





Mysql:

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `harshinee`

--

-- --------------------------------------------------------

--

-- Table structure for table `booking`

--

CREATE TABLE `booking` (

`Booking\_ID` int(4) NOT NULL,

`Booking\_Date` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`Room\_No` int(4) NOT NULL,

`Room\_Type` varchar(20) NOT NULL,

`Food\_ID` int(4) NOT NULL,

`Food\_Type` varchar(20) NOT NULL,

`Room\_Charges` decimal(10,0) NOT NULL,

`days` int(2) NOT NULL,

`amount` decimal(10,0) NOT NULL,

`Food\_Charges` decimal(10,0) NOT NULL,

`Grand\_Total` decimal(10,0) NOT NULL,

`UserName` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `booking`

--

INSERT INTO `booking` (`Booking\_ID`, `Booking\_Date`, `Room\_No`, `Room\_Type`, `Food\_ID`, `Food\_Type`, `Room\_Charges`, `days`, `amount`, `Food\_Charges`, `Grand\_Total`, `UserName`) VALUES

(1, '2019-08-28 10:51:50', 105, 'SUPER-DELUXE', 1, 'Veg', '7000', 2, '14000', '1200', '15200', 'kavya'),

(2, '2019-08-28 10:53:38', 102, 'AC', 3, 'Vegan', '4000', 5, '20000', '1800', '21800', 'tanu'),

(5, '2019-09-07 03:10:46', 103, 'DELUXE', 4, 'Beverages', '5000', 1, '5000', '2000', '7000', 'naman');

-- --------------------------------------------------------

--

-- Table structure for table `customers`

--

CREATE TABLE `customers` (

`id` int(4) NOT NULL,

`name` varchar(30) NOT NULL,

`username` varchar(20) NOT NULL,

`password` varchar(20) NOT NULL,

`mobile` varchar(10) NOT NULL,

`email` varchar(50) NOT NULL,

`address` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `customers`

--

INSERT INTO `customers` (`id`, `name`, `username`, `password`, `mobile`, `email`, `address`) VALUES

(1, 'Kavya', 'kavya', '12345', '9876543210', 'kb@gmail.com', 'Vashi'),

(8, 'Tanu', 'tanu', '12345', '1234567890', 'tanu@gmail.com', 'New Panvel'),

(9, 'Naman', 'naman', '12345', '9182736450', 'nb@gmail.com', 'Vashi');

-- --------------------------------------------------------

--

-- Table structure for table `employee`

--

CREATE TABLE `employee` (

`emp\_id` int(4) NOT NULL,

`Emp\_name` varchar(30) NOT NULL,

`desig` varchar(30) NOT NULL,

`username` varchar(20) NOT NULL,

`password` varchar(20) NOT NULL,

`mobile` varchar(10) NOT NULL,

`email` varchar(50) NOT NULL,

`address` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `employee`

--

INSERT INTO `employee` (`emp\_id`, `Emp\_name`, `desig`, `username`, `password`, `mobile`, `email`, `address`) VALUES

(111, 'Kavya Bansal', 'Manager', 'admin', '12345', '7208808034', 'kavya@gmail.com', 'New Panvel'),

(112, 'Tanu', 'Room Service', 'tanu', '12345', '9594927299', 'tanu@gmail.com', 'Panvel');

-- --------------------------------------------------------

--

-- Table structure for table `food`

--

CREATE TABLE `food` (

`food\_id` int(3) NOT NULL,

`Food\_Type` varchar(40) NOT NULL,

`Charges` decimal(10,0) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `food`

--

INSERT INTO `food` (`food\_id`, `Food\_Type`, `Charges`) VALUES

(1, 'chinese', '1200'),

(2, 'continental', '1500'),

(3, 'french', '1800'),

(4, 'italian', '1800'),

(5, 'indian', '1550'),

(6, 'mexican', '1500'),

(7, 'fusion', '3000');

-- --------------------------------------------------------

--

-- Table structure for table `room`

--

CREATE TABLE `room` (

`Room\_No` int(3) NOT NULL,

`Room\_Type` varchar(20) NOT NULL,

`Charges` decimal(10,0) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `room`

