A MAJOR PROJECT REPORT

on

HOTEL MANAGEMENT SYSTEM USING PYTHON GUI

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

THOTA JANU SRI - O180182

Under the Guidance of

Mrs. J.Mrudhula M.Tech

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

Ongole campus

2023-2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

ONGOLE CAMPUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled “**HOTEL MANAGEMENT USING PYTHON”** submitted by **THOTA JANU SRI**(**O180182)** in partial fulfillment of the requirements for the award of the degree of the Bachelor of Technology in Computer Science and Engineering in **RGUKT Ongole** is a record of bona fide work carried out under my guidance and supervision during the academic year 2023-2024.

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Mrs. J.Mrudhula M. Tech,
Assistant Professor,
Department of CSE,
RGUKT ONGOLE.

Mr. B. SAMPATH BABU M. Tech (Ph. D),
Head of Department,
Department of CSE,
RGUKT ONGOLE.

APPROVAL SHEET

The report entitled **HOTEL MANAGEMENT SYSTEM USING PYTHON** by **THOTA JANUSRI** (O180182) is approved for the degree of the Bachelor of Technology in Computer Science & Engineering.

Examiners: ________________________

____

____

Supervisor(s):

____

____

____

Date: ___________________

Place: __________________

ACKNOWLEDGEMENT

It is our privilege to express a profound sense of respect, gratitude and indebtedness to our guide **Mrs. J.MRUDHULA**, Assistant Professor, Department of Computer Science and Engineering RGUKT Ongole, for her indefatigable inspiration, guidance, cogent, discussion, constructive criticisms and encouragement throughout the dissertation work.

We express our sincere gratitude to **Mr. B. SAMPATH BABU,** Assistant Professor & Head of Department of Computer Science and Engineering, RGUKT Ongole for his suggestion motivations and co-operation for the successful completion of the work.

We extend our sincere thanks **Prof. B. JAYA RAMI REDDY** sir, Director, RGUKT Ongole for his encouragement.

With sincere Regards,

THOTA JANU SRI,

(O180182)

Date: ___________________

DECLARATION

We hereby declare that the project work entitles **“HOTEL MANAGEMENT SYSTEM USING PYTHON”** submitted to the **RGUKT Ongole** in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology (B Tech) in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mrs. J.MRUDHULA,** Assistant Professor and this project work have not been submitted to any university for the award of any other degree or diploma.

Signature:

THOTA JANU SRI (O180182)

Date: ___________________

Place: ___________________

ABSTRACT

As ‘Technology’ is improving day by day, all fields of industry are trying to be more globalized and get succeed by using new technologies and strategies that attract people. The hotel industry is a highly competitive field, and managing day-to-day operations can be challenging task. To address this issue, a Hotel Management System can be implemented to streamline operations and improve customer satisfaction. Here we are designing this system to automate the hotel’s various processes, such as managing room bookings, check-ins and check-outs, managing room inventory, billing and payment processing and generating reports.

In this project, we propose the development of the system using Python programming language. The system will have a user-friendly graphical user interface(GUI) using TKinter that will allow hotel staff to manage bookings, check-in and check-out guests, generate reports and manage inventory efficiently.

The main motto of this system is to provide a comprehensive solution for managing hotel operations and enhancing customer experience.

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

The Hotel Management System using Python is a comprehensive and sophisticated software solution crafted to streamline various hotel operations. With an intuitive interface and robust functionalities, this system is tailored to meet the unique needs of the hospitality industry, providing hoteliers with a powerful tool to efficiently manage their establishments.This system encompasses features such as room management, reservation management, billing and invoicing, customer management, and staff management. Python programming language has been employed to develop the system, offering flexibility, reliability, and scalability. The design ensures seamless integration with other systems and technologies, enhancing its adaptability.

Key features include the efficient management of room inventory, reservation processes, and billing generation. The system enables personalized customer service by maintaining a centralized database of guest profiles and preferences. Staff management is facilitated through task scheduling and access controls, ensuring data security.The technology stack includes Python for its versatility, readability, and extensive libraries, along with a robust Database Management System (DBMS) for efficient data handling. The system is designed for easy integration with third-party systems such as POS (Point of Sale) and CRM (Customer Relationship Management) software.

In conclusion, the Hotel Management System using Python is a strategic investment for hotels seeking operational efficiency, exceptional guest experiences, and revenue maximization. Its user-friendly interface and advanced features position it as a valuable asset for staying competitive in the hospitality industry.

1.1 MOTIVATION:

Building a hotel management system involves analyzing various aspects of hotel operations like to manage reservations, track room availability, handle check-ins and check-outs and the system will be very useful for the hotels to integrate it into them. Developing this using Python provides an opportunity to apply programming skills to solve real-world problems in the hospitality industry , integrate various skills like database management and user interface design. Its user-friendly interface and advanced features position it as a valuable asset for staying competitive in the hospitality industry.

1.2 PROBLEM DEFINITION

To develop a modular and scalable hotel management system using python that can efficiently manage and automate various tasks involved in the day-to-day operations of a hotel, such as room management, reservation management, billing and payment, reporting and analytic s, and staff management. The system should provide a user-friendly interface and prioritize data security and confidentiality while enabling easy maintenance and future enhancements. The current manual system for managing hotel operations is proving to be inefficient and error-prone, leading to operational bottlenecks, poor customer service, and compromised data security. To address these challenges, there is a pressing need for the development of a Hotel Management System (HMS) using Python.

1.3 OBJECTIVE OF THE PROJECT:

The objective of the Hotel Management System project using Python is to develop a comprehensive software solution that automates various hotel operations, including room management, reservation processes, billing and payment, reporting and analytics, and staff management. The primary goals are to enhance operational efficiency, improve customer service, and ensure data security. The system aims to provide a user-friendly interface, prioritize confidentiality, and facilitate easy maintenance and future enhancements. Through modularity and scalability, the project intends to address the limitations of the current manual system, offering a seamless transition to a more efficient and technologically advanced solution for the hotel industry.

It is a strategic investment for hotels seeking operational efficiency, exceptional guest experiences, and revenue maximization. Its user-friendly interface and advanced features position it as a valuable asset for staying competitive in the hospitality industry.

CHAPTER-2
LITERATURE REVIEW

There are many similar paper citiations regarding “Hotel Management System” in which some of them we reviewed and took reference to develop our project.They are:

In **2014, Room Allocation for Hotel Revenue Management Considering consumer choice behavior** by **Chaoyong Qin** Revenue management has been widely used in hotel management. However, researches do not take customer satisfaction into account. In this paper, utility function is construct based on consumer behavior theory. A consumer discrete choice model is established after that. Variable precision rough set then is employed to reduce the factors that influence consumer choice and to calculate the weights. A model of hotel allocation is established thereafter. Finally, the model is validated by an example, the results show validity of the model.

In **2016, Hotel’s online booking segmentation for heterogenous customers** by **Z.W. Miao, T.Wei, Y.Q.Lan** cited The online reservation system (ORS) is widely used in the hospitality industries. This paper studies the hotel's optimal pricing policies for two types of contracts: booking without prepayment and booking with prepayment. The hotel adopts these two contracts to segment heterogenous customers through the ORS. We get the hotel's three possible pricing strategies and the corresponding payoffs. It is found that the hotel can segment the customers effectively if it employs two types booking contracts by proper pricing strategies. Through comparison, the results show that when the variable operational cost is high, the hotel will conduct the second contract; otherwise, the hotel will conduct the first contract.

In **2018, Design and Implementation of Hotel Room Information Management System Based on Kendo UI Front-End Framework,** by **Mei Shan-Shan, Shan Chun, Xue Jing-Feng** cited In the rapid development of society today, along with the popularization of computer technology and the extensive application of the Internet, all walks of life are applying the computer management system to their own enterprises. As a member of the service industry, the hotel industry is also developing rapidly. This article describes a management platform for hotel room's online booking and guest information using kendo UI front-end framework design. Using SSM (Spring, Spring MVC, MyBatis) framework to deal with background programs. The overall mission of system development is to ensure office staff can quickly and easily complete the hotel guest room management task, providing better and faster service for residents. The interface is simple and his operation is easy to maintain.

In **2018, Analysis and Design og Five-Star Hotel Management Information System Based on UML** by **Guo Chen** With the development of the tourism industry, the competition of the hotel industry is becoming more and more intense. Effective integration of all kinds of resources inside the hotel can improve the efficiency of labor in the hotel, reduce the transaction cost, and improve the operating profit of the hotel. Besides, the innovation of management is an important factor to form the competitiveness of the hotel. Therefore, the idea of catering management system is put forward.

In **2019, Design and Implementation of Hotel Room Management System** by **Wei, Zhengwei Lou** cited that Hotel room management system through the collection of information, transmission, sorting, processing, maintenance and use, improve the management level and efficiency, so as to achieve the automation, standardization and humanization of hotel management.According to the characteristics of hotel room management, this system uses Java Swing technology and Mysql database to connect, and develops under the development tool of Myeclipse. The system has the functions of reservation, checkout and settlement. With this system, hotel room can be managed conveniently and quickly, which greatly improves the efficiency of processing and makes management more modern.

In **2020, Enhanced Hotel Booking Application using PEGA** by **R Dhanagopa;,N Archana, R Menaka** cited Flying colors is a Hotel Booking Application; this application provides different staying solutions to customers, it is to book customers a hotel room online based on date and location along with the type of hotel. This architecture is based on agile development which is specially designed to work on multiple regions, channels, and quick future enhancements. PEGA7 Business Process Management (BPM) Tool which enables significant enhancement in building the application and data management experiences mobile functionality, user experience, and analytics.

In **2021, Design and Implementation of Hotel Reservation Management Platform based on SOA Framework** by **Ximei Lv** To solve the problems of hotel information island, complex network reservation process and inconvenient management, this paper analyzes and studies the hotel business management, and designs an extreme reservation management platform of face-to-face service combined with SOA framework and Webservice system. The system adopts unified channel entrance to form centralized management and distributed operation. Base on the B/S architecture, we adopt the development framework SSH and front-end development technology (jQuery, bootstrap, etc.), combined with the lightweight database Mysql, to achieve seamless heterogeneous platform.

In **2021, Hotel Management Platform Design Based on Virtual Reality** by **Ding Xueyan** cited The administrator can complete the basic hotel design and management functions through 3D graphical user interface, realize the realistic display of complex network information, and set and present it to users through the secondary development function of VRGIS platform. At the same time, it introduces the implementation process of key technologies such as data preprocessing, scene fusion, Java structure and database connection, which provides hotel managers with an easy planning and entity operation interface.

In **2021, Review on Hotel Management System** by **Avneesh Pathak, Arun Snghal, Bibhas Kumar Rana** cited the operation of the hotel computer management has become a very important task.

These articles give a brief introduction of a hotel management system, based on software engineering methods to perform performance analysis, needs analysis and design. The whole system is divided into separate modules and introduces the function, while providing a set of conditions and logical structure of data management database. These analyses and guidelines basically meet the structure hotel management requirements based on information technology, to improve the quality of service and efficiency of hotel managers.

In **2022, CRM Hotel Management System and Intelligent Information Push Based on BP Neural Network** by **Li Tang** cited the popularization of the Internet has made the basic conditions for realizing the informationization of hotel management. The CRM hotel management system (HMS) can efficiently process and analyze massive passenger registration data, extract passenger information hidden in big data, use data warehouse and data mining technology to analyze and integrate, and make decisions for hotel operations. This article uses experimental analysis and data analysis to better understand the performance of the CRM HMS, so as to improve the user experience of hotel room residents. According to the experimental results, when the number of concurrent operators is less than 120, the average response time of the system does not exceed 3s. Therefore, the system has better performance, can meet the basic needs of users, and give users a better experience

CHAPTER-3
METHODOLOGY

3.1 EXISTING SYSTEMS

There are many existing systems similar to our project we are developing. Some of the websites that are similar are ‘Odoo(formely Open Erp), Django Hotel, InnGrid, Python Hotel Management, Hotel Management System by Shalini Chauhan, Hotel Management System using Python and MySQL ,Hotel Reservation System using Flask etc...’. All these have their own unique features which are helpful for hotels to use them for their management. There are having some of the features like Reservation System, room assignment, billing, guest management and reporting, check-ins and check-outs.

3.2 PROPOSED SYSTEM

Taking references from all the existing projects, here we are proposing a new application that provides basic User Authentication to access the system, Room Management that allow users to manage room details and users are able to add, edit and delete room information, Reservation Management that provide functionality to create, modify, and cancel room reservations, Check-in and check-out that enable users to handle guest check-in and check-out processes. This includes updating room status, generating room keys, capturing arrival and departure details, and handling room inspections. Data backup and security that Implement regular data backups and ensure the system has proper security measures in place to protect guest and hotel data. Billing and Invoicing that implement a billing system to generate invoices for guests, including room charges, additional services, and taxes. Track payment status and provide options for cash, credit card, or other payment methods.

