HOTEL MANAGEMENT SYSTEM

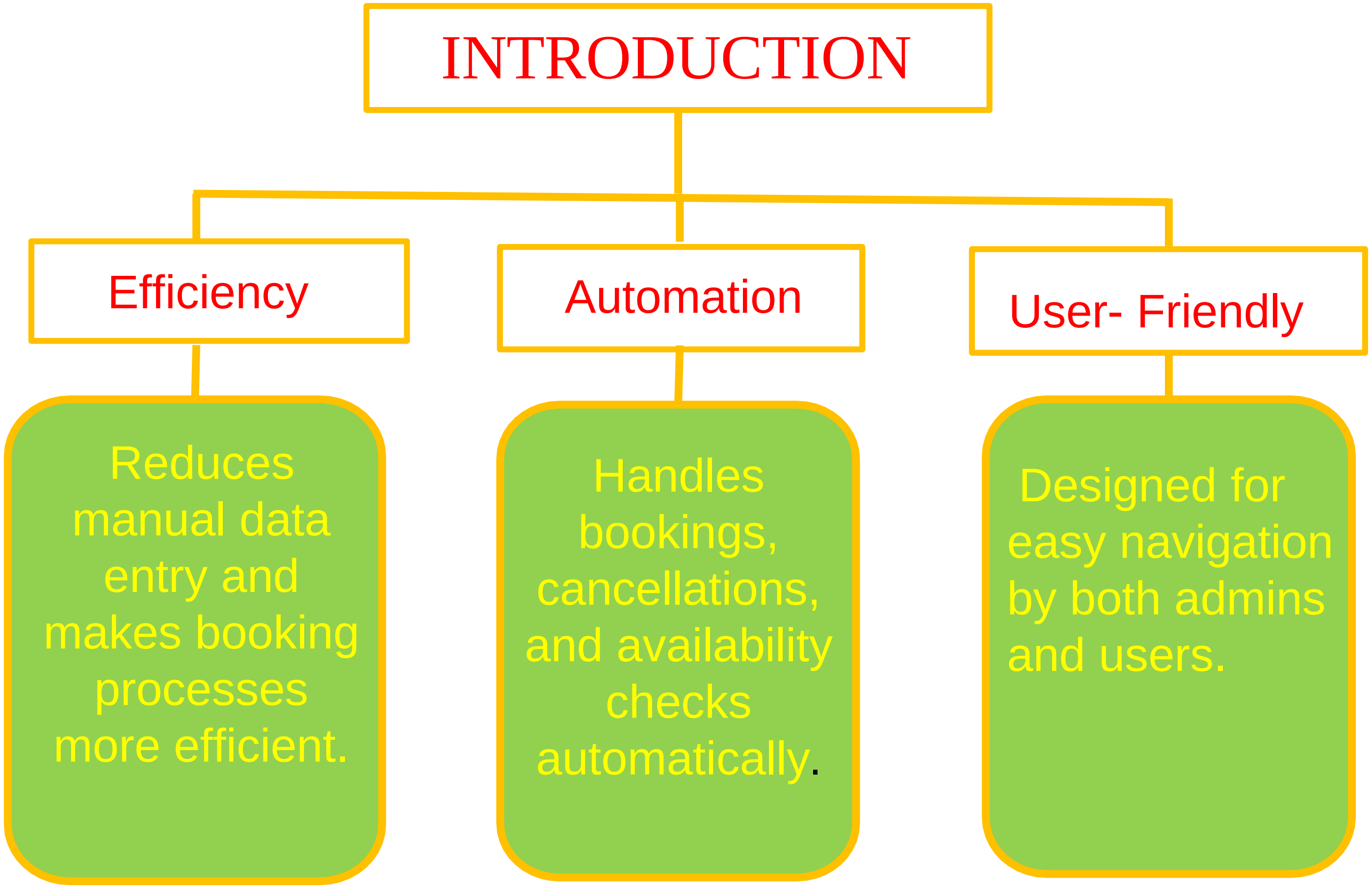
By

DHURGASHREE I	-	23ITR038
DIVYA R	-	23ITR042
GUHAN P	-	23ITR050
ARAVINDHAN T	-	23ITL183

ABSTRACT

Overview: The Hotel Management System is developed in Python to handle hotel operations efficiently. It includes an intuitive interface for managing rooms, bookings, and users (admins and guests).

Purpose: Simplify hotel management with automated room booking, room status updates, and secure user logins.



INTRODUCTION

Efficiency

Reduces manual data entry and makes booking processes more efficient.

