MAJOR PROJECT REPORT

(BCAN-691)

ON

Hotel management system

Session: 2019-2022

UNDER THE GUIDANCE OF:

Mr. Tapan Kumar Das

Assistant Professor

Department of Computer Application

SUBMITTED BY: ***(BCA 6th Sem)***

SIDDHARTHA KHAWAS (32201219075)

SHIVENDU SHIVAM (32201219074)

SUBHAM CHAKRABORTY(32201219061)

DIPAK KUMAR YADAV (32201219055)

RAHUL KUMAR (32201219012)

ASANSOL ENGINEERING COLLEGE (AEC)

Asansol, West Bengal

(Affiliated to Maulana Abul Kalam Azad University of Technology)

(Kolkata, West Bengal)

June 2022

Department of Computer Applications

Asansol Engineering College

Kalyanpur, Sen Raleigh Road, Asansol – 713304

CERTIFICATE

*This is to certify that the project work entitled "**HOTEL MANAGEMENT SYSTEM**" is a bonafide record of work carried out in the **Department of Computer Application**, Asansol Engineering College, Asansol.*

Name	*Roll Number*	*Registration Number*
SIDDHARTHA KHAWAS	32201219075	035811
SHIVENDU SHIVAM	32201219074	035771
SUBHAM CHAKRABORTY	32201219061	036112
DIPAK KUMAR YADAV	32201219055	036255
RAHUL KUMAR	32201219012	037177

The students of 6th Semester BCA 2021-22 under my / our supervision in requirement of partial fulfillment of the Award of Degree of BCA from Maulana Abul Kalam Azad University of Technology, West Bengal.

Signature of
The Project Guide
Name: **Dr./Mr. TAPAN KUMAR DAS**
Designation: Assistant Professor

Recommendation & Signature of
The Principal
Name: **Dr. P. P. Bhattacharya**

Recommendation &Signature of
The Head of Department
Name: **Dr. D. K. PAL**

Recommendation &Signature of
Internal / External Examiners

ACKNOWLEDGEMENT

We would like to express our special thanks and gratitude to our guide Mr. Tapan Kumar Das, Asst. Professor Department of Computer Application. We would like to thanks our HOD sir Dr. Dibyendu Kumar Pal to provide us the opportunity to carry out the project. We would also like to thanks our honorable principal sir Dr. P.P Bhattacharya who gave us the golden opportunity to do this wonderful project entitled "Hotel Management System", which also helped us in doing lots of research and we came to know about so many new things. We are really thankful to them.

We would also like to thank other faculty members of our Department, our parents and group members to complete the project within time.

Thank You all for your support.

SIDDHARTHA KHAWAS (32201219075) ----------

SHIVENDU SHIVAM (32201219074) ----------

SUBHAM CHAKRABORTY (32201219061) ----------

DIPAK KUMAR YADAV (32201219055) ----------

RAHUL KUMAR (32201219012) ----------

DECLARATION

We hereby declare that the project entitled- **"*Hotel management system*",** which is being submitted as Major Project of 6th semester in **Bachelor of Computer Applications (BCA**) to **Asansol Engineering College, Asansol (W.B)** is the record of our genuine work under the guidance of our guide **Mr. Tapan Kumar Das** Department of **Computer Application Asansol Engineering College, Asansol.**

SIDDHARTHA KHAWAS (32201219075)

SHIVENDU SHIVAM (32201219074)

SUBHAM CHAKRABORTY (32201219061)

DIPAK KUMAR YADAV (32201219055)

RAHUL KUMAR (32201219012)

Place:

Date:

Table of Content

1. TITLE OF THE PROJECT:

Hotel management system

2. INTRODUCTION OF THE PROJECT:

A hotel is a hive of numerous operations such as front office, booking, and reservation, banquet, finance, HR, inventory, material management, quality management, security, energy management, housekeeping, CRM and more. The hotel has some rooms, and these rooms are of different categories. By room category, each room hast the different price. A hotel has employees to manage the services provided to customers. It is quite important to store the customer record in hotel database which contains customer identity, his address, check in time, check out time, etc., for future and security purposes. Hotel provides food and beverages to their customers and generates the bill for this at the time of their check out.

Currently, there are many hotels which lack a common platform for all these functionalities to work together. Databases will be required in huge amount to store all this information and to be retrieved as and when required. Handling of such a huge database will also require a reliable and efficient software for application.

3. OBJECTIVE OF THE PROJECT:

Currently, there are many hotels which lack a common platform for all these functionalities to work together. Databases will be required in huge amount to store all this information and to be retrieved as and when required. Handling of such a huge database will also require a reliable and efficient software for application. It is capable of hosting almost all the features that can be required by a hotel.

4. HARDWARE AND SOFTWARE REQUIREMENTS:

The minimum hardware and software configuration for application to run is as follows.

HARDWARE:-

Microprocessor: - Intel® Core™ i3 processors (Or equivalent)

Ram:- 4 GB

Hard Drive: - 500 GB

SOFTWARE:-

Operating System: - WINDOWS 10/11

Languages: - Python, MySQL

5. Purpose of The Project:

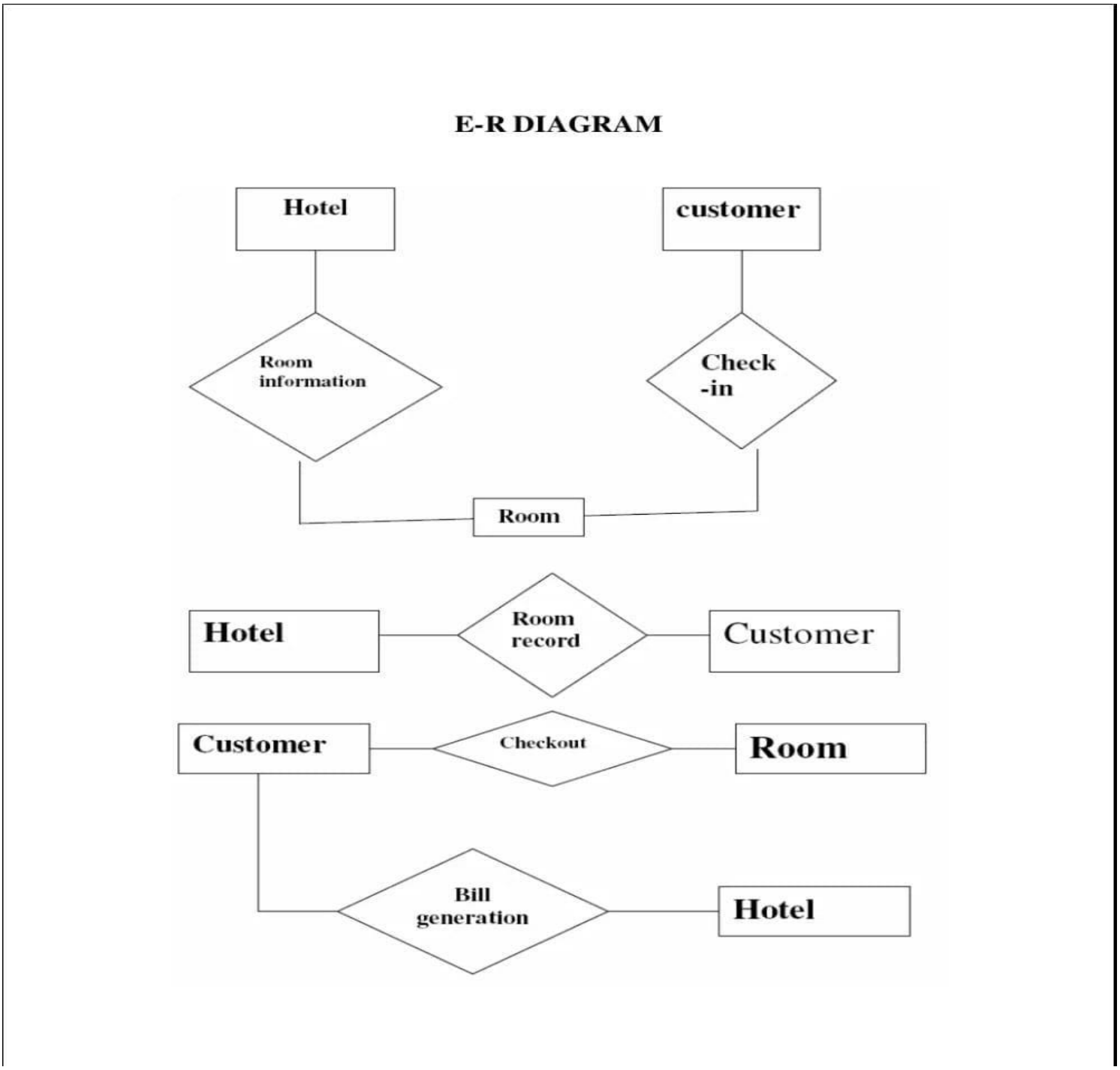
The system aims at the maintenance and management of the different Hotels. It mainly takes care of the Hotel management at the core area of the database.

In the front end we using Python and in the backend we using mysql under the Microsoft windows operating system.

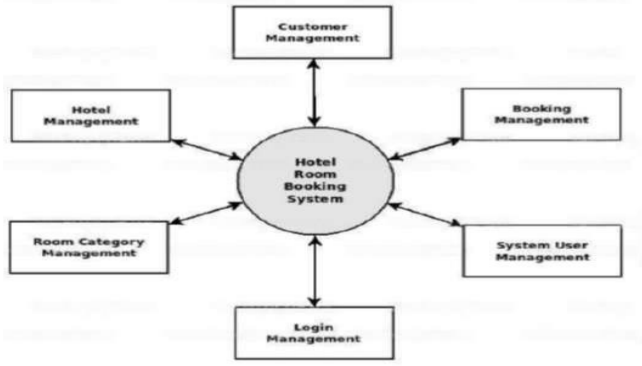
Features of this project:

- It saves user time in searching a room.
- The system is useful as it calculates an exact cost of rooms for requested number of days.
- It saves organization resources and expenses.
- This system is effective and saves time and cost of users.

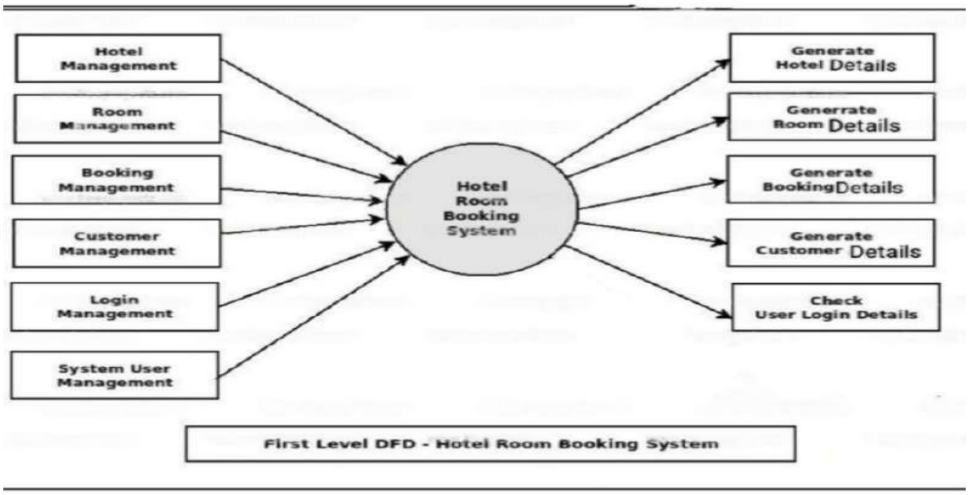
ER Diagrams


E-R DIAGRAM
Hotel
customer
Room information
Check -in
Room
Hotel
Room record
Customer
Customer
Checkout
Room
Bill generation
Hotel

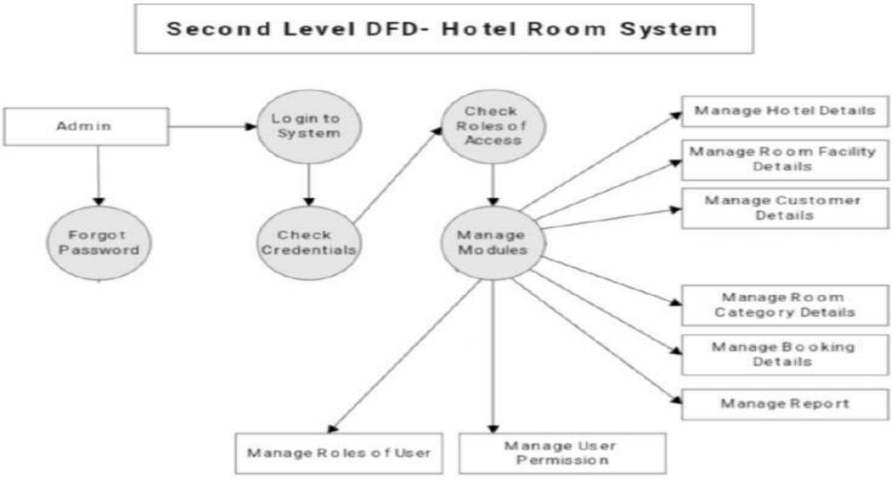
Dataflow Diagrams



Zero Level DFD - Hotel Room Booking System



First Level DFD - Hotel Room Booking System



Login to Hotel management system

This part of paper includes security for hotel management system which contains username &password implemented.

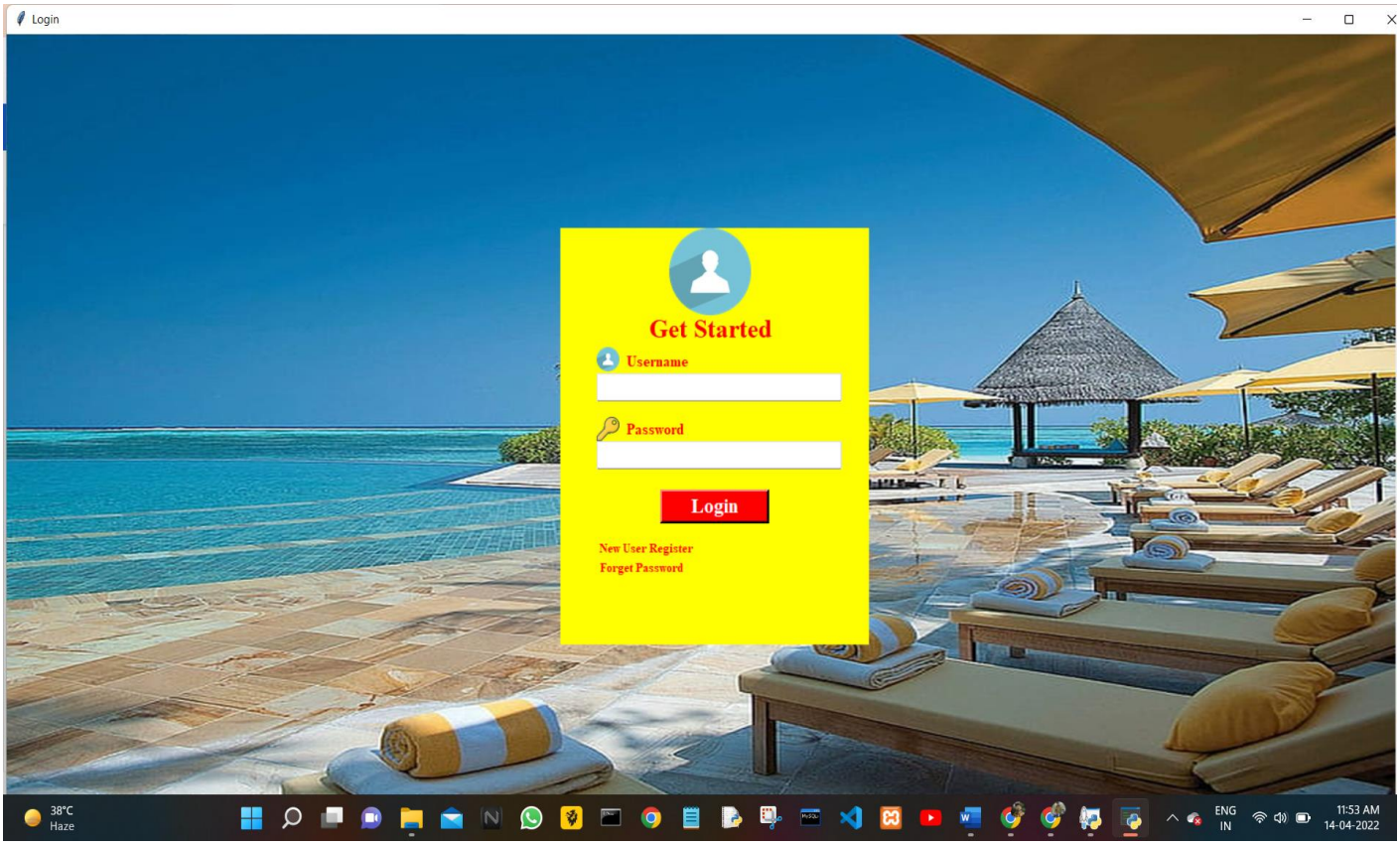
This design includes User name and Password, New User Register also forget password.

If the user entered the wrong password or the wrong user id, a message appears that the password and user id is wrong. We have created the login page using python and MySQL and then link the python with MySQL database, so if someone enters the right user's name and password then it will redirect to the hotel management system.