--

INSERT INTO `room` (`Room\_No`, `Room\_Type`, `Charges`) VALUES

(101, 'AC', '4000'),

(102, 'AC', '4000'),

(103, 'DELUXE', '5000'),

(104, 'DELUXE', '5000'),

(105, 'SUPER-DELUXE', '7000'),

(106, 'SUPER-DELUXE', '7000');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `booking`

--

ALTER TABLE `booking`

ADD PRIMARY KEY (`Booking\_ID`);

--

-- Indexes for table `customers`

--

ALTER TABLE `customers`

ADD PRIMARY KEY (`id`);

--

-- Indexes for table `food`

--

ALTER TABLE `food`

ADD PRIMARY KEY (`food\_id`);

--

-- Indexes for table `room`

--

ALTER TABLE `room`

ADD PRIMARY KEY (`Room\_No`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `booking`

--

ALTER TABLE `booking`

MODIFY `Booking\_ID` int(4) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

--

-- AUTO\_INCREMENT for table `customers`

--

ALTER TABLE `customers`

MODIFY `id` int(4) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=10;

--

-- AUTO\_INCREMENT for table `food`

--

ALTER TABLE `food`

MODIFY `food\_id` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;

--

-- AUTO\_INCREMENT for table `room`

--

ALTER TABLE `room`

MODIFY `Room\_No` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=107;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

Python :

#import modules

import mysql.connector

from tkinter import messagebox

from tkinter import \*

import os

# Designing window for registration

def view():

pass

def bookings():

pass

# Designing Main(first) window

def main\_menu():

global main\_menu

main\_menu = Tk()

main\_menu.geometry("600x500")

main\_menu.title("Radisson Blu")

Label(text=" \* \* \* W E L C O M E \* \* \* ", bg="cyan", width="300", height="2", font=("Bell MT", 18)).pack()

Label(text="").pack()

canvas = Canvas(main\_menu, width = 250, height = 200)

canvas.pack()

img = PhotoImage(file="logo.png")

canvas.create\_image(20,20, anchor=NW, image=img)

Label(text="").pack()

Button(text="CUSTOMER", height="2", width="30", bg="cyan",command = customer).pack()

Label(text="").pack()

Button(text="EMPLOYEE", height="2", width="30", bg="cyan",command=staff\_login).pack()

Label(text="").pack()

Button(text="EXIT", height="2", width="30", bg="cyan",command=ex6).pack()

main\_menu.mainloop()

def ex6():

main\_menu.destroy()

#customer

def customer():

global customer\_screen

customer\_screen = Toplevel(main\_menu)

#main\_screen = Tk()

customer\_screen.geometry("600x500")

customer\_screen.title("C U S T O M E R")

Label(customer\_screen,text="").pack()

canvas = Canvas(customer\_screen, width = 250, height = 200)

canvas.pack()

img = PhotoImage(file="customer.png")

canvas.create\_image(20,20, anchor=NW, image=img)

Label(customer\_screen,text="Great offers & packages when you book direct.", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(customer\_screen,text="").pack()

Button(customer\_screen,text="Login", height="2", width="30", bg="cyan",command = login).pack()

Label(customer\_screen,text="").pack()

Button(customer\_screen,text="Register", height="2", width="30", bg="cyan",command=register).pack()

Label(customer\_screen,text="").pack()

Button(customer\_screen,text="Exit", height="2", width="30", bg="cyan",command=ex7).pack()

customer\_screen.mainloop()

def ex7():

customer\_screen.destroy()

#staff login

def staff\_login():

global staff\_login\_screen

global susername\_verify

global spassword\_verify

susername\_verify = StringVar()

spassword\_verify = StringVar()

global susername\_login\_entry

global spassword\_login\_entry

staff\_login\_screen = Toplevel(main\_menu)

staff\_login\_screen.geometry("600x400")

staff\_login\_screen.title("Employee Login")

