HOTEL AAKSHAM

Prepared For
D.N.G. Sir

Date :- __/__/__

LIONS SCHOOL, MIRZAPUR

SESSION : 2023-24

AISSCE PROJECT
INFORMATICS PRACTICS - (065)
CLASS - XII

HOTEL MANAGMENT SYSTEM

SUBMITTED TO : MR . DWARIKA NATH GUPTA SIR

SUBMITTED BY :
- SAKSHAM JAISWAL
- ADITYA JAISWAL
- ARCHANA DUBEY
- YASH JAISWAL

DATE OF SUBMISSION : _____

SIGNATURE OF INVIGILATOR

SIGNATURE OF EXTERNAL INVIGILATOR

SIGNATURE OF PRINCIPAL

CERTIFICATE

OF COMPLETION

This Certificate Is Proudly Presented To

Saksham Jaiswal , Aditya Jaiswal ,

Yash Jaiswal , Archana Dubey

for successfully completing the G.U.I Application Of Hotel Aaksham

EXAMINER

TEACHER

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my I.P. Teacher D.N.G. Sir as well as our Principal, who gave me the golden opportunity to do this wonderful project. This project also helped me in doing a lot of research and of came to know about so many new things.

I am extremely grateful to my parents & my friends who gave valuable suggestions and guidance for completion of my Project This cooperation and healty critism come handy and useful with them.

Hence, I would like to thanks all the above mentioned people once again.

PREFACE

Welcome to the future of seamless and efficient hotel management! Aaksham Software is a cutting-edge solution designed to revolutionize the way you run your hospitality business. Whether you manage a small boutique hotel or a large resort, our software is crafted to meet your unique needs, providing an all-encompassing platform to streamline operations and enhance guest experiences.

In an era where precision and speed define success, Aaksham Software stands as a testament to innovation in the hospitality industry. Our team of dedicated professionals has poured countless hours into developing a robust and user-friendly software that empowers hoteliers to take control of every aspect of their operations.

Key Features:

1. ***Reservation Management:*** *Effortlessly handle bookings, check-ins, and check-outs with our intuitive reservation system. Say goodbye to overbookings and missed opportunities.*
2. ***Billing and Invoicing:*** *Simplify your financial processes with our comprehensive billing and invoicing module. Generate accurate invoices, track payments, and gain insights into your revenue streams.*
3. ***Guest Profiles:*** *Know your guests better with detailed profiles. Capture preferences and create personalized experiences that keep guests coming back.*
4. ***Inventory Control:*** *Manage your room inventory, track stock levels, and optimize resource allocation. Reduce waste and enhance operational efficiency.*
5. ***Reporting and Analytics:*** *Make informed decisions with powerful reporting tools. Analyze performance metrics, identify trends, and strategize for success.*

Why Aaksham Software?

- ***User-Friendly Interface:*** *Our software is designed for simplicity without compromising functionality. Users of all technical backgrounds can navigate effortlessly.*
- ***Scalability:*** *Whether you're a small boutique hotel or a large chain, Aaksham Software scales to meet your needs. Grow your business without outgrowing your software.*
- ***24/7 Support:*** *Our dedicated support team is always ready to assist you. From implementation to ongoing support, we're with you every step of the way.*

INDEX

INTRODUCTION

We are thrilled to introduce "Aaksham Software" a revolutionary hotel management software meticulously crafted by a team of innovative minds from Class 12. With passion, dedication, and a deep understanding of the evolving hospitality industry, we have designed a user-friendly and feature-rich solution that aims to simplify the complexities of hotel operations.

In the competitive realm of hotel management, efficiency and precision are paramount. "Aaksham Software" is not just a software; it's a testament to the creativity and technical prowess of our student team. This project represents the culmination of our academic journey, blending theoretical knowledge with practical application to create a tool that empowers hoteliers, big or small.

Our project emphasizes the efficient use of Python for the basic layout of the whole managment system and SQL for the data managment which leads to streamline processes, enhance customer satisfaction, and contribute to the evolving landscape of modern retail management.

Key Features:

1. *Intuitive Reservation System: Seamlessly manage bookings, check-ins, and check-outs with an easy-to-use reservation system. Our goal is to simplify the front desk process and enhance the overall guest experience.*
2. *Student-Crafted Design: Every pixel and line of code in "Aaksham Software" is a result of our team's dedication and passion. The user interface is not only functional but also aesthetically pleasing, reflecting the modern standards of software design.*
3. *Budget-Friendly Solution: Being students ourselves, we understand the importance of cost-effectiveness. "Aaksham Software" is a budget-friendly option without compromising on functionality, making it accessible to a wide range of hotel owners and managers.*
4. *Continuous Learning and Improvement: As students, we embrace the spirit of continuous learning. "Aaksham Software" is not a static project; it's a dynamic endeavor that will evolve with your feedback and the changing landscape of the hospitality industry.*

PROBLEM STATEMENT

The hospitality industry is renowned for its dynamic nature and ever-growing demand for seamless guest experiences. However, many hotels, especially smaller establishments, face significant challenges in effectively managing their day-to-day operations. Manual and outdated systems often lead to inefficiencies, errors, and a lack of real-time insights, hindering the overall performance and competitiveness of these establishments.

Key Problems:

1. ***Reservation Chaos:*** *Booking management is a fundamental aspect of hotel operations. However, many hotels struggle with overbookings, double bookings, and the inability to efficiently manage reservations, resulting in a negative impact on guest satisfaction and revenue.*
2. ***Inefficient Front Desk Operations:*** *Traditional front desk processes can be time-consuming and prone to errors. Paper-based check-ins, manual room assignments, and lack of real-time information often lead to delays and frustration for both staff and guests.*
3. ***Limited Guest Engagement:*** *Establishing and maintaining meaningful connections with guests is crucial for building loyalty. However, without a robust Guest Relationship Management (CRM) system, hotels may miss opportunities to provide personalized experiences, resulting in a generic and less memorable stay for guests.*
4. ***Inventory and Stock Challenges:*** *Managing inventory, especially in-house amenities, can be a significant headache for hoteliers. Stockouts, overstocking, and difficulties in tracking consumption can impact service delivery and operational costs.*
5. ***Financial Ambiguity:*** *Financial management is a critical aspect of any business, and hotels are no exception. Without an integrated accounting system, hotels may struggle with financial reporting, expense tracking, and compliance, leading to potential financial discrepancies.*

The Solution:

A comprehensive Hotel Management Software aims to address these challenges by providing an integrated platform that streamlines reservations, enhances front desk efficiency, improves guest engagement through CRM, optimizes inventory and stock control, and ensures accurate financial management. This solution is not only designed to simplify operations but also to empower hotel owners and managers with the tools necessary to stay competitive in the ever-evolving hospitality landscape. The objective is to develop a user-friendly, cost-effective, and feature-rich Hotel Management Software that caters to the unique needs of hotels, particularly smaller establishments, and enables them to provide exceptional guest experiences while optimizing their internal processes.

OBJECTIVE

The objectives of Hotel Management Software are multifaceted, aiming to streamline operations, enhance guest experiences, and optimize overall efficiency within a hotel setting. Here are the primary objectives:

1. ***Efficient Reservation Management:***
   - *Simplify the process of managing reservations, check-ins, and check-outs.*
   - *Avoid overbookings and streamline room allocation.*

2. ***Analytics and Reporting:***
   - *Provide detailed analytics and reports to track key performance indicators (KPIs).*
   - *Enable data-driven decision-making for hotel owners and managers.*

3. ***Scalability and Adaptability:***
   - *Design the software to be scalable, catering to the needs of hotels of various sizes.*
   - *Ensure adaptability to different types of hotel structures and business models.*

4. ***User-Friendly Interface:***
   - *Create an intuitive and user-friendly interface to facilitate quick adoption by hotel staff.*
   - *Provide training and support to ensure staff members can effectively use the software.*

5. ***Security and Compliance:***
   - *Implement robust security measures to protect sensitive guest and financial data.*
   - *Ensure compliance with industry regulations and standards.*

By achieving these objectives, Hotel Management Software contributes to a more organized, efficient, and guest-centric operation, ultimately enhancing the overall success and reputation of the hotel.

PROJECT SCOPE

The project scope of a Hotel Management Software outlines the boundaries, goals, and deliverables of the development effort. Here is a detailed breakdown of the project scope:

1. System Overview:

- Develop a comprehensive Hotel Management Software tailored to the needs of the hospitality industry.

- The system will include modules for reservations, front desk operations, guest relationship management, inventory control, financial management, analytics, and reporting.

2. User-Friendly Interface:

- Design an intuitive and user-friendly interface to ensure easy adoption by hotel staff.

- Provide training materials and support to help users make the most of the software.

3. Security Measures:

- Implement robust security protocols to protect guest data and financial information.

- Ensure compliance with data protection regulations.

4. Scalability and Adaptability:

- Design the software to be scalable, accommodating hotels of various sizes and structures.

- Ensure flexibility to adapt to different business models within the hospitality industry.

5. Testing and Quality Assurance:

- Conduct thorough testing of the software to identify and rectify bugs and issues.

- Ensure that the software meets quality standards and is reliable for day-to-day operations.

6. Documentation:

- Provide comprehensive documentation, including user manuals and technical documentation, to assist users and support teams.

By defining a clear and comprehensive project scope, the development team can work efficiently towards delivering a Hotel Management Software that meets the needs of the hospitality industry and its users.

SYSTEM REQUIRMENTS

- ***Brand & Model*** *- Nuvobook V1*
- ***RAM*** *- 8 GB*
- ***Processor*** *- 11th Gen Intel(R) Core(TM) i5-1155G7 @2.5 GHz upto max turbo frequency at 4.5 Ghz, 2496 Mhz, 4 Core(s), 8 Logical Processor(s)*
- ***Storage*** *- 512GB M.2 NVMe PCIe 3.0 SSD*
- ***Display*** *- 15.6-inch (39.62cm), FHD (1920 x 1080) 16:9 aspect ratio, 60Hz refresh rate, 250nits, 45% NTSC color gamut, Anti glare display ,IPS LCD DISPLAY*
- ***Integrated*** *- Intel(R) Iris(R) Xe Graphics*
- ***Operating System*** *- Windows 11 Home*

APPLICATIONS USED

In crafting our Hotel Management System project, we harnessed the power of two indispensable tools: ***PyCharm & MySQL.***

PyCharm, an intelligent Python IDE, facilitated seamless coding, debugging, and collaboration, ensuring the efficiency of our software development process.

Complementing this, MySQL, a robust relational database management system, enabled us to organize and manage vast datasets crucial for our store management system.

Together, these applications form the backbone of our project, ensuring a dynamic, scalable, and proficient Hotel Management System.

OVERVIEW OF PYTHON

Python, a versatile and high-level programming language, has emerged as a cornerstone in the world of software development, data science, and automation. Created by Guido van Rossum and first released in 1991, Python is celebrated for its readability, simplicity, and the ease with which developers can express concepts in fewer lines of code. Its expansive standard library and comprehensive ecosystem of thirdparty packages make Python a go-to choice for diverse applications. Python is an interpreted language, allowing developers to test and execute code swiftly, fostering an iterative development process. Python's readability is enhanced by its use of whitespace indentation, promoting clean and organized code structures. This, coupled with dynamic typing, makes it an accessible language for both beginners and experienced programmers. Python's applicability extends across web development, scientific computing, machine learning, and artificial intelligence. Frameworks like Django and Flask empower developers to build robust web applications, while libraries such as NumPy and Pandas facilitate complex data manipulation and analysis. Moreover, Python's open-source nature has fostered a vibrant community, contributing to its continuous evolution. The language's adaptability, versatility, and community support position Python as a language of choice for solving intricate problems and driving innovation in various domains.

OVERVIEW OF MYSQL

MySQL, a renowned relational database management system, stands as a stalwart in the realm of data management, offering a robust and scalable solution for diverse applications. Following the principles of relational database architecture, MySQL organizes data into structured tables with defined relationships, ensuring data integrity through ACID properties. Its compatibility with the widely accepted Structured Query Language (SQL) enables seamless interaction, facilitating integration with various programming languages and applications. MySQL is celebrated for its scalability and performance optimization features, making it adept at handling substantial datasets and high transaction volumes. Its open-source nature fosters collaboration, innovation, and adaptability, empowering users to modify and redistribute the source code. The cross-platform compatibility of MySQL ensures its versatility across different operating systems. Notably, MySQL incorporates security measures such as user authentication, access control, and encryption to safeguard sensitive data. With a thriving community, MySQL benefits from continuous improvements, extensive documentation, and compatibility with web technologies. It remains a preferred choice for applications ranging from small-scale projects to enterprise-level systems, providing a reliable and feature-rich environment for effective data management.

PROJECT MODULE

Hotel Managment System is divided into following modules :

Loading Module : *This consist of a loading page which shows loading of the app for the few seconds.*

Signup Module : *This allows user to signup to the software.*

Signin Module : *This allows user to login to the system if he/she has already registered.*

Forgetpass Module : *It allows user to change the password in case he/she has forgotten his password.*

Forget Username Module : *It allows user to change the user name in case he/she has forgotten his /her username.*

Homepage Module : *In this page , It is divided into 12 sub category :*

1. Guest Entry : *It shows basic details of the App & contain sub category :*

1.1 Services : *It allows to provide services to the customer like :*

Food Services, Laundry Service, Shift Room, Change Check Out Date

1.2 Room Availability : *It Shows the room status and contain 2 sub category : Room Reservation ,Check In*

2. Add Customer : *It allows user to add , modify or delete the Customer.*

PROJECT MODULE

3. **Check In Reserved Room :** *It allows to check in Reserved Room.*

4. **Check Out :** *It allows to check out the the customer.*

4.1 Check Out Customer Details : *It Contain list of number of check out.*

5. **Reservation List :** *It contain the list of room reservations .*

6.**Hall :** *It allows to book Hall in hotel and contain 2 sub category .*

6.1 Hall Billing : *It allows to to generate bill for hall booking.*

6.2 Hall Details : *It Contain list of hall booking.*

7. **Bar & Restaurant Billing :** *It allows to make bill for restaurant .*

8 . **Worker Details :** *It Contain information about worker in hotel.*

9 . **Guest Complaint :** *It contain list of guest complaints.*

10 . **Report :** *It contain analytics of hotel.*

11. **About Us :** *It contain information about hotel.*

11.1 About Team Members : *It contain information about team members*

12 . **Exit :** *It allows user to exit from the system.*

12 .1 Sign Out : *It allows user to sign out out from the system.*

LIBRARIES USED

For creating this project , I have used several libraries from which some are listed below which are majorly used –

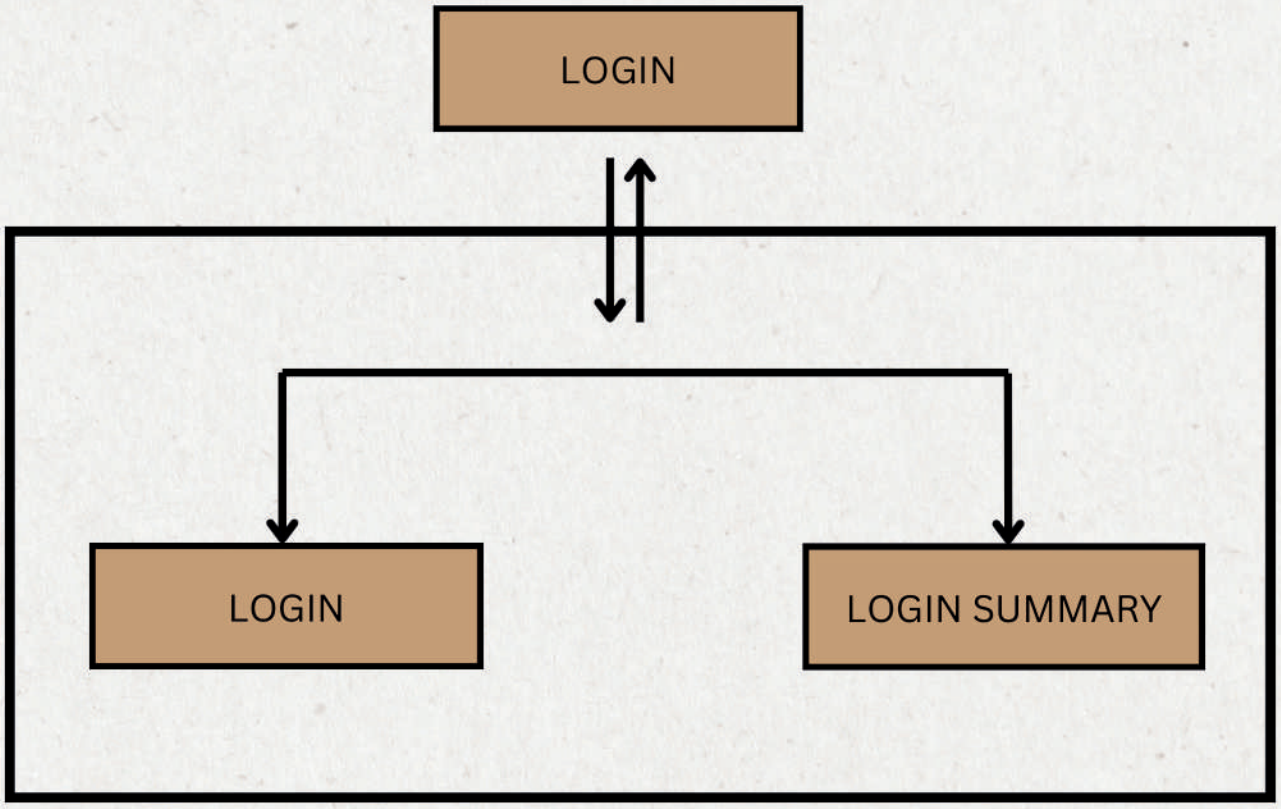
Tkinter : *Tkinter is a Python library for creating graphical user interfaces (GUIs). It provides a set of tools for building windows, dialogs, buttons, and other GUI elements. I have use this to prepare the frontend of the system .It is easy to use, making it a popular choice for beginners in GUI development. It's built on the Tk GUI toolkit and offers a simple way to create interactive applications.*

Python Image Library (PIL) : *The Python Imaging Library (PIL), now known as the Pillow library, is a powerful image processing library for Python. It provides a wide range of functionalities, including opening, manipulating, and saving various image formats. PIL/Pillow supports basic image editing operations like cropping, resizing, and filtering. It facilitates image enhancements, transformations, and supports diverse image file formats.*

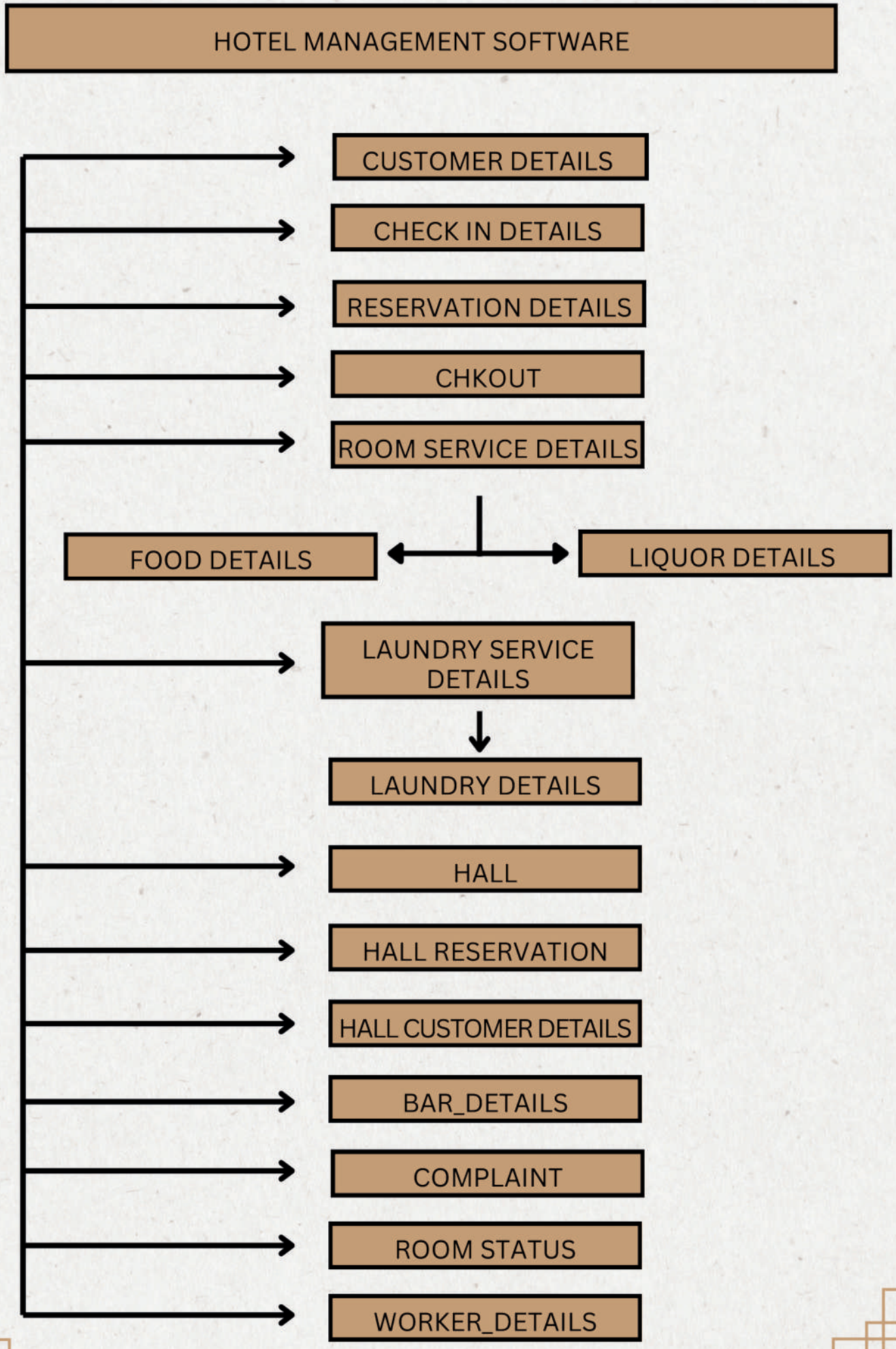
smtplib : *The smtplib module provides an interface to send emails using the SMTP protocol. You can integrate it into your Tkinter application by creating an entry form for the user to input the email content and credentials, and then use smtplib to send the email when the user clicks a button.*

DateEntry : *The `dateentry` library is an extension for Tkinter in Python, providing a widget for easy date input in graphical user interfaces. It simplifies date selection, allowing users to input and choose dates conveniently. This library enhances the Tkinter toolkit, making it more userfriendly for applications that require date-based interactions.*

DATA FLOW DIAGRAM



DATA FLOW DIAGRAM



Database Design and Table Structure

1. *Table Name – LOGIN*

	Field	Type	Null	Key	Default	Extra
▶	Email ID	varchar(40)	NO	UNI	NULL	
	UserName	varchar(20)	NO	PRI	NULL	
	Password	varchar(20)	NO	UNI	NULL	

2. *Table Name – LOGIN SUMMARY*

	Field	Type	Null	Key	Default	Extra
▶	UserName	varchar(20)	NO	PRI	NULL	
	Password	varchar(20)	NO	UNI	NULL	

3. *Table Name – CUSTOMER DETAILS*

	Field	Type	Null	Key	Default	Extra
▶	S No.	char(5)	NO	UNI	NULL	
	Guest ID	char(5)	NO	PRI	NULL	
	Guest Name	varchar(20)	NO		NULL	
	Gender	varchar(11)	NO		NULL	
	Religion	varchar(7)	NO		NULL	
	Address	varchar(25)	NO		NULL	
	City	varchar(20)	NO		NULL	
	Country	varchar(20)	NO		NULL	
	Contact No.	varchar(13)	NO		NULL	
	ID Type	varchar(20)	NO		NULL	
	ID Number	varchar(15)	NO	UNI	NULL	
	Status	varchar(10)	NO		NULL	
	Email ID	varchar(50)	NO	UNI	NULL	
	Image	varchar(83)	YES		NULL	

4 . Table Name – CHECK IN

	Field	Type	Null	Key	Default	Extra
▶	Guest ID	char(5)	NO		NULL	
	Guest Name	varchar(20)	NO		NULL	
	Room No	char(6)	NO		NULL	
	Room Type	varchar(15)	NO		NULL	
	Day In	date	NO		NULL	
	Day Out	date	NO		NULL	
	Room Price	int	NO		NULL	
	Payment	int	NO		NULL	
	Payment Mode	varchar(15)	NO		NULL	
	Payment Date	date	NO		NULL	
	Status	varchar(10)	NO		NULL	
	Booking Type	char(11)	NO		NULL	
	Room Original Price	float	NO		NULL	

5 . Table Name – RESERVATION DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Guest ID	char(5)	NO		NULL	
	Guest Name	varchar(20)	NO		NULL	
	Room No	char(6)	NO		NULL	
	Room Type	varchar(15)	NO		NULL	
	Day In	date	NO		NULL	
	Day Out	date	NO		NULL	
	Room Price	int	NO		NULL	
	Payment	int	NO		NULL	
	Payment Mode	varchar(15)	NO		NULL	
	Payment Date	date	NO		NULL	
	Status	varchar(10)	NO		NULL	
	Booking Type	char(11)	NO		NULL	

6 . Table Name – CHECK OUT

	Field	Type	Null	Key	Default	Extra
▶	Guest Id	char(5)	NO		NULL	
	Guest Name	varchar(50)	NO		NULL	
	Gender	varchar(10)	NO		NULL	
	Religion	varchar(15)	NO		NULL	
	Address	varchar(50)	NO		NULL	
	City	varchar(20)	NO		NULL	
	Country	varchar(20)	NO		NULL	
	Contact No	char(13)	NO		NULL	
	Email Id	varchar(50)	NO		NULL	
	Id Type	varchar(20)	NO		NULL	
	Id No	varchar(30)	NO		NULL	
	Room No	char(5)	NO		NULL	
	Date In	date	NO		NULL	
	Date Out	date	NO		NULL	
	Room Type	varchar(50)	NO		NULL	
	Room Charges	int	NO		NULL	
	Discount %	float	NO		NULL	
	SGST %	float	NO		NULL	
	CGST %	float	NO		NULL	
	Grand Total	float	NO		NULL	
	Reservation Payment	int	NO		NULL	
	Advance Payment	int	NO		0	DEFAUL...
	Remaining Price	float	NO		NULL	
	Payment Mode	varchar(20)	NO		NULL	
	Payment Date	date	NO		NULL	
	Payment	float	NO		NULL	

7 . Table Name – ROOM SERVICE DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Guest ID	char(5)	NO		NULL	
	Guest Name	varchar(20)	NO		NULL	
	Room No	char(6)	NO		NULL	
	Bill No.	varchar(15)	NO	PRI	NULL	
	Bill Date	date	NO		NULL	
	Payment Mode	varchar(15)	NO		NULL	

8 . Table Name – FOOD DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Bill No.	varchar(15)	YES		NULL	
	Food Name	varchar(20)	NO		NULL	
	Rate	int	NO		NULL	
	Quantity	char(3)	YES		NULL	
	Amount	int	NO		NULL	
	Discount	decimal(3,2)	NO		NULL	
	SGST	decimal(3,2)	NO		NULL	
	CGST	decimal(3,0)	NO		NULL	
	Total Amount	int	NO		NULL	
	Pavment Date	date	NO		NULL	

9 . Table Name – LIQUOR DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Bill No.	varchar(15)	YES		NULL	
	Liquor Name	varchar(25)	NO		NULL	
	Rate	int	NO		NULL	
	Quantity	char(3)	YES		NULL	
	Amount	int	NO		NULL	
	Discount	decimal(3,2)	NO		NULL	
	SGST	decimal(3,2)	NO		NULL	
	CGST	decimal(3,2)	NO		NULL	
	Total Amount	int	NO		NULL	
	Payment Date	date	NO		NULL	

10 . Table Name – LAUNDRY SERVICE DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Guest ID	char(5)	NO		NULL	
	Guest Name	varchar(20)	NO		NULL	
	Room No	char(6)	NO		NULL	
	Bill No.	varchar(15)	NO	PRI	NULL	
	Bill Date	date	NO		NULL	
	Payment Mode	varchar(15)	NO		NULL	

11 . Table Name – LAUNDRY DETAILS

Field	Type	Null	Key	Default	Extra
Bill No.	varchar(15)	YES		NULL	
Service Name	varchar(25)	NO		NULL	
Rate	int	NO		NULL	
Quantity	char(3)	YES		NULL	
Amount	int	NO		NULL	
Discount	decimal(3,2)	NO		NULL	
SGST	decimal(3,2)	NO		NULL	
CGST	decimal(3,2)	NO		NULL	
Total Amount	int	NO		NULL	
Payment Date	date	YES		NULL	

12 . Table Name – HALL

Field	Type	Null	Key	Default	Extra
Hall No	char(5)	NO	PRI	NULL	
Price	int	NO		NULL	

13 . Table Name – HALL RESERVATION

Field	Type	Null	Key	Default	Extra
Guest ID	char(5)	NO		NULL	
Guest Name	varchar(20)	NO		NULL	
Address	varchar(25)	NO		NULL	
Contact No.	varchar(13)	NO		NULL	
ID Type	varchar(20)	NO		NULL	
ID Number	varchar(15)	NO		NULL	
Email ID	varchar(70)	NO		NULL	
Hall Number	char(4)	NO		NULL	
Hall price	varchar(10)	NO		NULL	
From Date	date	NO		NULL	
To Date	date	NO		NULL	
Payment Date	date	NO		NULL	
Image	varchar(100)	NO		NULL	
Total	int	NO		NULL	
Advanve	int	NO		NULL	

14 . Table Name – HALL CUSTOMER DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Guest Id	char(5)	YES		NULL	
	Guest Name	varchar(50)	NO		NULL	
	Address	varchar(50)	NO		NULL	
	Contact No	char(13)	NO		NULL	
	Id Type	varchar(20)	NO		NULL	
	Id No	varchar(30)	NO		NULL	
	Email Id	varchar(50)	NO		NULL	
	Image	varchar(100)	NO		NULL	
	Hall No	char(5)	NO		NULL	
	Date In	date	NO		NULL	
	Date Out	date	NO		NULL	
	Hall Price	int	NO		NULL	
	Total Amount	int	NO		NULL	
	Advance Payment	int	NO		NULL	
	Payment	float	NO		NULL	
	Payment Mode	varchar(20)	NO		NULL	
	Payment Date	date	NO		NULL	

15 . Table Name – BAR_DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Bill No.	char(5)	NO		NULL	
	Guest Id	char(5)	NO		NULL	
	Guest Name	varchar(50)	NO		NULL	
	Gender	varchar(10)	NO		NULL	
	Contact No	char(13)	NO		NULL	
	Food Name	varchar(50)	NO		NULL	
	Rate	int	NO		NULL	
	Quantity	int	NO		NULL	
	Discount %	float	NO		NULL	
	SGST %	float	NO		NULL	
	CGST %	float	NO		NULL	
	Grand Total	float	NO		NULL	
	Payment Date	date	NO		NULL	

16 . Table Name – COMPLAINT

	Field	Type	Null	Key	Default	Extra
▶	Guest Id	char(5)	NO	PRI	NULL	
	Guest Name	varchar(50)	NO		NULL	
	Contact No	char(13)	NO		NULL	
	Complaint	varchar(50)	NO		NULL	

17 . Table Name – ROOM STATUS

	Field	Type	Null	Key	Default	Extra
▶	Room No.	varchar(15)	NO	PRI	NULL	
	Room Type	varchar(25)	NO		NULL	
	Status	varchar(10)	NO		NULL	
	Price	int	NO		NULL	

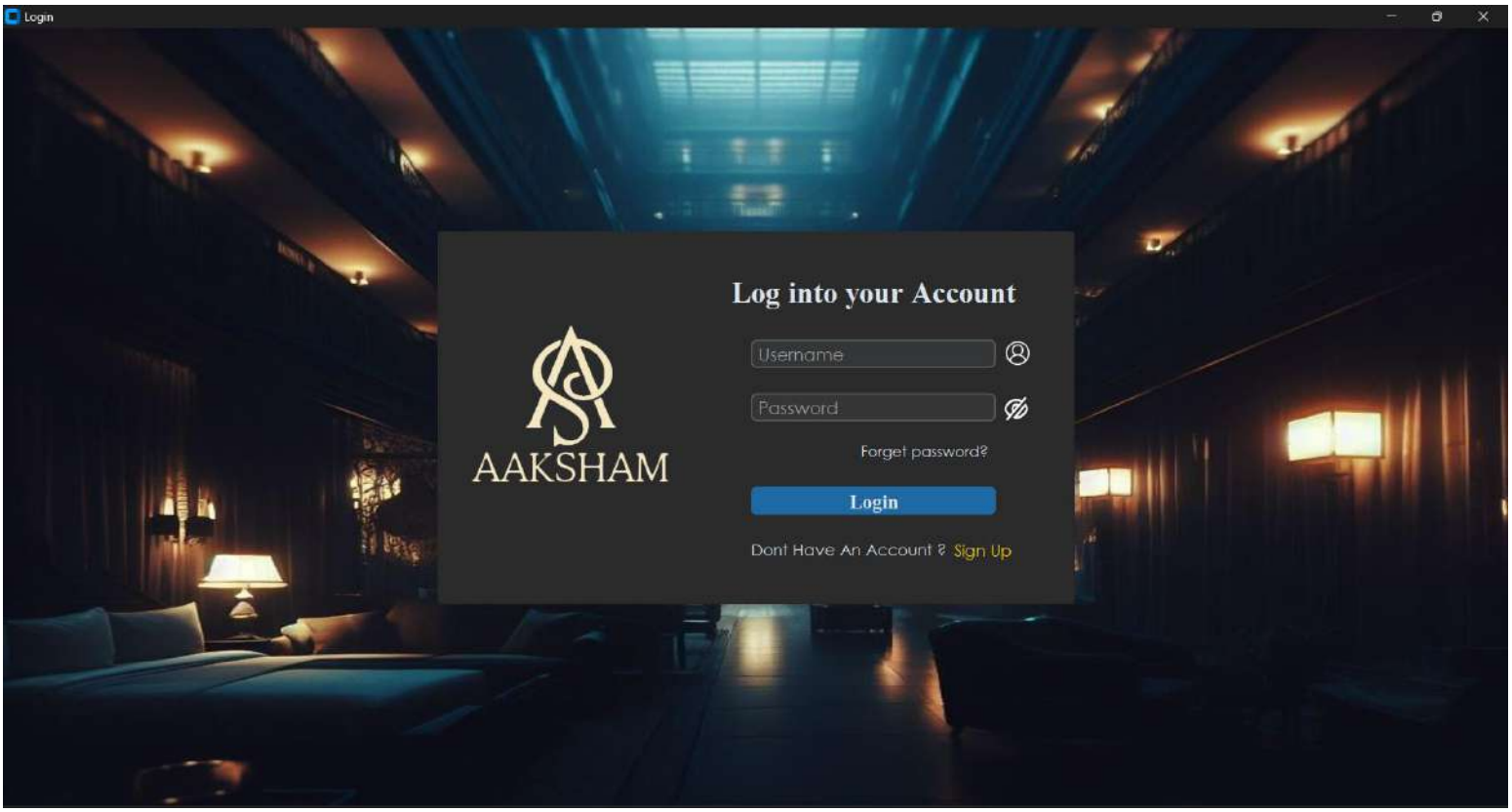
18 . Table Name – WORKER_DETAILS

	Field	Type	Null	Key	Default	Extra
▶	Worker Id	char(5)	NO	PRI	NULL	
	Worker Name	varchar(50)	NO		NULL	
	Gender	varchar(10)	NO		NULL	
	Religion	varchar(15)	NO		NULL	
	Address	varchar(50)	NO		NULL	
	Contact No	char(13)	NO		NULL	
	Date Of Joining	date	NO		NULL	
	Id Type	varchar(10)	NO		NULL	
	Id No.	varchar(13)	NO		NULL	
	Department	varchar(15)	NO		NULL	

PYTHON SOURCE CODE


```
from tkinter import *
from tkinter import ttk
from tkinter.ttk import Progressbar
from PIL import Image,ImageTk
import mysql.connector as connector
import os
root = Tk()
con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Login')
cur = con.cursor()
height = 400
width = 730
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y))
root.overrideredirect(True)
root.config(background="blue")
image = ImageTk.PhotoImage(Image.open("progress.png").resize((width,height)) )
bg_label= Label(root, image=image, bg="#2F6C60")
bg_label.place(x=0, y=0)
progress_label= Label(root, text="", font=("Ubuntu Mono", 1, "italic"), fg="#536ca6", bg="#536ca6")
progress_label.place (x=30, y=370)
progress=ttk.Style()
progress.theme_use('alt')
progress.configure ("red.Horizontal.TProgressbar", background='gold')
progress = Progressbar (root,orient=HORIZONTAL, length=730, mode='determinate',
style="red.Horizontal.TProgressbar")
progress.place(x=0, y=387)
def top():
   query ="select * from Login;"
```

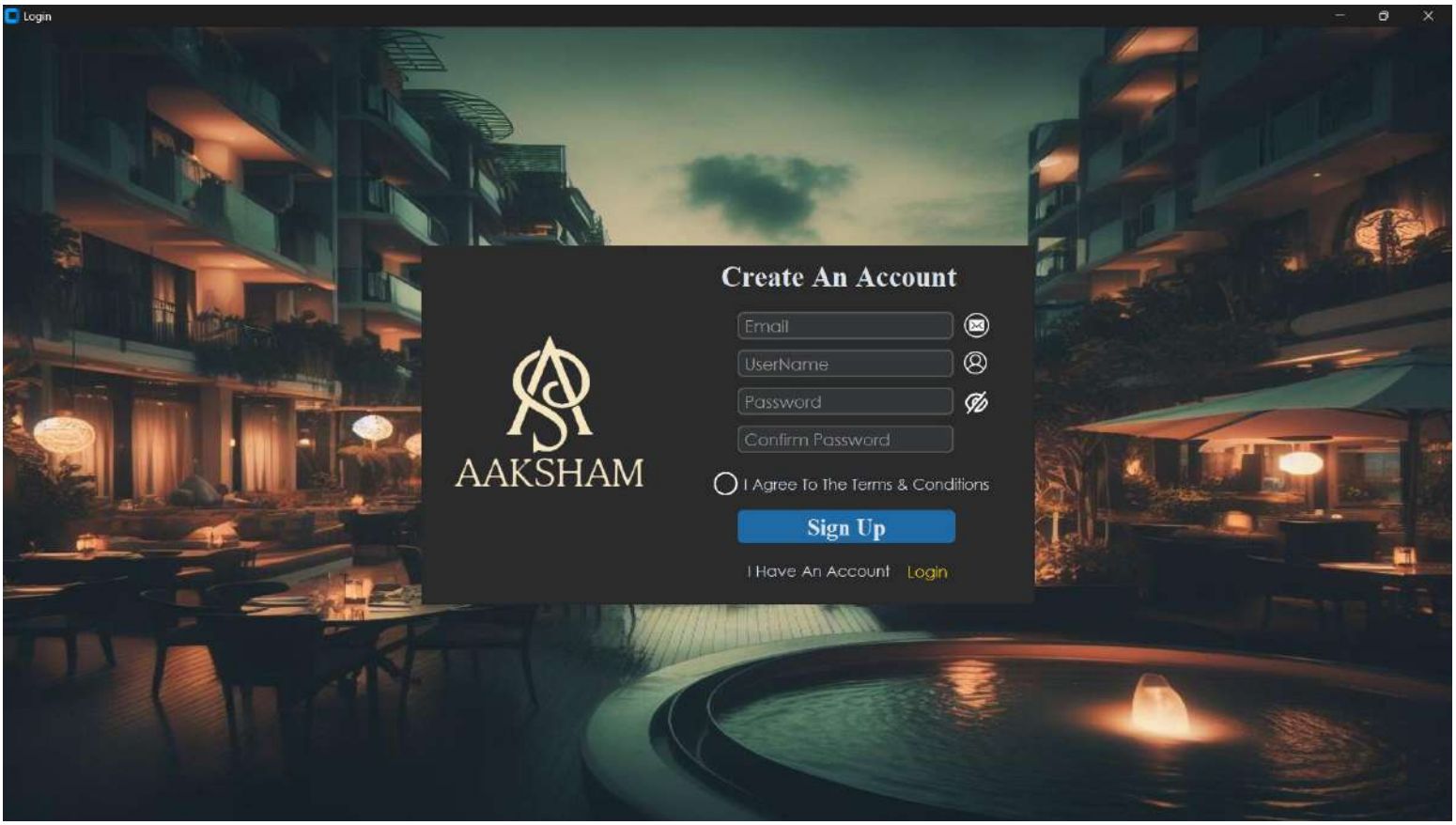
```
    cur.execute(query)
    l=[]
    for i in cur.fetchall():
        l.append(i[1:])
    # print(l)
    query ="select * from `Login Summary`;"
    cur.execute(query)
    j=[]
    for i in cur.fetchall():
        j.append(i)
    # print(j)
    mtch=False
    for i in j:
        for k in l:
            if i[0] == k[0] and i[1]==k[1]:
                mtch=True
    if mtch:
        root.destroy()
        os.system("python Main_BOX_TEST.py")
    else:
        root.destroy()
        os.system("python Login.py")
i = 0
def load():
    global i
    if i <= 100:
        txt = '' + (str(i)+'%')
        progress_label.config(text=txt)
        progress['value'] = i
        i += 1
        progress_label.after (50, load)
    else:
        top()
load()
root.mainloop()
```




```
import tkinter
import customtkinter
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import os
import random
import mysql.connector as connector
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
customtkinter.set_appearance_mode("Dark")  # Modes: system (default), light, dark
customtkinter.set_default_color_theme("blue")  # Themes: blue (default), dark-blue, green

app = customtkinter.CTk()
# creating cutstom tkinter window
height = 500
width = 930
x = (app.winfo_screenwidth()//2)-(width//2)
y = (app.winfo_screenheight()//2)-(height//2)
app.geometry('{}x{}+{}+{}'.format(width, height, x, y))
app.title('Login')

con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Login')
cur = con.cursor()

def button_function():
    # print(entry1.get())
    # print(entry2.get())
```

```
    if entry1.get()=="" or entry2.get()=="":
        messagebox.showerror("Error",f"Filed Complete All The Blanks ")
    else:
        # # print(ent.get())
        # # print(entry1.get())
        # app.destroy()  # destroy current window and creating new one
        # os.system("python Main_Box.py")
        cur.execute("select * from Login;")
        l = cur.fetchall()
        r=False
        for i in l:
            try:
                if entry1.get() == i[1] and entry2.get() == i[2]:
                    r=True
            except Exception as e:
                messagebox.showerror("Error", f"{e}")
        if r:
            query = f"insert into `Login Summary` values ('{entry1.get()}','{entry2.get()}');"
            try:
                cur.execute(query)
            except Exception as e:
                messagebox.showerror("Error",e)
            con.commit()
            con.close()
            app.destroy()
            os.system("python Main_BOX_TEST.py")

        else:
            messagebox.showerror("Error", "User Not Found")
img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=app,image=img1)
l1.pack()

# creating custom frame
# imgg = customtkinter.CTkImage(Image.open("").resize((20,20),Image.LANCZOS))
frame = customtkinter.CTkFrame(master=l1,width=650,height=380)
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

image = ImageTk.PhotoImage(Image.open("PhotoRoom-
20231022_184533.png").resize((266,213),Image.LANCZOS) )
iim = customtkinter.CTkLabel(master=frame, image=image,text="")
iim.place(x=30, y=90)
l2 = customtkinter.CTkLabel(master=frame,text="Log into your Account",font=('Times New
Roman',30,"bold"))
l2.place(x=300,y=45)
# customtkinter.CTkLabel(master=frame,text="User Name",font=('Century Gothic',18)).place(x=320,
y=85)
entry1 = customtkinter.CTkEntry(master=frame,font=('Century
Gothic',18),width=250,height=30,placeholder_text='Username')
entry1.place(x=320,y=110)
def hide():
    openeye.configure(image=imgg3)
    entry2.configure(show="*")
    openeye.configure(command=show)
```

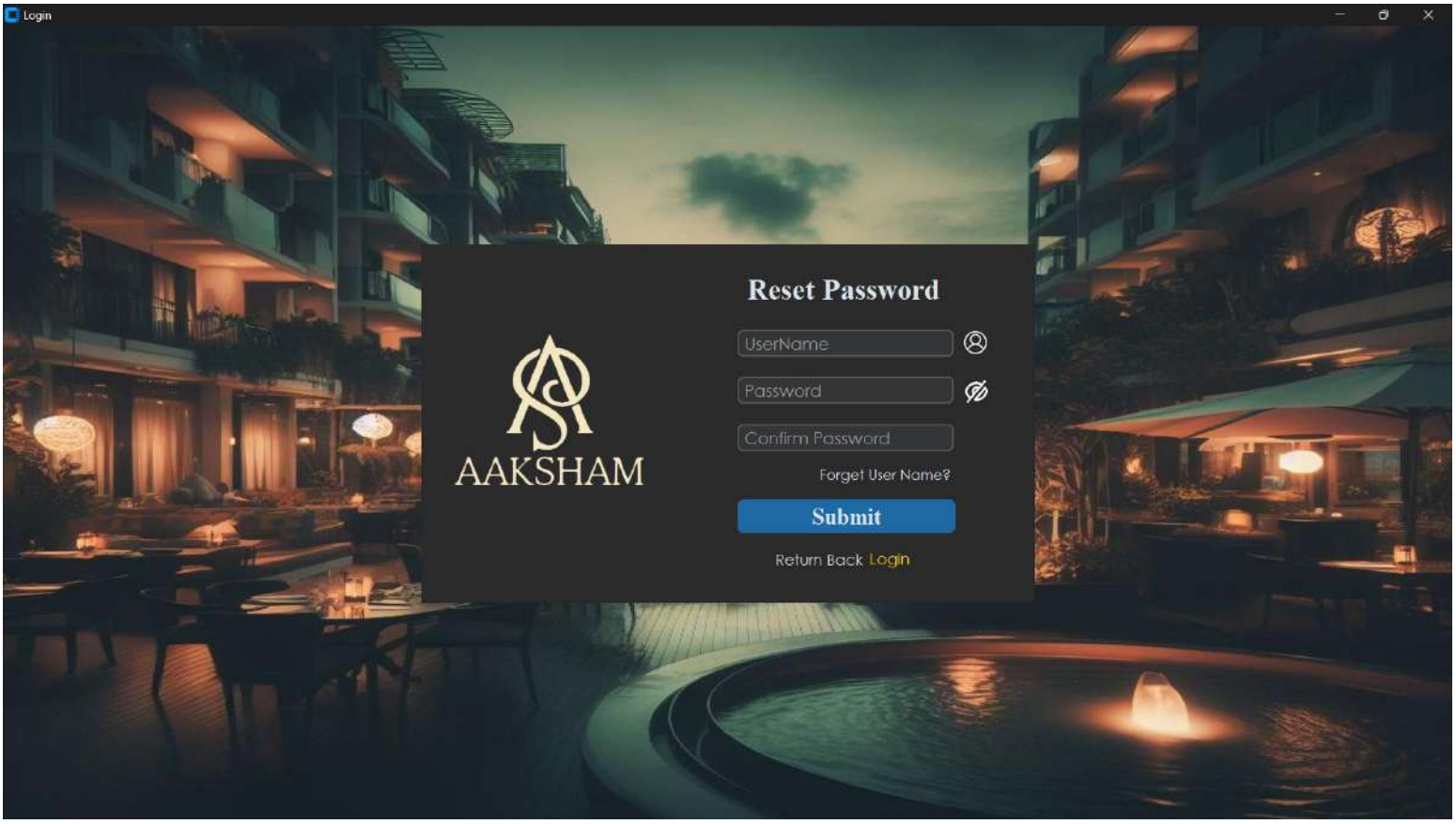
```
def show():
    openeye.configure(image=imgg2)
    entry2.configure(show="")
    openeye.configure(command=hide)
imgg1 = ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30)))
imgg2 = ImageTk.PhotoImage(Image.open("./images/openn.png").resize((30,30)))
imgg3 = ImageTk.PhotoImage(Image.open("./images/hide.png").resize((30,30)))
customtkinter.CTkLabel(master=frame, image=imgg1,text="").place(x=580, y=110)
openeye=customtkinter.CTkButton(master=frame,fg_color="#2b2b2b",image=imgg3,command=show,text
="",width=20,height=25,hover=False)
openeye.place(x=570,y=165)
# tkinter.Button(master=frame,image=img2,command=cc,activebackground="red").place(x=350,y=165)
entry2 = customtkinter.CTkEntry(master=frame,font=('Century
Gothic',18),width=250,height=30,placeholder_text='Password',show="*")
entry2.place(x=320,y=165)

dnt=customtkinter.CTkLabel(master=frame,text="Dont Have An Account ?",font=('Century Gothic',16))
dnt.place(x=320, y=310)

def ex():
    l1.configure(image=imgg)
    frame1.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
    l3.place(x=50,y=45-30)
tkinter.Button(master=frame,command=ex,text="Sign
Up",fg="gold",activeforeground="gold",relief="flat",activebackground="#2b2b2b",bg="#2b2b2b",cursor
="hand2",font=('Century Gothic',16)).place(x=650, y=385)
button1 = customtkinter.CTkButton(master=frame,font=('Times New
Roman',20,"bold"),width=250,text="Login",command=button_function,corner_radius=6)
button1.place(x=320,y=260)
```




```
imgg = ImageTk.PhotoImage(Image.open("./assets/7.jpg"))
frame1 = customtkinter.CTkFrame(master=l1,width=400,height=380)
l3 = customtkinter.CTkLabel(master=frame1,text="    Create An Account",font=('Times New
Roman',30,"bold"))
emailimgg = ImageTk.PhotoImage(Image.open("./images/email.png").resize((33,33)))
def hidee():
    openeyee.configure(image=imgg3)
    passs.configure(show="*")
    cnfpass.configure(show="*")
    openeyee.configure(command=showw)
def showw():
    openeyee.configure(image=imgg2)
    passs.configure(show="")
    cnfpass.configure(show="")
    openeyee.configure(command=hidee)
openeyee=customtkinter.CTkButton(master=frame1,fg_color="#2b2b2b",image=imgg3,command=showw,t
ext="",width=20,height=25,hover=False)
openeyee.place(x=333,y=180-30)
email = customtkinter.CTkEntry(master=frame1,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Email')
email.place(x=100,y=100-30)
# imgg1 = ImageTk.PhotoImage(Image.open("./images/user.png").resize((25,25)))
user = customtkinter.CTkEntry(master=frame1,font=('Century
Gothic',18),width=230,height=30,placeholder_text='UserName')
user.place(x=100,y=140-30)
customtkinter.CTkLabel(master=frame1,image=imgg1,text="").place(x=340,y=140-30)
customtkinter.CTkLabel(master=frame1,image=emailimgg,text="").place(x=340,y=100-30)
passs = customtkinter.CTkEntry(master=frame1,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Password',show="*")
passs.place(x=100,y=180-30)
cnfpass = customtkinter.CTkEntry(master=frame1,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Confirm Password',show="*")
```

```
cnfpass.place(x=100,y=220-30)
chk_terms=tkinter.StringVar(value=0)
chkbx=customtkinter.CTkCheckBox(master=frame1,font=('Century Gothic',16),text="I Agree To The
Terms &
Conditions",variable=chk_terms,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_ra
dius=25,fg_color="blue",border_color="white",hover=False,border_width=2)#checkmark_color="red"
chkbx.place(x=75,y=270-30)
def cmd():
   if email.get() == "" or user.get() == "" or passs.get() == "" or cnfpass.get() == "":
      messagebox.showinfo("Fill","Fill All The Blanks")
      # print("User Not Enter Complete Details")
   else:
      if passs.get() != cnfpass.get():
         messagebox.showerror("Error",f"Password And Confirm Password Are Not Same")
      else:
         if chkbx.get() == 0:
            # print(chkbx.get())
            messagebox.showinfo("Check","Ckeck Our Terms And Conditions")
            # print("Check Box Not Checked")
         else:
            query = f"insert into Login values('{email.get()}','{user.get()}','{passs.get()}');"
            print(query)
            cur.execute(query)
            con.commit()
            con.close()
            frame1.place(relx=5000,rely=5000)
            try:
               connect = smtplib.SMTP('smtp.gmail.com', 587)
               connect.ehlo()
               connect.starttls()
               sender_email = "aakshamhotel@gmail.com"
               sender_passwd = "kmko wohf Irdx gthw"
               connect.login(sender_email, sender_passwd)
               receiver_email = email.get()
               subject = "Aaksham Hotel Verification Details"
               msg_text = (f"User ID :- {user.get()}\n Password :- {passs.get()}\n Now You Are Registerd With
us...\nThanks & Regards \n Hotel Aaksham")
               message = MIMEMultipart()
               message["From"] = sender_email
               message["To"] = receiver_email
               message["Subject"] = subject
               message["Bcc"] = receiver_email
               message.attach(MIMEText(msg_text, "plain"))
               text = message.as_string()
               connect.sendmail(sender_email, receiver_email, text)
               print("Successfully email✉ sent")
            except Exception as e:
               messagebox.showerror("Error", e)
            finally:
               connect.quit()
customtkinter.CTkButton(master=frame1,font=('Times New Roman',25,"bold"),width=230,text="Sign
Up",command=cmd,corner_radius=6,height=30).place(x=100,y=310-30)
```

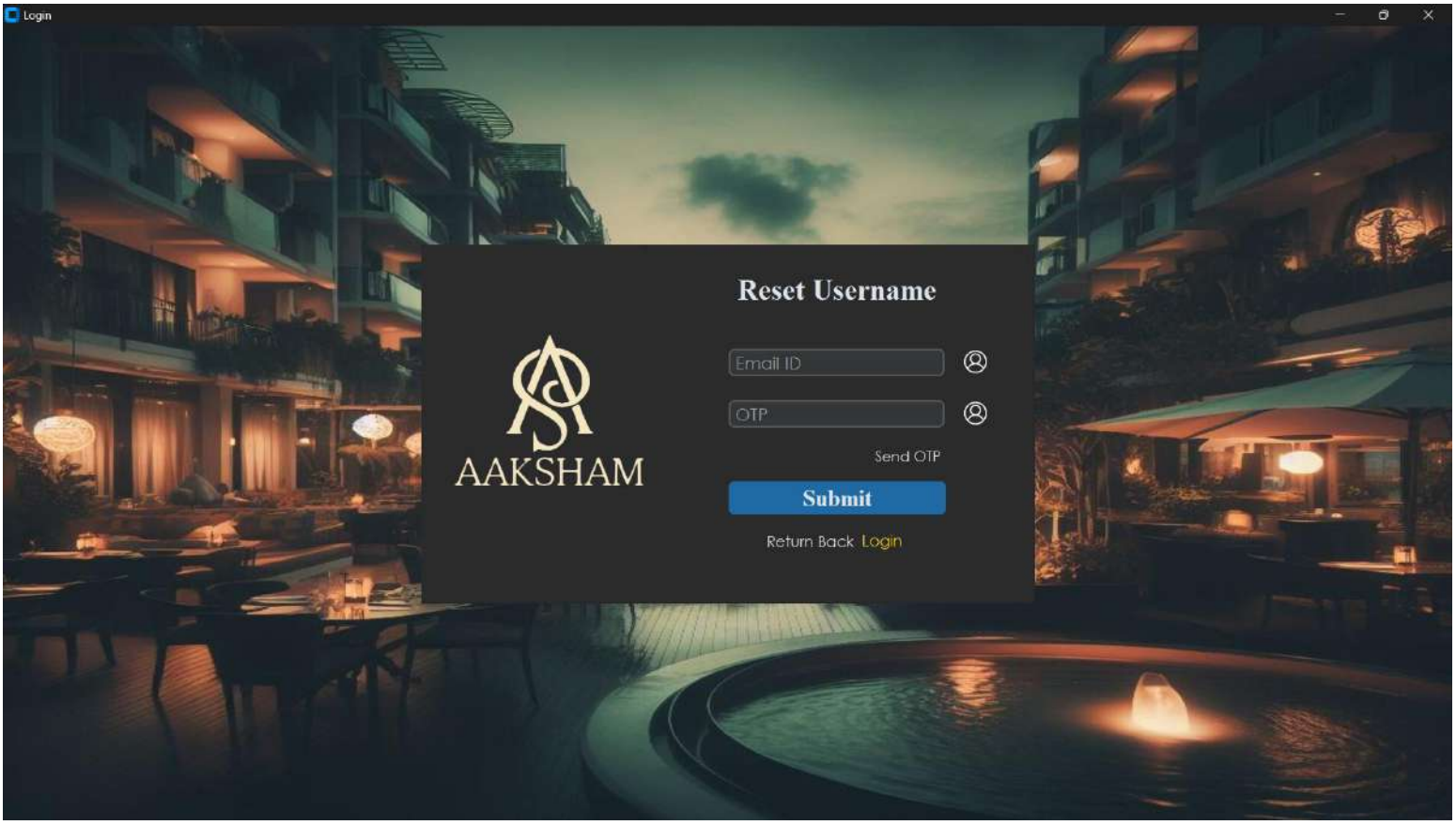
```
sign_UP_dnt=customtkinter.CTkLabel(master=frame1,text="I Have An Account",font=('Century
Gothic',16))
sign_UP_dnt.place(x=90+20, y=320+10)
def sign_UP_ex():
   frame1.place(relx=5000, rely=5000)
tkinter.Button(master=frame1,command=sign_UP_ex,text="Login",fg="gold",activeforeground="gold",rel
ief="flat",activebackground="#2b2b2b",bg="#2b2b2b",cursor="hand2",font=('Century
Gothic',16)).place(x=340, y=410)
```




```
#----------------------  FORGET PASSWORD  ------------------------------------------------------------------------

frame2 = customtkinter.CTkFrame(master=l1,width=400,height=380)
# frame2.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
def forgot():
   l1.configure(image=imgg)
   frame2.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
def forget_command():
   if forget_user.get() == "" or forget_cnf_password.get() == "" or forget_new_password.get() == "":
      messagebox.showinfo("Fill","Fill All The Blanks")
      # print("User Not Enter Complete Details")
   else:
      if forget_new_password.get() != forget_cnf_password.get():
         messagebox.showerror("Error",f"New Password And Confirm Password Are Not Same")
      else:
         forget_popup=messagebox.askyesno("confirm","Are You Sure You Want To Change Password")
         # print(forget_popup)
         if forget_popup:
            query=f"update Login set Password = '{forget_new_password.get()}' where
UserName='{forget_user.get()}'"
            try :
               cur.execute(query)
            except Exception as e:
```

```
            messagebox.showerror("Error",e)
        con.commit()
        con.close()
        frame2.place(relx=5000,rely=5000)
    else:
        pass
forget = customtkinter.CTkButton(master=frame,text="Forget password?",
                    font=('Century Gothic',15),
                    width=100,
                    corner_radius=6,
                    fg_color="#2b2b2b",
                    # hover=False,
                    command=forgot,hover_color="#3e3c3c")
forget.place(x=425,y=210)


customtkinter.CTkLabel(master=frame2,
image=ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30))),text="").place(x=340,
y=90)

def forget_ex():
    frame2.place(relx=5000, rely=5000)
tkinter.Button(master=frame2,command=forget_ex,text="Login",fg="gold",activeforeground="gold",relief
="flat",activebackground="#2b2b2b",bg="#2b2b2b",cursor="hand2",font=('Century
Gothic',16)).place(x=230+60, y=395)
def forget_hide():
    forget_eye.configure(image=imgg3)
    forget_new_password.configure(show="*")
    forget_cnf_password.configure(show="*")
    forget_eye.configure(command=forget_show)
def forget_show():
    forget_eye.configure(image=imgg2)
    forget_new_password.configure(show="")
    forget_cnf_password.configure(show="")
    forget_eye.configure(command=forget_hide)
forget_eye=customtkinter.CTkButton(master=frame2,fg_color="#2b2b2b",image=imgg3,command=forget_s
how,text="",width=20,height=25,hover=False)
forget_eye.place(x=333,y=140)


customtkinter.CTkLabel(master=frame2,text="Reset Password",font=('Times New
Roman',30,"bold")).place(x=110,y=30)
forget_dnt=customtkinter.CTkLabel(master=frame2,text="Return Back",font=('Century Gothic',16))
forget_dnt.place(x=90+50, y=320)
forget_user = customtkinter.CTkEntry(master=frame2,font=('Century
Gothic',18),width=230,height=30,placeholder_text='UserName')
forget_user.place(x=100,y=110-20)
forget_new_password = customtkinter.CTkEntry(master=frame2,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Password',show="*")
forget_new_password.place(x=100,y=150+10-20)
forget_cnf_password = customtkinter.CTkEntry(master=frame2,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Confirm Password',show="*")
forget_cnf_password.place(x=100,y=210-20)
forget_button =customtkinter.CTkButton(master=frame2,font=('Times New
```

