



THE CONTINENTAL HOTEL MANAGEMENT

Python Project



POWERED BY
MySQL™

Submitted to
Mrs Rashmi Singhai

Submitted by
Krishn Sharma

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my Computer Science teacher Rashmi Singhai for her able guidance , support and encouragement in completing my project.

I would also like to extend my gratitude to the principal madam Nandini for providing me with all the facilities required for the completion of this project.

DATE

NAME & CLASS

CERTIFICATE

This Certificate is Presented to

Krishn Sharma

This is to certify that Krishn sharma of class XII-O
has successfully completed the Computer
Science project under the guidance of Rashmi
Singhai during the academic session 2024-25
as per the guidelines issued by central board of
secondary education.

PRINCIPAL
NANDINI

TEACHER
RASHMI SINGHAI

INDEX

Introduction	05	Need Of Automation	07
Requirements	08	Advantages	09
Usage Documentary	10	Source Code	20
Bibliography	34		



INTRODUCTION: BRIEF OVERVIEW OF THE PROJECT



This project is based on Hotel Management. The program helps us to enter, display, or modify details of guests and room bookings in hotel records. Moreover, and most importantly, the program assists in tracking the current status of a room, guest details, and other related information. It includes various functional modules to accomplish these tasks.

Data file handling is effectively implemented within the program. The database is a collection of interrelated data to serve multiple applications. Database programs create files of information, making file management a critical part of the software.



DBMS

The software required for managing data is called DBMS (Database Management System). It has three models:

01 Relational Model

This is based on the concept of relations. A relation is essentially a table consisting of rows and columns. The rows of the table are referred to as tuples, while the columns are known as attributes. The total number of rows in a table is called cardinality, and the number of columns is referred to as the degree.

02 Hierarchical Model

In this model, each record has one parent record, and no child record can exist without a parent record. The records are organized in a tree-like structure, making it suitable for applications that require a parent-child relationship.

03 Network Model

Here, data is represented as a collection of records, and the relationships between data are represented through links or associations.



Need for Automation

Planning a stay at a hotel has never been easier, thanks to the efficiency and accessibility brought by modern hotel management systems. With computerized systems, reservations can be made through the hotel's website or via authorized travel agencies, making the entire process quick and hassle-free. Many hotels also offer mobile apps that allow users to book rooms, check availability, and manage their bookings from anywhere, without waiting in long queues or making phone calls.

This shift to automation in hotel management has revolutionized the hospitality industry. It reduces manual errors, enhances guest experience, and optimizes hotel operations. From inventory management to guest preferences, everything is efficiently handled through an integrated system. As a result, hotel staff can focus more on guest service rather than routine administrative tasks. This advancement is widely accepted and appreciated, offering both convenience to guests and operational efficiency for hotels.



System Requirements



SOFTWARE REQUIREMENTS

WINDOWS **11**

PYTHON **3.13.0**

MYSQL SERVER **8.0**

PYTHON => <https://www.python.org/downloads/>

MYSQL SERVER => <https://dev.mysql.com/downloads/>

HARDWARE REQUIREMENTS

INTEL i5 8100U @ 2.5Ghz

RAM: 8GB DDR4

STORAGE: 1 TB SSD



Advantages

Efficient Booking

Real-time
Availability

Centralized
Data

Increased
Revenue

Streamlined Booking Process:

Automates room reservations, making it easier for both staff and guests to handle bookings efficiently.

Real-time Room Availability:

Provides up-to-date information on room availability, helping to avoid overbooking.

Centralized Data: All guest and room data is stored in one place, ensuring easy access and efficient management.