Label(staff\_login\_screen, text="Please enter details below to login",bg="cyan").pack()

#img1 = PhotoImage(file="book.png")

Label(staff\_login\_screen, text="").pack()

#canvas1 = Canvas(staff\_login\_screen, width = 250, height = 200)

#canvas1.pack()

#img1 = PhotoImage(file="book.png")

#canvas1.create\_image(20,20, anchor=NW, image=img1)

Label(staff\_login\_screen, text="Username \* ").pack()

susername\_login\_entry = Entry(staff\_login\_screen, textvariable=susername\_verify)

susername\_login\_entry.pack()

Label(staff\_login\_screen, text="").pack()

Label(staff\_login\_screen, text="Password \* ").pack()

spassword\_login\_entry = Entry(staff\_login\_screen, textvariable=spassword\_verify, show= '\*')

spassword\_login\_entry.pack()

Label(staff\_login\_screen, text="").pack()

Button(staff\_login\_screen, text="Login", width=10, height=1, bg="cyan",command = slogin\_verify).pack()

Label(staff\_login\_screen, text="").pack()

Button(staff\_login\_screen,text="Cancel", width=10, height=1, bg="cyan", command=ex5).pack()

def ex5():

staff\_login\_screen.destroy()

def staff():

global staff\_screen

#staff\_screen = Tk()

staff\_screen = Toplevel(main\_menu)

staff\_screen.geometry("600x600")

staff\_screen.title("E M P L O Y E E")

Label(staff\_screen,text="LOGIN SUCCESSFUL").pack()

canvas = Canvas(staff\_screen, width = 250, height = 200)

canvas.pack()

img = PhotoImage(file="staff.png")

canvas.create\_image(20,20, anchor=NW, image=img)

Label(staff\_screen,text="Great offers & packages when you book direct.", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(staff\_screen,text="").pack()

Button(staff\_screen,text="View Customer", height="2", width="30", command=viewc).pack()

Label(staff\_screen,text="").pack()

Button(staff\_screen,text="View Bookings", height="2", width="30", command=viewb).pack()

Label(staff\_screen,text="").pack()

Button(staff\_screen,text="Cancel Bookings", height="2", width="30", command=cancel\_booking).pack()

Label(staff\_screen,text="").pack()

#Button(staff\_screen,text="Booking", height="2", width="30", command=Booking1).pack()

#Label(staff\_screen,text="").pack()

Button(staff\_screen,text="Exit", height="2", width="30", command=ex3).pack()

staff\_screen.mainloop()

def ex3():

staff\_screen.destroy()

def Booking1():

pass

def register():

global register\_screen

register\_screen = Toplevel(customer\_screen)

register\_screen.geometry("600x500")

register\_screen.title("R E G I S T E R")

Label(register\_screen,text="").pack()

global username

global password

global mobile

global name

global email

global address

global username\_entry

global password\_entry

global name\_entry

global mobile\_entry

global address\_entry

global email\_entry

username = StringVar()

password = StringVar()

mobile = StringVar()

name = StringVar()

address = StringVar()

email = StringVar()

Label(register\_screen, text="Please enter details below", bg="cyan").pack()

Label(register\_screen, text="").pack()

name\_lable = Label(register\_screen, text="Name \* ")

name\_lable.pack()

name\_entry = Entry(register\_screen, textvariable=name)

name\_entry.pack()

username\_lable = Label(register\_screen, text="Username \* ")

username\_lable.pack()

username\_entry = Entry(register\_screen, textvariable=username)

username\_entry.pack()

password\_lable = Label(register\_screen, text="Password \* ")

password\_lable.pack()

password\_entry = Entry(register\_screen, textvariable=password, show='\*')

password\_entry.pack()

mobile\_lable = Label(register\_screen, text="Mobile \* ")

mobile\_lable.pack()

mobile\_entry = Entry(register\_screen, textvariable=mobile)

mobile\_entry.pack()

email\_lable = Label(register\_screen, text="Email \* ")

email\_lable.pack()

email\_entry = Entry(register\_screen, textvariable=email)

email\_entry.pack()

address\_lable = Label(register\_screen, text="Address \* ")

address\_lable.pack()

address\_entry = Entry(register\_screen, textvariable=address)

address\_entry.pack()