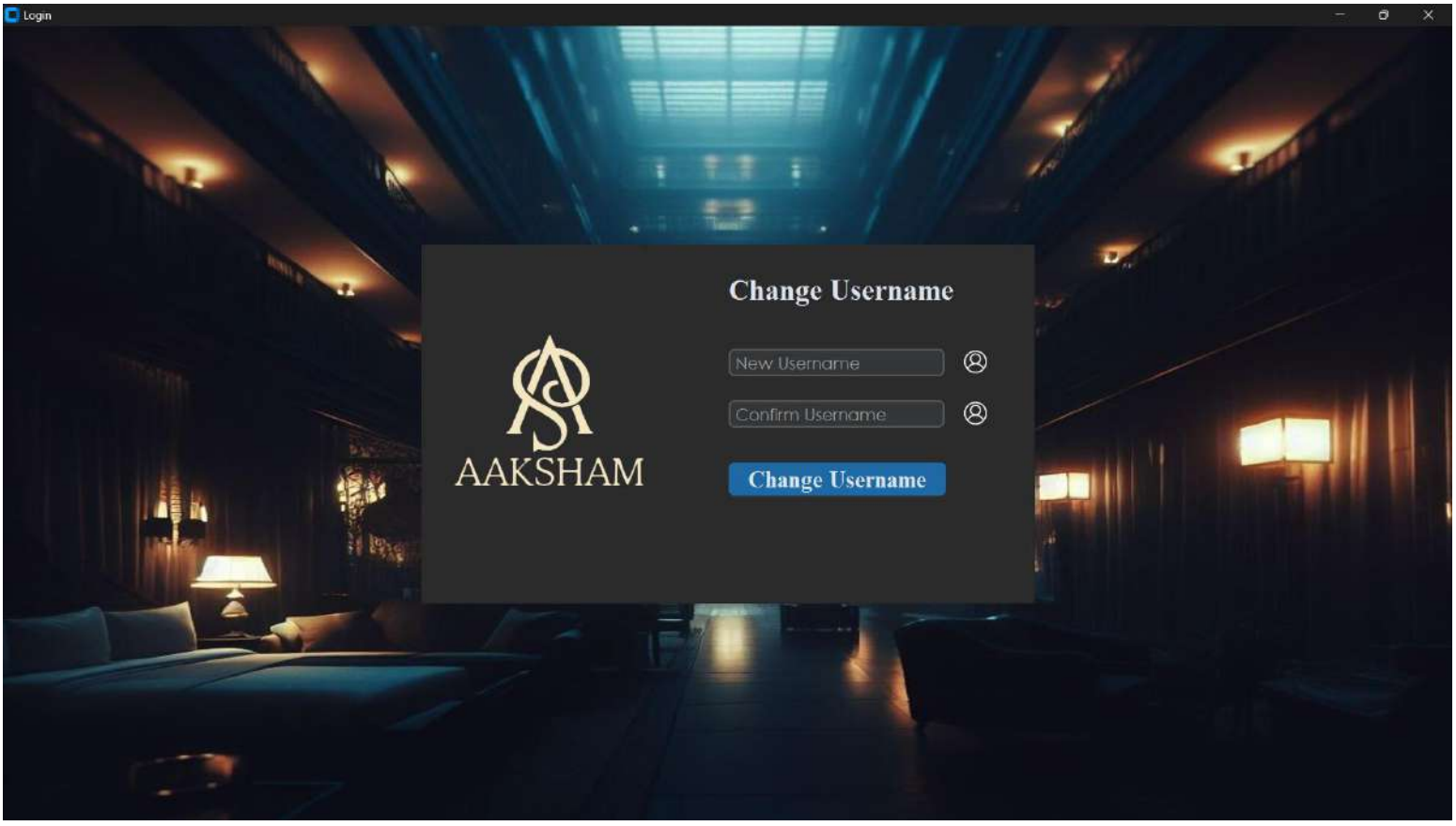
```
Roman',25,"bold"),width=230,text="Submit",command=forget_command,corner_radius=6,height=30)
forget_button.place(x=100,y=270)
def forgotUserNm():
   l1.configure(image=imgg)
   frame3.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
customtkinter.CTkButton(master=frame2,text="Forget User Name?",
                        font=('Century Gothic',15),
                        width=100,
                        corner_radius=6,
                        fg_color="#2b2b2b",
                        # hover=False,
                        command=forgotUserNm,hover_color="#3e3c3c").place(x=180,y=230)
```




```
#---------------------------------------------- Forget User Name ----------------------------------------------
frame3 = customtkinter.CTkFrame(master=l1,width=400,height=380)
# frame3.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
customtkinter.CTkLabel(master=frame3,text="Reset Username",font=('Times New
Roman',30,"bold")).place(x=100,y=30)

customtkinter.CTkLabel(master=frame3,
image=ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30))),text="").place(x=340,
y=110)
customtkinter.CTkLabel(master=frame3,
image=ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30))),text="").place(x=340,
y=165)

UserNmentry = customtkinter.CTkEntry(master=frame3,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Email ID')
UserNmentry.place(x=90,y=110)
UserNmentry1 = customtkinter.CTkEntry(master=frame3,font=('Century
Gothic',18),width=230,height=30,placeholder_text='OTP')
UserNmentry1.place(x=90,y=165)
```

```
OTP=random.randint(100000,999999)
def Sndotp():
   if UserNmentry.get()[-10:] != "@gmail.com":
      messagebox.showerror("Invalid","Invalid Mail Id")
   else:
      con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Login')
      cur = con.cursor()
      print(UserNmentry.get())
      query =f"select * from Login where `Email ID`='{UserNmentry.get()}'"
      cur.execute(query)
      Data=cur.fetchone()
      print(Data)
      if Data == None:
         messagebox.showinfo("Reset Username", "No Username Found With This Email ID")
      else:
         messagebox.showinfo("Resetting Username", "An OTP Is Send To Given Mail ID")
         try:
            connect = smtplib.SMTP('smtp.gmail.com', 587)
            connect.ehlo()
            connect.starttls()
            sender_email = “aakshamhotel@gmail.com”
            sender_passwd = “kmko wohf Irdx gthw”
            connect.login(sender_email, sender_passwd)
            receiver_email = Data[0]
            subject = "Aaksham Hotel Verification Code"
            msg_text = (f"Your OTP For Reseting Username Is : - {OTP}")
            message = MIMEMultipart()
            message["From"] = sender_email
            message["To"] = receiver_email
            message["Subject"] = subject
            message["Bcc"] = receiver_email
            message.attach(MIMEText(msg_text, "plain"))
            text = message.as_string()
            connect.sendmail(sender_email, receiver_email, text)
            print("Successfully email✉ sent")
         except Exception as e:
            messagebox.showerror("Error", e)
         finally:
            connect.quit()
customtkinter.CTkButton(master=frame3,text="Send OTP",
                        font=('Century Gothic',15),
                        width=100,
                        corner_radius=6,
                        fg_color="#2b2b2b",
                        # hover=False,
                        command= Sndotp,hover_color="#3e3c3c").place(x=230,y=210)
frame4 = customtkinter.CTkFrame(master=l1,width=400,height=380)

customtkinter.CTkLabel(master=frame3,text="Return Back",font=('Century Gothic',16)).place(x=90+40,
```

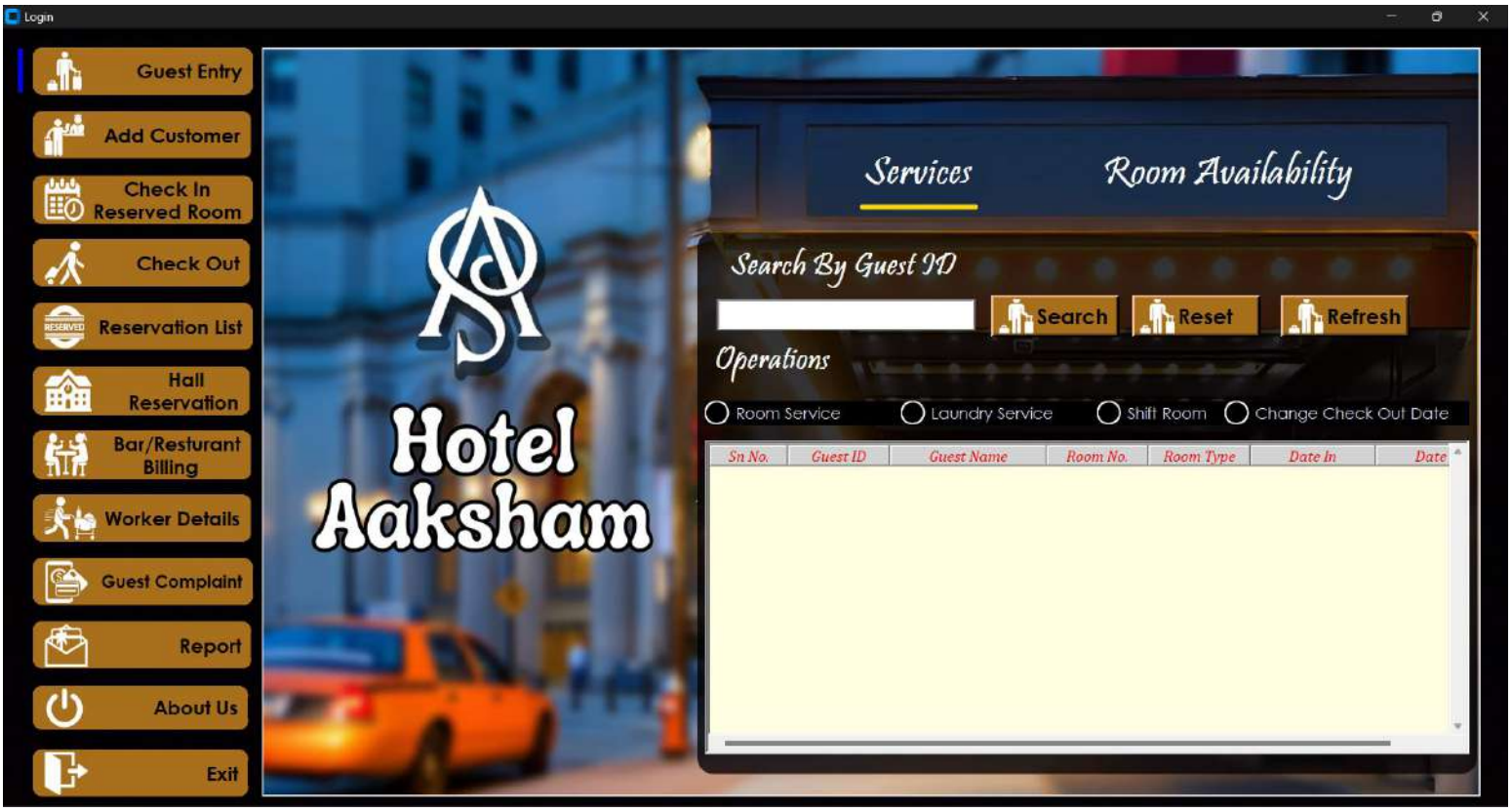
```
y=300)
def forget_ex():
    frame2.place(relx=5000, rely=5000)
    frame3.place(relx=5000, rely=5000)
tkinter.Button(master=frame3,command=forget_ex,text="Login",fg="gold",activeforeground="gold",relief
="flat",activebackground="#2b2b2b",bg="#2b2b2b",cursor="hand2",font=('Century
Gothic',16)).place(x=230+50, y=370)
def ChgUsrNm():
    try:
        print(OTP)
        print(UserNmentry1.get())
        if int(UserNmentry1.get()) == OTP:
            l1.configure(image=imgg)
            frame4.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
        else:
            messagebox.showerror("Error","OTP Doest Not Match")
    except Exception as e :
        messagebox.showerror("Error","Please Input Details Firstly")
UserNmforget_button =customtkinter.CTkButton(master=frame3,font=('Times New
Roman',25,"bold"),width=230,text="Submit",command=ChgUsrNm,corner_radius=6,height=30)
UserNmforget_button.place(x=90,y=250)
```




```
#------------------------------------------------------------ Change Username  ----------------------------------------

frame4 = customtkinter.CTkFrame(master=l1,width=400,height=380)
# frame4.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER,x=110)
customtkinter.CTkLabel(master=frame4,
image=ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30))),text="").place(x=340,
y=110)
customtkinter.CTkLabel(master=frame4,
image=ImageTk.PhotoImage(Image.open("./images/user.png").resize((30,30))),text="").place(x=340,
y=165)
```

```
CngUsrNmLbl = customtkinter.CTkLabel(master=frame4,text="Change Username",font=('Times New
Roman',30,"bold"))
CngUsrNmLbl.place(x=90,y=30)
CngUsrNmEntry = customtkinter.CTkEntry(master=frame4,font=('Century
Gothic',18),width=230,height=30,placeholder_text='New Username')
CngUsrNmEntry.place(x=90,y=110)
CngUsrNmEntry1 = customtkinter.CTkEntry(master=frame4,font=('Century
Gothic',18),width=230,height=30,placeholder_text='Confirm Username')
CngUsrNmEntry1.place(x=90,y=165)
def ChgUsrNm():
   if CngUsrNmEntry.get()==CngUsrNmEntry1.get():
      con = connector.connect(host='localhost',
                  port='3306',
                  user='root',
                  password='Password',
                  database='Login')
      cur = con.cursor()
      query=f"update login set UserName = '{CngUsrNmEntry.get()}' where `Email
ID`='{UserNmentry.get()}'"
      cur.execute(query)
      con.commit()
      messagebox.showinfo("Resetting Username","You UserName Is Changed")
      frame2.place(relx=5000, rely=5000)
      frame3.place(relx=5000, rely=5000)
      frame4.place(relx=5000, rely=5000)
   else:
      messagebox.showerror("Resetting Username","Username And Confirm Username are Not Same")
UserNmforget_button =customtkinter.CTkButton(master=frame4,font=('Times New
Roman',25,"bold"),width=230,text="Change
Username",command=ChgUsrNm,corner_radius=6,height=30)
UserNmforget_button.place(x=90,y=230)
```

```
# ----------------------------------------   Mysql Query ----------------------------------------------------------------
create database login;
use Login;
create table Login(
`Email ID` varchar(40) unique not null,
UserName varchar(20) unique not null,
Password varchar(20) unique not null
);
create table `Login Summary`(
UserName varchar(20) unique not null,
Password varchar(20) unique not null
);
```




```
import time
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import filedialog
from tkinter import  messagebox
from PIL import ImageTk,Image
from tkcalendar import DateEntry
from datetime import date
import datetime
import pandas as pd
```

```
import mysql.connector as connector
import os
import pygame
customtkinter.set_appearance_mode("dark")
app = customtkinter.CTk()
app.geometry('{}x{}+0+0'.format(app.winfo_screenwidth(),app.winfo_screenheight()))
app.title('Login')
# app.state("zoomed")
img1 = ImageTk.PhotoImage(Image.open("./assets/Black_colour.jpg"))
# img1=customtkinter.CTkImage(Image.open("./assets/8.jpg"))
l1 = customtkinter.CTkLabel(master=app,image=img1,text="")
l1.pack()
f1=customtkinter.CTkFrame(master=l1,fg_color="blue",width=5,height=45)
f1.place(x=15,y=21)
pygame.mixer.init()
pygame.mixer.music.load("Welcome.mp3")
def Guestt_Entry():
    ent.set("")
    chk_terms.set(value=0)
    Chk_Ot.set(value=0)
    Rm_Serve.set(value=0)
    Ldry_Serve.set(value=0)
    ShftRm.set(value=0)
    Cng_Date.set(value=0)
    Rm_Reverse.set(value=0)
    Chk_Rm.set(value=0)
    Rsvrent.set("")
    Rsvrent1.set("")
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    for tem in Aval.get_children():
        Aval.delete(tem)
    S_No = 1
    query = "select * from `Room Status` where `Status` not in ('Dirty','Repair') order by `Room No.` asc;"
    cur.execute(query)
    for i in cur.fetchall():
        Aval.insert("", END, values=(S_No, i[0], i[1], i[3]))
        S_No += 1
    for tm in ChkAval.get_children():
        ChkAval.delete(tm)
    S_No = 1
    query = "select * from `Room Status` where `Status`!= 'vacant' order by `Room No.` asc ;"
    cur.execute(query)
    for i in cur.fetchall():
        ChkAval.insert("", END, values=(S_No, i[0], i[1], i[2]))
        S_No += 1
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
```

```
                    database='Hotel Management Software')
    cur = con.cursor()
    for item in table.get_children():
        table.delete(item)
    query = "select * from `Check In Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(sn, row[0], row[1], row[2], row[3], row[4], row[5]))
        sn += 1
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Guest Entry.mp3")
    pygame.mixer.music.play()
def Ad_Mmber():
    con = connector.connect(host='localhost',
                    port='3306',
                    user='root',
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    Gs_val = pd.read_csv("Gs_ID.csv", index_col=[0])
    Gs_ENTRY_ID.set(value=Gs_val.Gs_Entry[1])
    # val3.close()
    Gs_ENTRY_Nm.set(value="")
    Gs_ENTRY_der.set(value="")
    Gs_ENTRY_gion.set(value="")
    Gs_ENTRY_Address.set(value="")
    Gs_ENTRY_City.set(value="")
    Gs_ENTRY_Cntry.set(value="")
    Gs_ENTRY_CntNO.set(value="")
    Gs_ENTRY_IDType.set(value="")
    Gs_ENTRY_IDNo.set(value="")
    Gs_ENTRY_EmailID.set(value="")
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
    con = connector.connect(host='localhost',
                    port='3306',
                    user='root',
```

```
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    for item in Gs_Entry_Tabke.get_children():
        Gs_Entry_Tabke.delete(item)
    query = "select * from `Customer Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        Gs_Entry_Tabke.insert("", END, values=(
        sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12],
row[13]))
        sn += 1
    f1.place(x=15,y=86)
    can_widgett.place(x=1000,y=1000)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=330, y=25)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Add Customer.mp3")
    pygame.mixer.music.play()
def Checkk_Out():
    global ChkotCstmrImg
    Rm.set("")
    RType.set("")
    DateIn.set("")
    DateOut.set("")
    RoomCharge.set("")
    Discount_Price.set("")
    SGST_Price.set("")
    CGST_Price.set("")
    GrandTotal.set("")
    AdvancePayment.set("")
    RemainingAmnt.set("")
    Pymnt.set("")
    GsID.set("")
    GsNm.set("")
    Gsder.set("")
    Gsgion.set("")
    GsAddress.set("")
    GsCity.set("")
    GsCntry.set("")
    GsCntNO.set("")
    GsIDType.set("")
```

```
    GsIDNo.set("")
    rsvrd_pymnt.set("")
    ChkotCstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    ChkotImglbl.configure(image=ChkotCstmrImg)
    f1.place(x=15,y=216)
    can_widget1.place(x=330, y=25)
    can_widgett.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Check Out.mp3")
    pygame.mixer.music.play()
def Chkout_Cstmr_Dtl():
    ChkoutE1Var.set("")
    ChkoutE2Var.set("")
    ChkoutE3Var.set("")
    for item in Chk_Gst_Dtl_Trvw.get_children():
        Chk_Gst_Dtl_Trvw.delete(item)
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    query = "select * from chkout;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Chk_Gst_Dtl_Trvw.insert("", END,
                                values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                        row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16], row[17],
                                        row[18], row[19], row[20], row[21], row[22], row[23], row[24], row[25]))
        sn += 1
    f1.place(x=15,y=216)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
```

```
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget13.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Check Out Details.mp3")
    pygame.mixer.music.play()
def Chk_INN():
    Rsvrd_Gst_GsID.set("")
    Rsvrd_Gst_GsNm.set("")
    Rsvrd_Gst_Gsder.set("")
    Rsvrd_Gst_Gsgion.set("")
    Rsvrd_Gst_GsAddress.set("")
    Rsvrd_Gst_GsCity.set("")
    Rsvrd_Gst_GsCntry.set("")
    Rsvrd_Gst_GsCntNO.set("")
    Rsvrd_Gst_GsIDType.set("")
    Rsvrd_Gst_GsIDNo.set("")
    Rsvrd_Gst_Rm.set("")
    Rsvrd_Gst_Rm_Type.set("")
    din.set("")
    dout.set("")
    Rsvrd_Gst_rm_price.set("")
    Rsvrd_Gst_Pymnt.set("")
    Rsvrd_Gst_mydata.set("Cash")
    Rsvrd_Gst_cal.set_date(date.today())
    f1.place(x=15, y=151)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=330, y=25)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Check In Reserved Room.mp3")
    pygame.mixer.music.play()
def Halll_Reservation():
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm .set("")
    Scrh_Hall_by_Address .set("")
    Scrh_Hall_by_Cnt_No .set("")
    Scrh_Hall_by_ID_Type .set("")
```

```
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
        Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
        Hall_resevr_TreeYu.insert("", END, values=(
            sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
        sn += 1
    f1.place(x=15,y=346)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=330, y=25)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Hall Reservation.mp3")
    pygame.mixer.music.play()
def Hll_Grd_Billingg():
    global HallCstmrBillImg
    Hallgd_Gst_GsID.set("")
    Hallgd_Gst_GsNm.set("")
    Hallgd_Gst_GsAddress.set("")
    Hallgd_Gst_GsCntNO.set("")
    Hallgd_Gst_GsIDType.set("")
    Hallgd_Gst_GsIDNo.set("")
    Hallgd_Gst_GsEmil_ID.set("")
    Hallgd_Gst_Rm.set("")
    Hallgd_Gst_rm_price.set("")
    Hallgd_Gst_Ttl_Price.set("")
    dtin.set("")
    dtout.set("")
    HallCstmrBillImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 90)))
    HallBillImgBtn.configure(image=HallCstmrBillImg)
```

```
    Hallgd_Gst_Adv_Price.set("")
    Hallgd_Gst_Pymnt.set("")
    Hallgd_Gst_mydata.set("Cash")
    f1.place(x=15, y=346)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget6.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Hall Billing.mp3")
    pygame.mixer.music.play()
def Hll_Details():
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    E1Var.set("")
    E2Var.set("")
    E3Var.set("")
    for item in Hall_Gst_Dtl_Trvw.get_children():
        Hall_Gst_Dtl_Trvw.delete(item)
    query = f"select * from `Hall Customer Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Hall_Gst_Dtl_Trvw.insert("", END,
                                 values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                         row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[11],
                                         row[12], row[13], row[14], row[15], row[16].strftime("%d/%m/%y")))
        sn += 1
    f1.place(x=15, y=346)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
```

```
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget12.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Hall Details.mp3")
    pygame.mixer.music.play()
def Reservationn_List():
    ResvrGS_ID.set("")
    ResvrGS_Nm.set("")
    ResvrGS_Rm_No.set("")
    ResvrGS_Cnt_No.set("")
    for item in Rm_resevr.get_children():
        Rm_resevr.delete(item)
    query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Rm_resevr.insert("", END, values=(
        sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10],row[12
        ], row[15], row[16], row[17],
        row[18], row[19], row[20]))
        sn+=1

    f1.place(x=15,y=281)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=330, y=25)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Reservation List.mp3")
    pygame.mixer.music.play()
def Barr_Resturant():
    Brrestornt_Gst_GsID.set("")
    Brrestornt_Gst_GsNm.set("")
    Brrestornt_Gst_Gsder.set("")
    Brrestornt_Gst_GsCntNO.set("")
    for item in barfd.get_children():
        barfd.delete(item)
    Brrestornt_Gst_GsID.set("")
    Brrestornt_Gst_GsNm.set("")
    Brrestornt_Gst_Gsder.set("")
```

```
    Brrestornt_Gst_GsCntNO.set("")
    bar_FdNm.set("")
    barrat.set(0)
    baramt.set(0)
    bardstprice.set(0)
    barss_GST.set(0)
    barcc_GST.set(0)
    barTl_Amnt.set(0)
    f1.place(x=15,y=411)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget7.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Bar And Restaurant.mp3")
    pygame.mixer.music.play()
def Wk_Details():
    wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
    Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
    Wk_Lst_Nm.set(value="")
    Wk_Lst_der.set(value="")
    Wk_Lst_gion.set(value="")
    Wk_Lst_Address.set(value="")
    Wk_Lst_Cntry.set(value="")
    Wk_Lst_CntNO.set(value="")
    Wk_Lst_IDType.set(value="")
    Wk_Lst_IDNo.set(value="")
    Dept.set(value="")
    for item in Wk_Lst_Tabke.get_children():
        Wk_Lst_Tabke.delete(item)
    query = "select * from Worker_Details;"
    cur.execute(query)
    sn = 1
    for i in cur.fetchall():
        Wk_Lst_Tabke.insert("", END, values=(sn, i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9]))
        sn += 1
    f1.place(x=15, y=476)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
```

```
    can_widget7.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget8.place(x=330, y=25)
    # can_widget9.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Worker Details.mp3")
    pygame.mixer.music.play()
def Gst_Cmplnt():
    Gst_cmpnt_ID.set(value="")
    Gst_cmpnt_Nm.set(value="")
    Gst_cmpnt_CntNO.set(value="")
    Gst_cmpnt_Cmplnt.set(value="")
    f1.place(x=15,y=541)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget9.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/Guest Complaints.mp3")
    pygame.mixer.music.play()
def ReportCmd():
    f1.place(x=15,y=606)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget10.place(x=330, y=25)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    pygame.mixer.music.load("./Voices/Report.mp3")
    pygame.mixer.music.play()
def AbtUs():
```

```
    f1.place(x=15,y=671)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=1000, y=1000)
    can_widget11.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/About us.mp3")
    pygame.mixer.music.play()
def AbtOurMmbr():
    f1.place(x=15,y=671)
    can_widgett.place(x=1000, y=1000)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
    can_widget14.place(x=330, y=25)
    pygame.mixer.music.load("./Voices/About Team Members.mp3")
    pygame.mixer.music.play()
def Ext():
    if messagebox.askyesno("Exit","Are You Sure You Want To Exit"):
        pygame.mixer.init()
        pygame.mixer.music.load("./Voices/Exit.mp3")
        pygame.mixer.music.play()
        time.sleep(1)
        exit()
def Sign_Outt():
    if messagebox.askyesno("SignOut","Are You Sure You Want To Sign Out"):
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Login')
        cur = con.cursor()
        query = "delete from `Login Summary`;"
        cur.execute(query)
```

```
    con.commit()
    pygame.mixer.music.load("./Voices/Sign Out.mp3")
    pygame.mixer.music.play()
    time.sleep(1)
    exit()


con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((50,50)))
Check_in = ImageTk.PhotoImage(Image.open("./images/Check_In.png").resize((50,55)))
Check_Out = ImageTk.PhotoImage(Image.open("./images/Check_Out.png").resize((50,55)))
Room_Reservation =
ImageTk.PhotoImage(Image.open("./images/Room_Reservation.png").resize((50,55)))
Hall_Reservation = ImageTk.PhotoImage(Image.open("./images/Hall_Reservation.png").resize((60,55)))
Reservation_List = ImageTk.PhotoImage(Image.open("./images/Reservation_List.png").resize((50,55)))
Room_Service = ImageTk.PhotoImage(Image.open("./images/Room_Service.png").resize((65,55)))
Bar_Resturant = ImageTk.PhotoImage(Image.open("./images/Bar_Resturant.png").resize((55,55)))
Hll_Grd_Billing = ImageTk.PhotoImage(Image.open("./images/Voucher.png").resize((55,55)))
Payment = ImageTk.PhotoImage(Image.open("./images/Payment.png").resize((55,55)))
Sign_Out = ImageTk.PhotoImage(Image.open("./images/Sign_Out.png").resize((50,50)))
Exit = ImageTk.PhotoImage(Image.open("./images/Exit.png").resize((55,55)))


customtkinter.CTkButton(master=l1,image=Guest_Entry,compound=LEFT,text_color="Black",width=200,
height=40,text="       Guest
Entry",corner_radius=10,fg_color="#a8701d",bg_color="black",anchor=W,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Guestt_Entry).place(x=30,y=20)


customtkinter.CTkButton(master=l1,image=Check_in,compound=LEFT,text_color="Black",width=200,hei
ght=40,text="  Add
Customer",corner_radius=10,fg_color="#a8701d",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",bg_color="black",command=Ad_Mmber).place(x=30,y=85)

customtkinter.CTkButton(master=l1,text="Check In\nReserved
Room",image=Room_Reservation,compound=LEFT,text_color="Black",width=210,height=40,corner_radi
us=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Chk_INN).place(x=30,y=150)

ChkOutBtn=customtkinter.CTkButton(master=l1,text="       Check
Out",image=Check_Out,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10,fg_
color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Checkk_Out)
ChkOutBtn.place(x=30,y=215)

ChkOutDtlFrm=tkinter.Frame(width=330,background="black",height=80)

customtkinter.CTkButton(master=ChkOutDtlFrm,text="Check Out
```

```
Details",image=Hll_Grd_Billing,compound=LEFT,text_color="Black",width=200,height=40,corner_radius
=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Chkout_Cstmr_Dtl).place(x=5,y=8)

customtkinter.CTkButton(master=l1,text=" Reservation
List",image=Reservation_List,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=1
0,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Reservationn_List).place(x=30,y=280)
def on_hover(e):
   ChkOutDtlFrm.place(x=320,y=259)
def on_exit(e):
   # print("Chala Gaya")
   time.sleep(0.1)
   ChkOutDtlFrm.place(x=5000, y=5000)
ChkOutBtn.bind('<Enter>',on_hover)
ChkOutBtn.bind('<Leave>',on_exit)
ChkOutDtlFrm.bind('<Enter>',on_hover)
ChkOutDtlFrm.bind('<Leave>',on_exit)

HallRsvr=customtkinter.CTkButton(master=l1,text="     Hall\n
Reservation",image=Hall_Reservation,compound=LEFT,text_color="Black",width=200,height=40,corner_
radius=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Halll_Reservation)
HallRsvr.place(x=30,y=345)
Hvrfrm=tkinter.Frame(width=270,background="black",height=170)
# Hvrfrm.place(x=300,y=420)
customtkinter.CTkButton(master=Hvrfrm,text="Hall
Billing",image=Hll_Grd_Billing,compound=LEFT,text_color="Black",width=200,height=40,corner_radius
=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Hll_Grd_Billingg).place(x=5,y=8)
customtkinter.CTkButton(master=Hvrfrm,text="Hall
Details",image=Hll_Grd_Billing,compound=LEFT,text_color="Black",width=200,height=40,corner_radius
=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Hll_Details).place(x=5,y=75)
def on_hover(e):
   Hvrfrm.place(x=320,y=420)
def on_exit(e):
   # print("Chala Gaya")
   time.sleep(0.1)
   Hvrfrm.place(x=5000, y=5000)
HallRsvr.bind('<Enter>',on_hover)
HallRsvr.bind('<Leave>',on_exit)
Hvrfrm.bind('<Enter>',on_hover)
Hvrfrm.bind('<Leave>',on_exit)

customtkinter.CTkButton(master=l1,text="  Bar/Resturant\n
Billing",image=Bar_Resturant,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=1
0,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Barr_Resturant).place(x=30,y=410)

customtkinter.CTkButton(master=l1,text="Worker
Details",image=Room_Service,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=
10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Wk_Details).place(x=30,y=475)
```

```
customtkinter.CTkButton(master=l1,text=" Guest
Complaint",image=Payment,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10
,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',18,"bold"),hover_color="#b68339",command=Gst_Cmplnt).place(x=30,y=540)

customtkinter.CTkButton(master=l1,text="
Report",image=Hll_Grd_Billing,compound=LEFT,text_color="Black",width=200,height=40,corner_radius
=10,fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=ReportCmd).place(x=30,y=605)
Abtusfrm=tkinter.Frame(width=370,background="black",height=75)

AbtusBtn=customtkinter.CTkButton(master=l1,text="          About
Us",image=Sign_Out,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10,fg_col
or="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=AbtUs)
AbtusBtn.place(x=30,y=670)
customtkinter.CTkButton(master=Abtusfrm,text="About Team
Members",image=Sign_Out,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10,
fg_color="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=AbtOurMmbr).place(x=6,y=8)

def Abtus_on_hover(e):
   Abtusfrm.place(x=320,y=830)
def Abtus_on_exit(e):
   # print("Chala Gaya")
   time.sleep(0.1)
   Abtusfrm.place(x=5000, y=5000)
AbtusBtn.bind('<Enter>',Abtus_on_hover)
AbtusBtn.bind('<Leave>',Abtus_on_exit)
Abtusfrm.bind('<Enter>',Abtus_on_hover)
Abtusfrm.bind('<Leave>',Abtus_on_exit)

ExitBtn=customtkinter.CTkButton(master=l1,text="
Exit",image=Exit,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10,fg_color=
"#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),command=Ext,hover_color="#b68339")
ExitBtn.place(x=30,y=735)
Extfrm=tkinter.Frame(width=270,background="black",height=70)
customtkinter.CTkButton(master=Extfrm,text="Sign
Out",image=Sign_Out,compound=LEFT,text_color="Black",width=200,height=40,corner_radius=10,fg_co
lor="#a8701d",bg_color="black",anchor=W,border_spacing=0,font=('Century
Gothic',20,"bold"),hover_color="#b68339",command=Sign_Outt).place(x=6,y=8)
def Ext_on_hover(e):
   Extfrm.place(x=320,y=910)
def Ext_on_exit(e):
   # print("Chala Gaya")
   time.sleep(0.1)
   Extfrm.place(x=5000, y=5000)
ExitBtn.bind('<Enter>',Ext_on_hover)
ExitBtn.bind('<Leave>',Ext_on_exit)
Extfrm.bind('<Enter>',Ext_on_hover)
Extfrm.bind('<Leave>',Ext_on_exit)
f2=customtkinter.CTkFrame(master=l1,fg_color="black",width=50,height=450)
```

```
img2 = ImageTk.PhotoImage(Image.open("./assets/output-modified (1).jpeg").resize((1570,955)))
img3 = ImageTk.PhotoImage(Image.open("./assets/ai.png").resize((1570,955)))
img4 = ImageTk.PhotoImage(Image.open("./assets/im1.png").resize((1570,955)))
img5 = ImageTk.PhotoImage(Image.open("./assets/OIG.jpeg").resize((1570,955)))
img6 = ImageTk.PhotoImage(Image.open("./assets/im1.png").resize((1570,955)))
img7 = ImageTk.PhotoImage(Image.open("./assets/im1.png").resize((1570,955)))
img8 = ImageTk.PhotoImage(Image.open("./assets/output_1.jpg").resize((1570,955)))
img9 = ImageTk.PhotoImage(Image.open("./assets/output_2.jpg").resize((1570,955)))
img10 = ImageTk.PhotoImage(Image.open("./assets/output_3.jpg").resize((1570,955)))
img11 = ImageTk.PhotoImage(Image.open("./assets/output_4.jpg").resize((1570,955)))
f11=customtkinter.CTkLabel(master=l1,height=0,width=0,bg_color="black",image=img11,text="")
can_widgett = Canvas(l1,width=1550,height=950,borderwidth=0,bd=0)
can_widgett.place(x=330,y=25)
can_widget = Canvas(l1,width=1550,height=950,borderwidth=0,bd=0)
img3 = ImageTk.PhotoImage(Image.open("./assets/T3.png").resize((1570,955)))
cn = ImageTk.PhotoImage(Image.open("./assets/bb.png").resize((995,690)))
can_widgett.create_image(0,0,anchor=NW,image=img3)
can_widgett.create_image(1052,580,image=cn)# can_widget.create_image(1160,475,image=cn))
Gstryselcton=customtkinter.CTkFrame(can_widgett,fg_color="gold",width=120,height=5,bg_color="#192
d48")
Gstryselcton.place(x=610,y=160)
def ChkCmd():
    ent.set("")
    chk_terms.set(value=0)
    Chk_Ot.set(value=0)
    Rm_Serve.set(value=0)
    Ldry_Serve.set(value=0)
    ShftRm.set(value=0)
    Cng_Date.set(value=0)
    Rm_Reverse.set(value=0)
    Chk_Rm.set(value=0)
    Rsvrent.set("")
    Rsvrent1.set("")
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    for item in table.get_children():
        table.delete(item)
    # query = "select * from `Check In Details` order by `Room No`;"
    query = "select * from `Check In Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(sn, row[0], row[1], row[2], row[3], row[4], row[5]))
        sn += 1
    Gstryselcton.configure(width=120)
    Gstryselcton.place(x=610, y=160)
    Rsvrrst.place(x=121000, y=200)
    RsvrRfrsh.place(x=136500, y=200)
    Rsvrff.place(x=80000, y=350)
    frm3.place(x=4500 + 520, y=400, width=975, height=200)
```

```
    can_widgett.coords(RsvrOpt1, 73000, 300)
    can_widgett.coords(RsvrOpt2, 65000, 370)
    Rsvrentry1.place(x=90000, y=278, width=100, height=40)
    RsvrScrh1.place(x=102000, y=270)
    Rsvrrst.place(x=120000, y=270)
    RsvrRfrsh.place(x=138000, y=270)

    can_widgett.coords(RsvrOpt, 73000, 660)
    Rsvrentry.place(x=90000, y=638, width=100, height=40)
    RsvrScrh.place(x=102000, y=630)
    RsvrRset.place(x=120000, y=630)
    RsvrRfrsh1.place(x=138000, y=630)
    frm4.place(x=4500 + 520, y=700, width=975, height=200)

    ChkCmdScrh.place(x=480 + 450, y=315)
    ChkCmdRset.place(x=450 + 660, y=315)
    ChkCmdRfrsh.place(x=600 + 700, y=315)
    ChkCmdentry.place(x=80 + 500, y=320, width=330, height=40)
    can_widgett.coords(ChkCmdScrhGstNm, 740, 280)
    can_widgett.coords(ChkCmdOpt, 650, 400)
    ChkCmdff.place(x=45 + 520, y=450)
    frm2.place(x=45 + 520, y=500, width=975, height=400)

    pygame.mixer.music.load("./Voices/Services.mp3")
    pygame.mixer.music.play()


ChkBttn=customtkinter.CTkButton(can_widgett,bg_color="#192d48",command=ChkCmd,hover_color="#
233b58",cursor="hand2",fg_color="#192d48",text_color="white",text="Services",font=('Pristina',40,"bol
d"))
ChkBttn.place(x=600,y=100)
def RsvCmd():
    ent.set("")
    chk_terms.set(value=0)
    Chk_Ot.set(value=0)
    Rm_Serve.set(value=0)
    Ldry_Serve.set(value=0)
    ShftRm.set(value=0)
    Cng_Date.set(value=0)
    Rm_Reverse.set(value=0)
    Chk_Rm.set(value=0)
    Rsvrent.set("")
    Rsvrent1.set("")
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    for tem in Aval.get_children():
        Aval.delete(tem)
    S_No = 1
    query = "select * from `Room Status` where `Status` not in ('Dirty','Repair') order by `Room No.` asc;"
    cur.execute(query)
```

```
    for i in cur.fetchall():
        Aval.insert("", END, values=(S_No, i[0], i[1], i[3]))
        S_No += 1
    for tm in ChkAval.get_children():
        ChkAval.delete(tm)
    S_No = 1
    query = "select * from `Room Status` where `Status`!= 'vacant' order by `Room No.` asc ;"
    cur.execute(query)
    for i in cur.fetchall():
        ChkAval.insert("", END, values=(S_No, i[0], i[1], i[2]))
        S_No += 1
    Gstryselcton.configure(width=250)
    Gstryselcton.place(x=860, y=160)
    ChkCmdScrh.place(x=48000 + 450, y=315)
    ChkCmdRset.place(x=45000 + 660, y=315)
    ChkCmdRfrsh.place(x=60000 + 700, y=315)
    ChkCmdentry.place(x=8000 + 500, y=320, width=330, height=40)
    can_widgett.coords(ChkCmdScrhGstNm, 74000, 280)
    can_widgett.coords(ChkCmdOpt, 65000, 400)
    ChkCmdff.place(x=4500 + 520, y=450)
    frm2.place(x=4500 + 520, y=500, width=975, height=400)

    Rsvrrst.place(x=1210, y=200)
    RsvrRfrsh.place(x=1365, y=200)
    Rsvrff.place(x=800, y=350)
    frm3.place(x=45 + 520, y=400, width=975, height=200)
    can_widgett.coords(RsvrOpt1, 730, 300)
    can_widgett.coords(RsvrOpt2, 650, 370)
    Rsvrentry1.place(x=900, y=278, width=100, height=40)
    RsvrScrh1.place(x=1020, y=270)
    Rsvrrst.place(x=1200, y=270)
    RsvrRfrsh.place(x=1380, y=270)

    can_widgett.coords(RsvrOpt, 730, 660)
    Rsvrentry.place(x=900, y=638, width=100, height=40)
    RsvrScrh.place(x=1020, y=630)
    RsvrRset.place(x=1200, y=630)
    RsvrRfrsh1.place(x=1380, y=630)
    frm4.place(x=45 + 520, y=700, width=975, height=200)

    pygame.mixer.music.load("./Voices/Room Availibility.mp3")
    pygame.mixer.music.play()


RsvrBttn=customtkinter.CTkButton(can_widgett,bg_color="#192d48",command=RsvCmd,hover_color="#
233b58",cursor="hand2",fg_color="#192d48",text_color="white",text="Room
Availability",font=('Pristina',40,"bold"))
RsvrBttn.place(x=850,y=100)
ent=tkinter.StringVar()
ChkCmdentry=Entry(can_widgett,highlightthickness=2,textvariable=ent,highlightbackground="grey",highl
ightcolor="black",fg="blue",font="consolas 17 italic")

def Scrh ():
    con = connector.connect(host='localhost',
```

```
                    port='3306',
                    user='root',
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    for item in table.get_children():
        table.delete(item)
    # query = "select * from `Check In Details` order by `Room No`;"
    query = f"select * from `Check In Details` where `Guest Id`='{ent.get()}';"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(sn, row[0], row[1], row[2], row[3], row[4], row[5]))
        sn += 1
    chk_terms.set(value=0)
    Chk_Ot.set(value=0)
    Rm_Serve.set(value=0)
    Ldry_Serve.set(value=0)

ChkCmdScrh=tkinter.Button(can_widgett,image=Guest_Entry,command=Scrh,compound=LEFT,fg="Black
",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Search",bg="#a870
1d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
def entt ():
    ent.set("")
    chk_terms.set(value=0)
    Chk_Ot.set(value=0)
    Rm_Serve.set(value=0)
    Ldry_Serve.set(value=0)
    ShftRm.set(value=0)
    Cng_Date.set(value=0)

ChkCmdRset=tkinter.Button(can_widgett,image=Guest_Entry,command=entt,compound=LEFT,fg="Black
",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Reset",bg="#a8701
d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
def refresh():
    con = connector.connect(host='localhost',
                    port='3306',
                    user='root',
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    for item in table.get_children():
        table.delete(item)
    # query = "select * from `Check In Details` order by `Room No`;"
    query = "select * from `Check In Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(sn, row[0], row[1], row[2], row[3], row[4], row[5]))
        sn += 1
ChkCmdRfrsh=tkinter.Button(can_widgett,image=Guest_Entry,command=refresh,compound=LEFT,fg="B
lack",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Refresh",bg="#
a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
ChkCmdScrhGstNm=can_widgett.create_text(4000,2200,text="Search By Guest
```

```
ID",font=('Pristina',30,"bold"),fill="white")
ChkCmdOpt=can_widgett.create_text(4000,2200,text="Operations",font=('Pristina',30,"bold"),fill="white
")
chk_terms=tkinter.IntVar(value=0)
Chk_Ot=tkinter.IntVar(value=0)
Rm_Serve=tkinter.IntVar(value=0)
Ldry_Serve=tkinter.IntVar(value=0)
ShftRm=tkinter.IntVar(value=0)
Cng_Date=tkinter.IntVar(value=0)
ChkCmdff=tkinter.Frame(can_widgett,background="black",height=31,width=975)
chkbx=customtkinter.CTkCheckBox(ChkCmdff,font=('Century Gothic',16),text="Room
Service",variable=Rm_Serve,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_radius
=25,fg_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
chkbx.place(x=0,y=0)
chkbx=customtkinter.CTkCheckBox(ChkCmdff,font=('Century Gothic',16),text="Laundry
Service",variable=Ldry_Serve,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_radiu
s=25,fg_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
chkbx.place(x=200,y=0)
Shft_rm=customtkinter.CTkCheckBox(ChkCmdff,font=('Century Gothic',16),text="Shift
Room",variable=ShftRm,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_radius=25,
fg_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
Shft_rm.place(x=400,y=0)
CngDate=customtkinter.CTkCheckBox(ChkCmdff,font=('Century Gothic',16),text="Change Check Out
Date",variable=Cng_Date,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_radius=2
5,fg_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
CngDate.place(x=530,y=0)
frm2=Frame(can_widgett,relief=SUNKEN,borderwidth=4)
scbr_x=Scrollbar(frm2,orient=HORIZONTAL)
scbr_y=Scrollbar(frm2,orient=VERTICAL)

s=ttk.Style()
s.theme_use("winnative") # classic , alt,default , winnative , xpnative , clam , vista

            #     FOR INSERT VALUES

s.configure(".",font=("consolas",14,"italic"),foreground="blue")

            #     TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview",foreground="black",background="light
yellow",rowheight=25,fieldbackground="light yellow")
s.map("Treeview",background=[("selected","blue")])
s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light grey")



            #     TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table=ttk.Treeview(frm2,cursor="hand2",columns=("SN_No.","Gs_ID","Gst_Name","Room_No","Roo
m_Type","D_In","D_Out"),selectmode="browse",xscrollcommand=scbr_x.set,yscrollcommand=scbr_y.set)
scbr_x.pack(side=BOTTOM,fill=X)
scbr_y.pack(side=RIGHT,fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
```

```
table.heading("SN_No.",text="Sn No.",anchor=CENTER)
table.heading("Room_No",text="Room No.",anchor=CENTER)
table.heading("Room_Type",text="Room Type",anchor=CENTER)
table.heading("Gs_ID",text="Guest ID",anchor=CENTER)
table.heading("Gst_Name",text="Guest Name",anchor=CENTER)
table.heading("D_In",text="Date In",anchor=CENTER)
table.heading("D_Out",text="Date Out",anchor=CENTER)
table.pack(fill=BOTH,expand=1)

table["show"]="headings"
table.column("SN_No.",width=100,anchor=CENTER,minwidth=50)
table.column("Room_No",width=130,anchor=CENTER,minwidth=130)
table.column("Room_Type",width=130,anchor=CENTER,minwidth=130)
table.column("Gs_ID",width=130,anchor=CENTER,minwidth=130)
table.column("Gst_Name",width=200,anchor=CENTER,minwidth=200)
table.column("D_In",width=160,anchor=CENTER,minwidth=160)
table.column("D_Out",width=175,anchor=CENTER,minwidth=175)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
query="select * from `Check In Details`;"
cur.execute(query)
sn=1
for row in cur.fetchall():
    table.insert("",END, values=(sn, row[0], row[1], row[2],row[3], row[4], row[5]))
    sn+=1
def item_select(_):
    Gs = pd.read_csv("BillNo.csv", index_col=[0])
    c = table.item(table.selection())['values']
    # print(Gs)
    # print(c)
    Gs.BillNo[2] =c[1]
    Gs.BillNo[3] =c[2]
    Gs.BillNo[4] =c[3]
    Gs.BillNo[5] =c[4]
    Gs.BillNo[6] =c[5]
    Gs.BillNo[7] =c[6]
    Gs.to_csv("BillNo.csv")
    if Rm_Serve.get()==1:
        if messagebox.askyesno("Room Service","Are You Sure You Want Room Service"):
            os.system("python Rm_Service.py")
    if Ldry_Serve.get()==1:
        if messagebox.askyesno("Laundry Service", "Are You Sure You Want Laundry Service"):
            os.system("python Lndry_Service.py")
    if ShftRm.get() == 1:
        if messagebox.askyesno("Shift Room", "Are You Sure You Want To Shift Room"):
            os.system("python Shft_Rm.py")
    if Cng_Date.get() == 1:
        if messagebox.askyesno("Change Date", "Are You Sure You Want To Change Check Out Date"):
            os.system("python CNG_DT.py")
table.bind('<<TreeviewSelect>>',item_select)
```

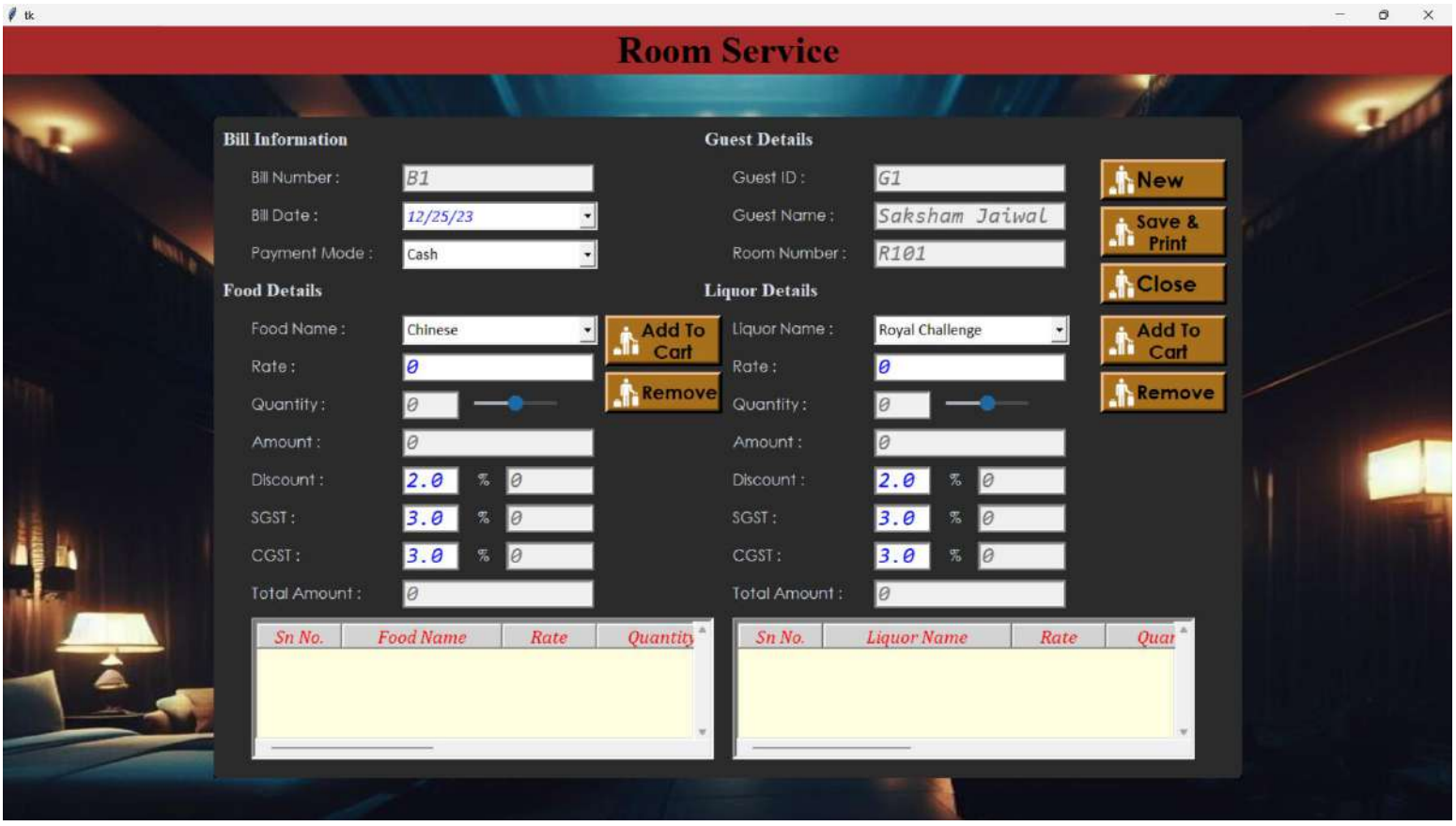
```
dt_in=tkinter.StringVar()
dt_ot=tkinter.StringVar()
ChkCmdScrh.place(x=480+450,y=315)
ChkCmdRset.place(x=450+660, y=315)
ChkCmdRfrsh.place(x=600+700, y=315)
ChkCmdentry.place(x=80+500, y=320, width=330, height=40)
can_widgett.coords(ChkCmdScrhGstNm, 740, 280)
can_widgett.coords(ChkCmdOpt, 650, 400)
ChkCmdff.place(x=45+520, y=450)
frm2.place(x=45+520, y=500, width=975, height=400)
def reset():
   Rm_Reverse.set(value=0)
   Chk_Rm.set(value=0)
   Rsvrent1.set("")
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   for tem in Aval.get_children():
      Aval.delete(tem)
   S_No = 1
   query = "select * from `Room Status` where `Status` not in ('Dirty','Repair') order by `Room No.` asc;"
   cur.execute(query)
   for i in cur.fetchall():
      Aval.insert("", END, values=(S_No, i[0], i[1], i[3]))
      S_No += 1
Rsvrrst=tkinter.Button(can_widgett,image=Guest_Entry,command=reset,compound=LEFT,fg="Black",wid
th=150,activeforeground="black",activebackground="#a8701d",height=40,text="Reset
",bg="#a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
def Aval_Refresh_ChkAval():
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   for tem in Aval.get_children():
      Aval.delete(tem)
   S_No = 1
   query = "select * from `Room Status` where `Status` not in ('Dirty','Repair') order by `Room No.` asc;"
   cur.execute(query)
   for i in cur.fetchall():
      Aval.insert("", END, values=(S_No, i[0], i[1], i[3]))
      S_No += 1
RsvrRfrsh=tkinter.Button(can_widgett,image=Guest_Entry,command=Aval_Refresh_ChkAval,compound=
LEFT,fg="Black",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Re
fresh",bg="#a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")

RsvrOpt1=can_widgett.create_text(4000,2200,text="Search By Room
No.",font=('Pristina',30,"bold"),fill="white")
RsvrOpt2=can_widgett.create_text(4000,2200,text="Operations",font=('Pristina',30,"bold"),fill="white")
Rsvrent1=StringVar()
```

```
Rsvrentry1=Entry(can_widgett,highlightthickness=2,textvariable=Rsvrent1,highlightbackground="grey",hig
hlightcolor="black",fg="blue",font="consolas 17 italic")
def RsvrCmdScrh1 ():
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   for tem in Aval.get_children():
      Aval.delete(tem)
   S_No = 1
   query = f"select * from `Room Status` where `Room No.`='{Rsvrentry1.get()}';"
   cur.execute(query)
   for i in cur.fetchall():
      Aval.insert("", END, values=(S_No, i[0], i[1], i[3]))
      S_No += 1
RsvrScrh1=tkinter.Button(can_widgett,image=Guest_Entry,command=RsvrCmdScrh1,compound=LEFT,fg
="Black",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Search",bg
="#a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")

RsvrOpt=can_widgett.create_text(4000,2200,text="Search By Room
No.",font=('Pristina',30,"bold"),fill="white")
Rsvrff=tkinter.Frame(can_widgett,background="black",borderwidth=0,width=740,height=35)
Rsvrent=StringVar()
Rsvrentry=Entry(can_widgett,highlightthickness=2,textvariable=Rsvrent,highlightbackground="grey",highl
ightcolor="black",fg="blue",font="consolas 17 italic")
def RsvrCmdScrh ():
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   for tm in ChkAval.get_children():
      ChkAval.delete(tm)
   S_No = 1
   query = f"select * from `Room Status` where `Room No.`='{Rsvrentry.get()}';"
   cur.execute(query)
   for i in cur.fetchall():
      ChkAval.insert("", END, values=(S_No, i[0], i[1], i[2]))
      S_No += 1
RsvrScrh=tkinter.Button(can_widgett,image=Guest_Entry,command=RsvrCmdScrh,compound=LEFT,fg="
Black",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Search",bg="
#a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
def RsvrCmdRset ():
   Rsvrent.set("")
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   for tm in ChkAval.get_children():
```

```
    ChkAval.delete(tm)
  S_No = 1
  query = "select * from `Room Status` where `Status`!= 'vacant' order by `Room No.` asc ;"
  cur.execute(query)
  for i in cur.fetchall():
    ChkAval.insert("", END, values=(S_No, i[0], i[1], i[2]))
    S_No += 1
RsvrRset=tkinter.Button(can_widgett,image=Guest_Entry,command=RsvrCmdRset,compound=LEFT,fg="
Black",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Reset",bg="#
a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
def RsvrCmdRfrsh():
  con = connector.connect(host='localhost',
                port='3306',
                user='root',
                password='Password',
                database='Hotel Management Software')
  cur = con.cursor()
  for tm in ChkAval.get_children():
    ChkAval.delete(tm)
  S_No = 1
  query = "select * from `Room Status` where `Status`!= 'vacant' order by `Room No.` asc ;"
  cur.execute(query)
  for i in cur.fetchall():
    ChkAval.insert("", END, values=(S_No, i[0], i[1], i[2]))
    S_No += 1
RsvrRfrsh1=tkinter.Button(can_widgett,image=Guest_Entry,command=RsvrCmdRfrsh,compound=LEFT,f
g="Black",width=150,activeforeground="black",activebackground="#a8701d",height=40,text="Refresh",b
g="#a8701d",anchor=W,font=('Century Gothic',20,"bold"),borderwidth=5,cursor="hand2")
#-------------------------------------------------------------------------------------------------------
Rm_Reverse=tkinter.IntVar(value=0)
Chk_Rm=tkinter.IntVar(value=0)
chkbx=customtkinter.CTkCheckBox(Rsvrff,font=('Century Gothic',16),text="Room
Reservation",variable=Rm_Reverse,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_
radius=25,fg_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
chkbx.place(x=0,y=0)
Chout=customtkinter.CTkCheckBox(Rsvrff,font=('Century Gothic',16),text="Check
In",variable=Chk_Rm,onvalue=1,offvalue=0,checkbox_width=25,checkbox_height=25,corner_radius=25,fg
_color="blue",bg_color="black",border_color="white",hover=False,border_width=2)
Chout.place(x=300,y=0)
#-------------------------------------------------------------------------------------------------------
frm3=Frame(can_widgett,relief=SUNKEN,borderwidth=4)

scbrr_x=Scrollbar(frm3,orient=HORIZONTAL)
scbrr_y=Scrollbar(frm3,orient=VERTICAL)
Aval=ttk.Treeview(frm3,cursor="hand2",columns=("SN_No.","Room_No","Room_Type","Room_Charge
s"),selectmode="browse",xscrollcommand=scbrr_x.set,yscrollcommand=scbrr_y.set)
scbrr_x.pack(side=BOTTOM,fill=X)
scbrr_y.pack(side=RIGHT,fill=Y)
scbrr_x.config(command=Aval.xview)
scbrr_y.config(command=Aval.yview)
Aval.heading("SN_No.",text="Sn No.",anchor=CENTER)
Aval.heading("Room_No",text="Room No.",anchor=CENTER)
Aval.heading("Room_Type",text="Room Type",anchor=CENTER)
Aval.heading("Room_Charges",text="Room Charges",anchor=CENTER)
```

```
Aval.pack(fill=BOTH,expand=1)

Aval["show"]="headings"
Aval.column("SN_No.",width=100,anchor=CENTER,minwidth=30)
Aval.column("Room_No",width=130,anchor=CENTER,minwidth=110)
Aval.column("Room_Type",width=200,anchor=CENTER,minwidth=180)
Aval.column("Room_Charges",width=160,anchor=CENTER,minwidth=130)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
S_No=1
query="select * from `Room Status` where `Status` not in ('Dirty','Repair');"
cur.execute(query)
for i in cur.fetchall():
   Aval.insert("",END,values=(S_No,i[0],i[1],i[3]))
   S_No+=1
def Ava_select(_):
   if Chk_Rm.get()==1:
      if messagebox.askyesno("Check In", "Are You Sure You Want To Check In"):
         so = pd.Series(data=Aval.item(Aval.selection())['values'], name="RoomDetails")
         sep = pd.DataFrame(so)
         sep.to_csv("RMNO_RMPRICE.csv")
         os.system("python Check_In.py")
   if Rm_Reverse.get()==1:
      if messagebox.askyesno("Room Reservation", "Are You Sure You Want To Reserve Room"):
         Rmso = pd.Series(data=Aval.item(Aval.selection())['values'], name="RoomDetails")
         Rmsep = pd.DataFrame(Rmso)
         Rmsep.to_csv("RM_Reservation.csv")
         os.system("python Room_Reservation.py")
Aval.bind('<<TreeviewSelect>>',Ava_select)
#-----------------------------------------------------------------------------------------------------------
frm4=Frame(can_widgett,relief=SUNKEN,borderwidth=4)

scbrrr_x=Scrollbar(frm4,orient=HORIZONTAL)
scbrrr_y=Scrollbar(frm4,orient=VERTICAL)
ChkAval=ttk.Treeview(frm4,cursor="hand2",columns=("SN_No.","Room_No","Room_Type","Status"),s
electmode="browse",xscrollcommand=scbrrr_x.set,yscrollcommand=scbrrr_y.set)
scbrrr_x.pack(side=BOTTOM,fill=X)
scbrrr_y.pack(side=RIGHT,fill=Y)
scbrrr_x.config(command=ChkAval.xview)
scbrrr_y.config(command=ChkAval.yview)
ChkAval.heading("SN_No.",text="Sn No.",anchor=CENTER)
ChkAval.heading("Room_No",text="Room No.",anchor=CENTER)
ChkAval.heading("Room_Type",text="Room Type",anchor=CENTER)
ChkAval.heading("Status",text="Status",anchor=CENTER)
ChkAval.pack(fill=BOTH,expand=1)

ChkAval["show"]="headings"
ChkAval.column("SN_No.",width=100,anchor=CENTER,minwidth=30)
ChkAval.column("Room_No",width=130,anchor=CENTER,minwidth=110)
ChkAval.column("Room_Type",width=200,anchor=CENTER,minwidth=180)
```

```
ChkAval.column("Status",width=160,anchor=CENTER,minwidth=130)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
S_No=1
query="select * from `Room Status` where `Status`!= 'vacant';"
cur.execute(query)
for i in cur.fetchall():
    ChkAval.insert("",END,values=(S_No,i[0],i[1],i[2]))
    S_No+=1

def ChkAval_select(_):
    if messagebox.askyesno("Room Status", "Are You Sure You Want To Change Room Status"):
        RmStatus = pd.Series(data=ChkAval.item(ChkAval.selection())['values'], name="RoomStatus")
        RmStatusDf = pd.DataFrame(RmStatus)
        RmStatusDf.to_csv("RM_Status.csv")
        os.system("python Room_Type.py")
ChkAval.bind('<<TreeviewSelect>>',ChkAval_select)
```