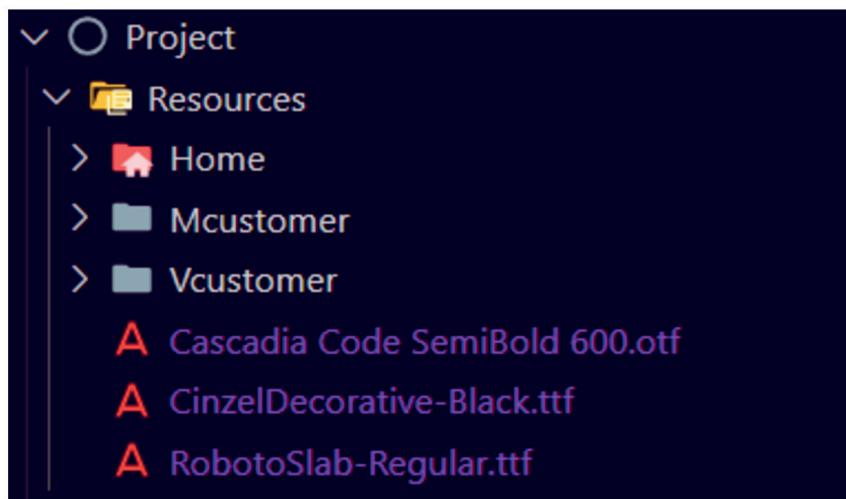


Usage

Documentary

SET UP

Install the required fonts inside of the Resources Folder



Run the MYSQL commands to create the structure of the database

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement

mysql> source C:\Users\sharm\OneDrive\Desktop\Project\SQL_cmds.sql
```

mysql> source path/XII-Project/SQL_cmds.sql



HOME PAGE

Open the “main.py” file inside of XII-Project



The following window will open



With a single click on any of the four buttons, you can open new windows to perform specific tasks and functions. Each button is designed for quick access, simplifying processes and boosting efficiency.



MANAGE CUSTOMERS

ADD CUSTOMERS

CONTINENTAL HOTEL | MANAGE CUSTOMERS

THE CONTINENTAL

ADD CUSTOMERS EDIT CUSTOMERS REMOVE CUSTOMERS

UNIQUE ID	7
FULL NAME	Rajesh Kumar
PHONE NO.	9432761580
EMAIL ADD.	KumaRaj@gmail.com
ADDRESS	South Delhi

SUBMIT



This window will open upon clicking the “MANAGE CUSTOMERS” button.

Here inside of the ADD CUSTOMERS section we can register a customer in our database and collect all the details.

A unique ID is automatically generated and we can enter

Full name

Phone

Email

Address

We can then click the “SUBMIT” button, this will save all the details and then refresh the page for a new entry.



EDIT CUSTOMERS

CONTINENTAL HOTEL | MANAGE CUSTOMERS

THE CONTINENTAL

[ADD CUSTOMERS](#) [EDIT CUSTOMERS](#) [REMOVE CUSTOMERS](#)

UNIQUE ID	<input type="text" value="7"/>
FULL NAME	Rajesh Kumar
PHONE NO.	6969769690
EMAIL ADD.	RajeshK@gmail.com
ADDRESS	North Delhi



SUBMIT

This window will open upon clicking the "EDIT CUSTOMERS" section. Here inside of the EDIT CUSTOMERS section we can modify existing customer in our database and update all the details.

We can choose which customer to modify by selecting their UID in a dropdown menu.

After selecting the UID the text box will fill up with their original details and we can change them as we wish.

We can then click the "SUBMIT" button, this will save all the details and then refresh the page for a new entry.



REMOVE CUSTOMERS



This window will open upon clicking the "REMOVE CUSTOMERS" section.

Here inside of the REMOVE CUSTOMERS section we can wipe the details of an existing customer

We can choose which customer to modify by selecting their name in a dropdown menu.

After selecting the name we can click the "REMOVE" button, this will remove all the details and then refresh the page.



MANAGE ROOMS

ADD ROOMS

CONTINENTAL HOTEL | MANAGE ROOMS

THE CONTINENTAL

ADD ROOMS CLEAR ROOMS BOOK ROOMS

ROOM ID	11
ROOM NAME	A123
FLOOR	12
AC/NON-AC	NON-AC
COST p/n	1200

SUBMIT

This window will open upon clicking the “MANAGE ROOMS” button. Here inside of the ADD ROOMS section we can register a new room in our database and save all the details.

A Room ID is automatically generated and we can enter

Room name

Floor

Email

AC/NON-AC

Cost per night

We can then click the “SUBMIT” button, this will save all the details and then refresh the page for a new entry.



CLEAR ROOMS

CONTINENTAL HOTEL | MANAGE ROOMS

THE CONTINENTAL

ADD ROOMS CLEAR ROOMS BOOK ROOMS

A123

SELECT WHICH ROOM
TO CLEAR/
REMOVE OCCUPANT FROM

REMOVE

THE CONTINENTAL HOTEL
CONTINENTAL
An Oasis of Calm & Civility

This window will open upon clicking the "CLEAR ROOMS" section. Here inside of the CLEAR ROOMS section we can remove the current occupant of the room(if any), this will allow the room to be available to be booked. This also clear their records on the customers table. We can choose which room to clear by selecting their name in a dropdown menu.

After selecting the name we can click the "REMOVE" button, this will clear the room and then refresh the page.



BOOK ROOMS

The screenshot shows a software interface for booking rooms. At the top, there are three buttons: 'ADD ROOMS', 'CLEAR ROOMS', and 'BOOK ROOMS', with 'BOOK ROOMS' being the active tab. Below these, there are three input fields: 'CUSTOMER' (containing 'Rajesh Kumar'), 'ROOM' (containing 'A123'), and 'Checkout' (containing '27-01-25'). To the right of the checkout date field is a large green 'SUBMIT' button. In the top left corner, there is a small window titled 'CONTINENTAL HOTEL | MANAGE ROOMS'. The background of the main window features the hotel's logo and name, 'THE CONTINENTAL' in large red letters.

This window will open upon clicking the "BOOK ROOMS" section. Here inside of the BOOK ROOMS section we can book a room for a designated customer.

We can choose which customer has to book the room by selecting their name in a dropdown menu.

We can choose which room to book by selecting its name in a dropdown menu.

We can also enter the checkout date

We can then click the "SUBMIT" button, this will save all the details and book the room.



VIEW CUSTOMERS

CONTINENTAL HOTEL CUSTOMER INFORMATION					
UNIQUE ID	CUSTOMER NAME	PHONE NO.	EMAIL ADD.	ADDRESS	ROOM
1	Krishn Sharma	8490657869	sharmak69@gmail.com	Assotech	A011
2	Shivam Sharma	6414004315	ShivamS@gmail.com	Shahberi	A024
3	Kapish Singh	7071737809	KSingh@gmail.com	Chipyana	A036
4	Anirvan	2951173282	Anirvan69@gmail.com	Shahberi	N/A
5	Arpit yadav	1210465593	Kammo@gmail.com	Hebatpur	N/A
6	Piyush Singh	6972988724	Piyush713@gmail.com	Gaur city	N/A
7	Rajesh Kumar	9432761580	KumaRaj@gmail.com	South Delhi	N/A

This window will open upon clicking the “VIEW CUSTOMERS ” button. Here inside of the VIEW CUSTOMERS section we can view all of the customer detail in a tabular format.

This allows us to check

UID

Full Name

Phone number

Email address

Home address

Room they're currently occupying



VIEW ROOMS

ALL ROOMS

CONTINENTAL HOTEL | ROOM INFORMATION

THE CONTINENTAL

ALL ROOMS OCCUPIED ROOMS AVAILABLE ROOMS

ROOM ID	ROOM NAME	FLOOR	AC/NON-AC	ROOM COST PER NIGHT₹	OCCUPANT
1	A011	1	AC	500	Krishna Sharma
2	B024	2	AC	1500	Shivam Sharma
3	A036	3	AC	800	Kapish Singh
4	B123	12	NON-AC	6000	N/A
5	B052	5	AC	3000	N/A
6	C032	3	NON-AC	1200	N/A
7	C103	10	AC	5000	N/A
8	A053	5	AC	200	N/A
9	B043	4	AC	2000	N/A
10	A034	3	AC	3000	N/A
11	A123	12	NON-AC	1200	N/A

OCCUPIED ROOMS

CONTINENTAL HOTEL | ROOM INFORMATION

THE CONTINENTAL

ALL ROOMS OCCUPIED ROOMS AVAILABLE ROOMS

ROOM ID	ROOM NAME	FLOOR	AC/NON-AC	ROOM COST PER NIGHT₹	OCCUPANT
1	A011	1	AC	500	Krishna Sharma
2	A024	2	AC	1500	Shivam Sharma
3	A036	3	AC	800	Kapish Singh

AVAILABLE ROOMS

CONTINENTAL HOTEL | ROOM INFORMATION

THE CONTINENTAL

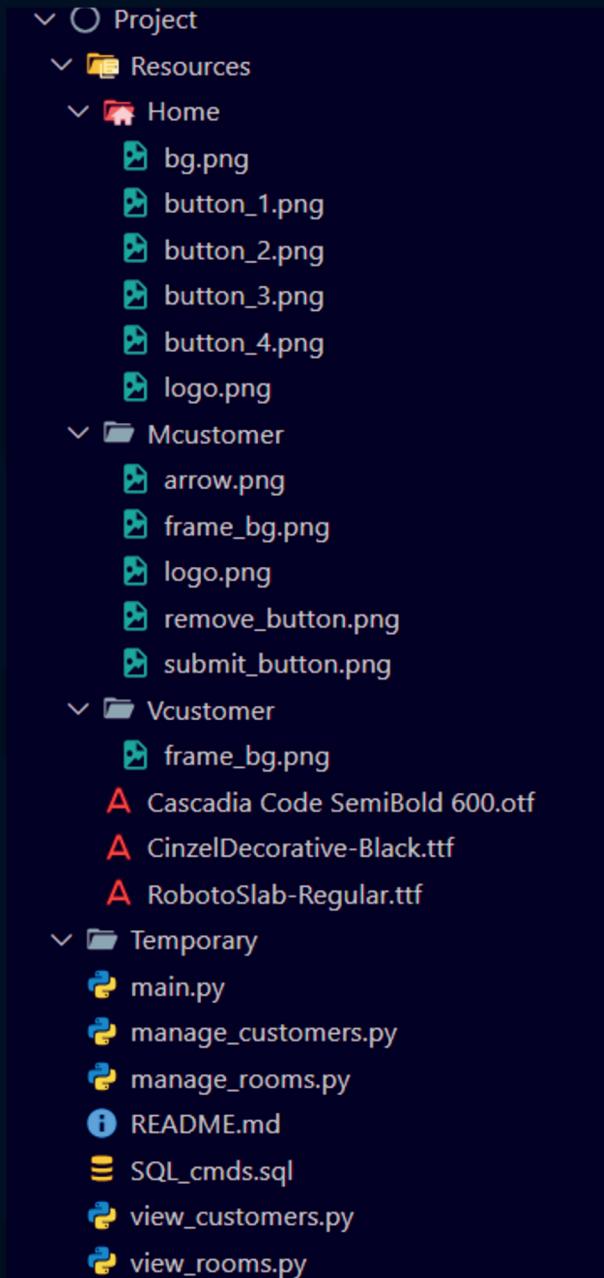
ALL ROOMS OCCUPIED ROOMS AVAILABLE ROOMS

ROOM ID	ROOM NAME	FLOOR	AC/NON-AC	ROOM COST PER NIGHT₹	OCCUPANT
4	B123	12	NON-AC	6000	N/A
5	B052	5	AC	3000	N/A
6	C032	3	NON-AC	1200	N/A
7	C103	10	AC	5000	N/A
8	A053	5	AC	200	N/A
9	B043	4	AC	2000	N/A
10	A034	3	AC	3000	N/A
11	A123	12	NON-AC	1200	N/A



Folder Directory

SOURCE CODE



Source code serves as the foundation of any software application, comprising all the critical instructions and logic written by developers in various programming languages. It dictates how a program functions, processes data, and interacts with users or other systems. Source code can be downloaded for completely free at <https://github.com/kr1shn-dll/XII-Project>

The image on the right displays the folder directory of the project, showing the structured organization of files, such as scripts, resources, and configuration settings.



main.py