Label(register\_screen, text="").pack()

Button(register\_screen, text="Register", width=10, height=1, bg="cyan", command = register\_user).pack()

Label(register\_screen, text="").pack()

Button(register\_screen, text="Cancel", width=10, height=1, bg="cyan", command = Ex8).pack()

def Ex8():

register\_screen.destroy()

# Designing window for login

def login():

global login\_screen

global username\_verify

global password\_verify

username\_verify = StringVar()

password\_verify = StringVar()

global username\_login\_entry

global password\_login\_entry

login\_screen = Toplevel(customer\_screen)

login\_screen.title("Customer Login")

login\_screen.geometry("300x350")

Label(login\_screen, text="Please enter details below to login",bg="cyan").pack()

Label(login\_screen, text="").pack()

Label(login\_screen, text="Username \* ").pack()

username\_login\_entry = Entry(login\_screen, textvariable=username\_verify)

username\_login\_entry.pack()

Label(login\_screen, text="").pack()

Label(login\_screen, text="Password \* ").pack()

password\_login\_entry = Entry(login\_screen, textvariable=password\_verify, show= '\*')

password\_login\_entry.pack()

Label(login\_screen, text="").pack()

Button(login\_screen, text="Login", width=10, height=1, bg="cyan",command = login\_verify).pack()

Label(login\_screen, text="").pack()

Button(login\_screen,text="Cancel", width=10, height=1, bg="cyan", command=ex4).pack()

def ex4():

login\_screen.destroy()

# Implementing event on register button

def register\_user():

try:

username\_info = username.get()

password\_info = password.get()

mobile\_info = mobile.get()

name\_info = name.get()

address\_info = address.get()

email\_info = email.get()

if username\_info=="" :

Label(register\_screen, text="User Name cannot be empty", fg="red", font=("calibri", 11)).pack()

#messagebox.showinfo("Registration ", "User Name cannot be empty")

elif password\_info=="" :

Label(register\_screen, text="Password Name cannot be empty", fg="red", font=("calibri", 11)).pack()

elif mobile\_info=="" :

Label(register\_screen, text="Mobile Name cannot be empty", fg="red", font=("calibri", 11)).pack()

elif name\_info=="" :

Label(register\_screen, text="Name Name cannot be empty", fg="red", font=("calibri", 11)).pack()

elif address\_info=="" :

Label(register\_screen, text="Address Name cannot be empty", fg="red", font=("calibri", 11)).pack()

elif email\_info=="" :

Label(register\_screen, text="Email Name cannot be empty", fg="red", font=("calibri", 11)).pack()

else:

mycursor = mydb.cursor()

sql = "INSERT INTO customers (name, username,password,mobile,email,address) VALUES (%s, %s,%s,%s,%s,%s)"

val = (name\_info,username\_info,password\_info,mobile\_info,email\_info,address\_info)

r=mycursor.execute(sql, val)

mydb.commit()

username\_entry.delete(0, END)

password\_entry.delete(0, END)

mobile\_entry.delete(0, END)

name\_entry.delete(0, END)

address\_entry.delete(0, END)

email\_entry.delete(0, END)

#//print(r)

Label(register\_screen, text="Registration Success", fg="green", font=("calibri", 11)).pack()

Label(register\_screen, text="").pack()

Button(register\_screen, text="Login", width=10, height=1, bg="cyan",command = login).pack()

except ConnectionError:

Label(register\_screen, text="Failed to insert record", fg="green", font=("calibri", 11)).pack()

# print("Failed to insert record into Laptop table {}".format(error))

# Implementing event on login button

def slogin\_verify():

username1 = susername\_verify.get()

password1 = spassword\_verify.get()

if username1=="" :

Label(login\_screen, text="UserName cannot be empty", fg="red", font=("calibri", 11)).pack()

