

VITYARTHI PROJECT: HOTEL MANAGEMENT SYSTEM

By
PRISHA ARORA
(25BAI10290)



**VIT Bhopal University
Kothrikalan -466114,
Madhya Pradesh,
India**

Algorithm:

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

Main Program:

```
❶ main.py(7,1) E: no file named __init__.py
  from tkinter import *
  from PIL import Image,ImageTk
  from customer import cust_wnd
  from room import room_wnd
  from details import details_wnd
  class HotelManagementSystem:
    def __init__(self,root):
      self.root = root
      self.root.title("Hotel Management System")
      self.root.geometry("1000x600x400")
      #-----[Image]-----
      img=Image.open("C:\Users\Abhishek\Desktop\Hotel\images\img1.jpg")
      img_resized=img.resize((100,100),Image.ANTIALIAS)
      self.photologo=ImageTk.PhotoImage(img_resized)
      self.photologo_label=Label(self.root,image=self.photologo,bd=0,relief="none")
      self.photologo_label.place(x=0,y=0,width=100,height=100)
      #-----[Title Label]-----
      self.title_label=Label(self.root,text="HOTEL MANAGEMENT SYSTEM",font=("times new roman",40,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
      self.title_label.place(x=0,y=150,width=350,height=60)
      #-----[Main Frame]-----
      self.main_frame=Frame(self.root,bd=4,relief=RIDGE)
      self.main_frame.place(x=0,y=150,width=350,height=350)
      #-----[Menu Label]-----
      self.menu_label=Label(self.main_frame,text="MENU",font=("times new roman",20,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
      self.menu_label.place(x=0,y=0,width=100)
      #-----[Main Frame Buttons]-----
      btn_Frame=Frame(self.main_frame,bd=4,relief=RIDGE)
      btn_Frame.place(x=0,y=0,width=100,height=30)
      #-----[Customer Details Buttons]-----
      cust_btn=Button(btn_Frame,text="Customer",command=self.cust_details,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")
      cust_btn.grid(row=0,column=0,pady=1)
      room_btn=Button(btn_Frame,text="Room",command=self.roombooking,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")
      room_btn.grid(row=0,column=1,pady=1)
      details_btn=Button(btn_Frame,text="Details",command=self.details_room,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")
      details_btn.grid(row=0,column=2,pady=1)
      logout_btn=Button(btn_Frame,text="Logout",command=self.logout,width=17,font=("times new roman",17,"bold"),bg="black",fg="gold",bd=0,cursor="hand2")
      logout_btn.grid(row=0,column=3,pady=1)
      #-----[Customer Details Images]-----
      self.photocust=Image.open("C:\Users\Abhishek\Desktop\Hotel\images\cust1.jpg")
      self.photocust_resized=ImageTk.PhotoImage(self.photocust)
      self.photocust_label=Label(self.main_frame,image=self.photocust_resized,bd=0,relief="none")
      self.photocust_label.place(x=0,y=40,width=250,height=250)
      #-----[Room Booking Images]-----
      self.photobook=Image.open("C:\Users\Abhishek\Desktop\Hotel\images\book1.jpg")
      self.photobook_resized=ImageTk.PhotoImage(self.photobook)
      self.photobook_label=Label(self.main_frame,image=self.photobook_resized,bd=0,relief="none")
      self.photobook_label.place(x=0,y=250,width=250,height=250)
      #-----[Room Details Images]-----
      self.photodet=Image.open("C:\Users\Abhishek\Desktop\Hotel\images\det1.jpg")
      self.photodet_resized=ImageTk.PhotoImage(self.photodet)
      self.photodet_label=Label(self.main_frame,image=self.photodet_resized,bd=0,relief="none")
      self.photodet_label.place(x=0,y=500,width=250,height=250)
      #-----[Logout Images]-----
      self.logoutimg=Image.open("C:\Users\Abhishek\Desktop\Hotel\images\logout1.jpg")
      self.logoutimg_resized=ImageTk.PhotoImage(self.logoutimg)
      self.logoutlabel=Label(self.main_frame,image=self.logoutimg_resized,bd=0,relief="none")
      self.logoutlabel.place(x=0,y=750,width=250,height=250)
      #-----[Customer Details]-----
      def cust_details(self):
        self.new_window=Toplevel(self.root)
        self.app=cust_wnd(self.new_window)
        self.createbooking(self.root)
```

```

        binding_label = tk.Label(Frame, image=wl_photograph, bd=0, relief="flat")
        binding_label.place(x=100, y=100, width=100, height=100)

    def cert_details(self):
        self.new_window = tk.Toplevel(self.root)
        self.app_cert_main(self.new_window)

    def roombinding(self):
        self.new_window = tk.Toplevel(self.root)
        self.app_roombinding(self.new_window)

    def details_main(self):
        self.new_window = tk.Toplevel(self.root)
        self.app_details_main(self.new_window)

    def logout(self):
        self.root.destroy()

if __name__ == "__main__":
    obj = login()
    obj.login()
    root.mainloop()