```
Project > main.py > ...
1  from tkinter import *
2  import os
3
4  def switch_1():
5      os.system("python manage_customers.py")
6  def switch_2():
7      os.system("python manage_rooms.py")
8  def switch_3():
9      os.system("python view_rooms.py")
10 def switch_4():
11     os.system("python view_customers.py")
12
13 window = Tk()
14
15 bg = PhotoImage(file=r"Resources\Home\bg.png")
16 logo = PhotoImage(file=r"Resources\Home\Logo.png")
17 button_image_1 = PhotoImage(file=r"Resources\Home\button_1.png")
18 button_image_2 = PhotoImage(file=r"Resources\Home\button_2.png")
19 button_image_3 = PhotoImage(file=r"Resources\Home\button_3.png")
20 button_image_4 = PhotoImage(file=r"Resources\Home\button_4.png")
21
22 window.geometry("1280x720")
23 window.configure(bg = "#FFFFFF")
24 window.title("CONTINENTAL HOTEL | HOME PAGE ")
25 window.iconphoto(True,logo)
26
27 canvas = Canvas(window,bg = "#FFFFFF",height = 720,width = 1280,highlightthickness = 0)
28 canvas.place(x = 0, y = 0)
29
30 canvas.create_image(640,360,image=bg)
31 canvas.create_image(1030,420,image=logo)
32
33 canvas.create_text(975,696,anchor="nw",text="Made by Krishn XII-O 2025",fill="#950B3C",font=("Cascadia Code SemiBold", 20 * -1))
34 canvas.create_text(47,0,anchor="nw",text="THE CONTINENTAL",fill="#950B3C",font=("Cinzel Decorative Black", 106 * -1))
35
36 button_1 = Button(image=button_image_1,borderwidth=0,highlightthickness=0,command=switch_1,relief="flat")
37 button_1.place(x=0,y=194,width=658,height=136)
38
39 button_2 = Button(image=button_image_2,borderwidth=0,highlightthickness=0,command=switch_2,relief="flat")
40 button_2.place(x=0,y=330,width=658,height=120)
41
42 button_3 = Button(image=button_image_3,borderwidth=0,highlightthickness=0,command=switch_3,relief="flat")
43 button_3.place(x=0,y=440,width=658,height=152)
44
45 button_4 = Button(image=button_image_4,borderwidth=0,highlightthickness=0,command=switch_4,relief="flat")
46 button_4.place(x=0,y=580,width=658,height=140)
47
48 window.resizable(False,False)
49 window.mainloop()
```



manage_customers.py

