column=0, padx=15, pady=10, sticky=E)
fname_entry.grid(row=0, column=1, pady=10, ipady=5, ipadx=60)
phone_label = Label(top_frame, text="Mobile Number : ", font=('Orbitron', 20))
phone_entry = Entry(top_frame, textvar=Phone, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
phone_label.grid(row=2, column=0, padx=15, pady=10, sticky=E)
phone_entry.grid(row=2, column=1, ipady=5, ipadx=60)
email_label = Label(top_frame, text="Email Address : ", font=('Orbitron', 20))
email_entry = Entry(top_frame, textvar=Email, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
email_label.grid(row=3, column=0, padx=15, pady=10, sticky=E)
```

```
email_entry.grid(row=3, column=1, ipady=5, ipadx=60)
ad_label = Label(top_frame, text=" City : ", font=('Orbitron', 20))
ad_entry = Entry(top_frame, bd=5, textvar=City, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
ad_label.grid(row=4, column=0, padx=15, pady=10, sticky=E)
ad_entry.grid(row=4, column=1, ipady=5, ipadx=60)
rn_label = Label(top_frame, text="State : ", font=('Orbitron', 20))
rn_entry = Entry(top_frame, textvar=State, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
rn_label.grid(row=5, column=0, padx=15, pady=10, sticky=E)
rn_entry.grid(row=5, column=1, ipady=5, ipadx=60, sticky=W)
n_label = Label(top_frame, text="Nation : ", font=('Orbitron', 20))
n_box = Entry(top_frame, textvar=Nation, bd=5, bg="#ccefff", fg='blue', width=20,
        font=('Arial', 15))
n_label.grid(row=6, column=0, padx=15, pady=10, sticky=E)
n_box.grid(row=6, column=1, ipady=5, ipadx=60, sticky=W)
rn_label = Label(top_frame, text="Room Number : ", font=('Orbitron', 20))
rn_entry = Entry(top_frame, textvar=Room_no, bd=5, bg="#ccefff", fg='blue', width=5,
font=('Arial', 15))
rn_label.grid(row=7, column=0, padx=15, pady=10, sticky=E)
rn_entry.grid(row=7, column=1, ipady=5, ipadx=60, sticky=W)
day_label = Label(top_frame, text="Number of Days : ", font=('Orbitron', 20))
day_box = Spinbox(top_frame, textvar=No_days, bg="#ccefff", fg='blue', from_=1, to=30,
width=5, bd=5,font=('Orbitron', 15))
day_label.grid(row=8, column=0, padx=15, pady=10, sticky=E)
day_box.grid(row=8, column=1, ipady=5, sticky=W)
room_label = Label(top_frame, text="Room Type : ", font=('Orbitron', 20))
ac_rb = Radiobutton(top_frame, variable=Room_type, text="AC Room", fg='blue',
font=('Arial', 12, 'bold'), value=1)
nac_rb = Radiobutton(top_frame, variable=Room_type, text="Non-AC Room", fg='blue',
font=('Arial', 12, 'bold'), value=2)
room_label.grid(row=9, column=0, padx=15, pady=10, sticky=E)
ac_rb.grid(row=9, column=1, sticky=W)
nac_rb.grid(row=9, column=1, sticky=E)
v_button = Button(top_frame, text="Vacancy", font=('ARIAL BLACK', 15), bg='#80002a',
        fg='White', width=10, command=click_vacancy)
v_button.grid(row=7, column=1, ipadx=7, sticky=E)

submit_button = Button(root, text="SUBMIT", width=15, bg="#269900", fg='Black',
font=('ARIAL BLACK', 20), relief=RAISED,
            command=click_submit)
submit_button.place(relx=0.5, rely=0.95, anchor=S)
root.mainloop()
```

4.7 MODULE 7 FUNCTIONALITY

CUSDETAILS

```
from tkinter import *
from subprocess import call
import mysql.connector
from tkinter import messagebox

root = Tk(className=" MINI PROJECT ")
root.geometry('1000x750')
F_name = StringVar()
contact = StringVar()
def click_search():
   text.delete('1.0', END)
   f_name = F_name.get()
   cont = contact.get()
   if f_name == '' or cont == 0:
      messagebox.showwarning("Warning", "Incomplete Data Entry")
   else:

       #mysql connect

       tup = []
      for i in res:
         tup = list(i)
         final_detail = "Customer Name :\t " + tup[0] + "\n\n" + "Contact :\t " + str(tup[1]) +
"\n\n" \+ "Email  :\t" + tup[2] + "\n\n" + "City :\t " + tup[3] + "\n\n" \+ "State :\t " + tup[4] +
"\n\n" + "Nation :\t " + tup[5] + "\n"

          text.insert(INSERT, final_detail)

      cust_entry.delete(0, 'end')
      ph_entry.delete(0, 'end')
heading_label = Label(root, text="---------  CUSTOMER DETAILS  ---------", font=('Orbitron',
25), bg="black",fg="white")
heading_label.pack(fill=X)

top_frame = Frame(root)
top_frame.pack()
blankspace = Label(top_frame, text="\n\n\n")
blankspace.grid(row=0)
cust_label = Label(top_frame, text="Customer Name : ", font=('Orbitron', 20))
cust_entry = Entry(top_frame, textvar=F_name, bd=5, bg="#ccefff", fg='blue', width=20,
font=('Arial', 15))
cust_label.grid(row=1, column=0, padx=15, pady=10, sticky=W)
cust_entry.grid(row=1, column=1, pady=10, ipady=5, ipadx=60)
ph_label = Label(top_frame, text="Phone number : ", font=('Orbitron', 20))
ph_entry = Entry(top_frame, textvar=contact, bd=5, bg="#ccefff", fg='blue', width=5,
```

```
font=('Arial', 15))
ph_label.grid(row=2, column=0, padx=15, pady=10, sticky=E)
ph_entry.grid(row=2, column=1, ipady=5, ipadx=60, sticky=W)

submit_button = Button(root, text="SEARCH", width=12, bg="#269900", fg='Black',
font=('ARIAL BLACK', 20), relief=RAISED,command=click_search)
submit_button.place(relx=0.55, rely=0.4, anchor=S)
text = Text(root, bd=5, bg="white", fg='blue', width=200, font=('Arial', 15))
text.place(rely=0.45)
root.mainloop()
```

4.8 MODULE 8 FUNCTIONALITY

EMPLOYEE

```
import mysql.connector
from tkinter import *
root = Tk(className=" MINI PROJECT ")
root.geometry('1000x750')
heading_label = Label(root, text="---------  EMPLOYEE DETAILS  ---------", font=('Orbitron', 25),
bg="black", fg="white")
heading_label.pack(fill=X)
text = Text(root, bd=5, bg="white", fg='blue', width=200, font=('Arial', 15))
text.place(rely=0.45)
ed = StringVar()
top_frame = Frame(root)
top_frame.pack()
blankspace = Label(top_frame, text="\n\n\n")
blankspace.grid(row=0)
cl = Label(top_frame, text="EID : ", font=('Orbitron', 20))
ce = Entry(top_frame, textvar=ed, bd=5, bg="#ccefff", fg='blue', width=20, font=('Arial', 15))
cl.grid(row=1, column=0, padx=15, pady=10, sticky=W)
ce.grid(row=1, column=1, pady=10, ipady=5, ipadx=60)
cl = Label(top_frame, text="*E001,E002..... : ", font=('Orbitron', 15))
cl.grid(row=2, column=1, padx=15, pady=10, sticky=W)

def click_search():
   text.delete('1.0', END)
   e = ed.get()

#mysql connect
       avail = 0
   for i in res:
      a = list(i)
      avail = a[0]
```

```
   if avail == 1:
      #select from database

     tup = []
       for i in res:
          tup = list(i)
          final_detail = "\n\n EID :\t " + str(tup[0]) + "\n\n" + "EMPLOYEE NAME :\t " + tup[1] +
"\n\n" \+ "SALARY: \t" + str(tup[2]) + "\n\n"+"JOB :\t "+tup[3]+"\n\n" + "CONTACT1 :\t " +
str(tup[4]) + "\n\n" \+ "CONTACT2:\t " + str(tup[5]) + "\n"
          text.insert(INSERT, final_detail)
   else:
       text.insert(INSERT, "INVALID DATA !!!!!!!\t\t Please Enter Correct Details   !!!!!")