3.3 SOFTWARE REQUIREMENT SPECIFICATION

My project was created using the Python programming language. Here I’ve used It provides a set of tools for creating desktop applications with graphical user interfaces. Tkinter is based on the Tk GUI toolkit and is included with most Python installations. It allows developers to create windows, dialogs, buttons, textboxes, and other GUI elements to build interactive applications.Tkinter library for GUI which serves a effective interaction with backend system here it is MySQL.

3.3.1 TECHNOLOGIES USED:

PYTHON:

In the "Text summarization and Feedback Classification using Python" project, Python's Graphical User Interface capabilities is employed in our project to enhance and manage hotel operations. A part form that python is a high-level, versatile, and dynamically typed programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is widely used for various purposes, including web development, data analysis, artificial intelligence, machine learning, scripting, and more.

TKINTER:

Tkinter is the standard GUI (Graphical User Interface) library for Python. It provides a set of tools for creating desktop applications with graphical interfaces. Tkinter is easy to learn and widely used due to its simplicity and effectiveness.Some basic components of Tkinter is as follows

Widgets: Tkinter provides various widgets (GUI components) like buttons, labels, entry widgets, etc., that you can use to create the graphical interface.

Geometry Managers: Tkinter uses geometry managers to control the placement and sizing of widgets within a window. Common geometry managers include pack(), grid(), and place().

Event Handling: Tkinter allows you to define functions that get triggered in response to events, such as button clicks or keypresses.

MYSQL:

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and manipulating structured data. It is known for its speed, reliability, and ease of use, making it one of the most popular database systems in the world. MySQL is often used in conjunction with web applications and is a crucial component of the LAMP (Linux, Apache, MySQL, PHP/Python/Perl) and MEAN (MongoDB, Express.js, AngularJS, Node.js) stacks.

Here I’ve used PyMySQL which is a python module that serves as a connector between Python applications and MySQL databases. It provides a programming interface for Python programs to communicate with MySQL servers, enabling the execution of SQL queries, data retrieval, and database manipulation.

3.3.2 MODULES

- Sign Up and Login
- Home Page
- Services
- Booking
- Check-out
- Ordering food
- Add Rooms
- Checking details
- Feedback

MODULES DESCRIPTION

Sign up& Login:

Here, user can login into our website "Hotel Management System", if one is new to our system they can register into our website by creating an account.

Home Page:

The website home page consists a glance of description about our system, what it consists of, and a brief introduction with services we are providing and also the contact information ,about us.

Services:

Services we are providing in the hotel management system are Room booking service by adding their details and check-out finally the feedback section by user.

Coming to the part of admin, he can add new rooms and also check all the details regarding booking, rooms, and feedback.

Booking:

User can book room of his choice by adding customer details before and can automatically generate bill and view details.

Check-out:

User can check-out from the hotel by simply opening the checkout section and entering the details like Reference number, Mobile number and Name.

Ordering food:

Customers who wish to have their food in the hotel itself can also order from our website where one can choose between breakfast , lunch and dinner.A brief menu will be displayed so that customer can know what are the food items currently available.

Feedback:

User can give his valuable feedback by submitting the form created in this module.

Add Rooms:

Admin can add, delete and update rooms in the hotel by using this module.

Checking Details:

Admin can check Customer details, Booking details, Room details and also feedbacks submitted.

3.4 SAMPLE CODE

```
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox
import pymysql
import ast
from tkinter import ttk
import mysql.connector
import random
from tkinter import ttk
from time import strptime
from datetime import datetime

root=Tk()
root.title("Hotel Management System Using Python")
root.geometry('1370x760')
my_img=ImageTk.PhotoImage(Image.open("C:\\Users\\Rgukt\\Desktop\\Hotel Management system\\minimages\\main.png"))
my_l=Label(image=my_img)
my_l.pack()
def addroom():
    root=Toplevel()
    root.geometry('1366x768')
    root.title("Adding Rooms")
    my_img=PhotoImage(file("C:\\Users\\Rgukt\\Desktop\\Hotel Management system\\minimages\\addrooms.png")
    k=Label(root,image=my_img).pack()
    logo=PhotoImage(file("C:\\Users\\Rgukt\\Desktop\\Hotel Management system\\minimages
\\LOGO.png")

b1=Button(root,image=logo,bd=0,bg="#050a30",activebackground="#050a30",cursor="hand2",comm
and=home).place(x=30,y=15)

    #---functions---
    def reset():
        var_floor.set("")
        var_room.set("")
        var_roty.set("")

    def delete_data():
            mDelete=messagebox.askyesno("Hotel Management System","Do you really want to delete this
customer",parent=root)
            if mDelete>0:

conn=mysql.connector.connect(host="localhost",username="root",password="Janu@1477",database="
userdata")
                my_cursor=conn.cursor()
                query="delete from details where RoomNo=%s"
                value=(var_room.get(),)
                my_cursor.execute(query,value)

            else:
                if not mDelete:
                    return
            conn.commit()
            fetch_data()
            conn.close()
    def update_data():
            if var_floor.get()=="":
                messagebox.showerror("Error","Please enter room number",parent=root)
            else:
```

```
conn=mysql.connector.connect(host="localhost",username="root",password="Janu@1477",database="
userdata")
            my_cursor=conn.cursor()
            my_cursor.execute("update details set Floor=%s,Roomtype=%s where
RoomNo=%s",(var_floor.get(),var_roty.get(),var_room.get()))

            conn.commit()
            fetch_data()
            conn.close()
            messagebox.showinfo("Update","Room details has been updated Successfully",parent=root)
    def get_cursor(event=""):
        cursor_row=room_table.focus()
        content=room_table.item(cursor_row)
        row=content["values"]

        var_floor.set(row[0])
        var_room.set(row[1])
        var_roty.set(row[2])

    def fetch_data():

conn=mysql.connector.connect(host="localhost",username="root",password="Janu@1477",database="
userdata")
        my_cursor=conn.cursor()
        my_cursor.execute("select * from details")
        rows=my_cursor.fetchall()
        if len(rows)!=0:
            room_table.delete(*room_table.get_children())
            for i in rows:
                room_table.insert("",END,values=i)
        conn.commit()
        conn.close()
    def add_data():
        if var_floor.get()=="" or var_roty.get()=="":
            messagebox.showerror("Error","All fields are required",parent=root)
        else:
            try:

conn=mysql.connector.connect(host="localhost",username="root",password="Janu@1477",database="
userdata")
                my_cursor=conn.cursor()
                my_cursor.execute("insert into details
values(%s,%s,%s)",(var_floor.get(),var_room.get(),var_roty.get()))

                conn.commit()
                fetch_data()
                conn.close()
                messagebox.showinfo("success","Room added",parent=root)

            except Exception as es:
                messagebox.showwarning("Warning",f"Something went wrong:{str(es)}",parent=root)



    #---variables----
    var_floor=StringVar()
    var_room=StringVar()
    var_roty=StringVar()

    #-------------customer details frame----------
    lleft=LabelFrame(root,bd=1,relief=RIDGE,text="Adding New Rooms",font=("times new
roman",18,'bold'),padx=2,pady=5,bg="#fbf5e9")
    lleft.place(x=70,y=150,width=425,height=510)

    #-------labels and entry---------------------
    #customer contact
    lcustcon=Label(lleft,text="Floor:-",font=("times new
roman",12,'bold'),padx=2,pady=7,bg="#fbf5e9")
```

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

    enty_con=ttk.Entry(lleft,width=20,font=("times new roman",13,'bold'),textvariable=var_floor)
    enty_con.grid(row=0,column=1,padx=3,sticky=W)

    #checkin date
    lcustcheckin=Label(lleft,text="Room No:-",font=("times new roman",12,'bold'),padx=2,pady=7,bg="#fbf5e9")
    lcustcheckin.grid(row=1,column=0,sticky=W)

    enty_checkin=ttk.Entry(lleft,width=20,font=("times new roman",13,'bold'),textvariable=var_room)
    enty_checkin.grid(row=1,column=1,padx=3)
    #checkout date
lcustcheckout=Label(lleft,text="Room Type:-",font=("times new roman",12,'bold'),padx=2,pady=7,bg="#fbf5e9")
fetch_data()
    root.mainloop()



def roombooking():
    root=Toplevel()
    root.geometry('1366x768')
    root.title("Room Booking")
    my_img=PhotoImage(file=("C:\\Users\\Rgukt\\Desktop\\Hotel Management system\\minimages\\roombook.png")
    k=Label(root,image=my_img).pack()
    logo=PhotoImage(file=("C:\\Users\\Rgukt\\Desktop\\Hotel Management system\\minimages\\LOGO.png")

b1=Button(root,image=logo,bd=0,bg="#050a30",activebackground="#050a30",cursor="hand2",command=home).place(x=30,y=15)
```

```
    #---functions---
    def search():