```
Project > manage_customers.py > ...
1  from tkinter import *
2  from tkinter import messagebox
3  import mysql.connector as sql
4
5  connection = sql.connect(host="localhost",
6                           user="root",
7                           passwd="root",
8                           database="Continental",
9                           auth_plugin='mysql_native_password')
10 cursor = connection.cursor()
11
12 def add_submit(uid,name,phone,email,address):
13     a = int(uid.get())
14     b = name.get()
15     c = str(phone.get())
16     d = email.get()
17     e = address.get()
18     try:
19         cursor.execute(f"INSERT INTO customers VALUES('{a}', '{b}', '{c}', '{d}', '{e}', 'N/A');")
20         connection.commit()
21     except:
22         messagebox.showerror(title="STATUS", message="UID already in use")
23     else:
24         messagebox.showinfo(title="STATUS", message="CUSTOMER ADDED SUCCESFULLY")
25     finally:
26         add_page.destroy()
27         add_switch()
28
29 def edit_submit(uid,name,phone,email,address):
30     a = uid.get()
31     b = name.get()
32     c = phone.get()
33     d = email.get()
34     e = address.get()
35     answer = messagebox.askyesno(title="CONFIRM CHANGES", message="ARE YOU SURE YOU WANT TO EDIT INFO?")
36     if answer:
37         cursor.execute(f"update customers set Name = '{b}', Phone = {c}, Email='{d}',Address='{e}' WHERE UID = '{a}';")
38         connection.commit()
39         messagebox.showinfo(title="STATUS", message="CUSTOMER EDITED SUCCESFULLY")
40         edit_page.destroy()
41         edit_switch()
42
43 def remove_submit(name):
44     answer = messagebox.askyesno(title="CONFIRM CHANGES", message="ARE YOU SURE YOU WANT TO DELETE CUSTOMER?")
45     a = name.get()
46     if answer:
47         cursor.execute(f"delete from customers where name = '{a}';")
48         connection.commit()
49         messagebox.showinfo(title="STATUS", message="CUSTOMER DELETED SUCCESFULLY")
50         remove_page.destroy()
51         remove_switch()
52
53
54 window = Tk()
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 23

Project > manage_customers.py > ...

```
56     bg = PhotoImage(file=r"Resources\Home\bg.png")
57     logo = PhotoImage(file=r"Resources\Mcustomer\Logo.png")
58     frame_bg = PhotoImage(file=r"Resources\Mcustomer\frame_bg.png")
59     submitButton = PhotoImage(file=r"Resources\Mcustomer\submit_button.png")
60     removeButton = PhotoImage(file=r"Resources\Mcustomer\remove_button.png")
61
62     window.geometry("1280x720")
63     window.configure(bg = "#FFFFFF")
64     window.title("CONTINENTAL HOTEL / MANAGE CUSTOMERS ")
65     window.iconphoto(True,logo)
66
67     canvas = Canvas(window,bg = "#FFFFFF",height = 720,width = 1280,highlightthickness = 0)
68     canvas.place(x = 0, y = 0)
69
70     canvas.create_image(640,360,image=bg)
71     canvas.create_text(47,0,anchor="nw",text="THE CONTINENTAL",fill="#950B3C",font=("Cinzel Decorative Black", 106 * -1))
72
73     menu = Frame(window,bg="#CDA84C")
74     menu.pack_propagate(False)
75     menu.configure(width=1280,height=100)
76     menu.pack(pady=135)
77
78     def switch(label,frame):
79         add_label['bg'] = "#CDA84C"
80         edit_label['bg'] = "#CDA84C"
81         remove_label['bg'] = "#CDA84C"
82         label['bg'] = "#950B3C"
83
84         add_page.destroy()
85         edit_page.destroy()
86         remove_page.destroy()
87         frame()
88
89     add_button = Button(menu,text="ADD CUSTOMERS",font=("Cascadia Code SemiBold",32),bd=0,
90                         fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
91                         command = lambda: switch(add_label,add_switch))
92     add_button.place(x=20,y=0,width=400)
93     add_label = Label(menu,bg="#950B3C")
94     add_label.place(x=20,y=90,width=400,height=3)
95
96     edit_button = Button(menu,text="EDIT CUSTOMERS",font=("Cascadia Code SemiBold",32),bd=0,
97                          fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
98                          command = lambda: switch(edit_label,edit_switch))
99     edit_button.place(x=440,y=0,width=400)
100    edit_label = Label(menu,bg="#CDA84C")
101    edit_label.place(x=440,y=90,width=400,height=3)
102
103    remove_button = Button(menu,text="REMOVE CUSTOMERS",font=("Cascadia Code SemiBold",32),bd=0,
104                           fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
105                           command = lambda: switch(remove_label,remove_switch))
106    remove_button.place(x=860,y=0,width=400)
107    remove_label = Label(menu,bg="#CDA84C")
108    remove_label.place(x=860,y=90,width=400,height=3)
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 24