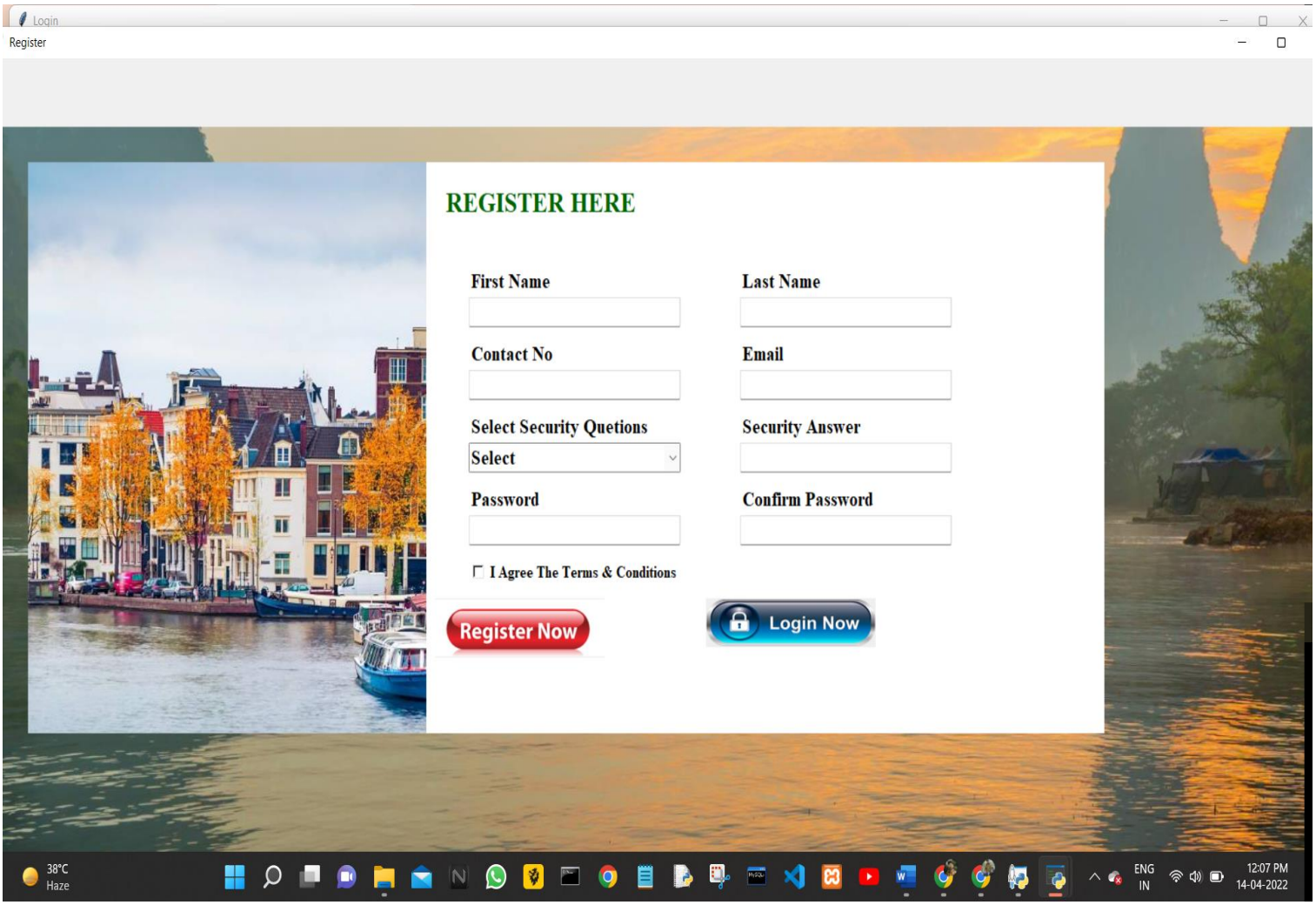
User also create a new user id and password.

We also provided another option the name is forget password. When the user can't remember his password then click this button. This option provided the user how to create a new password.

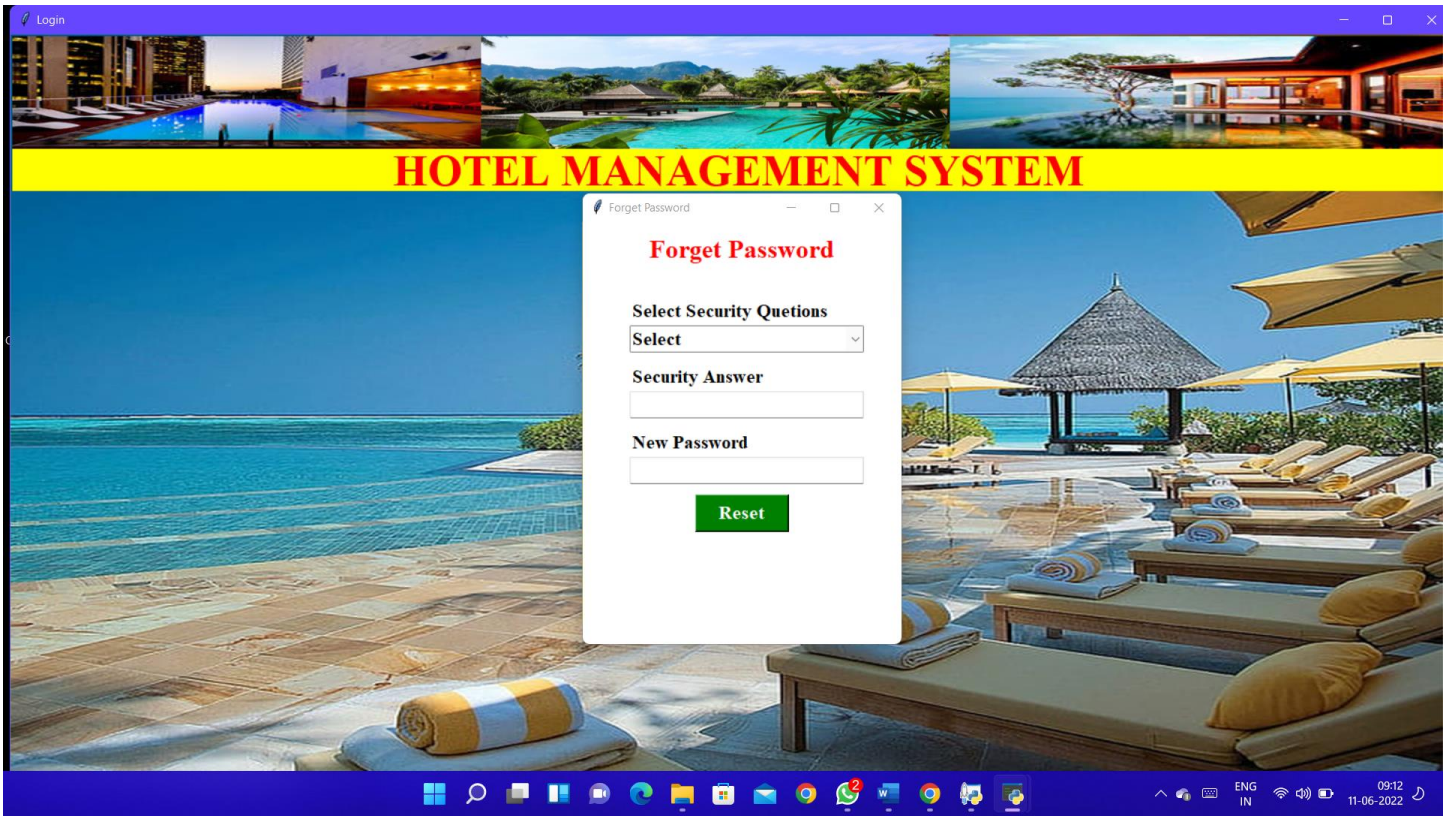
-:Login Window: -



-: New user window: -



-: Forget password portal: -



-: ***Database login table***: -

```
mysql> select * from register;
+-------------+------------+------------+-------------+--------------+------------+----------+
| fname       | lname      | contact    | email       | security_Q   | security_A | password |
+-------------+------------+------------+-------------+--------------+------------+----------+
|  DIPAK KUMAR | YADAV      | 8201219054 | dy@gmail.com | Your Pet Name | cat        | dy@1234  |
| RAHUL       | KUMAR      | 3201219012 | rk@gmail.com | Your Pet Name | dog        | rk@1234  |
| Mr.Subham   | Chakraborty | 9563742370 | sc@gmail.com | Your Pet Name | Dog        | 1234567  |
| SIDDHARTHA  | KHAWAS     | 6220121907 | sk@gmail.com | Your Pet Name | Cat        | sk@1234  |
| SHIVENDU    | SHIVAM     | 2201219065 | ss@gmail.com | Your Pet Name | Dog        | ss@1234  |
+-------------+------------+------------+-------------+--------------+------------+----------+
5 rows in set (0.00 sec)
```

-: ***Python code of login window***: -

```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk    #pip install pillow
from tkinter import messagebox
import random
import time
import datetime
import mysql.connector
from hotel import HotelManagementSystem




def main():
    win=Tk()
    app=Login_Window(win)
    win.mainloop()


class Login_Window:
    def __init__(self,root):
        self.root=root
        self.root.title("Login")
        self.root.geometry("1550x800+0+0")
```

```
        img1 = Image.open("images/ho1.jpg")

        img1 = img1.resize((1530,800), Image.ANTIALIAS)

        self.photoImg1 = ImageTk.PhotoImage(img1)

        bg_lbl=Label(self.root,image=self.photoImg1)

        bg_lbl.place(x=0,y=0,width=1530,height=800)


        title=Label(bg_lbl,text="ONLINE HOTEL MANAGEMENT SYSTEM",font=("times new
roman",35,"bold"),bg="yellow",fg="red")

        title.place(x=0,y=120,width=1550,height=45)


         # ==================== Project buttom(description)
================================================


        img10 = Image.open("images/image1.jpg")

        img10 = img10.resize((500,120), Image.ANTIALIAS)

        self.photoImg10 = ImageTk.PhotoImage(img10)

        bg_lbl1=Label(bg_lbl,image=self.photoImg10)

        bg_lbl1.place(x=0,y=0,width=500,height=120)


        img11 = Image.open("images/images.jpg")

        img11 = img11.resize((500,120), Image.ANTIALIAS)

        self.photoImg11 = ImageTk.PhotoImage(img11)
```

```
bg_lbl22=Label(bg_lbl,image=self.photoImg11)

bg_lbl22.place(x=500,y=0,width=500,height=120)


img13 = Image.open("images/download (5).jpg")

img13 = img13.resize((550,120), Image.ANTIALIAS)

self.photoImg13= ImageTk.PhotoImage(img13)

bg_lbl12=Label(bg_lbl,image=self.photoImg13)

bg_lbl12.place(x=1000,y=0,width=550,height=120)



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

frame.place(x=610,y=200,width=340,height=430)


img1=Image.open("images/LoginIconAppl.ico")

img1=img1.resize((90,90),Image.ANTIALIAS)

self.photoimage1=ImageTk.PhotoImage(img1)

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

lblimg1.place(x=730,y=200,width=90,height=90)


get_str=Label(frame,text="Get Started",font=("times new roman",20,"bold"),fg="red",bg="yellow")

get_str.place(x=95,y=85)


# label

username=lbl=Label(frame,text="Username",font=("times new roman",12,"bold"),fg="red",bg="yellow")
```

```
username.place(x=70,y=125)



self.txtuser=StringVar()

self.txtpass=StringVar()


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

txtuser.place(x=40,y=150,width=270)


password=lbl=Label(frame,text="Password",font=("times new roman",12,"bold"),fg="red",bg="yellow")

password.place(x=70,y=195)


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

txtpass.place(x=40,y=220,width=270)


# ======Icon Images=================

img2=Image.open("images/LoginIconAppl.ico")

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

self.photoimage2=ImageTk.PhotoImage(img2)

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

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



img3=Image.open("images/lock-512.ico")
```

```
        img3=img3.resize((25,25),Image.ANTIALIAS)

        self.photoimage3=ImageTk.PhotoImage(img3)

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

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



        # LoginButton


btn_login=Button(frame,text="Login",borderwidth=3,relief=RAISED,command=self.login,cursor="hand2",font=("times new roman",16,"bold"),fg="white",bg="red" ,activebackground="#B00857")

        btn_login.place(x=110,y=270,width=120,height=35)

        # registerbutton

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

        registerbtn.place(x=15,y=320,width=160)



        # forgetpassbtn

        registerbtn=Button(frame,text="Forget Password",command=self.forgot_password_window,font=("times new roman",10,"bold"),borderwidth=0,fg="red",bg="yellow",activeforeground="white",activebackground="black")

        registerbtn.place(x=10,y=340,width=160)



    def rigister_window(self):

        self.new_window=Toplevel(self.root)

        self.app=Register( self.new_window)



    def login(self):

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

```
    messagebox.showerror("Error","all field required")
elif self.txtuser.get()=="Skkk" and self.txtpass.get()=="shiv":
    messagebox.showinfo("Success","Welcome")
else:
    conn=mysql.connector.connect(host="localhost",user="root",password="sc956374",database="mydata")
    my_cursor=conn.cursor()
    my_cursor.execute("select * from register where email=%s and password=%s",(
                                                self.txtuser.get(),
                                                self.txtpass.get()
                                            ))

    row=my_cursor.fetchone()
    # print(row)
    if row==None:
        messagebox.showerror("Error","Inavalid Username & password")
    else:
        open_main=messagebox.askyesno("YesNo","Access only Authority Person")
        if open_main>0:
            self.new_window=Toplevel(self.root)     # Note: Connect Your Main Project
            self.app=HotelManagementSystem(self.new_window)
        else:
            if not open_main:
                return
    conn.commit()
```

```
        self.clear()
        conn.close()


def clear(self):
    self.txtuser.set("")
    self.txtpass.set("")


# ====================reset password=======================================
def reset_pass(self):
    if self.combo_securiy_Q.get()=="Select" or self.txt_security.get()=="" or self.txt_newpass=="":
        messagebox.showerror("Error","All fields are required",parent=self.root2)
    else:
        try:
            conn=mysql.connector.connect(host="localhost",user="root",password="sc956374",database="mydata")
            cur=conn.cursor()
            query=("select * from register where email=%s and securityQ=%s and securityA=%s")
            value=(self.txtuser.get(),self.combo_securiy_Q.get(),self.txt_security.get(),)
            cur.execute(query,value)
            row=cur.fetchone()
            # print(row)
            if row==None:
                messagebox.showerror("Error","Please select the correct security quetion/Enter answer",parent=self.root2)
            else:
                query=("update register set password=%s where email=%s")
```

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

            cur.execute(query,value)

            conn.commit()

            conn.close()

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

            self.root2.destroy()




        except Exception as es:

           messagebox.showerror("Error",f"Error Due To:{str(es)}",parent=self.root2)




  # ============================================forgrt password window============================

  def forgot_password_window(self):

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

        messagebox.showerror("Error","Plaese Enter the Email address to reset password")

     else:

        conn=mysql.connector.connect(host="localhost",user="root",password="sc956374",database="mydata")

        my_cursor=conn.cursor()

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

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

        my_cursor.execute(query,value)

        row=my_cursor.fetchone()

        # print(row)
```

```
        if row==None:
            messagebox.showerror("My Error","Plaese enter the valid user name")
        else:
            conn.close()
            self.root2=Toplevel()
            self.root2.title("Forget Password")
            self.root2.geometry("340x450+610+200")
            self.root2.configure(bg="white")

            l=Label(self.root2,text="Forget Password",font=("times new roman",20,"bold"),fg="red",bg="white")
            l.place(x=0,y=10,relwidth=1)

            security_Q=Label(self.root2,text="Select Security Quetions",font=("times new
roman",15,"bold"),bg="white",fg="black")
            security_Q.place(x=50,y=80)

            self.combo_securiy_Q=ttk.Combobox(self.root2,font=("times new roman",15,"bold"),state="readonly")
            self.combo_securiy_Q["values"]=("Select","Your Birth Place","Your Father name","Your Pet Name")
            self.combo_securiy_Q.place(x=50,y=110,width=250)
            self.combo_securiy_Q.current(0)

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

```
        security_A.place(x=50,y=150)



        self.txt_security=ttk.Entry(self.root2,font=("times new roman",15,"bold"))

        self.txt_security.place(x=50,y=180,width=250)




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

        new_password.place(x=50,y=220)



        self.txt_newpass=ttk.Entry(self.root2,font=("times new roman",15,"bold"))

        self.txt_newpass.place(x=50,y=250,width=250)



        btn=Button(self.root2,text="Reset",command=self.reset_pass,font=("times new
roman",15,"bold"),fg="White",bg="green")

        btn.place(x=120,y=290,width=100)






class Register:

    def __init__(self,root):

        self.root=root

        self.root.title("Register")

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

```
# ==================varibles====================

self.var_fname=StringVar()

self.var_lname=StringVar()

self.var_contact=StringVar()

self.var_email=StringVar()

self.var_securityQ=StringVar()

self.var_SecurityA=StringVar()

self.var_pass=StringVar()

self.var_confpass=StringVar()




# ============bg image==================

self.bg=ImageTk.PhotoImage(file="images/register_background.jpg")

bg_lbl=Label(self.root,image=self.bg)

bg_lbl.place(x=0,y=0,relwidth=1,relheight=1)




# ============left image==================

self.bg1=ImageTk.PhotoImage(file="images/register_left_background.jpg")

left_lbl=Label(self.root,image=self.bg1)

left_lbl.place(x=50,y=100,width=470,height=550)


# ==============main Frame====================
```

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

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


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

register_lbl.place(x=20,y=20)


# ==============label and entry============================


# ------------------row1

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

fname.place(x=50,y=100)


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

self.fname_entry.place(x=50,y=130,width=250)


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

l_name.place(x=370,y=100)


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

self.txt_lname.place(x=370,y=130,width=250)


# ------------------row2


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

```
        contact.place(x=50,y=170)


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

        self.txt_contact.place(x=50,y=200,width=250)


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

        email.place(x=370,y=170)


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

        self.txt_email.place(x=370,y=200,width=250)


        # ------------------row3


        security_Q=Label(frame,text="Select Security Quetions",font=("times new
roman",15,"bold"),bg="white",fg="black")

        security_Q.place(x=50,y=240)


        self.combo_securiy_Q=ttk.Combobox(frame,textvariable=self.var_securityQ,font=("times new
roman",15,"bold"),state="readonly")

        self.combo_securiy_Q["values"]=("Select","Your Birth Place","Your Father name","Your Pet Name")

        self.combo_securiy_Q.place(x=50,y=270,width=250)

        self.combo_securiy_Q.current(0)



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

```
        security_A.place(x=370,y=240)


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

        self.txt_security.place(x=370,y=270,width=250)


        # ----------------------row4


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

        pswd.place(x=50,y=310)


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

        self.txt_pswd.place(x=50,y=340,width=250)


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

        confirm_pswd.place(x=370,y=310)


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

        self.txt_confirm_pswd.place(x=370,y=340,width=250)


        # ====================checkbutton=================

        self.var_check=IntVar()

        self.checkbtn=Checkbutton(frame,variable=self.var_check,text="I Agree The Terms &
Conditions",bg='white',font=("times new roman",12,"bold"),onvalue=1,offvalue=0)