conn=mysql.connector.connect(host="localhost",username="root",password="Janu@1477",database="
userdata")
            my_cursor=conn.cursor()
            my_cursor.execute("select * from room where "+str(search_var.get())+" LIKE
'%"+str(txt_search.get())+"%'")
            rows=my_cursor.fetchall()
            if len(rows)!=0:
                room_table.delete(*room_table.get_children())
                for i in rows:
                    room_table.insert("",END,values=i)
                conn.commit()
            conn.close()
    def total():
        indate=var_checkin.get()
        outdate=var_checkout.get()
        indate=datetime.strptime(indate,"%d/%m/%Y")
        outdate=datetime.strptime(outdate,"%d/%m/%Y")
        k=(outdate-indate).days
        if k>0:
            var_nofdays.set(k)
        else:
            messagebox.showerror("Error","Please enter valid dates",parent=root)
        d={"Single":500,"Double":800,"Luxury":1000,"Duplex":1500}
        if(var_meal.get()=="BreakFast"):
            q1=float(300)
            q2=float(d[(var_roty.get()[4:])])
            q3=float(var_nofdays.get())
            q4=float(q1+q2)
            q5=float(q3+q4)
            tax="Rs."+str("%.2f"%((q5)*0.1))
            st="Rs."+str("%.2f"%((q5)))
            tt="Rs."+str("%.2f"%(q5+((q5)*0.1)))
            var_tax.set(tax)
            var_stotal.set(st)
            var_total.set(tt)
        elif(var_meal.get()=="Lunch"):
            q1=float(500)
            q2=float(d[(var_roty.get()[4:])])
            q3=float(var_nofdays.get())
            q4=float(q1+q2)
            q5=float(q3+q4)
            tax="Rs."+str("%.2f"%((q5)*0.1))
            st="Rs."+str("%.2f"%((q5)))
            tt="Rs."+str("%.2f"%(q5+((q5)*0.1)))
            var_tax.set(tax)
            var_stotal.set(st)
            var_total.set(tt)
        elif(var_meal.get()=="Dinner"):
            q1=float(400)
            q2=float(d[(var_roty.get()[4:])])
            q3=float(var_nofdays.get())
            q4=float(q1+q2)
            q5=float(q3+q4)
            tax="Rs."+str("%.2f"%((q5)*0.1))
            st="Rs."+str("%.2f"%((q5)))
            tt="Rs."+str("%.2f"%(q5+((q5)*0.1)))
            var_tax.set(tax)
            var_stotal.set(st)
            var_total.set(tt)
root.mainloop()
```

3.5 UML DIAGRAMS:

The abbreviation UML stands for Unified Modelling Language. In the area of object-oriented software engineering, UML is a general purpose modeling language that has been standardized. The Object Management Group oversees and developed the standard.