```

Customer Details Program:

```

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

class cust_wini:
    def __init__(self,root):
        self.root=root
        self.root.title("Hotel Management System")
        self.root.geometry("1600x600+230+95")
        ####Variables#####
        self.var_ref=StringVar()
        x=random.randint(1000,9999)
        self.var_ref.set(str(x))

        self.var_cust_name=StringVar()
        self.var_mother=StringVar()
        self.var_gender=StringVar()
        self.var_email=StringVar()
        self.var_mobile=StringVar()
        self.var_nationality=StringVar()
        self.var_id_proof=StringVar()
        self.var_id_number=StringVar()
        self.var_address=StringVar()

        #####Title#####
        lbl_title=Label(self.root,text="ADD CUSTOMER DETAILS",font=( "times new roman",30,"bold"),bg="black",fg="gold",bd=4,relief=RIDGE)
        lbl_title.place(x=0,y=0,width=1080,height=60)

        #####Image#####
        img=Image.open("C:\Users\valade\OneDrive\Desktop\hotel\images\logo1.png")
        img2=ImageTk.PhotoImage(img)
        self.photobtn=ttk.Button(self.root,image=img2,photofn=photoing,command=photoing,bd=4,relief=RIDGE)
        self.photobtn.place(x=0,y=0,width=110,height=60)

        #####Label#####
        label1=Label(self.root,bd=4,relief=RIDGE,font=( "times new roman",22,"bold"),padx=2)

```

```

label=Label(labelframeleft,labelframeleft.root,bd=2,relief=RIDGE,text="Customer Ref",font=("times new roman",22,"bold"),padx=2)
label.place(x=10,y=100, width=250, height=60)
label.config(bg="#FFFFCC")
label["text"]=""*10

#Labels and Entry
#cost ref
l1=Label(labelframeleft,text="Customer Ref",font=("times new roman",20,"bold")),padx=2,pady=4
l1.grid(row=0,column=0,sticky="w")
entry=l1.Entry(labelframeleft,width=24,testvariable=self.var_ref,font=("times new roman",21,"bold"),state="readonly")
entry.grid(row=0,column=1)

#cost name
cname=Label(labelframeleft,text="Customer Name:",font=("times new roman",20,"bold"),padx=2,pady=4)
cname.grid(row=1,column=0,sticky="w")
txtcname=ttk.Entry(labelframeleft,width=24,testvariable=self.var_cust_name,font=("times new roman",21,"bold"))
txtcname.grid(row=1,column=1)

#cost mother
l1name=Label(labelframeleft,text="Mother Name:",font=("times new roman",20,"bold"),padx=2,pady=4)
l1name.grid(row=2,column=0,sticky="w")
txtname=ttk.Entry(labelframeleft,width=24,testvariable=self.var_mother,font=("times new roman",21,"bold"))
txtname.grid(row=2,column=1)

#gender codbox
label_gender=Label(labelframeleft,text="Gender:",font=("times new roman",20,"bold"),padx=2,pady=4)
label_gender.grid(row=3,column=0,sticky="w")
combo_gender=ttk.Combobox(labelframeleft,width=23,testvariable=self.var_gender,font=("times new roman",21,"bold"),state="readonly")
combo_gender["value"]=[ "Male","Female","Other"]
combo_gender.current(0)
combo_gender.grid(row=3,column=1)

#postcode
l1postcode=Label(labelframeleft,text="Postcode:",font=("times new roman",20,"bold"),padx=2,pady=4)
l1postcode.grid(row=4,column=0,sticky="w")
txtpostcode=ttk.Entry(labelframeleft,width=24,testvariable=self.var_post,font=("times new roman",21,"bold"))
txtpostcode.grid(row=4,column=1)

#mobile number

```

```

70
71         addressnumber
72         lbbmobile=Label(labelframeleft,text="Mobile:",font=("times new roman",20,"bold"),padx=2,pady=6)
73         lbbmobile.grid(row=6,column=stickyW)
74         txbmobile=Entry(labelframeleft,width=14,textvariable=self.var_mobile,font=("times new roman",20,"bold"))
75         txbmobile.grid(row=6,column=1)

76         nationality
77         lbbnationality=Label(labelframeleft,text="Nationality:",font=("times new roman",20,"bold"),padx=2,pady=6)
78         lbbnationality.grid(row=7,column=stickyW)
79         combonationality=Combobox(labelframeleft,width=23,textvariable=self.var_nationality,font=("times new roman",20,"bold"),state="readonly")
80         combonationality['value']=['Indian','American','Russian']
81         combonationality.current(0)
82         combonationality.grid(row=7,column=1)

83         idprof
84         lbbidprof=Label(labelframeleft,text="ID Proof Types:",font=("times new roman",20,"bold"),padx=2,pady=6)
85         lbbidprof.grid(row=8,column=stickyW)
86         comboidprof=Combobox(labelframeleft,width=23,textvariable=self.var_id_proof,font=("times new roman",20,"bold"),state="readonly")
87         comboidprof['value']=['Aadharcard','Driving license','Passport']
88         comboidprof.current(0)
89         comboidprof.grid(row=8,column=1)

90         idnumber
91         lbbidnumber=Label(labelframeleft,text="ID Number:",font=("times new roman",20,"bold"),padx=2,pady=6)
92         lbbidnumber.grid(row=9,column=stickyW)
93         txtidnumber=Entry(labelframeleft,width=14,textvariable=self.var_id_number,font="times new roman",20,"bold"))
94         txtidnumber.grid(row=9,column=1)

95         address
96         lbbaddress=Label(labelframeleft,text="Address:",font=("times new roman",20,"bold"),padx=2,pady=6)
97         lbbaddress.grid(row=10,column=stickyW)
98         txtaddress=Text(labelframeleft,width=23, textvariable=self.var_address,font="times new roman",20,"bold"))
99         txtaddress.grid(row=10,column=1)