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

```
        # =================buttons=======================

        img=Image.open("images/register-now-button1.jpg")

        img=img.resize((200,55),Image.ANTIALIAS)

        self.photoimage=ImageTk.PhotoImage(img)

        b1=Button(frame,image=self.photoimage,command=self.register_data,borderwidth=0,cursor="hand2",font=("times new roman",15,"bold"),fg="white")

        b1.place(x=10,y=420,width=200)




        img1=Image.open("images/loginpng.png")

        img1=img1.resize((200,45),Image.ANTIALIAS)

        self.photoimage1=ImageTk.PhotoImage(img1)


b1=Button(frame,image=self.photoimage1,command=self.return_login,borderwidth=0,cursor="hand2",font=("times new roman",15,"bold"),fg="white")

        b1.place(x=330,y=420,width=200)



    # =================Function declaration=============================


    def register_data(self):

        if self.var_fname.get()=="" or self.var_email.get()=="" or self.var_securityQ.get()=="Select":

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

        elif self.var_pass.get()!=self.var_confpass.get():

            messagebox.showerror("Error","password & confirm password must be same",parent=self.root)

        elif self.var_check.get()==0:
```

```
    messagebox.showerror("Error","Plaese agree our terms ane condition",parent=self.root)
else:
    conn=mysql.connector.connect(host="localhost",user="root",password="sc956374",database="mydata")
    my_cursor=conn.cursor()
    query=("select * from register where email=%s")
    value=(self.var_email.get(),)
    my_cursor.execute(query,value)
    row=my_cursor.fetchone()
    if row!=None:
        messagebox.showerror("Error","User already exist,plaese try another email",parent=self.root)
    else:
        my_cursor.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s)",(
                                                self.var_fname.get(),
                                                self.var_lname.get(),
                                                self.var_contact.get(),
                                                self.var_email.get(),
                                                self.var_securityQ.get(),
                                                self.var_SecurityA.get(),
                                                self.var_pass.get()

                                            ))
    conn.commit()
    conn.close()
    messagebox.showinfo("Success","Register Successfully",parent=self.root)
```

```
    def return_login(self):
        self.root.destroy()


if __name__ == "__main__":
    main()
```

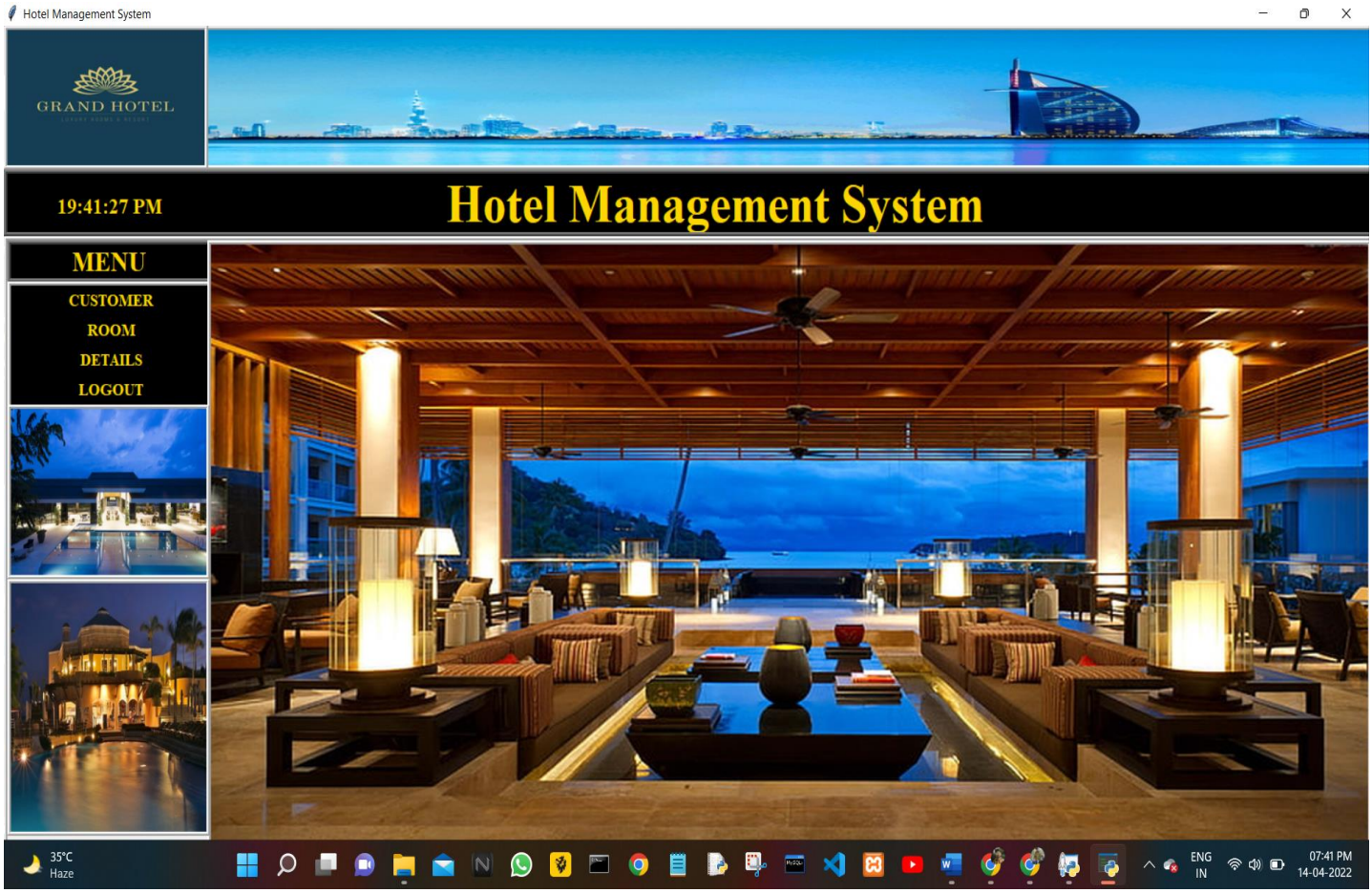
-: *MySQL code of login window*: -

```
create database mydata;


use mydata;


CREATE TABLE `register` (
  `fname` varchar(45) DEFAULT NULL,
  `lname` varchar(45) DEFAULT NULL,
  `contact` varchar(45) DEFAULT NULL,
  `email` varchar(45) NOT NULL,
  `security_Q` varchar(45) DEFAULT NULL,
  `security_A` varchar(45) DEFAULT NULL,
  `password` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

-:Hotel Management System Window:-



- **In this window we have four options:**

1. **Customer.**
2. **Room.**
3. **Details.**
4. **Logout.**

- **And we discuss about the four options below**.

- ***Customer: -***

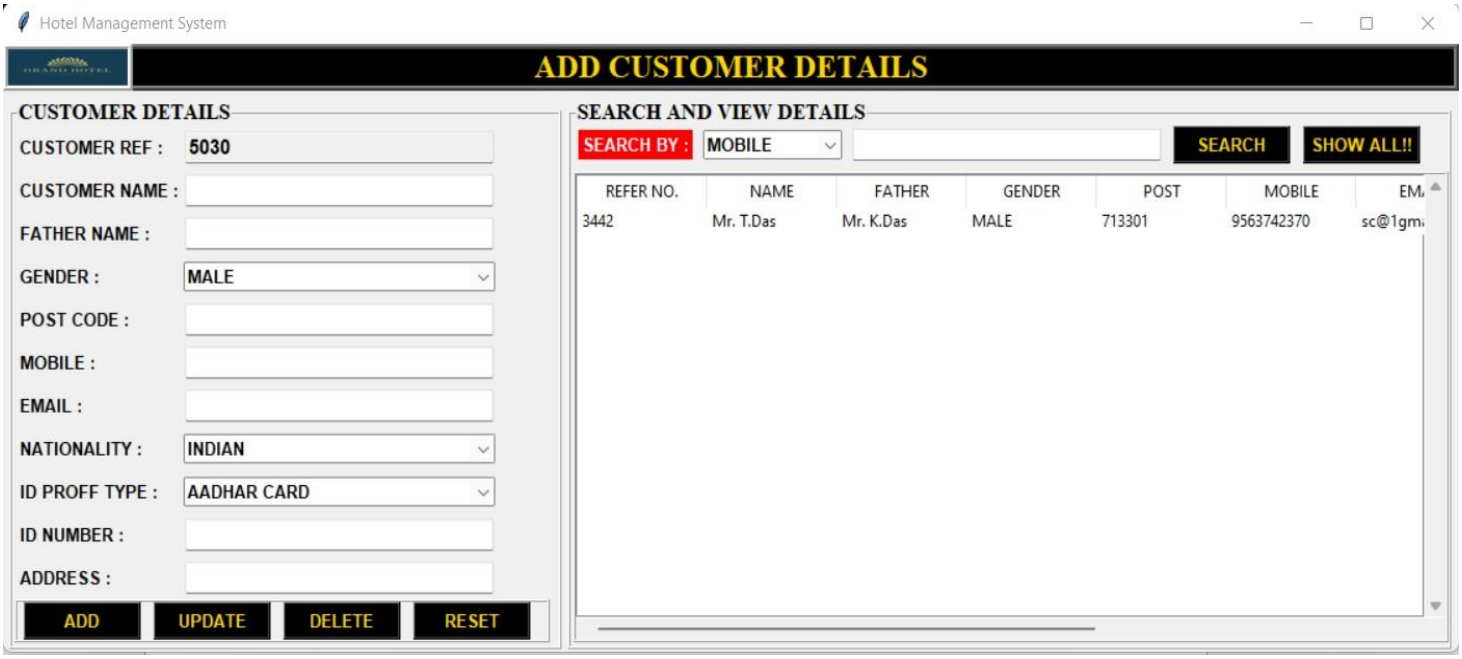
  When we want to insert the customers details in the database then we use the customer option. This option provided many features like (Customer Ref No, customer Name, Customer Father Name, Customer Gender, Postal code, Customer

mobile number, Customer email, Customer nationality, Customer ID Proof Type, ID Number and Customer Address) this all functions are help full for user to insert the data.

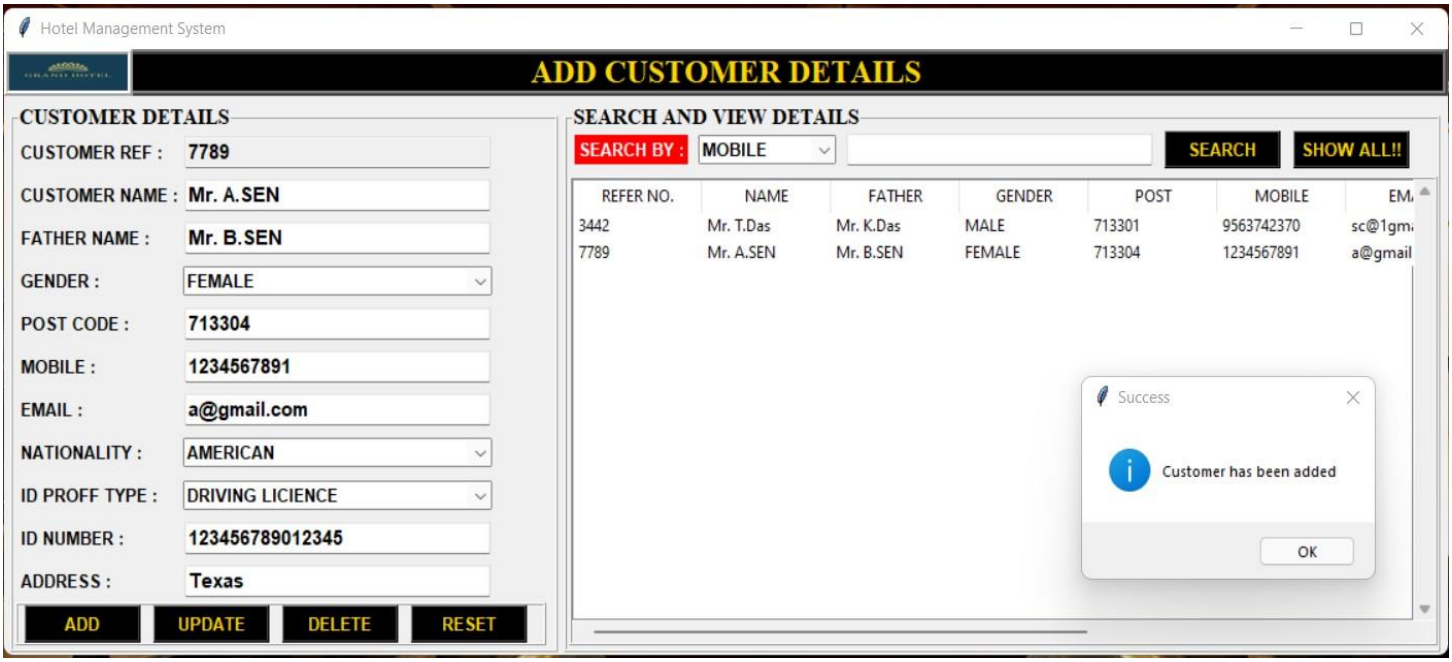
★ **Duplicate entries are not allowed here.**

★ **We have four buttons in customer window. The buttons are ADD, UPDATE DELETE and RESET. below we discuss the features of four button.**

- ***Customer window:-***



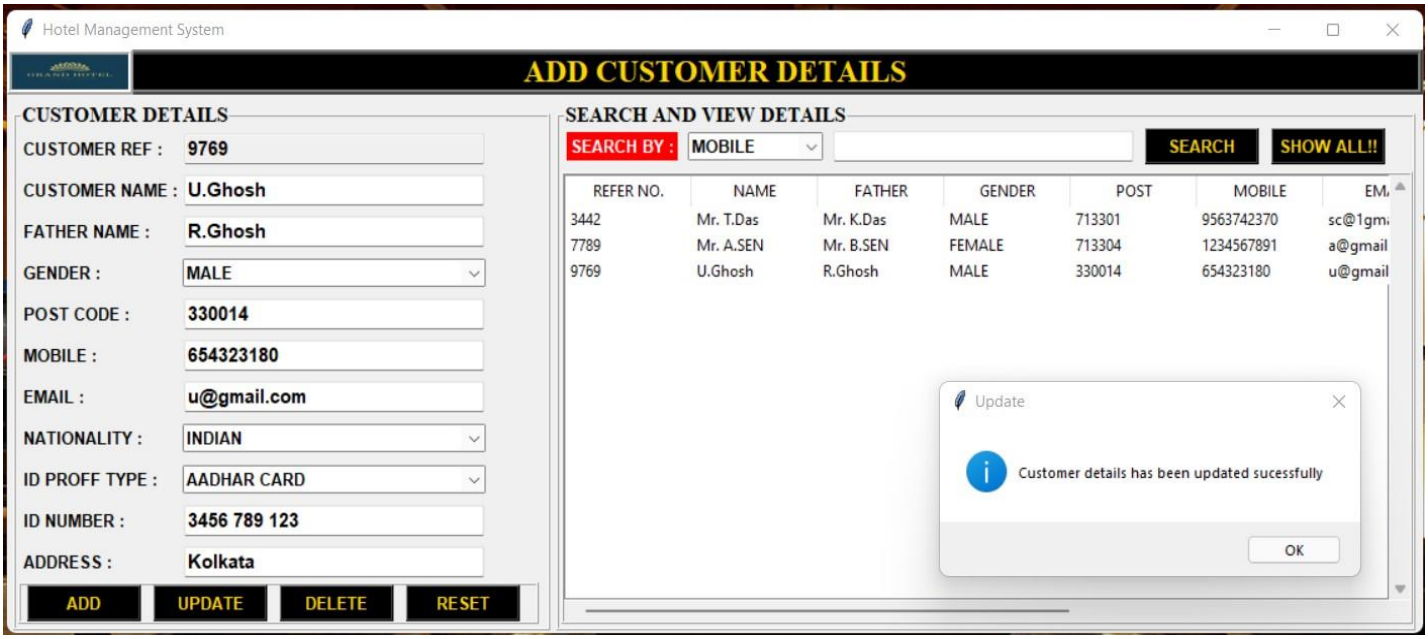
- ***Insert Customer data(Add button): -***



- Here we insert the customer data's and after inserted the message will be pop up "**Customer has added".**

- ***<u>Update button:-</u>***

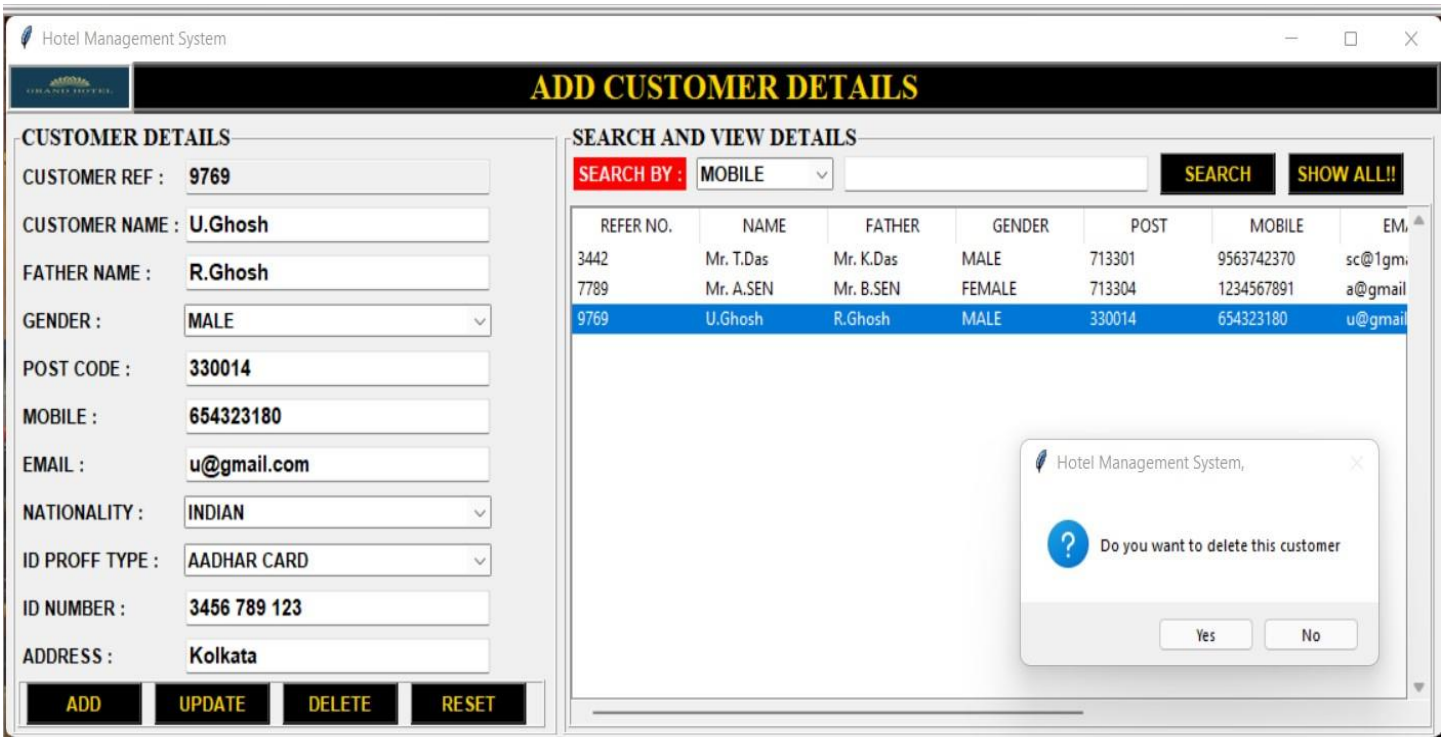
When the user wants to update the customer data for any reason then he can use the update button.



- after updated the message will be show "**customer details has been updated successfully."**

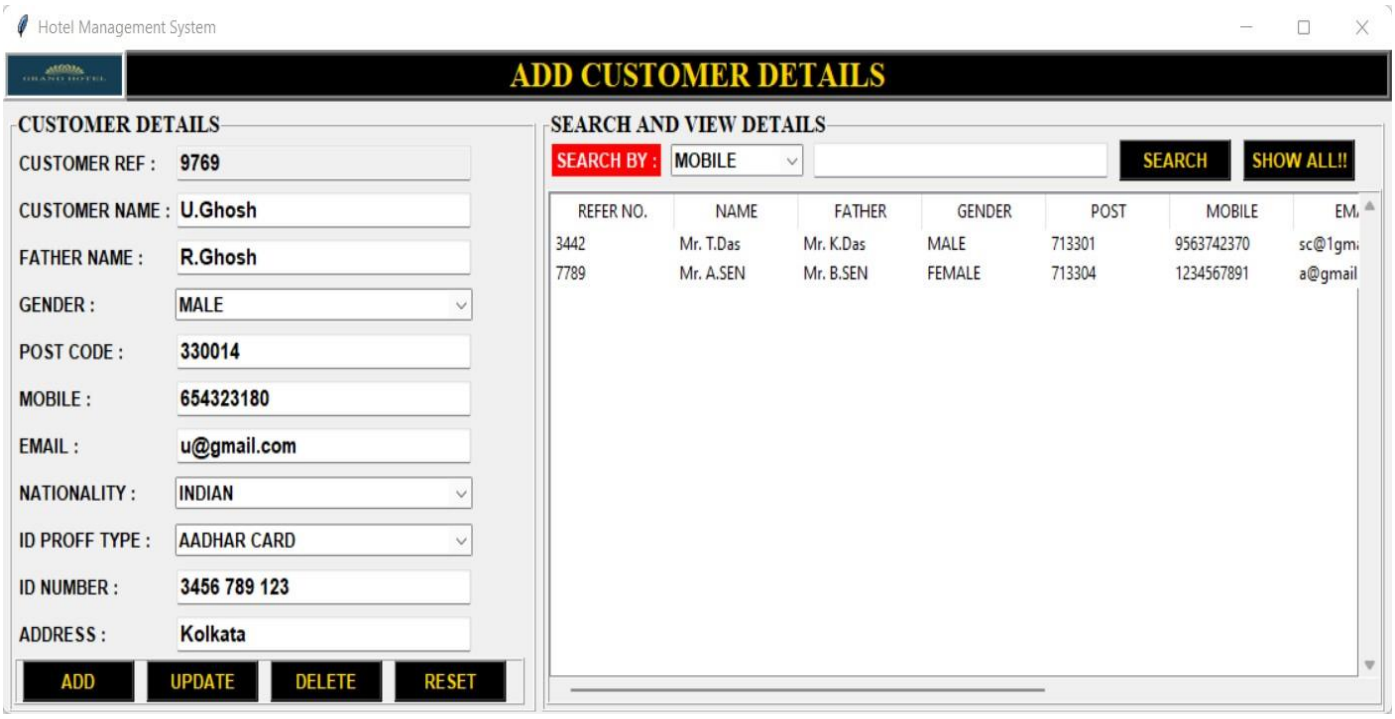
- ***<u>Delete button:-</u>***

If we want to delete some data then the user can use delete button.



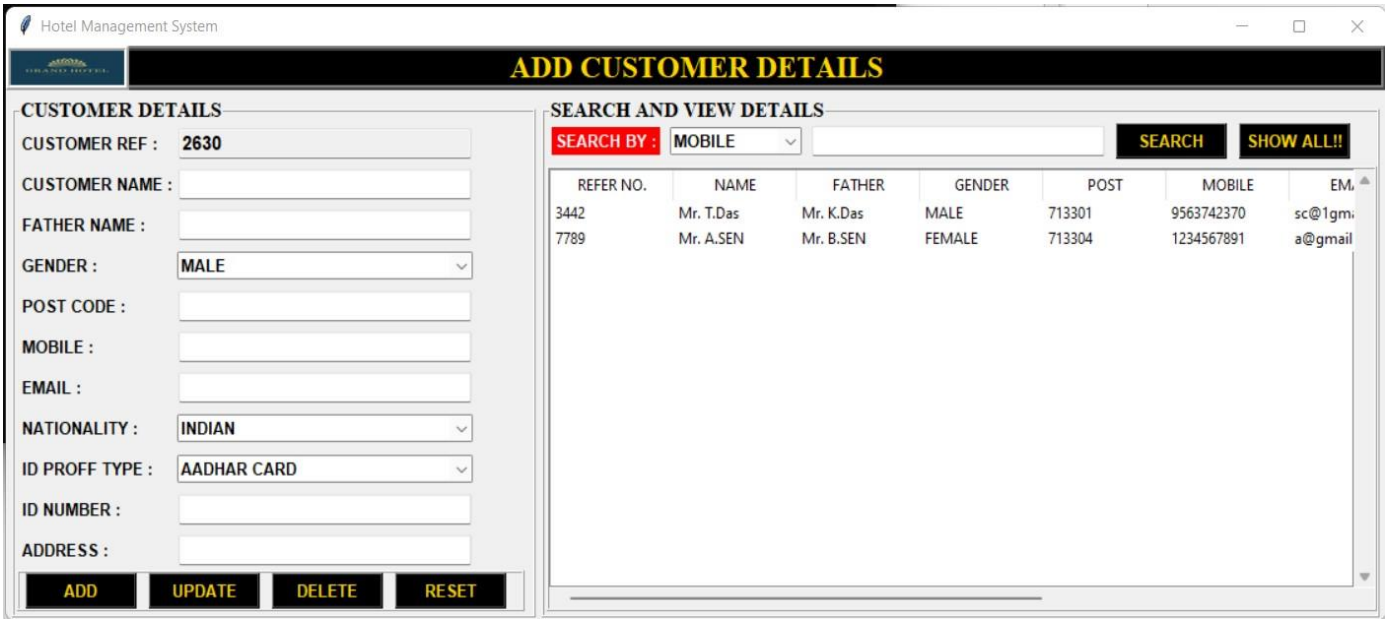
- ***After deleting data: -***

  After deleting the data REF NO. 9769 Name: U.Ghosh is no longer in the data base.



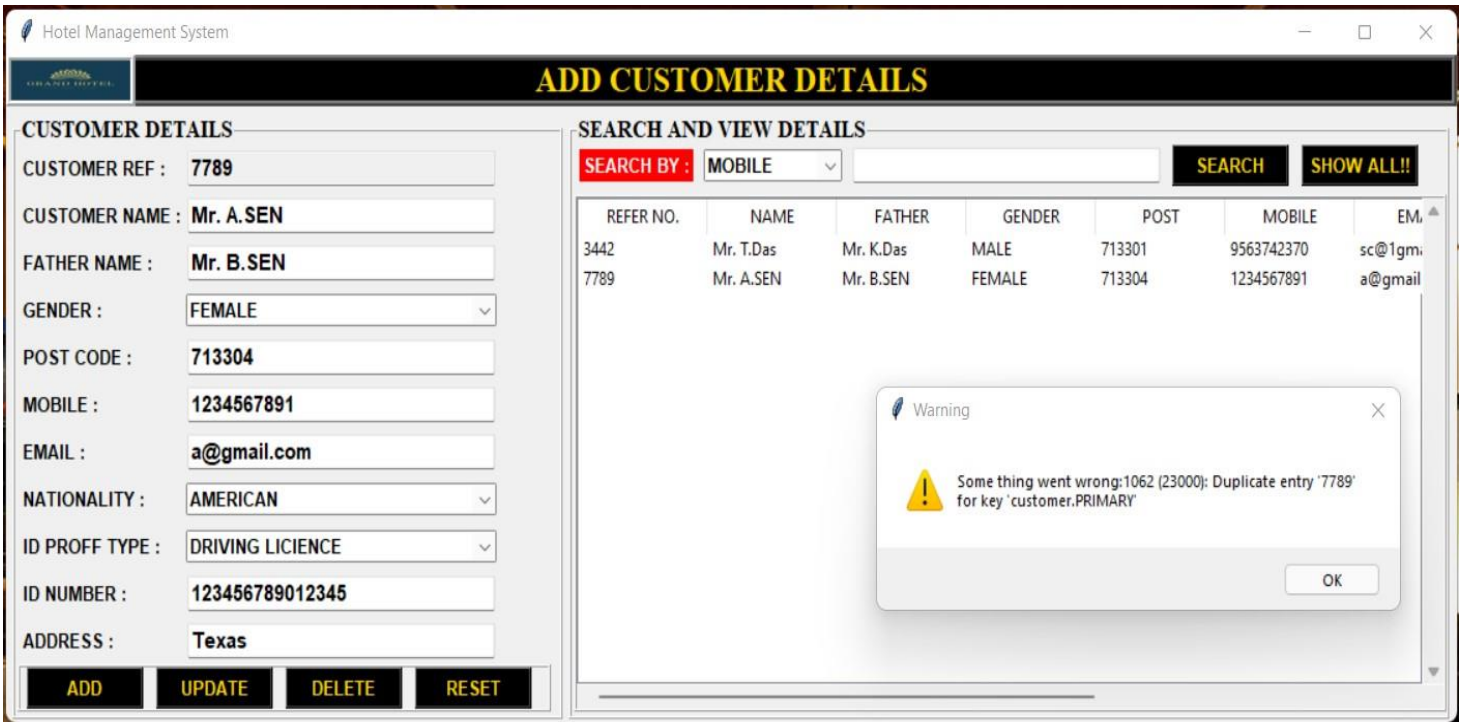
- ***Reset button: -***

This button used to clear the text area.



- ***<u>warning for duplicate entry:-</u>***

When a user intentionally or unintentionally uses the same REF Number or Same phone number the time of data's entry of the customer then he will be given a warning and he can’t enter the data In the system.



Python code of Customer window:-