The creation of a common modeling language for object-oriented software engineering is a key objective of UML. UML now consists of two main parts: a notation and a meta-model. In the future, UML might also be coupled with or added to in the form of a method or process. The Unified Modeling Language is a standard language used for business modeling, non-software systems, and specifying, building, and documenting the artifacts of software systems.

UML, or Unified Modeling Language, is the acronym. UML is a general purpose modeling language that has been standardized in the field of object-oriented software engineering. The standard was created and is supervised by the Object Management Group.

The following are the main objectives in the UML's design:

1. Offer users an expressive visual modeling language that is ready to use so they can create and trade meaningful models.
2. Offer methods for specialization and extendibility to expand the fundamental ideas.
3. Be unreliant on specific development methodologies and programming languages.
4. Promote the commercial expansion of OO tools.
5. Encourage the integration of best practices and support higher level development ideas like collaboration, frameworks, patterns, and components.

The goal of a UML Diagram, which is based on the UML (Unified Modeling Language), is to graphically represent a system together with its primary players, roles, actions, objects, or classes in order to better understand, edit, maintain, or document system-related information. Structural and behavioral UML diagrams make up the UML diagrams.

STRUCTURAL UML DIAGRAMS:

Static representations of a system's structure are shown in structural diagrams. It is frequently employed in software architecture documentation. Developers and stakeholders can better understand and convey the system architecture, linkages, and interactions between various classes thanks to these diagrams, which provide a visual depiction of a system's static structure. They are:

Class Diagram

Object Diagram

Component Diagram

Composite Structure Diagram

Deployment Diagram

Package Diagram

Profile Diagram

BEHAVIOURAL UML DIAGRAM:

A dynamic picture of a system or the behavior of a system, which explains the operation of the system, is provided by behavioral diagrams. These illustrations aid in explaining how objects interact, react to circumstances, and change states. They support the design, analysis, and validation of the system, ensuring that it operates as intended and satisfies the required specifications. There are seven diagrams. They are:

Use case Diagram

Sequence Diagram

Activity Diagram

State Machine Diagram

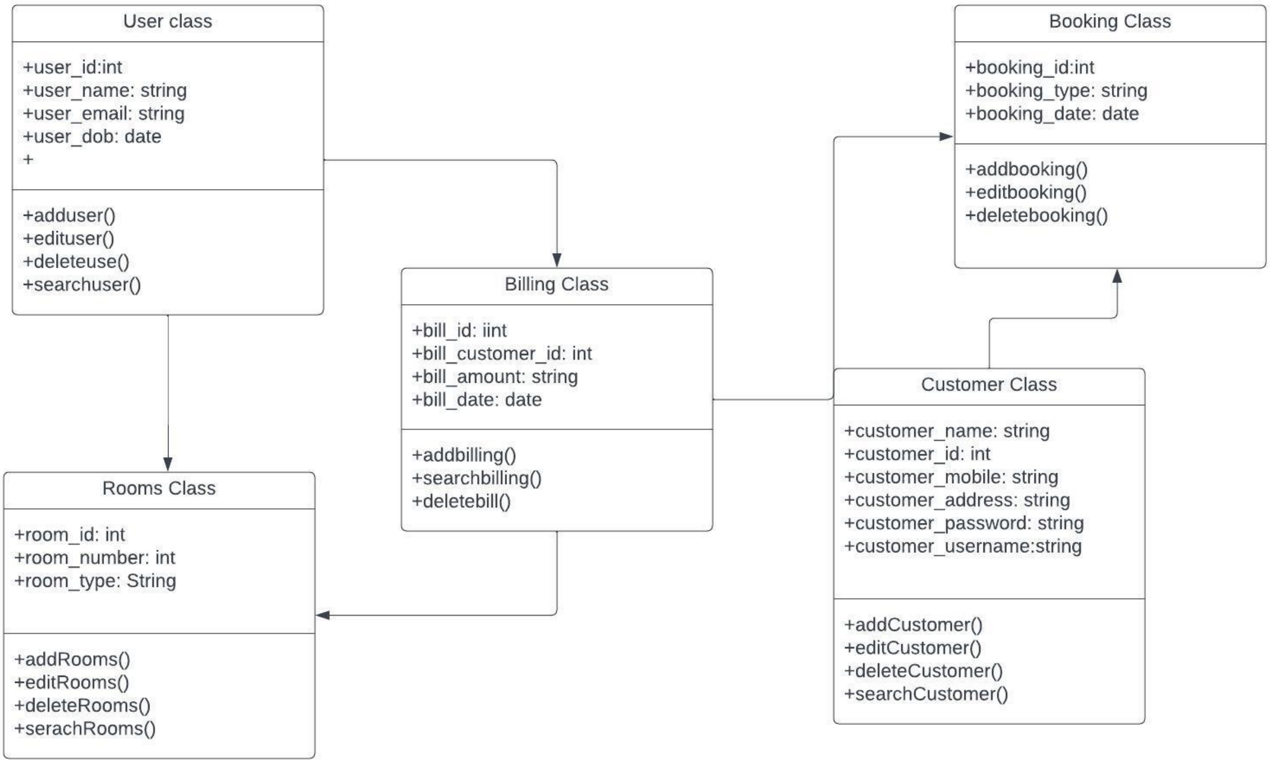
Interaction Overview Diagram

Communication Diagram

Timing Diagram

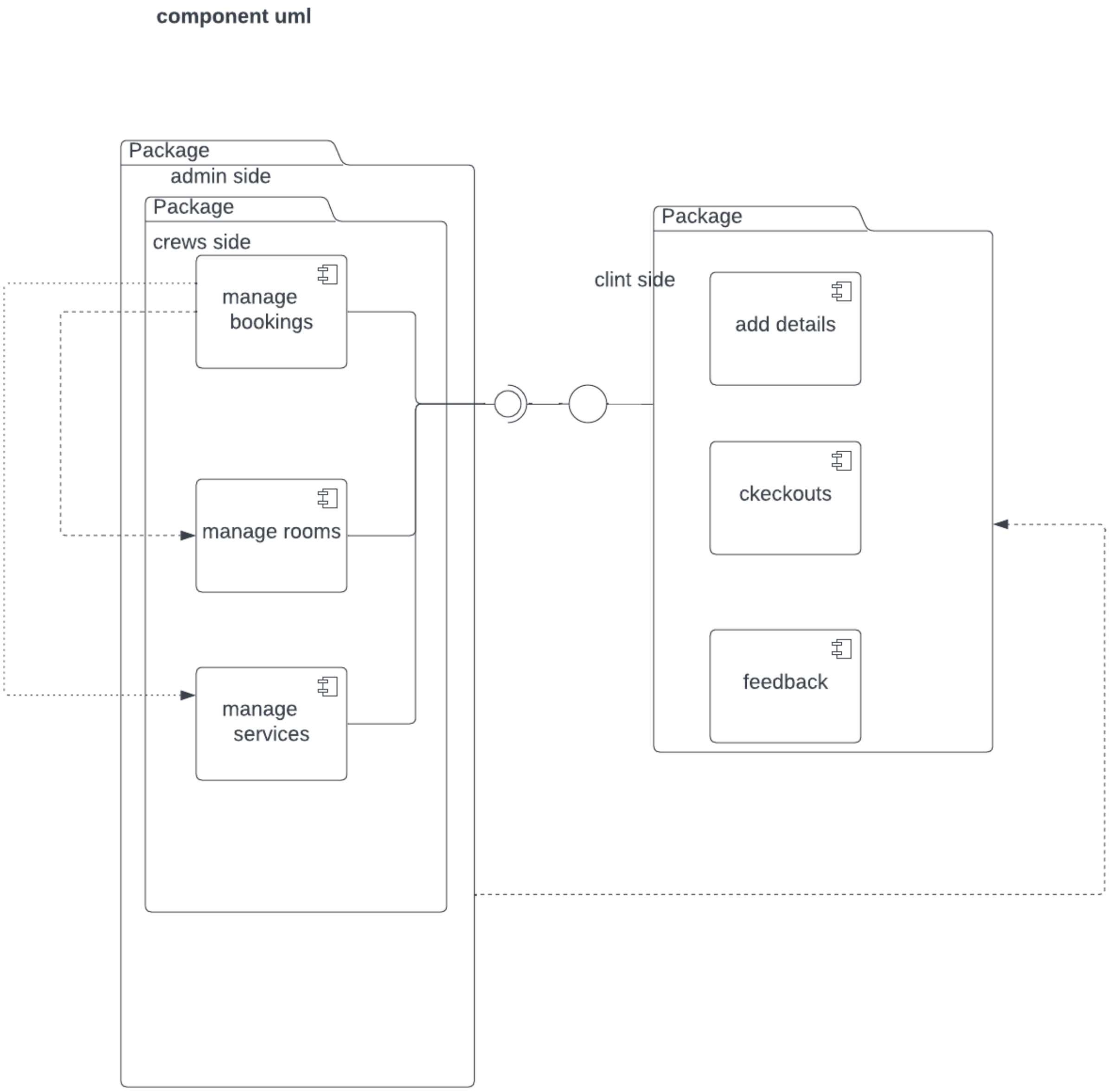
3.5.1 CLASS DIAGRAM