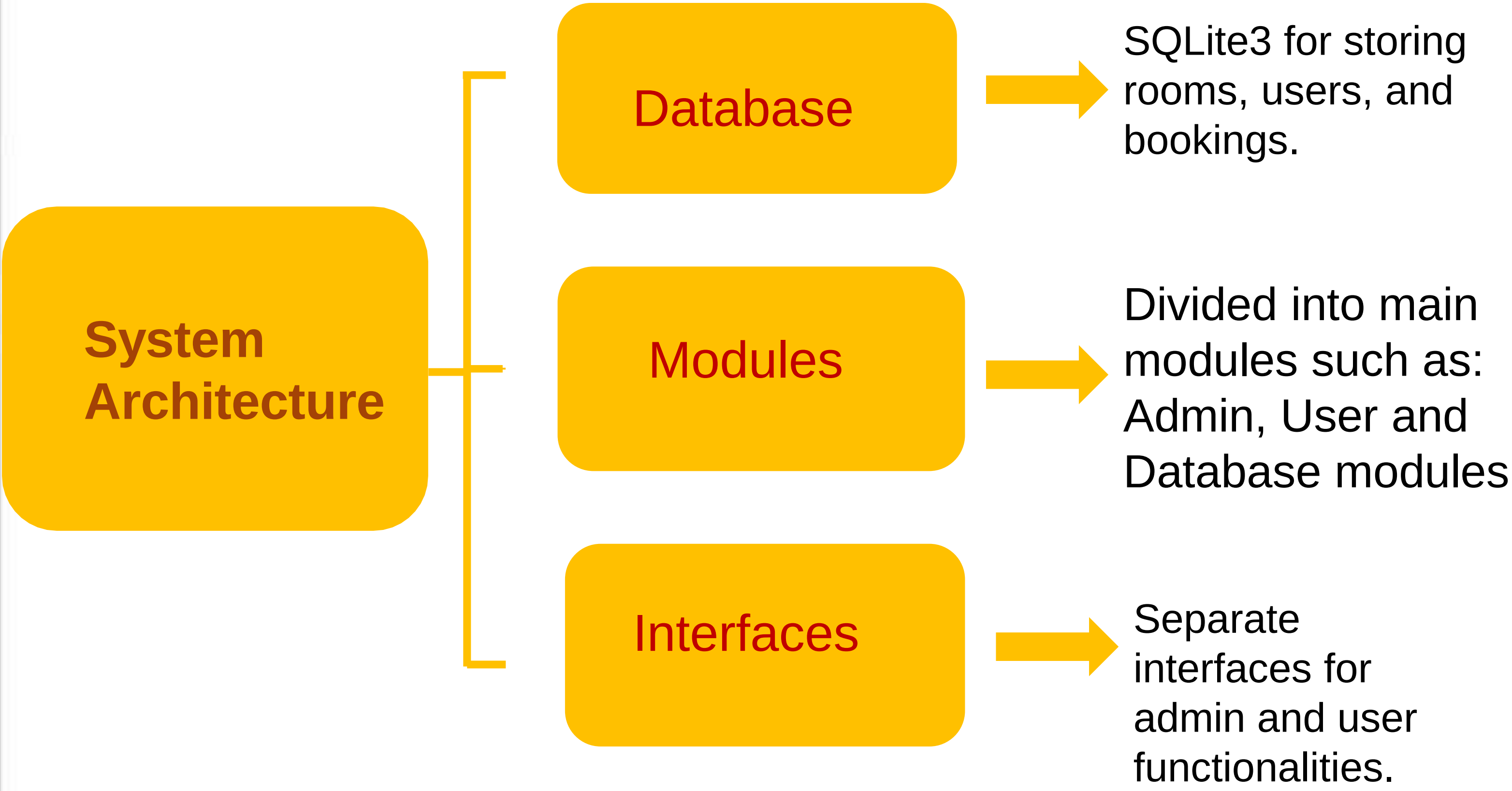
Automation

Handles bookings, cancellations, and availability checks automatically.

User- Friendly

Designed for easy navigation by both admins and users.

Hotel Management System



System Architecture

Database

SQLite3 for storing rooms, users, and bookings.

Modules

Divided into main modules such as: Admin, User and Database modules

Interfaces

Separate interfaces for admin and user functionalities.

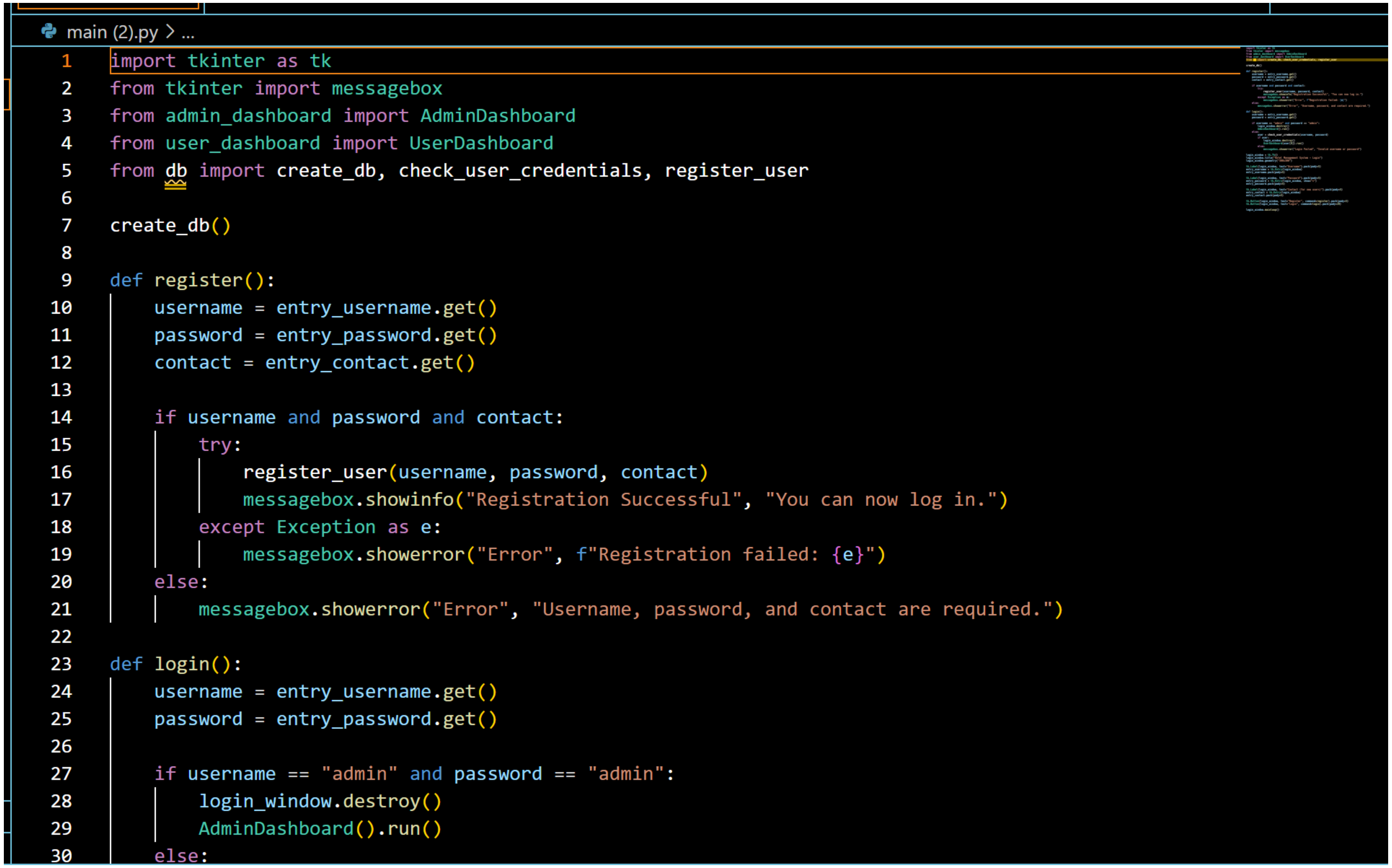
Features and Functionalities

Admin Dashboard ⟹ Add and update room details (price,status).
View available rooms and booking details.