submit_button = Button(root, text="SEARCH", width=12, bg="#269900", fg='Black',
font=('ARIAL BLACK', 20), relief=RAISED,command=click_search)
submit_button.place(relx=0.55, rely=0.4, anchor=S)
root.mainloop()
```

CHAPTER 5

RESULTS

5.1 MAIN PAGE

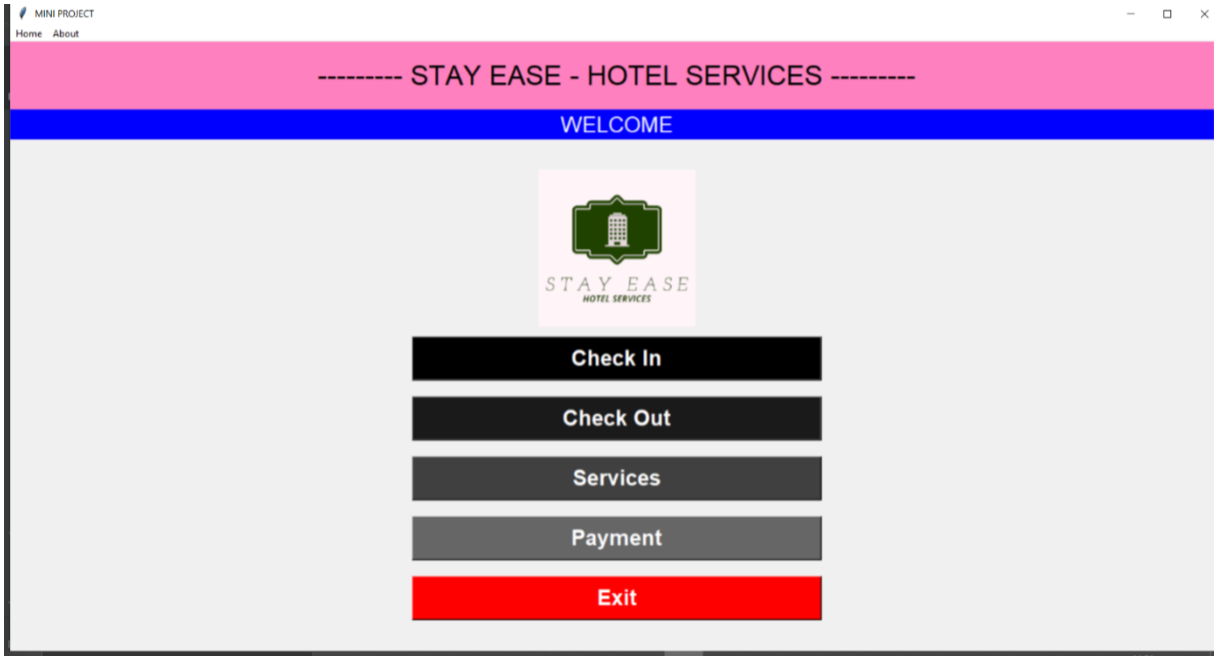


Fig 5.a main page

5.2 VACANCY

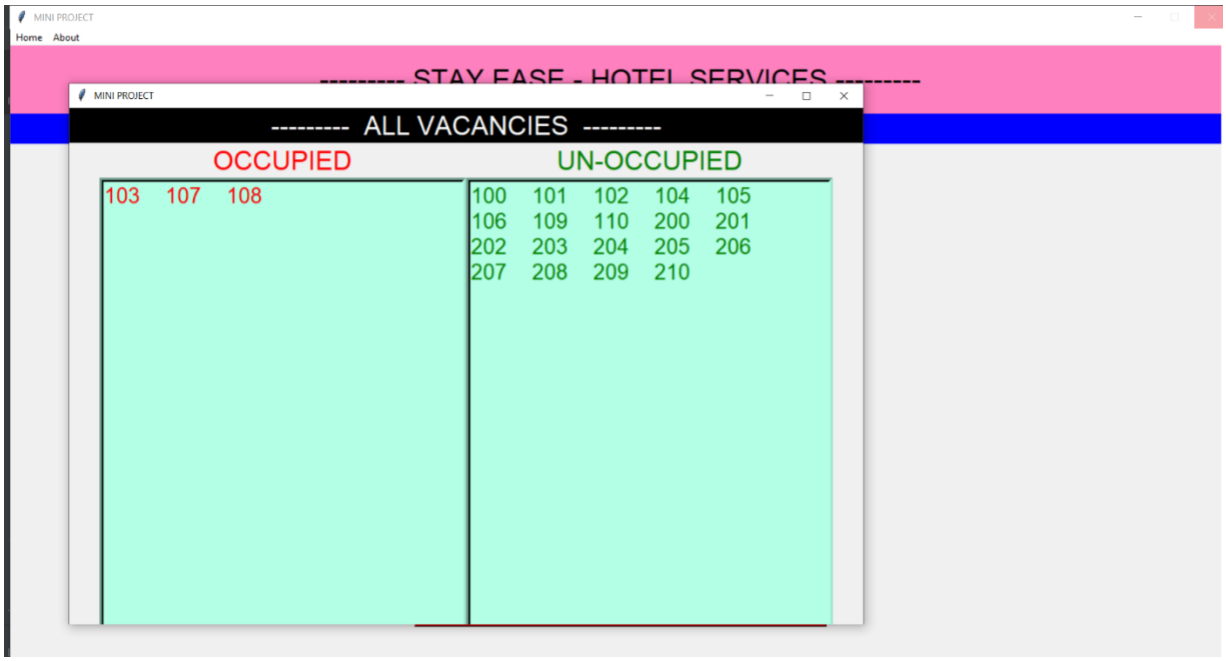


Fig 5.b vacancy page

5.3 CHECK IN

The user must enter the corresponding data required for the booking of the room.