```
 from tkinter import*
from PIL import Image, ImageTk #pip install pillow
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("1121x452+234+243")


        ################ VARIABLES ################
        self.var_ref = StringVar()
        x = random.randint(1000,9999)
        self.var_ref.set(str(x))
```

```
        self.var_cust_name = StringVar()

        self.var_cust_father = StringVar()

        self.var_cust_gender = StringVar()

        self.var_cust_post = StringVar()

        self.var_cust_mobile = StringVar()

        self.var_cust_email = StringVar()

        self.var_cust_nationality = StringVar()

        self.var_cust_id_proff = StringVar()

        self.var_cust_id_number = StringVar()

        self.var_cust_address = StringVar()



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

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

        lbl_title.place(x=0,y=0,width=1121,height=35)


        ################### Logo ######################

        img2 = Image.open("logohotel.png")

        img2 = img2.resize((100,35),Image.ANTIALIAS)

        self.photoimg2=ImageTk.PhotoImage(img2)
```

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

        lblimg.place(x=0,y=0,width=100,height=35)



        ################ LABEL FRAME ##################

        lblFrameLeft=LabelFrame(self.root,bd=2,relief=RIDGE,text="CUSTOMER
DETAILS",font=("times new roman",12,"bold"),padx=2)

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



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



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

        lbl_cust_ref = Label(lblFrameLeft,text="CUSTOMER REF
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        entry_ref =
ttk.Entry(lblFrameLeft,textvariable=self.var_ref,width=29,font=("arial",11,"bold"),state="reado
nly")

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



        ############## CUSTOMER NAME #############

        lbl_cust_name = Label(lblFrameLeft,text="CUSTOMER NAME
:",font=("arial",10,"bold"),padx=2,pady=6)
```

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



        entry_name =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_name,width=29,font=("arial",11,"bold"))

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



        ############## FATHER NAME ###############

        lbl_cust_mname = Label(lblFrameLeft,text="FATHER NAME
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        entry_fname =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_father,width=29,font=("arial",11,"bold"))

        entry_fname.grid(row=2,column=1)



        ############# GENDER BOX ###############

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

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



combo_gender=ttk.Combobox(lblFrameLeft,textvariable=self.var_cust_gender,font=("arial",10
,"bold"),width=31,state="readonly")

        combo_gender["value"]=("MALE","FEMALE","OTHERS")
```

```
        combo_gender.current(0)

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



        ########### POST CODE ###############

        lbl_cust_post = Label(lblFrameLeft,text="POST CODE
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        entry_post =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_post,width=29,font=("arial",11,"bold"))

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



        ########## MOBILE NUMBER ##############

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

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



        entry_mob =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_mobile,width=29,font=("arial",11,"bold"))

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



        ########## EMAIL ##################
```

```
        lbl_cust_email = Label(lblFrameLeft,text="EMAIL
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        entry_email =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_email,width=29,font=("arial",11,"bold"))

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



        ############ NATIONALITY ###############

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

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



combo_nationality=ttk.Combobox(lblFrameLeft,textvariable=self.var_cust_nationality,font=("a
rial",10,"bold"),width=31,state="readonly")

        combo_nationality["value"]=("INDIAN","AMERICAN","BRITISH")

        combo_nationality.current(0)

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



        ############# IDPROFF TYPE #############

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

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

```
combo_idproff=ttk.Combobox(lblFrameLeft,textvariable=self.var_cust_id_proff,font=("arial",10,"bold"),width=31,state="readonly")

        combo_idproff["value"]=("AADHAR CARD","PASSPORT","DRIVING LICIENCE","PAN CARD")

        combo_idproff.current(0)

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



        ############# ID NUMBER ##################

        lbl_cust_idno = Label(lblFrameLeft,text="ID NUMBER :",font=("arial",10,"bold"),padx=2,pady=6)

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



        entry_idno = ttk.Entry(lblFrameLeft,textvariable=self.var_cust_id_number,width=29,font=("arial",11,"bold"))

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



        ############ ADDRESS #################

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

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

```
        entry_addr =
ttk.Entry(lblFrameLeft,textvariable=self.var_cust_address,width=29,font=("arial",11,"bold"))

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



        ############### BUTTONS ###############

        btn_frame = Label(lblFrameLeft,bd=2,relief=RIDGE)

        btn_frame.place(x=0,y=352,width=412,height=32)



        btn_add =
Button(btn_frame,text="ADD",command=self.add_data,font=("arial",10,"bold"),bg="black",fg
="gold",width=10)

        btn_add.grid(row=0,column=0,padx=5)



        btn_update =
Button(btn_frame,text="UPDATE",command=self.update,font=("arial",10,"bold"),bg="black",f
g="gold",width=10)

        btn_update.grid(row=0,column=1,padx=5)



        btn_delete =
Button(btn_frame,text="DELETE",command=self.dat_Delete,font=("arial",10,"bold"),bg="black
",fg="gold",width=10)

        btn_delete.grid(row=0,column=2,padx=5)
```

```
        btn_reset =
Button(btn_frame,text="RESET",command=self.data_reset,font=("arial",10,"bold"),bg="black",
fg="gold",width=10)

        btn_reset.grid(row=0,column=3,padx=5)



        ############### TABLE FRAME SEARCH ###############

        table_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="SEARCH AND VIEW
DETAILS",font=("times new roman",12,"bold"),padx=2)

        table_frame.place(x=435,y=40,width=680,height=410)



        lblsearchby = Label(table_frame,text="SEARCH BY
:",font=("arial",10,"bold"),bg="red",fg="white")

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



        self.search_var = StringVar()



combo_search=ttk.Combobox(table_frame,textvariable=self.search_var,font=("arial",10,"bold
"),width=12,state="readonly")

        combo_search["value"]=("MOBILE","REF_NO")

        combo_search.current(0)

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



        self.txt_search = StringVar()
```

```
        entry_search =
ttk.Entry(table_frame,textvariable=self.txt_search,width=29,font=("arial",11,"bold"))

        entry_search.grid(row=0,column=2,padx=4)



        btn_search =
Button(table_frame,text="SEARCH",command=self.search_data,font=("arial",10,"bold"),bg="b
lack",fg="gold",width=10)

        btn_search.grid(row=0,column=3,padx=5)



        btn_showall = Button(table_frame,text="SHOW
ALL!!",command=self.fetch_data,font=("arial",10,"bold"),bg="black",fg="gold",width=10)

        btn_showall.grid(row=0,column=4,padx=5)



        ################ SHOW DATA TABLE ##############

        details_table = Label(table_frame,bd=2,relief=RIDGE)

        details_table.place(x=0,y=34,width=674,height=350)



        Scroll_x=ttk.Scrollbar(details_table,orient=HORIZONTAL)

        Scroll_y=ttk.Scrollbar(details_table,orient=VERTICAL)



self.Cust_details_table=ttk.Treeview(details_table,columns=("REF_NO","NAME","FATHER","GE
NDER","POST","MOBILE","EMAIL","NATIONALITY","ID_PROFF","ID_NO","ADDRESS"),xscrollco
mmand=Scroll_x.set,yscrollcommand=Scroll_y.set)
```

```
Scroll_x.pack(side=BOTTOM,fill=X)

Scroll_y.pack(side=RIGHT,fill=Y)


Scroll_x.config(command=self.Cust_details_table.xview)

Scroll_y.config(command=self.Cust_details_table.yview)


self.Cust_details_table.heading("REF_NO",text="REFER NO.")