```
Project > manage_customers.py > ...
110 main = Frame(window, bg="#CDA84C")
111 main.configure(width=1280, height=470)
112 main.place(x=0, y=250)
113
114 def add_switch():
115     global add_page
116     add_page = Frame(main, bg="#CDA84C")
117     add_page.configure(width=1280, height=470)
118     add_page.place(x=0, y=0)
119     canvas = Canvas(add_page, bg = "#FFFFFF", height = 470, width = 1280, bd = 0, highlightthickness = 0, relief = "ridge")
120     canvas.place(x = 0, y = 0)
121     canvas.create_image(640.0, 235.0, image=frame_bg)
122     canvas.create_image(1032.0, 151.0, image=logo)
123     canvas.create_text(36.0, 42.0, anchor="nw", text="UNIQUE ID", fill="#000000", font=("Roboto Slab", 40 * -1))
124     canvas.create_text(36.0, 125.0, anchor="nw", text="FULL NAME", fill="#000000", font=("Roboto Slab", 40 * -1))
125     canvas.create_text(36.0, 208.0, anchor="nw", text="PHONE NO.", fill="#000000", font=("Roboto Slab", 40 * -1))
126     canvas.create_text(36.0, 291.0, anchor="nw", text="EMAIL ADD.", fill="#000000", font=("Roboto Slab", 40 * -1))
127     canvas.create_text(36.0, 374.0, anchor="nw", text="ADDRESS", fill="#000000", font=("Roboto Slab", 40 * -1))
128     uid = Entry(add_page, bd=0, bg="#FFFFFF", fg="#000000", highlightthickness=0, font=("Calibri", 40))
129     uid.place(x=290.0, y=42.0, width=459.0, height=53.0)
130     cursor.execute("select MAX(UID) from customers")
131     data = cursor.fetchall()
132     uid.insert(0, data[0][0]+1)
133     name = Entry(add_page, bd=0, bg="#FFFFFF", fg="#000000", highlightthickness=0, font=("Calibri", 40))
134     name.place(x=290.0, y=125.0, width=459.0, height=53.0)
135     phone = Entry(add_page, bd=0, bg="#FFFFFF", fg="#000000", highlightthickness=0, font=("Calibri", 40))
136     phone.place(x=290.0, y=208.0, width=459.0, height=53.0)
137     email = Entry(add_page, bd=0, bg="#FFFFFF", fg="#000000", highlightthickness=0, font=("Calibri", 40))
138     email.place(x=290.0, y=291.0, width=459.0, height=53.0)
139     address = Entry(add_page, bd=0, bg="#FFFFFF", fg="#000000", highlightthickness=0, font=("Calibri", 40))
140     address.place(x=290.0, y=374.0, width=459.0, height=53.0)
141     Button(add_page, image=submitButton, borderwidth=0, highlightthickness=0,
142             command=lambda: add_submit(uid, name, phone, email, address), relief="flat").place(x=836.0, y=289.0,
143                                         width=391.0, height=113.0)
144
145 add_switch()
146
147 def edit_switch():
148     global edit_page
149     edit_page = Frame(main, bg="#CDA84C")
150     edit_page.configure(width=1280, height=470)
151     edit_page.place(x=0, y=0)
152     canvas = Canvas(edit_page, bg = "#FFFFFF", height = 470, width = 1280, bd = 0, highlightthickness = 0, relief = "ridge")
153     canvas.place(x = 0, y = 0)
154     canvas.create_image(640.0, 235.0, image=frame_bg)
155     canvas.create_image(1032.0, 151.0, image=logo)
156     canvas.create_text(36.0, 42.0, anchor="nw", text="UNIQUE ID", fill="#000000", font=("Roboto Slab", 40 * -1))
157     canvas.create_text(36.0, 125.0, anchor="nw", text="FULL NAME", fill="#000000", font=("Roboto Slab", 40 * -1))
158     canvas.create_text(36.0, 208.0, anchor="nw", text="PHONE NO.", fill="#000000", font=("Roboto Slab", 40 * -1))
159     canvas.create_text(36.0, 291.0, anchor="nw", text="EMAIL ADD.", fill="#000000", font=("Roboto Slab", 40 * -1))
160     canvas.create_text(36.0, 374.0, anchor="nw", text="ADDRESS", fill="#000000", font=("Roboto Slab", 40 * -1))
161     uid = StringVar()
162     cursor.execute("select * from customers;")
163     data = cursor.fetchall()
164     customers = []
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 25

```
Project > manage_customers.py > edit_switch
165     for i in data:
166         customers.append(i[0])
167     def select(uid):
168         name.delete(0,END)
169         phone.delete(0,END)
170         email.delete(0,END)
171         address.delete(0,END)
172         for i in data:
173             if i[0] == uid:
174                 name.insert(0,i[1])
175                 phone.insert(0,i[2])
176                 email.insert(0,i[3])
177                 address.insert(0,i[4])
178
179     OptionMenu(edit_page,uid,*customers,command= lambda uid: select(uid) ).place(x=290.0,y=42.0,width=459.0,height=53.0)
180     name = Entry(edit_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
181     name.place(x=290.0,y=127.0,width=459.0,height=53.0)
182     phone = Entry(edit_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
183     phone.place(x=290.0,y=212.0,width=459.0,height=53.0)
184     email = Entry(edit_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
185     email.place(x=290.0,y=297,width=459.0,height=53.0)
186     address = Entry(edit_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
187     address.place(x=290.0,y=382,width=459.0,height=53.0)
188     Button(edit_page,image=submitButton,borderwidth=0,highlightthickness=0,
189             command=lambda: edit_submit(uid,name,phone,email,address),relief="flat").place(x=836.0,y=289.0,
190                                         width=391.0,height=113.0)
191     edit_switch()
192     edit_page.destroy()
193
194 def remove_switch():
195     global remove_page
196     remove_page = Frame(main,bg="#CDA84C")
197     remove_page.configure(width=1280,height=470)
198     remove_page.place(x=0,y=0)
199     canvas = Canvas(remove_page,bg = "#FFFFFF",height = 470,width = 1280,bd = 0,highlightthickness = 0,relief = "ridge")
200     canvas.place(x = 0, y = 0)
201     canvas.create_image(640.0,235.0,image=frame_bg)
202     canvas.create_image(1032.0,151.0,image=logo)
203     canvas.create_text(36.0,125.0,anchor="nw",text="SELECT WHICH CUSTOMER\nTO DELETE",fill="#000000",font=("Roboto Slab", 40 * -1))
204     name = StringVar()
205     customers = []
206     cursor.execute("select name from customers;")
207     for i in cursor.fetchall():
208         customers.append(i[0])
209     OptionMenu(remove_page,name,*customers).place(x=290.0,y=42.0,width=459.0,height=53.0)
210     Button(remove_page,image=removeButton,borderwidth=0,highlightthickness=0,
211             command=lambda: remove_submit(name),relief="flat").place(x=836.0,y=289.0,
212                                         width=391.0,height=113.0)
213     remove_switch()
214     remove_page.destroy()
215
216
217
218 window.resizable(False,False)
219 window.mainloop()
```