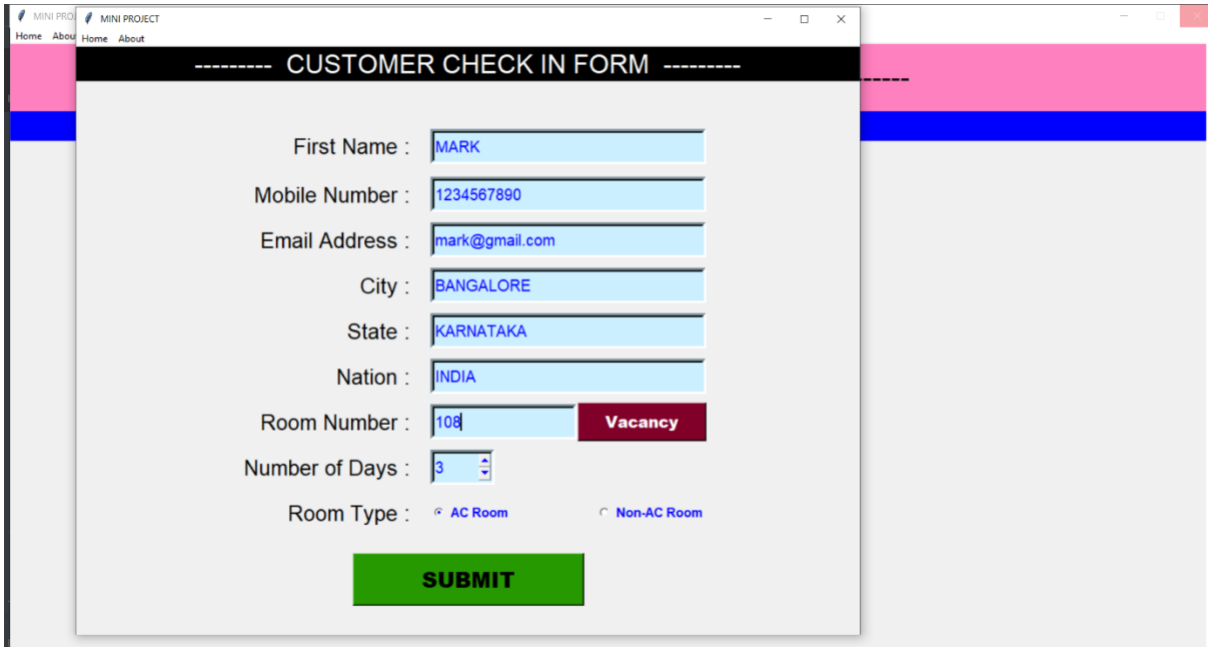


Fig 5.c check in page

5.4 SEVICE

The various services that are provided by the STAY EASE hotel to the customer are stored for the billing as well as Record-Keeping purposes.

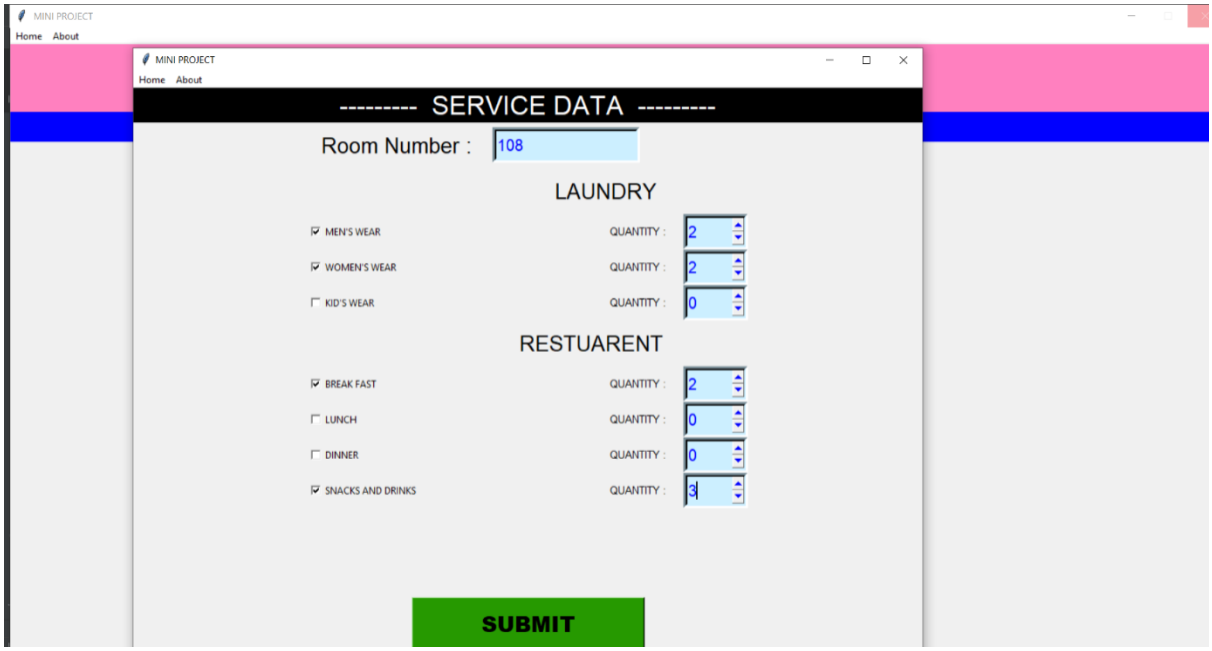


Fig 5.d service page

5.5 CHECK OUT

The application generates the check out summary for the room which is being vacated. The corresponding service bill, room bill and