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

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

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

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

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

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

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

self.Cust_details_table.heading("ID_PROFF",text="ID PROFF.")

self.Cust_details_table.heading("ID_NO",text="ID NO.")

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


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


self.Cust_details_table.column("REF_NO",width=100)
```

```
    self.Cust_details_table.column("NAME",width=100)
    self.Cust_details_table.column("FATHER",width=100)
    self.Cust_details_table.column("GENDER",width=100)
    self.Cust_details_table.column("POST",width=100)
    self.Cust_details_table.column("MOBILE",width=100)
    self.Cust_details_table.column("EMAIL",width=100)
    self.Cust_details_table.column("NATIONALITY",width=100)
    self.Cust_details_table.column("ID_PROFF",width=100)
    self.Cust_details_table.column("ID_NO",width=100)
    self.Cust_details_table.column("ADDRESS",width=100)

    self.Cust_details_table.pack(fill=BOTH,expand=1)
    self.Cust_details_table.bind("<ButtonRelease-1>",self.get_cursor)
    self.fetch_data()



def add_data(self):
    if self.var_cust_mobile.get()=="" or self.var_cust_father.get()=="":
        messagebox.showerror("Error","ALL FIELDS ARE REQUIRED!!!",parent=self.root)
    else:
        try:
```

```
            conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database=
"hotel")

            my_cursor = conn.cursor()

            my_cursor.execute("insert into customer
values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(self.var_ref.get(),self.var_cust_name.get(),self.v
ar_cust_father.get(),self.var_cust_gender.get(),self.var_cust_post.get(),self.var_cust_mobile.g
et(),self.var_cust_email.get(),self.var_cust_nationality.get(),self.var_cust_id_proff.get(),self.va
r_cust_id_number.get(),self.var_cust_address.get()))

            conn.commit()

            self.fetch_data()

            conn.close()

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

        except Exception as es:

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



    def fetch_data(self):

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database=
"hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select * from customer")

        rows = my_cursor.fetchall()

        if len(rows)!= 0:
```

```
        self.Cust_details_table.delete(*self.Cust_details_table.get_children())
        for i in rows:
            self.Cust_details_table.insert("",END,values=i)
        conn.commit()
    conn.close()

def get_cursor(self,event=""):
    cursor_row = self.Cust_details_table.focus()
    content = self.Cust_details_table.item(cursor_row)
    row = content["values"]

    self.var_ref.set(row[0]),
    self.var_cust_name.set(row[1]),
    self.var_cust_father.set(row[2]),
    self.var_cust_gender.set(row[3]),
    self.var_cust_post.set(row[4]),
    self.var_cust_mobile.set(row[5]),
    self.var_cust_email.set(row[6]),
    self.var_cust_nationality.set(row[7]),
    self.var_cust_id_proff.set(row[8]),
    self.var_cust_id_number.set(row[9]),
```

```
        self.var_cust_address.set(row[10])


    def update(self):
        if self.var_cust_mobile.get()=="":
            messagebox.showerror("Error","Please enter mobile number",parent=self.root)
        else:
            conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database=
"hotel")
            my_cursor = conn.cursor()
            my_cursor.execute("update customer set
NAME=%s,FATHER=%s,GENDER=%s,POST=%s,MOBILE=%s,EMAIL=%s,NATIONALITY=%s,ID_PR
OFF=%s,ID_NO=%s,ADDRESS=%s where
REF_NO=%s",(self.var_cust_name.get(),self.var_cust_father.get(),self.var_cust_gender.get(),se
lf.var_cust_post.get(),self.var_cust_mobile.get(),self.var_cust_email.get(),self.var_cust_nation
ality.get(),self.var_cust_id_proff.get(),self.var_cust_id_number.get(),self.var_cust_address.get
(),self.var_ref.get()))
            conn.commit()
            self.fetch_data()
            conn.close()
            messagebox.showinfo("Update","Customer details has been updated
sucessfully",parent=self.root)


    def dat_Delete(self):
        dat_Delete = messagebox.askyesno("Hotel Management System,","Do you want to delete
this customer",parent=self.root)
```

```
        if dat_Delete>0:
            conn =
mysql.connector.connect(host="localhost",username="root",password="1234",database="hot
el")
            my_cursor = conn.cursor()
            query = "delete from customer where REF_NO=%s"
            value = (self.var_ref.get(),)
            my_cursor.execute(query,value)
        else:
            if not dat_Delete:
                return
        conn.commit()
        self.fetch_data()
        conn.close()


    def data_reset(self):
        #self.var_ref.set(""),
        self.var_cust_name.set(""),
        self.var_cust_father.set(""),
        #self.var_cust_gender.set(""),
        self.var_cust_post.set(""),
        self.var_cust_mobile.set(""),
```

```
        self.var_cust_email.set(""),
        #self.var_cust_nationality.set(""),
        #self.var_cust_id_proff.set(""),
        self.var_cust_id_number.set(""),
        self.var_cust_address.set("")

        x = random.randint(1000,9999)
        self.var_ref.set(str(x))

    def search_data(self):
        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database=
"hotel")
        my_cursor = conn.cursor()
        my_cursor.execute("select * from customer where "+str(self.search_var.get())+" LIKE
'%"+str(self.txt_search.get())+"%'")
        rows_featch = my_cursor.fetchall()
        if len(rows_featch)!=0:
            self.Cust_details_table.delete(*self.Cust_details_table.get_children())
            for i in rows_featch:
                self.Cust_details_table.insert("",END,values=i)
            conn.commit()
        conn.close()
```

```
if __name__ == "__main__":
    root=Tk()
    Obj=Cust_Win(root)
    root.mainloop()
```

- ***<u>Customer table code in MySQL: -</u>***

```
create database hotel;
use hotel;


CREATE TABLE `customer` (
  `REF_NO` int NOT NULL,
  `NAME` varchar(45) DEFAULT NULL,
  `FATHER` varchar(45) DEFAULT NULL,
  `GENDER` varchar(45) DEFAULT NULL,
  `POST` varchar(45) DEFAULT NULL,
```

```
  `MOBILE` varchar(45) DEFAULT NULL,
  `EMAIL` varchar(45) DEFAULT NULL,
  `NATIONALITY` varchar(45) DEFAULT NULL,
  `ID_PROFF` varchar(45) DEFAULT NULL,
  `ID_NO` varchar(45) DEFAULT NULL,
  `ADDRESS` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`REF_NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

- ***<u>Customer table: -</u>***

```
mysql> select * from customer;
+--------+----------+----------+--------+--------+------------+--------------+-------------+-----------------+-----------------+---------+
| REF_NO | NAME     | FATHER   | GENDER | POST   | MOBILE     | EMAIL        | NATIONALITY | ID_PROFF        | ID_NO           | ADDRESS |
+--------+----------+----------+--------+--------+------------+--------------+-------------+-----------------+-----------------+---------+
|   3442 | Mr. T.Das | Mr. K.Das | MALE   | 713301 | 9563742370 | sc@1gmail.com | INDIAN      | AADHAR CARD     | 123 456 789     | Asansol |
|   7789 | Mr. A.SEN | Mr. B.SEN | FEMALE | 713304 | 1234567891 | a@gmail.com   | AMERICAN    | DRIVING LICIENCE | 123456789012345 | Texas   |
|   9769 | U.Ghosh  | R.Ghosh  | MALE   | 330014 | 654323180  | u@gmail.com   | INDIAN      | AADHAR CARD     | 3456 789 123    | Kolkata |
+--------+----------+----------+--------+--------+------------+--------------+-------------+-----------------+-----------------+---------+
3 rows in set (0.00 sec)
```

- ***Room: -***

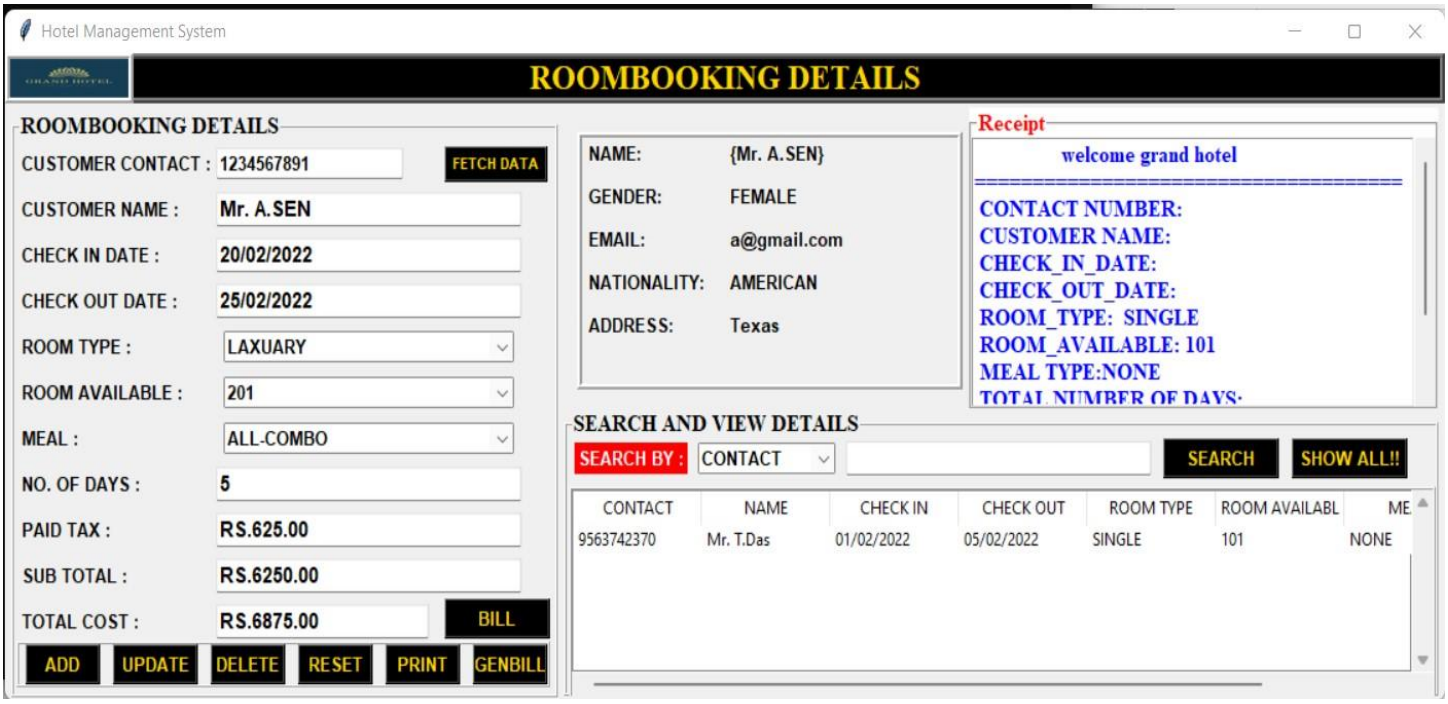
This window provides lots of features for the user like at first the user fetch data by using customer contact. Then he put the customer's name, check in and out dates, room types (Single, Duplex, Luxury), Available rooms, types of meal (Breakfast, Lunch, Dinner, All Combo).

And No. Of Days he lived, paid tax, subtotal, total cost is generated automatically by the pressing GENBILL button. Add, update, delete, reset buttons are used by like previous.

We also added one button is print button where the user can print the bill for the customer.

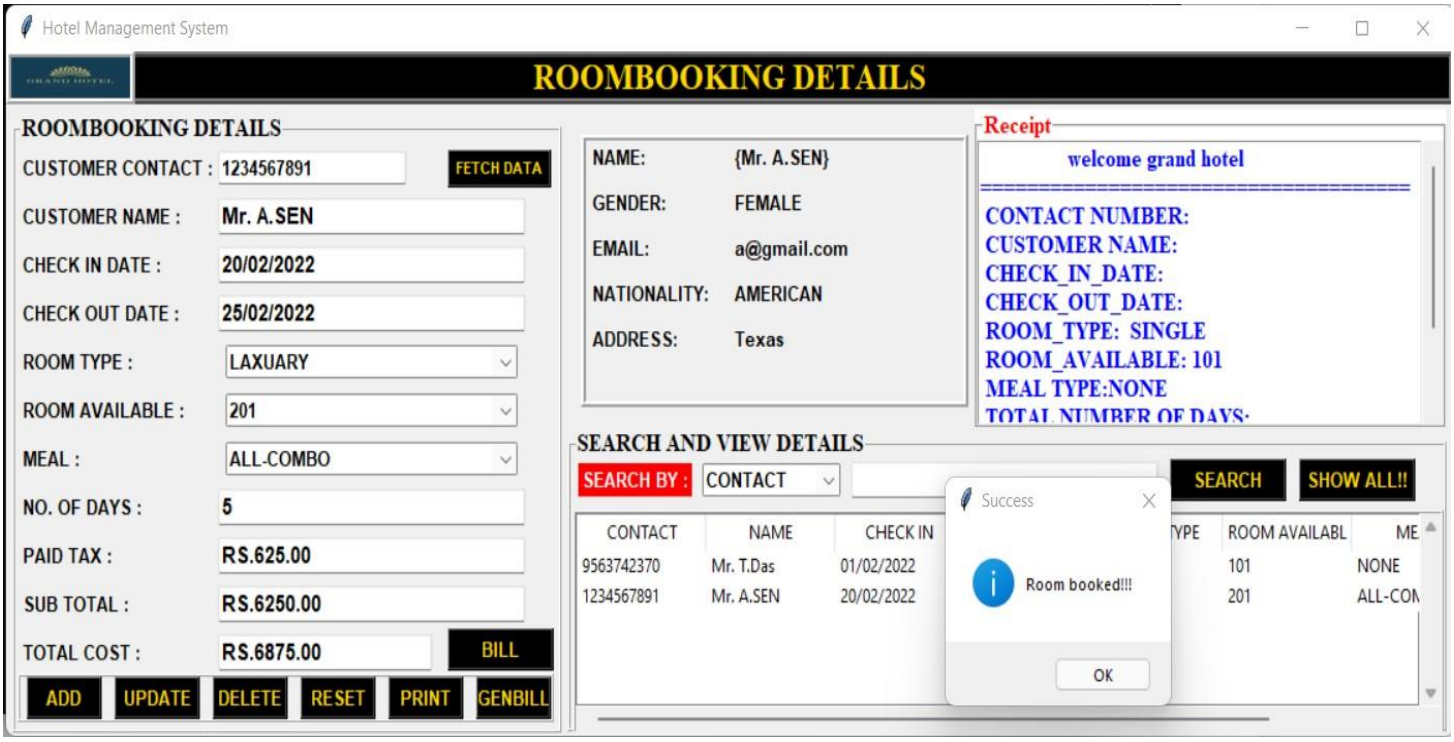
- ***Room window: -***

The user has entered some information here.



- ***After booking: -***

When the booking is confirmed one message pop up in the window **“Room Booked!!!".**



- ***After generating the bill: -***

```
          welcome grand hotel
=====================================
 CONTACT NUMBER:  1234567891
 CUSTOMER NAME:Mr. A.SEN
 CHECK_IN_DATE:20/02/2022
 CHECK_OUT_DATE:25/02/2022
 ROOM_TYPE:  LAXUARY
 ROOM_AVAILABLE: 201
 MEAL TYPE:ALL-COMBO
 TOTAL NUMBER OF DAYS:5
 PAID TAX:RS.625.00
 SUB TOTAL:RS.6250.00
 TOTAL COST:RS.6875.00
=====================================
```

- ***Python Code of Room window: -***

```
from os import stat

from tkinter import*

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

```
from tkinter import ttk
import random
from time import strftime
from datetime import datetime
import mysql.connector
from tkinter import messagebox
import tempfile
import os



class Roombooking:
    def __init__(self,root):
        self.root=root
        self.root.title("Hotel Management System")
        self.root.geometry("1121x452+234+243")



        ################ VARIABLES #############
        self.var_contact = StringVar()
        self.var_name = StringVar()
        self.var_check_in = StringVar()
        self.var_check_out = StringVar()
        self.var_room_type = StringVar()
```

```
        self.var_room_available = StringVar()

        self.var_meal = StringVar()

        self.var_no_of_days = StringVar()

        self.var_paid_tax = StringVar()

        self.var_actual_total = StringVar()

        self.var_total = StringVar()


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

        lbl_title = Label(self.root,text="ROOMBOOKING DETAILS",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)

        lbl_title.place(x=0,y=0,width=1121,height=35)


        ################## Logo ######################

        img2 = Image.open("logohotel.png")

        img2 = img2.resize((100,35),Image.ANTIALIAS)

        self.photoimg2=ImageTk.PhotoImage(img2)


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

        lblimg.place(x=0,y=0,width=100,height=35)


        ################ LABEL FRAME ##################

        lblFrameLeft=LabelFrame(self.root,bd=2,relief=RIDGE,text="ROOMBOOKING DETAILS",font=("times new roman",12,"bold"),padx=2)