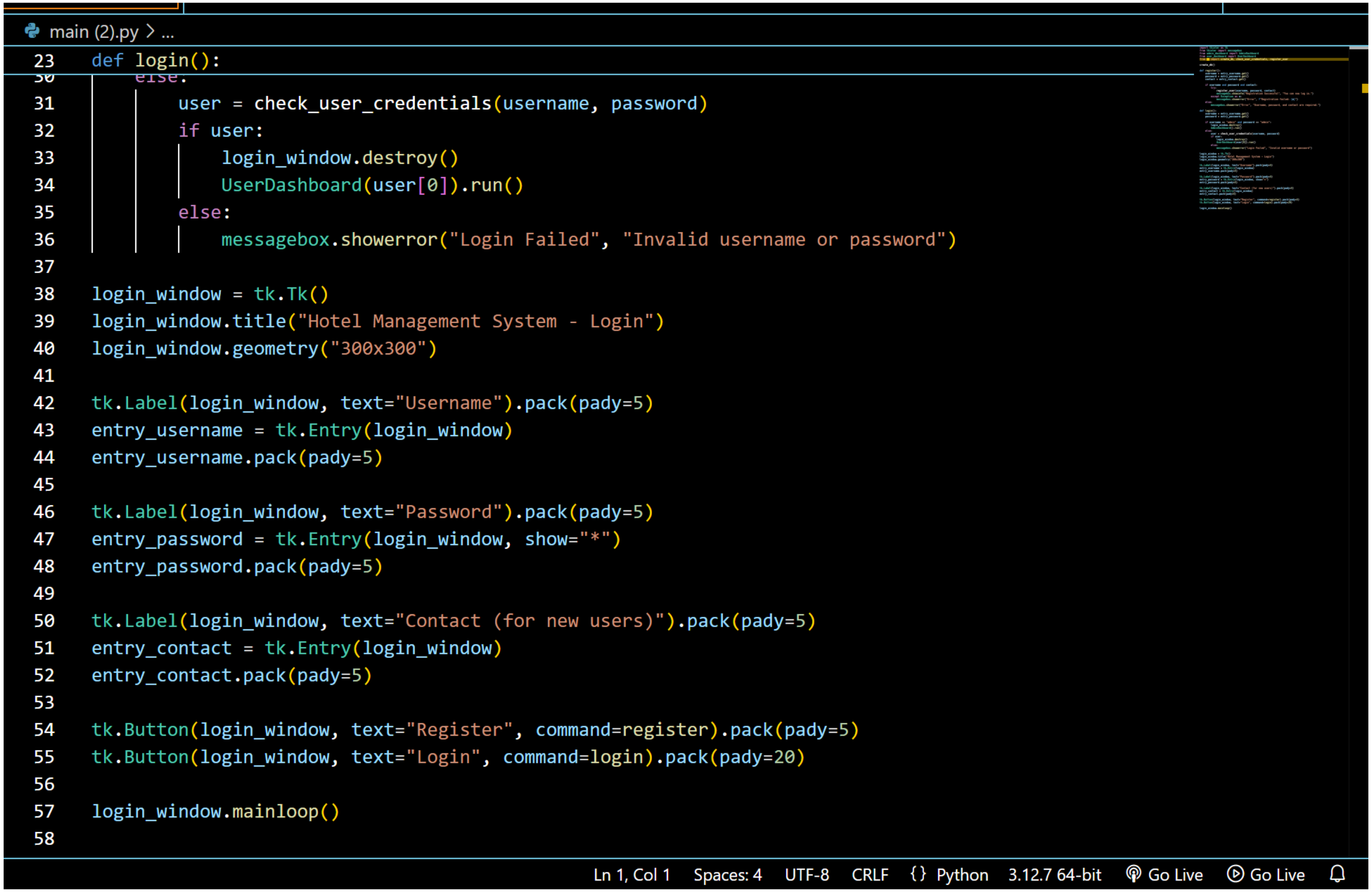
User Dashboard ⟹ View available rooms, book or cancel bookings, and view personal booking history.

Secure Login System ⟹ Admin and user logins with password validation.

main.py:

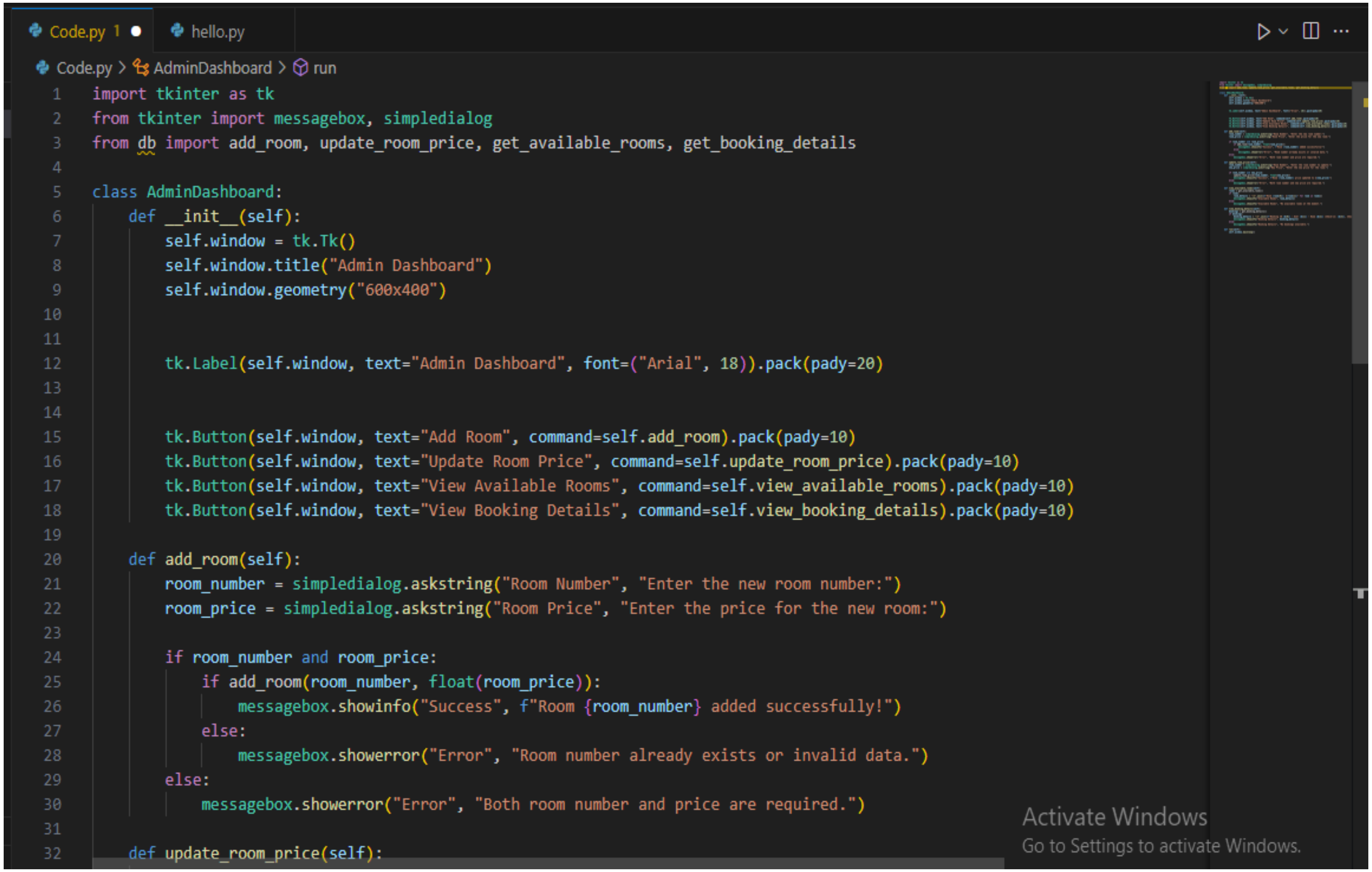


```python
import tkinter as tk
from tkinter import messagebox
from admin_dashboard import AdminDashboard
from user_dashboard import UserDashboard
from db import create_db, check_user_credentials, register_user

create_db()

def register():
    username = entry_username.get()
    password = entry_password.get()
    contact = entry_contact.get()

    if username and password and contact:
        try:
            register_user(username, password, contact)
            messagebox.showinfo("Registration Successful", "You can now log in.")
        except Exception as e:
            messagebox.showerror("Error", f"Registration failed: {e}")
    else:
        messagebox.showerror("Error", "Username, password, and contact are required.")

def login():
    username = entry_username.get()
    password = entry_password.get()

    if username == "admin" and password == "admin":
        login_window.destroy()
        AdminDashboard().run()
    else:
```