their total amount is calculated and displayed along the details of the customer.

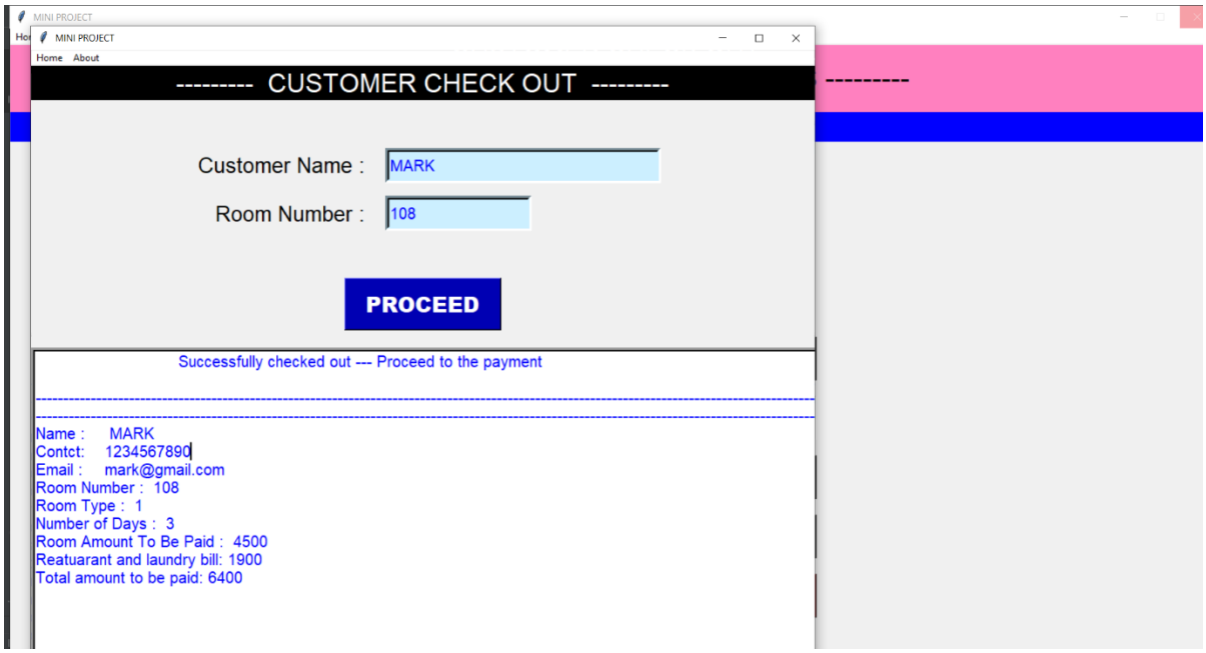


Fig 5.e check out page

5.6 PAYMENT

Payment for the room is done here and the corresponding details are stored in database.

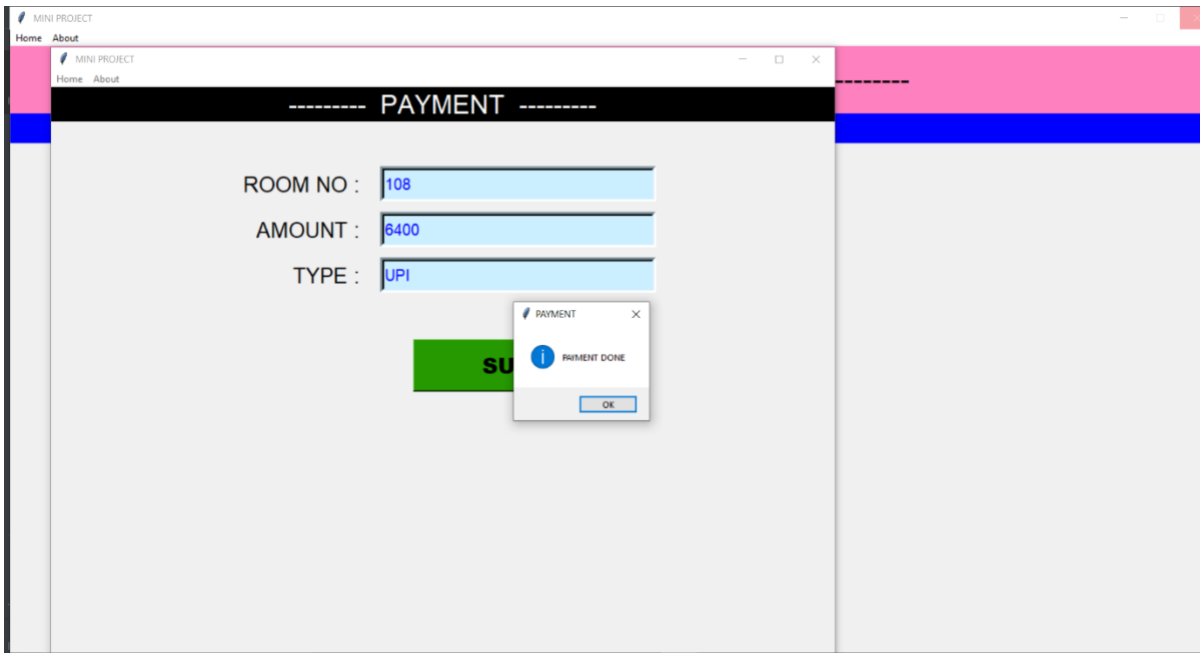


Fig 5.f payment page

5.7 CUSTOMER DETAILS

The application allows to search the details regarding the customer's who have already visited the STAY EASE hotel .