One of the most often used diagrams is the class diagram. All object-oriented software systems are built on it. It shows the system's static structure. It shows the class, properties, and operations of the system. It aids in understanding the relationships between various classes and objects.



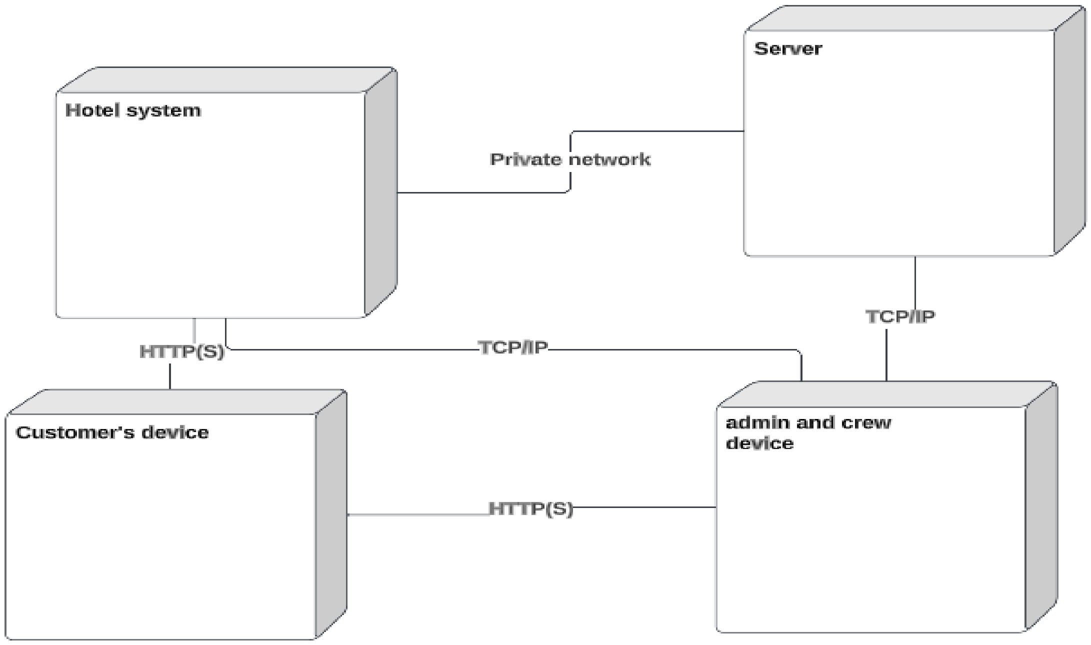
3.5.2 COMPONENT DIAGRAM

It illustrates how the system's physical components are arranged. It is employed for modeling specifics of execution. Because it contains structural linkages between the components of a software system, it can tell whether the planned development has taken into account the desired functional requirements or not.



3.5.3 DEPLOYMENT DIAGRAM

By describing the physical components that are now in place and the software components that are executing on them, it shows both the hardware and software of the system. It generates data on system software. Every time software is utilized, disseminated, or installed across a number of machines with various configurations, it is included.

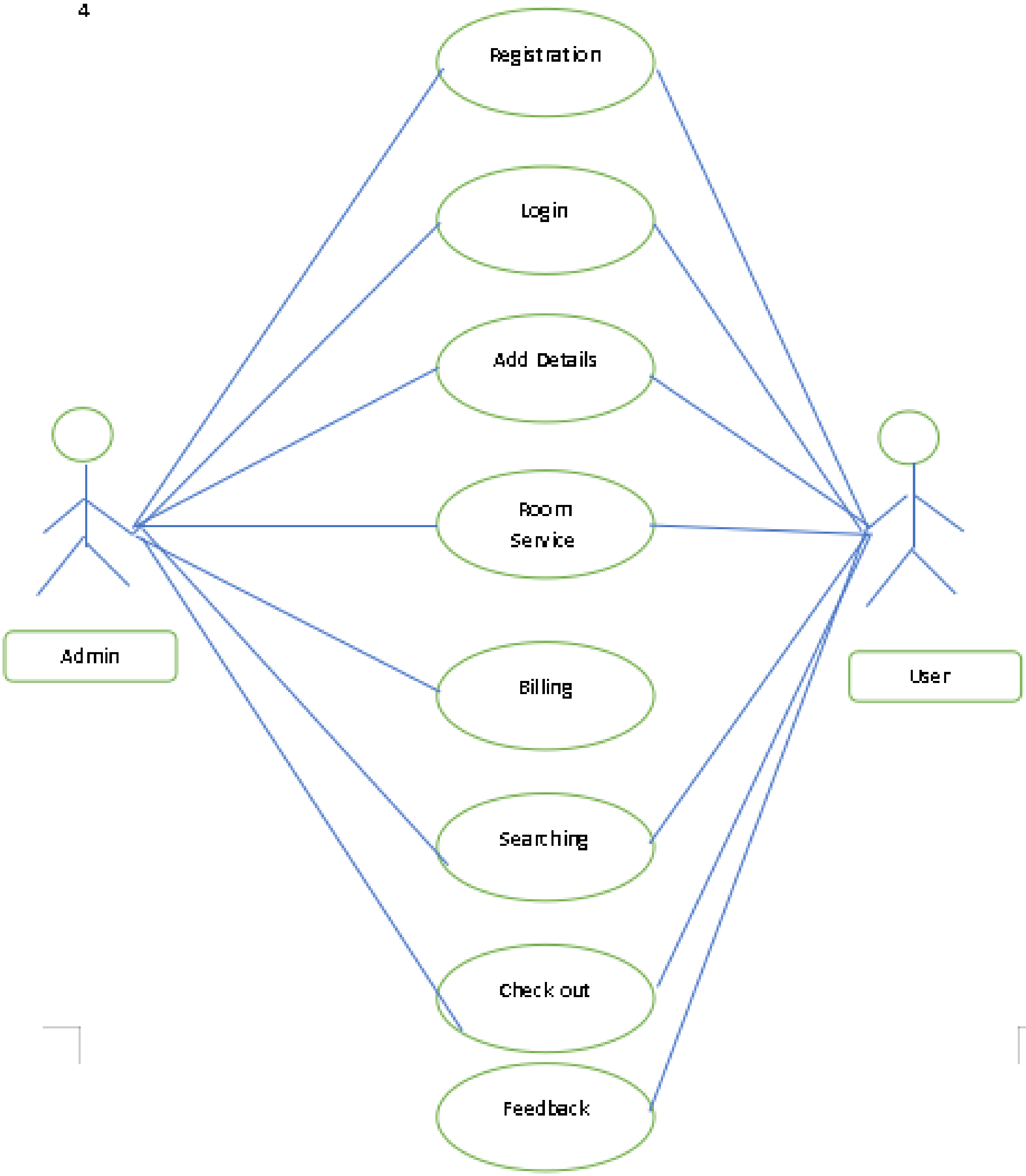


BEHAVIORAL DIAGRAMS

3.5.4 USECASE DIAGRAM

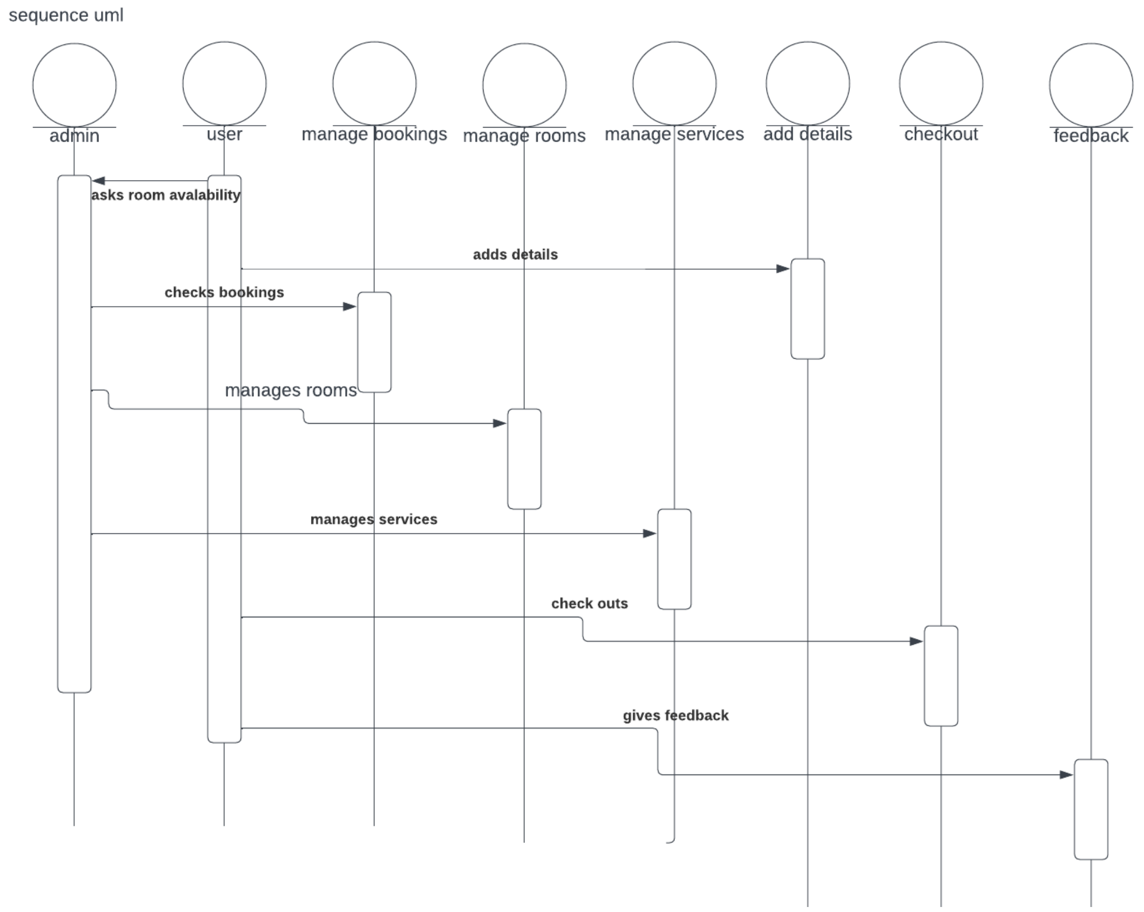
Use-case diagrams aid in capturing system requirements and depict a system's behavior in UML. The scope and high-level functions of a system are described in use-case diagrams. The interactions between the system and its actors are also depicted in these diagrams. utilize-case diagrams show what the system does and how the actors utilize it, but they do not show how the system works within.

4



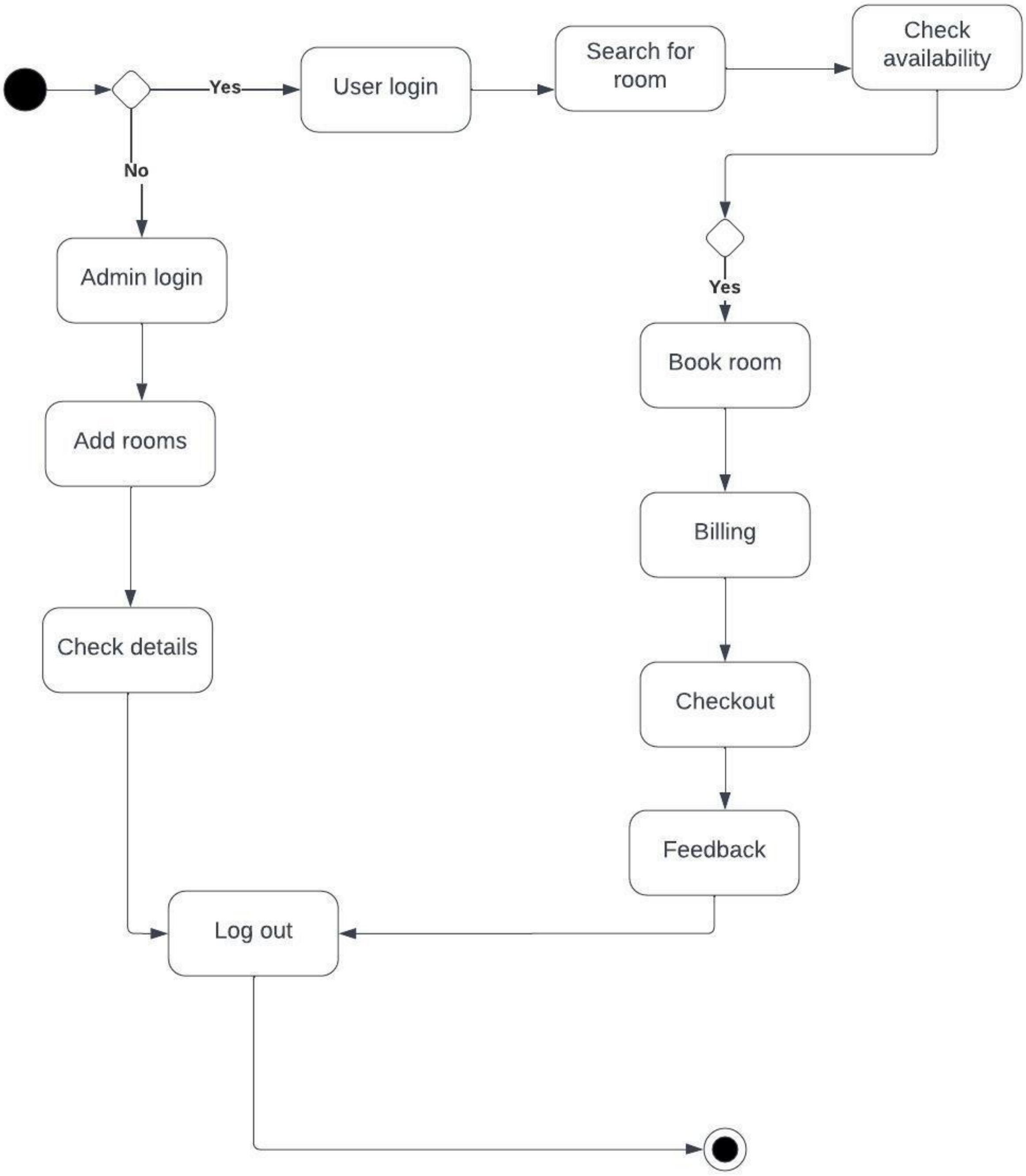
3.5.5 SEQUENCE DIAGRAM

The sequence diagram, which is also known as an event diagram, shows how messages move through the system. It aids in creating a variety of dynamic settings. It depicts communication between any two lifelines as a chronologically ordered series of activities, implying that these lifelines were active at the moment of communication. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.



3.5.6 ACTIVITY DIAGRAM

The activity diagram in UML is used to show the system's control flow rather than its implementation. Both concurrent and sequential activities are modeled. The workflow from one action to the next can be seen using the activity diagram. It placed emphasis on the existence of flow and the sequence in which it takes place.



CHAPTER-4

RESULTS & DISCUSSION

4.1 TEST CASES

Login:

Whenever admin wants to login into the system, with correct credentials whether one is able to login or not and when enters wrong username or password it is showing error message or not.

When a new user want to login into the system, initially as one has no account, whether it is showing “Account doesn’t exist” or not and enabling user to create new account in register interface.