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

```
        ################ LABELS & ENTRIES ##############



        ############### CUSTOMER CONTACT. #################

        self.lbl_cust_contact= Label(lblFrameLeft,text="CUSTOMER CONTACT
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        self.entry_contact = ttk.Entry(lblFrameLeft,textvariable=
self.var_contact,width=20,font=("arial",10,"bold"))

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



        ############## FEATCH DATA BUTTON ##########

        btn_featch_data = Button(lblFrameLeft,command=self.fetch_contact,text="FETCH
DATA",font=("arial",8,"bold"),bg="black",fg="gold",width=10)

        btn_featch_data.place(x=333,y=4)



        ############## CHECK IN DATE ################

        self.lblcheck_in_date = Label(lblFrameLeft,text="CHECK IN DATE
:",font=("arial",10,"bold"),padx=2,pady=6)

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



self.txtcheck_in_date=ttk.Entry(lblFrameLeft,textvariable=self.var_check_in,width=29,font=("arial",11,"bold"))

        self.txtcheck_in_date.grid(row=2,column=1)
```

```
        ############ CHECK OUT DATE ##############

        lblcheck_out_date = Label(lblFrameLeft,text="CHECK OUT DATE
:",font=("arial",10,"bold"),padx=2,pady=6)

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



txtcheck_out_date=ttk.Entry(lblFrameLeft,textvariable=self.var_check_out,width=29,font=("arial",11,"bold"))

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



        ############ ROOM TYPE ###############

        lblroom_type=Label(lblFrameLeft,text="ROOM TYPE :",font=("arial",10,"bold"),padx=2,pady=6)

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



        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select ROOM_TYPE from details")

        data_rows = my_cursor.fetchall()



        combo_room_type =
ttk.Combobox(lblFrameLeft,textvariable=self.var_room_type,font=("arial",10,"bold"),width=29,stat="readonly
")

        combo_room_type["value"]=("SINGLE","DUPLEX","LAXUARY")

        combo_room_type.current(0)

        combo_room_type.grid(row=4,column=1)
```

```
        ############ AVAILABLE ROOM ##########

        lblroom_available = Label(lblFrameLeft,text="ROOM AVAILABLE
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select ROOM_NO from details")

        rows = my_cursor.fetchall()



        combo_room_no =
ttk.Combobox(lblFrameLeft,textvariable=self.var_room_available,font=("arial",10,"bold"),width=29,stat="read
only")

        combo_room_no["value"]=rows

        combo_room_no.current(0)

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



        ############ MEAL ################

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

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



        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select MEAL from room")
```

```
data_rows = my_cursor.fetchall()


combo_meal=ttk.Combobox(lblFrameLeft,textvariable=self.var_meal,width=29,font=("arial",10,"bold"))

combo_meal["value"]=("NONE","Breakfast","Lunch","Dinner","ALL-COMBO")

combo_meal.current(0)

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


########### NUMBER OF DAYS ################

lblno_of_days = Label(lblFrameLeft,text="NO. OF DAYS :",font=("arial",10,"bold"),padx=2,pady=6)

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


txtno_of_days=ttk.Entry(lblFrameLeft,textvariable=self.var_no_of_days,width=29,font=("arial",11,"bold"))

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


########## PAID TAX ###############

lblno_of_days = Label(lblFrameLeft,text="PAID TAX :",font=("arial",10,"bold"),padx=2,pady=6)

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


txtno_of_days=ttk.Entry(lblFrameLeft,textvariable=self.var_paid_tax,width=29,font=("arial",11,"bold"))

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


########### SUB TOTAL ################

lblno_of_days = Label(lblFrameLeft,text="SUB TOTAL :",font=("arial",10,"bold"),padx=2,pady=6)
```

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



txtno_of_days=ttk.Entry(lblFrameLeft,textvariable=self.var_actual_total,width=29,font=("arial",11,"bold"))
        txtno_of_days.grid(row=9,column=1)

        ########## TOTAL COST ################
        lbLid_number = Label(lblFrameLeft,text="TOTAL COST :",font=("arial",10,"bold"),padx=2,pady=6)
        lbLid_number.grid(row=10,column=0,sticky=W)

        txtid_number = ttk.Entry(lblFrameLeft,textvariable=self.var_total,width=20,font=("arial",11,"bold"))
        txtid_number.grid(row=10,column=1,sticky=W)




        ############name######
        lbl_cust_name= Label(lblFrameLeft,text="CUSTOMER NAME :",font=("arial",10,"bold"),padx=2,pady=6)
        lbl_cust_name.grid(row=1,column=0,sticky=W)

        entry_name = ttk.Entry(lblFrameLeft,textvariable= self.var_name,width=29,font=("arial",11,"bold"))
        entry_name.grid(row=1,column=1,sticky=W)
```

```
        ############## BILL BUTTON ##########

        btn_bill_button =
Button(lblFrameLeft,text="BILL",command=self.total,font=("arial",10,"bold"),bg="black",fg="gold",width=9)

        btn_bill_button.place(x=333,y=320)



        ############## BUTTONS ##############

        btn_frame = Label(lblFrameLeft,bd=2,relief=RIDGE)

        btn_frame.place(x=0,y=350,width=415,height=32)



        btn_add =
Button(btn_frame,text="ADD",command=self.add_data,font=("arial",10,"bold"),bg="black",fg="gold",width=6
)

        btn_add.grid(row=0,column=0,padx=5)



        btn_update =
Button(btn_frame,text="UPDATE",command=self.update,font=("arial",10,"bold"),bg="black",fg="gold",width=
7)

        btn_update.grid(row=0,column=1,padx=5)



        btn_delete =
Button(btn_frame,text="DELETE",command=self.dat_Delete,font=("arial",10,"bold"),bg="black",fg="gold",wid
th=6)

        btn_delete.grid(row=0,column=2,padx=5)



        btn_reset =
Button(btn_frame,text="RESET",command=self.reset,font=("arial",10,"bold"),bg="black",fg="gold",width=6)
```

```
        btn_reset.grid(row=0,column=3,padx=5)



        btn_print =
Button(btn_frame,text="PRINT",command=self.print,font=("arial",10,"bold"),bg="black",fg="gold",width=6)

        btn_print.grid(row=0,column=4,padx=5)



        btn_genbill =
Button(btn_frame,text="GENBILL",command=self.gen_bill,font=("arial",10,"bold"),bg="black",fg="gold",width
=6)

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



        ############### RIGHT SIDE IMAGE #################

        lblFrameRight=LabelFrame(self.root,bd=2,text="Receipt",font=("times new
roman",12,"bold"),bg="white",fg="red")

        lblFrameRight.place(x=750,y=38,width=365,height=211)



        scroll_y=Scrollbar(lblFrameRight,orient=VERTICAL)

        self.textarea=Text(lblFrameRight,yscrollcommand=scroll_y.set,bg="white",fg="blue",font=("times new
roman",12,"bold"))

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

        scroll_y.config(command=self.textarea.yview)

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

        self.welcome()



        ############### TABLE FRAME SEARCH ################
```

```
table_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="SEARCH AND VIEW DETAILS",font=("times new roman",12,"bold"),padx=2)

table_frame.place(x=435,y=248,width=680,height=201)


lblsearchby = Label(table_frame,text="SEARCH BY :",font=("arial",10,"bold"),bg="red",fg="white")

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


self.search_var = StringVar()


combo_search=ttk.Combobox(table_frame,textvariable=self.search_var,font=("arial",10,"bold"),width=12,state="readonly")

combo_search["value"]=("CONTACT","ROOM")

combo_search.current(0)

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


self.txt_search = StringVar()

entry_search = ttk.Entry(table_frame,textvariable=self.txt_search,width=29,font=("arial",11,"bold"))

entry_search.grid(row=0,column=2,padx=4)


btn_search = Button(table_frame,text="SEARCH",command=self.search_data,font=("arial",10,"bold"),bg="black",fg="gold",width=10)

btn_search.grid(row=0,column=3,padx=5)
```

```
        btn_showall = Button(table_frame,text="SHOW
ALL!!",command=self.fetch_data,font=("arial",10,"bold"),bg="black",fg="gold",width=10)

        btn_showall.grid(row=0,column=4,padx=5)




        ################ SHOW DATA TABLE ##############

        details_table = Label(table_frame,bd=2,relief=RIDGE)

        details_table.place(x=0,y=34,width=674,height=148)


        Scroll_x=ttk.Scrollbar(details_table,orient=HORIZONTAL)

        Scroll_y=ttk.Scrollbar(details_table,orient=VERTICAL)



self.room_table=ttk.Treeview(details_table,columns=("CONTACT","NAME","CHECK_IN","CHECK_OUT","ROOM
_TYPE","ROOM_AVAILABLE","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("NAME",text="NAME")

        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("ROOM_AVAILABLE",text="ROOM AVAILABLE")

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=100)

self.room_table.column("NAME",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("ROOM_AVAILABLE",width=100)

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

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


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

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

self.fetch_data()
```

```
def welcome(self):
    self.textarea.delete(1.0,END)
    self.textarea.insert(END,"\t welcome grand hotel")
    self.textarea.insert(END,"\n====================================")
    self.textarea.insert(END,f"\n CONTACT NUMBER:  {self.var_contact.get()}")
    self.textarea.insert(END,f"\n CUSTOMER NAME:{self.var_name.get()}")
    self.textarea.insert(END,f"\n CHECK_IN_DATE:{self.var_check_in.get()}")
    self.textarea.insert(END,f"\n CHECK_OUT_DATE:{self.var_check_out.get()}")
    self.textarea.insert(END,f"\n ROOM_TYPE:  {self.var_room_type.get()}")
    self.textarea.insert(END,f"\n ROOM_AVAILABLE: {self.var_room_available.get()}")
    self.textarea.insert(END,f"\n MEAL TYPE:{self.var_meal.get()}")
    self.textarea.insert(END,f"\n TOTAL NUMBER OF DAYS:{self.var_no_of_days.get()}")
    self.textarea.insert(END,f"\n PAID TAX:{self.var_paid_tax.get()}")
    self.textarea.insert(END,f"\n SUB TOTAL:{ self.var_actual_total.get()}")
    self.textarea.insert(END,f"\n TOTAL COST:{self.var_total.get()}")

    self.textarea.insert(END,"\n====================================")


def add_data(self):
   if self.var_contact.get()=="" or self.var_check_in.get()="":
      messagebox.showerror("Error","ALL FIELDS ARE REQUIRED!!!",parent=self.root)
```

```
        else:
            try:
                conn = mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")
                my_cursor = conn.cursor()
                my_cursor.execute("insert into room values(%s,%s,%s,%s,%s,%s,%s,%s)",(self.var_contact.get(),self.var_name.get(),self.var_check_in.get(),self.var_check_out.get(),self.var_room_type.get(),self.var_room_available.get(),self.var_meal.get(),
        self.var_no_of_days.get()))
                conn.commit()
                self.fetch_data()
                conn.close()
                messagebox.showinfo("Success","Room booked!!!",parent=self.root)
            except Exception as es:
                messagebox.showwarning("Warning",f"Some thing went wrong:{str(es)}",parent=self.root)


    def gen_bill(self):
        self.welcome()


    def fetch_data(self):
        conn = mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")
        my_cursor = conn.cursor()
        my_cursor.execute("Select * from room")
        rows = my_cursor.fetchall()
```

```
    if len(rows)!= 0:
        self.room_table.delete(*self.room_table.get_children())
        for i in rows:
            self.room_table.insert("",END,values=i)
        conn.commit()
    conn.close()

def get_cursor(self,event=""):
    cursor_row = self.room_table.focus()
    content = self.room_table.item(cursor_row)
    row = content["values"]

    self.var_contact.set(row[0]),
    self.var_name.set(row[1])
    self.var_check_in.set(row[2]),
    self.var_check_out.set(row[3]),
    self.var_room_type.set(row[4]),
    self.var_room_available.set(row[5]),
    self.var_meal.set(row[6]),
    self.var_no_of_days.set(row[7])
```

```
    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="sc956374",database="hotel")

            my_cursor = conn.cursor()

            my_cursor.execute("update room set NAME=%s,
CHECK_IN=%s,CHECK_OUT=%s,ROOM_TYPE=%s,ROOM=%s,MEAL=%s,NO_OF_DAYS=%s where
CONTACT=%s",(self.var_name.get(),self.var_check_in.get(),self.var_check_out.get(),self.var_room_type.get(),s
elf.var_room_available.get(),self.var_meal.get(),self.var_no_of_days.get(),self.var_contact.get()))

            conn.commit()

            self.fetch_data()

            conn.close()

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



    def dat_Delete(self):

        dat_Delete = messagebox.askyesno("Hotel Management System,","Do you want to remove this
customers room",parent=self.root)

        if dat_Delete>0:

            conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

            my_cursor = conn.cursor()

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

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

            my_cursor.execute(query,value)
```

```
    else:
        if not dat_Delete:
            return
    conn.commit()
    self.fetch_data()
    conn.close()


def reset(self):
    self.var_contact.set(""),
    self.var_check_in.set(""),
    self.var_check_out.set(""),
    self.var_room_type.set(""),
    self.var_room_available.set(""),
    self.var_meal.set(""),
    self.var_no_of_days.set("")
    self.var_paid_tax.set("")
    self.var_actual_total.set("")
    self.var_total.set("")
    self.var_name.set("")



def print(self):
     q= self.textarea.get(1.0,"end-1c")
```

```
        filename=tempfile.mktemp('.txt')
        open(filename,'w').write(q)
        os.startfile(filename,"print")

    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="sc956374",database="hotel")
            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","Number not found",parent=self.root)
        else:
            conn.commit()
            conn.close()

            showdataframe=Frame(self.root,bd=4,relief=RIDGE,padx=2)
            showdataframe.place(x=445,y=55,width=300,height=180)
```

```
        ######## NAME ###############

        lblname=Label(showdataframe,text="NAME:",font=("arial",10,"bold"))

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


        lbl_data_name = Label(showdataframe,text=row,font=("arial",10,"bold"))

        lbl_data_name.place(x=110,y=0)


        ############ GENDER ############

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        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",10,"bold"))

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


        lbl_data_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

        lbl_data_gender.place(x=110,y=30)


        ############# EMAIL #################
```

```
        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        my_cursor = conn.cursor()

        query=("select EMAIL from customer where MOBILE=%s")

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

        my_cursor.execute(query,value)

        row = my_cursor.fetchone()


        lblgender=Label(showdataframe,text="EMAIL:",font=("arial",10,"bold"))

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


        lbl_data_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

        lbl_data_gender.place(x=110,y=60)


        ######## NATIONALITY #################

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        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()


        lblgender=Label(showdataframe,text="NATIONALITY:",font=("arial",10,"bold"))
```

```
        lblgender.place(x=0,y=90)


        lbl_data_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

        lbl_data_gender.place(x=110,y=90)


        ######### ADDRESS ###############

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        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()


        lblgender=Label(showdataframe,text="ADDRESS:",font=("arial",10,"bold"))

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


        lbl_data_gender = Label(showdataframe,text=row,font=("arial",10,"bold"))

        lbl_data_gender.place(x=110,y=120)



    ############## SEARCH ####################

    def search_data(self):
```

```
        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("select * from room where "+str(self.search_var.get())+" LIKE
'%"+str(self.txt_search.get())+"%'")

        rows_featch = my_cursor.fetchall()

        if len(rows_featch)!=0:

            self.room_table.delete(*self.room_table.get_children())

            for i in rows_featch:

                self.room_table.insert("",END,values=i)

            conn.commit()

        conn.close()




    def total(self):

        indate = self.var_check_in.get()

        outdate = self.var_check_out.get()

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

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

        self.var_no_of_days.set(abs(outdate-indate).days)



        if (self.var_meal.get()=="Breakfast" and self.var_room_type.get()=="LAXUARY"):

            q1 = float(300)

            q2 = float(700)
```

```
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Lunch" and self.var_room_type.get()=="LAXUARY"):
    q1 = float(300)
    q2 = float(750)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)
```

```
elif (self.var_meal.get()=="Dinner" and self.var_room_type.get()=="LAXUARY"):
    q1 = float(300)
    q2 = float(780)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)



elif (self.var_meal.get()=="NONE" and self.var_room_type.get()=="LAXUARY"):
    q1 = float(300)
    q2 = float(650)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
```

```
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="ALL-COMBO" and self.var_room_type.get()=="LAXUARY"):
    q1 = float(400)
    q2 = float(850)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Breakfast" and self.var_room_type.get()=="SINGLE"):
    q1 = float(100)
    q2 = float(300)
    q3 = float(self.var_no_of_days.get())
```

```
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Lunch" and self.var_room_type.get()=="SINGLE"):
    q1 = float(100)
    q2 = float(320)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Dinner" and self.var_room_type.get()=="SINGLE"):
```

```
    q1 = float(100)
    q2 = float(350)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="NONE" and self.var_room_type.get()=="SINGLE"):
    q1 = float(100)
    q2 = float(230)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
```

```
    self.var_total.set(TT)


elif (self.var_meal.get()=="ALL-COMBO" and self.var_room_type.get()=="SINGLE"):
    q1 = float(300)
    q2 = float(450)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)


elif (self.var_meal.get()=="Breakfast" and self.var_room_type.get()=="DUPLEX"):
    q1 = float(200)
    q2 = float(400)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
```

```
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Lunch" and self.var_room_type.get()=="DUPLEX"):
    q1 = float(200)
    q2 = float(410)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)

elif (self.var_meal.get()=="Dinner" and self.var_room_type.get()=="DUPLEX"):
    q1 = float(200)
    q2 = float(420)
    q3 = float(self.var_no_of_days.get())
```

```
    q4 = float(q1+q2)

    q5 = float(q3*q4)

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

    ST = "RS."+str("%.2f"%((q5)))

    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))

    self.var_paid_tax.set(Tax)

    self.var_actual_total.set(ST)

    self.var_total.set(TT)


elif (self.var_meal.get()=="NONE" and self.var_room_type.get()=="DUPLEX"):

    q1 = float(200)

    q2 = float(320)

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

    q4 = float(q1+q2)

    q5 = float(q3*q4)

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

    ST = "RS."+str("%.2f"%((q5)))

    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))

    self.var_paid_tax.set(Tax)

    self.var_actual_total.set(ST)

    self.var_total.set(TT)


elif (self.var_meal.get()=="ALL-COMBO" and self.var_room_type.get()=="DUPLEX"):
```

```
    q1 = float(400)
    q2 = float(350)
    q3 = float(self.var_no_of_days.get())
    q4 = float(q1+q2)
    q5 = float(q3*q4)
    Tax = "RS."+str("%.2f"%((q5)*0.1))
    ST = "RS."+str("%.2f"%((q5)))
    TT = "RS."+str("%.2f"%(q5+((q5)*0.1)))
    self.var_paid_tax.set(Tax)
    self.var_actual_total.set(ST)
    self.var_total.set(TT)
def welcome(self):
  self.textarea.delete(1.0,END)
  self.textarea.insert(END,"\t welcome grand hotel")
  self.textarea.insert(END,f"\n contact number:{self.var_contact.get()}")
  self.textarea.insert(END,f"\n check_IN_data:{self.var_check_in()}")
  self.textarea.insert(END,f"\n check_out_date:{self.var_check_out.get()}")
  self.textarea.insert(END,f"\n room_Type:{self.var_room_type.get()}")
  self.textarea.insert(END,f"\n room_ava:{self.var_room_available.get()}")
  self.textarea.insert(END,f"\n meal:{self.var_meal.get()}")
  self.textarea.insert(END,f"\n no_of_days:{self.var_no_of_days.get()}")

  self.textarea.insert(END,"\n==========================================")
```

```
if __name__ == "__main__":
    root=Tk()
    Obj=Roombooking(root)
    root.mainloop()
```

- ***<u>Room table code in MySQL: -</u>***