manage_rooms.py

```
Project > manage_rooms.py > book_submit
1  from tkinter import *
2  from tkinter import messagebox
3  import mysql.connector as sql
4
5
6  connection = sql.connect(host="localhost",
7  user="root",
8  passwd="root",
9  database="Continental",
10 auth_plugin='mysql_native_password')
11
12 cursor = connection.cursor()
13
14 window = Tk()
15
16 def add_submit(rid,name,floor,ac,cost):
17     a = rid.get()
18     b = name.get()
19     c = floor.get()
20     d = ac.get()
21     e = cost.get()
22     try:
23         cursor.execute(f"insert into rooms values('{a}', '{b}', '{c}', '{d}', '{e}', 'N/A', 'N/A');")
24         connection.commit()
25     except:
26         messagebox.showerror(title="STATUS",message="RID already in use")
27     else:
28         messagebox.showinfo(title="STATUS",message="Room Added successfully")
29     finally:
30         add_page.destroy()
31         add_switch()
32
33 def remove_submit(name):
34     a = name.get()
35     cursor.execute("select name,rid from rooms;")
36     for i in cursor.fetchall():
37         if i[0] == a:
38             b = i[1]
39             break
40     answer = messagebox.askyesno(title="STATUS",message="Are you sure you want to remove occupant")
41     if answer:
42         cursor.execute(f"update rooms set occupant = 'N/A',checkout = 'N/A' where rid = {b};")
43         cursor.execute(f"update customers set room = 'N/A' where room = '{a}'")
44         connection.commit()
45         messagebox.showinfo(title="STATUS",message="Room cleared successfully")
46         remove_page.destroy()
47         remove_switch()
48
49 def book_submit(customer,room,checkout):
50     a = customer.get()
51     b = room.get()
52     c = checkout.get()
53     cursor.execute("select name,room from customers;")
54     V1 = bool
55     V2 = bool
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 27