#messagebox.showinfo("LOGIN ", "UserName cannot be empty")

elif password1=="" :

Label(login\_screen, text="Password cannot be empty", fg="red", font=("calibri", 11)).pack()

else:

susername\_login\_entry.delete(0, END)

spassword\_login\_entry.delete(0, END)

mycursor = mydb.cursor()

sql = "select \* from employee where username='"+username1+"' and password='"+password1+"'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

#for x in myresult:

if count>0:

staff()

else:

spassword\_not\_recognised()

#else:

# user\_not\_found()

def login\_verify():

global u

global p

username1 = username\_verify.get()

password1 = password\_verify.get()

if username1=="" :

Label(login\_screen, text="UserName cannot be empty", fg="red", font=("calibri", 11)).pack()

#messagebox.showinfo("LOGIN ", "UserName cannot be empty")

elif password1=="" :

Label(login\_screen, text="Password cannot be empty", fg="red", font=("calibri", 11)).pack()

else:

username\_login\_entry.delete(0, END)

password\_login\_entry.delete(0, END)

mycursor = mydb.cursor()

sql = "select \* from customers where username='"+username1+"' and password='"+password1+"'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

#for x in myresult:

if count>0:

u=username1

p=password1

Customer\_Menu()

else:

password\_not\_recognised()

#else:

# user\_not\_found()

# Designing popup for login success

def login\_sucess():

global login\_success\_screen

login\_success\_screen = Toplevel(login\_screen)

login\_success\_screen.title("Success")

login\_success\_screen.geometry("250x100")

Label(login\_success\_screen, text="Login Success").pack()

Button(login\_success\_screen, text="Booking", command=Booking).pack()

# Designing popup for login invalid password

def spassword\_not\_recognised():

global spassword\_not\_recog\_screen

spassword\_not\_recog\_screen = Toplevel(staff\_login\_screen)

spassword\_not\_recog\_screen.title("LOGIN FAILED")

spassword\_not\_recog\_screen.geometry("250x100")

Label(spassword\_not\_recog\_screen, text="Invalid Password ").pack()

Button(spassword\_not\_recog\_screen, text="OK", command=sdelete\_password\_not\_recognised).pack()

def password\_not\_recognised():

global password\_not\_recog\_screen

password\_not\_recog\_screen = Toplevel(login\_screen)

password\_not\_recog\_screen.title("LOGIN FAILED")

password\_not\_recog\_screen.geometry("250x100")

Label(password\_not\_recog\_screen, text="Invalid Password ").pack()

Button(password\_not\_recog\_screen, text="OK", command=delete\_password\_not\_recognised).pack()

# Designing popup for user not found

def user\_not\_found():

global user\_not\_found\_screen

user\_not\_found\_screen = Toplevel(login\_screen)

user\_not\_found\_screen.title("Success")

user\_not\_found\_screen.geometry("250x100")

Label(user\_not\_found\_screen, text="User Not Found").pack()

Button(user\_not\_found\_screen, text="OK", command=delete\_user\_not\_found\_screen).pack()

def viewc():

global view\_customer

view\_customer = Toplevel(staff\_screen)

view\_customer.title("View All Customers")

view\_customer.geometry("450x450")

mycursor = mydb.cursor()

sql = "select \* from Customers"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

Lb1 = Listbox(view\_customer,bg="cyan", width="450", height="10")

Lb1.pack()

#Lb1.insert(END,mycursor.fetchall())# "ID =>Name =>Mobile =>Email =>Address")

i=1

if count>0:

for x in myresult:

#st=str(x[0])+" => "+str(x[1])+" => "+ str(x[4])+" => "+str(x[5])+" => "+str(x[6])

Lb1.insert(i,x)

i=i+1

else:

print('Customer not found')

Button(view\_customer, text="Exit", command=ex2).pack()

def ex2():

view\_customer.destroy()

def viewb():

global view\_booking

view\_booking = Toplevel(staff\_screen)

view\_booking.title("View All Bookings")

view\_booking.geometry("450x450")

mycursor = mydb.cursor()

sql = "select \* from Booking"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