Invoice

Billed to:

{{ name }}
{{ phone }}
{{Address}}

Invoice No.
{{Bill_No}}
{{Date}}

Item	Quantity	Unit Price	Total
	{{%tr for item in invoice_list %}}		
{{item[1]}}	{{item[3]}}	${{item[2]}}	${{item[4]}}
	{{%tr endfor %}}		

Subtotal	${{ subtotal }}
S.GST ({{S_gst}}%)	${{ sgsttax }}
C.GST ({{C_gst}}%)	${{ cgsttax }}
Discount ({{Dst}}%)	${{ dsttax }}
Total	**${{ total }}**

Thank you

Payment Information

Cnd Bank
Account Name: Aaksham
Account No: 123-456-7890

Aaksham Hotel

Navi, Mumbai,In Front Of Ambani House

```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import pandas as pd
from tkcalendar import DateEntry
import mysql.connector as connector
from datetime import date
import os

import pygame
height = 780
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-40))
Data = pd.read_csv("BillNo.csv", index_col=[0])
Label(text="Room Service",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
Data = pd.read_csv("BillNo.csv", index_col=[0])
query = f"select chkindtl.`Guest ID`,chkindtl.`Guest Name`,cstmdtl.`Contact No.`,CONCAT
(cstmdtl.Address, ', ', cstmdtl.City, ', ', cstmdtl.Country) as Address,cstmdtl.`Email ID`from `check in
details`chkindtl  right join `customer details` cstmdtl using (`Guest ID`) where chkindtl.`Room
No`='{Data.BillNo[4]}';"
cur.execute(query)
DataLst=cur.fetchone()
print(DataLst)
img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)
frame = customtkinter.CTkFrame(master=l1,width=1090,height=700,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))
customtkinter.CTkLabel(master=frame, text="Bill Information", font=('Times New Roman', 20,
"bold")).place(x=10, y=10)
customtkinter.CTkLabel(master=frame, text="Guest Details", font=('Times New Roman', 20,
"bold")).place(x=500+20, y=10)
customtkinter.CTkLabel(master=frame, text="Food Details", font=('Times New Roman', 20,
"bold")).place(x=10, y=170)
customtkinter.CTkLabel(master=frame, text="Liquor Details", font=('Times New Roman', 20,
"bold")).place(x=500+20, y=170)
#-------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Guest ID :", font=('Century Gothic', 16)).place(x=550, y=50)
customtkinter.CTkLabel(master=frame, text="Guest Name :", font=('Century Gothic', 16)).place(x=550,
```

```
y=90)
customtkinter.CTkLabel(master=frame, text="Room Number :", font=('Century Gothic', 16)).place(x=550,
y=130)
#-------------------------------------------------------------------------------------------------------------
Gs_ID=StringVar()
Gs_ID.set(Data.BillNo[2])
Entry(frame,highlightthickness=2,textvariable=Gs_ID,highlightbackground="grey",highlightcolor="black",
fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=50,width=203,height=30)
Gs_Name=StringVar()
Gs_Name.set(Data.BillNo[3])
Entry(frame,highlightthickness=2,textvariable=Gs_Name,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=90,width=203,height=30)
Gs_RMNO=StringVar()
Gs_RMNO.set(Data.BillNo[4])
Entry(frame,highlightthickness=2,textvariable=Gs_RMNO,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=130,width=203,height=30)
#-------------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Bill Number :", font=('Century Gothic', 16)).place(x=40,
y=50)
customtkinter.CTkLabel(master=frame, text="Bill Date :", font=('Century Gothic', 16)).place(x=40, y=90)
customtkinter.CTkLabel(master=frame, text="Payment Mode :", font=('Century Gothic', 16)).place(x=40,
y=130)
customtkinter.CTkLabel(master=frame, text="Food Name :", font=('Century Gothic', 16)).place(x=40,
y=210)
customtkinter.CTkLabel(master=frame, text="Rate :", font=('Century Gothic', 16)).place(x=40, y=250)
customtkinter.CTkLabel(master=frame, text="Quantity :", font=('Century Gothic', 16)).place(x=40, y=290)
customtkinter.CTkLabel(master=frame, text="Amount :", font=('Century Gothic', 16)).place(x=40, y=330)
customtkinter.CTkLabel(master=frame, text="Discount :", font=('Century Gothic', 16)).place(x=40, y=370)
customtkinter.CTkLabel(master=frame, text="SGST :", font=('Century Gothic', 16)).place(x=40, y=410)
customtkinter.CTkLabel(master=frame, text="CGST :", font=('Century Gothic', 16)).place(x=40, y=450)
customtkinter.CTkLabel(master=frame, text="Total Amount :", font=('Century Gothic', 16)).place(x=40,
y=490)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=410-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=450-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=490-40)
#-------------------------------------------------------------------------------------------------------------
Bill=StringVar()
Bill.set(str(Data.BillNo[0])+str(Data.BillNo[1]))
Entry(frame,highlightthickness=2,textvariable=Bill,highlightbackground="grey",highlightcolor="black",fg
="blue",font="consolas 17 italic",state="disabled").place(x=200,y=50,width=203,height=30)
cal = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue", width=20)
cal.place(x=200, y=90)
#-------------------------------------------------------------------------------------------------------------
mydata = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13",
width=20,background="grey", height=10)
mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
l = []
mydata.set("Cash")
mydata.place(x=200, y=130)
mydata1 = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13",
width=20,background="grey", height=10)
mydata1["value"]=["Bhel Puri","Aalo Tikki","Burger","Pizza","Italian","Chinese"]
mydata1.set("Chinese")
mydata1.place(x=200, y=210)
```

```
#-------------------------------------------------------------------------------------------------------------
Foodrat=IntVar()
Entry(frame,highlightthickness=2,textvariable=Foodrat,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic").place(x=200,y=290-40,width=203,height=30)
Foodqnt=IntVar()
Entry(frame,highlightthickness=2,textvariable=Foodqnt,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=330-40,width=60,height=30)
Foodamt=IntVar()
Entry(frame,highlightthickness=2,highlightbackground="grey",textvariable=Foodamt,highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=370-40,width=203,height=30)
Fooddst=IntVar()
Fooddst.set(2.0)
Entry(frame,highlightthickness=2,textvariable=Fooddst,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic").place(x=200,y=410-40,width=60,height=30)
Foods_GST=IntVar()
Foods_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Foods_GST,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic").place(x=200,y=450-40,width=60,height=30)
Foodc_GST=IntVar()
Foodc_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Foodc_GST,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic").place(x=200,y=490-40,width=60,height=30)
FoodTl_Amnt=IntVar()
Entry(frame,highlightthickness=2,textvariable=FoodTl_Amnt,highlightbackground="grey",highlightcolor=
"black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=530-40,width=203,height=30)
Fooddstprice=IntVar()
Entry(frame,highlightthickness=2,textvariable=Fooddstprice,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=410-40,width=93,height=30)
Foodss_GST=IntVar()
Entry(frame,highlightthickness=2,textvariable=Foodss_GST,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=450-40,width=93,height=30)
Foodcc_GST=IntVar()
Entry(frame,highlightthickness=2,textvariable=Foodcc_GST,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=490-40,width=93,height=30)
#-------------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Liquor Name :", font=('Century Gothic',
16)).place(x=530+20, y=210)
customtkinter.CTkLabel(master=frame, text="Rate :", font=('Century Gothic', 16)).place(x=530+20,
y=290-40)
customtkinter.CTkLabel(master=frame, text="Quantity :", font=('Century Gothic', 16)).place(x=530+20,
y=330-40)
customtkinter.CTkLabel(master=frame, text="Amount :", font=('Century Gothic', 16)).place(x=530+20,
y=370-40)
customtkinter.CTkLabel(master=frame, text="Discount :", font=('Century Gothic', 16)).place(x=530+20,
y=410-40)
customtkinter.CTkLabel(master=frame, text="SGST :", font=('Century Gothic', 16)).place(x=530+20,
y=450-40)
customtkinter.CTkLabel(master=frame, text="CGST :", font=('Century Gothic', 16)).place(x=530+20,
y=490-40)
customtkinter.CTkLabel(master=frame, text="Total Amount :", font=('Century Gothic',
16)).place(x=530+20, y=530-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=780, y=410-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=780, y=450-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=780, y=490-40)
```

```
#-----------------------------------------------------------------------------------------------------------
rat=IntVar()
Entry(frame,highlightthickness=2,textvariable=rat,highlightbackground="grey",highlightcolor="black",fg=
"blue",font="consolas 17 italic").place(x=700,y=290-40,width=203,height=30)
qnt=IntVar()
Entry(frame,highlightthickness=2,textvariable=qnt,highlightbackground="grey",highlightcolor="black",fg=
"blue",font="consolas 17 italic",state="disabled").place(x=700,y=330-40,width=60,height=30)
amt=IntVar()
Entry(frame,highlightthickness=2,highlightbackground="grey",textvariable=amt,highlightcolor="black",fg
="blue",font="consolas 17 italic",state="disabled").place(x=700,y=370-40,width=203,height=30)
dst=IntVar()
dst.set(2.0)
Entry(frame,highlightthickness=2,textvariable=Fooddst,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic").place(x=700,y=410-40,width=60,height=30)
s_GST=IntVar()
s_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Foods_GST,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic").place(x=700,y=450-40,width=60,height=30)
c_GST=IntVar()
c_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Foodc_GST,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic").place(x=700,y=490-40,width=60,height=30)
Tl_Amnt=IntVar()
Entry(frame,highlightthickness=2,textvariable=Tl_Amnt,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=530-40,width=203,height=30)
dstprice=IntVar()
Entry(frame,highlightthickness=2,textvariable=dstprice,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic",state="disabled").place(x=810,y=410-40,width=93,height=30)
ss_GST=IntVar()
Entry(frame,highlightthickness=2,textvariable=ss_GST,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic",state="disabled").place(x=810,y=450-40,width=93,height=30)
cc_GST=IntVar()
Entry(frame,highlightthickness=2,textvariable=cc_GST,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic",state="disabled").place(x=810,y=490-40,width=93,height=30)
#-----------------------------------------------------------------------------------------------------------
mydata2 = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13",
width=20,background="grey", height=10)
mydata2["value"]=["Jinro Soju","McDowell's No.1 Whisky","Royal Stag Whisky","Royal
Challenge","Smirnoff","Tuborg Brewery","Johnnie Walker"]
mydata2.set("Royal Challenge")
mydata2.place(x=700, y=210)
#-----------------------------------------------------------------------------------------------------------
def Liquorsliderevent(value):
    qnt.set(value=int(value))
    try:
        amt.set(value=int(value * rat.get()))
        dstprice.set(value=int((value*rat.get()*dst.get())/100))
        ss_GST.set(value=int((value*rat.get()*s_GST.get())/100))
        cc_GST.set(value=int((value*rat.get()*c_GST.get())/100))
        Tl_Amnt.set(value=str(rat.get()+dstprice.get()+ss_GST.get()+cc_GST.get()))
    except EXCEPTION as e:
        print(e)
    # print(value=int(value))
def Foodsliderevent(value):
```

```
    Foodqnt.set(value=int(value))
    try:
        Foodamt.set(value=int(value * Foodrat.get()))
        Fooddstprice.set(value=int((value * Foodrat.get() * Fooddst.get()) / 100))
        Foodss_GST.set(value=int((value * Foodrat.get() * Foods_GST.get()) / 100))
        Foodcc_GST.set(value=int((value * Foodrat.get() * Foodc_GST.get()) / 100))
        FoodTl_Amnt.set(value=str(Foodrat.get() + Fooddstprice.get() + Foodss_GST.get() +
Foodcc_GST.get()))
    except EXCEPTION as e:
        print(e)
    # print(value=int(value))
customtkinter.CTkSlider(frame,from_=0,to=10,command=Liquorsliderevent,number_of_steps=10,width=10
0).place(x=770,y=335-40)
customtkinter.CTkSlider(frame,from_=0,to=10,command=Foodsliderevent,number_of_steps=10,width=100)
.place(x=270,y=335-40)
#-------------------------------------------------------------------------------------------------------------
def New():
    for item in table.get_children():
        table.delete(item)
    for item in liqr.get_children():
        liqr.delete(item)
    cal.set_date(date.today())
    mydata2.set(value="Royal Challenge")
    mydata1.set(value="Bhel Puri")
    Foodrat.set(value=0)
    Foodqnt.set(value=0)
    Foodamt.set(value=0)
    Fooddstprice.set(value=0)
    Foodss_GST.set(value=0)
    Foodcc_GST.set(value=0)
    FoodTl_Amnt.set(value=0)
    rat.set(value=0)
    qnt.set(value=0)
    amt.set(value=0)
    dstprice.set(value=0)
    ss_GST.set(value=0)
    cc_GST.set(value=0)
    Tl_Amnt.set(value=0)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="New",command=New,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=940, y=40+20-15)
def Save_Print():
    if messagebox.askyesno("Laundry Bill","Are You Sure You Want To Bill Out"):
        LstData = []
        Bdt = cal.get_date().strftime("%Y-%m-%d")
        query = f"INSERT INTO `Room Service Details` VALUES ('{Gs_ID.get()}', '{Gs_Name.get()}',
'{Gs_RMNO.get()}', '{Bill.get()}', '{Bdt}','{mydata.get()}');"
        cur.execute(query)
        print(query)
        con.commit()
        for i in table.get_children():
            l = table.item(i)["values"]
            LstData.append(l)
```

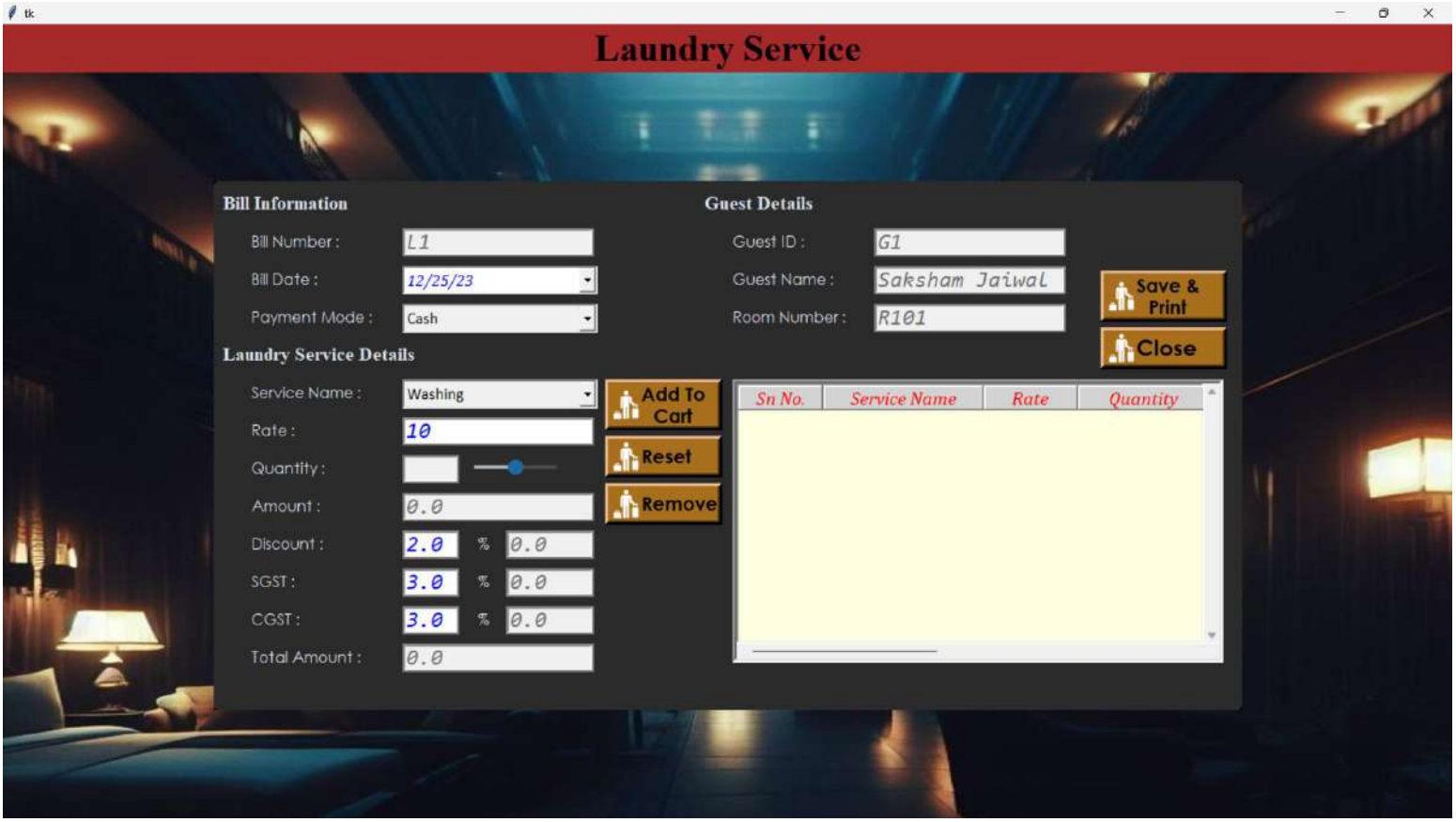
```
        print(l)
        query = f"INSERT INTO `Food Details` VALUES ('{Bill.get()}', '{l[1]}', '{l[2]}', '{l[3]}',
'{l[4]}','{l[5]}', '{l[6]}', '{l[7]}', '{l[8]}','{date.today()}');"
        cur.execute(query)
        print(query)
        con.commit()
    for j in liqr.get_children():
        w = liqr.item(j)["values"]
        LstData.append(w)
        print(w)
        query = f"INSERT INTO `Liquor Details` VALUES ('{Bill.get()}', '{w[1]}', '{w[2]}', '{w[3]}',
'{w[4]}','{w[5]}','{w[6]}', '{w[7]}', '{w[8]}','{date.today()}');"
        cur.execute(query)
        con.commit()
        print(query)
    Data = pd.read_csv("BillNo.csv", index_col=[0])
    Data.BillNo[1] = int(Data.BillNo[1]) + 1
    print(Data.BillNo[1])
    Data.to_csv("BillNo.csv")
    print("ho gaya")
    print(LstData)
    from docx2pdf import convert
    from docxtpl import DocxTemplate
    import smtplib
    from email import encoders
    from email.mime.base import MIMEBase
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    doc = DocxTemplate("Food_Invoice.docx")
    # invoice_list = []
    # tl = 0
    # for i in table.get_children():
    #     invoice_list.append(table.item(i)["values"])
    #     tl += float(table.item(i)["values"][8])
    # print(tl)
    # print(invoice_list)
    TtlAmnt=0
    for i in LstData:
        print(i)
        TtlAmnt+=float(i[4])
    doc.render({"Bill_No": f"{Data.BillNo[0] + str(Data.BillNo[1])}",
                "Date": f"{date.today().strftime('%d/%m/%Y')}",
                "name": f"{Gs_Name.get()}",
                "phone": DataLst[2],
                "Address": DataLst[3],
                "subtotal": f'{TtlAmnt}',
                "invoice_list": LstData,
                "S_gst": dst.get(),
                "C_gst": s_GST.get(),
                "D_gst": c_GST.get(),
                "sgsttax": TtlAmnt+TtlAmnt*s_GST.get(),
                "cgsttax": TtlAmnt+TtlAmnt*c_GST.get(),
                "dsttax": TtlAmnt+TtlAmnt*dst.get(),
                "total": TtlAmnt-
```

```
TtlAmnt+TtlAmnt*dst.get()+TtlAmnt+TtlAmnt*s_GST.get()+TtlAmnt+TtlAmnt*c_GST.get()})
    doc.save("Food invoice.docx")
    convert(r"D:\python\Project\Food invoice.docx", r"D:\python\Project\Room_Service.pdf")
    os.remove(r"D:\python\Project\Food invoice.docx")
    os.system("Room_Service.pdf")

    def Email():
      try:
        connect = smtplib.SMTP('smtp.gmail.com', 587)
        connect.ehlo()
        connect.starttls()
        sender_email = “aakshamhotel@gmail.com”
        sender_passwd = “kmko wohf Irdx gthw”
        connect.login(sender_email, sender_passwd)
        receiver_email = DataLst[-1]
        subject = "Laundry Service Bill"
        msg_text = "We Will Soon Pickup You Cloths….And Delivered Shortly"
        message = MIMEMultipart()
        message["From"] = sender_email
        message["To"] = receiver_email
        message["Subject"] = subject
        message["Bcc"] = receiver_email
        message.attach(MIMEText(msg_text, "plain"))
        filename = "Room_Service.pdf"
        with open(filename, "rb") as attachment:
          part = MIMEBase("application", "octet-stream")
          part.set_payload(attachment.read())
        encoders.encode_base64(part)
        part.add_header("Content-Disposition", f"attachment; filename= {filename}", )
        message.attach(part)
        text = message.as_string()
        connect.sendmail(sender_email, receiver_email, text)
        print("Successfully email✉ sent")
      except Exception as e:
        print(e)
      finally:
        connect.quit()
    Email()
    Food_Data = pd.read_csv("BillNo.csv", index_col=[0])
    Food_Data.BillNo[1] = int(Food_Data.BillNo[1]) + 1
    Food_Data.to_csv("BillNo.csv")
    root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Save_Print, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=40, text="Save &\nPrint",
bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=940, y=90+20-15)
def cls():
  root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=940,
y=90+20+50-5)
sn=1
```

```
def LiquorCart():
   # (sn, mydata2.get(), rat.get(), qnt.get(), amt.get(), dstprice.get(), ss_GST.get(), cc_GST.get(),
Tl_Amnt.get(),
   #  cal.get_date().strftime("%d-%m-%Y"), mydata.get()))
   global sn
   liqr.insert(parent="", index=0, values=(sn, mydata2.get(), rat.get(),
qnt.get(),amt.get(),dst.get(),s_GST.get(),c_GST.get(),Tl_Amnt.get()))
   sn += 1
   mydata2.set(value="Royal Challenge")
   rat.set(value=0)
   qnt.set(value=0)
   amt.set(value=0)
   dstprice.set(value=0)
   ss_GST.set(value=0)
   cc_GST.set(value=0)
   Tl_Amnt.set(value=0)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=LiquorCart, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=40, text="Add To\nCart",
bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=940, y=280-70)
def Liquorrm():
   try:
      liqr.delete(liqr.selection())
   except:
      pass
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Liquorrm, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Remove", bg="#a8701d",
anchor=W,font=('Century Gothic', 16, "bold"), borderwidth=5, cursor="hand2").place(x=940, y=340-70)
snn=1
def FoodCart():
   global snn
   table.insert(parent="", index=0, values=(
   snn, mydata1.get(), Foodrat.get(), Foodqnt.get(), Foodamt.get(), Fooddst.get(),Foods_GST.get(),
Foodc_GST.get(), FoodTl_Amnt.get()))
   snn += 1
   mydata1.set(value="Bhel Puri")
   Foodrat.set(value=0)
   Foodqnt.set(value=0)
   Foodamt.set(value=0)
   Fooddstprice.set(value=0)
   Foodss_GST.set(value=0)
   Foodcc_GST.set(value=0)
   FoodTl_Amnt.set(value=0)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=FoodCart, fg="Black", width=110,
activeforeground="black",activebackground="#a8701d", height=40, text="Add To\nCart", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=415, y=210)
def Foodrm():
   try:
      table.delete(table.selection())
   except:
      pass
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Foodrm, fg="Black", width=110,
activeforeground="black",activebackground="#a8701d", height=30, text="Remove", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=415, y=270)
```

```
#-----------------------------------------------------------------------------------------------------------
frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=40, y=530, width=490, height=150)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.","Fd_Nm", "Rate", "Quantity",
"Amount","Discount","S_GST","C_GST","Ttl_Amnt"),selectmode="browse",
xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Fd_Nm", text="Food Name", anchor=CENTER)
table.heading("Rate", text="Rate", anchor=CENTER)
table.heading("Quantity", text="Quantity", anchor=CENTER)
table.heading("Amount", text="Amount", anchor=CENTER)
table.heading("Discount", text="Discount", anchor=CENTER)
table.heading("S_GST", text="S GST", anchor=CENTER)
table.heading("C_GST", text="C GST", anchor=CENTER)
table.heading("Ttl_Amnt", text="Total Amount", anchor=CENTER)

table.pack(fill=BOTH, expand=1)


table["show"] = "headings"
table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
table.column("Fd_Nm", width=170, anchor=CENTER, minwidth=150)
table.column("Rate", width=100, anchor=CENTER, minwidth=70)
table.column("Quantity", width=140, anchor=CENTER, minwidth=120)
table.column("Amount", width=140, anchor=CENTER, minwidth=120)
table.column("Discount", width=140, anchor=CENTER, minwidth=120)
table.column("S_GST", width=140, anchor=CENTER, minwidth=120)
table.column("C_GST", width=140, anchor=CENTER, minwidth=120)
```

```
table.column("Ttl_Amnt", width=140, anchor=CENTER, minwidth=120)


pygame.mixer.init()
pygame.mixer.music.load("./Voices/Room Services.mp3")
pygame.mixer.music.play()


#-------------------------------------------------------------------------------------------------------------
frm2 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm2.place(x=550, y=530, width=490, height=150)
scbr_x = Scrollbar(frm2, orient=HORIZONTAL)
scbr_y = Scrollbar(frm2, orient=VERTICAL)
liqr = ttk.Treeview(frm2, cursor="hand2", columns=("SN_No.","Liqr_Nm", "Rate", "Quantity",
"Amount","Discount","S_GST","C_GST","Ttl_Amnt"),selectmode="browse",
xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=liqr.xview)
scbr_y.config(command=liqr.yview)
liqr.heading("SN_No.", text="Sn No.", anchor=CENTER)
liqr.heading("Liqr_Nm", text="Liquor Name", anchor=CENTER)
liqr.heading("Rate", text="Rate", anchor=CENTER)
liqr.heading("Quantity", text="Quantity", anchor=CENTER)
liqr.heading("Amount", text="Amount", anchor=CENTER)
liqr.heading("Discount", text="Discount", anchor=CENTER)
liqr.heading("S_GST", text="S GST", anchor=CENTER)
liqr.heading("C_GST", text="C GST", anchor=CENTER)
liqr.heading("Ttl_Amnt", text="Total Amount", anchor=CENTER)
liqr.pack(fill=BOTH, expand=1)


liqr["show"] = "headings"
liqr.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
liqr.column("Liqr_Nm", width=200, anchor=CENTER, minwidth=150)
liqr.column("Rate", width=100, anchor=CENTER, minwidth=70)
liqr.column("Quantity", width=140, anchor=CENTER, minwidth=120)
liqr.column("Amount", width=140, anchor=CENTER, minwidth=120)
liqr.column("Discount", width=140, anchor=CENTER, minwidth=120)
liqr.column("S_GST", width=140, anchor=CENTER, minwidth=120)
liqr.column("C_GST", width=140, anchor=CENTER, minwidth=120)
liqr.column("Ttl_Amnt", width=140, anchor=CENTER, minwidth=120)
#-------------------------------------------------------------------------------------------------------------
root.mainloop()
```


tk
Laundry Service
Bill Information
Bill Number :
L1
Bill Date :
12/25/23
Payment Mode :
Cash
Guest Details
Guest ID :
G1
Guest Name :
Saksham Jaiwal
Room Number :
R101
Save & Print
Close
Laundry Service Details
Service Name :
Washing
Rate :
10
Quantity :
Amount :
0.0
Discount :
2.0
%
0.0
SGST :
3.0
%
0.0
CGST :
3.0
%
0.0
Total Amount :
0.0
Add To Cart
Reset
Remove
Sn No.
Service Name
Rate
Quantity

Invoice

Billed to:

{{ name }}
{{ phone }}
{{Address}}

Invoice No.
{{Bill_No}}
{{Date}}

Item	Quantity	Unit Price	Total
	{{%tr for item in invoice_list %}}		
{{item[0]}}	{{item[1]}}	${{item[2]}}	${{item[3]}}
	{{%tr endfor %}}		

Subtotal	${{ subtotal }}
Tax (10%)	${{ salestax }}
Total	**${{ total }}**

Thank you

Payment Information

Cnd Bank
Account Name: Aaksham
Account No: 123-456-7890

Aaksham Hotel
Navi, Mumbai,In Front Of Ambani House

```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import pandas as pd
from tkcalendar import DateEntry
import mysql.connector as connector
from datetime import date
import os
height =700
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-20))
Label(text="Laundry Service",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
Data = pd.read_csv("BillNo.csv", index_col=[0])
query = f"select chkindtl.`Guest ID`,chkindtl.`Guest Name`,cstmdtl.`Contact No.`,CONCAT
(cstmdtl.Address, ', ', cstmdtl.City, ', ', cstmdtl.Country) as Address,cstmdtl.`Email ID`from `check in
details`chkindtl  right join `customer details` cstmdtl using (`Guest ID`) where chkindtl.`Room
No`='{Data.BillNo[4]}';"
cur.execute(query)
DataLst=cur.fetchone()
print(DataLst)
img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)
frame = customtkinter.CTkFrame(master=l1,width=1090,height=560,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))
customtkinter.CTkLabel(master=frame, text="Bill Information", font=('Times New Roman', 20,
"bold")).place(x=10, y=10)
customtkinter.CTkLabel(master=frame, text="Guest Details", font=('Times New Roman', 20,
"bold")).place(x=500+20, y=10)
customtkinter.CTkLabel(master=frame, text="Laundry Service Details", font=('Times New Roman', 20,
"bold")).place(x=10, y=170)
#-------------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Guest ID :", font=('Century Gothic', 16)).place(x=550, y=50)
customtkinter.CTkLabel(master=frame, text="Guest Name :", font=('Century Gothic', 16)).place(x=550,
y=90)
customtkinter.CTkLabel(master=frame, text="Room Number :", font=('Century Gothic', 16)).place(x=550,
y=130)
#-------------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Bill Number :", font=('Century Gothic', 16)).place(x=40,
```

```
y=50)
customtkinter.CTkLabel(master=frame, text="Bill Date :", font=('Century Gothic', 16)).place(x=40, y=90)
customtkinter.CTkLabel(master=frame, text="Payment Mode :", font=('Century Gothic', 16)).place(x=40,
y=130)
#-------------------------------------------------------------------------------------------------------------
# customtkinter.CTkLabel(master=frame, text="Bill Number :", font=('Century Gothic', 16)).place(x=40,
y=50)
# customtkinter.CTkLabel(master=frame, text="Bill Date :", font=('Century Gothic', 16)).place(x=40,
y=90)
# customtkinter.CTkLabel(master=frame, text="Payment Mode :", font=('Century Gothic',
16)).place(x=40, y=130)
# -------------------------------------------------------------------------------------------------------------
customtkinter.CTkLabel(master=frame, text="Service Name :", font=('Century Gothic', 16)).place(x=40,
y=210)
customtkinter.CTkLabel(master=frame, text="Rate :", font=('Century Gothic', 16)).place(x=40, y=250)
customtkinter.CTkLabel(master=frame, text="Quantity :", font=('Century Gothic', 16)).place(x=40, y=290)
customtkinter.CTkLabel(master=frame, text="Amount :", font=('Century Gothic', 16)).place(x=40, y=330)
customtkinter.CTkLabel(master=frame, text="Discount :", font=('Century Gothic', 16)).place(x=40, y=370)
customtkinter.CTkLabel(master=frame, text="SGST :", font=('Century Gothic', 16)).place(x=40, y=410)
customtkinter.CTkLabel(master=frame, text="CGST :", font=('Century Gothic', 16)).place(x=40, y=450)
customtkinter.CTkLabel(master=frame, text="Total Amount :", font=('Century Gothic', 16)).place(x=40,
y=490)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=410-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=450-40)
customtkinter.CTkLabel(master=frame, text="%", font=('Century Gothic', 16)).place(x=280, y=490-40)
#-------------------------------------------------------------------------------------------------------------
Gs_ID=StringVar()
Gs_ID.set(Data.BillNo[2])
Entry(frame,highlightthickness=2,textvariable=Gs_ID,highlightbackground="grey",highlightcolor="black",
fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=50,width=203,height=30)
Gs_Name=StringVar()
Gs_Name.set(Data.BillNo[3])
Entry(frame,highlightthickness=2,textvariable=Gs_Name,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=90,width=203,height=30)
Gs_RMNO=StringVar()
Gs_RMNO.set(Data.BillNo[4])
Entry(frame,highlightthickness=2,textvariable=Gs_RMNO,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic",state="disabled").place(x=700,y=130,width=203,height=30)
#-------------------------------------------------------------------------------------------------------------
Bill=StringVar()
Lndry_Data = pd.read_csv("Lndry_BillNo.csv", index_col=[0])
Bill.set(str(Lndry_Data.LndryBill[0])+str(Lndry_Data.LndryBill[1]))
Entry(frame,highlightthickness=2,textvariable=Bill,highlightbackground="grey",highlightcolor="black",fg
="blue",font="consolas 17 italic",state="disabled").place(x=200,y=50,width=203,height=30)
cal = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue", width=20)
cal.place(x=200, y=90)
#-------------------------------------------------------------------------------------------------------------
mydata = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=20,
state='readonly',background="grey", height=10)
mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
l = []
mydata.set("Cash")
mydata.place(x=200, y=130)
mydata1 = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=20,
```

```
state='readonly',background="grey", height=10)
mydata1["value"]=["Dry Cleaning","Washing"]
mydata1.set("Washing")
mydata1.place(x=200, y=210)
def mydata1_selected(e):
    if mydata1.get()=="Dry Cleaning":
        # print("Dry Cleaning")
        Lndryrat.set("55")

    else:
        # print("Washing")
        Lndryrat.set("10")
mydata1.bind('<<ComboboxSelected>>',mydata1_selected)

#-------------------------------------------------------------------------------------------------------------------
Lndryrat=StringVar()
Lndryrat.set("10")
Entry(frame,highlightthickness=2,textvariable=Lndryrat,highlightbackground="grey",highlightcolor="black
",fg="blue",font="consolas 17 italic").place(x=200,y=290-40,width=203,height=30)
Lndryqnt=StringVar()
Entry(frame,highlightthickness=2,textvariable=Lndryqnt,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=330-40,width=60,height=30)
Lndryamt=DoubleVar()
Entry(frame,highlightthickness=2,highlightbackground="grey",textvariable=Lndryamt,highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=370-40,width=203,height=30)
Lndrydst=DoubleVar()
Lndrydst.set(2.0)
Entry(frame,highlightthickness=2,textvariable=Lndrydst,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic").place(x=200,y=410-40,width=60,height=30)
Lndrys_GST=DoubleVar()
Lndrys_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Lndrys_GST,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic").place(x=200,y=450-40,width=60,height=30)
Lndryc_GST=StringVar()
Lndryc_GST.set(3.0)
Entry(frame,highlightthickness=2,textvariable=Lndryc_GST,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic").place(x=200,y=490-40,width=60,height=30)
LndryTl_Amnt=DoubleVar()
Entry(frame,highlightthickness=2,textvariable=LndryTl_Amnt,highlightbackground="grey",highlightcolor=
"black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=530-40,width=203,height=30)
Lndrydstprice=DoubleVar()
Entry(frame,highlightthickness=2,textvariable=Lndrydstprice,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=410-40,width=93,height=30)
Lndryss_GST=DoubleVar()
Entry(frame,highlightthickness=2,textvariable=Lndryss_GST,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=450-40,width=93,height=30)
Lndrycc_GST=DoubleVar()
Entry(frame,highlightthickness=2,textvariable=Lndrycc_GST,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="disabled").place(x=310,y=490-40,width=93,height=30)
#-------------------------------------------------------------------------------------------------------------------
def Lndrysliderevent(value):
    Lndryqnt.set(value=int(value))
    try:
        Lndryamt.set(value=float(value * float(Lndryrat.get())))
```

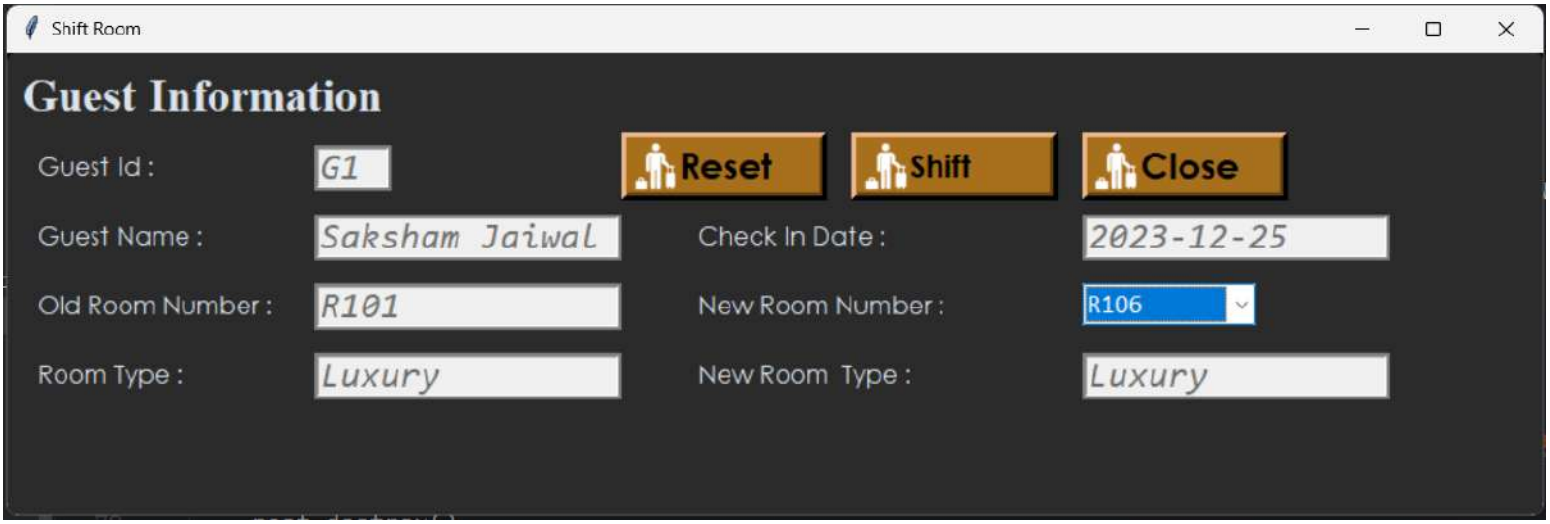
```
        Lndrydstprice.set(value=float((value * float(Lndryrat.get()) * float(Lndrydst.get()) / 100)))
        Lndryss_GST.set(value=float((value * float(Lndryrat.get()) * float(Lndrys_GST.get()) / 100)))
        Lndrycc_GST.set(value=float((value * float(Lndryrat.get()) * float(Lndryc_GST.get()) / 100)))
        LndryTl_Amnt.set(value=str(float(Lndryamt.get()) - float(Lndrydstprice.get()) +
float(Lndryss_GST.get()) + float(Lndrycc_GST.get())))
    except EXCEPTION as e:
        print(e)
    # print(value=int(value))
customtkinter.CTkSlider(frame,from_=0,to=10,command=Lndrysliderevent,number_of_steps=10,width=100
).place(x=270,y=335-40)
#-------------------------------------------------------------------------------------------------------------
snn=1
def LndryCart():
    global snn
    table.insert("",END, values=(
    snn, mydata1.get(), Lndryrat.get(), Lndryqnt.get(), Lndryamt.get(), Lndrydst.get(), Lndrys_GST.get(),
Lndryc_GST.get(), "{:.2f}".format(LndryTl_Amnt.get())))
    snn += 1
    mydata1.set(value="Washing")
    Lndryrat.set(value=10)
    Lndryqnt.set(value=0)
    Lndryamt.set(value=0)
    Lndrydstprice.set(value=0)
    Lndryss_GST.set(value=0)
    Lndrycc_GST.set(value=0)
    LndryTl_Amnt.set(value=0)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=LndryCart, fg="Black",
width=110, activeforeground="black",activebackground="#a8701d", height=40, text="Add To\nCart",
bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=415, y=210)
def rset():
    mydata1.set(value="Washing")
    Lndryrat.set(value=10)
    Lndryqnt.set(value=0)
    Lndryamt.set(value=0)
    Lndrydstprice.set(value=0)
    Lndryss_GST.set(value=0)
    Lndrycc_GST.set(value=0)
    LndryTl_Amnt.set(value=0)
    global snn
    for rmv in table.get_children():
        table.delete(rmv)
    snn=1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=rset, fg="Black", width=110,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=415, y=270)
def Lndryrm():
    try:
        table.delete(table.selection())
    except:
        pass
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Lndryrm, fg="Black", width=110,
activeforeground="black",activebackground="#a8701d", height=30, text="Remove", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=415, y=320)
```

```
#-----------------------------------------------------------------------------------------------------------
# def New():
#     for item in table.get_children():
#         table.delete(item)
#     cal.set_date(date.today())
#     mydata.set(value="Cash")
#     mydata1.set(value="Washing")
#     Lndryrat.set(value=0)
#     Lndryqnt.set(value=0)
#     Lndryamt.set(value=0)
#     Lndrydstprice.set(value=0)
#     Lndryss_GST.set(value=0)
#     Lndrycc_GST.set(value=0)
#     LndryTl_Amnt.set(value=0)
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="New",command=New,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=940, y=40+20-15)
def Save_Print():
    if messagebox.askyesno("Laundry Bill","Are You Sure You Want To Bill Out"):
        Bdt = cal.get_date().strftime("%Y-%m-%d")
        query = f"INSERT INTO `Laundry Service Details` VALUES ('{Gs_ID.get()}', '{Gs_Name.get()}',
'{Gs_RMNO.get()}', '{Bill.get()}', '{Bdt}','{mydata.get()}');"
        print(query)
        cur.execute(query)
        con.commit()
        for i in table.get_children():
            l = table.item(i)["values"]
            query = f"INSERT INTO `Laundry Details` VALUES ('{Bill.get()}', '{l[1]}', '{l[2]}', '{l[3]}',
'{l[4]}','{l[5]}', '{l[6]}', '{l[7]}', '{l[8]}','{date.today()}');"
            print(query)
            cur.execute(query)
            con.commit()
        from docx2pdf import convert
        from docxtpl import DocxTemplate
        import smtplib
        from email import encoders
        from email.mime.base import MIMEBase
        from email.mime.multipart import MIMEMultipart
        from email.mime.text import MIMEText
        doc = DocxTemplate("invoice_template.docx")
        invoice_list = []
        tl = 0
        for i in table.get_children():
            invoice_list.append(table.item(i)["values"])
            tl += float(table.item(i)["values"][8])
        print(tl)
        print(invoice_list)
        TtlAmnt=0
        for i in invoice_list:
            TtlAmnt+=float(i[8])
        Lndry_Data = pd.read_csv("Lndry_BillNo.csv", index_col=[0])
        doc.render({"Bill_No": f"{Lndry_Data.LndryBill[0] + Lndry_Data.LndryBill[1]}",
                    "Date": f"{date.today().strftime('%d/%m/%Y')}",
```

```
            "name": f"{Gs_Name.get()}",
            "phone": DataLst[2],
            "Address": DataLst[3],
            "date": f'{date.today().strftime("%d/%b/%y")}',
            "subtotal": f'{TtlAmnt}',
            "invoice_list": invoice_list,
            "salestax": "10%",
            "total": f"{tl+tl*0.1}"})
    doc.save("Laundry invoice.docx")
    convert(r"D:\python\Project\Laundry invoice.docx", r"D:\python\Project\Laundry.pdf")
    os.remove(r"D:\python\Project\Laundry invoice.docx")
    os.system("Laundry.pdf")

    def Email():
      try:
        connect = smtplib.SMTP('smtp.gmail.com', 587)
        connect.ehlo()
        connect.starttls()
        sender_email = "aakshamhotel@gmail.com"
        sender_passwd = "kmko wohf Irdx gthw"
        connect.login(sender_email, sender_passwd)
        receiver_email = DataLst[-1]
        subject = "Laundry Service Bill"
        msg_text = "We Will Soon Pickup You Cloths….And Delivered Shortly"
        message = MIMEMultipart()
        message["From"] = sender_email
        message["To"] = receiver_email
        message["Subject"] = subject
        message["Bcc"] = receiver_email
        message.attach(MIMEText(msg_text, "plain"))
        filename = "Laundry.pdf"
        with open(filename, "rb") as attachment:
          part = MIMEBase("application", "octet-stream")
          part.set_payload(attachment.read())
        encoders.encode_base64(part)
        part.add_header("Content-Disposition", f"attachment; filename= {filename}", )
        message.attach(part)
        text = message.as_string()
        connect.sendmail(sender_email, receiver_email, text)
        print("Successfully email✉ sent")
      except Exception as e:
        print(e)
      finally:
        connect.quit()
    Email()
    Lndry_Data = pd.read_csv("Lndry_BillNo.csv", index_col=[0])
    Lndry_Data.LndryBill[1] = int(Lndry_Data.LndryBill[1]) + 1
    Lndry_Data.to_csv("Lndry_BillNo.csv")
    root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Save_Print, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=40, text="Save &\nPrint",
bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=940, y=90+20-15)
```

```
def cls():
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=940,
y=90+20+50-5)

#-------------------------------------------------------------------------------------------------------------
frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=550, y=210, width=520, height=300)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.","Fd_Nm", "Rate", "Quantity",
"Amount","Discount","S_GST","C_GST","Ttl_Amnt"),selectmode="browse",
xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Fd_Nm", text="Service Name", anchor=CENTER)
table.heading("Rate", text="Rate", anchor=CENTER)
table.heading("Quantity", text="Quantity", anchor=CENTER)
table.heading("Amount", text="Amount", anchor=CENTER)
table.heading("Discount", text="Discount", anchor=CENTER)
table.heading("S_GST", text="S GST", anchor=CENTER)
table.heading("C_GST", text="C GST", anchor=CENTER)
table.heading("Ttl_Amnt", text="Total Amount", anchor=CENTER)

table.pack(fill=BOTH, expand=1)
def table_selected(_):
   pass

table["show"] = "headings"
```

```
table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
table.column("Fd_Nm", width=170, anchor=CENTER, minwidth=150)
table.column("Rate", width=100, anchor=CENTER, minwidth=70)
table.column("Quantity", width=140, anchor=CENTER, minwidth=120)
table.column("Amount", width=140, anchor=CENTER, minwidth=120)
table.column("Discount", width=140, anchor=CENTER, minwidth=120)
table.column("S_GST", width=140, anchor=CENTER, minwidth=120)
table.column("C_GST", width=140, anchor=CENTER, minwidth=120)
table.column("Ttl_Amnt", width=140, anchor=CENTER, minwidth=120)
table.bind('<<TreeviewSelect>>',table_selected)

pygame.mixer.init()
pygame.mixer.music.load("./Voices/Laundry Services.mp3")
pygame.mixer.music.play()


root.mainloop()
```



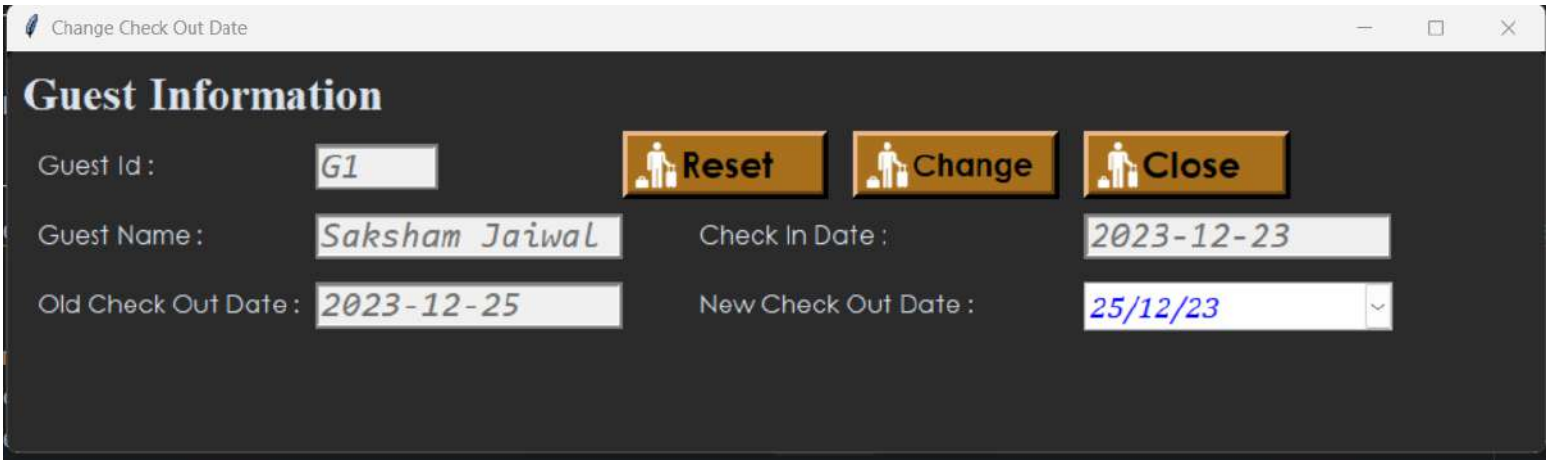

```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import mysql.connector as connector
from tkcalendar import DateEntry
from datetime import date
import pandas as pd
import os

import pygame
height = 600
width = 1000

root=Tk()
root.maxsize("1000","300")
root.minsize("1000","300")
root.geometry('{}x{}+{}+{}'.format(1000,300,300,300))
root.title("Shift Room")
frame = customtkinter.CTkFrame(master=root,bg_color="black")
frame.pack(fill=BOTH,expand=1)
```

```
Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Guest Information", font=('Times New Roman', 30,
"bold")).place(x=10, y=10)
# def Gstry():
#     # GsID.set("")
#     # GsNm.set("")
#     # Gsder.set("")
#     # Gsgion.set("")
#     # GsAddress.set("")
#     # GsCity.set("")
#     # GsCntry.set("")
#     # GsCntNO.set("")
#     # GsIDType.set("")
#     # GsIDNo.set("")
#     os.system("python Gs_Entry.py")
#     Gs = pd.read_csv("hi.csv", index_col=[0])
#     GsID.set(Data.hii[1])
#     GsNm.set(Data.hii[2])
#     Gsder.set(Data.hii[3])
#     Gsgion.set(Data.hii[4])
#     GsAddress.set(Data.hii[5])
#     GsCity.set(Data.hii[6])
#     GsCntry.set(Data.hii[7])
#     GsCntNO.set(Data.hii[8])
#     GsIDType.set(Data.hii[9])
#     GsIDNo.set(Data.hii[10])
#     # os.close("Main_Box.py")
#     # print(5656565655555555555555555555)
def reset():
    mydata.set("")
    NwRmType.set("")
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=reset,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=300+100, y=52)
def CHk():
    if mydata.get() == "":
        messagebox.showerror("Shift Room","Select The Room")
    else:
        Cnf=messagebox.askyesno("Change Room","Are You Sure You Want To Change")
        if Cnf:
            query = f"select `Room Price` from `Check In Details` where `Room No`='{OldRmNo.get()}';"
            cur.execute(query)
            query = f"insert into `room status` values
('{OldRmNo.get()}','{RmType.get()}','Dirty','{cur.fetchone()[0]}');"
            cur.execute(query)
            con.commit()
            # prc = 0
            # for i in dict:
            #     if i == mydata.get():
            #         prc = dict[i][1]
            query = (
                f"UPDATE `Check In Details` SET `Room No` = '{mydata.get()}',`Room
```

```
Type`='{NwRmType.get()}',`Room Original Price`='{DataLst[-1]}'"
            f"WHERE (`Guest ID` = '{Data.BillNo[2]}');")
        cur.execute(query)
        con.commit()
        con.close()
        root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=CHk, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Shift", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=400+150, y=52)
def cls():
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=500+200, y=52)
def idd():
   customtkinter.CTkLabel(master=frame,text="Guest Id :",font=('Century Gothic',16)).place(x=20, y=60)
   customtkinter.CTkLabel(master=frame,text="Guest Name :",font=('Century Gothic',16)).place(x=20,
y=105)
   customtkinter.CTkLabel(master=frame,text="Old Room Number :",font=('Century
Gothic',16)).place(x=20, y=150)
   customtkinter.CTkLabel(master=frame,text="Room Type :",font=('Century Gothic',16)).place(x=20,
y=195)
   customtkinter.CTkLabel(master=frame,text="Check In Date :",font=('Century Gothic',16)).place(x=450,
y=105)
   customtkinter.CTkLabel(master=frame,text="New Room Number :",font=('Century
Gothic',16)).place(x=450, y=150)
   customtkinter.CTkLabel(master=frame,text="New Room  Type :",font=('Century
Gothic',16)).place(x=450, y=195)
idd()
GsID=StringVar()
GsNm=StringVar()
OldRmNo=StringVar()
RmType=StringVar()
ChkInDt=StringVar()
NwRmType=StringVar()

con=connector.connect(host='localhost',
                port='3306',
                user='root',
                password='Password',
                database='Hotel Management Software')
cur=con.cursor()
Data = pd.read_csv("BillNo.csv", index_col=[0])
query = f"select * from `check in details` where `Room No`='{Data.BillNo[4]}';"
cur.execute(query)
DataLst=cur.fetchone()
print(DataLst)

GsID.set(Data.BillNo[2])
GsNm.set(Data.BillNo[3])
OldRmNo.set(Data.BillNo[4])
RmType.set(Data.BillNo[5])
ChkInDt.set(Data.BillNo[7])
NwRmType.set("")
```

```
Entry(frame,highlightthickness=2,textvariable=GsID,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=60,width=50,height=30)
Entry(frame,highlightthickness=2,textvariable=GsNm,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=105,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=OldRmNo,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=150,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=RmType,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=195,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=ChkInDt,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=700, y=105,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=NwRmType,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=700, y=195,width=200,height=30)
#
Entry(frame,highlightthickness=2,textvariable=GsCntry,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=700, y=195,width=200,height=30)
con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Hotel Management Software')
cur = con.cursor()
query = "select `Room No.`,`room Type`,Price from `room status` where `status`='vacant';"
cur.execute(query)
l=[]
dict={}
for i in cur.fetchall():
    l.append(i[0])
    dict[i[0]]=[i[1],i[2]]
# print(dict)
mydata = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=10,
state='readonly',background="grey", height=10)
mydata["value"]=l
def mydata_selected(_):
    for i in dict:
        if i == mydata.get():
            NwRmType.set(dict[i][0])
mydata.bind('<<ComboboxSelected>>',mydata_selected)
# for j in range(1, 501):
#     l.append(f"API {j}")
#
# mydata["value"] = 1
mydata.place(x=700, y=150)
# mydata1 = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=10,
state='readonly',background="grey", height=10)
# mydata1["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
# l = []
# for j in range(1, 501):
#     l.append(f"API {j}")
#
# mydata["value"] = 1
# mydata1.set("Cash")
# mydata1.place(x=700, y=195)
pygame.mixer.init()
```