After creating account successfully whether it is redirecting back to login page or not.

When user enters wrong password whether it is showing error message or not, and successfully login whenever user gives correct details or not.

Home Page:

All the elements present in navigation bar are successfully redirecting to their particular interface or not.

Services:

All the services Booking, Check-out, Feedback, Add rooms, Checking details buttons are correctly redirecting to their correct interface or not.

Booking:

Whenever booking interface is open, before goaing to booking room whether add customer details is clearly visible to the user and the form is perfectly working or not and the details added are visible on the screen.

After successfully adding details, book your room now button is redirecting to booking interface or not and room booking is perfectly working or not.

Check-out:

After a happy stay in the hotel, when the customer want to check-out by entering Reference No, Mobile No, Name and click on check-out button, all the details of user are deleted from database or not.

Feedback:

When the user wants to give feedback, there will be form to fill, the form submitted, the response is successfully stored into database or not.

Add Rooms:

Whenever admin wants to add new rooms to hotel, he can able to add rooms and whether they are visible in booking interface or not.

Checking Details:

Whenever admin wants to check details regarding booking, customer, room and feedback details all are visible correctly or not.

4.2 SCREENSHOTS

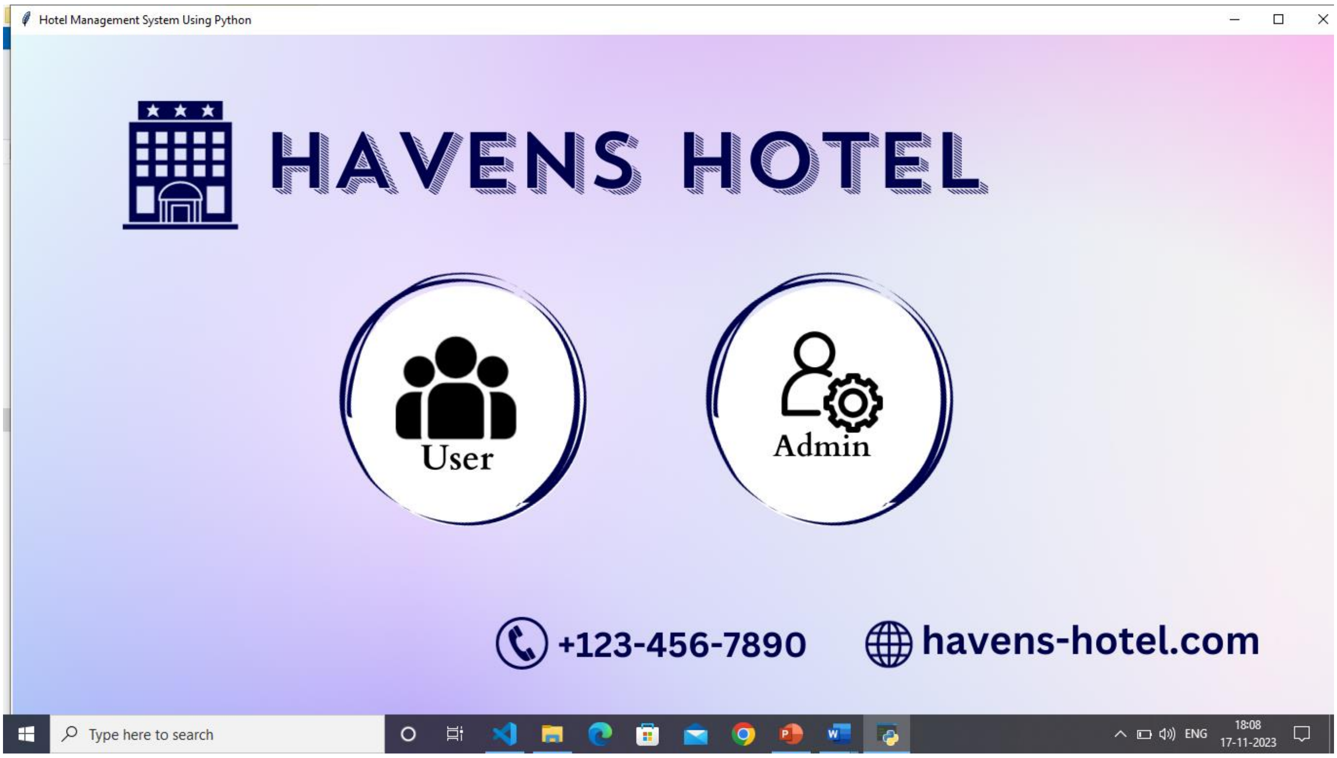
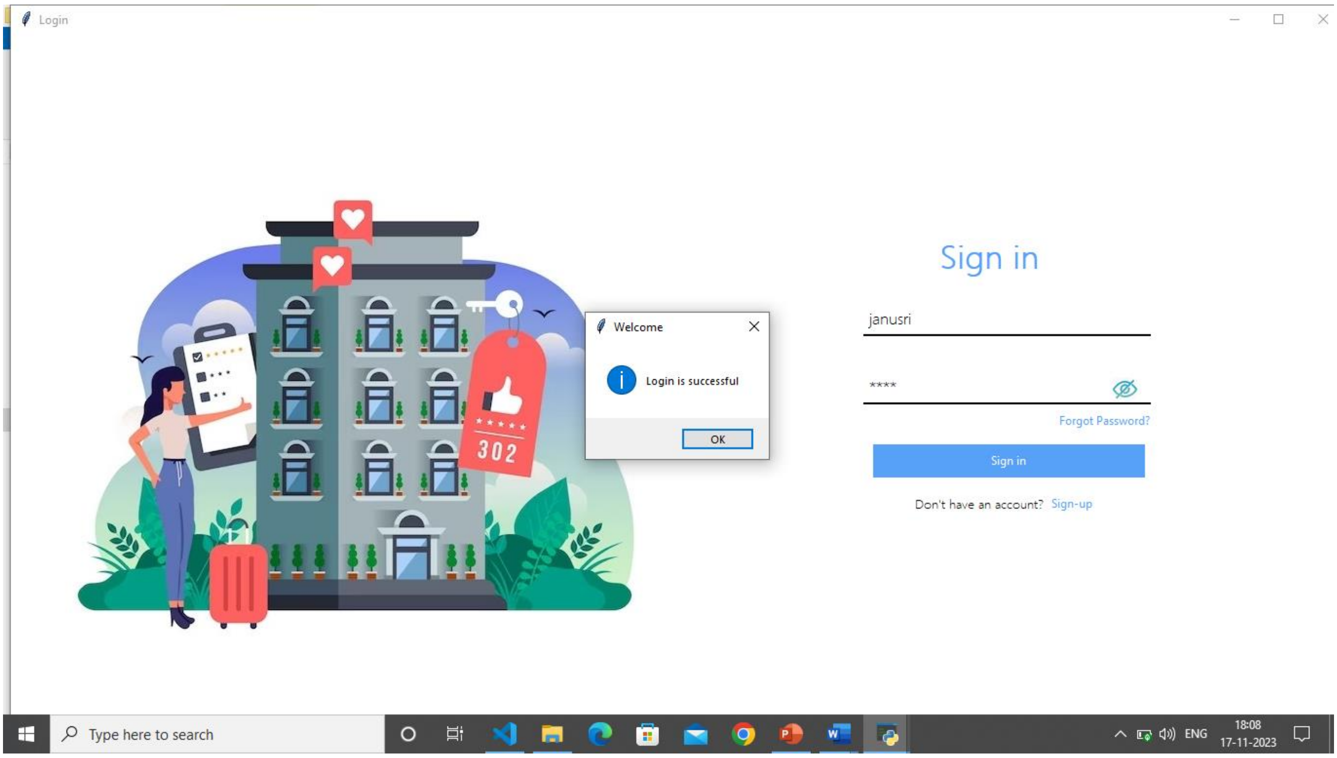


Fig: 4.2.1 Main page



Login is successful

Fig: 4.2.2 Sign in

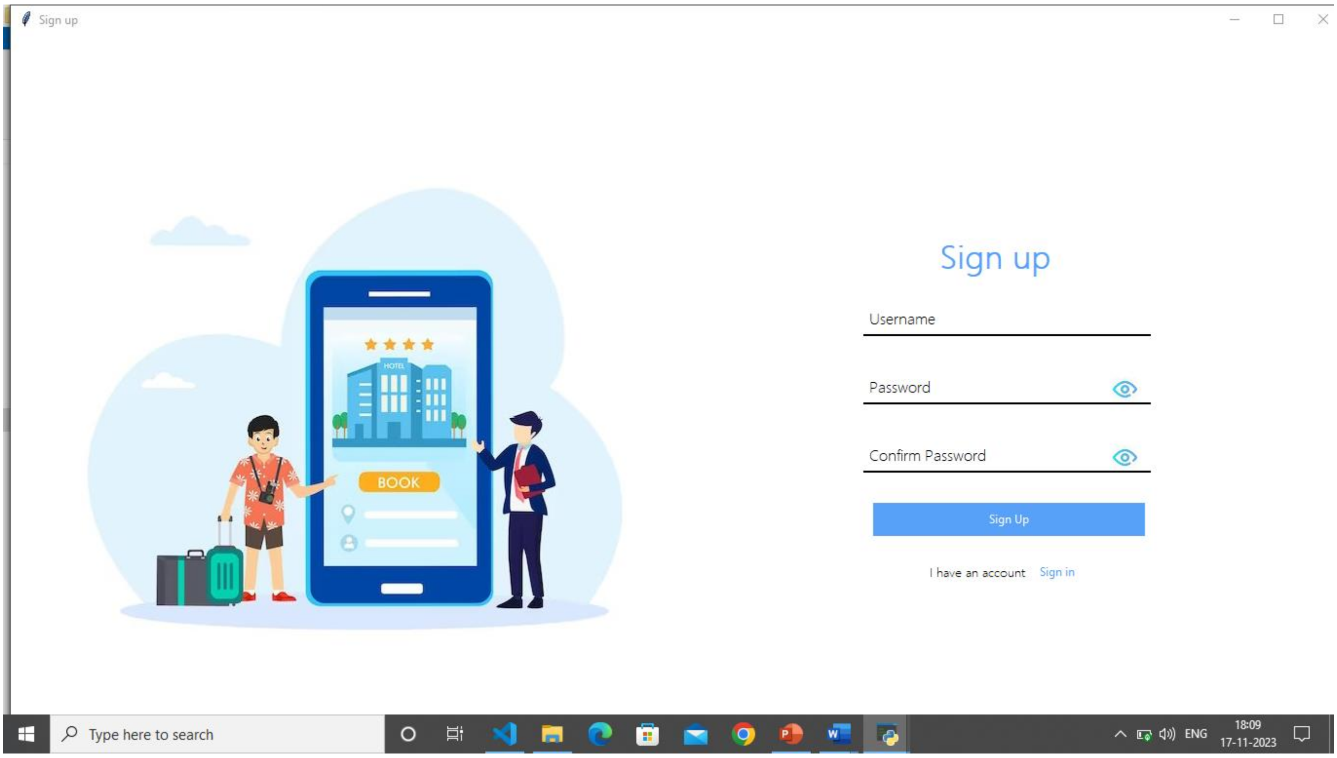


Fig: 4.2.3 Sign up

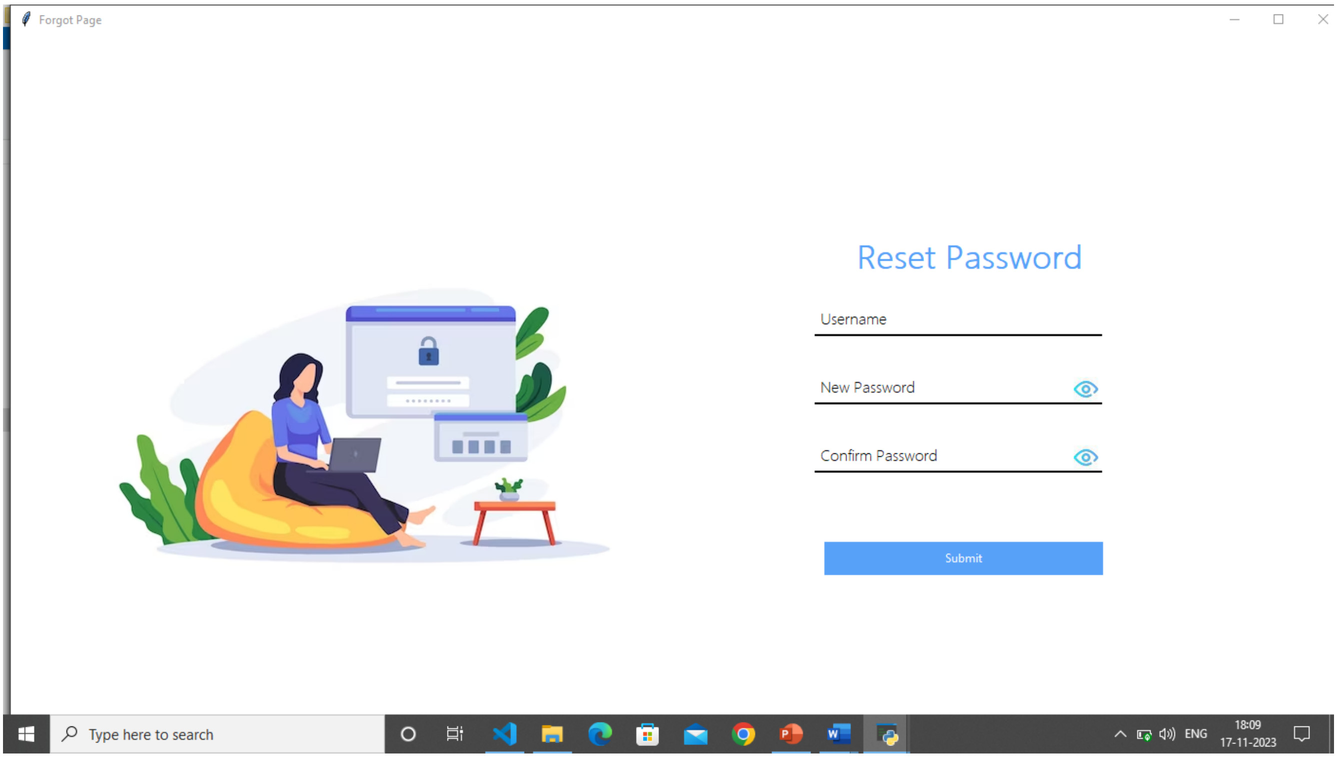
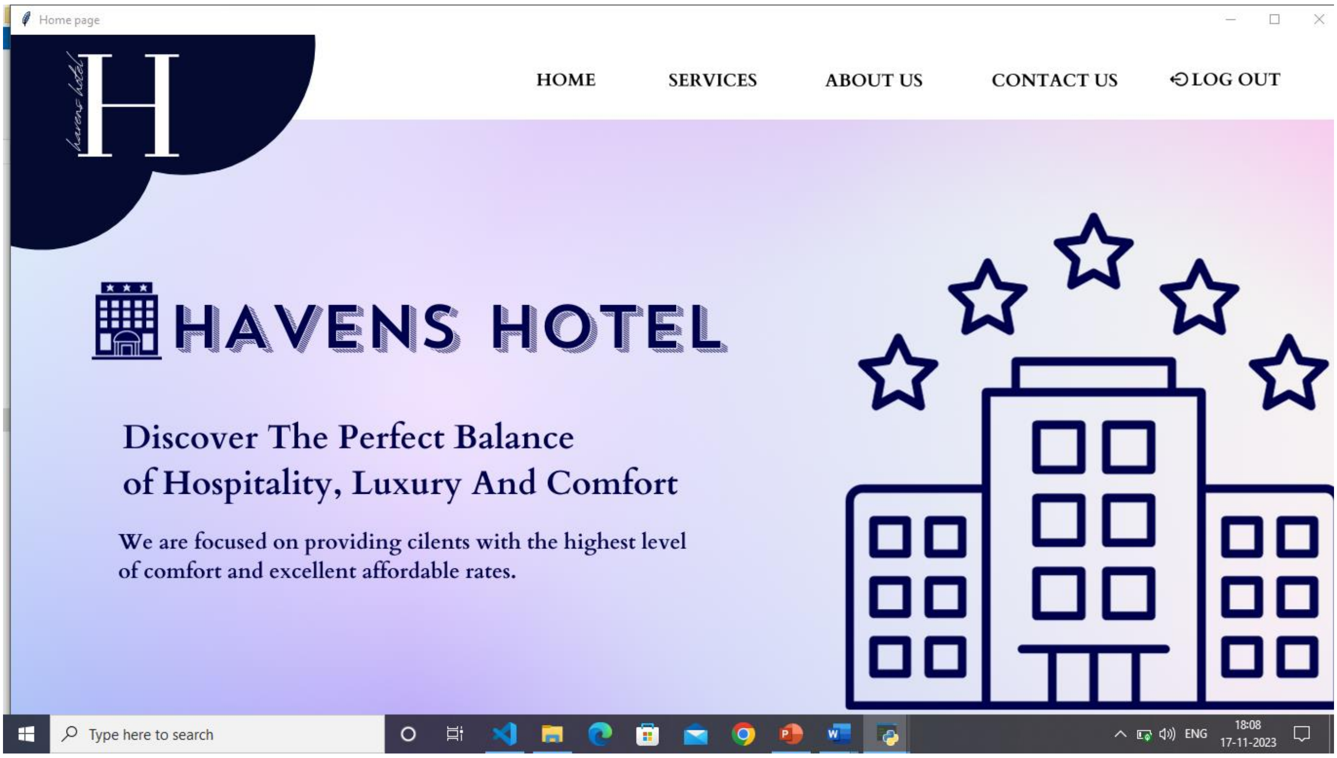


Fig: 4.2.4 Reset password



Discover The Perfect Balance of Hospitality, Luxury And Comfort

We are focused on providing cilents with the highest level of comfort and excellent affordable rates.

Fig: 4.2.5 Home page

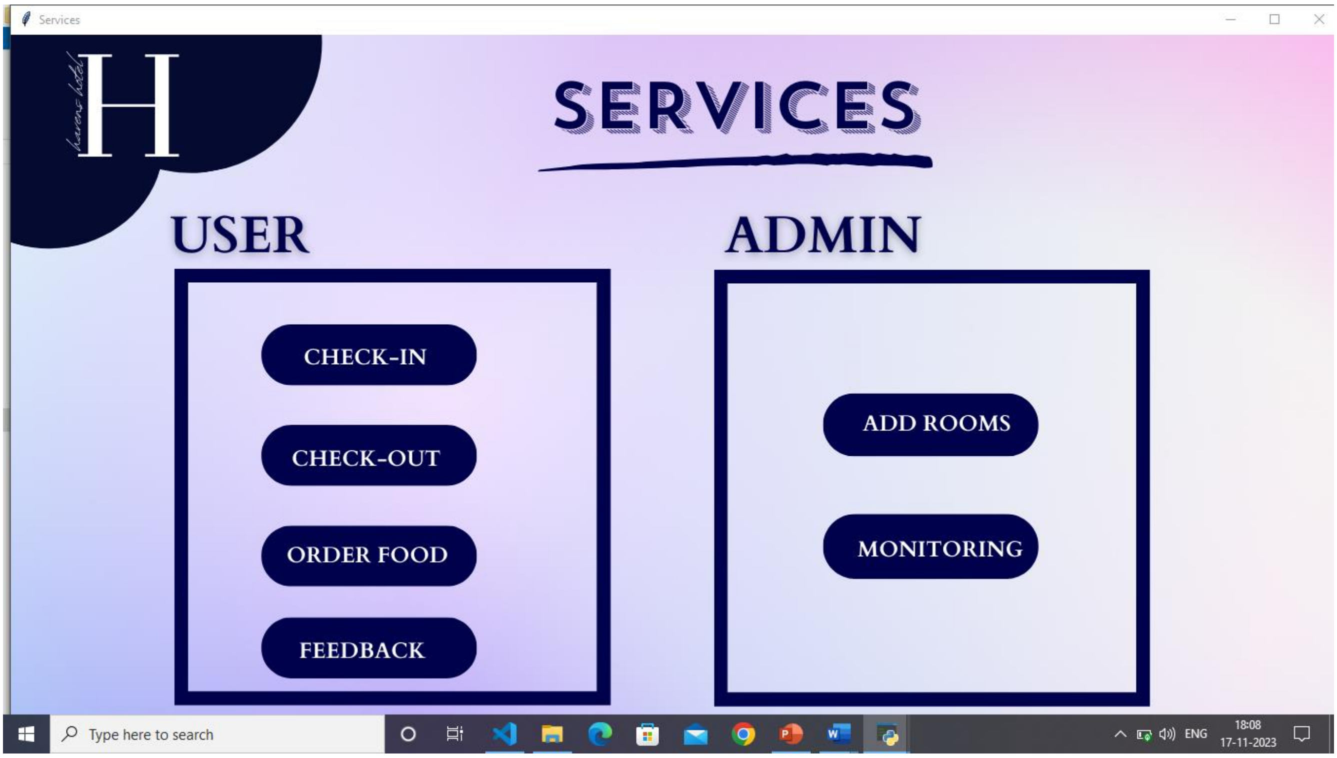
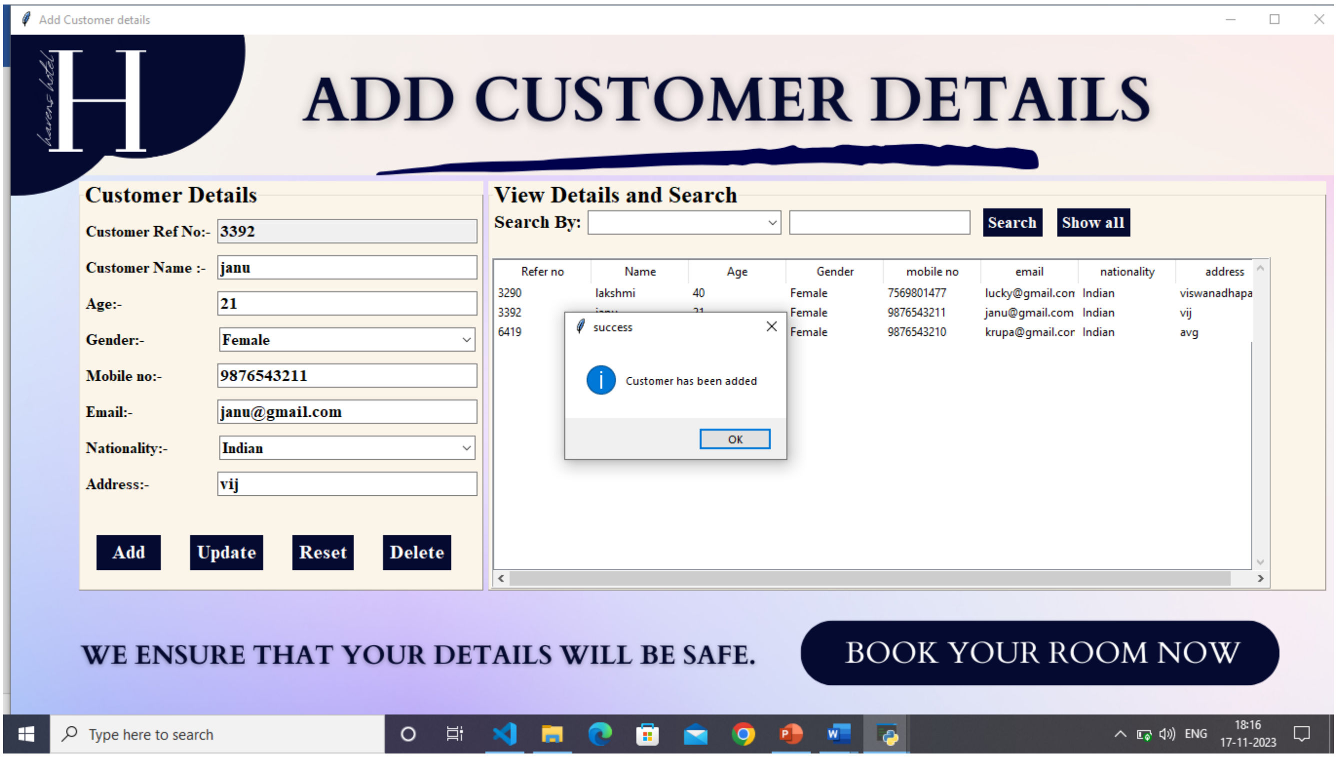


Fig 4.2.6 Services



ADD CUSTOMER DETAILS

Customer Details

Customer Ref No:-	3392
Customer Name :-	janu
Age:-	21
Gender:-	Female
Mobile no:-	9876543211
Email:-	janu@gmail.com
Nationality:-	Indian
Address:-	vij

View Details and Search

Search By:

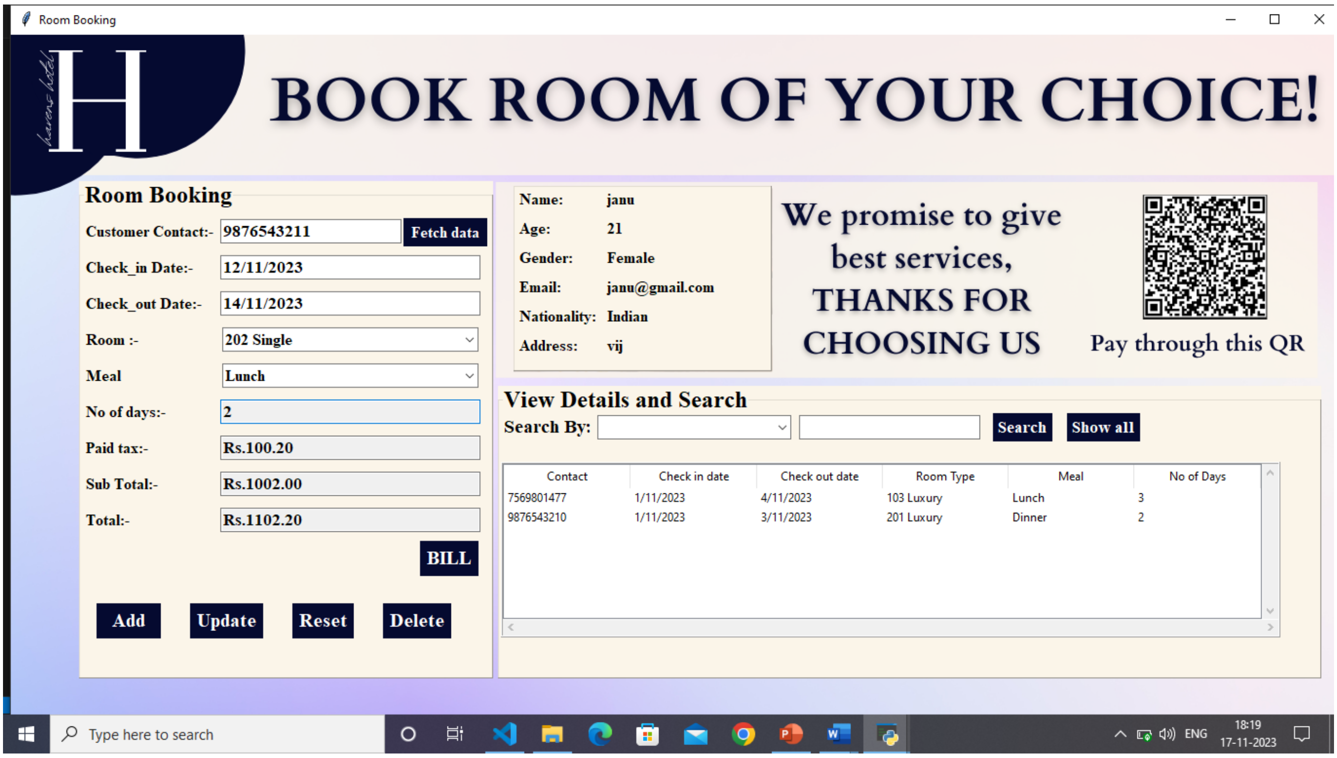
Refer no	Name	Age	Gender	mobile no	email	nationality	address
3290	lakshmi	40	Female	7569801477	lucky@gmail.con	Indian	viswanadhapa
3392	janu	21	Female	9876543211	janu@gmail.com	Indian	vij
6419			Female	9876543210	krupa@gmail.cor	Indian	avg

success

Customer has been added

WE ENSURE THAT YOUR DETAILS WILL BE SAFE.