Lb1 = Listbox(view\_booking,bg="cyan", width="450", height="10")

#Lb1.insert(1, "ID => Date => Room\_ID=>Room\_Type=>Food\_Id=>Food\_Type=>Room Charges=>Food\_Charges=>Grand\_Total=>UserName")

i=1

if count>0:

for x in myresult:

#st=str(x[0])+" => "+str(x[1])+" => "+ str(x[2])+" => "+str(x[3])+" => "+str(x[4])+" => "+str(x[5])+" => "+str(x[6])+" => "+str(x[7])+" => "+str(x[8])+" => "+str(x[9])

Lb1.insert(i,x)

i=i+1

else:

print('Booking not found')

Lb1.pack()

Button(view\_booking, text="Exit", command=ex1).pack()

def ex1():

view\_booking.destroy()

# Deleting popups

def delete\_staff\_login\_success():

staff\_login\_success\_screen.destroy()

def book\_room():

try:

room\_type=""

rch=0

food\_id=0

fch=0;

gt=0

room=""+tkvar.get() #roomno=tkvar.get()

food=""+tkvar1.get()

ds=days.get()

if room=="" :

Label(customer\_booking\_screen, text="Select a room no", fg="red", font=("calibri", 11)).pack()

#messagebox.showinfo("Registration ", "User Name cannot be empty")

elif food=="" :

Label(customer\_booking\_screen, text="Selec food item", fg="red", font=("calibri", 11)).pack()

else:

print(room)

print(food)

print(days)

mycursor = mydb.cursor()

sql = "select \* from Room where room\_no="+room

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

if count>0:

for x in myresult:

room\_type=x[1]

rch=x[2]

else:

print('not found')

mycursor = mydb.cursor()

sql = "select \* from Food where Food\_Type='"+food+"'"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

if count>0:

for x in myresult:

food\_id=x[0]

fch=x[2]

else:

print('not found')

mycursor = mydb.cursor()

amt=rch\*ds

gt=fch+amt

sql = "INSERT INTO booking(Room\_No, Room\_Type, Food\_ID, Food\_Type, Room\_Charges,days,amount, Food\_Charges, Grand\_Total, UserName) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"

val = (room,room\_type,food\_id,food,rch,ds,amt,fch,gt,u)

r=mycursor.execute(sql, val)

mydb.commit()

days\_entry.delete(0, END)

Label(customer\_booking\_screen, text="Booking Done Successfully", fg="red", font=("calibri", 11)).pack()

Label(customer\_booking\_screen, text="").pack()

#Button(customer\_booking\_screen, text="Exit", width=10, height=1, bg="cyan",command = ex).pack()

except ConnectionError:

Label(customer\_booking\_screen, text="Failed to insert record", fg="green", font=("calibri", 11)).pack()

def ex():

main\_menu.destroy();

#customer\_booking\_screen.destroy()

def cancel\_booking():

global customer\_Cancel\_booking\_screen

global popupMenu

global tkvar2

customer\_Cancel\_booking\_screen = Toplevel(staff\_screen)

customer\_Cancel\_booking\_screen.geometry("600x500")

customer\_Cancel\_booking\_screen.title("CANCEL BOOKING")

Label(customer\_Cancel\_booking\_screen,text=" \* \* \* BOOKING MENU \* \* \* ", bg="cyan", width="300", height="2", font=("Bell MT", 18)).pack()

Label(customer\_Cancel\_booking\_screen,text="").pack()

Label(customer\_Cancel\_booking\_screen,text="Select Room No To Cancel booking ", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(customer\_Cancel\_booking\_screen,text="").pack()

mycursor = mydb.cursor()

sql = "select room\_no from Booking"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

if count>0:

tkvar2 = StringVar(customer\_Cancel\_booking\_screen)

s=""

for x in myresult:

s=s+str(x[0])+","

choices=(s.split(","))

tkvar2.set(str(x[0]))

popupMenu=OptionMenu(customer\_Cancel\_booking\_screen,tkvar2,\*choices)

popupMenu.pack()