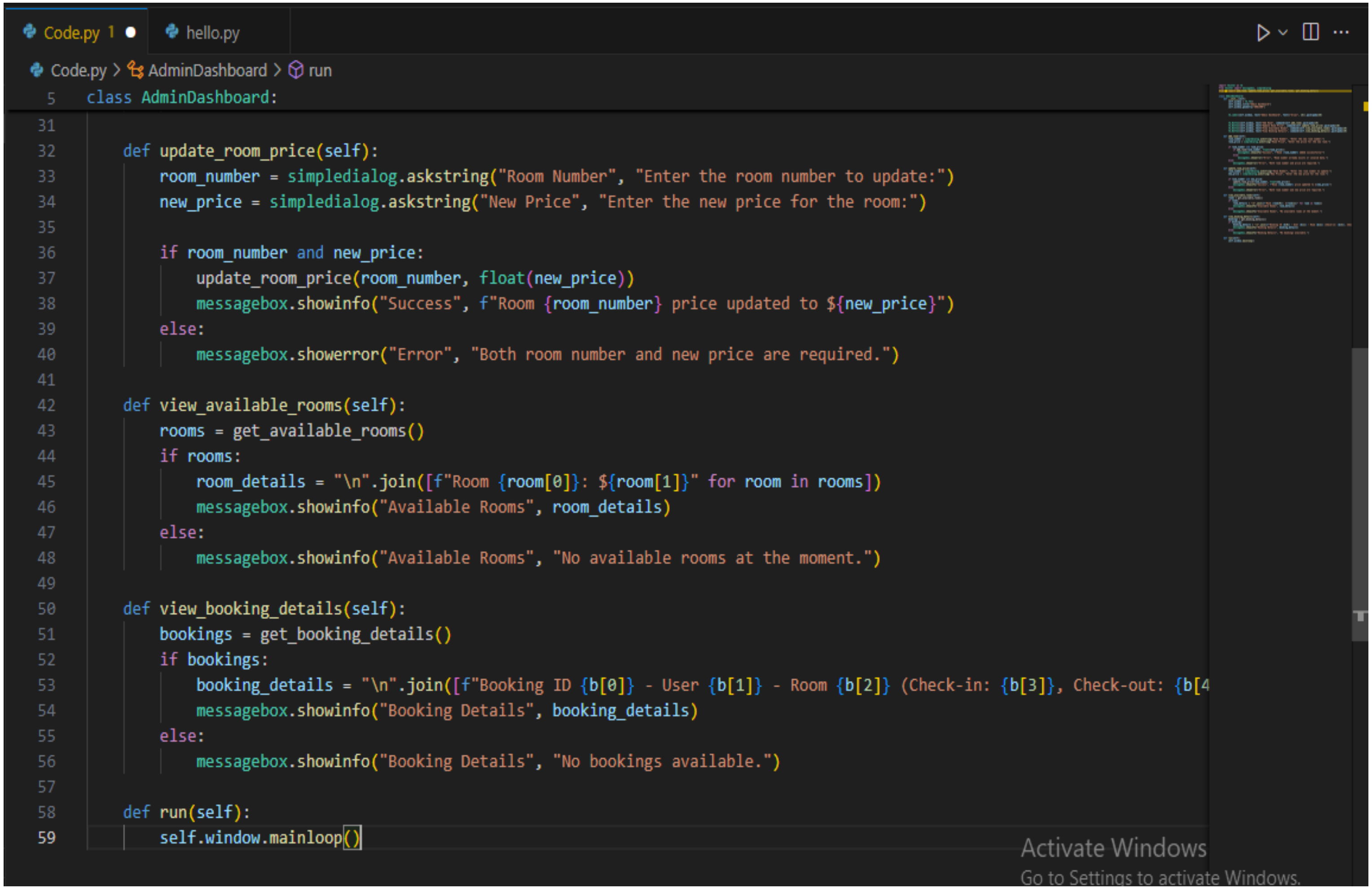


```python
def login():
    else:
        user = check_user_credentials(username, password)
        if user:
            login_window.destroy()
            UserDashboard(user[0]).run()
        else:
            messagebox.showerror("Login Failed", "Invalid username or password")

login_window = tk.Tk()
login_window.title("Hotel Management System - Login")
login_window.geometry("300x300")

tk.Label(login_window, text="Username").pack(pady=5)
entry_username = tk.Entry(login_window)
entry_username.pack(pady=5)

tk.Label(login_window, text="Password").pack(pady=5)
entry_password = tk.Entry(login_window, show="*")
entry_password.pack(pady=5)

tk.Label(login_window, text="Contact (for new users)").pack(pady=5)
entry_contact = tk.Entry(login_window)
entry_contact.pack(pady=5)

tk.Button(login_window, text="Register", command=register).pack(pady=5)
tk.Button(login_window, text="Login", command=login).pack(pady=20)

login_window.mainloop()