```
Project > manage_rooms.py > book_submit
49 def book_submit(customer,room,checkout):
50     for i in cursor.fetchall():
51         if a == i[0]:
52             if i[1] == "N/A":
53                 V1 = True
54                 break
55             else:
56                 V1 = False
57     cursor.execute("select name,occupant,checkout from rooms;")
58     for i in cursor.fetchall():
59         if b == i[0] and i[1] == "N/A" and i[2] == "N/A":
60             V2 = True
61             break
62         else:
63             V2 = False
64     if V1 and V2:
65         cursor.execute(f"update rooms set occupant = '{a}', checkout ='{c}' where name = '{b}';")
66         cursor.execute(f"update customers set room = '{b}' where name = '{a}'")
67         connection.commit()
68         messagebox.showinfo(title="STATUS",message="ROOM BOOKED SUCCESSFULLY")
69     elif not V1:
70         messagebox.showerror(title="STATUS",message="Customer already in a room ")
71     elif not V2:
72         messagebox.showerror(title="STATUS",message="Room already occupied")
73     book_page.destroy()
74     book_switch()
75
76 bg = PhotoImage(file=r"Resources\Home\bg.png")
77 logo = PhotoImage(file=r"Resources\Mcustomer\Logo.png")
78 arrow = PhotoImage(file=r"Resources\Mcustomer\arrow.png")
79 frame_bg = PhotoImage(file=r"Resources\Mcustomer\frame_bg.png")
80 submitButton = PhotoImage(file=r"Resources\Mcustomer\submit_button.png")
81 removeButton = PhotoImage(file=r"Resources\Mcustomer\remove_button.png")
82
83 window.geometry("1280x720")
84 window.configure(bg = "#FFFFFF")
85 window.title("CONTINENTAL HOTEL / MANAGE ROOMS ")
86 window.iconphoto(True,logo)
87
88 canvas = Canvas(window,bg = "#FFFFFF",height = 720,width = 1280,highlightthickness = 0)
89 canvas.place(x = 0, y = 0)
90
91 canvas.create_image(640,360,image=bg)
92 canvas.create_text(47,0,anchor="nw",text="THE CONTINENTAL",fill="#950B3C",font=("Cinzel Decorative Black", 106 * -1))
93
94 menu = Frame(window,bg="#CDA84C")
95 menu.pack_propagate(False)
96 menu.configure(width=1280,height=100)
97 menu.pack(pady=135)
98
99
100 def switch(label,frame):
101     add_label['bg'] = "#CDA84C"
102     remove_label['bg'] = "#CDA84C"
103     book_label['bg'] = "#CDA84C"
104     label['bg'] = "#950B3C"
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 28

```
① Project > manage_rooms.py > book_submit
105 def switch(label,frame):
111     add_page.destroy()
112     remove_page.destroy()
113     book_page.destroy()
114     frame()
115
116 add_button = Button(menu,text="ADD ROOMS",font=("Cascadia Code SemiBold",32),bd=0,
117                         fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
118                         command = lambda: switch(add_label,add_switch))
119 add_button.place(x=20,y=0,width=400)
120 add_label = Label(menu,bg="#950B3C")
121 add_label.place(x=20,y=90,width=400,height=3)
122
123 remove_button = Button(menu,text="CLEAR ROOMS",font=("Cascadia Code SemiBold",32),bd=0,
124                         fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
125                         command = lambda: switch(remove_label,remove_switch))
126 remove_button.place(x=440,y=0,width=400)
127 remove_label = Label(menu,bg="#CDA84C")
128 remove_label.place(x=440,y=90,width=400,height=3)
129
130 book_button = Button(menu,text="BOOK ROOMS",font=("Cascadia Code SemiBold",32),bd=0,
131                         fg="#950B3C",activeforeground="#950B3C",bg="#CDA84C",activebackground="#CDA84C",
132                         command = lambda: switch(book_label,book_switch))
133 book_button.place(x=860,y=0,width=400)
134 book_label = Label(menu,bg="#CDA84C")
135 book_label.place(x=860,y=90,width=400,height=3)
136
137 main = Frame(window,bg="#CDA84C")
138 main.configure(width=1280,height=470)
139 main.place(x=0,y=250)
140
141 def add_switch():
142     global add_page
143     add_page = Frame(main,bg="#CDA84C")
144     add_page.configure(width=1280,height=470)
145     add_page.place(x=0,y=0)
146     canvas = Canvas(add_page,bg = "#FFFFFF",height = 470,width = 1280,bd = 0,highlightthickness = 0,relief = "ridge")
147     canvas.place(x = 0, y = 0)
148     canvas.create_image(640.0,235.0,image=frame_bg)
149     canvas.create_image(1032.0,151.0,image=logo)
150     canvas.create_text(36.0,42.0,anchor="nw",text="ROOM ID",fill="#000000",font=("Roboto Slab", 40 * -1))
151     canvas.create_text(36.0,125.0,anchor="nw",text="ROOM NAME",fill="#000000",font=("Roboto Slab", 39 * -1))
152     canvas.create_text(36.0,208.0,anchor="nw",text="FLOOR",fill="#000000",font=("Roboto Slab", 40 * -1))
153     canvas.create_text(36.0,291.0,anchor="nw",text="AC/NON-AC",fill="#000000",font=("Roboto Slab", 40 * -1))
154     canvas.create_text(36.0,374.0,anchor="nw",text="COST p/n",fill="#000000",font=("Roboto Slab", 40 * -1))
155     rid = Entry(add_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
156     rid.place(x=290.0,y=42.0,width=459.0,height=53.0)
157     cursor.execute("select MAX(RID) from rooms")
158     data = cursor.fetchall()
159     rid.insert(0,data[0][0]+1)
160     name = Entry(add_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
161     name.place(x=290.0,y=127.0,width=459.0,height=53.0)
162     floor = Entry(add_page,bd=0,bg="#FFFFFF",fg="#000000",highlightthickness=0,font=("Calibri",40))
163     floor.place(x=290.0,y=212.0,width=459.0,height=53.0)
164     ac = StringVar()
```