Label(customer\_Cancel\_booking\_screen,text="").pack()

Button(customer\_Cancel\_booking\_screen,text="Cancel Booking", height="2", width="30", bg="cyan",command = cancel\_book).pack()

Label(customer\_Cancel\_booking\_screen,text="").pack()

else:

Label(customer\_Cancel\_booking\_screen, text="Room Details not found, contact Admin", fg="red", font=("calibri", 11)).pack()

def cancel\_book():

try:

if tkvar2=="" :

Label(customer\_Cancel\_booking\_screen, text="Select a room no to cancel", fg="red", font=("calibri", 11)).pack()

else:

mycursor = mydb.cursor()

room=""+tkvar2.get()

sql = "delete from booking where room\_no="+room

r=mycursor.execute(sql)

mydb.commit()

Label(customer\_Cancel\_booking\_screen, text="Record deleted successfully", fg="red", font=("calibri", 11)).pack()

Label(customer\_Cancel\_booking\_screen, text="").pack()

Button(customer\_Cancel\_booking\_screen,text="Exit", height="2", width="30", bg="cyan",command=Ex9).pack()

except ConnectionError:

Label(customer\_Cancel\_booking\_screen, text="Failed to delete record", fg="green", font=("calibri", 11)).pack()

def Ex9():

customer\_Cancel\_booking\_screen.destroy()

def viewr():

global view\_room

view\_room = Toplevel(customer\_menu\_screen)

view\_room.title("View Rooms")

view\_room.geometry("450x450")

mycursor = mydb.cursor()

sql = "select \* from Room"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

Lb1 = Listbox(view\_room,bg="cyan", width="450", height="10")

#Lb1.insert(1, "ID => Date => Room\_ID=>Room\_Type=>Food\_Id=>Food\_Type=>Room Charges=>Food\_Charges=>Grand\_Total=>UserName")

i=1

if count>0:

for x in myresult:

#st=str(x[0])+" => "+str(x[1])+" => "+ str(x[2])+" => "+str(x[3])+" => "+str(x[4])+" => "+str(x[5])+" => "+str(x[6])+" => "+str(x[7])+" => "+str(x[8])+" => "+str(x[9])

Lb1.insert(i,x)

i=i+1

else:

print('Rom Details not found')

Lb1.pack()

Button(view\_room, text="Exit", command=ex21).pack()

def ex21():

view\_room.destroy()

def viewf():

global view\_food

view\_food = Toplevel(customer\_menu\_screen)

view\_food.title("View Rooms")

view\_food.geometry("450x450")

mycursor = mydb.cursor()

sql = "select \* from Food"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

Lb1 = Listbox(view\_food,bg="cyan", width="450", height="10")

#Lb1.insert(1, "ID => Date => Room\_ID=>Room\_Type=>Food\_Id=>Food\_Type=>Room Charges=>Food\_Charges=>Grand\_Total=>UserName")

i=1

if count>0:

for x in myresult:

#st=str(x[0])+" => "+str(x[1])+" => "+ str(x[2])+" => "+str(x[3])+" => "+str(x[4])+" => "+str(x[5])+" => "+str(x[6])+" => "+str(x[7])+" => "+str(x[8])+" => "+str(x[9])

Lb1.insert(i,x)

i=i+1

else:

print('Fooditem not found')

Lb1.pack()

Button(view\_food, text="Exit", command=ex221).pack()

def ex221():

view\_food.destroy()

def Customer\_Menu():

global customer\_menu\_screen

#staff\_screen = Tk()

customer\_menu\_screen = Toplevel(login\_screen)

customer\_menu\_screen.geometry("600x600")

customer\_menu\_screen.title("CUSTOMER MENU")

Label(customer\_menu\_screen,text="LOGIN SUCCESSFUL").pack()

canvas = Canvas(customer\_menu\_screen, width = 250, height = 200)

canvas.pack()

img = PhotoImage(file="staff.png")

canvas.create\_image(20,20, anchor=NW, image=img)