```

admin_dashboard.py:

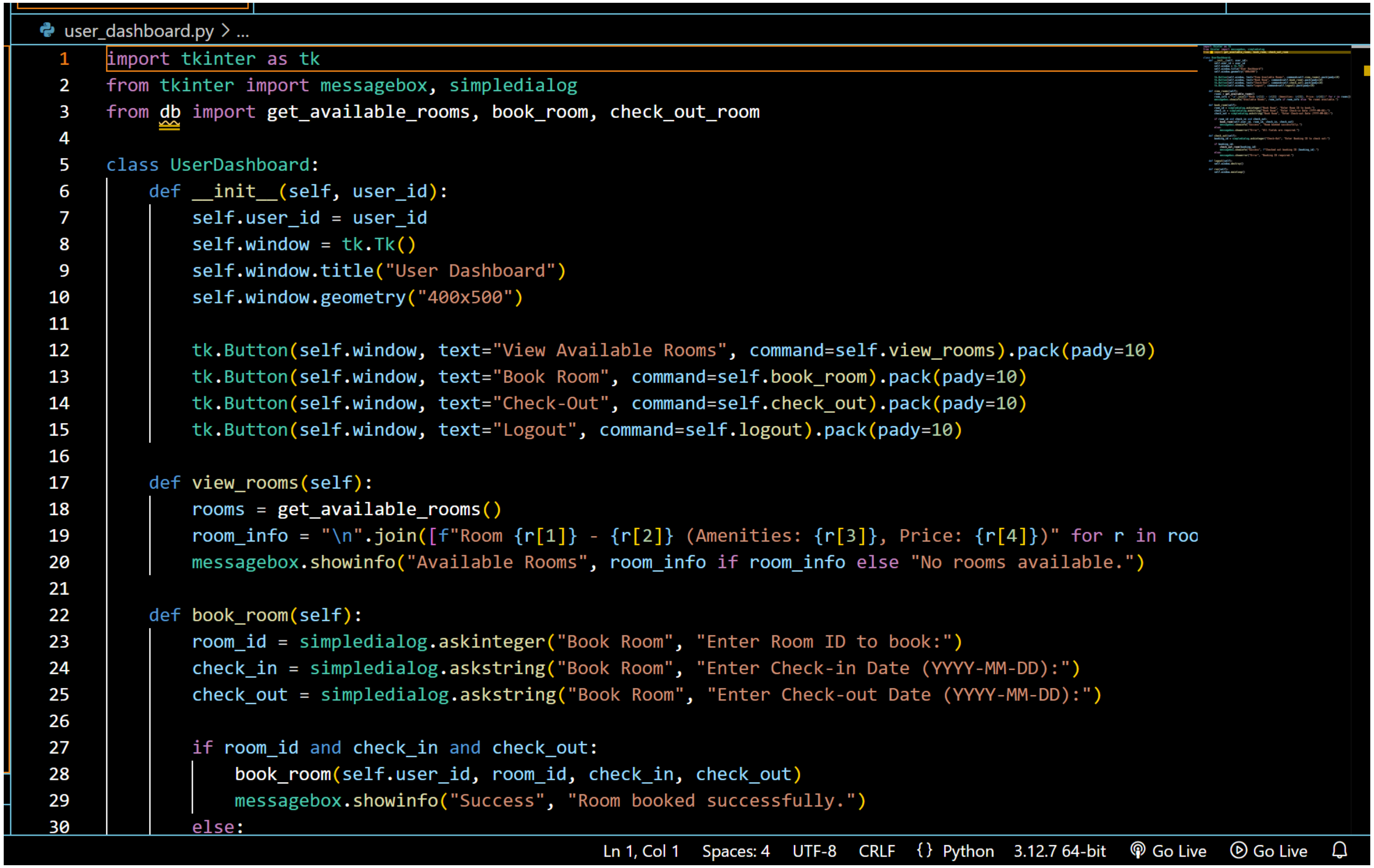


```python
import tkinter as tk
from tkinter import messagebox, simpledialog
from db import add_room, update_room_price, get_available_rooms, get_booking_details

class AdminDashboard:
    def __init__(self):
        self.window = tk.Tk()
        self.window.title("Admin Dashboard")
        self.window.geometry("600x400")


        tk.Label(self.window, text="Admin Dashboard", font=("Arial", 18)).pack(pady=20)


        tk.Button(self.window, text="Add Room", command=self.add_room).pack(pady=10)
        tk.Button(self.window, text="Update Room Price", command=self.update_room_price).pack(pady=10)
        tk.Button(self.window, text="View Available Rooms", command=self.view_available_rooms).pack(pady=10)
        tk.Button(self.window, text="View Booking Details", command=self.view_booking_details).pack(pady=10)

    def add_room(self):
        room_number = simpledialog.askstring("Room Number", "Enter the new room number:")
        room_price = simpledialog.askstring("Room Price", "Enter the price for the new room:")

        if room_number and room_price:
            if add_room(room_number, float(room_price)):
                messagebox.showinfo("Success", f"Room {room_number} added successfully!")
            else:
                messagebox.showerror("Error", "Room number already exists or invalid data.")
        else:
            messagebox.showerror("Error", "Both room number and price are required.")

    def update_room_price(self):
```