THE CONTINENTAL HOTEL MANAGEMENT

Page: 29



view_customers.py

Project > view_customers.py > ...

```
1  from tkinter import *
2  from tkinter import ttk
3  import mysql.connector as sql
4
5
6  connection = sql.connect(host="localhost",
7  user="root",
8  passwd="root",
9  database="Continental",
10 auth_plugin='mysql_native_password')
11
12 cursor = connection.cursor()
13
14 window = Tk()
15
16 bg = PhotoImage(file=r"Resources\Home\bg.png")
17 logo = PhotoImage(file=r"Resources\Home\Logo.png")
18 frame_bg = PhotoImage(file=r"Resources\Vcustomer\frame_bg.png")
19
20 window.geometry("1280x720")
21 window.configure(bg = "#FFFFFF")
22 window.title("CONTINENTAL HOTEL / CUSTOMER INFORMATION ")
23 window.iconphoto(True,logo)
24
25 canvas = Canvas(window,bg = "#FFFFFF",height = 720,width = 1280,highlightthickness = 0)
26 canvas.place(x = 0, y = 0)
27
28 canvas.create_image(640,360,image=bg)
29 canvas.create_text(47,0,anchor="nw",text="THE CONTINENTAL",fill="#950B3C",font=("Cinzel Decorative Black", 106 * -1))
30 canvas.create_image(640.0,422.5,image=frame_bg)
31
32 CustomerTable = ttk.Treeview(window,columns=("uid","name","phone","email","address","room"),show="headings")
33 CustomerTable.heading("uid",text="UNIQUE ID")
34 CustomerTable.heading("name",text="CUSTOMER NAME")
35 CustomerTable.heading("phone",text="PHONE NO.")
36 CustomerTable.heading("email",text="EMAIL ADD.")
37 CustomerTable.heading("address",text="ADDRESS")
38 CustomerTable.heading("room",text="ROOM")
39 CustomerTable.place(x=40,y=160,height=530)
40
41 cursor.execute("select * from customers;")
42 data = cursor.fetchall()
43 for i in data:
44     CustomerTable.insert(parent="",index=END,values=i)
45
46 window.resizable(False,False)
47 window.mainloop()
```



view_rooms.py



THE CONTINENTAL HOTEL MANAGEMENT

Page: 32



THE CONTINENTAL HOTEL MANAGEMENT

Page: 33

```
① Project > view_rooms.py > available_switch
111     for i in cursor.fetchall():
112         table.insert(parent="",index=END,values=i)
113     occupied_switch()
114     occupied_page.destroy()
115
116 def available_switch():
117     global available_page
118     available_page = Frame(main,bg="#CDA84C")
119     available_page.configure(width=1280,height=470)
120     available_page.place(x=0,y=0)
121     canvas = Canvas(available_page,bg = "#FFFFFF",height = 470,width = 1280,bd = 0,highlightthickness = 0,relief = "ridge")
122     canvas.place(x = 0, y = 0)
123     canvas.create_image(640.0,235.0,image=frame_bg)
124     table = ttk.Treeview(available_page,columns=("rid","name","floor","ac","cost","occupant"),show="headings")
125     table.heading("rid",text="ROOM ID")
126     table.heading("name",text="ROOM NAME")
127     table.heading("floor",text="FLOOR")
128     table.heading("ac",text="AC/NON-AC")
129     table.heading("cost",text="ROOM COST PER NIGHT(₹)")
130     table.heading("occupant",text="OCCUPANT")
131     cursor.execute("select * from rooms where occupant = 'N/A' and checkout = 'N/A';")
132     table.place(x=40,y=40,height=395)
133     for i in cursor.fetchall():
134         table.insert(parent="",index=END,values=i)
135     available_switch()
136     available_page.destroy()
137
138 window.resizable(False,False)
139 window.mainloop()
```

SQL_cmds.sql

```
① Project > SQL_cmds.sql
1  create database Continental;
2
3  use Continental;
4
5  CREATE TABLE customers(
6      `UID` INT NOT NULL,
7      `Name` VARCHAR(32) NOT NULL,
8      `Phone` VARCHAR(10) NOT NULL,
9      `Email` VARCHAR(45) NOT NULL,
10     `Address` VARCHAR(45) NOT NULL,
11     `Room` VARCHAR(45) NOT NULL,
12     PRIMARY KEY (`UID`));
13
14  create table Rooms(
15      `RID` INT NOT NULL,
16      `Name` VARCHAR(10) NOT NULL,
17      `Floor` INT NOT NULL,
18      `ac` VARCHAR(6) NOT NULL,
19      `Cost` INT NOT NULL,
20      `Occupant` VARCHAR(45) NOT NULL,
21      `Checkout` VARCHAR(45) NOT NULL,
22      PRIMARY KEY (`RID`));
23
24  insert into customers values(1,'Krishn Sharma','8490657869','sharmak69@gmail.com','Assotech','N/A');
25  insert into customers values(2,'Shivam Sharma','6414004315','Shivams@gmail.com','Shahberi','N/A');
26  insert into customers values(3,'Kapish Singh','7071737809','KSingh@gmail.com','Chipyana','N/A');
27
28  insert into rooms values(1,'A011',1,'AC',500,'N/A','N/A');
29  insert into rooms values(2,'A024',2,'AC',1500,'N/A','N/A');
30  insert into rooms values(3,'A036',3,'AC',800,'N/A','N/A');
```



Bibliography

Bro Code - YouTube



<https://www.youtube.com/>

@BroCodez

Clear Code - YouTube



<https://www.youtube.com/>

@ClearCode

Stack Overflow



<https://www.stackoverflow.com>

Leet Code



<https://www.leetcode.com/>

GitHub and Git



<https://www.github.com/>

Arora, Preeti. Computer Science with Python: Textbook for CBSE Class 12. New Delhi: Vikas Publishing, 2023.



**THE CONTINENTAL
HOTEL MANAGEMENT**

Page: 35

Thanks



<https://github.com/kr1shn-dll/XII-Project>