```



```
263     except Exception as es:
264         messagebox.showwarning("Warning","Something went wrong:[str(es)]",parent=self.root)
265         return "fetch data"
266 
267     def fetch_data(self):
268         conn=mysql.connector.connect(host="localhost",username="root",password="morni",database="student1")
269         my_cursor=conn.cursor()
270         my_cursor.execute("select * from customer")
271         rows=my_cursor.fetchall()
272         if len(rows)>0:
273             self.cust_details_table.delete(*self.cust_details_table.get_children())
274             for i in rows:
275                 self.cust_details_table.insert("",END,values=i)
276             conn.commit()
277         conn.close()
278 
279     *****get cursor*****
280     def get_cursor(self,events=""):
281         cursor = self.cust_details_table.focus()
282         content = self.cust_details_table.item(cursor['row'])
283         row_content["values"]
284 
285         self.var_ref.set(row[0]),
286         self.var_cust_name.set(row[1]),
287         self.var_mother.set(row[2]),
288         self.var_gender.set(row[3]),
289         self.var_post.set(row[4]),
290         self.var_mobile.set(row[5]),
291         self.var_nationality.set(row[6]),
292         self.var_id_proof.set(row[7]),
293         self.var_id_number.set(row[8]),
294         self.var_address.set(row[9])
295 
296     *****update*****
297     def update(self):
298         if self.var_mobile.get() == "":
299             messagebox.showerror("Error","Please enter mobile number",parent=self.root)
```

```

237
238     self.cursor.execute("select * from customer where Ref=%s", (self.var_Ref.get(),))
239     self.cursor.fetchone()
240     self.cursor.execute("update customer set name=%s,other=%s,Gender=%s,Postcode=%s,Ref=%s,Email=%s,Mobile=%s,Address=%s where Ref=%s",(
241         self.var_cust_name.get(),
242         self.var_other.get(),
243         self.var_gender.get(),
244         self.var_post.get(),
245         self.var_Ref.get(),
246         self.var_email.get(),
247         self.var_mobile.get(),
248         self.var_address.get(),
249         self.var_id_number.get(),
250         self.var_ref.get(),
251         ))
252
253     conn.commit()
254     self.fetch_data()
255     conn.close()
256     msg.showinfo("Update","Customer details has been updated successfully",parent=self.root)
257
258     self.addoption()
259     message=messagebox.askyesno("TITLE MANAGEMENT SYSTEM","Do you want delete this customer",parent=self.root)
260
261     if message:
262         conn=mysql.connector.connect(host="localhost",username="root",password="mormi",database="student")
263         my.cursor=conn.cursor()
264         query="delete from customer where Ref=%s"
265         value=(self.var_Ref.get(),)
266         my.cursor.execute(query,value)
267
268     else:
269         "Do not delete"
270         conn.rollback()
271     self.fetch_data()
272     conn.close()

```

```

221     def reset(self):
222         self.var_ref.set(""),
223         self.var_cust_name.set(""),
224         self.var_mother.set(""),
225         self.var_gender.set(""),
226         self.var_post.set(""),
227         self.var_mobile.set(""),
228         self.var_email.set(""),
229         self.var_id_proof.set(""),
230         self.var_id_number.set(""),
231         self.var_address.set("")
232
233         x=random.randint(1000,9999)
234         self.var_ref.set(str(x))
235
236         self.var_search.set("Search")
237
238     def search(self):
239         my.connector.connect(host="localhost",username="root",password="root",database="student")
240         my_cursor=my.connector.cursor()
241
242         my_cursor.execute("select * from customer where "+str(self.search_var.get())+" like '%"+str(self.txt_search.get())+"%'")
243         rows=my.cursor.fetchall()
244         if len(rows)>0:
245             self.cust_details_table.delete("self.cust_details_table.get_children()")
246             for i in rows:
247                 self.cust_details_table.insert("",END,values=i)
248             conn.commit()
249         conn.close()
250
251     if __name__ == "__main__":
252         root=Tk()
253         obj=cost(root)
254         root.mainloop()

```

Room Program:


```

1 package com.mkyong.jdbc;
2
3 import java.sql.*;
4
5 public class JdbcExample {
6     public static void main(String[] args) {
7         // Load JDBC driver
8         Class.forName("com.mysql.jdbc.Driver");
9
10        // Create connection
11        Connection conn = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test", "root", "root");
12
13        Statement stmt = conn.createStatement();
14        ResultSet rs = stmt.executeQuery("select * from user");
15
16        while(rs.next()) {
17            System.out.println("User ID: " + rs.getInt("id") + ", Name: " + rs.getString("name"));
18        }
19
20        rs.close();
21        stmt.close();
22        conn.close();
23    }
24 }