```
CREATE TABLE `room` (
  `CONTACT` varchar(45) DEFAULT NULL,
  `NAME` varchar(45) DEFAULT NULL,
  `CHECK_IN` varchar(45) DEFAULT NULL,
  `CHECK_OUT` varchar(45) DEFAULT NULL,
  `ROOM_TYPE` varchar(45) DEFAULT NULL,
  `ROOM` varchar(45) NOT NULL,
  `MEAL` varchar(45) DEFAULT NULL,
  `NO_OF_DAYS` varchar(45) DEFAULT NULL,
```

PRIMARY KEY (`ROOM`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

- ***Room table: -***

```
mysql> select * from room;
+------------+-----------+------------+------------+-----------+------+-----------+------------+
| CONTACT    | NAME      | CHECK_IN   | CHECK_OUT  | ROOM_TYPE | ROOM | MEAL      | NO_OF_DAYS |
+------------+-----------+------------+------------+-----------+------+-----------+------------+
| 9563742370 | Mr. T.Das | 01/02/2022 | 05/02/2022 | SINGLE    | 101  | Lunch     | 4          |
| 1234567891 | Mr. A.SEN | 20/02/2022 | 25/02/2022 | LAXUARY   | 201  | ALL-COMBO | 5          |
+------------+-----------+------------+------------+-----------+------+-----------+------------+
2 rows in set (0.00 sec)
```

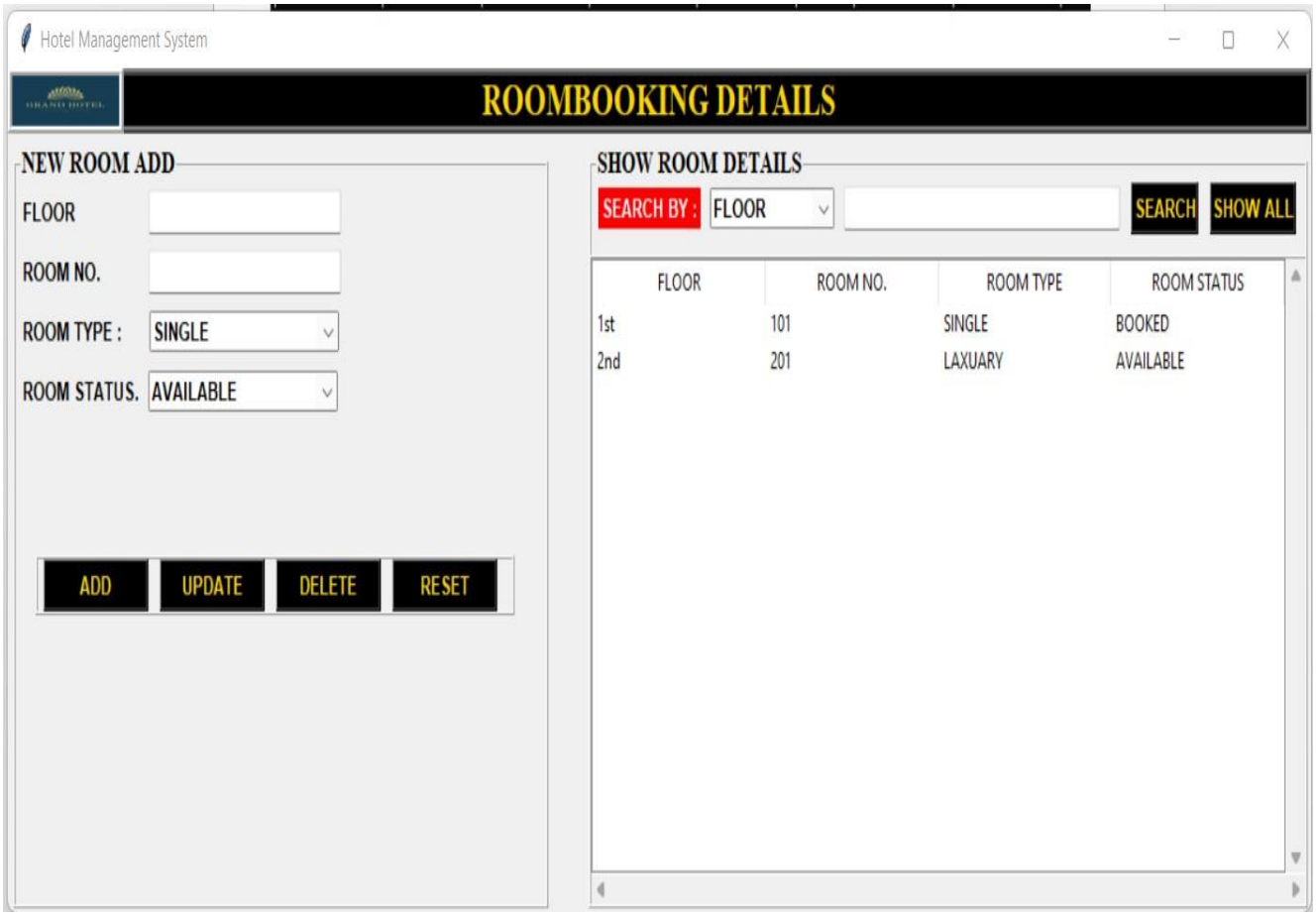
- ***Details options: -***

Here the user can create floor, room no, types of room like single, duplex, luxury

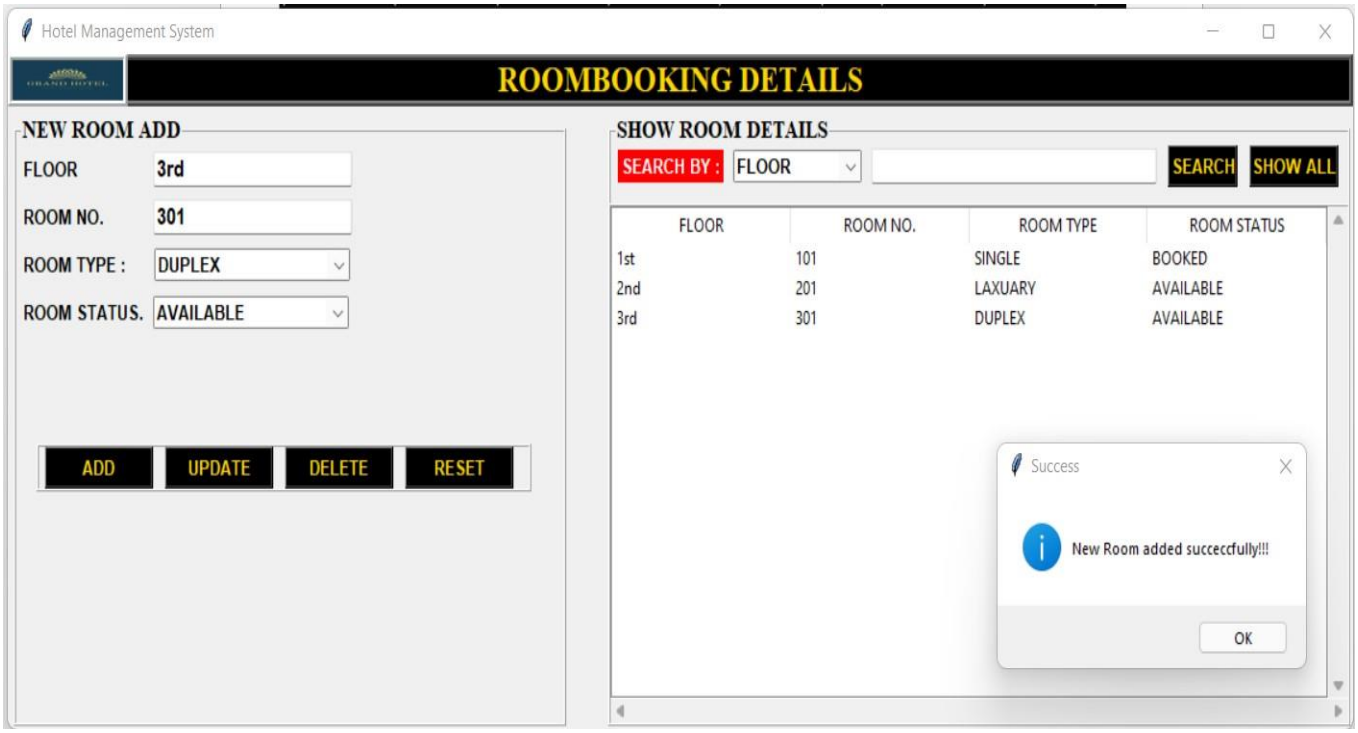
Also, we can change the status of room is available or booked. And add, update,

Reset buttons are also use here.

- ***Details window: -***



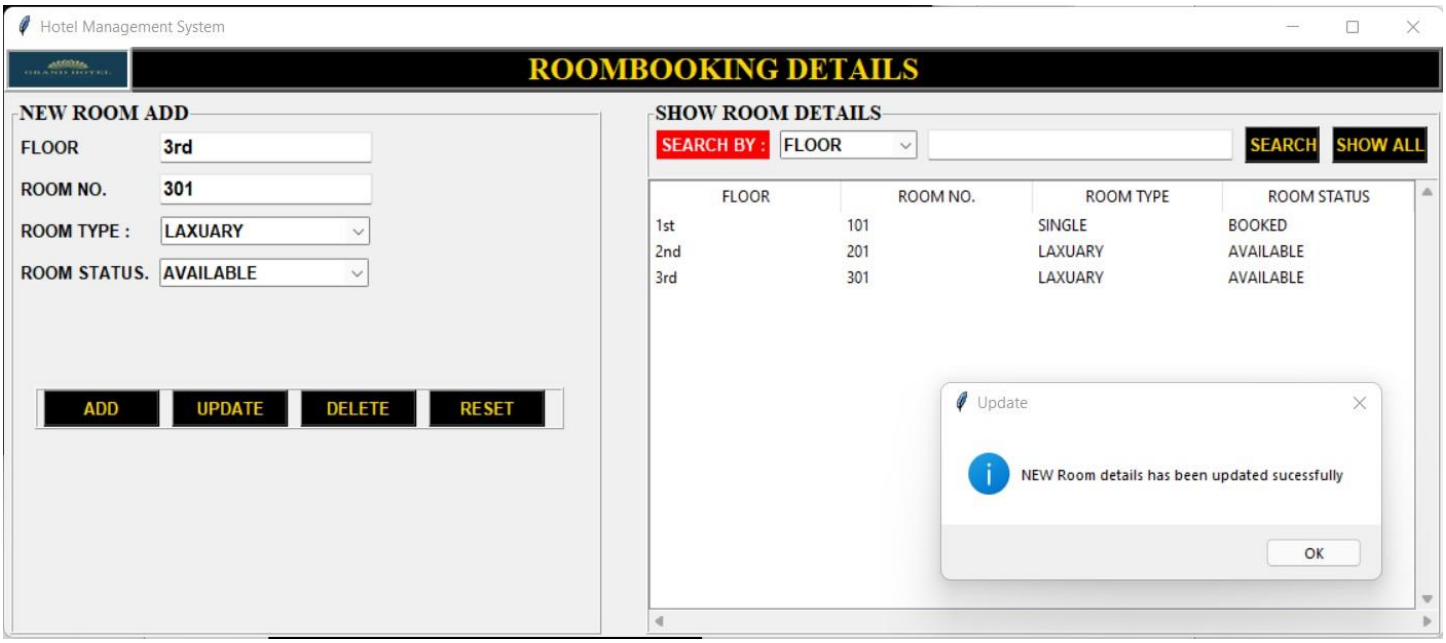
- ***<u>Add button use in Details window: -</u>***
- When admin add floor, room no, room types, room status then he use add button.



After added the message will be show **“New Room added successfully.”**

- ***<u>Update button use in Details window: -</u>***

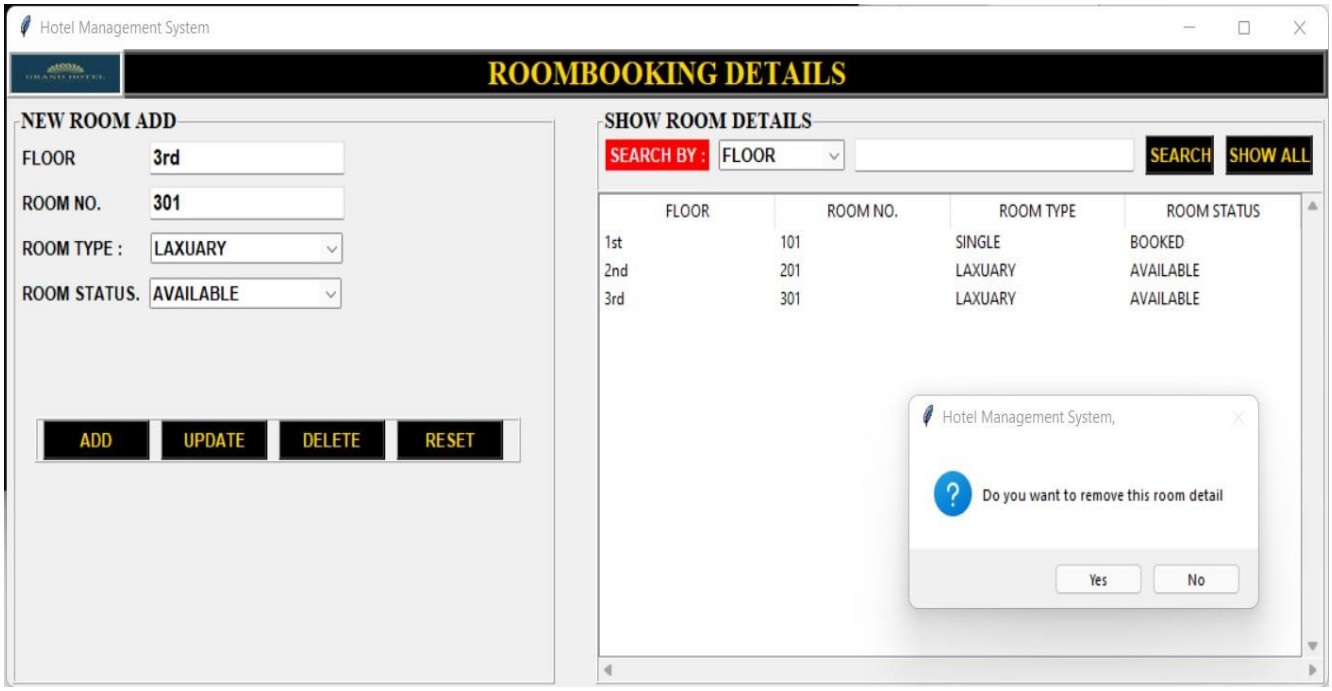
  When the user wants to update something.



After updating the message will be show**" New room details has been updated successfully".**

- ***<u>Delete button use in Details window: -</u>***

When we want to delete some data's then we use delete button.

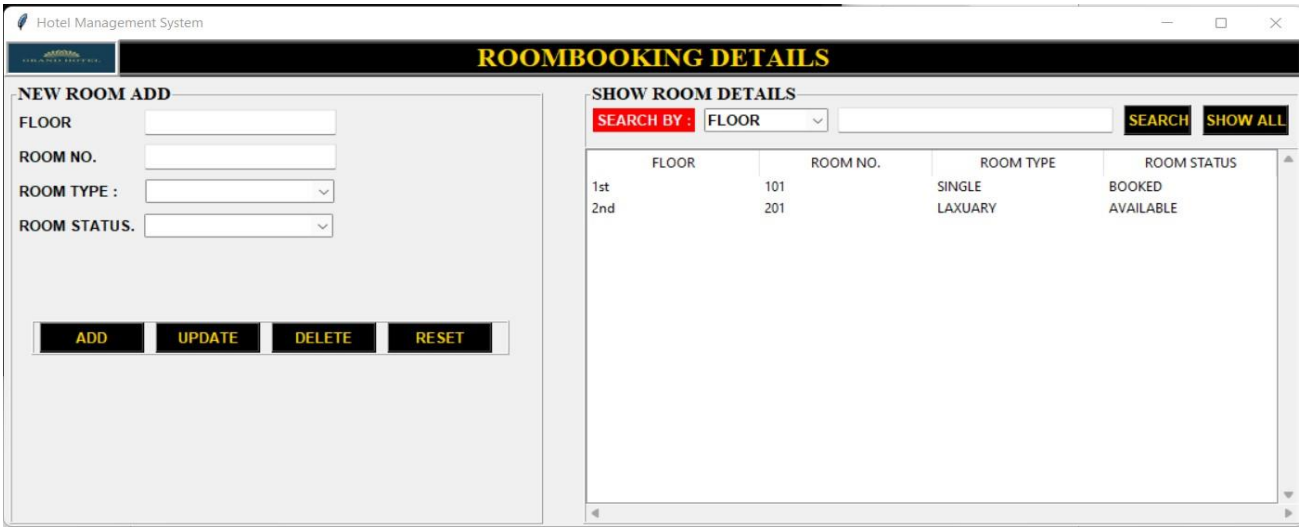


And the user pressed the delete button then the system asked the user **"Do you want to remove the data?"**

And you press yes then the data is deleted or press no then the data is not deleted.

- ***Reset button use in Details window: -***

When the user wants to reset the details window then just pressed reset button.



after reset the text area is cleaned.

- ***Python code of Details window: -***

```
from os import stat

from tkinter import*

from PIL import Image, ImageTk #pip install pillow

from tkinter import ttk

import random

from time import strftime

from datetime import datetime
```

```
import mysql.connector
from tkinter import messagebox


class Detailsroom:
    def __init__(self,root):
        self.root=root
        self.root.title("Hotel Management System")
        self.root.geometry("1121x452+234+243")

        ################## Title ###################
        lbl_title = Label(self.root,text="ROOMBOOKING DETAILS",font=("times new roman",18,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
        lbl_title.place(x=0,y=0,width=1121,height=35)

        ################### Logo ########################
        img2 = Image.open("logohotel.png")
        img2 = img2.resize((100,35),Image.ANTIALIAS)
        self.photoimg2=ImageTk.PhotoImage(img2)