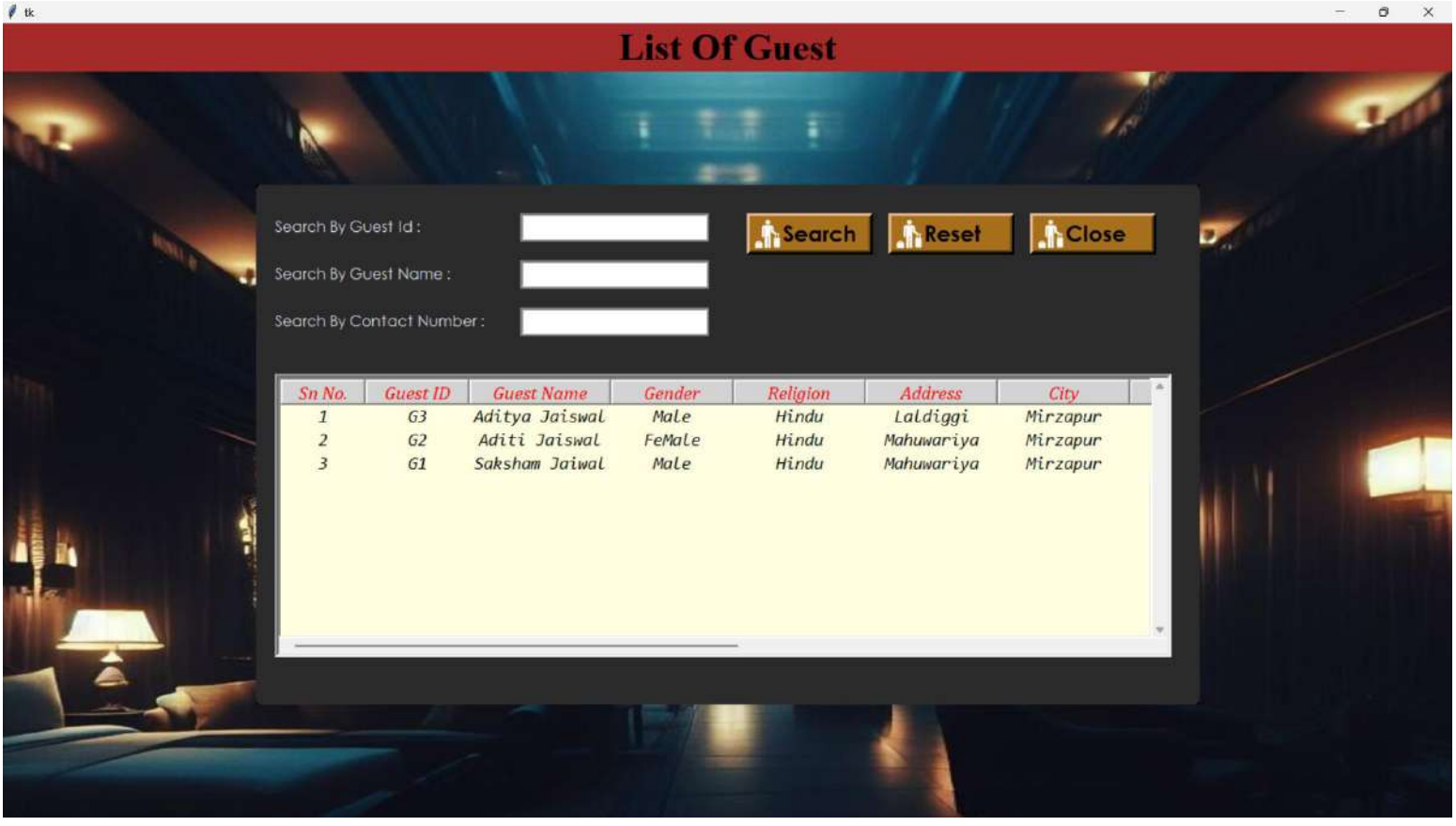
```
pygame.mixer.music.load("./Voices/Shift Room.mp3")
pygame.mixer.music.play()

root.mainloop()
```




```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
from tkcalendar import DateEntry
from datetime import date
import pandas as pd
import mysql.connector as connector
import os

import pygame
height = 600
width = 1000

root=Tk()
root.maxsize("1000","260")
root.minsize("1000","260")
root.geometry('{}x{}+{}+{}'.format(1000,260,300,300))
root.title("Change Check Out Date")
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
Data = pd.read_csv("BillNo.csv", index_col=[0])
query = f"select * from `check in details` where `Room No`='{Data.BillNo[4]}';"
cur.execute(query)
DataLst=cur.fetchone()
print(DataLst)
frame = customtkinter.CTkFrame(master=root,bg_color="black")
frame.pack(fill=BOTH,expand=1)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))
```

```
customtkinter.CTkLabel(master=frame, text="Guest Information", font=('Times New Roman', 30,
"bold")).place(x=10, y=10)
def reset():
   cal.set_date(date.today())
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=reset,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=300+100, y=52)
def Cng():
   from datetime import datetime
   chin_date = datetime.strptime(ChkinDt.get(), "%Y-%m-%d")
   past_date = datetime.strptime(cal.get_date().strftime("%Y-%m-%d"), "%Y-%m-%d")
   Nm_Dys=(past_date-chin_date).days*DataLst[-1]
   if Nm_Dys == 0.0:
      Nm_Dys = 1
   if (past_date-chin_date).days >= 0:
      chgval=messagebox.askyesno("Change Check Out Date","Are You Sure You Want To Change Date")
      if chgval:
         query = f"UPDATE `Check In Details` SET `Day Out` = '{past_date.date()}',`Room Price` =
'{Nm_Dys*DataLst[-1]}' WHERE (`Guest ID` = '{Data.BillNo[2]}');"
         cur.execute(query)
         con.commit()
         con.close()
         root.destroy()
      else:
         pass
   else:
      messagebox.showinfo("Change Check Out Date","Checkout Date Must Greater Than Check In Date")
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Cng, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Change", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=400+150, y=52)
def cls():
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=500+200, y=52)
def idd():
   customtkinter.CTkLabel(master=frame,text="Guest Id :",font=('Century Gothic',16)).place(x=20, y=60)
   customtkinter.CTkLabel(master=frame,text="Guest Name :",font=('Century Gothic',16)).place(x=20,
y=105)
   customtkinter.CTkLabel(master=frame,text="Old Check Out Date :",font=('Century
Gothic',16)).place(x=20, y=150)

   customtkinter.CTkLabel(master=frame,text="Check In Date :",font=('Century Gothic',16)).place(x=450,
y=105)
   customtkinter.CTkLabel(master=frame,text="New Check Out Date :",font=('Century
Gothic',16)).place(x=450, y=150)

idd()
GsID=StringVar()
GsNm=StringVar()
Gsder=StringVar()
oldChkDt=StringVar()
ChkinDt=StringVar()
```

```
GsID.set(value=Data.BillNo[2])
GsNm.set(value=Data.BillNo[3])
oldChkDt.set(value=Data.BillNo[7])
ChkinDt.set(value=Data.BillNo[6])
Entry(frame,highlightthickness=2,textvariable=GsID,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=60,width=80,height=30)
Entry(frame,highlightthickness=2,textvariable=GsNm,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=105,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=oldChkDt,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=200,y=150,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=ChkinDt,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=700, y=105,width=200,height=30)
#
Entry(frame,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=700, y=150,width=200,height=30)
s = ttk.Style()
s.theme_use("xpnative")  # classic , alt,default , winnative , xpnative , clam , vista
s.configure(".", font=("consolas", 14, "italic"), foreground="blue")
s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
# s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light grey")
cal = DateEntry(frame, selectmode="day", font=("Cambria", 15,
"italic"),date_pattern='dd/mm/yy',foreground="blue", width=16)
cal.place(x=700, y=150)

pygame.mixer.init()
pygame.mixer.music.load("./Voices/Change Check Out Date.mp3")
pygame.mixer.music.play()


root.mainloop()
```



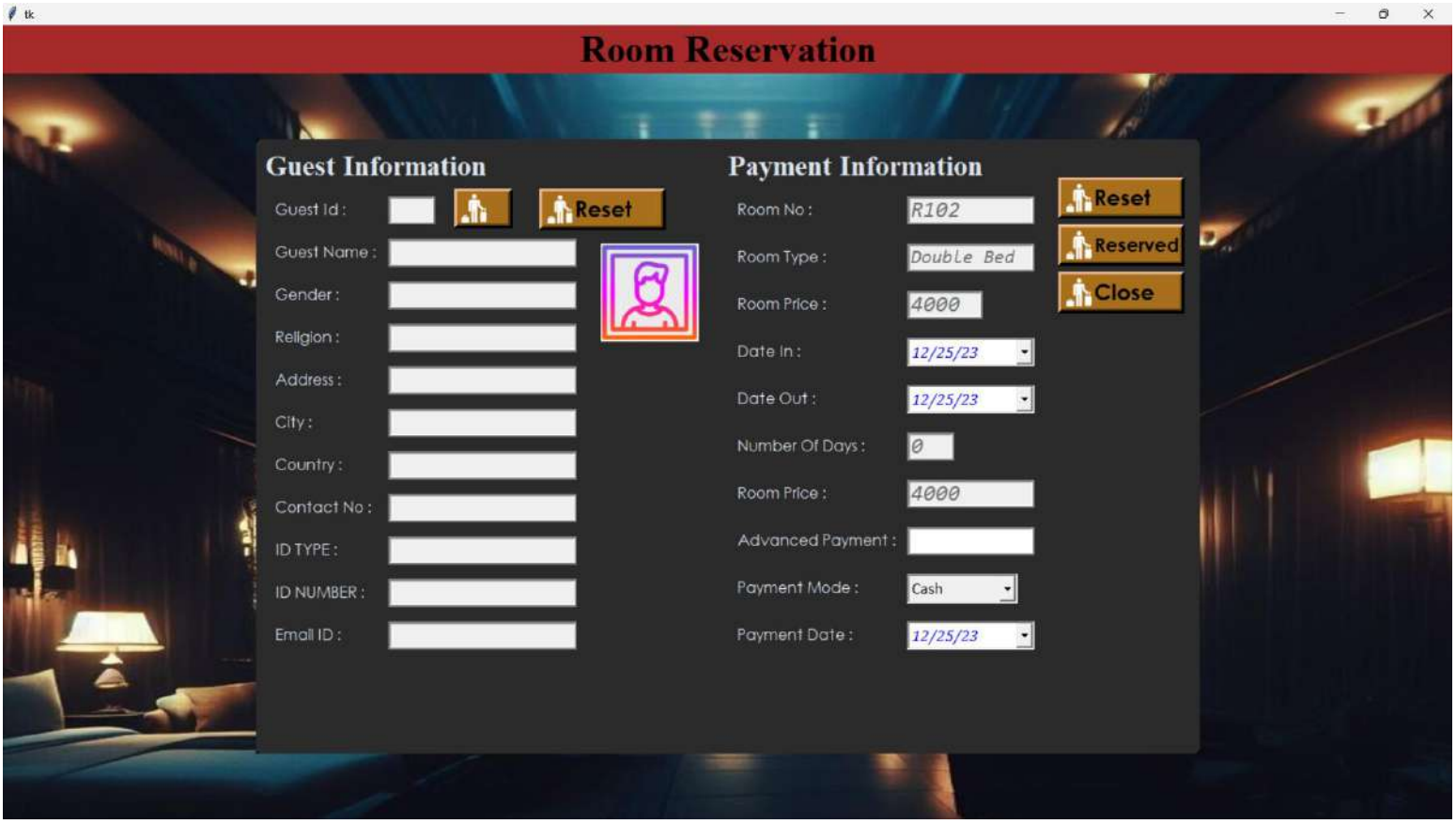
```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
import pandas as pd
import mysql.connector as connector
from tkinter import  messagebox
from PIL import ImageTk,Image
import pandas as pd
from tkcalendar import DateEntry
from datetime import date
import os
height = 730
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))

con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()

Label(text="List Of Guest",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)

img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=550,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Search By Guest Id :", font=('Century Gothic',
16)).place(x=20, y=30)
customtkinter.CTkLabel(master=frame, text="Search By Guest Name :", font=('Century Gothic',
16)).place(x=20, y=80)
customtkinter.CTkLabel(master=frame, text="Search By Contact Number :", font=('Century Gothic',
16)).place(x=20, y=130)
E1Var=StringVar()
E1Var.set("")
E2Var=StringVar()
E2Var.set("")
E3Var=StringVar()
E3Var.set("")
E1=Entry(frame,highlightthickness=2,textvariable=E1Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E1.place(x=280,y=30,width=200,height=30)
E2=Entry(frame,highlightthickness=2,textvariable=E2Var,highlightbackground="grey",highlightcolor="bla
```

```
ck",fg="blue",font="consolas 17 italic")
E2.place(x=280,y=80,width=200,height=30)
E3=Entry(frame,highlightthickness=2,textvariable=E3Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E3.place(x=280,y=130,width=200,height=30)
def Scrh():
    if E1.get()==""and E2.get()=="":
        for item in table.get_children():
            table.delete(item)
        query = f"select* from `Customer Details` where `Contact No.`='{E3.get()}' order by `S No.` desc;"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            table.insert("", END, values=(
                sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[12]))
            sn += 1
    elif E2.get()==""and E3.get()=="":
        for item in table.get_children():
            table.delete(item)
        query = f"select* from `Customer Details` where `Guest ID` ='{E1.get()}' order by `S No.` desc;"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            table.insert("", END, values=(
            sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[12]))
            sn += 1
    elif E1.get()==""and E3.get()=="":
        for item in table.get_children():
            table.delete(item)
        query = f"select* from `Customer Details` where `Guest Name`='{E2.get()}' order by `S No.` desc;"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            table.insert("", END, values=(
            sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[12]))
            sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Search",command=Scrh,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=500+20, y=30)
def Rset():
    E1Var.set("")
    E2Var.set("")
    E3Var.set("")
    for item in table.get_children():
        table.delete(item)
    query = "select* from `Customer Details` order by `S No.` desc;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(
        sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[12]))
        sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
```

```
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=Rset,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=650+20, y=30)
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Excel", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650, y=30)
def cls():
    Gs = pd.read_csv("hi.csv", index_col=[0])
    Gs.hii[0]=""
    Gs.hii[1]=""
    Gs.hii[2]=""
    Gs.hii[3]=""
    Gs.hii[4]=""
    Gs.hii[5]=""
    Gs.hii[6]=""
    Gs.hii[7]=""
    Gs.hii[8]=""
    Gs.hii[9]=""
    Gs.hii[10]=""
    Gs.hii[11]=""
    Gs.to_csv("hi.csv")
    root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=800+20, y=30)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=20, y=200, width=950, height=300)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Gs_ID", "Gs_Name",
"Gender","Religion","Address","City","Country","Contact_No","ID_Type","ID_No","Email_ID","Img
"),
                     selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

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

```
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Gs_ID", text="Guest ID", anchor=CENTER)
table.heading("Gs_Name", text="Guest Name", anchor=CENTER)
table.heading("Gender", text="Gender", anchor=CENTER)
table.heading("Religion", text="Religion", anchor=CENTER)
table.heading("Address", text="Address", anchor=CENTER)
table.heading("City", text="City", anchor=CENTER)
table.heading("Country", text="Country", anchor=CENTER)
table.heading("Contact_No", text="Contact Number", anchor=CENTER)
table.heading("ID_Type", text="ID Type", anchor=CENTER)
table.heading("ID_No", text="ID No", anchor=CENTER)
table.heading("Email_ID", text="Email ID", anchor=CENTER)
table.heading("Img", text="Image", anchor=CENTER)
table.pack(fill=BOTH, expand=1)


table["show"] = "headings"
table.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
table.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
table.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
table.column("Gender", width=130, anchor=CENTER, minwidth=130)
table.column("Religion", width=140, anchor=CENTER, minwidth=140)
table.column("Address", width=140, anchor=CENTER, minwidth=140)
table.column("City", width=140, anchor=CENTER, minwidth=140)
table.column("Country", width=140, anchor=CENTER, minwidth=140)
table.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
table.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
table.column("ID_No", width=140, anchor=CENTER, minwidth=120)
table.column("Email_ID", width=300, anchor=CENTER, minwidth=120)
table.column("Img", width=0, anchor=CENTER, minwidth=0)
#
table.insert(parent="",index=0,values=(1,101,"Anjali","Female","Hinduism","Jodhpur","Rajasthan","Bh
arat","+919102020202","Passport",'1P85DF256'))
# table.insert(parent="",index=1,values=(2,102,"Mohan","Male","Hinduism","Lucknow","Uttar
Pradesh","Bharat","+919105050505","Aadhar Card",'12121212121'))
#
table.insert(parent="",index=2,values=(3,103,"Rakesh","Transgender","None","Kerala","Kerala","Bhara
t","+919103003302","Passport",'1P25RF56456'))
query = "select* from `Customer Details` order by `S No.` desc;"
cur.execute(query)
sn=1
for row in cur.fetchall():
   table.insert("",END, values=(sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9],
row[10],row[12],row[13]))
   sn+=1
def table_select(_):
   fu=[]
   for i in table.selection():
      fu.append(table.item(i)['values'])
   so = pd.Series(data=fu[0], name="hii")
   sep = pd.DataFrame(so)
```

```
    sep.to_csv("hi.csv")
    # print(so)
    root.destroy()
table.bind('<<TreeviewSelect>>',table_select)

root.mainloop()
```




```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
from tkcalendar import DateEntry
from datetime import date
import pandas as pd
import mysql.connector as connector
import os

import pygame
height = 730
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
Label(text="Room Reservation",background="brown",font=('Times New
```

```
Roman',30,"bold")).pack(anchor=N,fill=X)

img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=650,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Guest Information", font=('Times New Roman', 30,
"bold")).place(x=10, y=10)
customtkinter.CTkLabel(master=frame, text="Payment Information", font=('Times New Roman', 30,
"bold")).place(x=500, y=10)
def Gstry():
    os.system("python Gs_Entry.py")
    Gs = pd.read_csv("hi.csv", index_col=[0])
    GsID.set(Gs.hii[1])
    GsNm.set(Gs.hii[2])
    Gsder.set(Gs.hii[3])
    Gsgion.set(Gs.hii[4])
    GsAddress.set(Gs.hii[5])
    GsCity.set(Gs.hii[6])
    GsCntry.set(Gs.hii[7])
    GsCntNO.set(Gs.hii[8])
    GsIDType.set(Gs.hii[9])
    GsIDNo.set(Gs.hii[10])
    GsEmlID.set(Gs.hii[11])
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open(Gs.hii[12]).resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
tkinter.Button(frame, image=Guest_Entry,compound=CENTER,command=Gstry, fg="Black", width=50,
activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", anchor=W,
borderwidth=5, cursor="hand2").place(x=210, y=52)
CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
Imglbl=Label(frame,image=CstmrImg)
Imglbl.place(x=365, y=110)
def reset():
    GsID.set("")
    GsNm.set("")
    Gsder.set("")
    Gsgion.set("")
    GsAddress.set("")
    GsCity.set("")
    GsCntry.set("")
    GsCntNO.set("")
    GsIDType.set("")
    GsIDNo.set("")
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=reset,
```

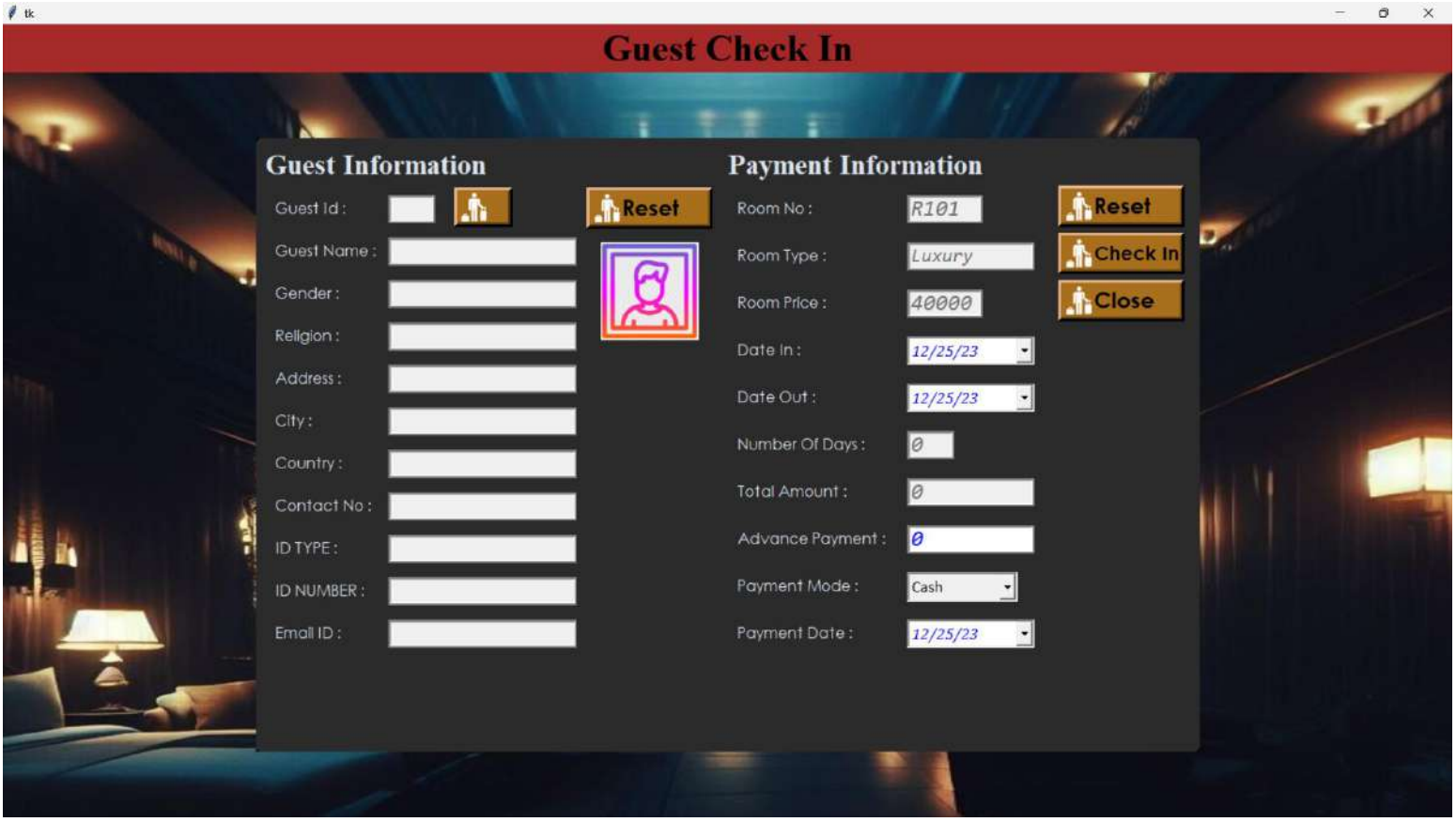
```
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=300, y=52)
def rr():
   Pymnt.set(0)
   mydata.set("Cash")
   cal.set_date(date.today())
   cal1.set_date(date.today())
   cal2.set_date(date.today())
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=rr, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=40)
sn=1
def Rsvrd():
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   c = cal1.get_date().strftime("%Y-%m-%d")
   c1 = cal2.get_date().strftime("%Y-%m-%d")
   c2 = cal.get_date().strftime("%Y-%m-%d")
   query = f"insert into `Reservation Details` values
('{GsID.get()}','{GsNm.get()}','{Rm.get()}','{Rm_Type.get()}','{c}','{c1}','{rm_price.get()}','{Pymnt.get()}','{
mydata.get()}','{c2}','Active','Reservation');"
   cur.execute(query)
   query = f"Delete from `Room Status` where `Room No.`='{Rm.get()}';"
   cur.execute(query)
   print(query)
   query = f"Update `customer details` set `Status`='Active' where `Guest ID`='{GsID.get()}';"
   cur.execute(query)
   print(query)
   con.commit()
   GsID.set("")
   GsNm.set("")
   Gsder.set("")
   Gsgion.set("")
   GsAddress.set("")
   GsCity.set("")
   GsCntry.set("")
   GsCntNO.set("")
   GsIDType.set("")
   GsIDNo.set("")
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Rsvrd, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reserved", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=90)
def cls():
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=140)
def idd():
   customtkinter.CTkLabel(master=frame,text="Guest Id :",font=('Century Gothic',16)).place(x=20, y=60)
```

```
   customtkinter.CTkLabel(master=frame,text="Guest Name :",font=('Century Gothic',16)).place(x=20,
y=105)
   customtkinter.CTkLabel(master=frame,text="Gender :",font=('Century Gothic',16)).place(x=20, y=150)
   customtkinter.CTkLabel(master=frame,text="Religion :",font=('Century Gothic',16)).place(x=20, y=195)
   customtkinter.CTkLabel(master=frame,text="Address :",font=('Century Gothic',16)).place(x=20, y=240)
   customtkinter.CTkLabel(master=frame,text="City :",font=('Century Gothic',16)).place(x=20, y=285)
   customtkinter.CTkLabel(master=frame,text="Country :",font=('Century Gothic',16)).place(x=20, y=330)
   customtkinter.CTkLabel(master=frame,text="Contact No :",font=('Century Gothic',16)).place(x=20,
y=375)
   customtkinter.CTkLabel(master=frame,text="ID TYPE :",font=('Century Gothic',16)).place(x=20,
y=420)
   customtkinter.CTkLabel(master=frame,text="ID NUMBER :",font=('Century Gothic',16)).place(x=20,
y=465)
   customtkinter.CTkLabel(master=frame,text="Email ID :",font=('Century Gothic',16)).place(x=20,
y=510)

   #-------------------------------------------------------------------------------------------
   customtkinter.CTkLabel(master=frame, text="Room No :", font=('Century Gothic', 16)).place(x=510,
y=60)
   customtkinter.CTkLabel(master=frame, text="Room Type :", font=('Century Gothic', 16)).place(x=510,
y=110)
   customtkinter.CTkLabel(master=frame, text="Room Price :", font=('Century Gothic', 16)).place(x=510,
y=160)
   customtkinter.CTkLabel(master=frame, text="Date In :", font=('Century Gothic', 16)).place(x=510,
y=180+30)
   customtkinter.CTkLabel(master=frame, text="Date Out :", font=('Century Gothic', 16)).place(x=510,
y=220+40)
   customtkinter.CTkLabel(master=frame, text="Number Of Days :", font=('Century Gothic',
16)).place(x=510, y=260+50)
   customtkinter.CTkLabel(master=frame, text="Room Price :", font=('Century Gothic', 16)).place(x=510,
y=300+60)
   customtkinter.CTkLabel(master=frame, text="Advanced Payment :", font=('Century Gothic',
16)).place(x=510,y=340+70)
   customtkinter.CTkLabel(master=frame, text="Payment Mode :", font=('Century Gothic',
16)).place(x=510, y=380+80)
   customtkinter.CTkLabel(master=frame, text="Payment Date :", font=('Century Gothic',
16)).place(x=510, y=380+40+90)
idd()
GsID=StringVar()
GsNm=StringVar()
Gsder=StringVar()
Gsgion=StringVar()
GsAddress=StringVar()
GsCity=StringVar()
GsCntry=StringVar()
GsCntNO=StringVar()
GsIDType=StringVar()
GsIDNo=StringVar()
GsEmlID=StringVar()
GsID.set("")
GsNm.set("")
Gsder.set("")
Gsgion.set("")
GsAddress.set("")
```

```
GsCity.set("")
GsCntry.set("")
GsCntNO.set("")
GsIDType.set("")
GsIDNo.set("")
GsEmlID.set("")
Entry(frame,highlightthickness=2,textvariable=GsID,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=60,width=50,height=30)
Entry(frame,highlightthickness=2,textvariable=GsNm,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=105,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=Gsder,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=150,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=Gsgion,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=195,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsAddress,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=240,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCity,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=285,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCntry,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=330,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCntNO,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=375,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsIDType,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=420,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsIDNo,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=465,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsEmlID,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=510,width=200,height=30)
Rm=StringVar()
# file1 = open("readme.txt", "r")
file = pd.read_csv('RM_Reservation.csv')
Rm.set(file.RoomDetails[1])
Rm_Type=StringVar()
Rm_Type.set(file.RoomDetails[2])
Rm_Prc=StringVar()
Rm_Prc.set(file.RoomDetails[3])
Days=StringVar()
rm_price=StringVar()
rm_price.set(value=file.RoomDetails[3])
Pymnt=StringVar()
Entry(frame,textvariable=Rm,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=60,width=135,height=30)
Entry(frame,textvariable=Rm_Type,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 15 italic",state="disabled").place(x=690,y=100+10,width=135,height=30)
Entry(frame,textvariable=Rm_Prc,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=100+40+20,width=80,height=30)
Entry(frame,highlightthickness=2,textvariable=Days,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=220+40+50,width=50,height=30)
Entry(frame,highlightthickness=2,state="disabled",textvariable=rm_price,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic").place(x=690,y=360,width=135,height=30)
en=Entry(frame,highlightthickness=2,textvariable=Pymnt,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
en.place(x=690,y=300+110,width=135,height=30)
cal1 = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue",
```

```
width=10+2)
cal1.place(x=690, y=140+40+30)
cal2 = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue",
width=10+2)
cal2.place(x=690, y=180+40+40)
cal = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue", width=10+2)
cal.place(x=690, y=380+40+90)
def cal2_calevent_create(_):
   rm_price.set(value=int(file.RoomDetails[3])*(cal2.get_date()-cal1.get_date()).days)
   print(int(file.RoomDetails[3])*(cal2.get_date()-cal1.get_date()).days)
   Days.set((cal2.get_date() - cal1.get_date()).days)
cal2.bind("<<DateEntrySelected>>", cal2_calevent_create)
Days.set((cal2.get_date() - cal1.get_date()).days)
   #----------------------------------     SLIDER     ----------------------------------------
   #-------------------------------------------------------------------------------------------
mydata = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=10,
state='readonly',background="grey", height=10)

mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
l = []
# for j in range(1, 501):
#     l.append(f"API {j}")
#
# mydata["value"] = l
mydata.set("Cash")
mydata.place(x=690, y=340+40+80)
#-------------------------------------------------------------------------------------------

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")
#
# #    TO APPLY ON COLUMNS
# # s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
# table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Payment_MD", "Payment",
"Payment_Dt"),
#                      selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)
#
# scbr_x.pack(side=BOTTOM, fill=X)
# scbr_y.pack(side=RIGHT, fill=Y)
# scbr_x.config(command=table.xview)
# scbr_y.config(command=table.yview)
```

```
# table.heading("SN_No.", text="Sn No.", anchor=CENTER)
# table.heading("Payment_MD", text="Payment Mode", anchor=CENTER)
# table.heading("Payment", text="Payment", anchor=CENTER)
# table.heading("Payment_Dt", text="Payment Date", anchor=CENTER)
# table.pack(fill=BOTH, expand=1)
#
#
# table["show"] = "headings"
# table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
# table.column("Payment_MD", width=170, anchor=CENTER, minwidth=150)
# table.column("Payment", width=100, anchor=CENTER, minwidth=70)
# table.column("Payment_Dt", width=140, anchor=CENTER, minwidth=120)

pygame.mixer.init()
pygame.mixer.music.load("./Voices/Room Reservation.mp3")
pygame.mixer.music.play()
root.mainloop()
```




```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
from tkcalendar import DateEntry
from datetime import date
import pandas as pd
import mysql.connector as connector
import os

import pygame
height = 730
width = 1200
```

```
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))

Label(text="Guest Check In",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=650,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Guest Information", font=('Times New Roman', 30,
"bold")).place(x=10, y=10)
customtkinter.CTkLabel(master=frame, text="Payment Information", font=('Times New Roman', 30,
"bold")).place(x=500, y=10)
def Gstry():
    os.system("python Gs_Entry.py")
    Gs = pd.read_csv("hi.csv", index_col=[0],keep_default_na=False)
    GsID.set(Gs.hii[1])
    GsNm.set(Gs.hii[2])
    Gsder.set(Gs.hii[3])
    Gsgion.set(Gs.hii[4])
    GsAddress.set(Gs.hii[5])
    GsCity.set(Gs.hii[6])
    GsCntry.set(Gs.hii[7])
    GsCntNO.set(Gs.hii[8])
    GsIDType.set(Gs.hii[9])
    GsIDNo.set(Gs.hii[10])
    GsEmailIDNO.set(Gs.hii[11])
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open(Gs.hii[12]).resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
tkinter.Button(frame, image=Guest_Entry,compound=CENTER,command=Gstry, fg="Black", width=50,
activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", anchor=W,
borderwidth=5, cursor="hand2").place(x=210, y=52)
CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
Imglbl=Label(frame,image=CstmrImg)
Imglbl.place(x=365, y=110)
def reset():
    GsID.set("")
    GsNm.set("")
    Gsder.set("")
    Gsgion.set("")
```

```
    GsAddress.set("")
    GsCity.set("")
    GsCntry.set("")
    GsCntNO.set("")
    GsIDType.set("")
    GsIDNo.set("")
    GsEmailIDNO.set("")
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=reset,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=350, y=52)
def rr():
    Pymnt.set(0)
    mydata.set("Cash")
    cal.set_date(date.today())
    cal1.set_date(date.today())
    cal2.set_date(date.today())
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=rr, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=50)
sn=1
# def Add():
#     global sn
#     if int(Pymnt.get())<=int(file.RoomDetails[3]):
#         table.insert("",END, values=(sn, mydata.get(), Pymnt.get(), cal.get_date().strftime("%d-%m-%Y")))
#         sn+=1
#     else:
#         print("Line Number 82  Import Message Box")
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Add",command=Add,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=850, y=100)
# def rm():
#     # print(table.selection())
#     try:
#         table.delete(table.selection())
#     except :
#         pass
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=rm, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Remove", bg="#a8701d",
anchor=W,font=('Century Gothic', 16, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=160)
# def upp():
#     j=table.item(table.selection())['values']
#     print(j)
#     cal.set_date(j[3])
#     Pymnt.set(j[2])
#     mydata.set(j[1])
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=upp, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Update", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=220)
def CHk():
```

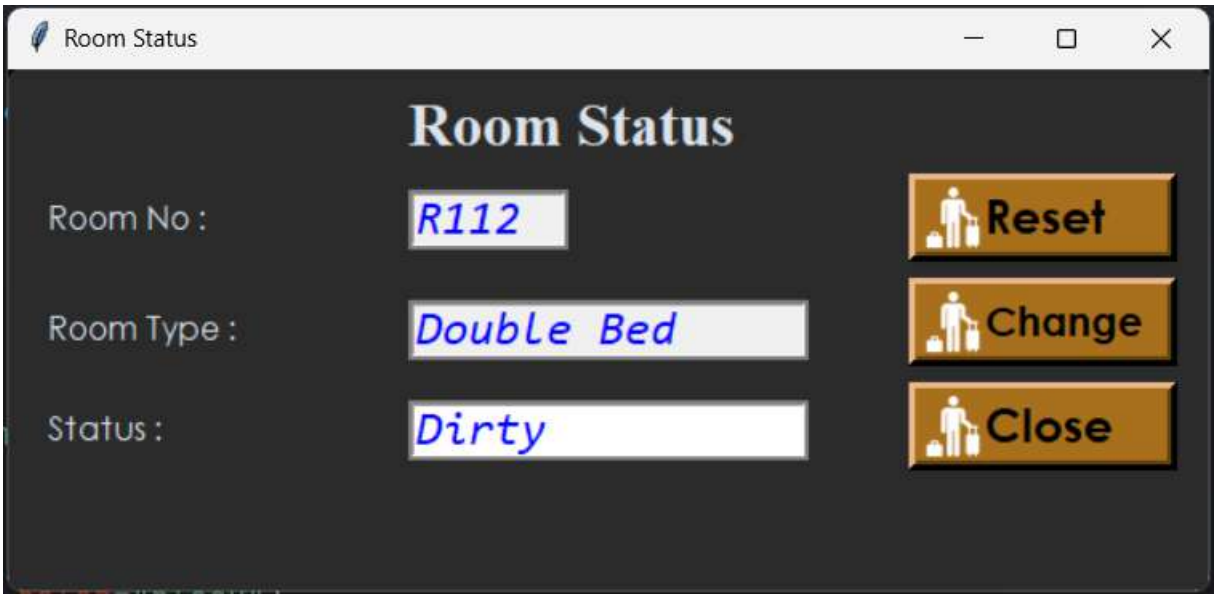
```
   if Pymnt.get() <= rm_price.get():
      if GsID.get() == "" or  GsNm.get() == "":
         err = messagebox.showerror("Fill Details", "Select The Customer")
         if err == "ok":
            pass
      else:
         Cnf = messagebox.askyesno("Check In", "Are You Sure You Want To Check In")
         if Cnf:
            con = connector.connect(host='localhost',
                                    port='3306',
                                    user='root',
                                    password='Password',
                                    database='Hotel Management Software')
            cur = con.cursor()
            c = cal1.get_date().strftime("%Y-%m-%d")
            c1 = cal2.get_date().strftime("%Y-%m-%d")
            c2 = cal.get_date().strftime("%Y-%m-%d")
            query = f"insert into `Check In Details` values
('{GsID.get()}','{GsNm.get()}','{Rm.get()}','{Rm_Type.get()}','{c}','{c1}','{rm_price.get()}','{Pymnt.get()}','{
mydata.get()}','{c2}','Active','Direct','{Rm_Prc.get()}');"
            # try:
            cur.execute(query)
            # print(query)
            # except Exception as e:
            #     print(e)
            query = f"Delete from `Room Status` where `Room No.`='{Rm.get()}';"
            cur.execute(query)
            # print(query)
            query = f"Update `customer details` set `Status`='Active' where `Guest ID`='{GsID.get()}';"
            cur.execute(query)
            # print(query)
            con.commit()
            GsID.set("")
            GsNm.set("")
            Gsder.set("")
            Gsgion.set("")
            GsAddress.set("")
            GsCity.set("")
            GsCntry.set("")
            GsCntNO.set("")
            GsIDType.set("")
            GsIDNo.set("")
            root.destroy()
   else :
      messagebox.showinfo("Check In Payment", "Advance Payment Must Be Smaller Than Or Equal To
Total Payment")
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=CHk, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Check In", bg="#a8701d",
anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=100)
def cls():
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=850, y=150)
```

```
def idd():
   customtkinter.CTkLabel(master=frame,text="Guest Id :",font=('Century Gothic',16)).place(x=20, y=60)
   customtkinter.CTkLabel(master=frame,text="Guest Name :",font=('Century Gothic',16)).place(x=20,
y=105)
   customtkinter.CTkLabel(master=frame,text="Gender :",font=('Century Gothic',16)).place(x=20, y=150)
   customtkinter.CTkLabel(master=frame,text="Religion :",font=('Century Gothic',16)).place(x=20, y=195)
   customtkinter.CTkLabel(master=frame,text="Address :",font=('Century Gothic',16)).place(x=20, y=240)
   customtkinter.CTkLabel(master=frame,text="City :",font=('Century Gothic',16)).place(x=20, y=285)
   customtkinter.CTkLabel(master=frame,text="Country :",font=('Century Gothic',16)).place(x=20, y=330)
   customtkinter.CTkLabel(master=frame,text="Contact No :",font=('Century Gothic',16)).place(x=20,
y=375)
   customtkinter.CTkLabel(master=frame,text="ID TYPE :",font=('Century Gothic',16)).place(x=20,
y=420)
   customtkinter.CTkLabel(master=frame,text="ID NUMBER :",font=('Century Gothic',16)).place(x=20,
y=465)
   customtkinter.CTkLabel(master=frame,text="Email ID :",font=('Century Gothic',16)).place(x=20,
y=510)

   #-------------------------------------------------------------------------------------------
   customtkinter.CTkLabel(master=frame, text="Room No :", font=('Century Gothic', 16)).place(x=510,
y=60)
   customtkinter.CTkLabel(master=frame, text="Room Type :", font=('Century Gothic', 16)).place(x=510,
y=110)
   customtkinter.CTkLabel(master=frame, text="Room Price :", font=('Century Gothic', 16)).place(x=510,
y=160)
   customtkinter.CTkLabel(master=frame, text="Date In :", font=('Century Gothic', 16)).place(x=510,
y=180+30)
   customtkinter.CTkLabel(master=frame, text="Date Out :", font=('Century Gothic', 16)).place(x=510,
y=220+40)
   customtkinter.CTkLabel(master=frame, text="Number Of Days :", font=('Century Gothic',
16)).place(x=510, y=260+50)
   customtkinter.CTkLabel(master=frame, text="Total Amount :", font=('Century Gothic',
16)).place(x=510, y=300+60)
   customtkinter.CTkLabel(master=frame, text="Advance Payment :", font=('Century Gothic',
16)).place(x=510,y=340+70)
   customtkinter.CTkLabel(master=frame, text="Payment Mode :", font=('Century Gothic',
16)).place(x=510, y=380+80)
   customtkinter.CTkLabel(master=frame, text="Payment Date :", font=('Century Gothic',
16)).place(x=510, y=380+40+90)
idd()
GsID=StringVar()
GsNm=StringVar()
Gsder=StringVar()
Gsgion=StringVar()
GsAddress=StringVar()
GsCity=StringVar()
GsCntry=StringVar()
GsCntNO=StringVar()
GsIDType=StringVar()
GsIDNo=StringVar()
GsEmailIDNO=StringVar()
GsID.set("")
GsNm.set("")
Gsder.set("")
```

```
Gsgion.set("")
GsAddress.set("")
GsCity.set("")
GsCntry.set("")
GsCntNO.set("")
GsIDType.set("")
GsIDNo.set("")
GsEmailIDNO.set("")
Entry(frame,highlightthickness=2,textvariable=GsID,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=60,width=50,height=30)
Entry(frame,highlightthickness=2,textvariable=GsNm,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=105,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=Gsder,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=150,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=Gsgion,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=195,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsAddress,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=240,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCity,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=285,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCntry,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=330,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsCntNO,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=375,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsIDType,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=420,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsIDNo,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=465,width=200,height=30)
Entry(frame,highlightthickness=2,textvariable=GsEmailIDNO,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=140,y=510,width=200,height=30)

# file1 = open("readme.txt", "r")
file = pd.read_csv('RMNO_RMPRICE.csv')
Rm=StringVar()
Rm.set(file.RoomDetails[1])
Rm_Type=StringVar()
Rm_Type.set(file.RoomDetails[2])
Rm_Prc=StringVar()
Rm_Prc.set(file.RoomDetails[3])
Days=StringVar()
rm_price=IntVar()
Pymnt=IntVar()
Entry(frame,textvariable=Rm,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=60,width=80,height=30)
Entry(frame,textvariable=Rm_Type,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 15 italic",state="disabled").place(x=690,y=60+40+10,width=135,height=30)
Entry(frame,textvariable=Rm_Prc,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=100+40+20,width=80,height=30)
Entry(frame,highlightthickness=2,textvariable=Days,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="disabled").place(x=690,y=220+40+50,width=50,height=30)
Entry(frame,highlightthickness=2,state="disabled",textvariable=rm_price,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic").place(x=690,y=260+40+60,width=135,height=30)
en=Entry(frame,highlightthickness=2,textvariable=Pymnt,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
```

```
en.place(x=690,y=300+40+70,width=135,height=30)
cal1 = DateEntry(frame,selectmode="day", font=("Cambria", 13, "italic"),foreground="blue",
width=10+2)
cal1.place(x=690, y=140+40+30)
cal2 = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue",
width=10+2)
cal2.place(x=690, y=180+40+40)
cal = DateEntry(frame, selectmode="day", font=("Cambria", 13, "italic"),foreground="blue", width=10+2)
cal.place(x=690, y=380+40+90)
def cal2_calevent_create(_):
   if (cal2.get_date() - cal1.get_date()).days >=0:
      Days.set((cal2.get_date() - cal1.get_date()).days)
      rm_price.set(value=int(file.RoomDetails[3])*(cal2.get_date()-cal1.get_date()).days)
      if rm_price.get() == 0:
         rm_price.set(int(file.RoomDetails[3]))
   else :
      messagebox.showinfo("Check Out Date", "Checkout Date Must Greater Than Check In Date")
   # print(int(file.RoomDetails[1])*(cal2.get_date()-cal1.get_date()).days)
cal2.bind("<<DateEntrySelected>>", cal2_calevent_create)
Days.set((cal2.get_date()-cal1.get_date()).days)
   #----------------------------------     SLIDER     ----------------------------------------
# def sliderevent(value):
#     Person.set(value=int(value))
#
customtkinter.CTkSlider(frame,from_=0,to=10,number_of_steps=10,width=80,command=sliderevent).place
(x=750,y=220)
   #-------------------------------------------------------------------------------------------
mydata = ttk.Combobox(frame, foreground="black", justify=LEFT, font="Calibri 13", width=10,
state='readonly',background="grey", height=10)
mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
# for j in range(1, 501):
#     l.append(f"API {j}")
#
# mydata["value"] = l
mydata.set("Cash")
mydata.place(x=690, y=340+40+80)
#-------------------------------------------------------------------------------------------
# frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
# # frm1.place(x=500, y=420, width=480, height=200)
# scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
# scbr_y = Scrollbar(frm1, orient=VERTICAL)
#
s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista
#
# #     FOR INSERT VALUES
#
s.configure(".", font=("consolas", 14, "italic"), foreground="blue")
#
# #    TO APPLY ON WHOLE TREEVIEW
#
s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
```

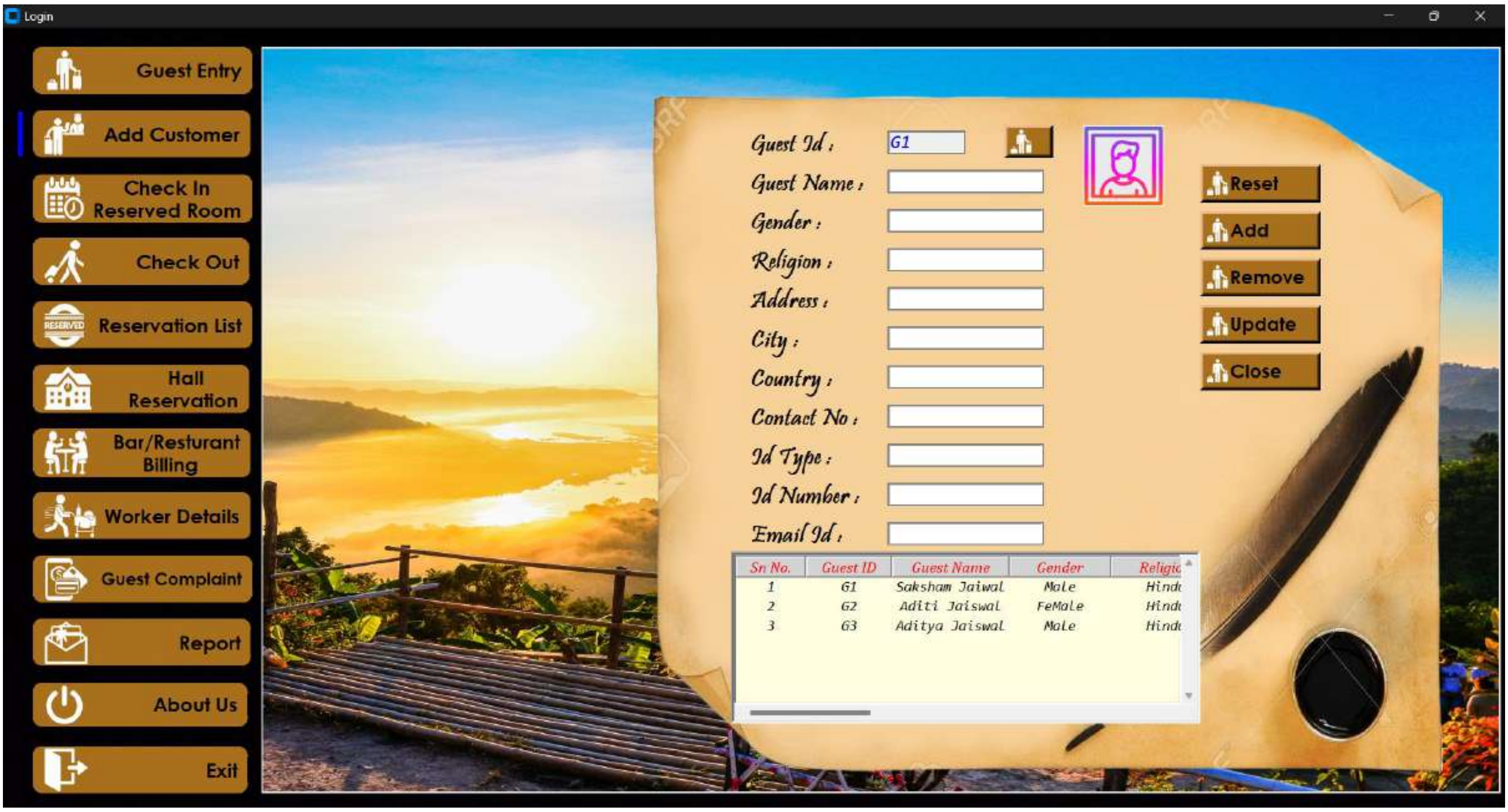
```
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light grey")
#
# #    TO APPLY ON COLUMNS
# # s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light grey")
# table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Payment_MD", "Payment", "Payment_Dt"),
#                      selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)
#
# scbr_x.pack(side=BOTTOM, fill=X)
# scbr_y.pack(side=RIGHT, fill=Y)
# scbr_x.config(command=table.xview)
# scbr_y.config(command=table.yview)
# table.heading("SN_No.", text="Sn No.", anchor=CENTER)
# table.heading("Payment_MD", text="Payment Mode", anchor=CENTER)
# table.heading("Payment", text="Payment", anchor=CENTER)
# table.heading("Payment_Dt", text="Payment Date", anchor=CENTER)
# table.pack(fill=BOTH, expand=1)
#
#
# table["show"] = "headings"
# table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
# table.column("Payment_MD", width=170, anchor=CENTER, minwidth=150)
# table.column("Payment", width=100, anchor=CENTER, minwidth=70)
# table.column("Payment_Dt", width=140, anchor=CENTER, minwidth=120)

pygame.mixer.init()
pygame.mixer.music.load("./Voices/Check In.mp3")
pygame.mixer.music.play()

root.mainloop()
```




```
try:
    import tkinter
    import customtkinter
    from tkinter import ttk
    from tkinter import *
    from tkinter import messagebox
    from PIL import ImageTk, Image
```

```
from tkcalendar import DateEntry
from datetime import date
import pandas as pd
import mysql.connector as connector
import os

height = 600
width = 1000

root = Tk()
root.maxsize("600", "260")
root.minsize("600", "260")
root.geometry('{}x{}+{}+{}'.format(600, 260, 600, 250))
root.title("Room Status")
con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Hotel Management Software')
cur = con.cursor()
Gs = pd.read_csv("RM_Status.csv", index_col=[0])
frame = customtkinter.CTkFrame(master=root, bg_color="black")
frame.pack(fill=BOTH, expand=1)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30, 30)))

customtkinter.CTkLabel(master=frame, text="Room Status", font=('Times New Roman', 30,
"bold")).place(x=200, y=10)


def reset():
    RmStatus.set(Gs.RoomStatus[3])


tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",
            activebackground="#a8701d", height=30, text="Reset", command=reset, bg="#a8701d",
anchor=W,
            font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=450, y=52)


def Cng():
    con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Hotel Management Software')
    cur = con.cursor()
    query = f"update `Room Status` set `Status`= '{RmStatus.get()}' where `Room No.` ='{RmNo.get()}';"
    cur.execute(query)
    con.commit()
    root.destroy()
```

```
    tkinter.Button(frame, image=Guest_Entry, compound=LEFT, command=Cng, fg="Black", width=120,
                   activeforeground="black", activebackground="#a8701d", height=30, text="Change",
bg="#a8701d",
                   anchor=W, font=('Century Gothic', 15, "bold"), borderwidth=5, cursor="hand2").place(x=450,
y=52 + 52)


    def cls():
        root.destroy()


    tkinter.Button(frame, image=Guest_Entry, compound=LEFT, command=cls, fg="Black", width=120,
                   activeforeground="black", activebackground="#a8701d", height=30, text="Close",
bg="#a8701d",
                   anchor=W, font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=450,
                                                                                                       y=52 + 52 + 52)


    def idd():
        customtkinter.CTkLabel(master=frame, text="Room No :", font=('Century Gothic', 16)).place(x=20,
y=60)
        customtkinter.CTkLabel(master=frame, text="Room Type :", font=('Century Gothic', 16)).place(x=20,
y=105 + 10)
        customtkinter.CTkLabel(master=frame, text="Status :", font=('Century Gothic', 16)).place(x=20,
y=150 + 10 + 5)


    idd()
    RmNo = StringVar()
    RmTyp = StringVar()
    RmStatus = StringVar()
    RmNo.set(Gs.RoomStatus[1])
    RmTyp.set(Gs.RoomStatus[2])
    RmStatus.set(Gs.RoomStatus[3])
    Entry(frame, highlightthickness=2, textvariable=RmNo, highlightbackground="grey",
highlightcolor="black", fg="blue",
          font="consolas 17 italic", state="readonly").place(x=200, y=60, width=80, height=30)
    Entry(frame, highlightthickness=2, textvariable=RmTyp, highlightbackground="grey",
highlightcolor="black",
          fg="blue", font="consolas 17 italic", state="readonly").place(x=200, y=105 + 10, width=200,
height=30)
    Entry(frame, highlightthickness=2, textvariable=RmStatus, highlightbackground="grey",
highlightcolor="black",
          fg="blue", font="consolas 17 italic").place(x=200, y=150 + 15, width=200, height=30)
    root.mainloop()
except Exception as e:
    messagebox.showerror("Error",e)
```




```
#--------------------------------------- ADD MEMBER -----------------------------------------------------------------------
------------------

can_widget2 = Canvas(l1,width=1550,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
AddMmBg = ImageTk.PhotoImage(Image.open("./assets/yellow-tourism-sun-life-beautiful-mountain-
1577681-pxhere.com.jpg").resize((1570,955)))
can_widget2.create_image(0,0,anchor=NW,image=AddMmBg)
AddMmImg = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231109_003231.png").resize((1020,890)))
can_widget2.create_image(1000,495,image=AddMmImg)
# can_widget2.place(x=330, y=25)
# customtkinter.CTkLabel(master=can_widget2, text="Add Customer", font=('Times New Roman', 30,
"bold"),fg_color="black",bg_color="black").place(x=500, y=10)
def Ad_Mem_idd():
   customtkinter.CTkLabel(master=can_widget2,text="Guest Id
```

```
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=60+25)
   customtkinter.CTkLabel(master=can_widget2,text="Guest Name
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=100+25)
   customtkinter.CTkLabel(master=can_widget2,text="Gender
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=140+25)
   customtkinter.CTkLabel(master=can_widget2,text="Religion
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=180+25)
   customtkinter.CTkLabel(master=can_widget2,text="Address
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=220+25)
   customtkinter.CTkLabel(master=can_widget2,text="City
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=260+25)
   customtkinter.CTkLabel(master=can_widget2,text="Country
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=300+25)
   customtkinter.CTkLabel(master=can_widget2,text="Contact No
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=340+25)
   customtkinter.CTkLabel(master=can_widget2,text="Id Type
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=380+25)
   customtkinter.CTkLabel(master=can_widget2,text="Id Number
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=420+25)
   customtkinter.CTkLabel(master=can_widget2,text="Email Id
:",text_color="Black",fg_color="#f7d19a",bg_color="#f7d19a",font=('Pristina',25,"bold")).place(x=500,
y=460+25)
Ad_Mem_idd()
#-------------------------------------------------------------------------------------------------------------
-------
Gs_ENTRY_ID=StringVar()
Gs_ENTRY_Nm=StringVar()
Gs_ENTRY_der=StringVar()
Gs_ENTRY_gion=StringVar()
Gs_ENTRY_Address=StringVar()
Gs_ENTRY_City=StringVar()
Gs_ENTRY_Cntry=StringVar()
Gs_ENTRY_CntNO=StringVar()
Gs_ENTRY_IDType=StringVar()
Gs_ENTRY_IDNo=StringVar()
Gs_ENTRY_EmailID=StringVar()
Gs_ENTRY_ID.set("")
Gs_ENTRY_Nm.set("")
Gs_ENTRY_der.set("")
Gs_ENTRY_gion.set("")
Gs_ENTRY_Address.set("")
Gs_ENTRY_City.set("")
Gs_ENTRY_Cntry.set("")
Gs_ENTRY_CntNO.set("")
```

```
Gs_ENTRY_IDType.set("")
Gs_ENTRY_IDNo.set("")
Gs_ENTRY_EmailID.set("")
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_ID,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic",state="readonly").place(x=900-
100,y=80+27,width=100,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_Nm,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=130+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_der,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=180+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_gion,highlightbackground="grey",highli
ghtcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=230+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_Address,highlightbackground="grey",hi
ghlightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=280+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_City,highlightbackground="grey",highli
ghtcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=330+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_Cntry,highlightbackground="grey",highl
ightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=380+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_CntNO,highlightbackground="grey",hig
hlightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=430+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_IDType,highlightbackground="grey",hig
hlightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=480+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_IDNo,highlightbackground="grey",high
lightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=530+27,width=200,height=30)
Entry(can_widget2,highlightthickness=2,textvariable=Gs_ENTRY_EmailID,highlightbackground="grey",hi
ghlightcolor="red",fg="blue",font="consolas 17 italic").place(x=900-100,y=580+27,width=200,height=30)

CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
Imglbl=Label(can_widget2,image=CstmrImg)
Imglbl.place(x=1050, y=100)
filename=0
def ImgFunc():
   global CstmrImg
   global filename
   filename = filedialog.askopenfilename(initialdir="/", title="Select A File",filetypes=(("JPG files",
"*.jpg"), ("All Files", "*.*")))
   CstmrImg = ImageTk.PhotoImage(Image.open(filename).resize((100, 100)))
   Imglbl.configure(image=CstmrImg)
tkinter.Button(can_widget2, image=Guest_Entry,compound=CENTER,command=ImgFunc, fg="Black",
width=50, activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", anchor=W,
borderwidth=5, cursor="hand2").place(x=950, y=100)
def GST_INFOrr():
   Gs_entry_read = pd.read_csv("Gs_ID.csv", index_col=[0])
   Gs_ENTRY_ID.set(value=Gs_entry_read.Gs_Entry[1])
   Gs_ENTRY_Nm.set(value="")
   Gs_ENTRY_der.set(value="")
   Gs_ENTRY_gion.set(value="")
   Gs_ENTRY_Address.set(value="")
   Gs_ENTRY_City.set(value="")
   Gs_ENTRY_Cntry.set(value="")
   Gs_ENTRY_CntNO.set(value="")
   Gs_ENTRY_IDType.set(value="")
   Gs_ENTRY_IDNo.set(value="")
   Gs_ENTRY_EmailID.set(value="")
```

```
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
    # Pymnt.set("")
    # for item in table.get_children():
    #     table.delete(item)
    # cal.set_date(date.today())
tkinter.Button(can_widget2, image=Guest_Entry, compound=LEFT,command=GST_INFOrr, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Reset",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=100+50)