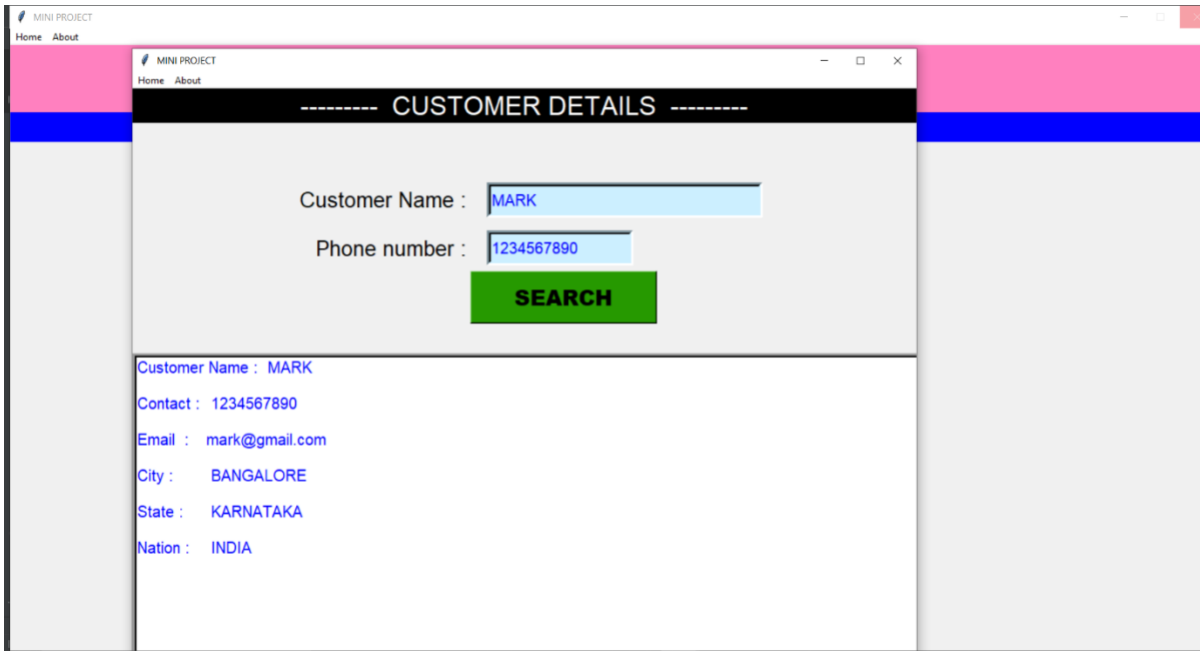


Fig 5.g customer details page

5.8 EMPLOYEE DETAILS

The application allows to search the information about the employees who are working in the hotel .

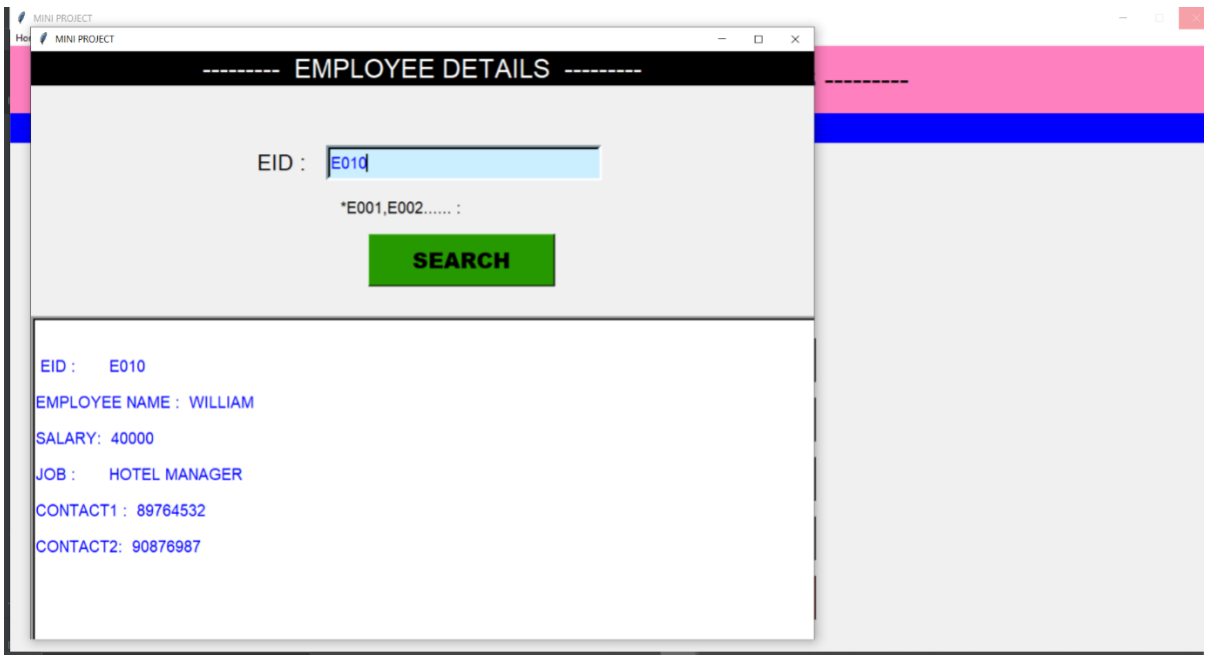


Fig 5.h employee details page

5.9 DATABASE TABLES 1

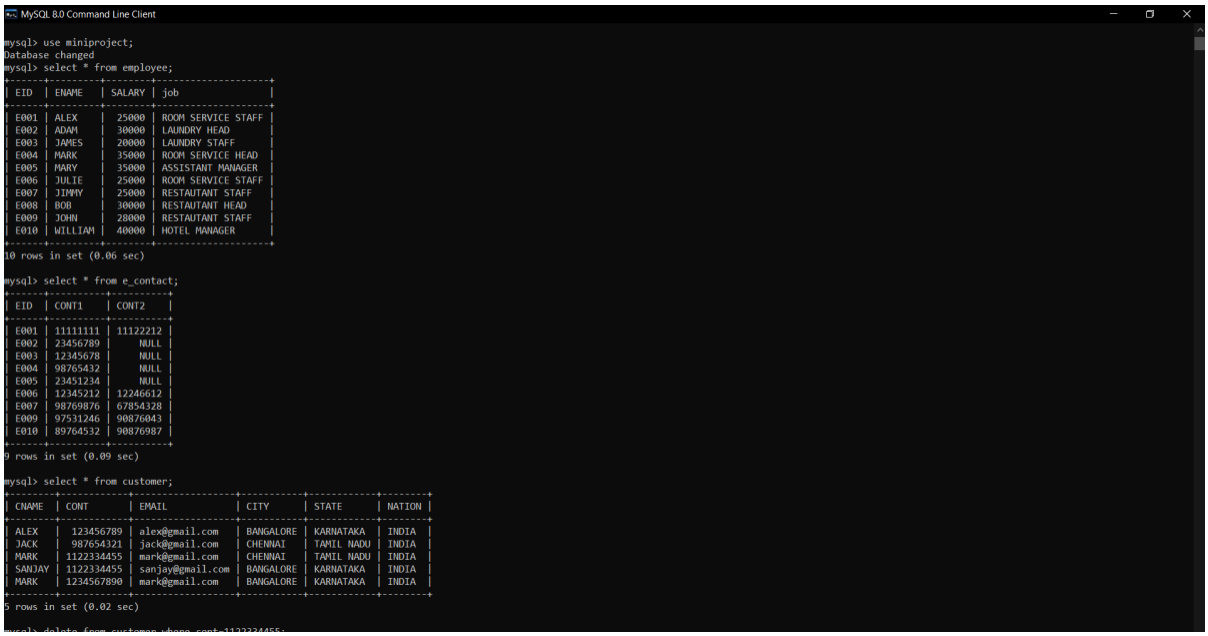


Fig 5.i Tables 1

5.10 DATABASE TABLES 2

Fig 5.j Tables 2

CHAPTER 6

CONCLUSIONS

The application for the **STAY EASE – HOTEL SERVICES** has been successfully implemented with the database in backend and the python GUI tkinter in the front end of the application. The objectives of the application has been fulfilled. The application increase efficiency of managing the rooms booking. It provide the searching facilities based on various factors, such as booking , Customer, payment services. It tracks all the information of the hotel room and payment etc. Manage the information of all aspects of the hotel. Show the information and description of the booking and customers details. Editing, adding and updating of records. Proper resource management of the booking data.

REFERENCES

[1] https://www.w3schools.com/python/python_mysql_getstarted.asp

[2]https://www.geeksforgeeks.org/pythonguitkinter/

[3] https://pythonspot.com/mysql-with-python/

[4] https://stackoverflow.coms

Other python and mysql database online websites.