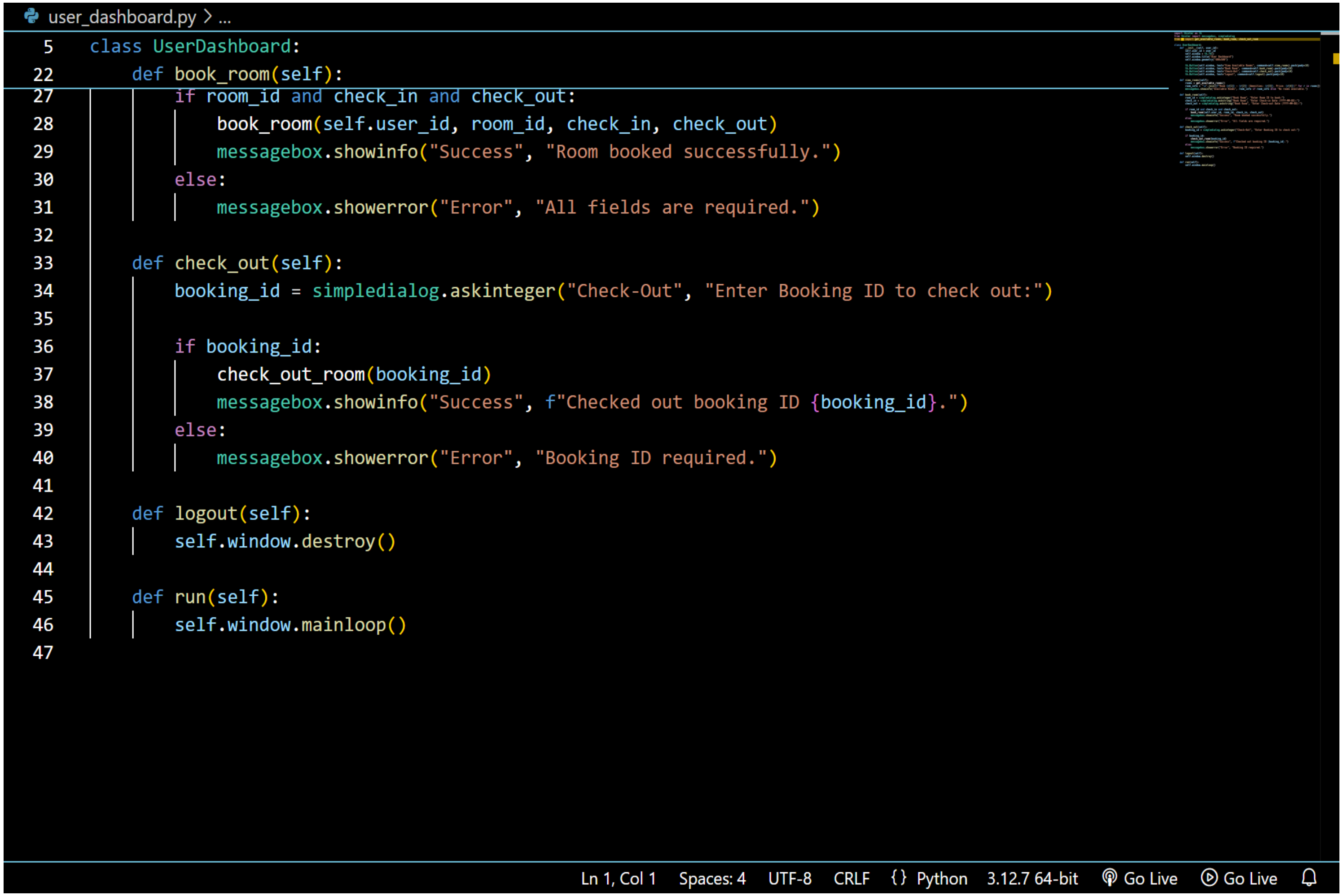

Code.py 1 ●
hello.py
Code.py > AdminDashboard > run
Activate Windows
Go to Settings to activate Windows.

```python
class AdminDashboard:

    def update_room_price(self):
        room_number = simpledialog.askstring("Room Number", "Enter the room number to update:")
        new_price = simpledialog.askstring("New Price", "Enter the new price for the room:")

        if room_number and new_price:
            update_room_price(room_number, float(new_price))
            messagebox.showinfo("Success", f"Room {room_number} price updated to ${new_price}")
        else:
            messagebox.showerror("Error", "Both room number and new price are required.")

    def view_available_rooms(self):
        rooms = get_available_rooms()
        if rooms:
            room_details = "\n".join([f"Room {room[0]}: ${room[1]}" for room in rooms])
            messagebox.showinfo("Available Rooms", room_details)
        else:
            messagebox.showinfo("Available Rooms", "No available rooms at the moment.")

    def view_booking_details(self):
        bookings = get_booking_details()
        if bookings:
            booking_details = "\n".join([f"Booking ID {b[0]} - User {b[1]} - Room {b[2]} (Check-in: {b[3]}, Check-out: {b[4
            messagebox.showinfo("Booking Details", booking_details)
        else:
            messagebox.showinfo("Booking Details", "No bookings available.")

    def run(self):
        self.window.mainloop()
```

user_dashboard.py:

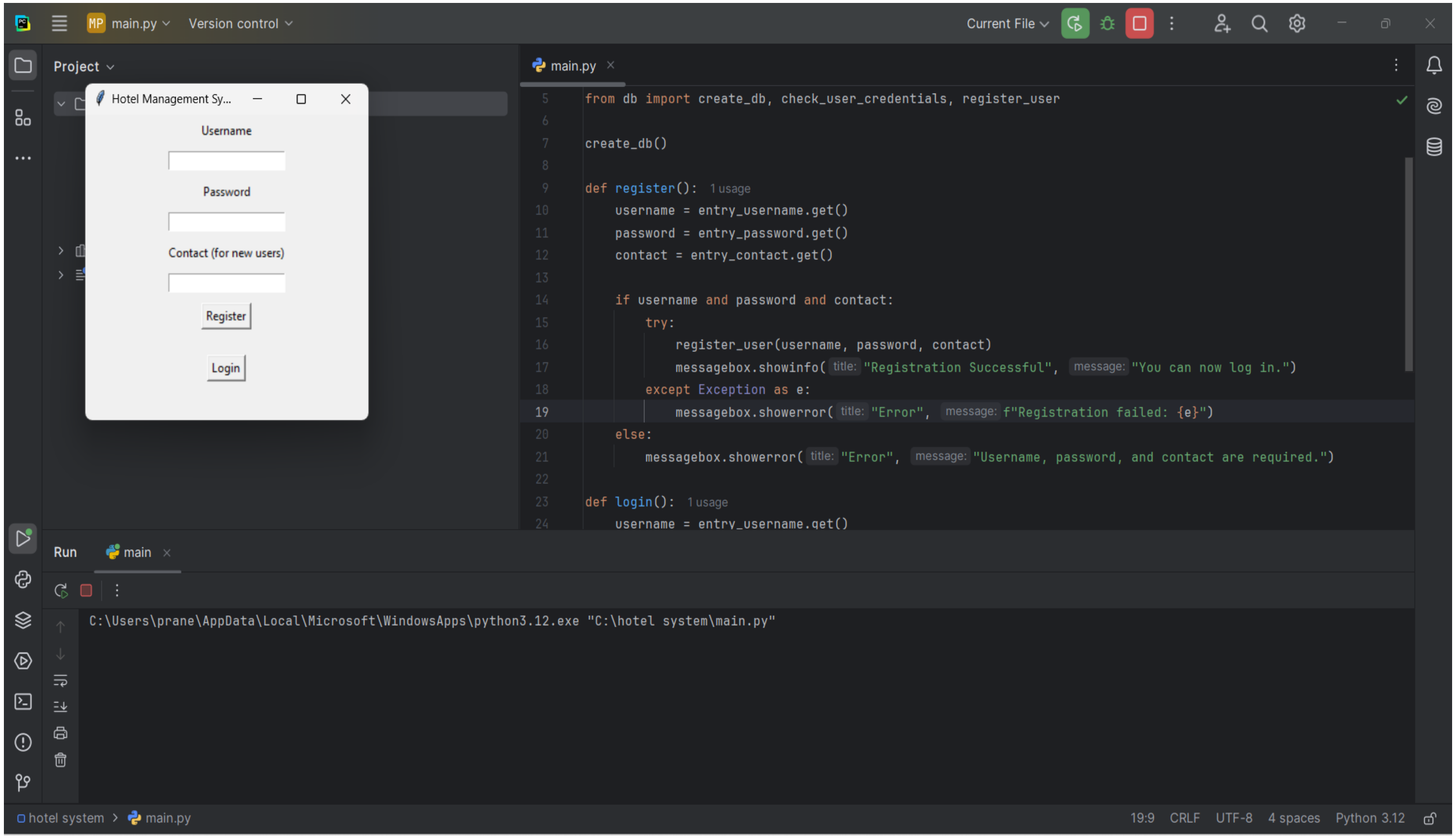


```python
import tkinter as tk
from tkinter import messagebox, simpledialog
from db import get_available_rooms, book_room, check_out_room

class UserDashboard:
    def __init__(self, user_id):
        self.user_id = user_id
        self.window = tk.Tk()
        self.window.title("User Dashboard")
        self.window.geometry("400x500")

        tk.Button(self.window, text="View Available Rooms", command=self.view_rooms).pack(pady=10)
        tk.Button(self.window, text="Book Room", command=self.book_room).pack(pady=10)
        tk.Button(self.window, text="Check-Out", command=self.check_out).pack(pady=10)
        tk.Button(self.window, text="Logout", command=self.logout).pack(pady=10)

    def view_rooms(self):
        rooms = get_available_rooms()
        room_info = "\n".join([f"Room {r[1]} - {r[2]} (Amenities: {r[3]}, Price: {r[4]})" for r in roo
        messagebox.showinfo("Available Rooms", room_info if room_info else "No rooms available.")

    def book_room(self):
        room_id = simpledialog.askinteger("Book Room", "Enter Room ID to book:")
        check_in = simpledialog.askstring("Book Room", "Enter Check-in Date (YYYY-MM-DD):")
        check_out = simpledialog.askstring("Book Room", "Enter Check-out Date (YYYY-MM-DD):")

        if room_id and check_in and check_out:
            book_room(self.user_id, room_id, check_in, check_out)
            messagebox.showinfo("Success", "Room booked successfully.")
        else:
```



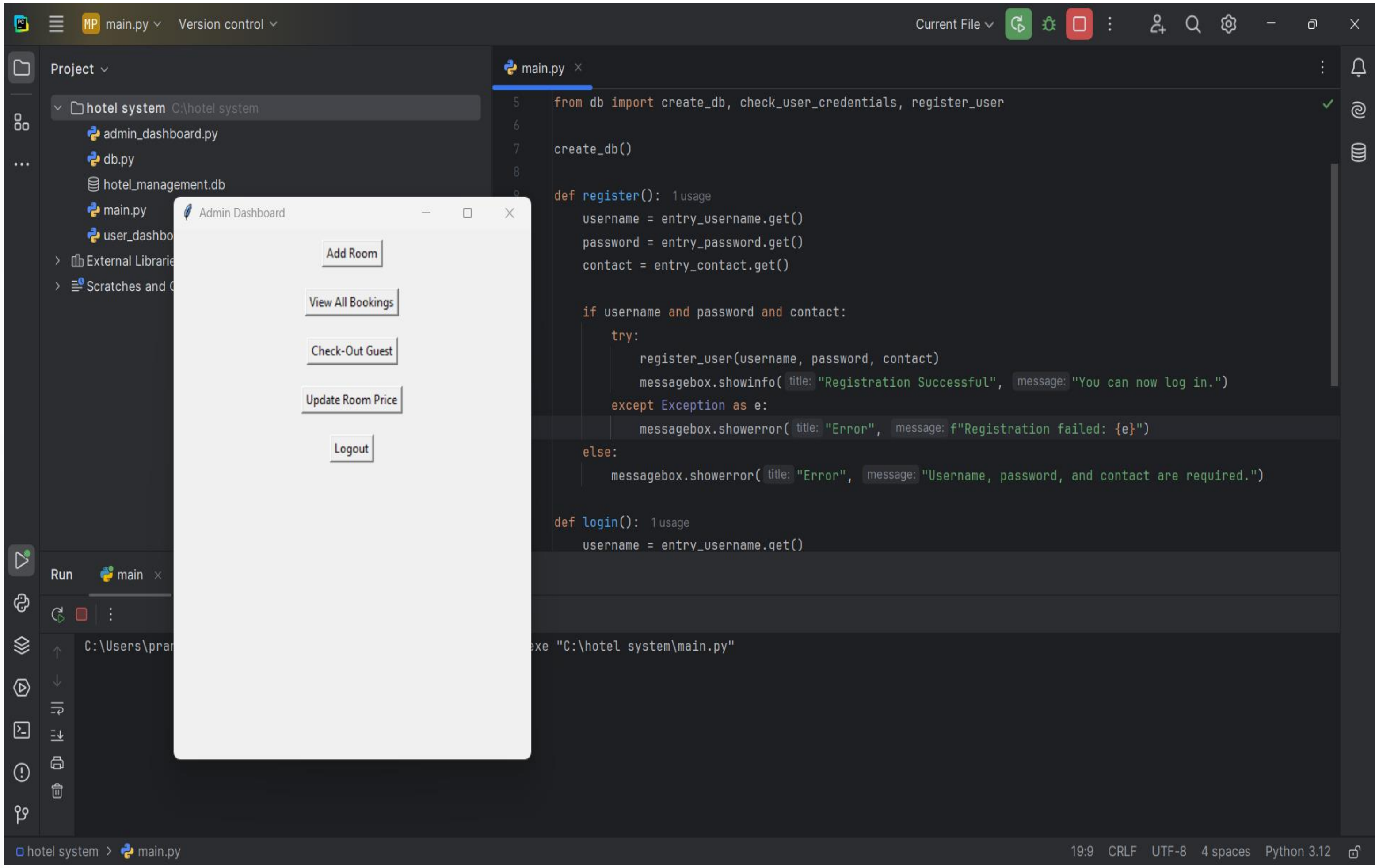
```
class UserDashboard:
    def book_room(self):
        if room_id and check_in and check_out:
            book_room(self.user_id, room_id, check_in, check_out)
            messagebox.showinfo("Success", "Room booked successfully.")
        else:
            messagebox.showerror("Error", "All fields are required.")

    def check_out(self):
        booking_id = simpledialog.askinteger("Check-Out", "Enter Booking ID to check out:")

        if booking_id:
            check_out_room(booking_id)
            messagebox.showinfo("Success", f"Checked out booking ID {booking_id}.")
        else:
            messagebox.showerror("Error", "Booking ID required.")

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

    def run(self):
        self.window.mainloop()