Gs_entry_read=pd.read_csv("Gs_ID.csv", index_col=[0])
Gs_ENTRY_ID.set(value=Gs_entry_read.Gs_Entry[1])

def GST_INFOAdd():
    global filename
    if messagebox.askyesno("Add Customer", "Are You Sure You Want To Add Customer"):
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Hotel Management Software')
        cur = con.cursor()
        query = 'desc `customer details`'
        cur.execute(query)
        if len(str(filename)) > int(cur.fetchall()[13][1][8:-1]):
            query = f'alter table `customer details` modify column Image varchar({len(str(filename))});'
            cur.execute(query)
            con.commit()
        if filename ==0:
            filename="passportsizephoto.webp"
        Gs_val = pd.read_csv("Gs_ID.csv", index_col=[0])
        query = f"insert into `Customer Details`
values('{Gs_val.Gs_Entry[0]}','{Gs_ENTRY_ID.get()}','{Gs_ENTRY_Nm.get()}','{Gs_ENTRY_der.get()}','
{Gs_ENTRY_gion.get()}','{Gs_ENTRY_Address.get()}','{Gs_ENTRY_City.get()}','{Gs_ENTRY_Cntry.get(
)}','{Gs_ENTRY_CntNO.get()}','{Gs_ENTRY_IDType.get()}','{Gs_ENTRY_IDNo.get()}','Inactive','{Gs_E
NTRY_EmailID.get()}','{filename}');"
        # print(query)
        try:
            cur.execute(query)
            Gs_entry_read.Gs_Entry[0] = str(int(Gs_entry_read.Gs_Entry[0]) + 1)
            Gs_entry_read.Gs_Entry[1] = "G" + str(int(Gs_entry_read.Gs_Entry[1][1:]) + 1)
            Gs_entry_read.to_csv("Gs_ID.csv")
        except Exception as e:
            print(e)
        con.commit()
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Hotel Management Software')
        cur = con.cursor()
        for item in Gs_Entry_Tabke.get_children():
```

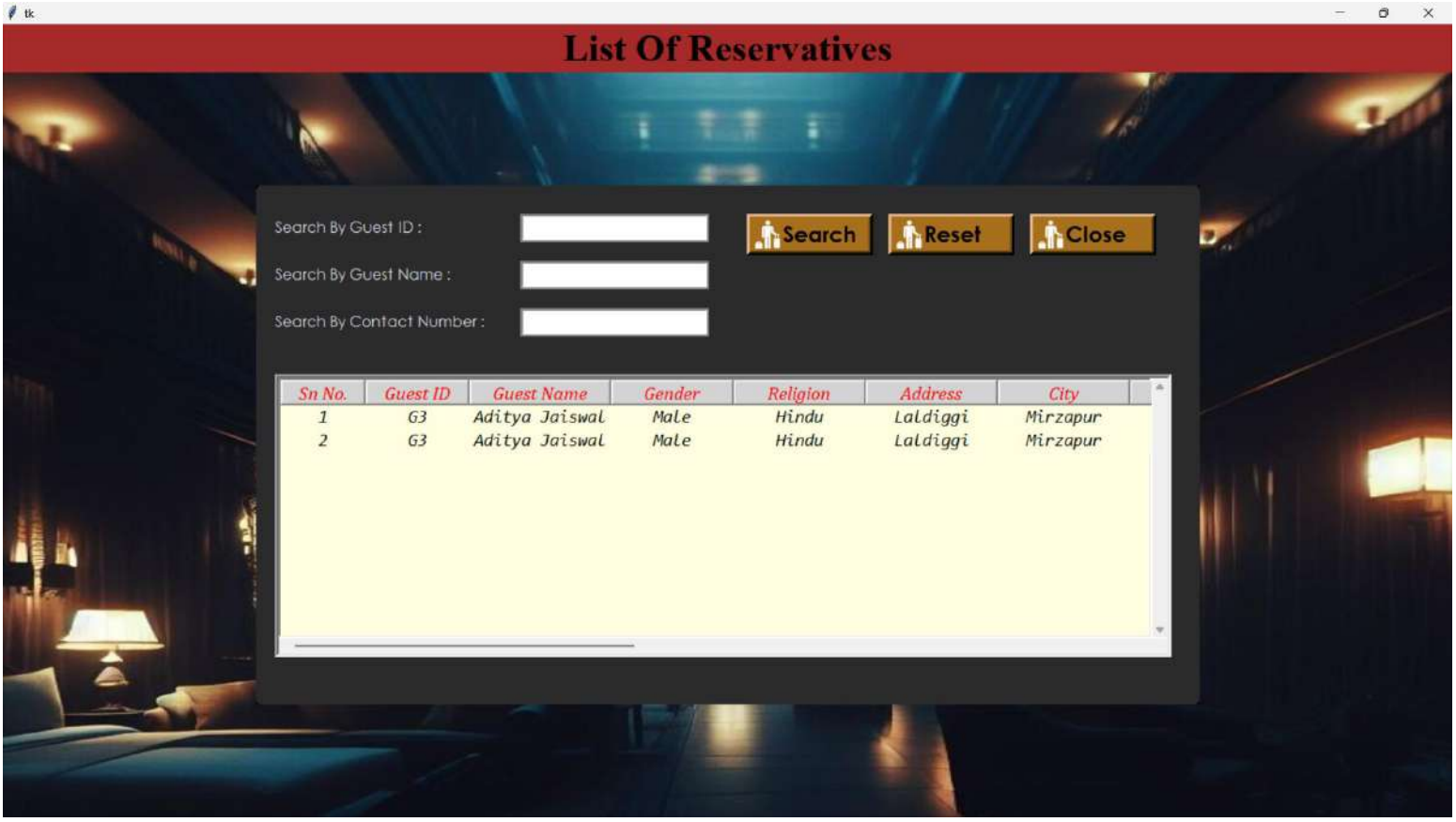
```
        Gs_Entry_Tabke.delete(item)
    query = "select * from `Customer Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        Gs_Entry_Tabke.insert("", END, values=(
            sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11],
row[12],
            row[13]))
        sn += 1
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
    Gs_val = pd.read_csv("Gs_ID.csv", index_col=[0])
    Gs_ENTRY_ID.set(value=Gs_val.Gs_Entry[1])
    # val3.close()
    Gs_ENTRY_Nm.set(value="")
    Gs_ENTRY_der.set(value="")
    Gs_ENTRY_gion.set(value="")
    Gs_ENTRY_Address.set(value="")
    Gs_ENTRY_City.set(value="")
    Gs_ENTRY_Cntry.set(value="")
    Gs_ENTRY_CntNO.set(value="")
    Gs_ENTRY_IDType.set(value="")
    Gs_ENTRY_IDNo.set(value="")
    Gs_ENTRY_EmailID.set(value="")
    # Gs_ENTRY_sn += 1
tkinter.Button(can_widget2, image=Guest_Entry, compound=LEFT, fg="Black", width=120+20,
activeforeground="black",activebackground="#a8701d", height=30+5,
text="Add",command=GST_INFOAdd, bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"),
borderwidth=5, cursor="hand2").place(x=1200, y=160+50)
def GST_INFOrm():
  if messagebox.askyesno("Remove Customer", "Are You Sure You Want To Remove Customer"):
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    rmve = Gs_Entry_Tabke.item(Gs_Entry_Tabke.selection())['values']
    query = f"DELETE FROM `Customer Details` WHERE (`Guest ID` = '{rmve[1]}');"
    try:
        cur.execute(query)
    except:
        pass
    con.commit()
    Gs_val = pd.read_csv("Gs_ID.csv", index_col=[0])
    Gs_ENTRY_ID.set(value=Gs_val.Gs_Entry[1])
    Gs_ENTRY_Nm.set(value="")
    Gs_ENTRY_der.set(value="")
    Gs_ENTRY_gion.set(value="")
    Gs_ENTRY_Address.set(value="")
    Gs_ENTRY_City.set(value="")
    Gs_ENTRY_Cntry.set(value="")
```

```
    Gs_ENTRY_CntNO.set(value="")
    Gs_ENTRY_IDType.set(value="")
    Gs_ENTRY_IDNo.set(value="")
    Gs_ENTRY_EmailID.set(value="")
    Gs_Entry_Tabke.delete(Gs_Entry_Tabke.selection()[0])
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    for item in Gs_Entry_Tabke.get_children():
        Gs_Entry_Tabke.delete(item)
    query = "select * from `Customer Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        Gs_Entry_Tabke.insert("", END, values=(
            sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11],
row[12],
            row[13]))
        sn += 1
tkinter.Button(can_widget2, image=Guest_Entry, compound=LEFT,command=GST_INFOrm, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Remove",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=220+50)
def GST_INFOupp():
    if messagebox.askyesno("Update Customer", "Are You Sure You Want To Update Customer Details"):
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Hotel Management Software')
        cur = con.cursor()
        query = f"UPDATE `Customer Details` SET `Guest Name` = '{Gs_ENTRY_Nm.get()}', `Gender` =
'{Gs_ENTRY_der.get()}', `Religion` = '{Gs_ENTRY_gion.get()}', `Address` = '{Gs_ENTRY_Address.get()}',
`City` = '{Gs_ENTRY_City.get()}', `Country` = '{Gs_ENTRY_Cntry.get()}', `Contact No.` =
'{Gs_ENTRY_CntNO.get()}', `ID Type` = '{Gs_ENTRY_IDType.get()}', `ID Number` =
'{Gs_ENTRY_IDNo.get()}',`Email ID` = '{Gs_ENTRY_EmailID.get()}',Image = '{filename}' WHERE
(`Guest ID` = '{Gs_ENTRY_ID.get()}');"
        # print(query)
        cur.execute(query)
        con.commit()
        for item in Gs_Entry_Tabke.get_children():
            Gs_Entry_Tabke.delete(item)
        sn = 1
        cur = con.cursor()
        query = "select * from `Customer Details`;"
        cur.execute(query)
        for row in cur.fetchall():
            Gs_Entry_Tabke.insert("", END, values=(
```

```
            sn, row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11],
row[12]))
            sn += 1
        Gs_val = pd.read_csv("Gs_ID.csv", index_col=[0])
        Gs_ENTRY_ID.set(value=Gs_val.Gs_Entry[1])
        Gs_ENTRY_Nm.set(value="")
        Gs_ENTRY_der.set(value="")
        Gs_ENTRY_gion.set(value="")
        Gs_ENTRY_Address.set(value="")
        Gs_ENTRY_City.set(value="")
        Gs_ENTRY_Cntry.set(value="")
        Gs_ENTRY_CntNO.set(value="")
        Gs_ENTRY_IDType.set(value="")
        Gs_ENTRY_IDNo.set(value="")
        Gs_ENTRY_EmailID.set(value="")
tkinter.Button(can_widget2, image=Guest_Entry, compound=LEFT,command=GST_INFOupp,
fg="Black", width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5,
text="Update", bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=280+50)
def GST_INFOcls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
tkinter.Button(can_widget2, image=Guest_Entry, compound=LEFT,command=GST_INFOcls, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=340+50)

#-------------------------------------------------------------------------------------------------------------
-------
s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
```

```
grey")

frm1 = Frame(can_widget2, relief=SUNKEN, borderwidth=4)
frm1.place(x=600, y=645, width=600, height=220)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Gs_Entry_Tabke = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Gs_ID", "Gs_Name",
"Gender","Religion","Address","City","Country","Contact_No","ID_Type","ID_No","Active","Email_
ID",'Img'),
                  selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=Gs_Entry_Tabke.xview)
scbr_y.config(command=Gs_Entry_Tabke.yview)
Gs_Entry_Tabke.heading("SN_No.", text="Sn No.", anchor=CENTER)
Gs_Entry_Tabke.heading("Gs_ID", text="Guest ID", anchor=CENTER)
Gs_Entry_Tabke.heading("Gs_Name", text="Guest Name", anchor=CENTER)
Gs_Entry_Tabke.heading("Gender", text="Gender", anchor=CENTER)
Gs_Entry_Tabke.heading("Religion", text="Religion", anchor=CENTER)
Gs_Entry_Tabke.heading("Address", text="Address", anchor=CENTER)
Gs_Entry_Tabke.heading("City", text="City", anchor=CENTER)
Gs_Entry_Tabke.heading("Country", text="Country", anchor=CENTER)
Gs_Entry_Tabke.heading("Contact_No", text="Contact Number", anchor=CENTER)
Gs_Entry_Tabke.heading("ID_Type", text="ID Type", anchor=CENTER)
Gs_Entry_Tabke.heading("ID_No", text="ID No", anchor=CENTER)
Gs_Entry_Tabke.heading("Active", text="Active", anchor=CENTER)
Gs_Entry_Tabke.heading("Email_ID", text="Email ID", anchor=CENTER)
Gs_Entry_Tabke.heading("Img", text="Image", anchor=CENTER)
Gs_Entry_Tabke.pack(fill=BOTH, expand=1)


Gs_Entry_Tabke["show"] = "headings"
Gs_Entry_Tabke.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
Gs_Entry_Tabke.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
Gs_Entry_Tabke.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
Gs_Entry_Tabke.column("Gender", width=130, anchor=CENTER, minwidth=130)
Gs_Entry_Tabke.column("Religion", width=140, anchor=CENTER, minwidth=140)
Gs_Entry_Tabke.column("Address", width=140, anchor=CENTER, minwidth=140)
Gs_Entry_Tabke.column("City", width=140, anchor=CENTER, minwidth=140)
Gs_Entry_Tabke.column("Country", width=140, anchor=CENTER, minwidth=140)
Gs_Entry_Tabke.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
Gs_Entry_Tabke.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
Gs_Entry_Tabke.column("ID_No", width=140, anchor=CENTER, minwidth=120)
Gs_Entry_Tabke.column("Active", width=140, anchor=CENTER, minwidth=120)
Gs_Entry_Tabke.column("Email_ID", width=300, anchor=CENTER, minwidth=120)
Gs_Entry_Tabke.column("Img", width=120, anchor=CENTER, minwidth=0)
def Gs_Entry_Tabke_select(_):
    j = Gs_Entry_Tabke.item(Gs_Entry_Tabke.selection())['values']
    Gs_ENTRY_ID.set(value=j[1])
    Gs_ENTRY_Nm.set(value=j[2])
```

```
    Gs_ENTRY_der.set(value=j[3])
    Gs_ENTRY_gion.set(value=j[4])
    Gs_ENTRY_Address.set(value=j[5])
    Gs_ENTRY_City.set(value=j[6])
    Gs_ENTRY_Cntry.set(value=j[7])
    Gs_ENTRY_CntNO.set(value=j[8])
    Gs_ENTRY_IDType.set(value=j[9])
    Gs_ENTRY_IDNo.set(value=j[10])
    Gs_ENTRY_EmailID.set(value=j[12])
    # print(j[13])
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open(j[13]).resize((100, 100)))
    Imglbl.configure(image=CstmrImg)
Gs_Entry_Tabke.bind('<<TreeviewSelect>>',Gs_Entry_Tabke_select)
```



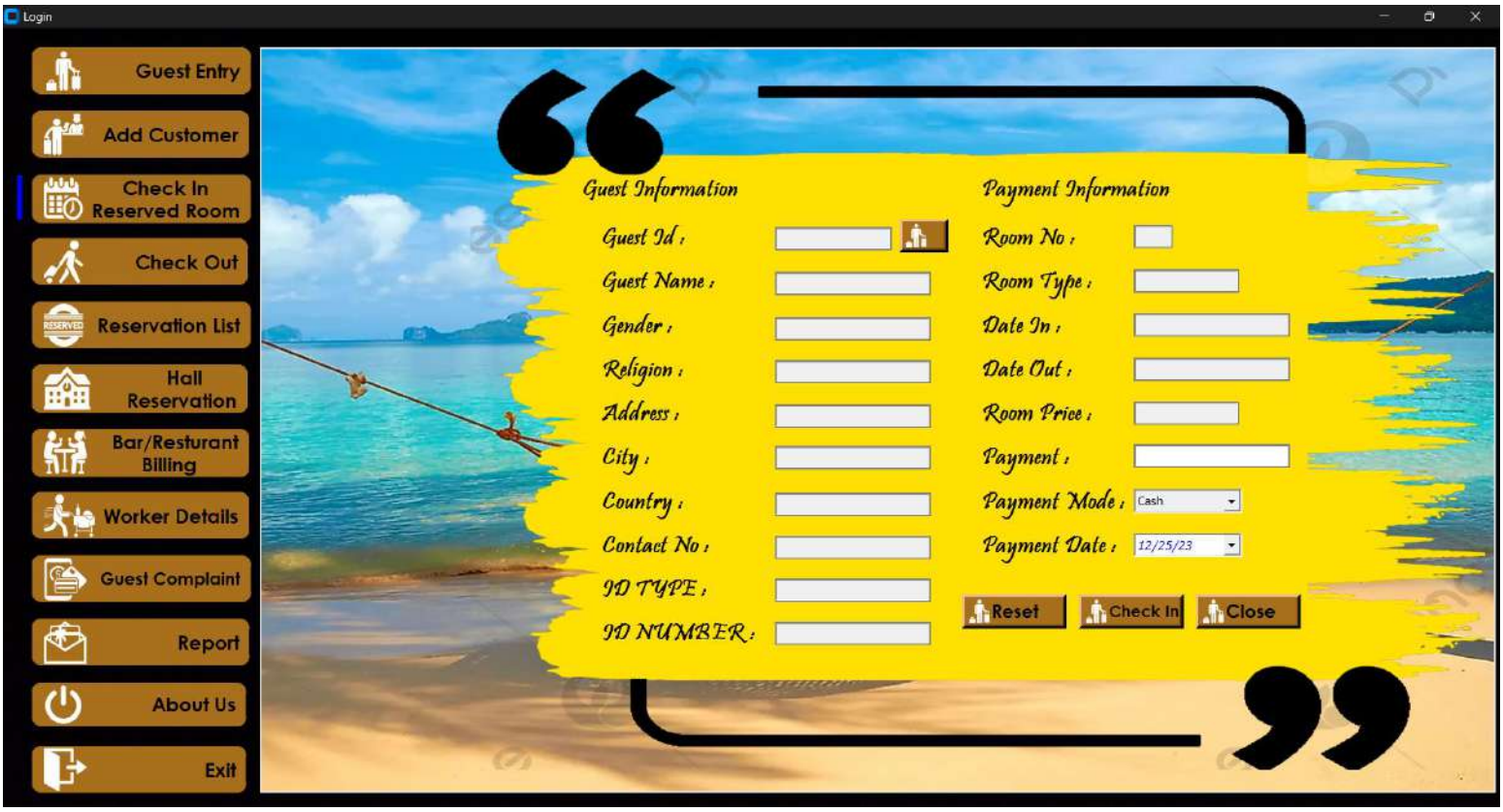

```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import pandas as pd
import mysql.connector as connector
from tkcalendar import DateEntry
from datetime import date
import os
height = 730
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))
```

```
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
Label(text="List Of Reservatives",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)

img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=550,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Search By Guest ID :", font=('Century Gothic',
16)).place(x=20, y=30)
customtkinter.CTkLabel(master=frame, text="Search By Guest Name :", font=('Century Gothic',
16)).place(x=20, y=80)
customtkinter.CTkLabel(master=frame, text="Search By Contact Number :", font=('Century Gothic',
16)).place(x=20, y=130)
E1Var=StringVar()
E1Var.set("")
E2Var=StringVar()
E2Var.set("")
E3Var=StringVar()
E3Var.set("")
E1=Entry(frame,highlightthickness=2,textvariable=E1Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E1.place(x=280,y=30,width=200,height=30)
E2=Entry(frame,highlightthickness=2,textvariable=E2Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E2.place(x=280,y=80,width=200,height=30)
E3=Entry(frame,highlightthickness=2,textvariable=E3Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E3.place(x=280,y=130,width=200,height=30)
def Scrh():
   if E1.get()==""and E2.get()=="":
      for item in table.get_children():
         table.delete(item)
      query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null and `Contact No.`='{E3.get()}';"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         table.insert("", END, values=(
         sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[13],
row[14],
         row[15], row[16], row[17], row[18]))
         sn += 1
   elif E2.get()==""and E3.get()=="":
```

```
        for item in table.get_children():
            table.delete(item)
        query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null  and `Guest ID`='{E1.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            table.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[13],
row[14],
            row[15], row[16], row[17], row[18]))
            sn += 1
    elif E1.get()==""and E3.get()=="":
        for item in table.get_children():
            table.delete(item)
        query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null;"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            table.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[13],
row[14],
            row[15], row[16], row[17], row[18]))
            sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Scrh, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Search", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=500+20, y=30)
def Rset():
    E1Var.set("")
    E2Var.set("")
    E3Var.set("")
    for item in table.get_children():
        table.delete(item)
    query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(
        sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[13], row[14],
row[15],
        row[16], row[17], row[18]))
        sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Rset, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650+20, y=30)
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Excel", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650, y=30)
def cls():
    fu = ["","","","","","","","","","","","","","","","",""]
    so = pd.Series(data=fu, name="hii")
    sep = pd.DataFrame(so)
```

```
   sep.to_csv("hi.csv")
   root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=800+20, y=30)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=20, y=200, width=950, height=300)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Gs_ID", "Gs_Name",
"Gender","Religion","Address","City","Country","Contact_No","ID_Type","ID_No","Room_No","Ro
om_Type","Date_In","Date_Out","Room_Charges","Payment"),
                     selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Gs_ID", text="Guest ID", anchor=CENTER)
table.heading("Gs_Name", text="Guest Name", anchor=CENTER)
table.heading("Gender", text="Gender", anchor=CENTER)
table.heading("Religion", text="Religion", anchor=CENTER)
table.heading("Address", text="Address", anchor=CENTER)
table.heading("City", text="City", anchor=CENTER)
table.heading("Country", text="Country", anchor=CENTER)
table.heading("Contact_No", text="Contact Number", anchor=CENTER)
table.heading("ID_Type", text="ID Type", anchor=CENTER)
table.heading("ID_No", text="ID No", anchor=CENTER)
table.heading("Room_No", text="Room No", anchor=CENTER)
table.heading("Room_Type", text="Room Type", anchor=CENTER)
table.heading("Date_In", text="Date In", anchor=CENTER)
table.heading("Date_Out", text="Date Out", anchor=CENTER)
table.heading("Room_Charges", text="Room Charges", anchor=CENTER)
table.heading("Payment", text="Payment", anchor=CENTER)
```

```
table.pack(fill=BOTH, expand=1)


table["show"] = "headings"
table.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
table.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
table.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
table.column("Gender", width=130, anchor=CENTER, minwidth=130)
table.column("Religion", width=140, anchor=CENTER, minwidth=140)
table.column("Address", width=140, anchor=CENTER, minwidth=140)
table.column("City", width=140, anchor=CENTER, minwidth=140)
table.column("Country", width=140, anchor=CENTER, minwidth=140)
table.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
table.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
table.column("ID_No", width=140, anchor=CENTER, minwidth=120)
table.column("Room_No", width=140, anchor=CENTER, minwidth=120)
table.column("Room_Type", width=140, anchor=CENTER, minwidth=120)
table.column("Date_In", width=140, anchor=CENTER, minwidth=120)
table.column("Date_Out", width=140, anchor=CENTER, minwidth=120)
table.column("Room_Charges", width=140, anchor=CENTER, minwidth=120)
table.column("Payment", width=140, anchor=CENTER, minwidth=120)
query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null;"
cur.execute(query)
sn=1
for row in cur.fetchall():
   table.insert("",END, values=(sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9],
row[10],row[15], row[16], row[17], row[18], row[19], row[20]))
   sn+=1
#
table.insert(parent="",index=0,values=(1,101,"Anjali","Female","Hinduism","Jodhpur","Rajasthan","Bh
arat","+919102020202","Passport",'1P85DF256'))
# table.insert(parent="",index=1,values=(2,102,"Mohan","Male","Hinduism","Lucknow","Uttar
Pradesh","Bharat","+919105050505","Aadhar Card",'12121212121'))
#
table.insert(parent="",index=2,values=(3,103,"Rakesh","Transgender","None","Kerala","Kerala","Bhara
t","+919103003302","Passport",'1P25RF56456'))
def table_select(_):
   fu=[]
   for i in table.selection():
      fu.append(table.item(i)['values'])
   so = pd.Series(data=fu[0], name="hii")
   sep = pd.DataFrame(so)
   sep.to_csv("hi.csv")
   # print(so)
   root.destroy()
table.bind('<<TreeviewSelect>>',table_select)

root.mainloop()
```




```
#---------------------------------------------------- Check In ----------------------------------------------------
can_widget5 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
ChkInBg= ImageTk.PhotoImage(Image.open("./assets/shutterstock-329818472.webp").resize((1585,955)))
can_widget5.create_image(0,0,anchor=NW,image=ChkInBg)
ChkInImg = ImageTk.PhotoImage(Image.open("./assets/pngtree-speech-bubble-quote-text-box-origami-banner-png-image_7947840.png").resize((1300,900)))
can_widget5.create_image(920,475,image=ChkInImg)
# can_widget5.place(x=330, y=25)
# Label(can_widget5,text="Reserved Customer Check In",background="brown",font=('Times New Roman',30,"bold")).pack(anchor=N,fill=X)
# customtkinter.CTkLabel(master=can_widget5, text="Reserved Customer Check In", font=('Times New Roman', 30, "bold"),fg_color="black",bg_color="black").place(x=500, y=10)
#
# Rsvrd_Gst_Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))
#
customtkinter.CTkLabel(master=can_widget5, text="Guest Information",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=330, y=10+50+70)
customtkinter.CTkLabel(master=can_widget5, text="Payment Information",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=740, y=10+50+70)

def Rsvrd_Gst_chkin():
    os.system("python Resvrd_List_checkin.py")
    Gs = pd.read_csv("hi.csv", index_col=[0])
    Rsvrd_Gst_GsID.set(Gs.hii[1])
    Rsvrd_Gst_GsNm.set(Gs.hii[2])
    Rsvrd_Gst_Gsder.set(Gs.hii[3])
    Rsvrd_Gst_Gsgion.set(Gs.hii[4])
    Rsvrd_Gst_GsAddress.set(Gs.hii[5])
    Rsvrd_Gst_GsCity.set(Gs.hii[6])
```

```
    Rsvrd_Gst_GsCntry.set(Gs.hii[7])
    Rsvrd_Gst_GsCntNO.set(Gs.hii[8])
    Rsvrd_Gst_GsIDType.set(Gs.hii[9])
    Rsvrd_Gst_GsIDNo.set(Gs.hii[10])
    Rsvrd_Gst_Rm.set(Gs.hii[11])
    Rsvrd_Gst_Rm_Type.set(Gs.hii[12])
    din.set(Gs.hii[13])
    dout.set(Gs.hii[14])
    Rsvrd_Gst_rm_price.set(Gs.hii[15])
# ------------------------------------------------------ BUTTON ------------------------------------------------------
tkinter.Button(can_widget5, image=Guest_Entry,compound=CENTER,command=Rsvrd_Gst_chkin,
fg="Black", width=50, activeforeground="black",activebackground="#a8701d", height=30,
bg="#a8701d", anchor=W, borderwidth=5, cursor="hand2").place(x=820, y=220)
def Rsvrd_Gst_rr():
    Rsvrd_Gst_GsID.set("")
    Rsvrd_Gst_GsNm.set("")
    Rsvrd_Gst_Gsder.set("")
    Rsvrd_Gst_Gsgion.set("")
    Rsvrd_Gst_GsAddress.set("")
    Rsvrd_Gst_GsCity.set("")
    Rsvrd_Gst_GsCntry.set("")
    Rsvrd_Gst_GsCntNO.set("")
    Rsvrd_Gst_GsIDType.set("")
    Rsvrd_Gst_GsIDNo.set("")
    Rsvrd_Gst_Rm.set("")
    Rsvrd_Gst_Rm_Type.set("")
    din.set("")
    dout.set("")
    Rsvrd_Gst_rm_price.set("")
    Rsvrd_Gst_Pymnt.set("")
    Rsvrd_Gst_mydata.set("Cash")
    Rsvrd_Gst_cal.set_date(date.today())
tkinter.Button(can_widget5, image=Guest_Entry, compound=LEFT,command=Rsvrd_Gst_rr, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Reset",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=900, y=700)
Rsvrd_Gst_sn=1

def Rsvrd_Gst_CHk():
    if messagebox.askyesno("Check In", "Are You Sure You Want To Check In Reserved Room"):
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Hotel Management Software')
        cur = con.cursor()
        query = f"insert into `Check In Details` values
('{Rsvrd_Gst_GsID.get()}','{Rsvrd_Gst_GsNm.get()}','{Rsvrd_Gst_Rm.get()}','{Rsvrd_Gst_Rm_Type.get()}'
,'{din.get()}','{dout.get()}','{Rsvrd_Gst_rm_price.get()}','{Rsvrd_Gst_Pymnt.get()}','{Rsvrd_Gst_mydata.get(
)}','{Rsvrd_Gst_cal.get_date().strftime('%Y-%m-
%d')}','Active','Reservation','{Rsvrd_Gst_rm_price.get()}');"
        # print(query)
        cur.execute(query)
        con.commit()
```

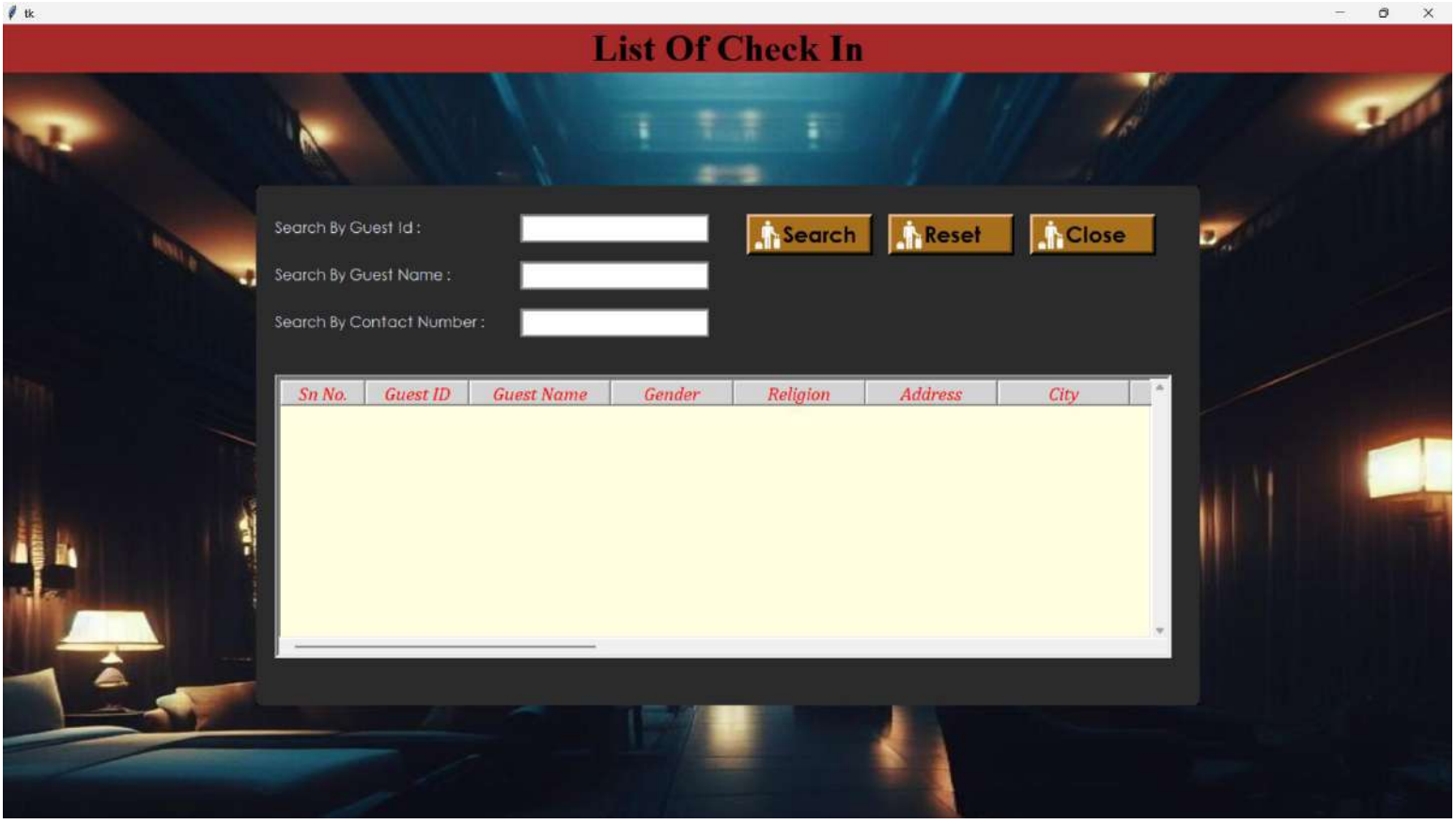
```
        Rsvrd_Gst_GsID.set("")
        Rsvrd_Gst_GsNm.set("")
        Rsvrd_Gst_Gsder.set("")
        Rsvrd_Gst_Gsgion.set("")
        Rsvrd_Gst_GsAddress.set("")
        Rsvrd_Gst_GsCity.set("")
        Rsvrd_Gst_GsCntry.set("")
        Rsvrd_Gst_GsCntNO.set("")
        Rsvrd_Gst_GsIDType.set("")
        Rsvrd_Gst_GsIDNo.set("")
        Rsvrd_Gst_Rm.set("")
        Rsvrd_Gst_Rm_Type.set("")
        din.set("")
        dout.set("")
        Rsvrd_Gst_rm_price.set("")
        Rsvrd_Gst_Pymnt.set("")
        Rsvrd_Gst_mydata.set("")
        Rsvrd_Gst_cal.set_date(date.today())
    # root.destroy()
tkinter.Button(can_widget5, image=Guest_Entry, compound=LEFT,command=Rsvrd_Gst_CHk,
fg="Black", width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Check
In", bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=1050, y=700)
def Rsvrd_Gst_cls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    # root.destroy()
tkinter.Button(can_widget5, image=Guest_Entry, compound=LEFT,command=Rsvrd_Gst_cls, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=700)
def idd():
    customtkinter.CTkLabel(master=can_widget5,text="Guest Id
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=60+120)
    customtkinter.CTkLabel(master=can_widget5,text="Guest Name
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=105+120)
    customtkinter.CTkLabel(master=can_widget5,text="Gender
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=150+120)
    customtkinter.CTkLabel(master=can_widget5,text="Religion
```

```
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=195+120)
   customtkinter.CTkLabel(master=can_widget5,text="Address
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=240+120)
   customtkinter.CTkLabel(master=can_widget5,text="City
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=285+120)
   customtkinter.CTkLabel(master=can_widget5,text="Country
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=330+120)
   customtkinter.CTkLabel(master=can_widget5,text="Contact No
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=375+120)
   customtkinter.CTkLabel(master=can_widget5,text="ID TYPE
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=420+120)
   customtkinter.CTkLabel(master=can_widget5,text="ID NUMBER
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=20+130
+200, y=465+120)

   #-----------------------------------------------------------------------------------------
   customtkinter.CTkLabel(master=can_widget5, text="Room No
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=180)
   customtkinter.CTkLabel(master=can_widget5, text="Room Type
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=105+120)
   customtkinter.CTkLabel(master=can_widget5, text="Date In
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=150+120)
   customtkinter.CTkLabel(master=can_widget5, text="Date Out
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=195+120)
   customtkinter.CTkLabel(master=can_widget5, text="Room Price
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=240+120)
   customtkinter.CTkLabel(master=can_widget5, text="Payment
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=285+120)
   customtkinter.CTkLabel(master=can_widget5, text="Payment Mode
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100,y=330+120)
   customtkinter.CTkLabel(master=can_widget5, text="Payment Date
:",text_color="Black",fg_color="#ffe000",bg_color="#ffe000",font=('Pristina',25,"bold")).place(x=510+13
0+100, y=375+120)
idd()
Rsvrd_Gst_GsID=StringVar()
Rsvrd_Gst_GsNm=StringVar()
Rsvrd_Gst_Gsder=StringVar()
Rsvrd_Gst_Gsgion=StringVar()
Rsvrd_Gst_GsAddress=StringVar()
Rsvrd_Gst_GsCity=StringVar()
Rsvrd_Gst_GsCntry=StringVar()
```

```
Rsvrd_Gst_GsCntNO=StringVar()
Rsvrd_Gst_GsIDType=StringVar()
Rsvrd_Gst_GsIDNo=StringVar()
Rsvrd_Gst_GsID.set("")
Rsvrd_Gst_GsNm.set("")
Rsvrd_Gst_Gsder.set("")
Rsvrd_Gst_Gsgion.set("")
Rsvrd_Gst_GsAddress.set("")
Rsvrd_Gst_GsCity.set("")
Rsvrd_Gst_GsCntry.set("")
Rsvrd_Gst_GsCntNO.set("")
Rsvrd_Gst_GsIDType.set("")
Rsvrd_Gst_GsIDNo.set("")
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsID,highlightbackground="grey",highli
ghtcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+65+100,width=150,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsNm,highlightbackground="grey",highli
ghtcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+105+12+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_Gsder,highlightbackground="grey",highli
ghtcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+150+25+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_Gsgion,highlightbackground="grey",highl
ightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+195+35+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsAddress,highlightbackground="grey",h
ighlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+240+47+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsCity,highlightbackground="grey",highl
ightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+285+60+100,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsCntry,highlightbackground="grey",hig
hlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+330+70+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsCntNO,highlightbackground="grey",hi
ghlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+375+80+105,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsIDType,highlightbackground="grey",h
ighlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+420+95+100,width=200,height=30)
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_GsIDNo,highlightbackground="grey",hig
hlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+320,y=65+420+95+50+105,width=200,height=30)


Rsvrd_Gst_Rm=StringVar()
# file1 = open("readme.txt", "r")
file = pd.read_csv('RMNO_RMPRICE.csv',index_col=[0])
# Rsvrd_Gst_Rm.set(file.RoomDetails[1])
Entry(can_widget5,textvariable=Rsvrd_Gst_Rm,highlightthickness=2,highlightbackground="grey",highligh
tcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=690+350+80,y=65+65+97,width=50,height=30)#690+350+80
Rsvrd_Gst_Rm_Type=StringVar()
# Rsvrd_Gst_Rm_Type.set(file.RoomDetails[2])
```

```
Entry(can_widget5,textvariable=Rsvrd_Gst_Rm_Type,highlightthickness=2,highlightbackground="grey",hi
ghlightcolor="black",fg="blue",font="consolas 15
italic",state="readonly").place(x=690+350+80,y=65+105+12+102,width=135,height=30)
# Rsvrd_Gst_Person=StringVar()
#
Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_Person,highlightbackground="grey",highl
ightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=690+350+80,y=270+20+65,width=50,height=30)
#    #-----------------------------------------------------------------------------------------
din=StringVar()
Entry(can_widget5,highlightthickness=2,textvariable=din,highlightbackground="grey",highlightcolor="blac
k",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+350+80,
y=70+140+25+105,width=200,height=30)

dout=StringVar()
Entry(can_widget5,highlightthickness=2,textvariable=dout,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+350+80,
y=70+180+45+102,width=200,height=30)
Rsvrd_Gst_rm_price=StringVar()
Entry(can_widget5,highlightthickness=2,state="readonly",textvariable=Rsvrd_Gst_rm_price,highlightbackg
round="grey",highlightcolor="black",fg="blue",font="consolas 17
italic").place(x=690+350+80,y=65+150+25+213,width=135,height=30)
Rsvrd_Gst_Pymnt=StringVar()
Rsvrd_Gst_en=Entry(can_widget5,highlightthickness=2,textvariable=Rsvrd_Gst_Pymnt,highlightbackgroun
d="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
Rsvrd_Gst_en.place(x=690+350+80,y=65+195+35+213,width=200,height=30)

Rsvrd_Gst_cal = DateEntry(can_widget5, selectmode="day", font=("Cambria", 13,
"italic"),foreground="blue", width=10+2)
Rsvrd_Gst_cal.place(x=690+350+80, y=65+240+47+270)

Rsvrd_Gst_mydata = ttk.Combobox(can_widget5, foreground="black", justify=LEFT, font="Calibri 13",
width=12, state='readonly',background="grey", height=10)

Rsvrd_Gst_mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
Rsvrd_Gst_mydata.set("Cash")
Rsvrd_Gst_mydata.place(x=690+350+80, y=65+285+60+155)
# #-----------------------------------------------------------------------------------------
Rsvrd_Gst_frm = Frame(can_widget5, relief=SUNKEN, borderwidth=4)
# Rsvrd_Gst_frm.place(x=700, y=700, width=480, height=200)
Rsvrd_Gst_scbr_x = Scrollbar(Rsvrd_Gst_frm, orient=HORIZONTAL)
Rsvrd_Gst_scbr_y = Scrollbar(Rsvrd_Gst_frm, orient=VERTICAL)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
```

```
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light grey")

#     TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light grey")
Rsvrd_Gst_table = ttk.Treeview(Rsvrd_Gst_frm, cursor="hand2", columns=("SN_No.", "Payment_MD", "Payment", "Payment_Dt"),selectmode="browse",
xscrollcommand=Rsvrd_Gst_scbr_x.set,yscrollcommand=Rsvrd_Gst_scbr_y.set)

Rsvrd_Gst_scbr_x.pack(side=BOTTOM, fill=X)
Rsvrd_Gst_scbr_y.pack(side=RIGHT, fill=Y)
Rsvrd_Gst_scbr_x.config(command=Rsvrd_Gst_table.xview)
Rsvrd_Gst_scbr_y.config(command=Rsvrd_Gst_table.yview)
Rsvrd_Gst_table.heading("SN_No.", text="Sn No.", anchor=CENTER)
Rsvrd_Gst_table.heading("Payment_MD", text="Payment Mode", anchor=CENTER)
Rsvrd_Gst_table.heading("Payment", text="Payment", anchor=CENTER)
Rsvrd_Gst_table.heading("Payment_Dt", text="Payment Date", anchor=CENTER)
Rsvrd_Gst_table.pack(fill=BOTH, expand=1)


Rsvrd_Gst_table["show"] = "headings"
Rsvrd_Gst_table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
Rsvrd_Gst_table.column("Payment_MD", width=170, anchor=CENTER, minwidth=150)
Rsvrd_Gst_table.column("Payment", width=100, anchor=CENTER, minwidth=70)
Rsvrd_Gst_table.column("Payment_Dt", width=140, anchor=CENTER, minwidth=120)
```




```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
```

```
import pandas as pd
import mysql.connector as connector
from tkinter import  messagebox
from PIL import ImageTk,Image
import pandas as pd
from tkcalendar import DateEntry
from datetime import date
import os
height = 730
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))

con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()

Label(text="List Of Check In",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)

img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=550,bg_color="black")
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Search By Guest Id :", font=('Century Gothic',
16)).place(x=20, y=30)
customtkinter.CTkLabel(master=frame, text="Search By Guest Name :", font=('Century Gothic',
16)).place(x=20, y=80)
customtkinter.CTkLabel(master=frame, text="Search By Contact Number :", font=('Century Gothic',
16)).place(x=20, y=130)
E1Var=StringVar()
E1Var.set("")
E2Var=StringVar()
E2Var.set("")
E3Var=StringVar()
E3Var.set("")
E1=Entry(frame,highlightthickness=2,textvariable=E1Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E1.place(x=280,y=30,width=200,height=30)
E2=Entry(frame,highlightthickness=2,textvariable=E2Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E2.place(x=280,y=80,width=200,height=30)
E3=Entry(frame,highlightthickness=2,textvariable=E3Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
```

```
E3.place(x=280,y=130,width=200,height=30)

def Scrh():
   if E1.get()==""and E2.get()=="":
      for item in table.get_children():
         table.delete(item)
      query = f"select * from `check in details`  right join `customer details` using (`Guest ID`) where `check in
details`.`Room No` is not null and `Contact No.`='{E3.get()}';"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         table.insert("", END, values=(
         sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[12], row[9], row[10],
row[15],
         row[16], row[17], row[18], row[19], row[20], row[24]))
         sn += 1
   elif E2.get()==""and E3.get()=="":
      for item in table.get_children():
         table.delete(item)
      query = f"select * from `check in details`  right join `customer details` using (`Guest ID`) where `check in
details`.`Room No` is not null  and `Guest ID`='{E1.get()}';"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         table.insert("", END, values=(
         sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[12], row[9], row[10],
row[15],
         row[16], row[17], row[18], row[19], row[20], row[24]))
         sn += 1
   elif E1.get()==""and E3.get()=="":
      for item in table.get_children():
         table.delete(item)
      query = f"select * from `check in details`  right join `customer details` using (`Guest ID`) where `check in
details`.`Room No` is not null  and `Guest Name`='{E2.get()}';"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         table.insert("", END, values=(
         sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[12], row[9], row[10],
row[15],
         row[16], row[17], row[18], row[19], row[20], row[24]))
         sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Scrh, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Search", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=500+20, y=30)
def Rset():
   E1Var.set("")
   E2Var.set("")
   E3Var.set("")
   for item in table.get_children():
      table.delete(item)
   query = "select * from `check in details`  right join `customer details` using (`Guest ID`) where `check in
details`.`Room No` is not null;"
   cur.execute(query)
```

```
    sn = 1
    for row in cur.fetchall():
        table.insert("", END, values=(
        sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[12], row[9], row[10], row[15],
row[16],
        row[17], row[18], row[19], row[20], row[24]))
        sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Rset, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650+20, y=30)
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Excel", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650, y=30)
def cls():
    fu = ["","","","","","","","","","","","","","","","","","","","","","","passportsizephoto.webp"]
    so = pd.Series(data=fu, name="ChkIn")
    sep = pd.DataFrame(so)
    sep.to_csv("Checkin_list.csv")
    root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=800+20, y=30)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=20, y=200, width=950, height=300)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Gs_ID", "Gs_Name",
"Gender","Religion","Address","City","Country","Contact_No","Email
Id","ID_Type","ID_No","Room_No","Room_Type","Date_In","Date_Out","Room_Charges","Payment
","Booking_Type",'Image'),
                    selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
```

```
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Gs_ID", text="Guest ID", anchor=CENTER)
table.heading("Gs_Name", text="Guest Name", anchor=CENTER)
table.heading("Gender", text="Gender", anchor=CENTER)
table.heading("Religion", text="Religion", anchor=CENTER)
table.heading("Address", text="Address", anchor=CENTER)
table.heading("City", text="City", anchor=CENTER)
table.heading("Country", text="Country", anchor=CENTER)
table.heading("Contact_No", text="Contact Number", anchor=CENTER)
table.heading("Email Id", text="Email Id", anchor=CENTER)
table.heading("ID_Type", text="ID Type", anchor=CENTER)
table.heading("ID_No", text="ID No", anchor=CENTER)
table.heading("Room_No", text="Room No", anchor=CENTER)
table.heading("Room_Type", text="Room Type", anchor=CENTER)
table.heading("Date_In", text="Date In", anchor=CENTER)
table.heading("Date_Out", text="Date Out", anchor=CENTER)
table.heading("Room_Charges", text="Room Charges", anchor=CENTER)
table.heading("Payment", text="Payment", anchor=CENTER)
table.heading("Booking_Type", text="Booking Type", anchor=CENTER)
table.heading("Image", text="Image", anchor=CENTER)
table.pack(fill=BOTH, expand=1)


table["show"] = "headings"
table.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
table.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
table.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
table.column("Gender", width=130, anchor=CENTER, minwidth=130)
table.column("Religion", width=140, anchor=CENTER, minwidth=140)
table.column("Address", width=140, anchor=CENTER, minwidth=140)
table.column("City", width=140, anchor=CENTER, minwidth=140)
table.column("Country", width=140, anchor=CENTER, minwidth=140)
table.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
table.column("Email Id", width=160, anchor=CENTER, minwidth=140)
table.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
table.column("ID_No", width=140, anchor=CENTER, minwidth=120)
table.column("Room_No", width=140, anchor=CENTER, minwidth=120)
table.column("Room_Type", width=140, anchor=CENTER, minwidth=120)
table.column("Date_In", width=140, anchor=CENTER, minwidth=120)
table.column("Date_Out", width=140, anchor=CENTER, minwidth=120)
table.column("Room_Charges", width=140, anchor=CENTER, minwidth=120)
table.column("Payment", width=140, anchor=CENTER, minwidth=120)
table.column("Booking_Type", width=140, anchor=CENTER, minwidth=120)
table.column("Image", width=0, anchor=CENTER, minwidth=0)
query = "select * from `check in details`  right join `customer details` using (`Guest ID`) where `check in
details`.`Room No` is not null;"
cur.execute(query)
sn=1
for row in cur.fetchall():
   table.insert("",END, values=(sn,row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8],row[12],
row[9], row[10],row[15], row[16], row[17], row[18], row[19],row[20],row[24],row[13]))
   sn+=1
# query = "select * from `check in details`  right join `customer details` using (`Guest ID`);"
```

```
# cur.execute(query)
# for row in cur.fetchall():
#     table.insert("",END, values=(row[1], row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
row[9], row[10], row[13], row[14], row[15], row[16], row[17], row[18]))

def table_select(_):
    fu=[]
    for i in table.selection():
        fu.append(table.item(i)['values'])
    so = pd.Series(data=fu[0], name="ChkIn")
    sep = pd.DataFrame(so)
    sep.to_csv("Checkin_list.csv")
    # print(so)
    root.destroy()
table.bind('<<TreeviewSelect>>',table_select)

root.mainloop()
```


```
#-------------------------          CHECK OUT      ------------------------------------------------------------------

can_widget1 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
ChkOutBg = ImageTk.PhotoImage(Image.open("./assets/main-pool.jpg").resize((1585,955)))
can_widget1.create_image(0,0,anchor=NW,image=ChkOutBg)
ChkOUTImg = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231108_195456.png").resize((1150,1080)))
can_widget1.create_image(830,530,image=ChkOUTImg)
# can_widget1.place(x=330, y=25)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=can_widget1,text="Guest
```

```
Information",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).pla
ce(x=480, y=80)
Pymnt=DoubleVar()
ChkotCstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
ChkotImglbl=Label(can_widget1,image=ChkotCstmrImg)
ChkotImglbl.place(x=855, y=110)
def Gstry():
    global ChkotCstmrImg
    os.system("python Check_In_List.py")
    Gs = pd.read_csv("Checkin_list.csv", index_col=[0])
    if Gs.ChkIn[18]=="Reservation":
        # print("Reservation")
        rsvrdpymnt.place(x=920 + 250, y=677, width=200, height=30)
        prermamnt.place(x=920 + 250, y=727, width=200, height=30)
        # prermamnt.place(x=920 + 250, y=677, width=200, height=30)
        rmamnt.place(x=510 + 50 + 210, y=580)
        rsvpmnt.place(x=510 + 50 + 210, y=540)
        PymMd.place(x=510 + 50 + 210, y=620)
        ChkPym.place(x=510 + 50 + 210, y=660)
        ChkOutmydata.place(x=920 + 250, y=777)
        en.place(x=920 + 250, y=827, width=140, height=30)
        GsID.set(Gs.ChkIn[1])
        GsNm.set(Gs.ChkIn[2])
        Gsder.set(Gs.ChkIn[3])
        Gsgion.set(Gs.ChkIn[4])
        GsAddress.set(Gs.ChkIn[5])
        GsCity.set(Gs.ChkIn[6])
        GsCntry.set(Gs.ChkIn[7])
        GsCntNO.set(Gs.ChkIn[8])
        GsIDType.set(Gs.ChkIn[9+1])
        GsIDNo.set(Gs.ChkIn[10+1])
        Rm.set(Gs.ChkIn[11+1])
        RType.set(Gs.ChkIn[12+1])
        DateIn.set(Gs.ChkIn[13+1])
        DateOut.set(Gs.ChkIn[14+1])
        RoomCharge.set(Gs.ChkIn[15+1])
        AdvancePayment.set(Gs.ChkIn[16+1])
        #TODO
        Discount_Price.set(value=float(Gs.ChkIn[15+1])*(Discount.get()/100))
        SGST_Price.set(value=float(Gs.ChkIn[15+1])*(SGST.get()/100))
        CGST_Price.set(value=float(Gs.ChkIn[15+1])*(CGST.get())/100)
        GrandTotal.set(float(Gs.ChkIn[15+1])-Discount_Price.get()+SGST_Price.get()+CGST_Price.get())
        query=f"select * from `reservation details` where `Guest ID` = '{Gs.ChkIn[1]}';"
        cur.execute(query)
        ve=cur.fetchone()
        # print(cur.fetchone()[8])
        rsvrd_pymnt.set(ve[7])
        RemainingAmnt.set(GrandTotal.get()-AdvancePayment.get()+rsvrd_pymnt.get())
        ChkotCstmrImg =ImageTk.PhotoImage(Image.open(Gs.ChkIn[19]).resize((100, 100)))
        ChkotImglbl.configure(image=ChkotCstmrImg)
    else:
        rmamnt.place(x=510 + 50 + 210, y=540)
        PymMd.place(x=510 + 50 + 210, y=580)
        ChkPym.place(x=510 + 50 + 210, y=620)
```

```
    prermamnt.place(x=920 + 250, y=677, width=200, height=30)
    ChkOutmydata.place(x=920 + 250, y=727)
    ChkOutmydata.place(x=920 + 250, y=727)
    en.place(x=920 + 250, y=777, width=140, height=30)
    rsvrdpymnt.place(x=920 + 250, y=60000, width=200, height=30)
    rsvpmnt.place(x=510 + 50 + 210000, y=540)
    # print("Direct")
    GsID.set(Gs.ChkIn[1])
    GsNm.set(Gs.ChkIn[2])
    Gsder.set(Gs.ChkIn[3])
    Gsgion.set(Gs.ChkIn[4])
    GsAddress.set(Gs.ChkIn[5])
    GsCity.set(Gs.ChkIn[6])
    GsCntry.set(Gs.ChkIn[7])
    GsCntNO.set(Gs.ChkIn[8])
    GsIDType.set(Gs.ChkIn[10])
    GsIDNo.set(Gs.ChkIn[10+1])
    Rm.set(Gs.ChkIn[11+1])
    RType.set(Gs.ChkIn[12+1])
    DateIn.set(Gs.ChkIn[13+1])
    DateOut.set(Gs.ChkIn[14+1])
    RoomCharge.set(Gs.ChkIn[15+1])
    AdvancePayment.set(Gs.ChkIn[16+1])
    rsvrd_pymnt.set(0)
    Discount_Price.set(value=float(Gs.ChkIn[15+1])*(Discount.get()/100))
    SGST_Price.set(value=float(Gs.ChkIn[15+1])*(SGST.get()/100))
    CGST_Price.set(value=float(Gs.ChkIn[15+1])*(CGST.get())/100)
    GrandTotal.set(float(Gs.ChkIn[15+1])-Discount_Price.get()+SGST_Price.get()+CGST_Price.get())
    RemainingAmnt.set(GrandTotal.get()-AdvancePayment.get())
    ChkotCstmrImg =ImageTk.PhotoImage(Image.open(Gs.ChkIn[19]).resize((100, 100)))
    ChkotImglbl.configure(image=ChkotCstmrImg)
  # print(56565656555555555555555555555)
tkinter.Button(can_widget1, image=Guest_Entry,compound=CENTER,command=Gstry, fg="Black",
width=50, activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", anchor=W,
borderwidth=5, cursor="hand2").place(x=1390, y=150)
def reset():
  global ChkotCstmrImg
  Rm.set("")
  RType.set("")
  DateIn.set("")
  DateOut.set("")
  RoomCharge.set("")
  Discount_Price.set("")
  SGST_Price.set("")
  CGST_Price.set("")
  GrandTotal.set("")
  AdvancePayment.set("")
  RemainingAmnt.set("")
  Pymnt.set("")
  GsID.set("")
  GsNm.set("")
  Gsder.set("")
  Gsgion.set("")
  GsAddress.set("")
```

```
    GsCity.set("")
    GsCntry.set("")
    GsCntNO.set("")
    GsIDType.set("")
    GsIDNo.set("")
    rsvrd_pymnt.set("")
    ChkotCstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    ChkotImglbl.configure(image=ChkotCstmrImg)
tkinter.Button(can_widget1, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset",command=reset,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=1390, y=210)
def CHk():
    if Pymnt.get()==RemainingAmnt.get():
        if messagebox.askyesno("Check Out", "Are You Sure You Want To Check Out"):
            Gs = pd.read_csv("Checkin_list.csv", index_col=[0])
            con = connector.connect(host='localhost',
                                    port='3306',
                                    user='root',
                                    password='Password',
                                    database='Hotel Management Software')
            cur = con.cursor()

            if rsvrd_pymnt.get() == 0:
                # print("Direct")
                query = (f"insert into ChkOut values("

f"'{GsID.get()}','{GsNm.get()}','{Gsder.get()}','{Gsgion.get()}','{GsAddress.get()}','{GsCity.get()}','{GsCnt
ry.get()}',"

f"'{GsCntNO.get()}','{Gs.ChkIn[9]}','{GsIDType.get()}','{GsIDNo.get()}','{Rm.get()}','{DateIn.get()}','{Da
teOut.get()}','{RType.get()}',"

f"'{RoomCharge.get()}','{Discount.get()}','{SGST.get()}','{CGST.get()}','{GrandTotal.get()}',{rsvrd_pymnt.
get()},'{AdvancePayment.get()}','{RemainingAmnt.get()}','{ChkOutmydata.get()}','{date.today()}','{en.get()}
');")
            else:
                # print("reservation")
                query = f"delete from `reservation details` where `Room No`= ('{Rm.get()}');"
                cur.execute(query)
                con.commit()
                query = (f"insert into ChkOut values("

f"'{GsID.get()}','{GsNm.get()}','{Gsder.get()}','{Gsgion.get()}','{GsAddress.get()}','{GsCity.get()}','{GsCnt
ry.get()}',"

f"'{GsCntNO.get()}','{Gs.ChkIn[9]}','{GsIDType.get()}','{GsIDNo.get()}','{Rm.get()}','{DateIn.get()}','{Da
teOut.get()}','{RType.get()}',"

f"'{RoomCharge.get()}','{Discount.get()}','{SGST.get()}','{CGST.get()}','{GrandTotal.get()}','{rsvrd_pymnt.
get()}','{AdvancePayment.get()}','{RemainingAmnt.get()}','{ChkOutmydata.get()}','{date.today()}','{en.get()
}');")
            # print(query)
            cur.execute(query)
```

```
        con.commit()
        query = f"DELETE FROM `Check In Details` WHERE (`Room No` = '{Rm.get()}');"
        cur.execute(query)
        con.commit()
        query = f"insert into `Room Status` values
('{Rm.get()}','{RType.get()}','Dirty','{RoomCharge.get()}');"
        cur.execute(query)
        con.commit()
        query = f"update `Customer Details` set `Status`='Inactive' where `Guest ID`='{GsID.get()}';"
        cur.execute(query)
        con.commit()
        Rm.set("")
        RType.set("")
        DateIn.set("")
        DateOut.set("")
        RoomCharge.set("")
        Discount.set("5.3")
        SGST.set("9.2")
        CGST.set("9.4")
        Discount_Price.set("")
        SGST_Price.set("")
        CGST_Price.set("")
        GrandTotal.set("")
        AdvancePayment.set("")
        RemainingAmnt.set("")
        rsvrd_pymnt.set("")
        GsID.set("")
        GsNm.set("")
        Gsder.set("")
        Gsgion.set("")
        GsAddress.set("")
        GsCity.set("")
        GsCntry.set("")
        GsCntNO.set("")
        GsIDType.set("")
        GsIDNo.set("")
        Pymnt.set("")
        global ChkotCstmrImg
        ChkotCstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
        ChkotImglbl.configure(image=ChkotCstmrImg)
    else:
        messagebox.showerror("Check Out","Payment Not Equal To Remaining Amount")
tkinter.Button(can_widget1, image=Guest_Entry, compound=LEFT,command=CHk, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30, text="Check Out",
bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=1390, y=270)
def cls():
    f1.place(x=15,y=21)
    # f2.place(x=260,y=20)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
```

```
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
tkinter.Button(can_widget1, image=Guest_Entry, compound=LEFT,command=cls, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=1390, y=330)
def idd():
    customtkinter.CTkLabel(master=can_widget1,text="Guest Id
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=137)
    customtkinter.CTkLabel(master=can_widget1,text="Guest Name
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=182)
    customtkinter.CTkLabel(master=can_widget1,text="Gender
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=227)
    customtkinter.CTkLabel(master=can_widget1,text="Religion
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=272)
    customtkinter.CTkLabel(master=can_widget1,text="Address
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=317)
    customtkinter.CTkLabel(master=can_widget1,text="City
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=362)
    customtkinter.CTkLabel(master=can_widget1,text="Country
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=407)
    customtkinter.CTkLabel(master=can_widget1,text="Contact No
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=452)
    customtkinter.CTkLabel(master=can_widget1,text="ID TYPE
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=450,
y=497)
    # customtkinter.CTkLabel(master=can_widget1,fg_color="black",bg_color="black",text="ID NUMBER
:",font=('Century Gothic',16)).place(x=450, y=50+465)

    #-------------------------------------------------------------------------------------
    customtkinter.CTkLabel(master=can_widget1,text="Room No
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=60+80)
    customtkinter.CTkLabel(master=can_widget1,text="Date In
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=100+80)
    customtkinter.CTkLabel(master=can_widget1,text="Date Out
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=140+80)
```

```
   customtkinter.CTkLabel(master=can_widget1,text="Room Type
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=180+80)
   # customtkinter.CTkLabel(master=can_widget1,fg_color="black",bg_color="black", text="Room Service
Charge :", font=('Century Gothic', 16)).place(x=510+50, y=220)
   # customtkinter.CTkLabel(master=can_widget1,fg_color="black",bg_color="black", text="Laundry
Charge :", font=('Century Gothic', 16)).place(x=510+50, y=260)
   customtkinter.CTkLabel(master=can_widget1,text="Room Charges
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=300)
   customtkinter.CTkLabel(master=can_widget1, text="Discount
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=340)
   customtkinter.CTkLabel(master=can_widget1, text="SGST
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=380)
   customtkinter.CTkLabel(master=can_widget1, text="CGST
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=420)
   customtkinter.CTkLabel(master=can_widget1, text="Grand Total
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=460)
   customtkinter.CTkLabel(master=can_widget1, text="Advance Payment
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x=510+5
0+210, y=500)
   customtkinter.CTkLabel(master=can_widget1,
text="%",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x
=510+50+245+210, y=380 - 40)
   customtkinter.CTkLabel(master=can_widget1,
text="%",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x
=510+50+245+210, y=420 - 40)
   customtkinter.CTkLabel(master=can_widget1,
text="%",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold")).place(x
=510+50+245+210, y=460 - 40)
rmamnt=customtkinter.CTkLabel(master=can_widget1, text="Remaining Amount
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold"))
rmamnt.place(x=510+50+210, y=540)
rsvpmnt= customtkinter.CTkLabel(master=can_widget1, text="Reservation Payment
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold"))
PymMd=customtkinter.CTkLabel(master=can_widget1,text="Payment Mode
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold"))
PymMd.place(x=510+50+210, y=580)
ChkPym=customtkinter.CTkLabel(master=can_widget1,text="Payment
:",text_color="Black",fg_color="#f5d79e",bg_color="#f5d79e",font=('Pristina',25,"bold"))
ChkPym.place(x=510+50+210, y=620)
#-----------------------------------------------------------------------------------------------------------
Rm=StringVar()
RType=StringVar()
DateIn=StringVar()
DateOut=StringVar()
RoomCharge=DoubleVar()
Discount=DoubleVar()
SGST=DoubleVar()
CGST=DoubleVar()
```