```

```

    class Mockingbird {
        def fetch_contacts():
            contacts = []
            #Initial contact
            contacts.append({
                "id": 1,
                "first_name": "John",
                "last_name": "Doe",
                "email": "johndoe@example.com",
                "phone": "+1234567890",
                "nationality": "American"
            })
            #Customer contact
            contacts.append({
                "id": 2,
                "first_name": "Jane",
                "last_name": "Doe",
                "email": "janedoe@example.com",
                "phone": "+1234567890",
                "nationality": "American"
            })
            #Employee contact
            contacts.append({
                "id": 3,
                "first_name": "Mike",
                "last_name": "Smith",
                "email": "mikesmith@example.com",
                "phone": "+1234567890",
                "nationality": "American"
            })
            return contacts
    }

```

```

    context->createCommand("SELECT
        id,
        name,
        description,
        totalValue,
        sum((q1*2) + q2*3 + q3*4 + q4*5 + q5*6 + q6*7 + q7*8 + q8*9 + q9*10 + q10*11 + q11*12 + q12*13 + q13*14 + q14*15 + q15*16 + q16*17 + q17*18 + q18*19 + q19*20 + q20*21 + q21*22 + q22*23 + q23*24 + q24*25 + q25*26 + q26*27 + q27*28 + q28*29 + q29*30 + q30*31 + q31*32 + q32*33 + q33*34 + q34*35 + q35*36 + q36*37 + q37*38 + q38*39 + q39*40 + q40*41 + q41*42 + q42*43 + q43*44 + q44*45 + q45*46 + q46*47 + q47*48 + q48*49 + q49*50 + q50*51 + q51*52 + q52*53 + q53*54 + q54*55 + q55*56 + q56*57 + q57*58 + q58*59 + q59*60 + q60*61 + q61*62 + q62*63 + q63*64 + q64*65 + q65*66 + q66*67 + q67*68 + q68*69 + q69*70 + q70*71 + q71*72 + q72*73 + q73*74 + q74*75 + q75*76 + q76*77 + q77*78 + q78*79 + q79*80 + q80*81 + q81*82 + q82*83 + q83*84 + q84*85 + q85*86 + q86*87 + q87*88 + q88*89 + q89*90 + q90*91 + q91*92 + q92*93 + q93*94 + q94*95 + q95*96 + q96*97 + q97*98 + q98*99 + q99*100 + q100*101 + q101*102 + q102*103 + q103*104 + q104*105 + q105*106 + q106*107 + q107*108 + q108*109 + q109*110 + q110*111 + q111*112 + q112*113 + q113*114 + q114*115 + q115*116 + q116*117 + q117*118 + q118*119 + q119*120 + q120*121 + q121*122 + q122*123 + q123*124 + q124*125 + q125*126 + q126*127 + q127*128 + q128*129 + q129*130 + q130*131 + q131*132 + q132*133 + q133*134 + q134*135 + q135*136 + q136*137 + q137*138 + q138*139 + q139*140 + q140*141 + q141*142 + q142*143 + q143*144 + q144*145 + q145*146 + q146*147 + q147*148 + q148*149 + q149*150 + q150*151 + q151*152 + q152*153 + q153*154 + q154*155 + q155*156 + q156*157 + q157*158 + q158*159 + q159*160 + q160*161 + q161*162 + q162*163 + q163*164 + q164*165 + q165*166 + q166*167 + q167*168 + q168*169 + q169*170 + q170*171 + q171*172 + q172*173 + q173*174 + q174*175 + q175*176 + q176*177 + q177*178 + q178*179 + q179*180 + q180*181 + q181*182 + q182*183 + q183*184 + q184*185 + q185*186 + q186*187 + q187*188 + q188*189 + q189*190 + q190*191 + q191*192 + q192*193 + q193*194 + q194*195 + q195*196 + q196*197 + q197*198 + q198*199 + q199*200 + q200*201 + q201*202 + q202*203 + q203*204 + q204*205 + q205*206 + q206*207 + q207*208 + q208*209 + q209*210 + q210*211 + q211*212 + q212*213 + q213*214 + q214*215 + q215*216 + q216*217 + q217*218 + q218*219 + q219*220 + q220*221 + q221*222 + q222*223 + q223*224 + q224*225 + q225*226 + q226*227 + q227*228 + q228*229 + q229*230 + q230*231 + q231*232 + q232*233 + q233*234 + q234*235 + q235*236 + q236*237 + q237*238 + q238*239 + q239*240 + q240*241 + q241*242 + q242*243 + q243*244 + q244*245 + q245*246 + q246*247 + q247*248 + q248*249 + q249*250 + q250*251 + q251*252 + q252*253 + q253*254 + q254*255 + q255*256 + q256*257 + q257*258 + q258*259 + q259*260 + q260*261 + q261*262 + q262*263 + q263*264 + q264*265 + q265*266 + q266*267 + q267*268 + q268*269 + q269*270 + q270*271 + q271*272 + q272*273 + q273*274 + q274*275 + q275*276 + q276*277 + q277*278 + q278*279 + q279*280 + q280*281 + q281*282 + q282*283 + q283*284 + q284*285 + q285*286 + q286*287 + q287*288 + q288*289 + q289*290 + q290*291 + q291*292 + q292*293 + q293*294 + q294*295 + q295*296 + q296*297 + q297*298 + q298*299 + q299*300 + q300*301 + q301*302 + q302*303 + q303*304 + q304*305 + q305*306 + q306*307 + q307*308 + q308*309 + q309*310 + q310*311 + q311*312 + q312*313 + q313*314 + q314*315 + q315*316 + q316*317 + q317*318 + q318*319 + q319*320 + q320*321 + q321*322 + q322*323 + q323*324 + q324*325 + q325*326 + q326*327 + q327*328 + q328*329 + q329*330 + q330*331 + q331*332 + q332*333 + q333*334 + q334*335 + q335*336 + q336*337 + q337*338 + q338*339 + q339*340 + q340*341 + q341*342 + q342*343 + q343*344 + q344*345 + q345*346 + q346*347 + q347*348 + q348*349 + q349*350 + q350*351 + q351*352 + q352*353 + q353*354 + q354*355 + q355*356 + q356*357 + q357*358 + q358*359 + q359*360 + q360*361 + q361*362 + q362*363 + q363*364 + q364*365 + q365*366 + q366*367 + q367*368 + q368*369 + q369*370 + q370*371 + q371*372 + q372*373 + q373*374 + q374*375 + q375*376 + q376*377 + q377*378 + q378*379 + q379*380 + q380*381 + q381*382 + q382*383 + q383*384 + q384*385 + q385*386 + q386*387 + q387*388 + q388*389 + q389*390 + q390*391 + q391*392 + q392*393 + q393*394 + q394*395 + q395*396 + q396*397 + q397*398 + q398*399 + q399*400 + q400*401 + q401*402 + q402*403 + q403*404 + q404*405 + q405*406 + q406*407 + q407*408 + q408*409 + q409*410 + q410*411 + q411*412 + q412*413 + q413*414 + q414*415 + q415*416 + q416*417 + q417*418 + q418*419 + q419*420 + q420*421 + q421*422 + q422*423 + q423*424 + q424*425 + q425*426 + q426*427 + q427*428 + q428*429 + q429*430 + q430*431 + q431*432 + q432*433 + q433*434 + q434*435 + q435*436 + q436*437 + q437*438 + q438*439 + q439*440 + q440*441 + q441*442 + q442*443 + q443*444 + q444*445 + q445*446 + q446*447 + q447*448 + q448*449 + q449*450 + q450*451 + q451*452 + q452*453 + q453*454 + q454*455 + q455*456 + q456*457 + q457*458 + q458*459 + q459*460 + q460*461 + q461*462 + q462*463 + q463*464 + q464*465 + q465*466 + q466*467 + q467*468 + q468*469 + q469*470 + q470*471 + q471*472 + q472*473 + q473*474 + q474*475 + q475*476 + q476*477 + q477*478 + q478*479 + q479*480 + q480*481 + q481*482 + q482*483 + q483*484 + q484*485 + q485*486 + q486*487 + q487*488 + q488*489 + q489*490 + q490*491 + q491*492 + q492*493 + q493*494 + q494*495 + q495*496 + q496*497 + q497*498 + q498*499 + q499*500 + q500*501 + q501*502 + q502*503 + q503*504 + q504*505 + q505*506 + q506*507 + q507*508 + q508*509 + q509*510 + q510*511 + q511*512 + q512*513 + q513*514 + q514*515 + q515*516 + q516*517 + q517*518 + q518*519 + q519*520 + q520*521 + q521*522 + q522*523 + q523*524 + q524*525 + q525*526 + q526*527 + q527*528 + q528*529 + q529*530 + q530*531 + q531*532 + q532*533 + q533*534 + q534*535 + q535*536 + q536*537 + q537*538 + q538*539 + q539*540 + q540*541 + q541*542 + q542*543 + q543*544 + q544*545 + q545*546 + q546*547 + q547*548 + q548*549 + q549*550 + q550*551 + q551*552 + q552*553 + q553*554 + q554*555 + q555*556 + q556*557 + q557*558 + q558*559 + q559*560 + q560*561 + q561*562 + q562*563 + q563*564 + q564*565 + q565*566 + q566*567 + q567*568 + q568*569 + q569*570 + q570*571 + q571*572 + q572*573 + q573*574 + q574*575 + q575*576 + q576*577 + q577*578 + q578*579 + q579*580 + q580*581 + q581*582 + q582*583 + q583*584 + q584*585 + q585*586 + q586*587 + q587*588 + q588*589 + q589*590 + q590*591 + q591*592 + q592*593 + q593*594 + q594*595 + q595*596 + q596*597 + q597*598 + q598*599 + q599*600 + q600*601 + q601*602 + q602*603 + q603*604 + q604*605 + q605*606 + q606*607 + q607*608 + q608*609 + q609*610 + q610*611 + q611*612 + q612*613 + q613*614 + q614*615 + q615*616 + q616*617 + q617*618 + q618*619 + q619*620 + q620*621 + q621*622 + q622*623 + q623*624 + q624*625 + q625*626 + q626*627 + q627*628 + q628*629 + q629*630 + q630*631 + q631*632 + q632*633 + q633*634 + q634*635 + q635*636 + q636*637 + q637*638 + q638*639 + q639*640 + q640*641 + q641*642 + q642*643 + q643*644 + q644*645 + q645*646 + q646*647 + q647*648 + q648*649 + q649*650 + q650*651 + q651*652 + q652*653 + q653*654 + q654*655 + q655*656 + q656*657 + q657*658 + q658*659 + q659*660 + q660*661 + q661*662 + q662*663 + q663*664 + q664*665 + q665*666 + q666*667 + q667*668 + q668*669 + q669*670 + q670*671 + q671*672 + q672*673 + q673*674 + q674*675 + q675*676 + q676*677 + q677*678 + q678*679 + q679*680 + q680*681 + q681*682 + q682*683 + q683*684 + q684*685 + q685*686 + q686*687 + q687*688 + q688*689 + q689*690 + q690*691 + q691*692 + q692*693 + q693*694 + q694*695 + q695*696 + q696*697 + q697*698 + q698*699 + q699*700 + q700*701 + q701*702 + q702*703 + q703*704 + q704*705 + q705*706 + q706*707 + q707*708 + q708*709 + q709*710 + q710*711 + q711*712 + q712*713 + q713*714 + q714*715 + q715*716 + q716*717 + q717*718 + q718*719 + q719*720 + q720*721 + q721*722 + q722*723 + q723*724 + q724*725 + q725*726 + q726*727 + q727*728 + q728*729 + q729*730 + q730*731 + q731*732 + q732*733 + q733*734 + q734*735 + q735*736 + q736*737 + q737*738 + q738*739 + q739*740 + q740*741 + q741*742 + q742*743 + q743*744 + q744*745 + q745*746 + q746*747 + q747*748 + q748*749 + q749*750 + q750*751 + q751*752 + q752*753 + q753*754 + q754*755 + q755*756 + q756*757 + q757*758 + q758*759 + q759*760 + q760*761 + q761*762 + q762*763 + q763*764 + q764*765 + q765*766 + q766*767 + q767*768 + q768*769 + q769*770 + q770*771 + q771*772 + q772*773 + q773*774 + q774*775 + q775*776 + q776*777 + q777*778 + q778*779 + q779*780 + q780*781 + q781*782 + q782*783 + q783*784 + q784*785 + q785*786 + q786*787 + q787*788 + q788*789 + q789*790 + q790*791 + q791*792 + q792*793 + q793*794 + q794*795 + q795*796 + q796*797 + q797*798 + q798*799 + q799*7100 + q7100*7101 + q7101*7102 + q7102*7103 + q7103*7104 + q7104*7105 + q7105*7106 + q7106*7107 + q7107*7108 + q7108*7109 + q7109*7110 + q7110*7111 + q7111*7112 + q7112*7113 + q7113*7114 + q7114*7115 + q7115*7116 + q7116*7117 + q7117*7118 + q7118*7119 + q7119*7120 + q7120*7121 + q7121*7122 + q7122*7123 + q7123*7124 + q7124*7125 + q7125*7126 + q7126*7127 + q7127*7128 + q7128*7129 + q7129*7130 + q7130*7131 + q7131*7132 + q7132*7133 + q7133*7134 + q7134*7135 + q7135*7136 + q7136*7137 + q7137*7138 + q7138*7139 + q7139*7140 + q7140*7141 + q7141*7142 + q7142*7143 + q7143*7144 + q7144*7145 + q7145*7146 + q7146*7147 + q7147*7148 + q7148*7149 + q7149*7150 + q7150*7151 + q7151*7152 + q7152*7153 + q7153*7154 + q7154*7155 + q7155*7156 + q7156*7157 + q7157*7158 + q7158*7159 + q7159*7160 + q7160*7161 + q7161*7162 + q7162*7163 + q7163*7164 + q7164*7165 + q7165*7166 + q7166*7167 + q7167*7168 + q7168*7169 + q7169*7170 + q7170*7171 + q7171*7172 + q7172*7173 + q7173*7174 + q7174*7175 + q7175*7176 + q7176*7177 + q7177*7178 + q7178*7179 + q7179*7180 + q7180*7181 + q7181*7182 + q7182*7183 + q7183*7184 + q7184*7185 + q7185*7186 + q7186*7187 + q7187*7188 + q7188*7189 + q7189*7190 + q7190*7191 + q7191*7192 + q7192*7193 + q7193*7194 + q7194*7195 + q7195*7196 + q7196*7197 + q7197*7198 + q7198*7199 + q7199*71100 + q71100*71101 + q71101*71102 + q71102*71103 + q71103*71104 + q71104*71105 + q71105*71106 + q71106*71107 + q71107*71108 + q71108*71109 + q71109*71110 + q71110*71111 + q71111*71112 + q71112*71113 + q71113*71114 + q71114*71115 + q71115*71116 + q71116*71117 + q71117*71118 + q71118*71119 + q71119*71120 + q71120*71121 + q71121*71122 + q71122*71123 + q71123*71124 + q71124*71125 + q71125*71126 + q71126*71127 + q71127*71128 + q71128*71129 + q71129*71130 + q71130*71131 + q71131*71132 + q71132*71133 + q71133*71134 + q71134*71135 + q71135*71136 + q71136*71137 + q71137*71138 + q71138*71139 + q71139*71140 + q71140*71141 + q71141*71142 + q71142*71143 + q71143*71144 + q71144*71145 + q71145*71146 + q71146*71147 + q71147*71148 + q71148*71149 + q71149*71150 + q71150*71151 + q71151*71152 + q71152*71153 + q71153*71154 + q71154*71155 + q71155*71156 + q71156*71157 + q71157*71158 + q71158*71159 + q71159*71160 + q71160*71161 + q71161*71162 + q71162*71163 + q71163*71164 + q71164*71165 + q71165*71166 + q71166*71167 + q71167*71168 + q71168*71169 + q71169*71170 + q71170*71171 + q71171*71172 + q71172*71173 + q71173*71174 + q71174*71175 + q71175*71176 + q71176*71177 + q71177*71178 + q71178*71179 + q71179*71180 + q71180*71181 + q71181*71182 + q71182*71183 + q71183*71184 + q71184*71185 + q71185*71186 + q71186*71187 + q71187*71188 + q71188*71189 + q71189*71190 + q71190*71191 + q71191*71192 + q71192*71193 + q71193*71194 + q71194*71195 + q71195*71196 + q71196*71197 + q71197*71198 + q71198*71199 + q71199*71200 + q71200*71201 + q71201*71202 + q71202*71203 + q71203*71204 + q71204*71205 + q71205*71206 + q71206*71207 + q71207*71208 + q71208*71209 + q71209*71210 + q71210*71211 + q71211*71212 + q71212*71213 + q71213*71214 + q71214*71215 + q71215*71216 + q71216*71217 + q71217*71218 + q71218*71219 + q71219*71220 + q71220*71221 + q71221*71222 + q71222*71223 + q71223*71224 + q71224*71225 + q71225*71226 + q71226*71227 + q71227*71228 + q71228*71229 + q71229*71230 + q71230*71231 + q71231*71232 + q71232*71233 + q71233*71234 + q71234*71235 + q71235*71236 + q71236*71237 + q71237*71238 + q71238*71239 + q71239*71240 + q71240*71241 + q71241*71242 + q71242*71243 + q71243*71244 + q71244*71245 + q71245*71246 + q71246*71247 + q71247*71248 + q71248*71249 + q71249*71250 + q71250*71251 + q71251*71252 + q71252*71253 + q71253*71254 + q71254*71255 + q71255*71256 + q71256*71257 + q71257*71258 + q71258*71259 + q71259*71260 + q71260*71261 + q71261*71262 + q71262*71263 + q71263*71264 + q71264*71265 + q71265*71266 + q71266*71267 + q71267*71268 + q71268*71269 + q71269*71270 + q71270*71271 + q71271*71272 + q71272*71273 + q71273*71274 + q71274*71275 + q71275*71276 + q71276*71277 + q71277*71278 + q71278*71279 + q71279*71280 + q71280*71281 + q71281*71282 + q71282*71283 + q71283*71284 + q71284*71285 + q71285*71286 + q71286*71287 + q71287*71288 + q71288*71289 + q71289*71290 + q71290*71291 + q71291*71292 + q71292*71293 + q71293*71294 + q71294*71295 + q71295*71296 + q71296*71297 + q71297*71298 + q71298*71299 + q71299*71300 + q71300*71301 + q71301*71302 + q71302*71303 + q71303*71304 + q71304*71305 + q71305*71306 + q71306*71307 + q71307*71308 + q71308*71309 + q71309*71310 + q71310*71311 + q71311*71312 + q71312*71313 + q71313*71314 + q71314*71315 + q71315*71316 + q71316*71317 + q71317*71318 + q71318*71319 + q71319*71320 + q71320*71321 + q71321*71322 + q71322*71323 + q71323*71324 + q71324*71325 + q71325*71326 + q71326*71327 + q71327*71328 + q71328*71329 + q71329*71330 + q71330*71331 + q71331*71332 + q71332*71333 + q71333*71334 + q71334*71335 + q71335*71336 + q71336*71337 + q71337*71338 + q71338*71339 + q71339*71340 + q71340*71341 + q71341*71342 + q71342*71343 + q71343*71344 + q71344*71345 + q71345*71346 + q71346*71347 + q71347*71348 + q71348*71349 + q71349*71350 + q71350*71351 + q71351*71352 + q71352*71353 + q71353*71354 + q71354*71355 + q71355*71356 + q71356*71357 + q71357*71358 + q71358*71359 + q71359*71360 + q71360*71361 + q71361*71362 + q71362*71363 + q71363*71364 + q71364*71365 + q71365*71366 + q71366*71367 + q71367*71368 + q71368*71369 + q71369*71370 + q71370*71371 + q71371*71372 + q71372*71373 + q71373*71374 + q71374*71375 + q71375*71376 + q71376*71377 + q71377*71378 + q71378*71379 + q71379*71380 + q71380*71381 + q71381*71382 + q71382*71383 + q71383*71384 + q71384*71385 + q71385*71386 + q71386*71387 + q71387*71388 + q71388*71389 + q71389*71390 + q71390*71391 + q71391*71392 + q71392*71393 + q71393*71394 + q71394*71395 + q71395*71396 + q71396*71397 + q71397*71398 + q71398*71399 + q71399*71400 + q71400*71401 + q71401*71402 + q71402*71403 + q71403*71404 + q71404*71405 + q71405*71406 + q71406*71407 + q71407*71408 + q71408*71409 + q71409*71410 + q71410*71411 + q71411*71412 + q71412*71413 + q71413*71414 + q71414*71415 + q71415*71416 + q71416*71417 + q71417*71418 + q71418*71419 + q71419*71420 + q71420*71421 + q71421*71422 + q71422*71423 + q71423*71424 + q71424*71425 + q71425*71426 + q71426*71427 +
```

```
    class rebookings {
        int totalSelf();
        void var_update_set(Taxi);
        void var_actualtotal_set(Taxi);
        void var_total_set(Taxi);

        void var_update_get(){"last" and self.var_createtype.get() == "double"};
        void var_actualtotal_get(){"last" and self.var_createtype.get() == "double"};
        void var_total_get(){"last" and self.var_createtype.get() == "double"};

        void var_update_set(0);
        void var_actualtotal_set(0);
        void var_total_set(0);

        void var_update_get(){"sum" and self.var_createtype.get() == "double"};
        void var_actualtotal_get(){"sum" and self.var_createtype.get() == "double"};
        void var_total_get(){"sum" and self.var_createtype.get() == "double"};

        void var_update_set(0);
        void var_actualtotal_set(0);
        void var_total_set(0);

        void var_update_get(){"lastFirst" and self.var_createtype.get() == "array"};
        void var_actualtotal_get(){"lastFirst" and self.var_createtype.get() == "array"};
        void var_total_get(){"lastFirst" and self.var_createtype.get() == "array"};
```