Fig 4.2.7 Check in



BOOK ROOM OF YOUR CHOICE!

Room Booking

Customer Contact:-	9876543211
Check_in Date:-	12/11/2023
Check_out Date:-	14/11/2023
Room :-	202 Single
Meal	Lunch
No of days:-	2
Paid tax:-	Rs.100.20
Sub Total:-	Rs.1002.00
Total:-	Rs.1102.20

Name:	janu
Age:	21
Gender:	Female
Email:	janu@gmail.com
Nationality:	Indian
Address:	vij

We promise to give best services, THANKS FOR CHOOSING US

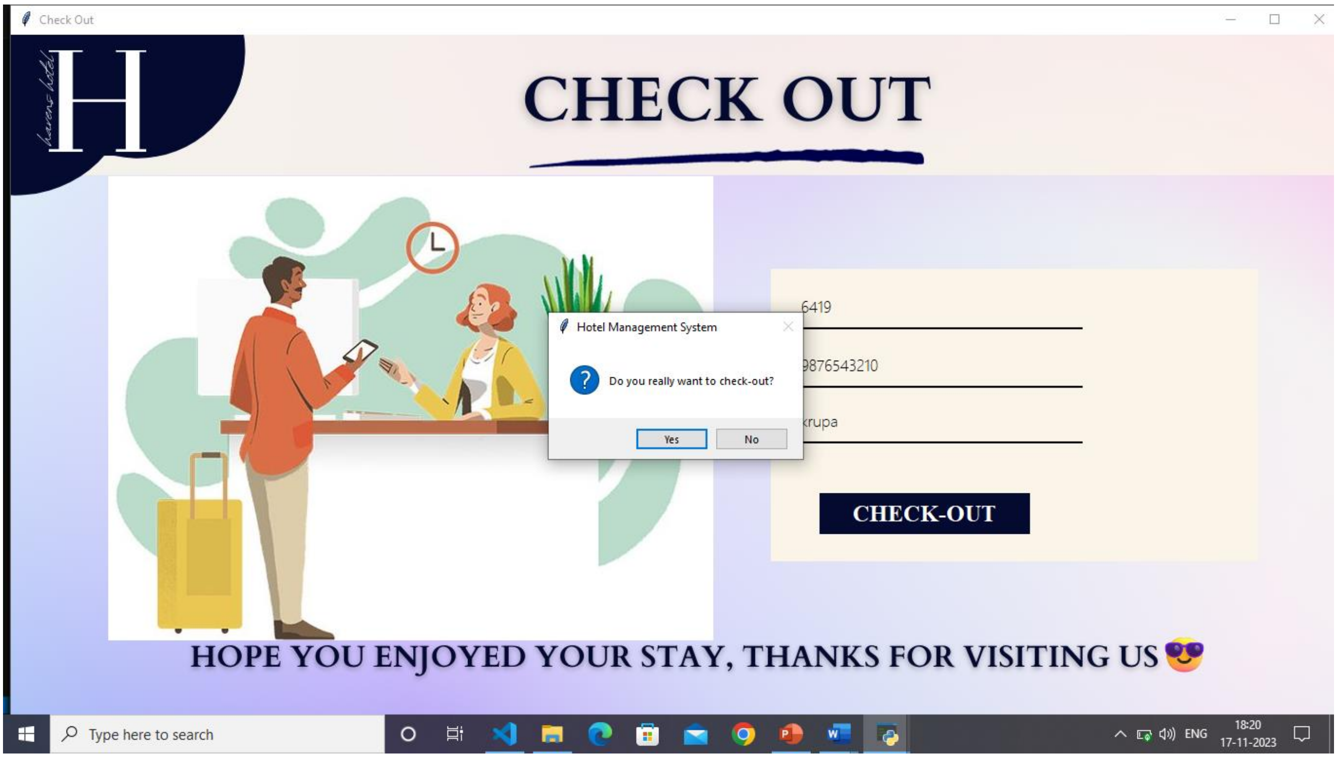
Pay through this QR

View Details and Search

Search By:

Contact	Check in date	Check out date	Room Type	Meal	No of Days
7569801477	1/11/2023	4/11/2023	103 Luxury	Lunch	3
9876543210	1/11/2023	3/11/2023	201 Luxury	Dinner	2

Fig 4.2.8 Booking



CHECK OUT

Hotel Management System

Do you really want to check-out?

Yes No

6419

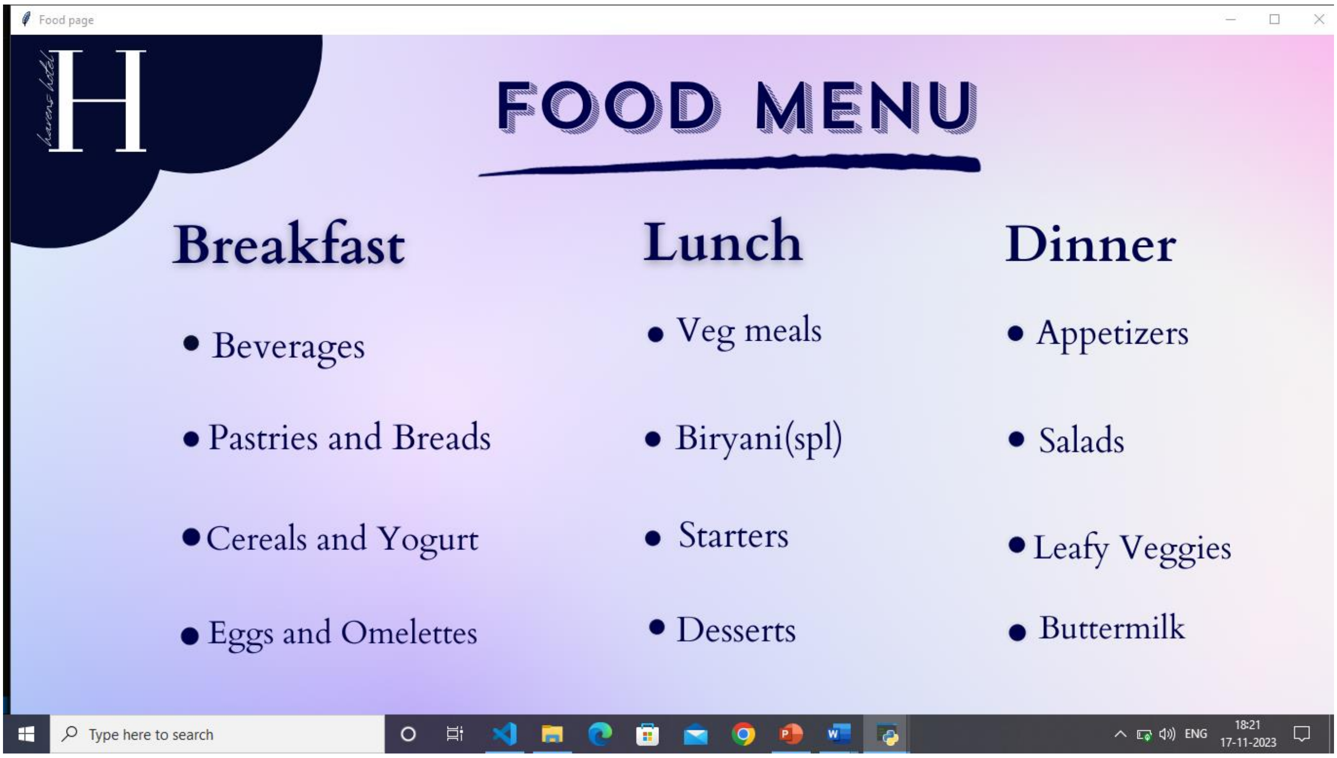
9876543210

krupa

CHECK-OUT

HOPE YOU ENJOYED YOUR STAY, THANKS FOR VISITING US 😎

Fig 4.2.9.check out



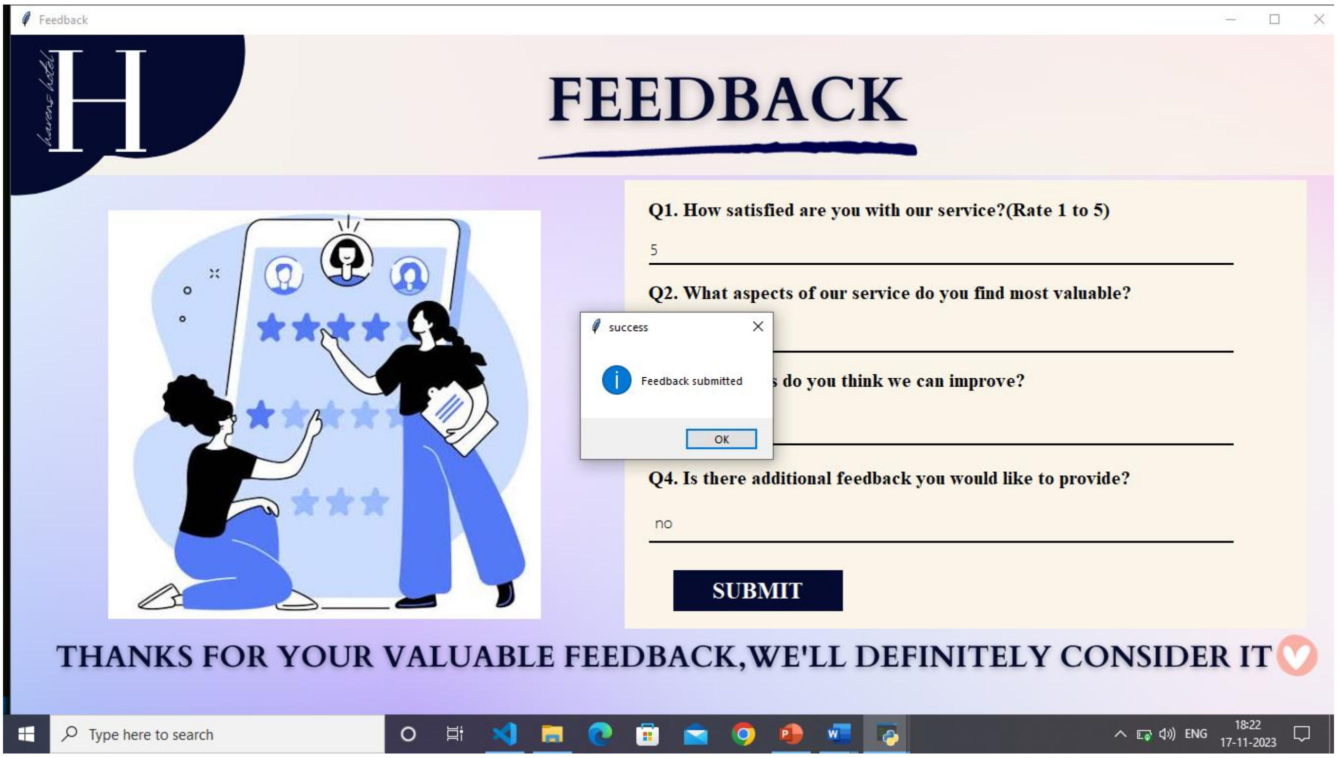
FOOD MENU

Breakfast
- Beverages
- Pastries and Breads
- Cereals and Yogurt
- Eggs and Omelettes

Lunch
- Veg meals
- Biryani(spl)
- Starters
- Desserts

Dinner
- Appetizers
- Salads
- Leafy Veggies
- Buttermilk

Fig 4.2.10.Ordering food



Feedback

FEEDBACK

Q1. How satisfied are you with our service?(Rate 1 to 5)

5

Q2. What aspects of our service do you find most valuable?

success

Feedback submitted

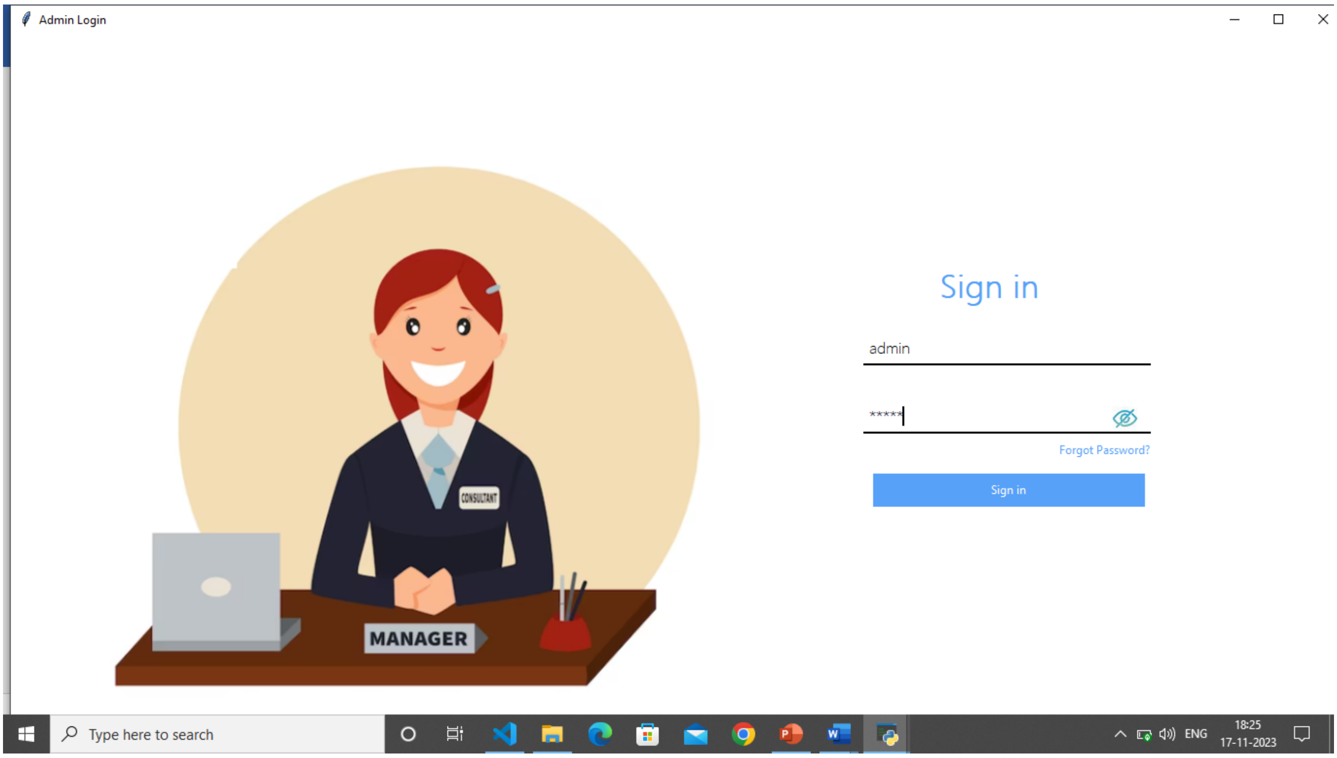
s do you think we can improve?

Q4. Is there additional feedback you would like to provide?

no

THANKS FOR YOUR VALUABLE FEEDBACK,WE'LL DEFINITELY CONSIDER IT

Fig 4.2.11.Feedback



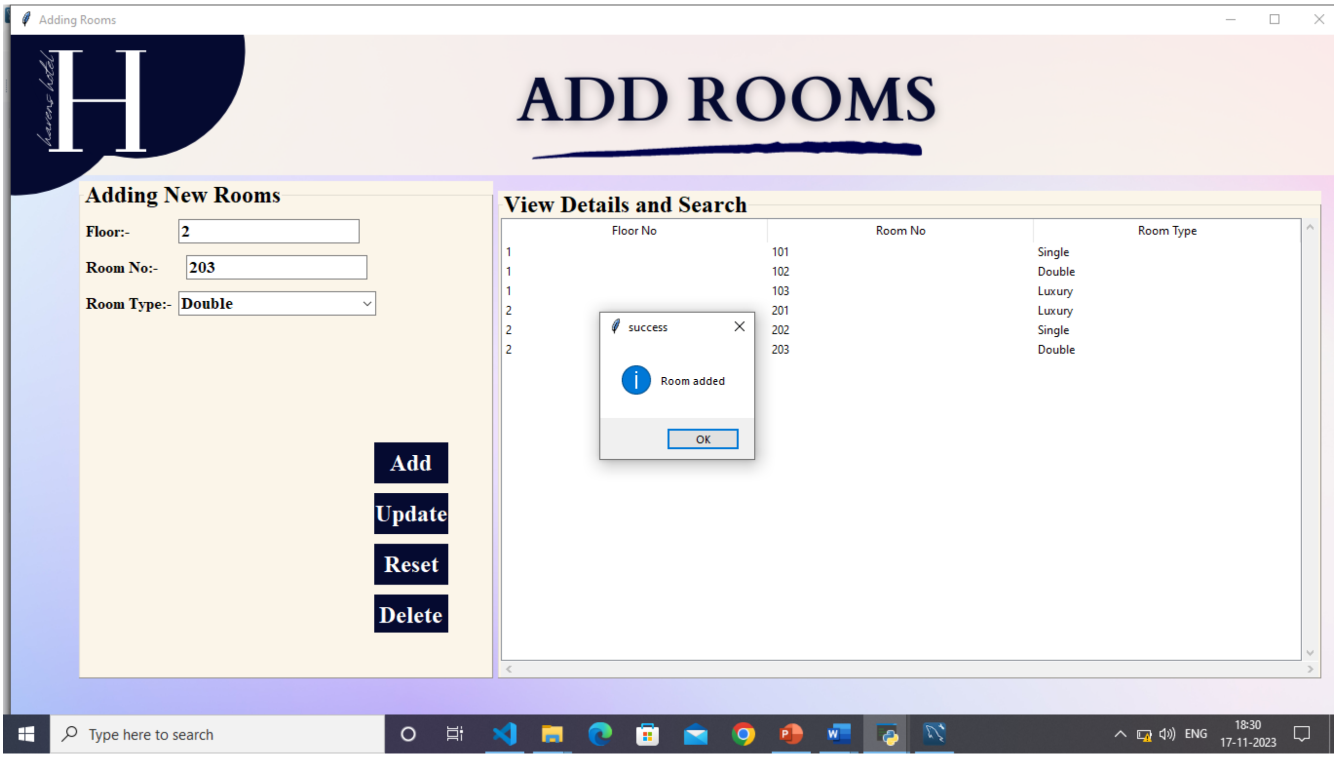
Admin Login

Sign in

admin

*****

Fig 4.2.12 Admin sign in



Adding New Rooms

Floor:- 2

Room No:- 203

Room Type:- Double

View Details and Search

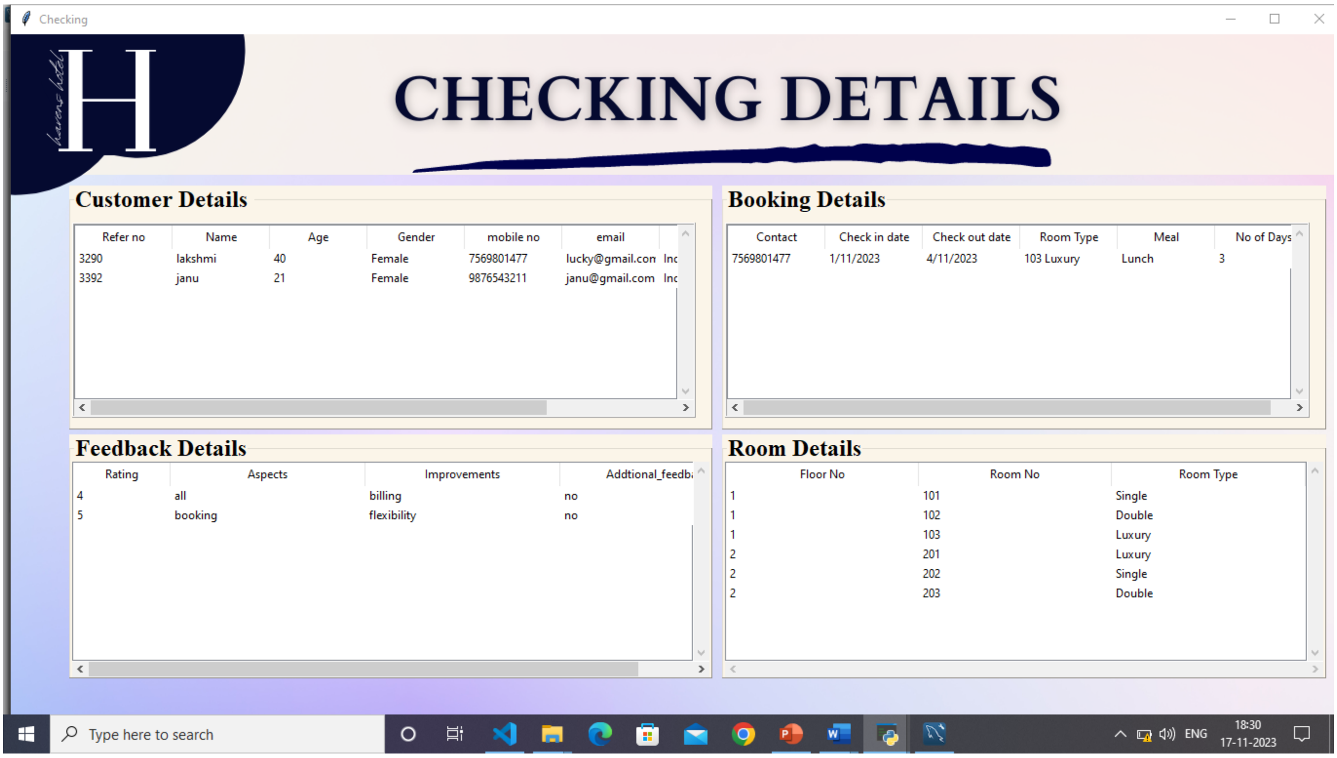
Floor No	Room No	Room Type
1	101	Single
1	102	Double
1	103	Luxury
2	201	Luxury
2	202	Single
2	203	Double

success

Room added

OK

Fig 4.2.13 Adding rooms



Customer Details

Refer no	Name	Age	Gender	mobile no	email	
3290	lakshmi	40	Female	7569801477	lucky@gmail.con	Inc
3392	janu	21	Female	9876543211	janu@gmail.com	Inc

Booking Details

Contact	Check in date	Check out date	Room Type	Meal	No of Days
7569801477	1/11/2023	4/11/2023	103 Luxury	Lunch	3

Feedback Details

Rating	Aspects	Improvements	Addtional_feedb
4	all	billing	no
5	booking	flexibility	no

Room Details

Floor No	Room No	Room Type
1	101	Single
1	102	Double
1	103	Luxury
2	201	Luxury
2	202	Single
2	203	Double

Fig 4.2.14 Checking details

CHAPTER-5
SUMMARY & CONCLUSION