```
Discount_Price=DoubleVar()
SGST_Price=DoubleVar()
CGST_Price=DoubleVar()
GrandTotal=DoubleVar()
AdvancePayment=DoubleVar()
RemainingAmnt=DoubleVar()
Rm.set("")
RType.set("")
DateIn.set("")
DateOut.set("")
RoomCharge.set("")
Discount.set(5.3)
SGST.set(9.2)
CGST.set(9.4)
Discount_Price.set("")
SGST_Price.set("")
CGST_Price.set("")
GrandTotal.set("")
AdvancePayment.set("")
RemainingAmnt.set("")
Entry(can_widget1,textvariable=Rm,highlightthickness=2,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17
italic",state="readonly").place(x=690+50+180+250,y=60+20+100,width=100,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=RType,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17
italic",state="readonly").place(x=690+50+180+250,y=210+20+100,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=DateIn,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+50+180+250,
y=110+20+100,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=DateOut,highlightbackground="grey",highlightcolor
="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+50+180+250,
y=160+20+100,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=RoomCharge,highlightbackground="grey",highlightc
olor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=920+250,y=378,width=200,height=30)

Entry(can_widget1,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",fo
nt="consolas 17 italic",textvariable=Discount).place(x=920+250,y=427,width=90,height=30)
Entry(can_widget1,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",fo
nt="consolas 17 italic",textvariable=SGST).place(x=920+250,y=475,width=90,height=30)
Entry(can_widget1,highlightthickness=2,highlightbackground="grey",highlightcolor="black",fg="blue",fo
nt="consolas 17 italic",textvariable=CGST).place(x=920+250,y=525,width=90,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=Discount_Price,highlightbackground="grey",highligh
tcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=1040+250,y=427,width=80,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=SGST_Price,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=1040+250,y=475,width=80,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=CGST_Price,highlightbackground="grey",highlightco
lor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=1040+250,y=525,width=80,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GrandTotal,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=920+250,y=575,width=200,height=30)
```

```
Entry(can_widget1,highlightthickness=2,textvariable=AdvancePayment,highlightbackground="grey",highli
ghtcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=920+250,y=627,width=200,height=30)
rsvrd_pymnt=DoubleVar()
rsvrd_pymnt.set("")
rsvrdpymnt=Entry(can_widget1,highlightthickness=2,textvariable=rsvrd_pymnt,highlightbackground="grey
",highlightcolor="black",fg="blue",font="consolas 17 italic",state="readonly")
prermamnt=Entry(can_widget1,highlightthickness=2,textvariable=RemainingAmnt,highlightbackground="g
rey",highlightcolor="black",fg="blue",font="consolas 17 italic",state="readonly")
prermamnt.place(x=920+250,y=677,width=200,height=30)
ChkOutmydata = ttk.Combobox(can_widget1, foreground="black", justify=LEFT, font="Calibri 13",
width=10, state='readonly',background="grey", height=10)

ChkOutmydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
l = []
ChkOutmydata.set("Cash")
ChkOutmydata.place(x=920+250,y=727)
en=Entry(can_widget1,highlightthickness=2,textvariable=Pymnt,highlightbackground="grey",highlightcolo
r="black",fg="blue",font="consolas 17 italic")
en.place(x=920+250,y=777,width=140,height=30)

#-----------------------------------------------------------------------------------------------------------------------


idd()
GsID=StringVar()
GsNm=StringVar()
Gsder=StringVar()
Gsgion=StringVar()
GsAddress=StringVar()
GsCity=StringVar()
GsCntry=StringVar()
GsCntNO=StringVar()
GsIDType=StringVar()
GsIDNo=StringVar()
GsID.set("")
GsNm.set("")
Gsder.set("")
Gsgion.set("")
GsAddress.set("")
GsCity.set("")
GsCntry.set("")
GsCntNO.set("")
GsIDType.set("")
GsIDNo.set("")
Entry(can_widget1,highlightthickness=2,textvariable=GsID,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=175,width=100,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsNm,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=230,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=Gsder,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=280,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=Gsgion,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=330,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsAddress,highlightbackground="grey",highlightcolo
```

```
r="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=380,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsCity,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=430,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsCntry,highlightbackground="grey",highlightcolor
="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=480,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsCntNO,highlightbackground="grey",highlightcolo
r="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=530,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsIDType,highlightbackground="grey",highlightcolo
r="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=580,width=200,height=30)
Entry(can_widget1,highlightthickness=2,textvariable=GsIDNo,highlightbackground="grey",highlightcolor
="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=750,y=630,width=200,height=30)

Chk_Out_Frame = Frame(can_widget1, relief=SUNKEN, borderwidth=4)
# Chk_Out_Frame.place(x=700, y=720, width=480, height=200)
scbr_x = Scrollbar(Chk_Out_Frame, orient=HORIZONTAL)
scbr_y = Scrollbar(Chk_Out_Frame, orient=VERTICAL)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Chk_Out_table = ttk.Treeview(Chk_Out_Frame, cursor="hand2", columns=("SN_No.", "Payment_MD",
"Payment", "Payment_Dt"),
                 selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=Chk_Out_table.xview)
scbr_y.config(command=Chk_Out_table.yview)
Chk_Out_table.heading("SN_No.", text="Sn No.", anchor=CENTER)
Chk_Out_table.heading("Payment_MD", text="Payment Mode", anchor=CENTER)
Chk_Out_table.heading("Payment", text="Payment", anchor=CENTER)
Chk_Out_table.heading("Payment_Dt", text="Payment Date", anchor=CENTER)
Chk_Out_table.pack(fill=BOTH, expand=1)


Chk_Out_table["show"] = "headings"
Chk_Out_table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
Chk_Out_table.column("Payment_MD", width=170, anchor=CENTER, minwidth=150)
```

```
Chk_Out_table.column("Payment", width=100, anchor=CENTER, minwidth=70)
Chk_Out_table.column("Payment_Dt", width=140, anchor=CENTER, minwidth=120)
```


```
can_widget13 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
RsvrDetailsimgbg =
ImageTk.PhotoImage(Image.open("1525415163_shutterstock_462136018_(1).jpg.webp").resize((1585,955)))

can_widget13.create_image(0,0,anchor=NW,image=RsvrDetailsimgbg)
can_widget13.place(x=330, y=25)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#     TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#     TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Chk_Gst_Dtl = Frame(can_widget13, relief=SUNKEN, borderwidth=4)
Chk_Gst_Dtl.place(x=190, y=400, width=1200, height=300)
Chk_Gst_Dtl_scbr_x = Scrollbar(Chk_Gst_Dtl, orient=HORIZONTAL)
Chk_Gst_Dtl_scbr_y = Scrollbar(Chk_Gst_Dtl, orient=VERTICAL)
```

```
Chk_Gst_Dtl_Trvw = ttk.Treeview(Chk_Gst_Dtl, cursor="hand2", columns=("SN_No.", "Gst_ID",
"Gst_Nm","Gndr","rlign","Address","City","Cntry","Cnt_No","EmID","ID_Type","ID_No","RmNo",
"Dt_In","Dt_Ot","RmTyp","RmChrg","Dst","Sgst","Cgst","Ttl","Rsvr","Adv","Rmant","Pymnt_Md"
,"PymDt","Pmnt"),selectmode="browse", xscrollcommand=Chk_Gst_Dtl_scbr_x.set,
yscrollcommand=Chk_Gst_Dtl_scbr_y.set)

Chk_Gst_Dtl_scbr_x.pack(side=BOTTOM, fill=X)
Chk_Gst_Dtl_scbr_y.pack(side=RIGHT, fill=Y)
Chk_Gst_Dtl_scbr_x.config(command=Chk_Gst_Dtl_Trvw.xview)
Chk_Gst_Dtl_scbr_y.config(command=Chk_Gst_Dtl_Trvw.yview)
Chk_Gst_Dtl_Trvw.heading("SN_No.", text="Sn No.", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Gst_ID", text="Guest ID", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Gst_Nm", text="Guest Name", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Gndr", text="Gender", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("rlign", text="Religion", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Address", text="Address", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("City", text="City", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Cntry", text="Country", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Cnt_No", text="Contact No.", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("EmID", text="Email ID", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("ID_Type", text="ID Type", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("ID_No", text="ID No.", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("RmNo", text="Room No.", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Dt_In", text="From Date", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Dt_Ot", text="To Date", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("RmTyp", text="Room Type", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("RmChrg", text="Room Price", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Dst", text="Discount", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Sgst", text="S.GST", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Cgst", text="C.GST", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Ttl", text="Total Amount", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Rsvr", text="Reservation Amount", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Adv", text="Advance", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Rmant", text="Remaining", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Pymnt_Md", text="Payment Mode", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("PymDt", text="Payment Date", anchor=CENTER)
Chk_Gst_Dtl_Trvw.heading("Pmnt", text="Payment", anchor=CENTER)
Chk_Gst_Dtl_Trvw.pack(fill=BOTH, expand=1)


Chk_Gst_Dtl_Trvw["show"] = "headings"
Chk_Gst_Dtl_Trvw.column("SN_No.", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Gst_ID", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Gst_Nm", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Gndr", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("rlign", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Address", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("City", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Cntry", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Cnt_No", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("EmID", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("ID_Type", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("ID_No", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("RmNo", width=150, anchor=CENTER, minwidth=50)
```

```
Chk_Gst_Dtl_Trvw.column("Dt_In", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Dt_Ot", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("RmTyp", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("RmChrg", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Dst", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Sgst", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Cgst", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Ttl", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Rsvr", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Adv", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Rmant", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Pymnt_Md", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("PymDt", width=150, anchor=CENTER, minwidth=50)
Chk_Gst_Dtl_Trvw.column("Pmnt", width=150, anchor=CENTER, minwidth=50)

can_widget13.create_text(800,100,text="CUSTOMER DETAILS",font=("Pristina", 50, "bold"))
can_widget13.create_text(370,50+190,text="Search By Guest ID :",font=("Pristina", 30, "bold"))
can_widget13.create_text(385,100+190,text="Search By Guest Name :",font=("Pristina", 30, "bold"))
can_widget13.create_text(410,150+190,text="Search By Contact Number :",font=("Pristina", 30, "bold"))
ChkoutE1Var=StringVar()
ChkoutE1Var.set("")
ChkoutE2Var=StringVar()
ChkoutE2Var.set("")
ChkoutE3Var=StringVar()
ChkoutE3Var.set("")
ChkoutE1=Entry(can_widget13,highlightthickness=2,textvariable=ChkoutE1Var,highlightbackground="gre
y",highlightcolor="black",fg="blue",font="consolas 17 italic")
ChkoutE1.place(x=280+380,y=30+190,width=200,height=30)
ChkoutE2=Entry(can_widget13,highlightthickness=2,textvariable=ChkoutE2Var,highlightbackground="gre
y",highlightcolor="black",fg="blue",font="consolas 17 italic")
ChkoutE2.place(x=280+380,y=80+190,width=200,height=30)
ChkoutE3=Entry(can_widget13,highlightthickness=2,textvariable=ChkoutE3Var,highlightbackground="gre
y",highlightcolor="black",fg="blue",font="consolas 17 italic")
ChkoutE3.place(x=280+380,y=130+190,width=200,height=30)
def ChkoutScrh():
  con = connector.connect(host='localhost',
                port='3306',
                user='root',
                password='Password',
                database='Hotel Management Software')
  cur = con.cursor()
  if ChkoutE1.get()==""and ChkoutE2.get()=="":
    for item in Chk_Gst_Dtl_Trvw.get_children():
      Chk_Gst_Dtl_Trvw.delete(item)
    query = f"select * from chkout where `Contact No`='{ChkoutE3.get()}';"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
      # print(row)
      Chk_Gst_Dtl_Trvw.insert("", END,
                values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                  row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16],
                  row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24],
                  row[25]))
```

```
        sn += 1
    elif ChkoutE2.get()==""and ChkoutE3.get()=="":
        for item in Chk_Gst_Dtl_Trvw.get_children():
            Chk_Gst_Dtl_Trvw.delete(item)
        query = f"select * from chkout where `Guest ID`='{ChkoutE1.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            # print(row)
            Chk_Gst_Dtl_Trvw.insert("", END,
                            values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16],
                                row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24],
                                row[25]))
            sn += 1
    elif ChkoutE1.get()==""and ChkoutE3.get()=="":
        for item in Chk_Gst_Dtl_Trvw.get_children():
            Chk_Gst_Dtl_Trvw.delete(item)
        query = f"select * from chkout where `Guest Name`='{ChkoutE2.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            # print(row)
            Chk_Gst_Dtl_Trvw.insert("", END,
                            values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16],
                                row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24],
                                row[25]))
            sn += 1
tkinter.Button(can_widget13, image=Guest_Entry, compound=LEFT,command=ChkoutScrh, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Search",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=500+420, y=210)
def ChkoutRset():
    ChkoutE1Var.set("")
    ChkoutE2Var.set("")
    ChkoutE3Var.set("")
    for item in Chk_Gst_Dtl_Trvw.get_children():
        Chk_Gst_Dtl_Trvw.delete(item)
    con = connector.connect(host='localhost',
                    port='3306',
                    user='root',
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    query = "select * from chkout;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Chk_Gst_Dtl_Trvw.insert("", END,
                        values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                            row[9], row[10], row[11], row[12], row[13], row[14], row[15], row[16], row[17],
                            row[18], row[19], row[20], row[21], row[22], row[23], row[24], row[25]))
```

```
        sn += 1
tkinter.Button(can_widget13, image=Guest_Entry, compound=LEFT,command=ChkoutRset, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Reset",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=650+420, y=210)
def cls():
    f1.place(x=15, y=21)
    can_widgett.place(x=350, y=25)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget13.place(x=1000, y=1000)
tkinter.Button(can_widget13, image=Guest_Entry, compound=LEFT,command=cls, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=800+420, y=210)
```

Aaksham Hotel

Your most memorable days await you

Booking Confrmation

Your reservation is confirmed!

Hi {{Name}},

Thank you for choosing to stay at Aaksham Hotel.

We're pleased to confirm your reservation.

Reservation Details

Lead Guest Name:	{{Name}}
Guest ID:	{{Gs_ID}}
Address:	{{Address}}
Room Number : Room Number :	{{Room_Number}} {{Room_Type}}

Check-In Date:	{{Check_In_Date}}
Check-Out Date:	{{Check_Out_Date}}

Payment Details

Room Charges	${{Room_Price}}
Reservation Charges	${{Payment}}
Total Charge	${{Ttl}}

Cancellation Policy

This booking is non-refundable.

We look forward to welcoming you to Aaksham Hotel.

Got more questions?

Send them to us!

+91-8957488214
aakshamhotel@gmail.com
www.aakshamhotel.com

```
#--------------------------------------------------------          Room Reservation       ------------------------------------------
---------------------------------------
can_widget3 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
RsvrBg = ImageTk.PhotoImage(Image.open("./assets/cinnamon-dhonveli-
maldives(1).jpg").resize((1585,955)))
can_widget3.create_image(0,0,anchor=NW,image=RsvrBg)
RsvrImg = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231107_190710.png").resize((1000,800)))
can_widget3.create_image(500,450,image=RsvrImg)
# can_widget3.place(x=330, y=25)
customtkinter.CTkLabel(master=can_widget3, text="Reservation
List",text_color="Black",fg_color="#f55c3c",bg_color="#f55c3c",font=('Pristina',25,"bold")).place(x=300
, y=120)
# customtkinter.CTkLabel(master=can_widget3, text="Search By", font=('Times New Roman', 20,
"bold"),fg_color="black",bg_color="black").place(x=130, y=50)

def Resvr_idd():
   customtkinter.CTkLabel(master=can_widget3,text="Guest Id
:",text_color="Black",fg_color="#f55c3c",bg_color="#f55c3c",font=('Pristina',25,"bold")).place(x=20+12
0, y=60+130)
   customtkinter.CTkLabel(master=can_widget3,text="Guest Name
:",text_color="Black",fg_color="#f55c3c",bg_color="#f55c3c",font=('Pristina',25,"bold")).place(x=20+12
0, y=105+130)
   customtkinter.CTkLabel(master=can_widget3,text="Room No.
:",text_color="Black",fg_color="#f55c3c",bg_color="#f55c3c",font=('Pristina',25,"bold")).place(x=20+12
0, y=150+130)
   customtkinter.CTkLabel(master=can_widget3,text="Contact No.
:",text_color="Black",fg_color="#f55c3c",bg_color="#f55c3c",font=('Pristina',25,"bold")).place(x=20+12
0, y=195+130)
Resvr_idd()

ResvrGS_ID=StringVar()
ResvrGS_Nm=StringVar()
ResvrGS_Rm_No=StringVar()
ResvrGS_Cnt_No=StringVar()



Entry(can_widget3,highlightthickness=2,textvariable=ResvrGS_ID,highlightbackground="grey",highlightco
lor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150,y=75+65+100,width=100,height=30)
Entry(can_widget3,highlightthickness=2,textvariable=ResvrGS_Nm,highlightbackground="grey",highlightc
olor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150,y=135+65+100,width=200,height=30)
Entry(can_widget3,highlightthickness=2,textvariable=ResvrGS_Rm_No,highlightbackground="grey",highli
ghtcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150,y=190+65+100,width=200,height=30)
Entry(can_widget3,highlightthickness=2,textvariable=ResvrGS_Cnt_No,highlightbackground="grey",highli
ghtcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150,y=245+65+100,width=200,height=30)

Rm_resevrr = Frame(can_widget3, relief=SUNKEN, borderwidth=4)
Rm_resevrr.place(x=365, y=480, width=580, height=250)
scbr_x = Scrollbar(Rm_resevrr, orient=HORIZONTAL)
```

```
scbr_y = Scrollbar(Rm_resevrr, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Rm_resevr = ttk.Treeview(Rm_resevrr, cursor="hand2", columns=("SN_No.", "Gs_ID", "Gs_Name",
"Gender","Religion","Address","City","Country"
,"Contact_No","ID_Type","ID_No","Email_ID","Room_No","Room_Type","Date_In","Date_Out","R
oom_Charges","Payment"),
                    selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=Rm_resevr.xview)
scbr_y.config(command=Rm_resevr.yview)
Rm_resevr.heading("SN_No.", text="Sn No.", anchor=CENTER)
Rm_resevr.heading("Gs_ID", text="Guest ID", anchor=CENTER)
Rm_resevr.heading("Gs_Name", text="Guest Name", anchor=CENTER)
Rm_resevr.heading("Gender", text="Gender", anchor=CENTER)
Rm_resevr.heading("Religion", text="Religion", anchor=CENTER)
Rm_resevr.heading("Address", text="Address", anchor=CENTER)
Rm_resevr.heading("City", text="City", anchor=CENTER)
Rm_resevr.heading("Country", text="Country", anchor=CENTER)
Rm_resevr.heading("Contact_No", text="Contact Number", anchor=CENTER)
Rm_resevr.heading("ID_Type", text="ID Type", anchor=CENTER)
Rm_resevr.heading("ID_No", text="ID No", anchor=CENTER)
Rm_resevr.heading("Email_ID", text="Email ID", anchor=CENTER)
Rm_resevr.heading("Room_No", text="Room No", anchor=CENTER)
Rm_resevr.heading("Room_Type", text="Room Type", anchor=CENTER)
Rm_resevr.heading("Date_In", text="Date In", anchor=CENTER)
Rm_resevr.heading("Date_Out", text="Date Out", anchor=CENTER)
Rm_resevr.heading("Room_Charges", text="Room Charges", anchor=CENTER)
Rm_resevr.heading("Payment", text="Payment", anchor=CENTER)
Rm_resevr.pack(fill=BOTH, expand=1)


Rm_resevr["show"] = "headings"
Rm_resevr.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
Rm_resevr.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
Rm_resevr.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
Rm_resevr.column("Gender", width=130, anchor=CENTER, minwidth=130)
Rm_resevr.column("Religion", width=140, anchor=CENTER, minwidth=140)
Rm_resevr.column("Address", width=140, anchor=CENTER, minwidth=140)
Rm_resevr.column("City", width=140, anchor=CENTER, minwidth=140)
Rm_resevr.column("Country", width=140, anchor=CENTER, minwidth=140)
Rm_resevr.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
Rm_resevr.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("ID_No", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Email_ID", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Room_No", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Room_Type", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Date_In", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Date_Out", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Room_Charges", width=140, anchor=CENTER, minwidth=120)
Rm_resevr.column("Payment", width=140, anchor=CENTER, minwidth=120)
```

```
#TODO
def Rm_resvr_EntryGstry():
   con = connector.connect(host='localhost',
                           port='3306',
                           user='root',
                           password='Password',
                           database='Hotel Management Software')
   cur = con.cursor()
   if ResvrGS_ID.get() != "" and ResvrGS_Nm.get() != "" and ResvrGS_Rm_No.get() != "" and
ResvrGS_Cnt_No.get() != "":
      # print("Worked")
      for item in Rm_resevr.get_children():
         Rm_resevr.delete(item)
      query = f"select * from `Reservation Details` r  right join `customer details` using (`Guest ID`) where
r.`Room No` is not null and r.`Guest ID`= '{ResvrGS_ID.get()}'and r.`Guest Name`=
'{ResvrGS_Nm.get()}'and `Room No`= '{ResvrGS_Rm_No.get()}'and `Contact No.`=
'{ResvrGS_Cnt_No.get()}' ;"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         # print(row)
         Rm_resevr.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15],
            row[16],
            row[17],
            row[18], row[19], row[20]))

   elif ResvrGS_ID.get() != "" and ResvrGS_Nm.get() == "" and ResvrGS_Rm_No.get() == "" and
ResvrGS_Cnt_No.get() == "":
      con = connector.connect(host='localhost',
                              port='3306',
                              user='root',
                              password='Password',
                              database='Hotel Management Software')
      cur = con.cursor()
      # print("Worked")
      for item in Rm_resevr.get_children():
         Rm_resevr.delete(item)
      query = f"select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null and `Guest ID`= '{ResvrGS_ID.get()}';"
      cur.execute(query)
      sn = 1
      for row in cur.fetchall():
         # print(row)
         Rm_resevr.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15],
row[16],
            row[17],
            row[18], row[19], row[20]))
   elif ResvrGS_ID.get() == "" and ResvrGS_Nm.get() != "" and ResvrGS_Rm_No.get() == "" and
ResvrGS_Cnt_No.get() == "":
      # print("Worked")
      for item in Rm_resevr.get_children():
```

```
        Rm_resevr.delete(item)
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    query = f"select * from `Reservation Details` r right join `customer details` using (`Guest ID`) where
r.`Room No` is not null and r.`Guest Name`= '{ResvrGS_Nm.get()}' ;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Rm_resevr.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15],
row[16],
            row[17],
            row[18], row[19], row[20]))
  elif ResvrGS_ID.get() == "" and ResvrGS_Nm.get() == "" and ResvrGS_Rm_No.get() != "" and
ResvrGS_Cnt_No.get() == "":
    # print("Worked")
    for item in Rm_resevr.get_children():
        Rm_resevr.delete(item)
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    query = f"select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null and `Room No`= '{ResvrGS_Rm_No.get()}';"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Rm_resevr.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15],
row[16],
            row[17],
            row[18], row[19], row[20]))
  elif ResvrGS_ID.get() == "" and ResvrGS_Nm.get() == "" and ResvrGS_Rm_No.get() == "" and
ResvrGS_Cnt_No.get() != "":
    # print("Worked")
    for item in Rm_resevr.get_children():
        Rm_resevr.delete(item)
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    query = f"select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null and `Contact No.`= '{ResvrGS_Cnt_No.get()}';"
    cur.execute(query)
```

```
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Rm_resevr.insert("", END, values=(
            sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15],
row[16],
            row[17],
            row[18], row[19], row[20]))

tkinter.Button(can_widget3,
image=Guest_Entry,compound=LEFT,command=Rm_resvr_EntryGstry,text="Show",
anchor=W,font=('Century Gothic', 17, "bold"), fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", borderwidth=5,
cursor="hand2").place(x=520,y=75+58+100)
def Rm_resvr_Entryreset():
    ResvrGS_ID.set("")
    ResvrGS_Nm.set("")
    ResvrGS_Rm_No.set("")
    ResvrGS_Cnt_No.set("")
    for item in Rm_resevr.get_children():
        Rm_resevr.delete(item)
    query = "select * from `Reservation Details`  right join `customer details` using (`Guest ID`) where
`Reservation Details`.`Room No` is not null;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Rm_resevr.insert("", END, values=(
        sn, row[0], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[15], row[16],
row[17],
        row[18], row[19], row[20]))
        sn+=1
tkinter.Button(can_widget3, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Reset",command=Rm_resvr_Entryreset, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=670,y=75+58+100)
def Rm_resvr_EntryAdd():
    from docx2pdf import convert
    from docxtpl import DocxTemplate
    doc = DocxTemplate("Hotel Reservation Confirmation.docx")
    RsvrdMsg = Rm_resevr.item(Rm_resevr.selection())['values']
    # print(RsvrdMsg)
    doc.render(
        {"Name": RsvrdMsg[2],
        "Gs_ID": RsvrdMsg[1],
        "Address": f"{RsvrdMsg[5]+', '+RsvrdMsg[6]}",
        "Room_Number": RsvrdMsg[12],
        "Room_Type": RsvrdMsg[13],
        "Check_In_Date": RsvrdMsg[14],
        "Check_Out_Date": RsvrdMsg[15],
        "Room_Price": RsvrdMsg[16],
        "Payment": RsvrdMsg[17],
        "Ttl": RsvrdMsg[17]+RsvrdMsg[16]})
    doc.save("Reservation Letter.docx")
```

```
    convert(r"D:\python\Project\Reservation Letter.docx", r"D:\python\Project\Reservation_Letter.pdf")
    os.remove(r"D:\python\Project\Reservation Letter.docx")
    os.system("Reservation_Letter.pdf")
tkinter.Button(can_widget3, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Print",command=Rm_resvr_EntryAdd, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=580,y=75+70+200)
def Rm_resvr_Entrycls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)

tkinter.Button(can_widget3, image=Guest_Entry, compound=LEFT,command=Rm_resvr_Entrycls,
fg="Black", width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=720,y=75+70+200)
def Rm_resvr_Entryrm():
    if messagebox.askyesno("Confirmation Letter", "Are You Sure You Want To Send Confirmation Letter"):
        from docx2pdf import convert
        from docxtpl import DocxTemplate
        doc = DocxTemplate("Hotel Reservation Confirmation.docx")
        RsvrdMsg = Rm_resevr.item(Rm_resevr.selection())['values']
        # print(RsvrdMsg)
        doc.render(
            {"Name": RsvrdMsg[2],
             "Gs_ID": RsvrdMsg[1],
             "Address": f"{RsvrdMsg[5] + ', ' + RsvrdMsg[6]}",
             "Room_Number": RsvrdMsg[12],
             "Room_Type": RsvrdMsg[13],
             "Check_In_Date": RsvrdMsg[14],
             "Check_Out_Date": RsvrdMsg[15],
             "Room_Price": RsvrdMsg[16],
             "Payment": RsvrdMsg[17],
             "Ttl": RsvrdMsg[17] + RsvrdMsg[16]})
        doc.save("Reservation Letter.docx")
        convert(r"D:\python\Project\Reservation Letter.docx",
r"D:\python\Project\Reservation_Letter.pdf")
        os.remove(r"D:\python\Project\Reservation Letter.docx")
        # os.system("Reservation_Letter.pdf")
        # RsvrdMsg = Rm_resevr.item(Rm_resevr.selection())['values']
        import smtplib
        from email import encoders
        from email.mime.base import MIMEBase
```

```
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    try:
      connect = smtplib.SMTP('smtp.gmail.com', 587)
      connect.ehlo()
      connect.starttls()
      sender_email = “aakshamhotel@gmail.com”
      sender_passwd = “kmko wohf Irdx gthw”
      connect.login(sender_email, sender_passwd)
      receiver_email = RsvrdMsg[11]
      subject = "Reservation Of Room"
      msg_text = (f"Guest ID :- {RsvrdMsg[1]}\n"
            f"Guest Name :- {RsvrdMsg[2]}\n"
            f"Address :- {RsvrdMsg[5]+', '+RsvrdMsg[6]}\n"
            f"Room No :- {RsvrdMsg[12]}\n"
            f"Room Type :- {RsvrdMsg[13]}\n"
            f"Day In :- {RsvrdMsg[14]}\n"
            f"Day Out :- {RsvrdMsg[15]}\n"
            f"Room Price :- {RsvrdMsg[16]}\n"
            f"Reservation Amount :- {RsvrdMsg[17]}\n"
            f"Thank You For Booking Your Room In Our Hotel, We Will Try To Serve You Best.....\n
Hope You Enjoy")
      message = MIMEMultipart()
      message["From"] = sender_email
      message["To"] = receiver_email
      message["Subject"] = subject
      message["Bcc"] = receiver_email
      message.attach(MIMEText(msg_text, "plain"))
      filename = "Reservation_Letter.pdf"
      with open(filename, "rb") as attachment:
        part = MIMEBase("application", "octet-stream")
        part.set_payload(attachment.read())
      encoders.encode_base64(part)
      part.add_header("Content-Disposition", f"attachment; filename= {filename}", )
      message.attach(part)
      text = message.as_string()
      connect.sendmail(sender_email, receiver_email,text)
      # print("Successfully email✉ sent")
      messagebox.showinfo("Mail","Successfully email✉ sent")
    except Exception as e:
      # print(e)
      messagebox.showerror("Error",e)
    finally:
      connect.quit()
tkinter.Button(can_widget3, image=Guest_Entry, compound=LEFT,command=Rm_resvr_Entryrm,
fg="Black", width=260, activeforeground="black",activebackground="#a8701d", height=30,
text="Confirmation Letter", bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"),
borderwidth=5, cursor="hand2").place(x=580,y=75+120+200)
```


Login
Guest Entry
Add Customer
Check In Reserved Room
Check Out
Reservation List
Hall Reservation
Bar/Resturant Billing
Worker Details
Guest Complaint
Report
About Us
Exit
Add Guest
Guest Id :
H1
Guest Name :
Address :
Contact No :
Id Type :
Id Number :
Email Id :
From Date :
25/12/23
To Date :
25/12/23
Hall No. :
Hall Price :
No Of Days :
Total :
Advance :
Update
Remove
Book
Reset
Print
Close
Mail
Sn No.
Guest ID
Guest Name
Address
Contact Number
ID Type
1
H7
Aditya
Laldiggi
999999999
Pancard

Aaksham Hotel

Your most memorable days await you

Booking Confrmation

Your reservation is confirmed!

Hi {{Name}},

Thank you for choosing to stay at Aaksham Hotel.

We're pleased to confirm your reservation.

Reservation Details

Lead Guest Name:	{{Name}}
Guest ID:	{{Gs_ID}}
Address:	{{Address}}
Room Number :	{{Room_Number}}
Check-In Date:	{{Check_In_Date}}
Check-Out Date:	{{Check_Out_Date}}

Payment Details

Room Charges	${{Room_Price}}
Reservation Charges	${{Payment}}
Total Charge	${{Ttl}}

Cancellation Policy

This booking is non-refundable.

We look forward to welcoming you to Aaksham Hotel.

Got more questions?

Send them to us!

+91-8957488214

aakshamhotel@gmail.com

www.aakshamhotel.com

```
#-----------------------------------------------------Hall ------     ------------------------------------------------------------
------------

can_widget4 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
Hall_Rsvr_bg= ImageTk.PhotoImage(Image.open("./assets/Conrad Bora Bora Nui French Polynesia   Hotel Review Cond Nast.jpg").resize((1585,955)))
can_widget4.create_image(0,0,anchor=NW,image=Hall_Rsvr_bg)

Hall_Rsvr_img = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-20231107_191351.png").resize((1200,850)))
can_widget4.create_image(550,495,image=Hall_Rsvr_img)
can_widget4.place(x=330, y=25)
# customtkinter.CTkLabel(master= can_widget4, text="Hall Reservation", font=('Times New Roman', 30, "bold"),fg_color="black",bg_color="black").place(x=450, y=10)
customtkinter.CTkLabel(master=can_widget4, text="Add Guest",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=120, y=150)

def Hall_Resvr_idd():
   customtkinter.CTkLabel(master=can_widget4,text="Guest Id :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=20+120-70, y=60+50+90)
   customtkinter.CTkLabel(master=can_widget4,text="Guest Name :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=20+120-70, y=105+50+90)
   customtkinter.CTkLabel(master=can_widget4, text="Address :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=140-70, y=200+90)
   customtkinter.CTkLabel(master=can_widget4, text="Contact No :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=140-70, y=245+90)
   customtkinter.CTkLabel(master=can_widget4, text="Id Type :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=140-70, y=290+90)
   customtkinter.CTkLabel(master=can_widget4, text="Id Number :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=140-70, y=335+90)
   customtkinter.CTkLabel(master=can_widget4, text="Email Id :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=140-70, y=335+90+40)
   customtkinter.CTkLabel(master=can_widget4, text="From Date :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-70, y=110+90)
   customtkinter.CTkLabel(master=can_widget4, text="To Date :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-70, y=155+90)
   customtkinter.CTkLabel(master=can_widget4, text="Hall No. :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-70, y=200+90)
   customtkinter.CTkLabel(master=can_widget4, text="Hall Price :",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-70, y=245+90)
   customtkinter.CTkLabel(master=can_widget4, text="No Of Days
```

```
:",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-
70, y=245+90+45)
   customtkinter.CTkLabel(master=can_widget4, text="Total
:",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-
70, y=245+90+45+45)
   customtkinter.CTkLabel(master=can_widget4, text="Advance
:",text_color="Black",fg_color="#fd5471",bg_color="#fd5471",font=('Pristina',25,"bold")).place(x=460-
70, y=245+90+45+45+40)
Hall_Resvr_idd()

Scrh_Hall_by_Gst_Id=StringVar()
Scrh_Hall_by_Gst_Nm=StringVar()
Scrh_Hall_by_Address=StringVar()
Scrh_Hall_by_Cnt_No=StringVar()
Scrh_Hall_by_ID_Type=StringVar()
Scrh_Hall_by_ID_Number=StringVar()
Scrh_Hall_Email_ID=StringVar()
# Hall_Gs=pd.DataFrame(["H","100"])
# Hall_Gs.to_csv("Hall_ID.csv")
Hall_Gs=pd.read_csv("Hall_ID.csv",index_col=[0])
Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_Gst_Id,highlightbackground="grey",hi
ghlightcolor="red",fg="blue",font="consolas 17 italic",state="readonly").place(
   x=210+150-100,y=75+65+110,width=100,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_Gst_Nm,highlightbackground="grey",h
ighlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=135+65+110,width=200,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_Address,highlightbackground="grey",h
ighlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=190+65+110,width=200,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_Cnt_No,highlightbackground="grey",h
ighlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=245+65+110,width=200,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_ID_Type,highlightbackground="grey",
highlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=365+110,width=200,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_by_ID_Number,highlightbackground="gre
y",highlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=420+110,width=200,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Scrh_Hall_Email_ID,highlightbackground="grey",hig
hlightcolor="red",fg="blue",font="consolas 17 italic").place(
   x=210+150-100,y=420+110+55,width=200,height=30)

HallCstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
Hallfilename=0
def Flenm():

   global CstmrImg
   global Hallfilename
   Hallfilename = filedialog.askopenfilename(initialdir="/", title="Select A File",filetypes=(("JPG files",
"*.jpg"), ("All Files", "*.*")))
   CstmrImg = ImageTk.PhotoImage(Image.open(Hallfilename).resize((100, 100)))
   HallImgBtn.configure(image=CstmrImg)
```

```
HallImgBtn=Button(can_widget4,image=HallCstmrImg,relief=RAISED,command=Flenm)
HallImgBtn.place(x=370, y=200)

l=[]
Dict={}
query=f"select * from `Hall`;"
cur.execute(query)
for row in cur.fetchall():
   # print(row)
   l.append(row[0])
   Dict[row[0]]=row[1]
Hall_No = ttk.Combobox(can_widget4, foreground="black", justify=LEFT, font="Calibri 13", width=10,
state='readonly',background="grey", height=10)
Hall_No["value"]=l
Hall_No.place(x=700-70, y=200-5+60+110)
def Hall_No_selected(e):
   for i in Dict:
      if i == Hall_No.get():
         Hall_Prc.set(Dict[i])
   Hall_Nm_Dys.set((Hall_DtOut_cal.get_date() - Hall_DtIn_cal.get_date()).days)
   Hall_Ttl.set(Hall_Prc.get()*int(Hall_Nm_Dys.get()))
   if Hall_Nm_Dys.get()==0:
      Hall_Ttl.set(Hall_Prc.get())
Hall_No.bind('<<ComboboxSelected>>',Hall_No_selected)
Hall_Prc = DoubleVar()
Hall_Prc.set("")
Hall_Nm_Dys = DoubleVar()
Hall_Nm_Dys.set("")
Hall_Ttl = DoubleVar()
Hall_Ttl.set("")
Hall_Adv = DoubleVar()
Hall_Adv.set("")
Entry(can_widget4,highlightthickness=2,textvariable=Hall_Prc,highlightbackground="grey",highlightcolor
="red",fg="blue",font="consolas 17 italic",state="readonly").place(x=700-70, y=260-
9+60+110,width=120,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Hall_Nm_Dys,highlightbackground="grey",highlight
color="red",fg="blue",font="consolas 17 italic",state="readonly").place(
   x=700-70,y=365+55+55,width=120,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Hall_Ttl,highlightbackground="grey",highlightcolor=
"red",fg="blue",font="consolas 17 italic",state="readonly").place(
   x=700-70,y=365+55+55+55,width=120,height=30)
Entry(can_widget4,highlightthickness=2,textvariable=Hall_Adv,highlightbackground="grey",highlightcolor
="red",fg="blue",font="consolas 17 italic").place(
   x=700-70,y=420+55+55+55,width=120,height=30)
Hall_DtIn_cal = DateEntry(can_widget4, selectmode="day", font=("Cambria", 13,
"italic"),date_pattern='dd/mm/yy',foreground="blue", width=10)
Hall_DtIn_cal.place(x=700-70, y=140+110)
Hall_DtOut_cal = DateEntry(can_widget4, selectmode="day", font=("Cambria", 13,
"italic"),date_pattern='dd/mm/yy',foreground="blue", width=10)
Hall_DtOut_cal.place(x=700-70,y=200-5+115)

Hall_resevr = Frame(can_widget4, relief=SUNKEN, borderwidth=4)
Hall_resevr.place(x=175-90, y=630, width=800, height=170)
Hall_scbr_x = Scrollbar(Hall_resevr, orient=HORIZONTAL)
```

```
Hall_scbr_y = Scrollbar(Hall_resevr, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Hall_resevr_TreeYu = ttk.Treeview(Hall_resevr, cursor="hand2", columns=("SN_No.", "Gs_ID",
"Gs_Name","Address","Contact_No","ID_Type","ID_No","Email_ID",'Image',"Hall_No","Hall_Pri","
Frm_Dt","To_Dt",'Ttl','Adv'),
                    selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

Hall_scbr_x.pack(side=BOTTOM, fill=X)
Hall_scbr_y.pack(side=RIGHT, fill=Y)
Hall_scbr_x.config(command=Hall_resevr_TreeYu.xview)
Hall_scbr_y.config(command=Hall_resevr_TreeYu.yview)
Hall_resevr_TreeYu.heading("SN_No.", text="Sn No.", anchor=CENTER)
Hall_resevr_TreeYu.heading("Gs_ID", text="Guest ID", anchor=CENTER)
Hall_resevr_TreeYu.heading("Gs_Name", text="Guest Name", anchor=CENTER)
Hall_resevr_TreeYu.heading("Address", text="Address", anchor=CENTER)
Hall_resevr_TreeYu.heading("Contact_No", text="Contact Number", anchor=CENTER)
Hall_resevr_TreeYu.heading("ID_Type", text="ID Type", anchor=CENTER)
Hall_resevr_TreeYu.heading("ID_No", text="ID No", anchor=CENTER)
Hall_resevr_TreeYu.heading("Email_ID", text="Email ID", anchor=CENTER)
Hall_resevr_TreeYu.heading("Image", text="Image", anchor=CENTER)
Hall_resevr_TreeYu.heading("Hall_No", text="Hall No", anchor=CENTER)
Hall_resevr_TreeYu.heading("Hall_Pri", text="Hall Price", anchor=CENTER)
Hall_resevr_TreeYu.heading("Frm_Dt", text="From Date", anchor=CENTER)
Hall_resevr_TreeYu.heading("To_Dt", text="To Date", anchor=CENTER)
Hall_resevr_TreeYu.heading("Ttl", text="Total", anchor=CENTER)
Hall_resevr_TreeYu.heading("Adv", text="Advance", anchor=CENTER)
Hall_resevr_TreeYu.pack(fill=BOTH, expand=1)



Hall_resevr_TreeYu["show"] = "headings"
Hall_resevr_TreeYu.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
Hall_resevr_TreeYu.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
Hall_resevr_TreeYu.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
Hall_resevr_TreeYu.column("Address", width=140, anchor=CENTER, minwidth=140)
Hall_resevr_TreeYu.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
Hall_resevr_TreeYu.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("ID_No", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Email_ID", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Image", width=0, anchor=CENTER, minwidth=0)
Hall_resevr_TreeYu.column("Hall_No", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Hall_Pri", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Frm_Dt", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("To_Dt", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Ttl", width=140, anchor=CENTER, minwidth=120)
Hall_resevr_TreeYu.column("Adv", width=140, anchor=CENTER, minwidth=120)
sn=1
query=f"select * from `Hall Reservation`;"
cur.execute(query)
for row in cur.fetchall():
   Hall_resevr_TreeYu.insert("", END, values=(
   sn, row[0],row[1], row[2], row[3], row[4], row[5], row[6],row[12],
row[7],row[8],row[9].strftime("%d/%m/%y"),row[10].strftime("%d/%m/%y"),row[13],row[14]))
```

```
    sn+=1
HallNumber=0
HallPrice=0
def Hall_resevr_selected(_):
    global HallCstmrImg
    global HallNumber
    global HallPrice
    global Hallfilename
    Hall_Data=Hall_resevr_TreeYu.item(Hall_resevr_TreeYu.selection())['values']
    # print(Hall_Data)
    Scrh_Hall_by_Gst_Id.set(Hall_Data[1])
    Scrh_Hall_by_Gst_Nm.set(Hall_Data[2])
    Scrh_Hall_by_Address.set(Hall_Data[3])
    Scrh_Hall_by_Cnt_No.set(Hall_Data[4])
    Scrh_Hall_by_ID_Type.set(Hall_Data[5])
    Scrh_Hall_by_ID_Number.set(Hall_Data[6])
    Scrh_Hall_Email_ID.set(Hall_Data[7])
    Hall_No.set(Hall_Data[9])
    Hall_Prc.set(Hall_Data[10])
    Hall_DtIn_cal.set_date(Hall_Data[11])
    Hall_DtOut_cal.set_date(Hall_Data[12])
    Hall_Nm_Dys.set((Hall_DtOut_cal.get_date()-Hall_DtIn_cal.get_date()).days)
    Hall_Ttl.set(Hall_Data[13])
    Hall_Adv.set(Hall_Data[14])
    HallCstmrImg = ImageTk.PhotoImage(Image.open(Hall_Data[8]).resize((100, 100)))
    HallImgBtn.configure(image=HallCstmrImg)
    Hallfilename=Hall_Data[8]
    HallNumber=Hall_Data[9]
    HallPrice=Hall_Data[10]

Hall_resevr_TreeYu.bind('<<TreeviewSelect>>',Hall_resevr_selected)

def Hall_resvr_Update():
    if messagebox.askyesno("Update Customer Details", "Are You Sure You Want To Update Details"):
        global Hallfilename
        if HallNumber != Hall_No.get():
            query = f"insert into Hall values ('{HallNumber}','{HallPrice}')"
            cur.execute(query)
            con.commit()
            query = f"delete from Hall where `Hall No` = '{Hall_No.get()}';"
            cur.execute(query)
            con.commit()
        if Hallfilename == 0:
            Hallfilename="passportsizephoto.webp"
        query = f"update `Hall Reservation` set `Guest ID`='{Scrh_Hall_by_Gst_Id.get()}' ,`Guest
Name`='{Scrh_Hall_by_Gst_Nm.get()}',`Address`='{Scrh_Hall_by_Address.get()}',`Contact
No.`='{Scrh_Hall_by_Cnt_No.get()}',`ID Type`='{Scrh_Hall_by_ID_Type.get()}',`ID
Number`='{Scrh_Hall_by_ID_Number.get()}',`Email ID`='{Scrh_Hall_Email_ID.get()}',`Hall
Number`='{Hall_No.get()}',`Hall Price`='{Hall_Prc.get()}',`From
Date`='{Hall_DtIn_cal.get_date().strftime('%Y-%m-%d')}',`To
Date`='{Hall_DtOut_cal.get_date().strftime('%Y-%m-
%d')}',`Image`='{Hallfilename}',`Total`='{Hall_Ttl.get()}',`Advanve`='{Hall_Adv.get()}' where `Guest
ID`='{Scrh_Hall_by_Gst_Id.get()}';"
        # print(query)
```

```
    cur.execute(query)
    con.commit()
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm .set("")
    Scrh_Hall_by_Address .set("")
    Scrh_Hall_by_Cnt_No .set("")
    Scrh_Hall_by_ID_Type .set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
      Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
      Hall_resevr_TreeYu.insert("", END, values=(
        sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
        row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
      sn += 1
tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Update",command=Hall_resvr_Update, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=770, y=250+30)

def Hall_resvr_Remove():
  if messagebox.askyesno("Remove Details", "Are You Sure You Want To Remove Details"):
    global Dict
    global Hall_No
    con = connector.connect(host='localhost',
                    port='3306',
                    user='root',
                    password='Password',
                    database='Hotel Management Software')
    cur = con.cursor()
    Hall_Data=Hall_resevr_TreeYu.item(Hall_resevr_TreeYu.selection())['values']
    query = f"insert into hall values ('{Hall_Data[9]}','{Hall_Data[10]}');"
    cur.execute(query)
    con.commit()
    query=f"delete from `Hall Reservation` where Hall No` = '{Hall_No.get()}';"
    cur.execute(query)
    # con.commit()
    Dict = {}
    query = f"select * from `Hall`;"
```

```
    cur.execute(query)
    l=[]
    for row in cur.fetchall():
        # print(row)
        l.append(row[0])
        Dict[row[0]] = row[1]
    Hall_No["value"] = 1
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm .set("")
    Scrh_Hall_by_Address .set("")
    Scrh_Hall_by_Cnt_No .set("")
    Scrh_Hall_by_ID_Type .set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
        Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
        Hall_resevr_TreeYu.insert("", END, values=(
            sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
        sn += 1
tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Remove",command=Hall_resvr_Remove, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=770, y=300+30)

def Hall_resvr_Book():
  if messagebox.askyesno("Reserved Hall", "Are You Sure You Want To Rrserved Hall"):
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    query='desc `Hall Reservation`;'
    cur.execute(query)
    if len(str(Hallfilename)) > int(cur.fetchall()[12][1][8:-1]):
        query=f'alter table `Hall Reservation` modify column Image varchar({len(str(filename))});'
        cur.execute(query)
        con.commit()
```

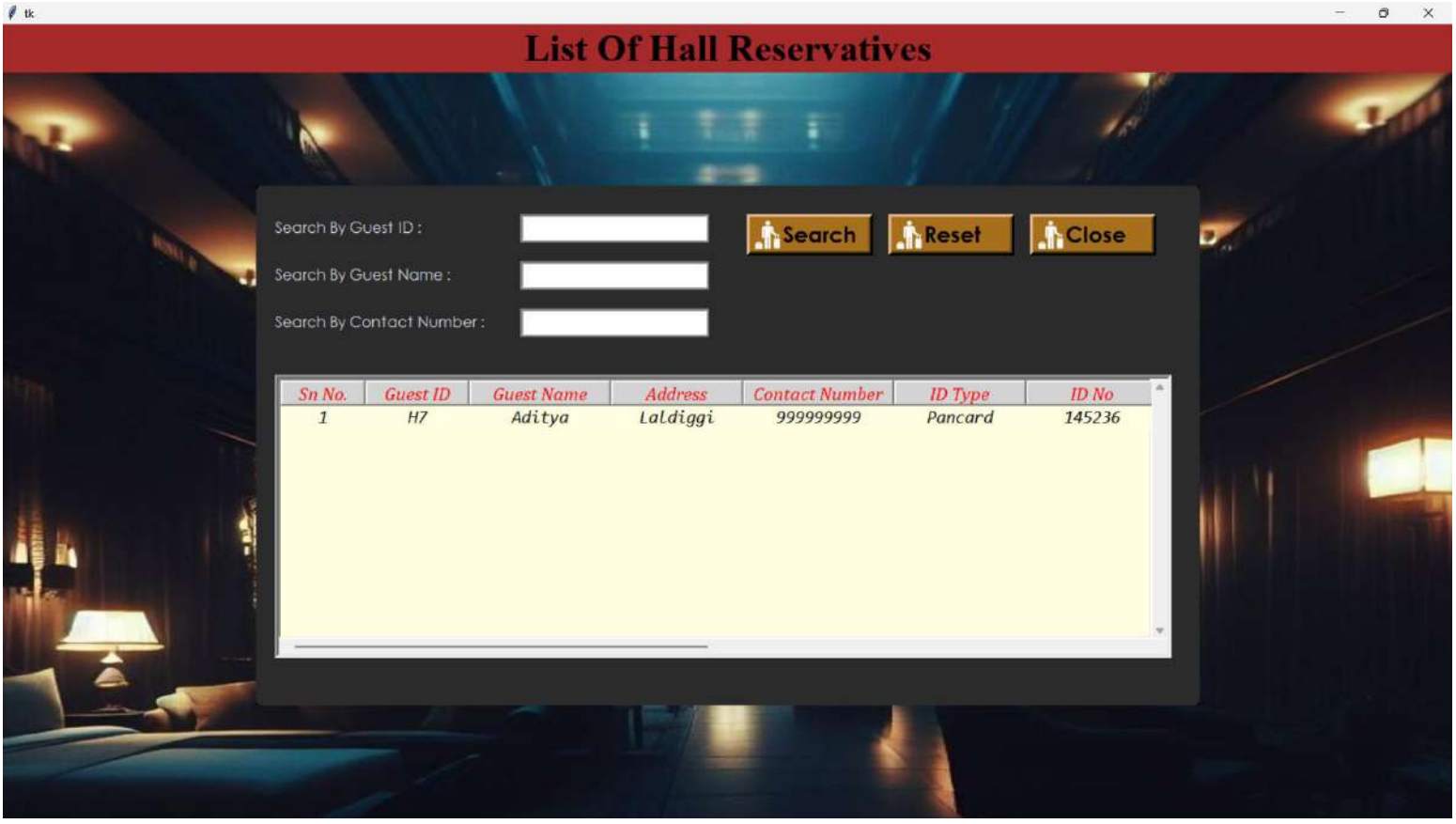
```
    hin=Hall_DtIn_cal.get_date().strftime("%Y-%m-%d")
    hout=Hall_DtOut_cal.get_date().strftime("%Y-%m-%d")
    query=f"insert into `Hall Reservation`
values('{Scrh_Hall_by_Gst_Id.get()}','{Scrh_Hall_by_Gst_Nm.get()}','{Scrh_Hall_by_Address.get()}','{Scrh_
Hall_by_Cnt_No.get()}','{Scrh_Hall_by_ID_Type.get()}','{Scrh_Hall_by_ID_Number.get()}','{Scrh_Hall_Em
ail_ID.get()}','{Hall_No.get()}','{Hall_Prc.get()}','{hin}','{hout}','{date.today()}','{Hallfilename}','{Hall_Ttl.
get()}','{Hall_Adv.get()}');"
    # print(query)
    cur.execute(query)
    con.commit()
    Hall_Gs = pd.read_csv("Hall_ID.csv",index_col=[0])
    Hall_Gs.Hall[1]=int(Hall_Gs.Hall[1])+1
    Hall_Gs.to_csv("Hall_ID.csv")
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    for i in Hall_resevr_TreeYu.get_children():
      Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
      Hall_resevr_TreeYu.insert("", END, values=(
        sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
        row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
      sn += 1
    l = []
    query = f"delete from `Hall` where `Hall No`='{Hall_No.get()}';"
    cur.execute(query)
    con.commit()
    query = f"select * from `Hall`;"
    cur.execute(query)
    for row in cur.fetchall():
      l.append(row[0])
    Hall_No["value"] = 1
    Hall_No.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm .set("")
    Scrh_Hall_by_Address .set("")
    Scrh_Hall_by_Cnt_No .set("")
    Scrh_Hall_by_ID_Type .set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Nm_Dys.set("")
    Hall_Prc.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
tkinter.Button(can_widget4,
image=Guest_Entry,compound=LEFT,command=Hall_resvr_Book,text="Book",
anchor=W,font=('Century Gothic', 17, "bold"), fg="Black", width=120,
```

```
activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", borderwidth=5,
cursor="hand2").place(x=770, y=350+30)
def Hall_resvr_Entryreset():
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm .set("")
    Scrh_Hall_by_Address .set("")
    Scrh_Hall_by_Cnt_No .set("")
    Scrh_Hall_by_ID_Type .set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
        Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
        Hall_resevr_TreeYu.insert("", END, values=(
            sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
        sn += 1
tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Reset",command=Hall_resvr_Entryreset, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=770, y=400+30)
def Hall_resvr_Print():
    from docx2pdf import convert
    from docxtpl import DocxTemplate
    doc = DocxTemplate("Hall Reservation.docx")
    HallMsg = Hall_resevr_TreeYu.item(Hall_resevr_TreeYu.selection())['values']
    # print(HallMsg)
    doc.render({"Name": HallMsg[2],
        "Gs_ID": HallMsg[1],
        "Address":HallMsg[3],
        "Room_Number": HallMsg[9],
        "Check_In_Date": HallMsg[11],
        "Check_Out_Date": HallMsg[12],
        "Room_Price": HallMsg[10],
        "Payment": HallMsg[14],
        "Ttl": HallMsg[13]})
    # doc.save("Hall Reservation Letter.docx")
    convert(r"D:\python\Project\Hall Reservation Letter.docx",
r"D:\python\Project\Hall_Reservation_Letter.pdf")
    os.remove(r"D:\python\Project\Hall Reservation Letter.docx")
```

```
    os.system("Hall_Reservation_Letter.pdf")
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm.set("")
    Scrh_Hall_by_Address.set("")
    Scrh_Hall_by_Cnt_No.set("")
    Scrh_Hall_by_ID_Type.set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
        Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
        Hall_resevr_TreeYu.insert("", END, values=(
            sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
        sn += 1
tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Print",command=Hall_resvr_Print, bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"),
borderwidth=5, cursor="hand2").place(x=770, y=480)
def Hall_resvr_Entrycls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)

tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT,command=Hall_resvr_Entrycls,
fg="Black", width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=770, y=480+50)
def Hall_resvr_cnf():
```

```
  if messagebox.askyesno("Mail", "Are You Sure You Want To Mail"):
    from docx2pdf import convert
    from docxtpl import DocxTemplate
    doc = DocxTemplate("Hall Reservation.docx")
    HallMsg = Hall_resevr_TreeYu.item(Hall_resevr_TreeYu.selection())['values']
    # print(HallMsg)
    doc.render({"Name": HallMsg[2],
            "Gs_ID": HallMsg[1],
            "Address": HallMsg[3],
            "Room_Number": HallMsg[9],
            "Check_In_Date": HallMsg[11],
            "Check_Out_Date": HallMsg[12],
            "Room_Price": HallMsg[10],
            "Payment": HallMsg[14],
            "Ttl": HallMsg[13]})
    doc.save("Hall Reservation Letter.docx")
    convert(r"D:\python\Project\Hall Reservation Letter.docx",
r"D:\python\Project\Hall_Reservation_Letter.pdf")
    os.remove(r"D:\python\Project\Hall Reservation Letter.docx")
    # os.system("Hall_Reservation_Letter.pdf")
    # print(HallMsg)
    import smtplib
    from email import encoders
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    try:
      connect = smtplib.SMTP('smtp.gmail.com', 587)
      connect.ehlo()
      connect.starttls()
      sender_email = "aakshamhotel@gmail.com"
      sender_passwd = "kmko wohf Irdx gthw"
      connect.login(sender_email, sender_passwd)
      receiver_email = HallMsg[7]
      subject = "Hall Reservation"
      msg_text = (f"Guest ID :- {HallMsg[1]}\n"
            f"Guest Name :- {HallMsg[2]}\n"
            f"Address :- {HallMsg[3]}\n"
            f"Contact No. :- {HallMsg[4]}\n"
            f"Hall No :- {HallMsg[9]}\n"
            f"Hall Price :- {HallMsg[10]}\n"
            f"Day In :- {HallMsg[11]}\n"
            f"Day Out :- {HallMsg[12]}\n"
            f"Total Price :- {HallMsg[13]}\n"
            f"Reservation Amount :- {HallMsg[14]}\n"
            f"Thank You For Booking Hall In Our Hotel, We Will Try To Serve You Best……\nHope
You Enjoy")
      # print(msg_text)
      message = MIMEMultipart()
      message["From"] = sender_email
      message["To"] = receiver_email
      message["Subject"] = subject
      message["Bcc"] = receiver_email
      message.attach(MIMEText(msg_text, "plain"))
```