Label(customer\_menu\_screen,text="Great offers & packages when you book direct.", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(customer\_menu\_screen,text="").pack()

Button(customer\_menu\_screen,text="Booking", height="2", width="30", command=Booking).pack()

Label(customer\_menu\_screen,text="").pack()

Button(customer\_menu\_screen,text="View Room", height="2", width="30", command=viewr).pack()

Label(customer\_menu\_screen,text="").pack()

Button(customer\_menu\_screen,text="View Food", height="2", width="30", command=viewf).pack()

Label(customer\_menu\_screen,text="").pack()

Button(customer\_menu\_screen,text="Exit", height="2", width="30", command=ex33).pack()

customer\_menu\_screen.mainloop()

def ex33():

customer\_menu\_screen.destroy()

def Booking():

#login\_success\_screen.destroy()

global customer\_booking\_screen

global popupMenu

global tkvar

global tkvar1

global days

days = IntVar()

global days\_entry

customer\_booking\_screen = Toplevel(customer\_menu\_screen)

customer\_booking\_screen.geometry("600x500")

customer\_booking\_screen.title("Book a Room")

Label(customer\_booking\_screen,text=" \* \* \* BOOKING MENU \* \* \* ", bg="cyan", width="300", height="2", font=("Bell MT", 18)).pack()

Label(customer\_booking\_screen,text="").pack()

Label(customer\_booking\_screen,text="Select Room No.", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(customer\_booking\_screen,text="").pack()

mycursor = mydb.cursor()

sql = "select room\_no from Room"

mycursor.execute(sql)

myresult = mycursor.fetchall()

count = mycursor.rowcount

if count>0:

tkvar = StringVar(customer\_booking\_screen)

s=""

for x in myresult:

s=s+str(x[0])+","

choices=(s.split(","))

tkvar.set(str(x[0]))

popupMenu=OptionMenu(customer\_booking\_screen,tkvar,\*choices)

popupMenu.pack()

Label(customer\_booking\_screen, text="").pack()

name\_lable = Label(customer\_booking\_screen, text="Enter no of Days \* ")

name\_lable.pack()

days\_entry = Entry(customer\_booking\_screen, textvariable=days)

days\_entry.pack()

Label(customer\_booking\_screen,text="").pack()

mycursor1 = mydb.cursor()

sql = "select food\_type from food"

mycursor1.execute(sql)

myresult1 = mycursor1.fetchall()

count1 = mycursor1.rowcount

if count1>0:

tkvar1 = StringVar(customer\_booking\_screen)

#('101','102','103','104','201','202')

s=""

for x in myresult1:

s=s+str(x[0])+","

choices1=(s.split(","))

tkvar1.set(str(x[0]))

Label(customer\_booking\_screen,text="Select Food Item.", bg="cyan", width="300", height="2", font=("Calibri", 13)).pack()

Label(customer\_booking\_screen,text="").pack()

popupMenu1=OptionMenu(customer\_booking\_screen,tkvar1,\*choices1)

popupMenu1.pack()

Label(customer\_booking\_screen,text="").pack()

Button(customer\_booking\_screen,text="Book", height="2", width="30", bg="cyan",command = book\_room).pack()

Label(customer\_booking\_screen,text="").pack()

Button(customer\_booking\_screen,text="Exit", height="2", width="30", bg="cyan",command=delete\_booking).pack()

else:

Label(customer\_booking\_screen, text="Food item not found, contact Admin", fg="red", font=("calibri", 11)).pack()

else:

Label(customer\_booking\_screen, text="Room Details not found, contact Admin", fg="red", font=("calibri", 11)).pack()

def delete\_booking():

customer\_booking\_screen.destroy()

def sdelete\_password\_not\_recognised():

spassword\_not\_recog\_screen.destroy()

def delete\_password\_not\_recognised():

password\_not\_recog\_screen.destroy()

def delete\_user\_not\_found\_screen():

user\_not\_found\_screen.destroy()

mydb = mysql.connector.connect(host="localhost",user="root", passwd="root",database="harshinee")

main\_menu()