In conclusion, a hotel management system using Python is a powerful tool that automates various hotel operations and administrative tasks. It streamlines processes, enhances efficiency, and improves guest experiences. The system enables centralized management of reservations, guest information, billing, reporting, and housekeeping tasks. Additionally, it can be extended in the future to incorporate additional features to meet the evolving needs of the hospitality industry. Overall, a hotel management system using Python is a valuable investment for hotels of all sizes looking to improve operational efficiency and increase guest satisfaction.

5.1 FUTURE SCOPE

Future enhancements for a hotel management system using Python may include online reputation management, CRM integration for personalized guest communication, revenue management optimization, loyalty program management, mobile app development, advanced analytics for data-driven decision-making, integration with IoT devices for enhanced guest experiences, and multi-property management capabilities for centralized control and coordination across multiple hotels. These enhancements aim to improve guest satisfaction, streamline operations, maximize revenue, and adapt to emerging industry trends and technologies.

Some additional features we might add are as follows:

Integrating ordering food from popular websites like Zomato and Swiggy,

Introducing careers or job roles present in the hotel.

Mobile app development for the project.

5.2. BIBLIOGRAPHY

•In 2014, Room Allocation for Hotel Revenue Management Considering consumer choice behavior by Chaoyong Qin

•In 2016, Hotel's online booking segmentation for heterogenous customers by Z.W. Miao, T.Wei, Y.Q.Lan

•In 2018, Design and Implementation of Hotel Room Information Management System Based on Kendo UI Front-End Framework by Mei Shan-Shan, Shan Chun, Xue Jing-Feng

•In 2018, Analysis and Design og Five-Star Hotel Management Information System Based on UML by Guo Chen

•In 2019, Design and Implementation of Hotel Room Management System by Wei, Zhengwei Lou

•In 2020, Enhanced Hotel Booking Application using PEGA by R Dhanagopa;,N Archana, R Menaka

•In 2021, Design and Implementation of Hotel Reservation Management Platform based on SOA Framework by Ximei Lv

•In 2021, Hotel Management Platform Design Based on Virtual Reality by Ding Xueyan

•In 2021, Review on Hotel Management System by Avneesh Pathak, Arun Snghal, Bibhas Kumar Rana

•In 2022, CRM Hotel Management System and Intelligent Information Push Based on BP Neural Network by Li Tang

•In 2022, Implementation of Hotel Intelligent Management System based on Big Data, by Zhichen Zhong

References

o https://www.odoo.com/

o https://www.inngrid.net/pricing

o https://www.freshworks.com/crm/

o https://www.hotelogix.com/

o https://www.techavidus.com/case-studies/hotel-management-system

o https://www.hotelopro.com/