```
        filename = "Hall_Reservation_Letter.pdf"
        with open(filename, "rb") as attachment:
            part = MIMEBase("application", "octet-stream")
            part.set_payload(attachment.read())
        message.attach(part)
        encoders.encode_base64(part)
        text = message.as_string()
        connect.sendmail(sender_email, receiver_email,text)
        # print("Successfully email✉ sent")
        messagebox.showinfo("Email Send","Successfully email✉ sent")
    except Exception as e:
        messagebox.showerror("Error",e)
    finally:
        connect.quit()
    Hall_No.set("")
    Hall_Nm_Dys.set("")
    Hall_Gs = pd.read_csv("Hall_ID.csv", index_col=[0])
    Scrh_Hall_by_Gst_Id.set(value=f"{Hall_Gs.Hall[0]}{Hall_Gs.Hall[1]}")
    Scrh_Hall_by_Gst_Nm.set("")
    Scrh_Hall_by_Address.set("")
    Scrh_Hall_by_Cnt_No.set("")
    Scrh_Hall_by_ID_Type.set("")
    Scrh_Hall_by_ID_Number.set("")
    Scrh_Hall_Email_ID.set("")
    Hall_Ttl.set("")
    Hall_Adv.set("")
    Hall_Prc.set("")
    Hall_DtIn_cal.set_date(date.today())
    Hall_DtOut_cal.set_date(date.today())
    global CstmrImg
    CstmrImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 100)))
    HallImgBtn.configure(image=CstmrImg)
    for i in Hall_resevr_TreeYu.get_children():
        Hall_resevr_TreeYu.delete(i)
    sn = 1
    query = f"select * from `Hall Reservation`;"
    cur.execute(query)
    for row in cur.fetchall():
        Hall_resevr_TreeYu.insert("", END, values=(
            sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[12], row[7], row[8],
            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[13], row[14]))
        sn += 1
tkinter.Button(can_widget4, image=Guest_Entry, compound=LEFT,command=Hall_resvr_cnf, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Mail",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=770, y=480+50+50)
```



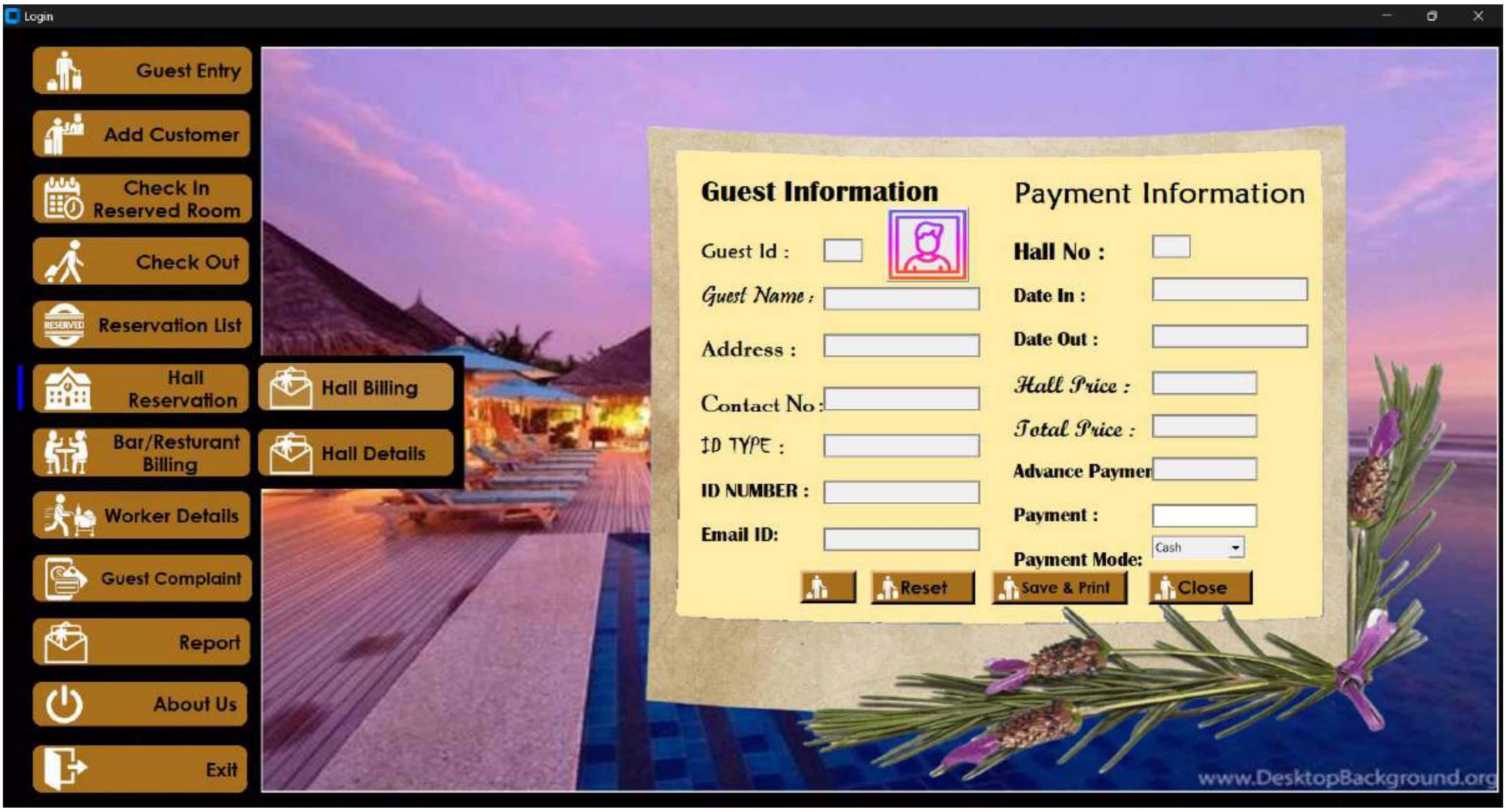

```
import tkinter
import customtkinter
from tkinter import  ttk
from tkinter import *
from tkinter import  messagebox
from PIL import ImageTk,Image
import mysql.connector as connector
import pandas as pd
con=connector.connect(host='localhost',
                      port='3306',
                      user='root',
                      password='Password',
                      database='Hotel Management Software')
cur=con.cursor()
from tkcalendar import DateEntry
from datetime import date
import os
height = 730
width = 1200
root=Tk()
x = (root.winfo_screenwidth()//2)-(width//2)
y = (root.winfo_screenheight()//2)-(height//2)
root.geometry('{}x{}+{}+{}'.format(width, height, x, y-30))

Label(text="List Of Hall Reservatives",background="brown",font=('Times New
Roman',30,"bold")).pack(anchor=N,fill=X)

img1 = ImageTk.PhotoImage(Image.open("./assets/3.jpg"))
l1 = customtkinter.CTkLabel(master=root,text="",image=img1)
l1.pack(fill=BOTH,anchor=N)

frame = customtkinter.CTkFrame(master=l1,width=1000,height=550,bg_color="black")
```

```
frame.place(relx=0.5,rely=0.5,anchor=tkinter.CENTER)

Guest_Entry = ImageTk.PhotoImage(Image.open("./images/Guest_Entry.png").resize((30,30)))

customtkinter.CTkLabel(master=frame, text="Search By Guest ID :", font=('Century Gothic',
16)).place(x=20, y=30)
customtkinter.CTkLabel(master=frame, text="Search By Guest Name :", font=('Century Gothic',
16)).place(x=20, y=80)
customtkinter.CTkLabel(master=frame, text="Search By Contact Number :", font=('Century Gothic',
16)).place(x=20, y=130)
E1Var=StringVar()
E1Var.set("")
E2Var=StringVar()
E2Var.set("")
E3Var=StringVar()
E3Var.set("")
E1=Entry(frame,highlightthickness=2,textvariable=E1Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E1.place(x=280,y=30,width=200,height=30)
E2=Entry(frame,highlightthickness=2,textvariable=E2Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E2.place(x=280,y=80,width=200,height=30)
E3=Entry(frame,highlightthickness=2,textvariable=E3Var,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic")
E3.place(x=280,y=130,width=200,height=30)
def Scrh():
    if E1.get()==""and E2.get()=="":
        for item in table.get_children():
            table.delete(item)
        sn = 1
        query = f"select * from `Hall Reservation` where `Contact No.`='{E3.get()}';"
        cur.execute(query)
        for row in cur.fetchall():
            table.insert("", END, values=(
                sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
row[9].strftime('%d/%m/%Y'),
                row[10].strftime('%d/%m/%Y'), row[12], row[13], row[14]))
            sn += 1
    elif E2.get()==""and E3.get()=="":
        for item in table.get_children():
            table.delete(item)
        sn = 1
        query = f"select * from `Hall Reservation` where `Guest ID`='{E1.get()}';"
        cur.execute(query)
        for row in cur.fetchall():
            # print(row)
            table.insert("", END, values=(
                sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
row[9].strftime('%d/%m/%Y'),
                row[10].strftime('%d/%m/%Y'), row[12], row[13], row[14]))
            sn += 1
    elif E1.get()==""and E3.get()=="":
        for item in table.get_children():
            table.delete(item)
```

```
    sn = 1
    query = f"select * from `Hall Reservation` where `Guest Name`='{E2.get()}';"
    cur.execute(query)
    for row in cur.fetchall():
      table.insert("", END, values=(
        sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
row[9].strftime('%d/%m/%Y'),
        row[10].strftime('%d/%m/%Y'), row[12], row[13], row[14]))
      sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Scrh, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Search", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=500+20, y=30)
def Rset():
  E1Var.set("")
  E2Var.set("")
  E3Var.set("")
  for item in table.get_children():
    table.delete(item)
  sn = 1
  query = f"select * from `Hall Reservation`;"
  cur.execute(query)
  for row in cur.fetchall():
    # print(row)
    table.insert("", END, values=(
      sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
row[9].strftime('%d/%m/%Y'),
      row[10].strftime('%d/%m/%Y'), row[12], row[13], row[14]))
    sn += 1
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=Rset, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Reset", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650+20, y=30)
# tkinter.Button(frame, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Excel", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=650, y=30)
def cls():
  fu = ["","","","","","","","","","","","","","passportsizephoto.webp","",""]
  so = pd.Series(data=fu, name="Hall")
  sep = pd.DataFrame(so)
  sep.to_csv("Hall_List.csv")
  root.destroy()
tkinter.Button(frame, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d",
anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=800+20, y=30)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
```

```
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

frm1 = Frame(frame, relief=SUNKEN, borderwidth=4)
frm1.place(x=20, y=200, width=950, height=300)
scbr_x = Scrollbar(frm1, orient=HORIZONTAL)
scbr_y = Scrollbar(frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
table = ttk.Treeview(frm1, cursor="hand2", columns=("SN_No.", "Gs_ID",
"Gs_Name","Address","Contact_No","ID_Type","ID_No","Email_ID","Hall_No","Hall_Pri","Frm_Dt
","To_Dt","Image","Ttl","Adv"),
                     selectmode="browse", xscrollcommand=scbr_x.set, yscrollcommand=scbr_y.set)

scbr_x.pack(side=BOTTOM, fill=X)
scbr_y.pack(side=RIGHT, fill=Y)
scbr_x.config(command=table.xview)
scbr_y.config(command=table.yview)
table.heading("SN_No.", text="Sn No.", anchor=CENTER)
table.heading("Gs_ID", text="Guest ID", anchor=CENTER)
table.heading("Gs_Name", text="Guest Name", anchor=CENTER)
table.heading("Address", text="Address", anchor=CENTER)
table.heading("Contact_No", text="Contact Number", anchor=CENTER)
table.heading("ID_Type", text="ID Type", anchor=CENTER)
table.heading("ID_No", text="ID No", anchor=CENTER)
table.heading("Email_ID", text="Email ID", anchor=CENTER)
table.heading("Hall_No", text="Hall No", anchor=CENTER)
table.heading("Hall_Pri", text="Hall Price", anchor=CENTER)
table.heading("Frm_Dt", text="From Date", anchor=CENTER)
table.heading("To_Dt", text="To Date", anchor=CENTER)
table.heading("Image", text="Image", anchor=CENTER)
table.heading("Ttl", text="Total", anchor=CENTER)
table.heading("Adv", text="Advance", anchor=CENTER)
table.pack(fill=BOTH, expand=1)


table["show"] = "headings"
table.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
table.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
table.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
table.column("Address", width=140, anchor=CENTER, minwidth=140)
table.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
table.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
table.column("ID_No", width=140, anchor=CENTER, minwidth=120)
table.column("Email_ID", width=140, anchor=CENTER, minwidth=120)
table.column("Hall_No", width=140, anchor=CENTER, minwidth=120)
table.column("Hall_Pri", width=140, anchor=CENTER, minwidth=120)
table.column("Frm_Dt", width=140, anchor=CENTER, minwidth=120)
table.column("To_Dt", width=140, anchor=CENTER, minwidth=120)
table.column("Image", width=0, anchor=CENTER, minwidth=0)
table.column("Ttl", width=140, anchor=CENTER, minwidth=120)
```

```
table.column("Adv", width=140, anchor=CENTER, minwidth=120)
sn=1
query=f"select * from `Hall Reservation`;"
cur.execute(query)
for row in cur.fetchall():
    # print(row)
    table.insert("", END, values=(
    sn, row[0],row[1], row[2], row[3], row[4], row[5], row[6], row[7],
row[8],row[9],row[10],row[12],row[13],row[14]))
    sn+=1
def table_select(_):
    fu=[]
    for i in table.selection():
        fu.append(table.item(i)['values'])
    so = pd.Series(data=fu[0], name="Hall")
    sep = pd.DataFrame(so)
    sep.to_csv("Hall_List.csv")
    root.destroy()
table.bind('<<TreeviewSelect>>',table_select)

root.mainloop()
```




```
#-------------------------------------------------------------    Hall Billing    ---------------------------------------------------
-------------

can_widget6 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
Hallbg = ImageTk.PhotoImage(Image.open("./assets/1deaJz.jpg").resize((1585,955)))
Hallimg = ImageTk.PhotoImage(Image.open("./assets/Bar.png").resize((1050,850)))
can_widget6.create_image(0,0,anchor=NW,image=Hallbg)
can_widget6.create_image(1010,520,image=Hallimg)
# can_widget6.place(x=330, y=25)
customtkinter.CTkLabel(master=can_widget6, text="Guest
```

```
Information",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
30)).place(x=10+120+320, y=10+50+70)
customtkinter.CTkLabel(master=can_widget6, text="Payment
Information",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=("Maiandra GD", 30,
"bold")).place(x=500+120+150, y=10+50+70)
def idd():
   customtkinter.CTkLabel(master=can_widget6,text="Guest Id
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Maiandra GD', 20,
"bold")).place(x=20+130+300, y=60+40+160-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="Guest Name
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Pristina',25,"bold")).place(x=20+13
0+300, y=105+40+160-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="Address
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Garamond',25,'bold')).place(x=20+1
30+300, y=240+40+80-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="Contact No
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Poor
Richard',25,'bold')).place(x=20+130+300, y=375+40-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="ID TYPE
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Chiller',25,'bold')).place(x=20+130+
300, y=420+40-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="ID NUMBER
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Britannic
Bold',20)).place(x=20+130+300, y=465+40-35-30)
   customtkinter.CTkLabel(master=can_widget6,text="Email
ID:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black",font=('Britannic
Bold',20)).place(x=20+130+300, y=465+40-35-30+45)

   #-------------------------------------------------------------------------------------------
   customtkinter.CTkLabel(master=can_widget6, text="Hall No
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
25)).place(x=510+130+130, y=40+105+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Date In
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
20)).place(x=510+130+130, y=190+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Date Out
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
20)).place(x=510+130+130, y=235+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Hall Price
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Script MT Bold',
25)).place(x=510+130+130, y=280+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Total Price
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Script MT Bold',
25)).place(x=510+130+130, y=325+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Advance Payment
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
20)).place(x=510+130+130, y=370+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Payment
:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=('Britannic Bold',
20)).place(x=510+130+130,y=415+80-30)
   customtkinter.CTkLabel(master=can_widget6, text="Payment
Mode:",fg_color="#fdeca6",bg_color="#fdeca6",text_color="black", font=("Britannic Bold",
20)).place(x=510+130+130, y=415+80-30+45)
idd()
```

```
Hallgd_Gst_GsID=StringVar()
Hallgd_Gst_GsNm=StringVar()
Hallgd_Gst_GsAddress=StringVar()
Hallgd_Gst_GsCntNO=StringVar()
Hallgd_Gst_GsIDType=StringVar()
Hallgd_Gst_GsIDNo=StringVar()
Hallgd_Gst_GsEmil_ID=StringVar()
dtin=StringVar()
dtout=StringVar()
Hallgd_Gst_GsID.set("")
Hallgd_Gst_GsNm.set("")
Hallgd_Gst_GsAddress.set("")
Hallgd_Gst_GsCntNO.set("")
Hallgd_Gst_GsIDType.set("")
Hallgd_Gst_GsIDNo.set("")
Hallgd_Gst_GsEmil_ID.set("")
dtin.set("")
dtout.set("")
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsID,highlightbackground="grey",highli
ghtcolor="black",fg="blue",font="'Britannic' 17
italic",state="readonly").place(x=140+200+350+30,y=65+65+200-45-40,width=50,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsNm,highlightbackground="grey",high
lightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+350+30,y=65+105+12+200-45-30,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsAddress,highlightbackground="grey",
highlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+350+30,y=65+240+47+100-45-40,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsCntNO,highlightbackground="grey",h
ighlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+350+30,y=65+375+80-45-40,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsIDType,highlightbackground="grey",
highlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+350+30,y=65+420+95-45-40,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsIDNo,highlightbackground="grey",hi
ghlightcolor="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=140+200+350+30,
y=640-45-40,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_GsEmil_ID,highlightbackground="grey",
highlightcolor="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=140+200+350+30,
y=640-45-40+60,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=dtin,highlightbackground="grey",highlightcolor="bla
ck",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+350+100, y=70+140+25+150-50-
40,width=200,height=30)
Entry(can_widget6,highlightthickness=2,textvariable=dtout,highlightbackground="grey",highlightcolor="bl
ack",fg="blue",font="consolas 17 italic",state="readonly").place(x=690+350+100, y=70+180+45+150-50-
40,width=200,height=30)


Hallgd_Gst_Rm=StringVar()
Entry(can_widget6,textvariable=Hallgd_Gst_Rm,highlightthickness=2,highlightbackground="grey",highlig
htcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=690+350+100,y=130+150+50-50-40,width=50,height=30)
Hallgd_Gst_rm_price=StringVar()
Entry(can_widget6,highlightthickness=2,state="readonly",textvariable=Hallgd_Gst_rm_price,highlightback
ground="grey",highlightcolor="black",fg="blue",font="consolas 17
```

```
italic").place(x=690+350+100,y=270+20+65+150-50-40,width=135,height=30)
Hallgd_Gst_Ttl_Price=DoubleVar()
Hallgd_Gst_Ttl=Entry(can_widget6,highlightthickness=2,state="readonly",textvariable=Hallgd_Gst_Ttl_Pr
ice,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
Hallgd_Gst_Ttl.place(x=690+350+100,y=270+60+80+150-50-40,width=135,height=30)
Hallgd_Gst_Adv_Price=DoubleVar()
Hallgd_Gst_Adv=Entry(can_widget6,highlightthickness=2,state="readonly",textvariable=Hallgd_Gst_Adv_
Price,highlightbackground="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
Hallgd_Gst_Adv.place(x=690+350+100, y=370+95+150-50-40,width=135,height=30)
Hallgd_Gst_Pymnt=DoubleVar()
Hallgd_Gst_en=Entry(can_widget6,highlightthickness=2,textvariable=Hallgd_Gst_Pymnt,highlightbackgrou
nd="grey",highlightcolor="black",fg="blue",font="consolas 17 italic")
Hallgd_Gst_en.place(x=690+350+100, y=370+95+150-50-40+60,width=135,height=30)
#-----------------------------------------------------------------------------------------
Hallgd_Gst_mydata = ttk.Combobox(can_widget6, foreground="black", justify=LEFT, font="Calibri 13",
width=10, state='readonly',background="grey", height=10)
Hallgd_Gst_mydata["value"]=["Cash","UPI","Debit Card","Credit Card","Net Banking"]
Hallgd_Gst_mydata.set("Cash")
Hallgd_Gst_mydata.place(x=690+350+100, y=70+340+110+150-50-40+45)
# #----------------------------------------------------------------------------------------------
Hallgd_Gst_frm = Frame(can_widget6, relief=SUNKEN, borderwidth=4)
# Hallgd_Gst_frm.place(x=700, y=700, width=480, height=200)
Hallgd_Gst_scbr_x = Scrollbar(Hallgd_Gst_frm, orient=HORIZONTAL)
Hallgd_Gst_scbr_y = Scrollbar(Hallgd_Gst_frm, orient=VERTICAL)

HallCstmrBillImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 90)))
HallBillfilename=0
def BillFlenm():
   global HallCstmrBillImg
   global HallBillfilename
   HallBillfilename = filedialog.askopenfilename(initialdir="/", title="Select A File",filetypes=(("JPG
files", "*.jpg"), ("All Files", "*.*")))
   HallCstmrBillImg= ImageTk.PhotoImage(Image.open(HallBillfilename).resize((100, 100)))
   HallBillImgBtn.configure(image=HallCstmrBillImg)
HallBillImgBtn=Button(can_widget6,image=HallCstmrBillImg,relief=RAISED,command=BillFlenm)
HallBillImgBtn.place(x=800, y=205)
HallBillCstmrImg="passportsizephoto.webp"
def Hallgd_Gst_chkin():
   global HallCstmrBillImg
   global HallBillCstmrImg
   os.system("python Hall_Billing.py")
   BillData = pd.read_csv("Hall_List.csv", index_col=[0])
   # print(BillData)
   Hallgd_Gst_GsID.set(BillData.Hall[1])
   Hallgd_Gst_GsNm.set(BillData.Hall[2])
   Hallgd_Gst_GsAddress.set(BillData.Hall[3])
   Hallgd_Gst_GsCntNO.set(BillData.Hall[4])
   Hallgd_Gst_GsIDType.set(BillData.Hall[5])
   Hallgd_Gst_GsIDNo.set(BillData.Hall[6])
   Hallgd_Gst_GsEmil_ID.set(BillData.Hall[7])
   Hallgd_Gst_Rm.set(BillData.Hall[8])
   Hallgd_Gst_rm_price.set(BillData.Hall[9])
   Hallgd_Gst_Ttl_Price.set(BillData.Hall[13])
   dtin.set(BillData.Hall[10])
```

```
    dtout.set(BillData.Hall[11])
    HallCstmrBillImg = ImageTk.PhotoImage(Image.open(BillData.Hall[12]).resize((100, 90)))
    HallBillImgBtn.configure(image=HallCstmrBillImg)
    HallBillCstmrImg=BillData.Hall[12]
    Hallgd_Gst_Adv_Price.set(BillData.Hall[14])
    Hallgd_Gst_Pymnt.set(Hallgd_Gst_Ttl_Price.get()-Hallgd_Gst_Adv_Price.get())
# ------------------------------------------------------ BUTTON ------------------------------------------------
tkinter.Button(can_widget6, image=Guest_Entry,compound=CENTER,command=Hallgd_Gst_chkin,
fg="Black", width=60, activeforeground="black",activebackground="#a8701d", height=30,
bg="#a8701d", anchor=W, borderwidth=5, cursor="hand2").place(x=560+130, y=650+20)
def Hallgd_Gst_Reset():
    global HallCstmrBillImg
    Hallgd_Gst_GsID.set("")
    Hallgd_Gst_GsNm.set("")
    Hallgd_Gst_GsAddress.set("")
    Hallgd_Gst_GsCntNO.set("")
    Hallgd_Gst_GsIDType.set("")
    Hallgd_Gst_GsIDNo.set("")
    Hallgd_Gst_GsEmil_ID.set("")
    Hallgd_Gst_Rm.set("")
    Hallgd_Gst_rm_price.set("")
    Hallgd_Gst_Ttl_Price.set("")
    dtin.set("")
    dtout.set("")
    HallCstmrBillImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 90)))
    HallBillImgBtn.configure(image=HallCstmrBillImg)
    Hallgd_Gst_Adv_Price.set("")
    Hallgd_Gst_Pymnt.set("")
    Hallgd_Gst_mydata.set("Cash")
tkinter.Button(can_widget6, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Reset",command=Hallgd_Gst_Reset, bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"),
borderwidth=5, cursor="hand2").place(x=650+130, y=650+20)

def Hallgd_Gst_CHk():
    if messagebox.askyesno("Billed", "Are You Sure You Want Billing Of Hall"):
        con = connector.connect(host='localhost',
                                port='3306',
                                user='root',
                                password='Password',
                                database='Hotel Management Software')
        cur = con.cursor()
        if Hallgd_Gst_Pymnt.get() == Hallgd_Gst_Ttl_Price.get()-Hallgd_Gst_Adv_Price.get():
            query = f"insert into `Hall Customer Details` values
('{Hallgd_Gst_GsID.get()}','{Hallgd_Gst_GsNm.get()}','{Hallgd_Gst_GsAddress.get()}','{Hallgd_Gst_GsCn
tNO.get()}','{Hallgd_Gst_GsIDType.get()}','{Hallgd_Gst_GsIDNo.get()}','{Hallgd_Gst_GsEmil_ID.get()}','
{HallBillCstmrImg}','{Hallgd_Gst_Rm.get()}','{dtin.get()}','{dtout.get()}','{Hallgd_Gst_rm_price.get()}','{Ha
llgd_Gst_Ttl_Price.get()}','{Hallgd_Gst_Adv_Price.get()}','{Hallgd_Gst_Pymnt.get()}','{Hallgd_Gst_mydata.
get()}','{date.today()}');"
            cur.execute(query)
            # con.commit()
            query=f"delete from `Hall Reservation` where `Hall Number` = '{Hallgd_Gst_Rm.get()}';"
            cur.execute(query)
            # con.commit()
```

```
from docx2pdf import convert
from docxtpl import DocxTemplate
doc = DocxTemplate("Hall Billing.docx")
doc.render({"Name": Hallgd_Gst_GsNm.get(),
        "Gs_ID": Hallgd_Gst_GsID.get(),
        "Address": Hallgd_Gst_GsAddress.get(),
        "Room_Number": Hallgd_Gst_Rm.get(),
        "Check_In_Date": dtin.get(),
        "Check_Out_Date": dtout.get(),
        "Room_Price": Hallgd_Gst_rm_price.get(),
        "Payment": Hallgd_Gst_Adv_Price.get(),
        "Pymnt":Hallgd_Gst_Pymnt.get(),
        "Ttl": Hallgd_Gst_Ttl_Price.get()})
doc.save("Hall_Billing.docx")
convert(r"D:\python\Project\Hall_Billing.docx", r"D:\python\Project\Hall_Billing.pdf")
os.remove(r"D:\python\Project\Hall_Billing.docx")
os.system("Hall_Billing.pdf")
import smtplib
from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
def Email():
   try:
      connect = smtplib.SMTP('smtp.gmail.com', 587)
      connect.ehlo()
      connect.starttls()
      sender_email = "aakshamhotel@gmail.com"
      sender_passwd = "kmko wohf Irdx gthw"
      connect.login(sender_email, sender_passwd)
      receiver_email = Hallgd_Gst_GsEmil_ID.get()
      subject = "Hall Bill"
      msg_text = "Thank You For Choosing Our Hall ....."
      message = MIMEMultipart()
      message["From"] = sender_email
      message["To"] = receiver_email
      message["Subject"] = subject
      message["Bcc"] = receiver_email
      message.attach(MIMEText(msg_text, "plain"))
      filename = "Hall_Billing.pdf"
      with open(filename, "rb") as attachment:
         part = MIMEBase("application", "octet-stream")
         part.set_payload(attachment.read())
      encoders.encode_base64(part)
      part.add_header("Content-Disposition", f"attachment; filename= {filename}", )
      message.attach(part)
      text = message.as_string()
      connect.sendmail(sender_email, receiver_email, text)
      # print("Successfully email✉ sent")
      messagebox.showinfo("Mail","Invoice Is Send To You Mail")
   except Exception as e:
      # print(e)
      messagebox.showerror("Error",e)
```

```
            finally:
                connect.quit()
            Email()
        global HallCstmrBillImg
        Hallgd_Gst_GsID.set("")
        Hallgd_Gst_GsNm.set("")
        Hallgd_Gst_GsAddress.set("")
        Hallgd_Gst_GsCntNO.set("")
        Hallgd_Gst_GsIDType.set("")
        Hallgd_Gst_GsIDNo.set("")
        Hallgd_Gst_GsEmil_ID.set("")
        Hallgd_Gst_Rm.set("")
        Hallgd_Gst_rm_price.set("")
        Hallgd_Gst_Ttl_Price.set("")
        dtin.set("")
        dtout.set("")
        HallCstmrBillImg = ImageTk.PhotoImage(Image.open("passportsizephoto.webp").resize((100, 90)))
        HallBillImgBtn.configure(image=HallCstmrBillImg)
        Hallgd_Gst_Adv_Price.set("")
        Hallgd_Gst_Pymnt.set("")
        Hallgd_Gst_mydata.set("Cash")


tkinter.Button(can_widget6, image=Guest_Entry, compound=LEFT,command=Hallgd_Gst_CHk,
fg="Black", width=160, activeforeground="black",activebackground="#a8701d", height=30, text="Save &
Print", bg="#a8701d", anchor=W,font=('Century Gothic', 15, "bold"), borderwidth=5,
cursor="hand2").place(x=745+190, y=650+20)
def Hallgd_Gst_cls():
    f1.place(x=15, y=21)
    can_widgett.place(x=350, y=25)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
    # root.destroy()
tkinter.Button(can_widget6, image=Guest_Entry, compound=LEFT,command=Hallgd_Gst_cls, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=835+300, y=650+20)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")
```

```
#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Hallgd_Gst_table = ttk.Treeview(Hallgd_Gst_frm, cursor="hand2", columns=("SN_No.", "Payment_MD",
"Payment", "Payment_Dt"),
                    selectmode="browse", xscrollcommand=
Hallgd_Gst_scbr_x.set, yscrollcommand=
Hallgd_Gst_scbr_y.set)

Hallgd_Gst_scbr_x.pack(side=BOTTOM, fill=X)
Hallgd_Gst_scbr_y.pack(side=RIGHT, fill=Y)
Hallgd_Gst_scbr_x.config(command=Hallgd_Gst_table.xview)
Hallgd_Gst_scbr_y.config(command=Hallgd_Gst_table.yview)
Hallgd_Gst_table.heading("SN_No.", text="Sn No.", anchor=CENTER)
Hallgd_Gst_table.heading("Payment_MD", text="Payment Mode", anchor=CENTER)
Hallgd_Gst_table.heading("Payment", text="Payment", anchor=CENTER)
Hallgd_Gst_table.heading("Payment_Dt", text="Payment Date", anchor=CENTER)
Hallgd_Gst_table.pack(fill=BOTH, expand=1)


Hallgd_Gst_table["show"] = "headings"
Hallgd_Gst_table.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
Hallgd_Gst_table.column("Payment_MD", width=170, anchor=CENTER, minwidth=150)
Hallgd_Gst_table.column("Payment", width=100, anchor=CENTER, minwidth=70)
Hallgd_Gst_table.column("Payment_Dt", width=140, anchor=CENTER, minwidth=120)
```


```
#---------------------------------------------------  Halll Details  ----------------------------------------------------
can_widget12 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
HallDetailsimgbg = ImageTk.PhotoImage(Image.open("Photo by Kashish Lamba on
Unsplash.jpeg").resize((1585,955)))
HallDetailsimgfg = ImageTk.PhotoImage(Image.open("DtlBg.png").resize((1400,750)))

can_widget12.create_image(0,0,anchor=NW,image=HallDetailsimgbg)
can_widget12.create_image(90,110,anchor=NW,image=HallDetailsimgfg)
# can_widget12.place(x=330, y=25)

s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#     TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

#     TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Hall_Gst_dtl = Frame(can_widget12, relief=SUNKEN, borderwidth=4)
Hall_Gst_dtl.place(x=200, y=430, width=1200, height=300)
Hall_Gst_Dtl_scbr_x = Scrollbar(Hall_Gst_dtl, orient=HORIZONTAL)
Hall_Gst_Dtl_scbr_y = Scrollbar(Hall_Gst_dtl, orient=VERTICAL)
```

```
Hall_Gst_Dtl_Trvw = ttk.Treeview(Hall_Gst_dtl, cursor="hand2", columns=("SN_No.", "Gst_ID",
"Gst_Nm","Address","Cnt_No","ID_Type","ID_No","Eml_ID","Image","Hall_No","Dt_In","Dt_Ot","
Hall_Prc","Ttl","Adv","Pmnt","Pymnt_Md","PymDt"),selectmode="browse",
xscrollcommand=Hall_Gst_Dtl_scbr_x.set, yscrollcommand=Hall_Gst_Dtl_scbr_y.set)

Hall_Gst_Dtl_scbr_x.pack(side=BOTTOM, fill=X)
Hall_Gst_Dtl_scbr_y.pack(side=RIGHT, fill=Y)
Hall_Gst_Dtl_scbr_x.config(command=Hall_Gst_Dtl_Trvw.xview)
Hall_Gst_Dtl_scbr_y.config(command=Hall_Gst_Dtl_Trvw.yview)
Hall_Gst_Dtl_Trvw.heading("SN_No.", text="Sn No.", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Gst_ID", text="Guest ID", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Gst_Nm", text="Guest Name", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Address", text="Address", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Cnt_No", text="Contact No.", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("ID_Type", text="ID Type", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("ID_No", text="ID No.", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Eml_ID", text="Email ID", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Image", text="Image", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Hall_No", text="Hall No.", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Dt_In", text="From Date", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Dt_Ot", text="To Date", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Hall_Prc", text="Hall Price", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Ttl", text="Total Amount", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Adv", text="Advance", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Pmnt", text="Payment", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("Pymnt_Md", text="Payment Mode", anchor=CENTER)
Hall_Gst_Dtl_Trvw.heading("PymDt", text="Payment Date", anchor=CENTER)
Hall_Gst_Dtl_Trvw.pack(fill=BOTH, expand=1)


Hall_Gst_Dtl_Trvw["show"] = "headings"
Hall_Gst_Dtl_Trvw.column("SN_No.", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Gst_ID", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Gst_Nm", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Address", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Cnt_No", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("ID_Type", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("ID_No", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Eml_ID", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Image", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Hall_No", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Dt_In", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Dt_Ot", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Hall_Prc", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Ttl", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Adv", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Pmnt", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("Pymnt_Md", width=150, anchor=CENTER, minwidth=50)
Hall_Gst_Dtl_Trvw.column("PymDt", width=150, anchor=CENTER, minwidth=50)


# customtkinter.CTkLabel(can_widget12, text="Search By Guest ID :", font=('Century Gothic',
16)).place(x=20, y=30)
# customtkinter.CTkLabel(can_widget12, text="Search By Guest Name :", font=('Century Gothic',
```

```
16)).place(x=20, y=80)
# customtkinter.CTkLabel(master=can_widget12, text="Search By Contact Number :", font=('Century
Gothic', 16)).place(x=20, y=130)
can_widget12.create_text(800,100,text="CUSTOMER DETAILS",font=("Pristina", 50, "bold"))
can_widget12.create_text(370,50+190+30,text="Search By Guest ID :",font=("Pristina", 30, "bold"))
can_widget12.create_text(385,100+190+30,text="Search By Guest Name :",font=("Pristina", 30, "bold"))
can_widget12.create_text(410,150+190+30,text="Search By Contact Number :",font=("Pristina", 30,
"bold"))
E1Var=StringVar()
E1Var.set("")
E2Var=StringVar()
E2Var.set("")
E3Var=StringVar()
E3Var.set("")
E1=Entry(can_widget12,highlightthickness=2,textvariable=E1Var,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17 italic")
E1.place(x=280+380,y=30+190+30,width=200,height=30)
E2=Entry(can_widget12,highlightthickness=2,textvariable=E2Var,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17 italic")
E2.place(x=280+380,y=80+190+30,width=200,height=30)
E3=Entry(can_widget12,highlightthickness=2,textvariable=E3Var,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17 italic")
E3.place(x=280+380,y=130+190+30,width=200,height=30)
def Scrh():
    con = connector.connect(host='localhost',
                            port='3306',
                            user='root',
                            password='Password',
                            database='Hotel Management Software')
    cur = con.cursor()
    if E1.get()==""and E2.get()=="":
        for item in Hall_Gst_Dtl_Trvw.get_children():
            Hall_Gst_Dtl_Trvw.delete(item)
        query = f"select * from `Hall Customer Details` where `Contact No`='{E3.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            # print(row)
            Hall_Gst_Dtl_Trvw.insert("", END,
                                     values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                         row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[11],
                                         row[12], row[13], row[14], row[15], row[16].strftime("%d/%m/%y")))
            sn += 1
    elif E2.get()==""and E3.get()=="":
        for item in Hall_Gst_Dtl_Trvw.get_children():
            Hall_Gst_Dtl_Trvw.delete(item)
        query = f"select * from `Hall Customer Details` where `Guest ID`='{E1.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            # print(row)
            Hall_Gst_Dtl_Trvw.insert("", END,
                                     values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                         row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[11],
```

```
                                row[12], row[13], row[14], row[15], row[16].strftime("%d/%m/%y")))
            sn += 1
    elif E1.get()==""and E3.get()=="":
        for item in Hall_Gst_Dtl_Trvw.get_children():
            Hall_Gst_Dtl_Trvw.delete(item)
        query = f"select * from `Hall Customer Details` where `Guest Name`='{E2.get()}';"
        cur.execute(query)
        sn = 1
        for row in cur.fetchall():
            # print(row)
            Hall_Gst_Dtl_Trvw.insert("", END,
                            values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                                row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[11],
                                row[12], row[13], row[14], row[15], row[16].strftime("%d/%m/%y")))
            sn += 1
tkinter.Button(can_widget12, image=Guest_Entry, compound=LEFT,command=Scrh, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Search",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=500+420, y=210+30)
def Rset():
    E1Var.set("")
    E2Var.set("")
    E3Var.set("")
    for item in Hall_Gst_Dtl_Trvw.get_children():
        Hall_Gst_Dtl_Trvw.delete(item)
    query = f"select * from `Hall Customer Details`;"
    cur.execute(query)
    sn = 1
    for row in cur.fetchall():
        # print(row)
        Hall_Gst_Dtl_Trvw.insert("", END,
                        values=(sn, row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8],
                            row[9].strftime("%d/%m/%y"), row[10].strftime("%d/%m/%y"), row[11],
                            row[12], row[13], row[14], row[15], row[16].strftime("%d/%m/%y")))
        sn += 1
tkinter.Button(can_widget12, image=Guest_Entry, compound=LEFT,command=Rset, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Reset",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=650+420, y=210+30)
def cls():
    f1.place(x=15, y=21)
    can_widgett.place(x=350, y=25)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
```

tkinter.Button(can_widget12, image=Guest_Entry, compound=LEFT,command=cls, fg="Black", width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Close", bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5, cursor="hand2").place(x=800+420, y=210+30)

Invoice

Invoice to:

{{ name }}
{{ phone }}

Invoice no:

{{ Billno }}

Date:

{{ date }}

No	Item	Price	Qty	Discount	S.Gst	C.Gst	Total
			{{%tr for item in invoice_list %}}				
{{item[0] }}	{{item[1]}}	${{item[2]}}	{{item[3]}}	{{item[5]}}%	{{item[6]}}%	{{item[7]}}%	${{item[8]}}
			{{%tr endfor %}}				
					TOTAL:		${{total }}

BANK INFO

Bank Name: Borcelle
Bank Account: 123-456-7890

```
#---------------------------------------------------------  Barr Billing  ---------------------------------------------

can_widget7 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
bar = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-20231108_160936.png").resize((1200,1000)))
bar_Resto = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231108_171648.png").resize((950,200)))
barbg = ImageTk.PhotoImage(Image.open("./assets/Cliffside Dinner Restaurant San Antonio Santorini
Hotel.jpg").resize((1585,955)))

can_widget7.create_image(0,0,anchor=NW,image=barbg)
can_widget7.create_image(1000,470,image=bar)
can_widget7.create_image(550,0,anchor=NW,image=bar_Resto)
# can_widget7.place(x=330, y=25)
customtkinter.CTkLabel(master=can_widget7, text="Guest
Information",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place
(x=530, y=10+50+80)
customtkinter.CTkLabel(master=can_widget7, text="Payment
Information",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place
(x=500+150+200, y=10+50+80)
Mail_Id="sakshamjais100@gmail.com"

def Brrestornt_Gst_chkin():
    global Mail_Id
    os.system("python bar.py")
    Gs = pd.read_csv("BAR.csv", index_col=[0])
    Brrestornt_Gst_GsID.set(Gs.hii[1])
    Brrestornt_Gst_GsNm.set(Gs.hii[2])
    Brrestornt_Gst_Gsder.set(Gs.hii[3])
    Brrestornt_Gst_GsCntNO.set(Gs.hii[8])
    Mail_Id=Gs.hii[11]
    # print(56565656555555555555555555555)
# ---------------------------------------------------------  BUTTON  ---------------------------------------------------
----------
tkinter.Button(can_widget7, image=Guest_Entry,compound=CENTER,command=Brrestornt_Gst_chkin,
fg="Black", width=50, activeforeground="black",activebackground="#a8701d", height=30,
bg="#a8701d", anchor=W, borderwidth=5, cursor="hand2").place(x=500+300, y=235+7)
def Brrestornt_Gst_Reset():
    Brrestornt_Gst_GsID.set("")
    Brrestornt_Gst_GsNm.set("")
    Brrestornt_Gst_Gsder.set("")
    Brrestornt_Gst_GsCntNO.set("")
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30,
text="Reset",command=Brrestornt_Gst_Reset, bg="#a8701d", anchor=W,font=('Century Gothic', 17,
"bold"), borderwidth=5, cursor="hand2").place(x=590+300, y=233+7)
def idd():
    customtkinter.CTkLabel(master=can_widget7,text="Guest Id
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130
+310, y=60+40+100)
    customtkinter.CTkLabel(master=can_widget7,text="Guest Name
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130
+310, y=105+40+100)
    customtkinter.CTkLabel(master=can_widget7,text="Gender
```

```
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130
+310, y=150+40+100)
   customtkinter.CTkLabel(master=can_widget7, text="Contact No
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20 +
130 + 310, y=195 + 40 + 100)
   # customtkinter.CTkLabel(master=can_widget7,text="Religion
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=195+40)
   # customtkinter.CTkLabel(master=can_widget7,text="Address
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=240+40)
   # customtkinter.CTkLabel(master=can_widget7,text="City
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=285+40)
   # customtkinter.CTkLabel(master=can_widget7,text="Country
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=330+40)
   # customtkinter.CTkLabel(master=can_widget7,text="ID TYPE
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=420+40)
   # customtkinter.CTkLabel(master=can_widget7,text="ID NUMBER
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+130,
y=465+40)
idd()

Brrestornt_Gst_GsID=StringVar()
Brrestornt_Gst_GsNm=StringVar()
Brrestornt_Gst_Gsder=StringVar()
Brrestornt_Gst_GsCntNO=StringVar()
Brrestornt_Gst_GsID.set("")
Brrestornt_Gst_GsNm.set("")
Brrestornt_Gst_Gsder.set("")
Brrestornt_Gst_GsCntNO.set("")
Entry(can_widget7,highlightthickness=2,textvariable=Brrestornt_Gst_GsID,highlightbackground="grey",hi
ghlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+390,y=85+65+100,width=50,height=30)
Entry(can_widget7,highlightthickness=2,textvariable=Brrestornt_Gst_GsNm,highlightbackground="grey",h
ighlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+390,y=85+105+12+100,width=200,height=30)
Entry(can_widget7,highlightthickness=2,textvariable=Brrestornt_Gst_Gsder,highlightbackground="grey",hi
ghlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+390,y=85+150+25+100,width=200,height=30)
Entry(can_widget7,highlightthickness=2,textvariable=Brrestornt_Gst_GsCntNO,highlightbackground="gre
y",highlightcolor="black",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+200+390,y=85+195+40+100,width=200,height=30)

#-----------------------------------------------------------------------------------------------------------
------
customtkinter.CTkLabel(master=can_widget7, text="Food Name
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=210-15)
customtkinter.CTkLabel(master=can_widget7, text="Rate
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=290-40-15)
```

```
customtkinter.CTkLabel(master=can_widget7, text="Quantity
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=330-40-15)
customtkinter.CTkLabel(master=can_widget7, text="Amount
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=370-40-15)
customtkinter.CTkLabel(master=can_widget7, text="Discount
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=410-40-15)
customtkinter.CTkLabel(master=can_widget7, text="SGST
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=450-40-15)
customtkinter.CTkLabel(master=can_widget7, text="CGST
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=490-40-15)
customtkinter.CTkLabel(master=can_widget7, text="Total Amount
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=530+20
+300, y=530-40-15)
customtkinter.CTkLabel(master=can_widget7,
text="%",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=
780+265, y=410-40-15)
customtkinter.CTkLabel(master=can_widget7,
text="%",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=
780+265, y=450-40-15)
customtkinter.CTkLabel(master=can_widget7,
text="%",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=
780+265, y=490-40-15)

#-----------------------------------------------------------------------------------------------------------------------
------
#-------------------------------------------------------------------------------------------------------------
bar_FdNm=StringVar()
bar_FdNm.set("")
barmydata=Entry(can_widget7,highlightthickness=2,textvariable=bar_FdNm,highlightbackground="grey",h
ighlightcolor="black",fg="blue",font="consolas 17 italic").place(x=700+350+170,y=60+290-82+50-
70,width=203,height=30)
barrat=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=barrat,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic").place(x=700+350+170,y=60+290-40+50-
70+5,width=203,height=30)
barqnt=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=barqnt,highlightbackground="grey",highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=700+350+170,y=60+330-40+50-
70+15,width=60,height=30)
baramt=IntVar()
Entry(can_widget7,highlightthickness=2,highlightbackground="grey",textvariable=baramt,highlightcolor="
black",fg="blue",font="consolas 17 italic",state="readonly").place(x=700+350+170,y=60+370-40+50-
70+20,width=203,height=30)
bardst=IntVar()
bardst.set(2.0)
Entry(can_widget7,highlightthickness=2,textvariable=bardst,highlightbackground="grey",highlightcolor="b
lack",fg="blue",font="consolas 17 italic").place(x=700+350+170,y=60+410-40+50-
70+35,width=60+20,height=30)
bars_GST=IntVar()
```

```
bars_GST.set(3.0)
Entry(can_widget7,highlightthickness=2,textvariable=bars_GST,highlightbackground="grey",highlightcolor
="black",fg="blue",font="consolas 17 italic").place(x=700+350+170,y=60+450-40+50-
70+30+15,width=60+20,height=30)
barc_GST=IntVar()
barc_GST.set(3.0)
Entry(can_widget7,highlightthickness=2,textvariable=barc_GST,highlightbackground="grey",highlightcolor
="black",fg="blue",font="consolas 17 italic").place(x=700+350+170,y=60+490-40+50-
70+55,width=60+20,height=30)
barTl_Amnt=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=barTl_Amnt,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=700+350+170,y=60+530-40+50-
70+65,width=203,height=30)
bardstprice=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=bardstprice,highlightbackground="grey",highlightcolo
r="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=810+350+170,y=60+410-40+50-
70+35,width=93,height=30)
barss_GST=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=barss_GST,highlightbackground="grey",highlightcolo
r="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=810+350+170,y=60+450-40+50-
70+30+15,width=93,height=30)
barcc_GST=IntVar()
Entry(can_widget7,highlightthickness=2,textvariable=barcc_GST,highlightbackground="grey",highlightcol
or="black",fg="blue",font="consolas 17 italic",state="readonly").place(x=810+350+170,y=60+490-40+50-
70+55,width=93,height=30)
#-------------------------------------------------------------------------------------------------------------
def barsliderevent(value):
    barqnt.set(value=int(value))
    try:
        baramt.set(value=int(value * barrat.get()))
        bardstprice.set(value=int((value*barrat.get()*bardst.get())/100))
        barss_GST.set(value=int((value*barrat.get()*bars_GST.get())/100))
        barcc_GST.set(value=int((value*barrat.get()*barc_GST.get())/100))
        barTl_Amnt.set(value=str(baramt.get()-bardstprice.get()+barss_GST.get()+barcc_GST.get()))
    except EXCEPTION as e:
        messagebox.showerror("Errror",e)
    # print(value=int(value))

customtkinter.CTkSlider(can_widget7,from_=0,to=10,command=barsliderevent,number_of_steps=10,width
=100).place(x=770+260,y=60+220)
#-------------------------------------------------------------------------------------------------------------
def barNew():
    for item in barfd.get_children():
        barfd.delete(item)
    Brrestornt_Gst_GsID.set("")
    Brrestornt_Gst_GsNm.set("")
    Brrestornt_Gst_Gsder.set("")
    Brrestornt_Gst_GsCntNO.set("")
    bar_FdNm.set("")
    barrat.set(0)
    baramt.set(0)
    bardstprice.set(0)
    barss_GST.set(0)
    barcc_GST.set(0)
```

```
    barTl_Amnt.set(0)
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT, fg="Black", width=120,
activeforeground="black",activebackground="#a8701d", height=30, text="New",command=barNew,
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=580, y=500)
# barbill=pd.DataFrame(["B",1])
# barbill.to_csv("BarBillno.csv")
def barsv():
    if messagebox.askyesno("Bar Billing", "Are You Sure You Want To Save & Print"):
        barbill = pd.read_csv("BarBillno.csv", index_col=[0])
        for j in barfd.get_children():
            i=barfd.item(j)["values"]
            query=f"insert into Bar_Details
values('{barbill.Bill[0]+barbill.Bill[1]}','{Brrestornt_Gst_GsID.get()}','{Brrestornt_Gst_GsNm.get()}','{Brres
tornt_Gst_Gsder.get()}','{Brrestornt_Gst_GsCntNO.get()}','{i[1]}','{i[2]}','{i[3]}','{i[5]}','{i[6]}','{i[7]}','{i[8
]}','{i[9]}');"
            # print(query)
            cur.execute(query)
            con.commit()
        from docx2pdf import convert
        from docxtpl import DocxTemplate
        doc = DocxTemplate("Bar & Restaurant Invoice Template.docx")
        # invoice_list = [[2, "pen", 0.5, 1],
        #                 [1, "paper pack", 5, 5],
        #                 [2, "notebook", 2, 4]]
        invoice_list =[]
        tl=0
        for i in barfd.get_children():
            invoice_list.append(barfd.item(i)["values"])
            tl+=int(barfd.item(i)["values"][8])

        # print(tl)
        # print(invoice_list)
        barbill=pd.read_csv("BarBillno.csv", index_col=[0])
        doc.render({"Billno": f"{barbill.Bill[0]+barbill.Bill[1]}",
                    "name": f"{Brrestornt_Gst_GsNm.get()}",
                    "phone": f"{Brrestornt_Gst_GsCntNO.get()}",
                    "date": f'{date.today().strftime("%d/%b/%y")}',
                    "invoice_list": invoice_list,
                    "salestax": "10%",
                    "total": f"{tl}"})
        doc.save("new_invoice.docx")
        convert(r"D:\python\Project\new_invoice.docx", r"D:\python\Project\new_invoice.pdf")
        os.remove(r"D:\python\Project\new_invoice.docx")
        barbill.Bill=int(barbill.Bill[1])+1
        barbill.Bill[0]="B"
        barbill.to_csv("BarBillno.csv")
        os.system("new_invoice.pdf")
        import smtplib
        from email import encoders
        from email.mime.base import MIMEBase
        from email.mime.multipart import MIMEMultipart
        from email.mime.text import MIMEText
        def Email():
```

```
        try:
            connect = smtplib.SMTP('smtp.gmail.com', 587)
            connect.ehlo()
            connect.starttls()
            sender_email = "aakshamhotel@gmail.com"
            sender_passwd = "kmko wohf Irdx gthw"
            connect.login(sender_email, sender_passwd)
            receiver_email =Mail_Id
            subject = "Resturant Bill"
            msg_text = "Thanks….For Ordering Food"
            message = MIMEMultipart()
            message["From"] = sender_email
            message["To"] = receiver_email
            message["Subject"] = subject
            message["Bcc"] = receiver_email
            message.attach(MIMEText(msg_text, "plain"))
            filename = "new_invoice.pdf"
            with open(filename, "rb") as attachment:
                part = MIMEBase("application", "octet-stream")
                part.set_payload(attachment.read())
            encoders.encode_base64(part)
            part.add_header("Content-Disposition", f"attachment; filename= {filename}", )
            message.attach(part)
            text = message.as_string()
            connect.sendmail(sender_email, receiver_email, text)
            # print("Successfully email✉ sent")
            messagebox.showinfo("Mailed","Successfully email✉ sent")
        except Exception as e:
            # print(e)
            messagebox.showerror("Error",e)
        finally:
            connect.quit()
    Email()
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT,command=barsv, fg="Black",
width=260, activeforeground="black",activebackground="#a8701d", height=30, text="Save & Print",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=580, y=550)
def barcls():
    f1.place(x=15, y=21)
    can_widgett.place(x=350, y=25)
    can_widget1.place(x=1000, y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT,command=barcls, fg="Black",
```

```
width=160, activeforeground="black",activebackground="#a8701d", height=30, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 17, "bold"), borderwidth=5,
cursor="hand2").place(x=860, y=550)
sn=1
def BarCart():
   global sn
   barfd.insert("", END, values=(sn,bar_FdNm.get(), barrat.get(),
barqnt.get(),baramt.get(),bardst.get(),bars_GST.get(),barc_GST.get(),barTl_Amnt.get(),date.today()))
   sn += 1
   bar_FdNm.set("")
   barrat.set(0)
   baramt.set(0)
   barqnt.set("0")
   bardstprice.set(0)
   barss_GST.set(0)
   barcc_GST.set(0)
   barTl_Amnt.set(0)
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT,command=BarCart, fg="Black",
width=160, activeforeground="black",activebackground="#a8701d", height=30, text="Add To Cart",
bg="#a8701d", anchor=W,font=('Century Gothic', 16, "bold"), borderwidth=5,
cursor="hand2").place(x=860, y=500)
def Barrm():
   try:
      barfd.delete(barfd.selection())
   except:
      pass
tkinter.Button(can_widget7, image=Guest_Entry, compound=LEFT,command=Barrm, fg="Black",
width=120, activeforeground="black",activebackground="#a8701d", height=30, text="Remove",
bg="#a8701d", anchor=W,font=('Century Gothic', 16, "bold"), borderwidth=5,
cursor="hand2").place(x=720, y=500)

#-------------------------------------------------------------------------------------------------------------
------
barfrm = Frame(can_widget7, relief=SUNKEN, borderwidth=4)
barfrm.place(x=560, y=640, width=900, height=170)
barscbr_x = Scrollbar(barfrm, orient=HORIZONTAL)
barscbr_y = Scrollbar(barfrm, orient=VERTICAL)
barfd = ttk.Treeview(barfrm, cursor="hand2", columns=("SN_No.","Fd_Nm", "Rate", "Quantity",
"Amount","Discount","S_GST","C_GST","Ttl_Amnt","Bill_Date"),selectmode="browse",
xscrollcommand=barscbr_x.set, yscrollcommand=barscbr_y.set)

barscbr_x.pack(side=BOTTOM, fill=X)
barscbr_y.pack(side=RIGHT, fill=Y)
barscbr_x.config(command=barfd.xview)
barscbr_y.config(command=barfd.yview)
barfd.heading("SN_No.", text="Sn No.", anchor=CENTER)
barfd.heading("Fd_Nm", text="Food Name", anchor=CENTER)
barfd.heading("Rate", text="Rate", anchor=CENTER)
barfd.heading("Quantity", text="Quantity", anchor=CENTER)
barfd.heading("Amount", text="Amount", anchor=CENTER)
barfd.heading("Discount", text="Discount", anchor=CENTER)
barfd.heading("S_GST", text="S GST", anchor=CENTER)
barfd.heading("C_GST", text="C GST", anchor=CENTER)
barfd.heading("Ttl_Amnt", text="Total Amount", anchor=CENTER)
```

```
barfd.heading("Bill_Date", text="Bill Date", anchor=CENTER)
# barfd.heading("Pymnt_Mde", text="Payment Mode", anchor=CENTER)
barfd.pack(fill=BOTH, expand=1)


barfd["show"] = "headings"
barfd.column("SN_No.", width=90, anchor=CENTER, minwidth=50)
barfd.column("Fd_Nm", width=200, anchor=CENTER, minwidth=150)
barfd.column("Rate", width=100, anchor=CENTER, minwidth=70)
barfd.column("Quantity", width=140, anchor=CENTER, minwidth=120)
barfd.column("Amount", width=140, anchor=CENTER, minwidth=120)
barfd.column("Discount", width=140, anchor=CENTER, minwidth=120)
barfd.column("S_GST", width=140, anchor=CENTER, minwidth=120)
barfd.column("C_GST", width=140, anchor=CENTER, minwidth=120)
barfd.column("Ttl_Amnt", width=140, anchor=CENTER, minwidth=120)
barfd.column("Bill_Date", width=140, anchor=CENTER, minwidth=120)
# barfd.column("Pymnt_Mde", width=140, anchor=CENTER, minwidth=120)
```


```
#------------------------------------ Worker List ----------------------------------------------------------
-----

can_widget8 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget.set_appearance_mode("Dark")
wkdtimg1 = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231107_203022.png").resize((500,100)))
wkdtimg2 = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231107_202211.png").resize((900,870)))
wkdtimg3 = ImageTk.PhotoImage(Image.open("./assets/PhotoRoom-
20231107_202923.png").resize((650,400)))
Wkdtimg = ImageTk.PhotoImage(Image.open("./assets/8vUcyZ.jpg").resize((1585,955)))
can_widget8.create_image(0,0,anchor=NW,image=Wkdtimg)
can_widget8.create_image(1050,475,image=wkdtimg2)
can_widget8.create_image(1050,100,image=wkdtimg1)
```