Details:

```
scroll_y.pack(side=RIGHT,fill=Y)
scroll_x.config(command=self.room_table.xview)
scroll_y.config(command=self.room_table.yview)

self.room_table.heading('Floor',text='Floor')
self.room_table.heading('Roomno',text='Roomno')
self.room_table.heading('Roomtype',text='Roomtype')

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

self.room_table.column('Floor',width=200)
self.room_table.column('Roomno',width=200)
self.room_table.column('Roomtype',width=200)
self.room_table.bind("buttonRelease-1",self.get_cursor)
self.room_table.pack(fill=BOTH,expand=1)
self.fetch_data()

*****Add function*****
def add_data(self):
    if self.var_floor.get() == "" or self.var_roomtype.get() == "":
        messagebox.showerror("Error","All fields are required",parent=self.root)
    else:
        try:
            conn=mysql.connector.connect(host="localhost",username="host",password="host",database="student1")
            my_cursor=conn.cursor()
            my_cursor.execute("insert into details values(%s,%s,%s) ")
            self.var_floor.get(),
            self.var_roomno.get(),
            self.var_roomtype.get()

            conn.commit()
            self.fetch_data()
            conn.close()
            messagebox.showinfo("Success","New Room Added Successfully",parent=self.root)
        except Exception as es:
            messagebox.showwarning("Warning","Something went wrong:",str(es)),parent=self.root)
```

```

1 def __init__(self):
2     self.fetch_data()
3
4     self.con=mysql.connector.connect(host="localhost",username="host",password="moni",database="student")
5     self.cur=con.cursor()
6
7     self.cur.execute("select * from details")
8     rows=self.cur.fetchall()
9
10    if len(rows)>0:
11        self.room_table.delete("self.room_table.get_children()")
12        for row in rows:
13            self.room_table.insert("",0,values=row)
14
15    con.commit()
16    con.close()
17
18    #-----get room-----#
19
20    def get_cursor(self,events=""):
21        cursor=self.room_table.focus()
22        content=self.room_table.item(cursor,"values")
23
24        self.var_floor.set(content[0])
25        self.var_roomno.set(content[1])
26        self.var_RoomType.set(content[2])
27
28    #-----update room-----#
29
30    def update(self):
31        if self.var_floor.get()!="":
32            messagebox.showerror("Error","Please enter floor number",parent=self.root)
33        else:
34            con=mysql.connector.connect(host="localhost",username="host",password="moni",database="student")
35            self.cur=con.cursor()
36            self.cur.execute("update details set Floor=%s,RoomType=%s where RoomNo=%s",
37                            (self.var_floor.get(),
38                             self.var_RoomType.get(),
39                             self.var_roomno.get()))
40
41            con.commit()
42            self.fetch_data()

```

```

74     """Delete function"""
75     def delete(self):
76         delete=messagebox.askyesno("HOTEL MANAGEMENT SYSTEM","Do you want delete this Room Details",parent=self.root)
77         if delete==True:
78             conn=mysql.connector.connect(host="localhost",username="root",password="moni",database="student1")
79             my_cursor=conn.cursor()
80             query="delete from details where RoomNo=%s"
81             value=(self.var_roomno.get(),)
82             my_cursor.execute(query,value)
83         else:
84             if not delete:
85                 return
86             conn.commit()
87             self.fetch_data()
88             conn.close()
89
90     """Reset function"""
91     def reset_data(self):
92         self.var_floor.set("")
93         self.var_roomno.set("")
94         self.var_roomtype.set("")
95
96     if __name__ == "__main__":
97         root=Tk()
98         obj=detailsroom(root)
99         root.mainloop()

```

System Architecture:

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



L

L

L

L

L

L

L

L