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

```
        lblimg.place(x=0,y=0,width=100,height=35)


        ################ LABEL FRAME ###################

        lblFrameLeft=LabelFrame(self.root,bd=2,relief=RIDGE,text="NEW ROOM
ADD",font=("times new roman",12,"bold"),padx=2)

        lblFrameLeft.place(x=5,y=40,width=460,height=410)


        ############### FLOOR #################

        lbl_floor= Label(lblFrameLeft,text="FLOOR
",font=("arial",10,"bold"),padx=2,pady=6)

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


        self.var_floor=StringVar()

        entry_floor =
ttk.Entry(lblFrameLeft,textvariable=self.var_floor,width=20,font=("arial",11,"bold"
))

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


        ############## ROOM NUMBER #################

        lbl_room_no= Label(lblFrameLeft,text="ROOM NO.
",font=("arial",10,"bold"),padx=2,pady=6)

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

```
        self.var_room_no=StringVar()

        entry_room_no =
ttk.Entry(lblFrameLeft,textvariable=self.var_room_no,width=20,font=("arial",11,"b
old"))

        entry_room_no.grid(row=1,column=1,sticky=W)



        ############### ROOM TYPE ##################

        lblroom_type=Label(lblFrameLeft,text="ROOM TYPE
:",font=("arial",10,"bold"),padx=2,pady=6)

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



        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select ROOM_TYPE from details")

        data_rows = my_cursor.fetchall()



        self.var_room_type=StringVar()
```

```
    combo_room_type =
ttk.Combobox(lblFrameLeft,textvariable=self.var_room_type,font=("arial",10,"bol
d"),width=20,stat="readonly")

    combo_room_type["value"]=("SINGLE","DUPLEX","LAXUARY")

    combo_room_type.current(0)

    combo_room_type.grid(row=2,column=1)




    ############### ROOM Status ##################

    lbl_room_status= Label(lblFrameLeft,text="ROOM STATUS.
",font=("arial",10,"bold"),padx=2,pady=6)

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



    conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")

    my_cursor = conn.cursor()

    my_cursor.execute("Select ROOM_STATUS from details")

    data_rows = my_cursor.fetchall()



    self.var_room_status=StringVar()
```

```
        combo_room_status =
ttk.Combobox(lblFrameLeft,textvariable=self.var_room_status,font=("arial",10,"bo
ld"),width=20,stat="readonly")

        combo_room_status["value"]=("AVAILABLE","BOOKED")

        combo_room_status.current(0)

        combo_room_status.grid(row=3,column=1,sticky=W)



        ############### BUTTONS ###############

        btn_frame = Label(lblFrameLeft,bd=2,relief=RIDGE)

        btn_frame.place(x=15,y=200,width=412,height=32)



        btn_add =
Button(btn_frame,text="ADD",command=self.add_data,font=("arial",10,"bold"),bg
="black",fg="gold",width=10)

        btn_add.grid(row=0,column=0,padx=5)



        btn_update =
Button(btn_frame,text="UPDATE",command=self.update,font=("arial",10,"bold"),
bg="black",fg="gold",width=10)

        btn_update.grid(row=0,column=1,padx=5)
```

```
        btn_delete =
Button(btn_frame,text="DELETE",command=self.dat_Delete,font=("arial",10,"bold
"),bg="black",fg="gold",width=10)

        btn_delete.grid(row=0,column=2,padx=5)



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

        btn_reset.grid(row=0,column=3,padx=5)



        ############### TABLE FRAME SEARCH ################

        table_frame=LabelFrame(self.root,bd=2,relief=RIDGE,text="SHOW ROOM
DETAILS",font=("times new roman",12,"bold"),padx=2)

        table_frame.place(x=500,y=40,width=618,height=350)







        lblsearchby = Label(table_frame,text="SEARCH BY
:",font=("arial",10,"bold"),bg="red",fg="white")

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



        self.search_var = StringVar()
```

```
combo_search=ttk.Combobox(table_frame,textvariable=self.search_var,font=("ari
al",10,"bold"),width=12,state="readonly")

    combo_search["value"]=("FLOOR","ROOM_TYPE","ROOM_STATUS")

    combo_search.current(0)

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



    self.txt_search = StringVar()

    entry_search =
ttk.Entry(table_frame,textvariable=self.txt_search,width=29,font=("arial",11,"bold
"))

    entry_search.grid(row=0,column=2,padx=4)



    btn_search =
Button(table_frame,text="SEARCH",command=self.search_data,font=("arial",10,"b
old"),bg="black",fg="gold",width=6)

    btn_search.grid(row=0,column=3,padx=5)



    btn_showall = Button(table_frame,text="SHOW
ALL",command=self.fetch_data,font=("arial",10,"bold"),bg="black",fg="gold",widt
h=8)

    btn_showall.grid(row=0,column=4,padx=5)
```

```
 ############### TABLE SEARCH ################

table_frame=LabelFrame(self.root,bd=2,relief=RIDGE)

table_frame.place(x=500,y=100,width=618,height=350)


Scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)

Scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)


self.room_table=ttk.Treeview(table_frame,columns=("FLOOR","ROOM_NO","ROOM_TYPE","ROOM_STATUS"),xscrollcommand=Scroll_x.set,yscrollcommand=Scroll_y.set)

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

Scroll_y.pack(side=RIGHT,fill=Y)


Scroll_x.config(command=self.room_table.xview)

Scroll_y.config(command=self.room_table.yview)


self.room_table.heading("FLOOR",text="FLOOR")
```

```
    self.room_table.heading("ROOM_NO",text="ROOM NO.")
    self.room_table.heading("ROOM_TYPE",text="ROOM TYPE")
    self.room_table.heading("ROOM_STATUS",text="ROOM STATUS")

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

    self.room_table.column("FLOOR",width=100)
    self.room_table.column("ROOM_NO",width=100)
    self.room_table.column("ROOM_TYPE",width=100)
    self.room_table.column("ROOM_STATUS",width=100)

    self.room_table.pack(fill=BOTH,expand=1)
    self.room_table.bind("<ButtonRelease-1>",self.get_cursor)
    self.fetch_data()



def search_data(self):
```

```
        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("select * from details where "+str(self.search_var.get())+"
LIKE '%"+str(self.txt_search.get())+"%'")

        rows_featch = my_cursor.fetchall()

        if len(rows_featch)!=0:

            self.room_table.delete(*self.room_table.get_children())

            for i in rows_featch:

                self.room_table.insert("",END,values=i)

            conn.commit()

        conn.close()



    def add_data(self):

        if self.var_floor.get()=="" or self.var_room_type.get()=="":

            messagebox.showerror("Error","ALL FIELDS ARE
REQUIRED!!!",parent=self.root)

        else:

            try:
```

```
            conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")

            my_cursor = conn.cursor()

            my_cursor.execute("insert into details
values(%s,%s,%s,%s)",(self.var_floor.get(),self.var_room_no.get(),self.var_room_ty
pe.get(),self.var_room_status.get()))

            conn.commit()

            self.fetch_data()

            conn.close()

            messagebox.showinfo("Success","New Room added
succeccfully!!!",parent=self.root)

        except Exception as es:

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



    def fetch_data(self):

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("Select * from details")

        rows = my_cursor.fetchall()
```

```
    if len(rows)!= 0:
        self.room_table.delete(*self.room_table.get_children())
        for i in rows:
            self.room_table.insert("",END,values=i)
        conn.commit()
    conn.close()

def get_cursor(self,event=""):
    cursor_row = self.room_table.focus()
    content = self.room_table.item(cursor_row)
    row = content["values"]

    self.var_floor.set(row[0]),
    self.var_room_no.set(row[1]),
    self.var_room_type.set(row[2])
    self.var_room_status.set(row[3])

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="root",password="sc956374
",database="hotel")

        my_cursor = conn.cursor()

        my_cursor.execute("update details set
FLOOR=%s,ROOM_TYPE=%s,ROOM_STATUS=%s where
ROOM_NO=%s",(self.var_floor.get(),self.var_room_type.get(),self.var_room_statu
s.get(),self.var_room_no.get(),))

        conn.commit()

        self.fetch_data()

        conn.close()

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



  def dat_Delete(self):

    dat_Delete = messagebox.askyesno("Hotel Management System,","Do you
want to remove this room detail",parent=self.root)

    if dat_Delete>0:

        conn =
mysql.connector.connect(host="localhost",username="root",password="sc956374
",database="hotel")
```

```
            my_cursor = conn.cursor()
            query = "delete from details where ROOM_NO=%s"
            value = (self.var_room_no.get(),)
            my_cursor.execute(query,value)
        else:
            if not dat_Delete:
                return
        conn.commit()
        self.fetch_data()
        conn.close()

    def reset(self):
        self.var_floor.set(""),
        self.var_room_no.set(""),
        self.var_room_type.set("")
        self.var_room_status.set("")


if __name__ == "__main__":
    root=Tk()
```

```
Obj=Detailsroom(root)

root.mainloop()
```

- ***MySQL code of details table: -***

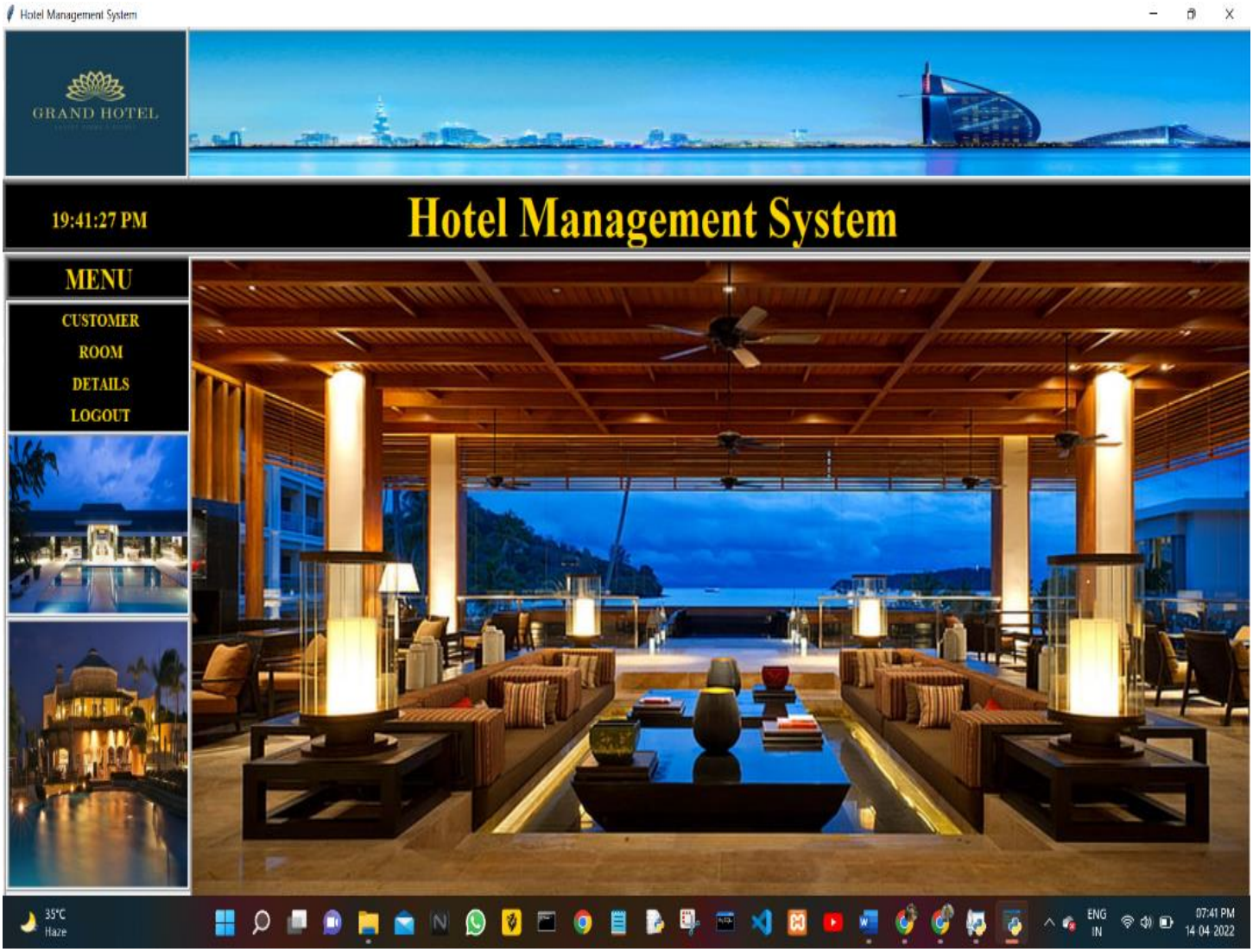
```
CREATE TABLE `details` (
  `FLOOR` varchar(45) DEFAULT NULL,
  `ROOM_NO` varchar(45) NOT NULL,
  `ROOM_TYPE` varchar(45) DEFAULT NULL,
  `ROOM_STATUS` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`ROOM_NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

- ***MYSQL details table:*** -

```
mysql> select * from details;
+-------+---------+-----------+--------------+
| FLOOR | ROOM_NO | ROOM_TYPE | ROOM_STATUS |
+-------+---------+-----------+--------------+
| 1st   | 101     | SINGLE    | BOOKED       |
| 2nd   | 201     | LAXUARY   | AVAILABLE    |
+-------+---------+-----------+--------------+
2 rows in set (0.00 sec)
```

- ***<u>Logout option: -</u>***

When the user wants to exit from the system just pressed logout button



Hotel Management system Design Code:-

```
from tkinter import*
from PIL import Image,ImageTk  #pip install pillow
from customer import Cust_Win, Cust_Win
from room import Roombooking
from details import Detailsroom
from time import strftime


class HotelManagementSystem:
    def __init__(self, root):
        self.root = root
        self.root.title("Hotel Management System")
        self.root.geometry("1600x700+0+0")

        ###################### First image ################
        img1 = Image.open("hotel1.jpg")
        img1 = img1.resize((1600,140),Image.ANTIALIAS)
        self.photoimg1=ImageTk.PhotoImage(img1)
```

```
        lblimg =
Label(self.root,image=self.photoimg1,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=1600,height=140)



        ###################### Logo ########################
        img2 = Image.open("logohotel.png")
        img2 = img2.resize((230,140),Image.ANTIALIAS)
        self.photoimg2=ImageTk.PhotoImage(img2)

        lblimg =
Label(self.root,image=self.photoimg2,bd=4,relief=RIDGE)
        lblimg.place(x=0,y=0,width=230,height=140)



        ################### Title ####################
        lbl_title = Label(self.root,text="Hotel Management
System",font=("times new
roman",40,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE
)
        lbl_title.place(x=0,y=140,width=1600,height=65)
```

```
        #time
        def time():
            string = strftime('%H:%M:%S %p')
            lbl.config(text = string)
            lbl.after(1000, time)

        lbl = Label(lbl_title, font = ('times new
roman',16,'bold'),background = 'black' ,foreground = 'gold')
        lbl.place(x=0,y=10,width=230,height=45)
        time()

Main_Frame=Frame(self.root,bd=5,relief=GROOVE,bg="blac
k")

        ################## MAIN FRAME #################
        main_frame = Frame(self.root,bd=4,relief=RIDGE)
        main_frame.place(x=0,y=205,width=1600,height=620)
```

```
        ################ LABEL #################
        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=229,height=190)

        cust_btn =
Button(btn_frame,text="CUSTOMER",command=self.Cust_d
etails,width=22,font=("times new
roman",13,"bold"),bg="black",fg="gold",bd=0,cursor="hand
1")
        cust_btn.grid(row=0,column=0)

        room_btn =
Button(btn_frame,text="ROOM",command=self.roombooking
,width=22,font=("times new
```

```
roman",13,"bold"),bg="black",fg="gold",bd=0,cursor="hand
1")
        room_btn.grid(row=1,column=0)


        details_btn =
Button(btn_frame,text="DETAILS",command=self.Details_ro
om,width=22,font=("times new
roman",13,"bold"),bg="black",fg="gold",bd=0,cursor="hand
1")
        details_btn.grid(row=2,column=0)


        logout_btn =
Button(btn_frame,text="LOGOUT",command=self.logout,wid
th=22,font=("times new
roman",13,"bold"),bg="black",fg="gold",bd=0,cursor="hand
1")
        logout_btn.grid(row=4,column=0)



        ################## RIGHT SIDE IMAGE
######################
        img3 = Image.open("slide3.jpg")
```

```
        img3 = img3.resize((1310,590),Image.ANTIALIAS)
        self.photoimg3=ImageTk.PhotoImage(img3)

        lblimg1 =
Label(main_frame,image=self.photoimg3,bd=4,relief=RIDGE)
        lblimg1.place(x=225,y=0,width=1310,height=590)

        ################## DOWN IMAGES
######################
        img4 = Image.open("sideimg.jpg")
        img4 = img4.resize((230,210),Image.ANTIALIAS)
        self.photoimg4=ImageTk.PhotoImage(img4)

        lblimg1 =
Label(main_frame,image=self.photoimg4,bd=4,relief=RIDGE)
        lblimg1.place(x=0,y=160,width=228,height=170)

        img5 = Image.open("inside.jpg")
        img5 = img5.resize((230,250),Image.ANTIALIAS)
        self.photoimg5=ImageTk.PhotoImage(img5)
```

```
        lblimg1 =
Label(main_frame,image=self.photoimg5,bd=4,relief=RIDGE)
        lblimg1.place(x=0,y=330,width=228,height=250)



    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()
```

7. ***Advantages of the proposed system: -***

i. Save time on admin tasks

ii. Develop strong relationships with your guests

iii. Implement an effective revenue management system

iv. Manage distribution functions

v. Increase bookings

vi. Accurate daily reports

vii. Prevent double bookings and manual errors

viii. Analyze your customer base

8. Limitation of the System:-

1. Customers can't able to book the hotel in online mode through this system.

2. Admin can't able to book a room for a guest through mobile because this system only support windows 10 or higher versions.

9. Future scope:-

1. Making an online hotel booking website.

2. Making an android app for the system where the customer can book hotel by himself.

3. Enable online payment system in our site/app so the customer can pay the bill via online mode.

10. Bibliography & References: -

WEBSITE:- **https://www.geeksforgeeks.org/**

WEBSITE:- **https://www.w3schools.com/python/**

WEBSITE:- **https://stackoverflow.com/**

YOUTUBE:- **https://www.youtube.com/channel/UCP1HhCg0ZwOyLFssftBCDIA**

BOOKS:- **Python: The Complete Reference**