```
can_widget8.create_image(400,750,image=wkdtimg3)
# can_widget8.place(x=330, y=25)
# customtkinter.CTkLabel(master=can_widget8, text="Worker Details", font=('Times New Roman', 30,
"bold"),fg_color="black",bg_color="black").place(x=500, y=10)
def Wk_Lst_idd():
   customtkinter.CTkLabel(master=can_widget8,text="Worker Id
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=150-20)
   customtkinter.CTkLabel(master=can_widget8,text="Worker Name
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=190-20)
   customtkinter.CTkLabel(master=can_widget8,text="Gender
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=230-20)
   customtkinter.CTkLabel(master=can_widget8,text="Religion
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=270-20)
   customtkinter.CTkLabel(master=can_widget8,text="Address
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=310-20)
   customtkinter.CTkLabel(master=can_widget8,text="Contact No
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=350-20)
   customtkinter.CTkLabel(master=can_widget8,text="D.O.J
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=390-20)
   customtkinter.CTkLabel(master=can_widget8,text="Id Type
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Pristina',
25,"bold")).place(x=20+120+450, y=430-20)
   customtkinter.CTkLabel(master=can_widget8,text="Id Number
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Pristina',
25,"bold")).place(x=20+120+450, y=470-20)
   customtkinter.CTkLabel(master=can_widget8,text="Department
:",text_color="Black",fg_color="#fdeba6",bg_color="#fdeba6",font=('Script MT Bold',
25)).place(x=20+120+450, y=510-20)
Wk_Lst_idd()
#-----------------------------------------------------------------------------------------------------------
-------
Wk_Lst_ID=StringVar()
Wk_Lst_Nm=StringVar()
Wk_Lst_der=StringVar()
Wk_Lst_gion=StringVar()
Wk_Lst_Address=StringVar()
Wk_Lst_Cntry=StringVar()
Wk_Lst_CntNO=StringVar()
Wk_Lst_IDType=StringVar()
Wk_Lst_IDNo=StringVar()
Dept=StringVar()
Wk_Lst_ID.set("")
Wk_Lst_Nm.set("")
Wk_Lst_der.set("")
Wk_Lst_gion.set("")
Wk_Lst_Address.set("")
Wk_Lst_Cntry.set("")
```

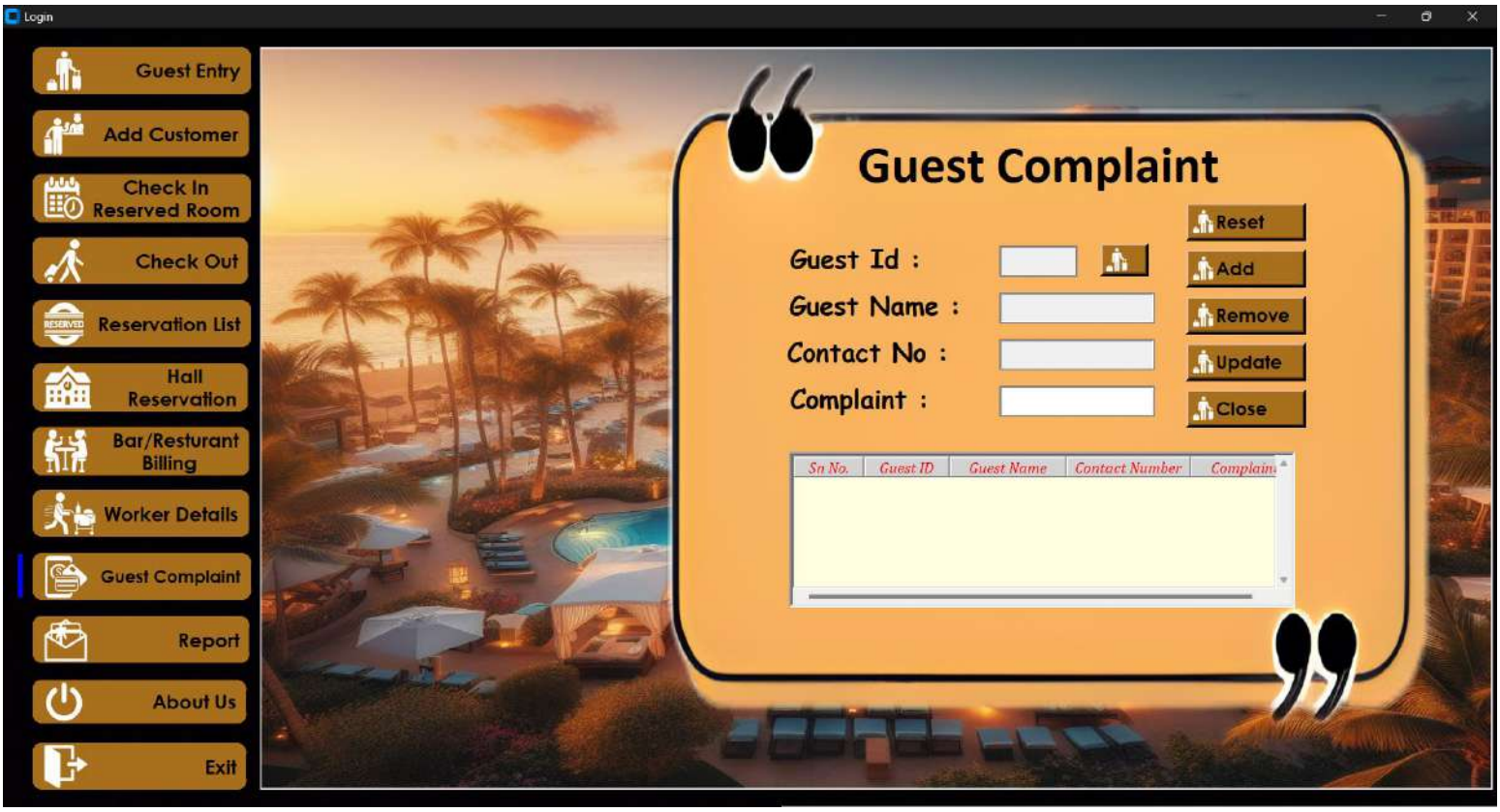
```
Wk_Lst_CntNO.set("")
Wk_Lst_IDType.set("")
Wk_Lst_IDNo.set("")
Dept.set("")
# pd.DataFrame(data=["W",1]).to_csv("WorkerDtl.csv")
wd=pd.read_csv("WorkerDtl.csv",index_col=[0])
Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_ID,highlightbackground="grey",highlightcol
or="red",fg="blue",font="consolas 17
italic",state="readonly").place(x=140+800,y=90+75,width=50,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_Nm,highlightbackground="grey",highlightco
lor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=80+135,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_der,highlightbackground="grey",highlightco
lor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=75+190,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_gion,highlightbackground="grey",highlightc
olor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=70+245,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_Address,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=65+300,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_Cntry,highlightbackground="grey",highlight
color="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=55+360,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_CntNO,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=50+415,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_IDType,highlightbackground="grey",highlig
htcolor="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=45+470,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Wk_Lst_IDNo,highlightbackground="grey",highlight
color="red",fg="blue",font="consolas 17 italic").place(x=140+800,y=40+525,width=200,height=30)
Entry(can_widget8,highlightthickness=2,textvariable=Dept,highlightbackground="grey",highlightcolor="re
d",fg="blue",font="consolas 17 italic").place(x=140+800,y=30+585,width=200,height=30)

def Wk_Lst_INFOrr():
    wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
    Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
    Wk_Lst_Nm.set(value="")
    Wk_Lst_der.set(value="")
    Wk_Lst_gion.set(value="")
    Wk_Lst_Address.set(value="")
    Wk_Lst_Cntry.set(value="")
    Wk_Lst_CntNO.set(value="")
    Wk_Lst_IDType.set(value="")
    Wk_Lst_IDNo.set(value="")
    Dept.set(value="")
    for item in Wk_Lst_Tabke.get_children():
        Wk_Lst_Tabke.delete(item)
    query = "select * from Worker_Details;"
    cur.execute(query)
    sn = 1
    for i in cur.fetchall():
        Wk_Lst_Tabke.insert("", END, values=(sn, i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9]))
        sn += 1
tkinter.Button(can_widget8, image=Guest_Entry, compound=LEFT,command=Wk_Lst_INFOrr,
fg="Black", width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5,
text="Reset", bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=100+60)
Wk_Lst_sn=1
```

```
def GST_INFOAdd():
   if messagebox.askyesno("Worker Details","Are You Sure You Want To Add Worker"):
      query=f"insert into Worker_Details values
('{Wk_Lst_ID.get()}','{Wk_Lst_Nm.get()}','{Wk_Lst_der.get()}','{Wk_Lst_gion.get()}','{Wk_Lst_Address.g
et()}','{Wk_Lst_Cntry.get()}','{Wk_Lst_CntNO.get()}','{Wk_Lst_IDType.get()}','{Wk_Lst_IDNo.get()}','{D
ept.get()}');"
      cur.execute(query)
      con.commit()
      global Wk_Lst_sn
      Wk_Lst_Tabke.insert(parent="", index=0,
values=(Wk_Lst_sn,Wk_Lst_ID.get(),Wk_Lst_Nm.get(),Wk_Lst_der.get(),Wk_Lst_gion.get(),Wk_Lst_Add
ress.get(),Wk_Lst_Cntry.get(),Wk_Lst_CntNO.get(),Wk_Lst_IDType.get(),Wk_Lst_IDNo.get()))
      wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
      wd.wt[1]=int(wd.wt[1])+1
      wd.to_csv("WorkerDtl.csv")
      wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
      Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
      Wk_Lst_Nm.set(value="")
      Wk_Lst_der.set(value="")
      Wk_Lst_gion.set(value="")
      Wk_Lst_Address.set(value="")
      Wk_Lst_Cntry.set(value="")
      Wk_Lst_CntNO.set(value="")
      Wk_Lst_IDType.set(value="")
      Wk_Lst_IDNo.set(value="")
      Dept.set(value="")
      for item in Wk_Lst_Tabke.get_children():
         Wk_Lst_Tabke.delete(item)
      query = "select * from Worker_Details;"
      cur.execute(query)
      sn = 1
      for i in cur.fetchall():
         Wk_Lst_Tabke.insert("", END, values=(sn, i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9]))
         sn += 1
tkinter.Button(can_widget8, image=Guest_Entry, compound=LEFT, fg="Black", width=120+20,
activeforeground="black",activebackground="#a8701d", height=30+5,
text="Add",command=GST_INFOAdd, bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"),
borderwidth=5, cursor="hand2").place(x=1200, y=160+60)
def GST_INFOrm():
   if messagebox.askyesno("Remove Worker Details","Are You Sure You Want To Remove Worker
Details"):
      query = f"DELETE FROM Worker_Details where `Worker Id`
='{(Wk_Lst_Tabke.item(Wk_Lst_Tabke.selection())['values'][1])}'"
      cur.execute(query)
      con.commit()
      for item in Wk_Lst_Tabke.get_children():
         Wk_Lst_Tabke.delete(item)
      query = "select * from Worker_Details;"
      cur.execute(query)
      sn = 1
      for i in cur.fetchall():
         Wk_Lst_Tabke.insert("", END, values=(sn, i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9]))
         sn += 1
      wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
```

```
        Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
        Wk_Lst_Nm.set(value="")
        Wk_Lst_der.set(value="")
        Wk_Lst_gion.set(value="")
        Wk_Lst_Address.set(value="")
        Wk_Lst_Cntry.set(value="")
        Wk_Lst_CntNO.set(value="")
        Wk_Lst_IDType.set(value="")
        Wk_Lst_IDNo.set(value="")
        Dept.set(value="")
tkinter.Button(can_widget8, image=Guest_Entry, compound=LEFT,command=GST_INFOrm, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Remove",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=220+60)
def GST_INFOupp():
    if messagebox.askyesno("Update Worker Details","Are you Sure You Want To Update Worker Details"):
        query =f"update Worker_Details set `Worker Name`='{Wk_Lst_Nm.get()}',
`Gender`='{Wk_Lst_der.get()}', `Religion`='{Wk_Lst_gion.get()}', `Address`='{Wk_Lst_Address.get()}',
`Contact No`='{Wk_Lst_Cntry.get()}', `Date Of Joining`='{Wk_Lst_CntNO.get()}', `Id
Type`='{Wk_Lst_IDType.get()}', `Id No.`='{Wk_Lst_IDNo.get()}', `Department`='{Dept.get()}';"
        cur.execute(query)
        con.commit()
        for item in Wk_Lst_Tabke.get_children():
            Wk_Lst_Tabke.delete(item)
        query = "select * from Worker_Details;"
        cur.execute(query)
        sn = 1
        for i in cur.fetchall():
            Wk_Lst_Tabke.insert("", END, values=(sn, i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9]))
            sn += 1
        wd = pd.read_csv("WorkerDtl.csv", index_col=[0])
        Wk_Lst_ID.set(f"{wd.wt[0]}{wd.wt[1]}")
        Wk_Lst_Nm.set(value="")
        Wk_Lst_der.set(value="")
        Wk_Lst_gion.set(value="")
        Wk_Lst_Address.set(value="")
        Wk_Lst_Cntry.set(value="")
        Wk_Lst_CntNO.set(value="")
        Wk_Lst_IDType.set(value="")
        Wk_Lst_IDNo.set(value="")
        Dept.set(value="")
tkinter.Button(can_widget8, image=Guest_Entry, compound=LEFT,command=GST_INFOupp,
fg="Black", width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5,
text="Update", bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=280+60)
def GST_INFOcls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
```

```
   can_widget7.place(x=1000, y=1000)
   can_widget8.place(x=1000, y=1000)
   can_widget9.place(x=1000, y=1000)
   can_widget10.place(x=1000, y=1000)
   can_widget11.place(x=1000, y=1000)
   can_widget12.place(x=1000, y=1000)
tkinter.Button(can_widget8, image=Guest_Entry, compound=LEFT,command=GST_INFOcls, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200, y=340+60)



#-------------------------------------------------------------------------------------------------------------------
-------
s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

Wk_Lst_frm1 = Frame(can_widget8, relief=SUNKEN, borderwidth=4)
Wk_Lst_frm1.place(x=740, y=660, width=650, height=150)
Wk_Lst_scbr_x = Scrollbar(Wk_Lst_frm1, orient=HORIZONTAL)
Wk_Lst_scbr_y = Scrollbar(Wk_Lst_frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Wk_Lst_Tabke = ttk.Treeview(Wk_Lst_frm1, cursor="hand2", columns=("SN_No.", "Gs_ID",
"Gs_Name", "Gender","Religion","Address","Contact_No","D.O.J","ID_Type","ID_No","Dept"),
                    selectmode="browse", xscrollcommand=Wk_Lst_scbr_x.set,
yscrollcommand=Wk_Lst_scbr_y.set)

Wk_Lst_scbr_x.pack(side=BOTTOM, fill=X)
Wk_Lst_scbr_y.pack(side=RIGHT, fill=Y)
Wk_Lst_scbr_x.config(command=Wk_Lst_Tabke.xview)
Wk_Lst_scbr_y.config(command=Wk_Lst_Tabke.yview)
Wk_Lst_Tabke.heading("SN_No.", text="Sn No.", anchor=CENTER)
Wk_Lst_Tabke.heading("Gs_ID", text="Guest ID", anchor=CENTER)
Wk_Lst_Tabke.heading("Gs_Name", text="Guest Name", anchor=CENTER)
Wk_Lst_Tabke.heading("Gender", text="Gender", anchor=CENTER)
Wk_Lst_Tabke.heading("Religion", text="Religion", anchor=CENTER)
Wk_Lst_Tabke.heading("Address", text="Address", anchor=CENTER)
Wk_Lst_Tabke.heading("Contact_No", text="Country", anchor=CENTER)
Wk_Lst_Tabke.heading("D.O.J", text="Contact Number", anchor=CENTER)
```

```
Wk_Lst_Tabke.heading("ID_Type", text="ID Type", anchor=CENTER)
Wk_Lst_Tabke.heading("ID_No", text="ID No", anchor=CENTER)
Wk_Lst_Tabke.heading("Dept", text="Department", anchor=CENTER)
Wk_Lst_Tabke.pack(fill=BOTH, expand=1)


Wk_Lst_Tabke["show"] = "headings"
Wk_Lst_Tabke.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
Wk_Lst_Tabke.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
Wk_Lst_Tabke.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
Wk_Lst_Tabke.column("Gender", width=130, anchor=CENTER, minwidth=130)
Wk_Lst_Tabke.column("Religion", width=140, anchor=CENTER, minwidth=140)
Wk_Lst_Tabke.column("Address", width=140, anchor=CENTER, minwidth=140)
Wk_Lst_Tabke.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
Wk_Lst_Tabke.column("D.O.J", width=140, anchor=CENTER, minwidth=140)
Wk_Lst_Tabke.column("ID_Type", width=140, anchor=CENTER, minwidth=120)
Wk_Lst_Tabke.column("ID_No", width=140, anchor=CENTER, minwidth=120)
Wk_Lst_Tabke.column("Dept", width=140, anchor=CENTER, minwidth=120)
query="select * from Worker_Details;"
cur.execute(query)
sn=1
for i in cur.fetchall():
    Wk_Lst_Tabke.insert("",END,values=(sn,i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7],i[8],i[9]))
    sn+=1
def Wk_Lst_Tabke_select(_):
    j=Wk_Lst_Tabke.item(Wk_Lst_Tabke.selection())['values']
    # print(j)
    Wk_Lst_ID.set(value=j[1])
    Wk_Lst_Nm.set(value=j[2])
    Wk_Lst_der.set(value=j[3])
    Wk_Lst_gion.set(value=j[4])
    Wk_Lst_Address.set(value=j[5])
    Wk_Lst_Cntry.set(value=j[6])
    Wk_Lst_CntNO.set(value=j[7])
    Wk_Lst_IDType.set(value=j[8])
    Wk_Lst_IDNo.set(value=j[9])
    Dept.set(value=j[10])

    # sep.to_csv("hi.csv")
    # print(so)
    # root.destroy()
Wk_Lst_Tabke.bind('<<TreeviewSelect>>',Wk_Lst_Tabke_select)
```



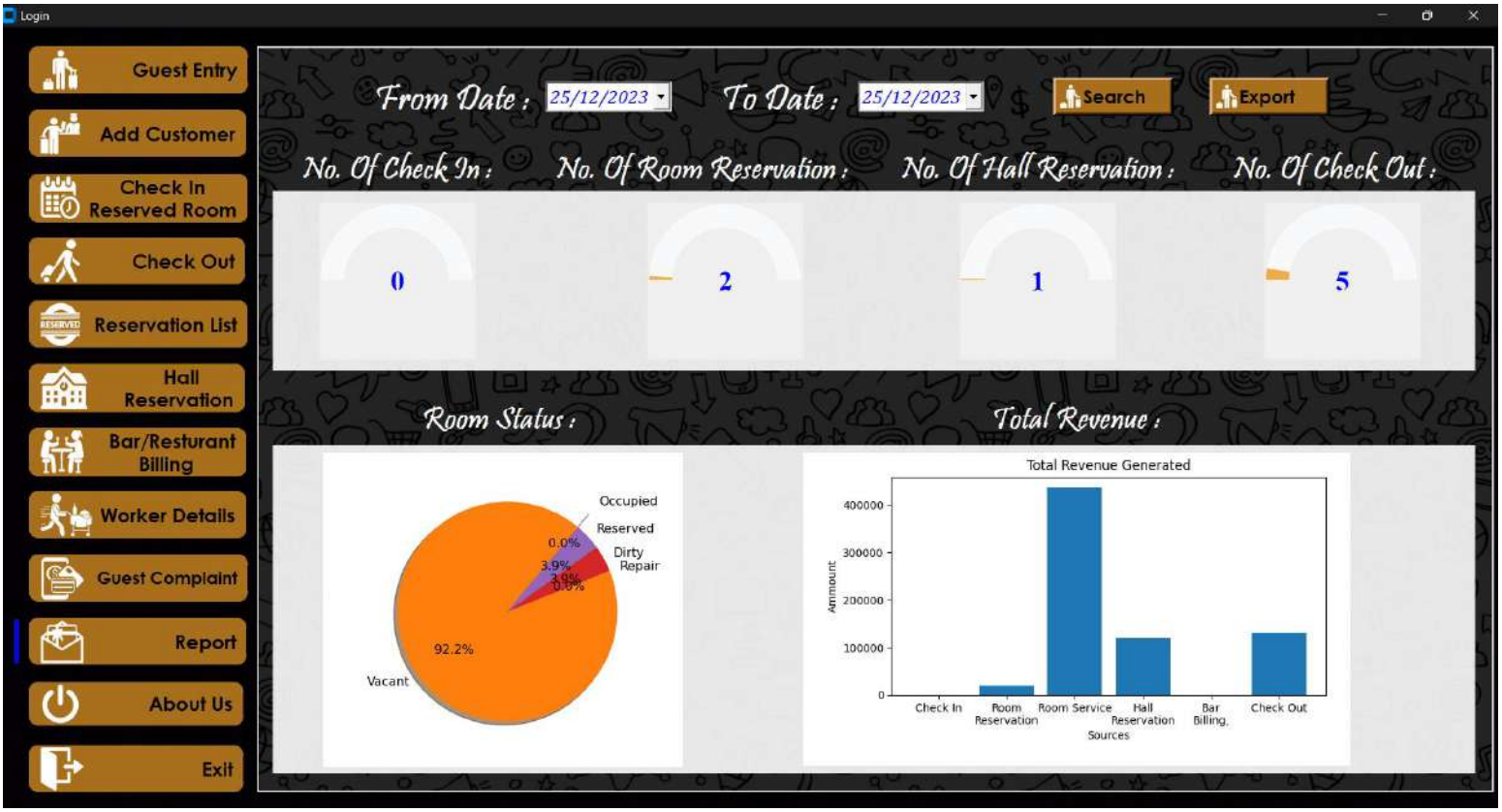

```
#------------------------------------  Guest Complaint  ---------------------------------------------
# img3 = ImageTk.PhotoImage(Image.open("./assets/11tst.jpeg").resize((1585,955)))

can_widget9 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget9.place(x=330, y=25)
# can_widget.set_appearance_mode("Dark")
can_widget9.create_image(0,0,anchor=NW,image=img5)
img45 = ImageTk.PhotoImage(Image.open("./assets/4 (1).png").resize((1100,1700)))
can_widget9.create_image(1010,505,image=img45)
can_widget9.create_text(1000, 150, text="Guest Complaint", font=('Calibri', 50, "bold"), fill="black")
# can_widget9.create_image(780,475,image=ChkOUT_cn)
# can_widget9.place(x=330, y=25)
# customtkinter.CTkLabel(master=can_widget9, text="Guest Complaint", font=('Times New Roman', 50,
"bold"),fg_color="black",bg_color="black").place(x=500, y=10)
def Gst_cmpnt_idd():
    can_widget9.create_text(300+500-35, 270, text="Guest Id :", font=("Comic Sans MS", 25, "bold"),
fill="black")
    can_widget9.create_text(300+500+5-15, 330, text="Guest Name :", font=("Comic Sans MS", 25, "bold"),
fill="black")
    can_widget9.create_text(300+500-5-15, 390, text="Contact No :", font=("Comic Sans MS", 25, "bold"),
fill="black")
    can_widget9.create_text(300+500-15-15, 450, text="Complaint :", font=("Comic Sans MS", 25, "bold"),
fill="black")
    # customtkinter.CTkLabel(master=can_widget9,text="Guest Id
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+120,
y=60)
    # customtkinter.CTkLabel(master=can_widget9,text="Guest Name
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+120,
y=105)
    # customtkinter.CTkLabel(master=can_widget9,text="Contact No
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+120,
y=375)
    # customtkinter.CTkLabel(master=can_widget9,text="Complaint
```

```
:",text_color="Black",fg_color="#8cfff9",bg_color="#8cfff9",font=('Pristina',25,"bold")).place(x=20+120,
y=420)
Gst_cmpnt_idd()
#-------------------------------------------------------------------------------------------------------------
Gst_cmpnt_ID=StringVar()
Gst_cmpnt_Nm=StringVar()
Gst_cmpnt_CntNO=StringVar()
Gst_cmpnt_Cmplnt=StringVar()
Gst_cmpnt_ID.set("")
Gst_cmpnt_Nm.set("")
Gst_cmpnt_CntNO.set("")
Gst_cmpnt_Cmplnt.set("")
Entry(can_widget9,highlightthickness=2,textvariable=Gst_cmpnt_ID,highlightbackground="grey",highlight
color="black",fg="#0012A8",font=("Comic Sans MS", 20, "italic
bold"),state="readonly",bg="#FFAB74").place(x=140+210+600,y=255,width=100,height=40)
Entry(can_widget9,highlightthickness=2,textvariable=Gst_cmpnt_Nm,highlightbackground="grey",highligh
tcolor="black",fg="#0012A8",font=("Comic Sans MS", 20, "italic
bold"),state="readonly",bg="#FFAB74").place(x=140+210+600,y=315,width=200,height=40)
Entry(can_widget9,highlightthickness=2,textvariable=Gst_cmpnt_CntNO,highlightbackground="grey",high
lightcolor="Black",fg="#0012A8",font=("Comic Sans MS", 20, "italic
bold"),state="readonly",bg="#FFAB74").place(x=140+210+600,y=375,width=200,height=40)
Entry(can_widget9,highlightthickness=2,textvariable=Gst_cmpnt_Cmplnt,highlightbackground="grey",high
lightcolor="black",fg="#0012A8",bg="white",font=("Comic Sans MS", 20, "italic
bold")).place(x=140+210+600,y=435,width=200,height=40)
# ------------------------------------------------------------------------------------------------------------
def Gstry():
   os.system("python Check_In_List.py")
   Gs = pd.read_csv("Checkin_list.csv", index_col=[0])
   Gst_cmpnt_ID.set(Gs.ChkIn[1])
   Gst_cmpnt_Nm.set(Gs.ChkIn[2])
   Gst_cmpnt_CntNO.set(Gs.ChkIn[8])
tkinter.Button(can_widget9, image=Guest_Entry,compound=CENTER,command=Gstry, fg="Black",
width=50, activeforeground="black",activebackground="#a8701d", height=30, bg="#a8701d", anchor=W,
borderwidth=5, cursor="hand2").place(x=1080, y=252)
def Gst_cmpnt_INFOrr():
   Gst_cmpnt_ID.set(value="")
   Gst_cmpnt_Nm.set(value="")
   Gst_cmpnt_CntNO.set(value="")
   Gst_cmpnt_Cmplnt.set(value="")
tkinter.Button(can_widget9, image=Guest_Entry, compound=LEFT,command=Gst_cmpnt_INFOrr,
fg="Black", width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5,
text="Reset", bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200-10, y=100+100)
Gst_cmpnt_sn=1
def GST_INFOAdd():
   if messagebox.askyesno("Add Complaint","Are You Sure You Want To Register Complaint"):
      query=f"insert into Complaint
values('{Gst_cmpnt_ID.get()}','{Gst_cmpnt_Nm.get()}','{Gst_cmpnt_CntNO.get()}','{Gst_cmpnt_Cmplnt.ge
t()}');"
      cur.execute(query)
      con.commit()
      Gst_cmpnt_ID.set(value="")
      Gst_cmpnt_Nm.set(value="")
      Gst_cmpnt_CntNO.set(value="")
```

```
    Gst_cmpnt_Cmplnt.set(value="")
    query="select * from Complaint;"
    cur.execute(query)
    sn=1
    for i in Gst_cmpnt_Tabke.get_children():
        Gst_cmpnt_Tabke.delete(i)
    for i in cur.fetchall():
        Gst_cmpnt_Tabke.insert(parent="", index=0, values=(sn,i[0],i[1],i[2],i[3]))
        sn+=1

tkinter.Button(can_widget9, image=Guest_Entry, compound=LEFT, fg="Black", width=120+20,
activeforeground="black",activebackground="#a8701d", height=30+5,
text="Add",command=GST_INFOAdd, bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"),
borderwidth=5, cursor="hand2").place(x=1200-10, y=160+100)
def GST_INFOrm():
    if messagebox.askyesno("Remove Complaint","Are You Sure You Want To Remove Complaint"):
        query=f"delete from complaint where `Guest Id`=
'{(Gst_cmpnt_Tabke.item(Gst_cmpnt_Tabke.selection())['values'])[1]}';"
    cur.execute(query)
    con.commit()
    Gst_cmpnt_ID.set(value="")
    Gst_cmpnt_Nm.set(value="")
    Gst_cmpnt_CntNO.set(value="")
    Gst_cmpnt_Cmplnt.set(value="")
    query="select * from Complaint;"
    cur.execute(query)
    sn=1
    for i in Gst_cmpnt_Tabke.get_children():
        Gst_cmpnt_Tabke.delete(i)
    for i in cur.fetchall():
        Gst_cmpnt_Tabke.insert(parent="", index=0, values=(sn,i[0],i[1],i[2],i[3]))
        sn+=1
tkinter.Button(can_widget9, image=Guest_Entry, compound=LEFT,command=GST_INFOrm, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Remove",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200-10, y=220+100)
def GST_INFOupp():
    if messagebox.askyesno("Update Complaint", "Are You Sure You Want To Update Complaint"):
        query=f"update Complaint set Complaint = '{Gst_cmpnt_Cmplnt.get()}' where `Guest
Id`='{(Gst_cmpnt_Tabke.item(Gst_cmpnt_Tabke.selection())['values'])[1]}';"
    cur.execute(query)
    con.commit()
    Gst_cmpnt_ID.set(value="")
    Gst_cmpnt_Nm.set(value="")
    Gst_cmpnt_CntNO.set(value="")
    Gst_cmpnt_Cmplnt.set(value="")
    query="select * from Complaint;"
    cur.execute(query)
    sn=1
    for i in Gst_cmpnt_Tabke.get_children():
        Gst_cmpnt_Tabke.delete(i)
    for i in cur.fetchall():
        Gst_cmpnt_Tabke.insert(parent="", index=0, values=(sn,i[0],i[1],i[2],i[3]))
        sn+=1
```

```
tkinter.Button(can_widget9, image=Guest_Entry, compound=LEFT,command=GST_INFOupp,
fg="Black", width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5,
text="Update", bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200-10, y=280+100)

def GST_INFOcls():
    f1.place(x=15,y=21)
    can_widgett.place(x=330, y=25)
    can_widget1.place(x=1000,y=1000)
    can_widget2.place(x=1000, y=1000)
    can_widget3.place(x=1000, y=1000)
    can_widget4.place(x=1000, y=1000)
    can_widget5.place(x=1000, y=1000)
    can_widget6.place(x=1000, y=1000)
    can_widget7.place(x=1000, y=1000)
    can_widget8.place(x=1000, y=1000)
    can_widget9.place(x=1000, y=1000)
    can_widget10.place(x=1000, y=1000)
    can_widget11.place(x=1000, y=1000)
    can_widget12.place(x=1000, y=1000)
tkinter.Button(can_widget9, image=Guest_Entry, compound=LEFT,command=GST_INFOcls, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Close",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1200-10, y=340+100)



#-------------------------------------------------------------------------------------------------------------
-------
s = ttk.Style()
s.theme_use("winnative")  # classic , alt,default , winnative , xpnative , clam , vista

#     FOR INSERT VALUES

s.configure(".", font=("consolas", 14, "italic"), foreground="blue")

#    TO APPLY ON WHOLE TREEVIEW

s.configure("Treeview", foreground="black", background="light yellow", rowheight=25,
fieldbackground="light yellow")
s.map("Treeview", background=[("selected", "blue")])
s.configure("Treeview.Heading", font=("Cambria", 15, "italic"), foreground="red", background="light
grey")

Gst_cmpnt_frm1 = Frame(can_widget9, relief=SUNKEN, borderwidth=4)
Gst_cmpnt_frm1.place(x=680, y=520, width=650, height=200)
Gst_cmpnt_scbr_x = Scrollbar(Gst_cmpnt_frm1, orient=HORIZONTAL)
Gst_cmpnt_scbr_y = Scrollbar(Gst_cmpnt_frm1, orient=VERTICAL)
#    TO APPLY ON COLUMNS
# s.configure("Treeview.Heading",font=("Cambria",17,"italic"),foreground="red",background="light
grey")
Gst_cmpnt_Tabke = ttk.Treeview(Gst_cmpnt_frm1, cursor="hand2", columns=("SN_No.", "Gs_ID",
"Gs_Name","Contact_No","Cmplnt"),selectmode="browse", xscrollcommand=Gst_cmpnt_scbr_x.set,
yscrollcommand=Gst_cmpnt_scbr_y.set)
```

```
Gst_cmpnt_scbr_x.pack(side=BOTTOM, fill=X)
Gst_cmpnt_scbr_y.pack(side=RIGHT, fill=Y)
Gst_cmpnt_scbr_x.config(command=Gst_cmpnt_Tabke.xview)
Gst_cmpnt_scbr_y.config(command=Gst_cmpnt_Tabke.yview)
Gst_cmpnt_Tabke.heading("SN_No.", text="Sn No.", anchor=CENTER)
Gst_cmpnt_Tabke.heading("Gs_ID", text="Guest ID", anchor=CENTER)
Gst_cmpnt_Tabke.heading("Gs_Name", text="Guest Name", anchor=CENTER)
Gst_cmpnt_Tabke.heading("Contact_No", text="Contact Number", anchor=CENTER)
Gst_cmpnt_Tabke.heading("Cmplnt", text="Complaint", anchor=CENTER)
Gst_cmpnt_Tabke.pack(fill=BOTH, expand=1)


Gst_cmpnt_Tabke["show"] = "headings"
Gst_cmpnt_Tabke.column("SN_No.", width=90, anchor=CENTER, minwidth=90)
Gst_cmpnt_Tabke.column("Gs_ID", width=110, anchor=CENTER, minwidth=110)
Gst_cmpnt_Tabke.column("Gs_Name", width=150, anchor=CENTER, minwidth=150)
Gst_cmpnt_Tabke.column("Contact_No", width=160, anchor=CENTER, minwidth=140)
Gst_cmpnt_Tabke.column("Cmplnt", width=140, anchor=CENTER, minwidth=120)
sn=1
query = "select * from Complaint;"
cur.execute(query)
for i in cur.fetchall():
    Gst_cmpnt_Tabke.insert(parent="", index=0, values=(sn, i[0], i[1], i[2], i[3]))
    sn += 1
def Gst_cmpnt_Tabke_select(_):
    i= Gst_cmpnt_Tabke.item(Gst_cmpnt_Tabke.selection())['values']
    Gst_cmpnt_ID.set(i[1])
    Gst_cmpnt_Nm.set(i[2])
    Gst_cmpnt_CntNO.set(i[3])
    Gst_cmpnt_Cmplnt.set(i[4])
Gst_cmpnt_Tabke.bind('<<TreeviewSelect>>',Gst_cmpnt_Tabke_select)
```



```
#-------------------------------------------------------------------------------------------------------------------------
-----
can_widget10 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
Reprt = ImageTk.PhotoImage(Image.open("./assets/pattern.png").resize((1585,955)))
can_widget10.create_image(0,0,anchor=NW,image=Reprt)

 #-----------------------------------------------------------------------------
# can_widget10.place(x=330, y=25)
WhiteBg=ImageTk.PhotoImage(Image.open("./assets/whitebg.png").resize((1540,230)))
can_widget10.create_image(20,185,anchor=NW,image=WhiteBg)
WhiteBg1=ImageTk.PhotoImage(Image.open("./assets/whitebg.png").resize((1540,420)))
can_widget10.create_image(20,510,anchor=NW,image=WhiteBg1)
def label():
   can_widget10.create_text(130+120, 50+20, text="From Date :", font=("Pristina", 35, "bold"),
fill="white")
   can_widget10.create_text(550+120, 50+20, text="To Date :", font=("Pristina", 35, "bold"), fill="white")
label()
Reportdt_in=tkinter.StringVar()
Reportdt_ot=tkinter.StringVar()
ReprtFrmDate=DateEntry(can_widget10,selectmode="day",font=("Cambria",18,"italic"),foreground="blu
e",textvariable=dt_in,width=10,locale='en_US',date_pattern='dd/MM/yyyy') #date_pattern='yyyy-MM-dd'
ReprtFrmDate.place(x=250+120,y=25+20)
ReprtToDate=DateEntry(can_widget10,selectmode="day",font=("Cambria",18,"italic"),foreground="blue
",textvariable=Reportdt_ot,width=10,locale='en_US',date_pattern='dd/MM/yyyy') #date_pattern='yyyy-
MM-dd'
ReprtToDate.place(x=650+120,y=25+20)
def ReprtScrh():
   if messagebox.askyesno("Analytics", "Are You Sure You Want To Seen Analytics"):
      con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Hotel Management Software')
      cur = con.cursor()
      query = f"select count(*) from `check in details` where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
      cur.execute(query)
      m1.configure(amountused=cur.fetchone()[0], bootstyle='primary')
      query = f"select count(*) from `Reservation Details`where`Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
      cur.execute(query)
      m2.configure(amountused=cur.fetchone()[0], bootstyle='danger')
      query = f"select count(*) from `hall reservation` where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
      cur.execute(query)
      m3.configure(amountused=cur.fetchone()[0], bootstyle='success')
      query = f"select count(*) from chkout where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
      cur.execute(query)
      m4.configure(amountused=cur.fetchone()[0], bootstyle='info')
```

```
    import matplotlib.pyplot as plt
    from matplotlib.backends.backend_tkagg import (
      FigureCanvasTkAgg)
    import numpy as np

    query = f"select sum(Payment) from `check in details` where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    ChkInPrc = cur.fetchone()[0]
    if ChkInPrc ==None:
      ChkInPrc =0

    query = f"select sum(Payment) from `Reservation Details` where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    RsvrdPrc = cur.fetchone()[0]
    if RsvrdPrc ==None:
      RsvrdPrc =0

    query = f"select sum(f.`Total Amount`)+sum(l.`Total Amount`)+sum(ld.`Total Amount`) from `food
details`f,`liquor details`l,`laundry details`ld where f.`Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    RmSvrPrc = cur.fetchone()[0]
    if RmSvrPrc ==None:
      RmSvrPrc =0

    query = f"select sum(`Hall Price`) from `hall reservation` where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    HallRsvrdPrc = cur.fetchone()[0]
    if HallRsvrdPrc ==None:
      HallRsvrdPrc =0

    query = f"select sum(`Grand Total`) from Bar_Details where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    BarPrc = cur.fetchone()[0]
    if BarPrc ==None:
      BarPrc =0

    query = f"select sum(`Payment`) from chkout where `Payment Date` between
'{ReprtFrmDate.get_date().strftime('%Y-%m-%d')}'AND'{ReprtToDate.get_date().strftime('%Y-%m-
%d')}';"
    cur.execute(query)
    ChkoutPrc = cur.fetchone()[0]
    if ChkoutPrc ==None:
      ChkoutPrc =0
```

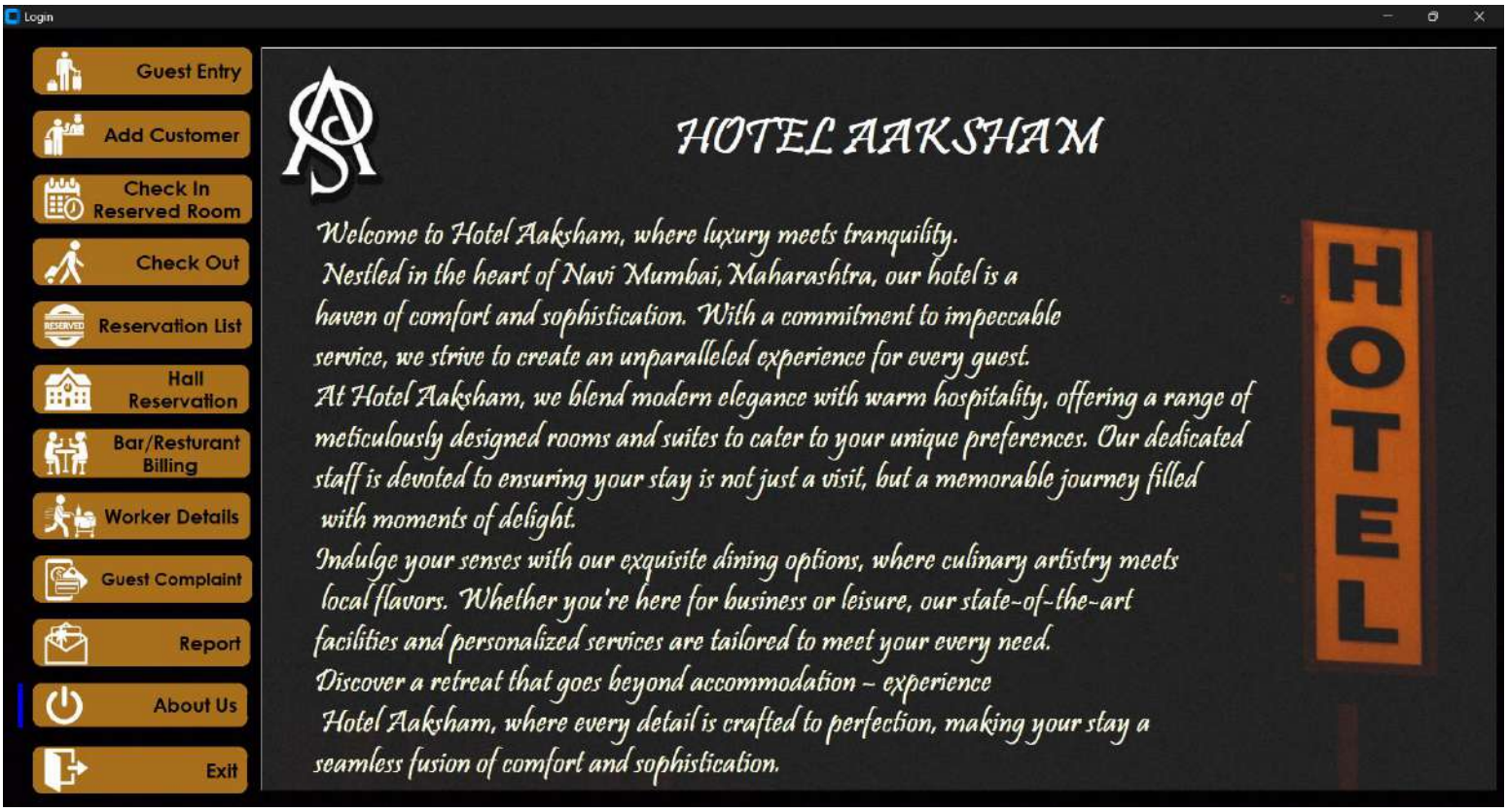
```
    fig = plt.figure(figsize=(7, 4), dpi=100)
    labels = ("Check In", "Room\nReservation", "Room Service", "Hall\nReservation", "Bar\nBilling,",
"Check Out")
    labelpos = np.arange(len(labels))
    y = [ChkInPrc, RsvrdPrc, RmSvrPrc, HallRsvrdPrc, BarPrc, ChkoutPrc]
    plt.bar(labelpos, y, align='center', alpha=1.0)
    plt.xticks(labelpos, labels)
    plt.ylabel('Ammount')
    plt.xlabel('Sources')
    plt.tight_layout(pad=2.2, w_pad=8.5, h_pad=8.1)
    plt.title("Total Revenue Generated")
    plt.xticks(rotation=0, horizontalalignment="center")
    canvasbar2 = FigureCanvasTkAgg(fig, master=can_widget10)
    canvasbar2.get_tk_widget().place(x=700, y=520)
    canvasbar2.draw()

    query = "select Status,count(*) from `room status` group by Status;"
    cur.execute(query)
    RmChrt = []
    RmChrtVcnt = 0
    RmChrtRpir = 0
    RmChrtDrty = 0
    for row, val in cur.fetchall():
        RmChrt.append(row)
        if row == 'Vacant':
            RmChrtVcnt = val
        if row == 'Repair':
            RmChrtRpir = val
        if row == 'Dirty':
            RmChrtDrty = val
    # print(RmChrt)
    # print(RmChrtVcnt)
    # print(RmChrtRpir)
    # print(RmChrtDrty)
    # if len(RmChrt) == 1:
    #     # print("hiii")
    #     RmChrtDrty=0
    #     RmChrtRpir=0
    #     RmChrtVcnt=0
    # elif len(RmChrt) == 2:
    #     # print("hiii")
    #     RmChrtDrty=0
    #     RmChrtRpir=0
    #     RmChrtVcnt=0
    # else:
    #     RmChrtDrty=RmChrt[0][1]
    #     RmChrtRpir=RmChrt[1][1]
    #     RmChrtVcnt=RmChrt[2][1]
    query = "select count(*) from `Reservation Details`;"
    cur.execute(query)
    RmChrtRsvrd = cur.fetchone()[0]
    query = "select count(*) from `check in details`;"
    cur.execute(query)
```

```
      RmChrtOcpd = cur.fetchone()[0]
      # print(RmChrtOcpd)
      import matplotlib.pyplot as plt
      from matplotlib.backends.backend_tkagg import (
        FigureCanvasTkAgg)
      import numpy as np
      if RmChrtOcpd == 0 and RmChrtVcnt == 0 and RmChrtRpir == 0 and RmChrtDrty == 0 and
RmChrtRsvrd == 0:
        pass
      else:
        fig = plt.figure(figsize=(5, 5), dpi=115)
        fig.set_size_inches(4, 3.5)
        labels = ['Occupied', 'Vacant', 'Repair', 'Dirty', 'Reserved']
        sizes = [RmChrtOcpd, RmChrtVcnt, RmChrtRpir, RmChrtDrty, RmChrtRsvrd]
        explode = (0.2, 0, 0, 0, 0)
        plt.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", shadow=True, startangle=50)
        plt.axis('equal')

        canvasbar = FigureCanvasTkAgg(fig, master=can_widget10)
        canvasbar.draw()
        canvasbar.get_tk_widget().place(x=85, y=520)

tkinter.Button(can_widget10, image=Guest_Entry, compound=LEFT,command=ReprtScrh, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Search",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=900+120, y=20+20)
def ReprtExprt():
  if messagebox.askyesno("Print Analytics", "Are You Sure You Want To Print Analytics"):
    import pyautogui
    region = (330, 52, 1555, 960)
    image = pyautogui.screenshot(region=region)
    image.save("D:\python\Project\screenshot.pdf")
    os.system("screenshot.pdf")
tkinter.Button(can_widget10, image=Guest_Entry, compound=LEFT,command=ReprtExprt, fg="Black",
width=120+20, activeforeground="black",activebackground="#a8701d", height=30+5, text="Export",
bg="#a8701d", anchor=W,font=('Century Gothic', 18, "bold"), borderwidth=5,
cursor="hand2").place(x=1100+120, y=20+20)
def label():
  can_widget10.create_text(180, 160, text="No. Of Check In :", font=("Pristina", 30, "bold"),
fill="white")
  can_widget10.create_text(570, 160, text="No. Of Room Reservation :", font=("Pristina", 30, "bold"),
fill="white")
  can_widget10.create_text(1000, 160, text="No. Of Hall Reservation :", font=("Pristina", 30, "bold"),
fill="white")
  can_widget10.create_text(1380, 160, text="No. Of Check Out :", font=("Pristina", 30, "bold"),
fill="white")
  can_widget10.create_text(310,480, text="Room Status :", font=("Pristina", 30, "bold"), fill="white")
  can_widget10.create_text(1050,480, text="Total Revenue :", font=("Pristina", 30, "bold"), fill="white")
label()
con = connector.connect(host='localhost',
                        port='3306',
                        user='root',
                        password='Password',
                        database='Hotel Management Software')
```

```
cur = con.cursor()
import ttkbootstrap as ttk
from ttkbootstrap.constants import *
Image.CUBIC = Image.BICUBIC
query="select count(*) from `check in details`;"
cur.execute(query)
m1 = ttk.Meter(
   can_widget10,
   amountused=cur.fetchone()[0],
   metersize=200,
   meterthickness=30,
   bootstyle=WARNING,metertype=SEMI,
   arcrange=(180),
   arcoffset=(540),
   textfont=['Times',26,'bold'],
   # textright='%',
   subtextfont=['Times',10,'normal']
)
query="select count(*) from `Reservation Details`;"
cur.execute(query)
m1.place(x=80,y=200)
m2 = ttk.Meter(
   can_widget10,
   amountused=cur.fetchone()[0],
   metersize=200,
   meterthickness=30,
   bootstyle=WARNING,metertype=SEMI,
   arcrange=(180),
   arcoffset=(540),
   textfont=['Times',26,'bold'],
   # textright='%',
   subtextfont=['Times',10,'normal']#    textleft='Speed',subtext='Performance',
)
query="select count(*) from `hall reservation`;"
cur.execute(query)
m2.place(x=500,y=200)
m3 = ttk.Meter(
   can_widget10,
   amountused=cur.fetchone()[0],
   metersize=200,
   meterthickness=30,
   bootstyle=WARNING,metertype=SEMI,
   arcrange=(180),
   arcoffset=(540),
   textfont=['Times',26,'bold'],
   # textright='%',
   subtextfont=['Times',10,'normal']
)
m3.place(x=900,y=200)
query="select count(*) from chkout ;"
cur.execute(query)
m4 = ttk.Meter(
   can_widget10,
   amountused=cur.fetchone()[0],
```

```
    metersize=200,
    meterthickness=30,
    bootstyle=WARNING,metertype=SEMI,
    arcrange=(180),
    arcoffset=(540),
    textfont=['Times',26,'bold'],
    subtextfont=['Times',10,'normal']
)
m4.place(x=1290,y=200)
s.theme_use("winnative")
query="select Status,count(*) from `room status` group by Status;"
cur.execute(query)
RmChrt=[]
RmChrtVcnt =0
RmChrtRpir =0
RmChrtDrty =0
for row,val in cur.fetchall():
    RmChrt.append(row)
    if row == 'Vacant' :
        RmChrtVcnt = val
    if row == 'Repair' :
        RmChrtRpir = val
    if row == 'Dirty' :
        RmChrtDrty = val
query="select count(*) from `Reservation Details`;"
cur.execute(query)
RmChrtRsvrd=cur.fetchone()[0]
query="select count(*) from `check in details`;"
cur.execute(query)
RmChrtOcpd=cur.fetchone()[0]
# print(RmChrtOcpd)
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import (
    FigureCanvasTkAgg)
import numpy as np
if RmChrtOcpd ==0 and RmChrtVcnt ==0 and RmChrtRpir == 0 and RmChrtDrty ==0 and RmChrtRsvrd
==0 :
    pass
else:
    fig=plt.figure(figsize=(5, 5), dpi=115)
    fig.set_size_inches (4, 3.5)
    labels=[ 'Occupied','Vacant','Repair','Dirty','Reserved']
    sizes = [RmChrtOcpd,RmChrtVcnt,RmChrtRpir,RmChrtDrty,RmChrtRsvrd]
    explode = (0.2, 0, 0, 0, 0)
    plt.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", shadow=True, startangle=50)
    plt.axis('equal')

    canvasbar=FigureCanvasTkAgg(fig, master=can_widget10)
    canvasbar.draw()
    canvasbar.get_tk_widget().place(x=85,y=520)

query="select sum(Payment) from `check in details`;"
cur.execute(query)
ChkInPrc=cur.fetchone()[0]
```

```
if ChkInPrc == None:
   ChkInPrc = 0

query="select sum(Payment) from `Reservation Details`;"
cur.execute(query)
RsvrdPrc=cur.fetchone()[0]
if RsvrdPrc == None:
   RsvrdPrc = 0

query="select sum(f.`Total Amount`)+sum(l.`Total Amount`)+sum(ld.`Total Amount`) from `food
details`f,`liquor details`l,`laundry details`ld;"
cur.execute(query)
RmSvrPrc=cur.fetchone()[0]
if RmSvrPrc == None:
   RmSvrPrc = 0


query="select sum(`Hall Price`) from `hall reservation`;"
cur.execute(query)
HallRsvrdPrc=cur.fetchone()[0]
if HallRsvrdPrc == None:
   HallRsvrdPrc = 0

query = f"select sum(`Grand Total`) from Bar_Details where `Payment Date`='2023-12-17';"
cur.execute(query)
BarPrc = cur.fetchone()[0]
if BarPrc == None:
   BarPrc = 0

query="select sum(`Payment`) from chkout;"
cur.execute(query)
ChkoutPrc=cur.fetchone()[0]
if ChkoutPrc == None:
   ChkoutPrc = 0

fig=plt.figure(figsize=(7,4),dpi=100)
labels=("Check In", "Room\nReservation", "Room Service", "Hall\nReservation",
"Bar\nBilling,","Check Out")
labelpos=np.arange(len(labels))
y=[ChkInPrc,RsvrdPrc,RmSvrPrc,HallRsvrdPrc,BarPrc,ChkoutPrc]
plt.bar(labelpos, y, align='center', alpha=1.0)
plt.xticks(labelpos, labels)
plt.ylabel('Ammount')
plt.xlabel('Sources')
plt.tight_layout(pad= 2.2,w_pad=8.5, h_pad=8.1)
plt.title("Total Revenue Generated")
plt.xticks(rotation=0, horizontalalignment="center")
canvasbar2= FigureCanvasTkAgg(fig, master=can_widget10)
canvasbar2.get_tk_widget().place(x=700,y=520)
canvasbar2.draw()
```




```
#-------------------------------------------------------------------------------------------------------------------
can_widget11 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget11.place(x=330, y=25)
AboutUsBg = ImageTk.PhotoImage(Image.open("AbtusBg - Copy.jpeg").resize((1585,955)))
can_widget11.create_image(1,1,anchor=NW,image=AboutUsBg)
AboutUsLogo = ImageTk.PhotoImage(Image.open("./assets/Logo.png").resize((140,186)))
can_widget11.create_image(20,20,anchor=NW,image=AboutUsLogo)

can_widget11.create_text(800, 120, text="HOTEL AAKSHAM", font=("Pristina", 50, "bold"),
fill="white")
can_widget11.create_text(670, 580, text="Welcome to Hotel Aaksham, where luxury meets tranquility.\n
Nestled in the heart of Navi Mumbai,Maharashtra, our hotel is a\nhaven of comfort and sophistication. With
a commitment to impeccable\nservice, we strive to create an unparalleled experience for every guest.\nAt
Hotel Aaksham, we blend modern elegance with warm hospitality, offering a range of \nmeticulously
designed rooms and suites to cater to your unique preferences. Our dedicated \nstaff is devoted to ensuring
your stay is not just a visit, but a memorable journey filled\n with moments of delight.\nIndulge your senses
with our exquisite dining options, where culinary artistry meets\n local flavors. Whether you're here for
business or leisure, our state-of-the-art \nfacilities and personalized services are tailored to meet your every
need.\nDiscover a retreat that goes beyond accommodation – experience\n Hotel Aaksham, where every
detail is crafted to perfection, making your stay a \nseamless fusion of comfort and sophistication.",
font=("Pristina",30, "bold"), fill="lightyellow")
```


```
can_widget14 = Canvas(l1,width=1580,height=950,borderwidth=0,bd=0)
# can_widget14.place(x=330, y=25)
AboutOurMmbrBg = ImageTk.PhotoImage(Image.open("AbtusBg - Copy.jpeg").resize((1585,955)))
can_widget14.create_image(1,1,anchor=NW,image=AboutOurMmbrBg)
AboutOurMmbrLogo = ImageTk.PhotoImage(Image.open("./assets/Logo.png").resize((80,100)))
ThnkuLogo = ImageTk.PhotoImage(Image.open("Thanku.png").resize((400,350)))
can_widget14.create_image(850,480,anchor=NW,image=ThnkuLogo)
can_widget14.create_image(20,20,anchor=NW,image=AboutOurMmbrLogo)
can_widget14.create_text(800, 70, text="HOTEL AAKSHAM", font=("Pristina", 50, "bold"), fill="white")
can_widget14.create_text(750, 200, text="In the context of hotel software development, a team typically
consists of professionals with diverse skills who work collaboratively to \ndesign, develop, implement, and
maintain software solutions tailored to the needs of the hotel industry.\nHere are some key roles and
responsibilities that might be present in a hotel software development team:", font=("Pristina",25, "bold"),
fill="lightyellow")
can_widget14.create_text(600, 370, text="1. Saksham Jaiswal ( Software Developer ) :\n- Front-end
Developers: Design and implement the user interface and user experience of the software.\n- Back-end
Developers: Work on server-side logic, databases, and application integration.\n- Full-stack Developers:
Have expertise in both front-end and back-end development.", font=("Pristina",23, "bold"),
fill="lightyellow")
can_widget14.create_text(400, 560, text="2. Aditya Jaiswal ( UI/UX Designers ) :\n- Create visually
appealing and user-friendly interfaces.\n- Consider user experience and accessibility in the design process.\n-
Design the overall structure of the software system.", font=("Pristina",23, "bold"), fill="lightyellow")
can_widget14.create_text(390, 730, text="3. Archana Dubey ( Database Administrators ( DBAs ) ) :\n-
Design, implement, and maintain the database architecture.\n- Ensure data integrity, security, and optimal
performance.", font=("Pristina",23, "bold"), fill="lightyellow")
can_widget14.create_text(470, 880, text="4. Yash Jaiswal ( Technical Writers ) : \n- Create documentation
for end-users, developers, and system administrators.\n- Ensure that manuals and guides are clear and
comprehensive.",font=("Pristina",23, "bold"), fill="lightyellow")

time.sleep(0.1)
pygame.mixer.music.play()
app.mainloop()
```

```
# ---------------------------------------  Mysql Query ----------------------------------------------------------------
create database `Hotel Management Software`;
use `Hotel Management Software`;
create table `Customer Details`(
`S No.` char(5) unique not null,
`Guest ID` char(5) primary key,
`Guest Name` varchar(20) not null,
`Gender` varchar(11) not null,
`Religion` varchar(7) not null,
`Address` varchar(25) not null,
`City` varchar(20) not null,
`Country` varchar(20) not null,
`Contact No.` varchar(13) not null,
`ID Type` varchar(20) not null,
`ID Number` varchar(15) unique not null,
`Status` varchar(10) not null,
`Email ID` varchar(50) not null unique
);
create table `Check In Details`(
`Guest ID` char(5) primary key,
`Guest Name` varchar(20) not null,
`Room No` char(6) not null,
`Room Type` varchar(15) not null,
`Day In` date not null,
`Day Out` date not null,
`Room Price` int(10) not null,
`Payment`int(10) not null,
`Payment Mode` varchar(15) not null,
`Payment Date` date not null,
`Status` varchar(10) not null
);
create table `Room Service Details`(
`Guest ID` char(5) not null,
`Guest Name` varchar(20) not null,
`Room No` char(6) not null,
```

```
`Bill No.` varchar(15) primary key,
`Bill Date` date not null,
`Payment Mode`varchar(15) not null
);
create table `Food Details`(
`Bill No.` varchar(15),
`Food Name` varchar(20) not null,
`Rate` int(5) not null,
`Quantity` char(3),
`Amount`int(6) not null,
`Discount` decimal(3,2) not null,
`SGST` decimal(3,2) not null,
`CGST` decimal(3.2) not null,
`Total Amount` int(6) not null
);
create table `Liquor Details`(
`Bill No.` varchar(15),
`Liquor Name` varchar(25) not null,
`Rate` int(5) not null,
`Quantity` char(3),
`Amount`int(6) not null,
`Discount` decimal(3,2) not null,
`SGST` decimal(3,2) not null,
`CGST` decimal(3,2) not null,
`Total Amount` int(6) not null);
create table `Laundry Service Details`(
`Guest ID` char(5) not null,
`Guest Name` varchar(20) not null,
`Room No` char(6) not null,
`Bill No.` varchar(15) primary key,
`Bill Date` date not null,
`Payment Mode`varchar(15) not null
);
create table `Laundry Details`(
`Bill No.` varchar(15),
```

```
`Service Name` varchar(25) not null,
`Rate` int(5) not null,
`Quantity` char(3),
`Amount`int(6) not null,
`Discount` decimal(3,2) not null,
`SGST` decimal(3,2) not null,
`CGST` decimal(3,2) not null,
`Total Amount` int(6) not null);
create table `Room Status`(
`Room No.` varchar(15) not null,
`Room Type` varchar(25) not null,
`Status` varchar(10) not null);
create table `Reservation Details`(
`Guest ID` char(5) primary key,
`Guest Name` varchar(20) not null,
`Room No` char(6) not null,
`Room Type` varchar(15) not null,
`Day In` date not null,
`Day Out` date not null,
`Room Price` int(10) not null,
`Payment`int(10) not null,
`Payment Mode` varchar(15) not null,
`Payment Date` date not null,
`Status` varchar(10) not null
);
create table `Hall Reservation`(
`Guest ID` char(5) primary key,
`Guest Name` varchar(20) not null,
`Address` varchar(25) not null,
`Contact No.` varchar(13) not null,
`ID Type` varchar(20) not null,
`ID Number` varchar(15) unique not null,
`Hall Number` char(4) unique not null,
`From Date` date not null,
`To Date` date not null
```

```
);
create table `Hall`(
`Hall No` char(5) primary key
);
create table `Hall Customer Details`(
`Guest Id`char(5),
`Guest Name`varchar(50) not null,
`Address`varchar(50) not  null,
`Contact No`char(13) not  null,
`Id Type`varchar(20) not  null,
`Id No`varchar(30) not  null,
`Email Id`varchar(50) not  null,
`Image` varchar(100) not  null,
`Hall No`char(5) not  null,
`Date In`date not  null,
`Date Out`date not  null,
`Hall Price`int not  null,
`Total Amount`int not  null,
`Advance Payment`int not  null,
`Payment`float not  null,
`Payment Mode`varchar(20) not  null,
`Payment Date`date not  null
);
create table ChkOut(
`Guest Id`char(5) primary key,
`Guest Name`varchar(50) not null,
`Gender`varchar(10)not null,
`Religion`varchar(15) not null,
`Address`varchar(50) not  null,
`City`varchar(20) not  null,
`Country`varchar(20) not  null,
`Contact No`char(13) not  null,
`Email Id`varchar(50) not  null,
`Id Type`varchar(20) not  null,
`Id No`varchar(30) not  null,
```

```
`Room No`char(5) not  null,
`Date In`date not  null,
`Date Out`date not  null,
`Room Type`varchar(10) not  null,
`Room Charges`int not  null,
`Discount %`float not  null,
`SGST %`float not  null,
`CGST %`float not  null,
`Grand Total`float not  null,
`Reservation Payment`int not  null,
`Advance Payment`int not  null,
`Remaining Price`float not  null,
`Payment Mode`varchar(20) not  null,
`Payment`float not  null
);
 create table Bar_Details(
`Bill No.`char(5) not null,
`Guest Id`char(5) not null,
`Guest Name`varchar(50) not null,
`Gender`varchar(10)not null,
`Contact No`char(13) not  null,
`Food Name`varchar(10) not  null,
`Rate`int not  null,
`Quantity`int not  null,
`Discount %`float not  null,
`SGST %`float not  null,
`CGST %`float not  null,
`Grand Total`float not  null,
`Payment Date`date not  null
);
create table Worker_Details(
`Worker Id`char(5) primary key,
`Worker Name`varchar(50) not null,
`Gender`varchar(10)not null,
`Religion`varchar(15)not null,
```

```
`Address`varchar(50) not null,
`Contact No`char(13) not  null,
`Date Of Joining`date not  null,
`Id Type`varchar(10) not  null,
`Id No.`varchar(13) not  null,
`Department`varchar(15) not  null
);
create table Complaint(
`Guest Id`char(5) primary key,
`Guest Name`varchar(50) not null,
`Contact No`char(13) not  null,
`Complaint`varchar(50) not  null
);
```

CONCLUSION

In conclusion, hotel management software plays a crucial role in streamlining and enhancing various aspects of the hospitality industry. Its multifaceted capabilities contribute to improved operational efficiency, guest satisfaction, and overall business success.

hotel management software is an indispensable tool for modern hotels aiming to stay competitive, deliver exceptional guest experiences, and optimize their overall operations. Its multifunctional capabilities make it an integral part of the evolving landscape of the hospitality industry.

BIBLIOGRAPHY

- *www.freepik.com*
- *www.google.com*
- *www.leonardo.com*
- *microsoft-copilot*
- *www.d-id.com*