```

OUTPUT OF THE PROJECT



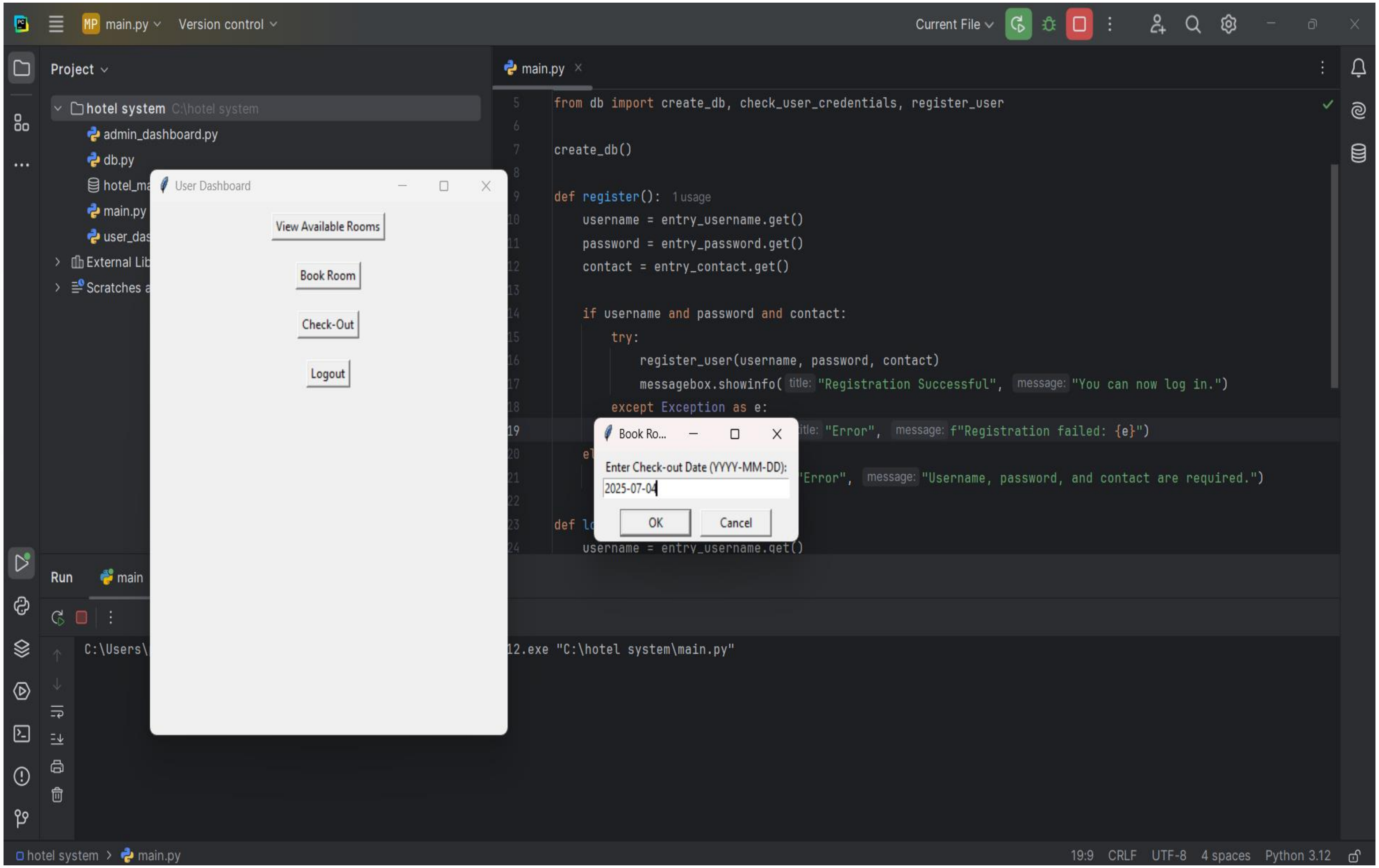
```
from db import create_db, check_user_credentials, register_user

create_db()

def register():  1 usage
    username = entry_username.get()
    password = entry_password.get()
    contact = entry_contact.get()

    if username and password and contact:
        try:
            register_user(username, password, contact)
            messagebox.showinfo( title: "Registration Successful",  message: "You can now log in.")
        except Exception as e:
            messagebox.showerror( title: "Error",  message: f"Registration failed: {e}")
    else:
        messagebox.showerror( title: "Error",  message: "Username, password, and contact are required.")

def login():  1 usage
    username = entry_username.get()
```

```
C:\Users\prane\AppData\Local\Microsoft\WindowsApps\python3.12.exe "C:\hotel system\main.py"
```



```
from db import create_db, check_user_credentials, register_user

create_db()

def register(): 1 usage
    username = entry_username.get()
    password = entry_password.get()
    contact = entry_contact.get()

    if username and password and contact:
        try:
            register_user(username, password, contact)
            messagebox.showinfo( title: "Registration Successful",  message: "You can now log in.")
        except Exception as e:
            messagebox.showerror( title: "Error",  message: f"Registration failed: {e}")
    else:
        messagebox.showerror( title: "Error",  message: "Username, password, and contact are required.")

def login(): 1 usage
    username = entry_username.get()
```

```
C:\Users\prar                              exe "C:\hotel system\main.py"
```

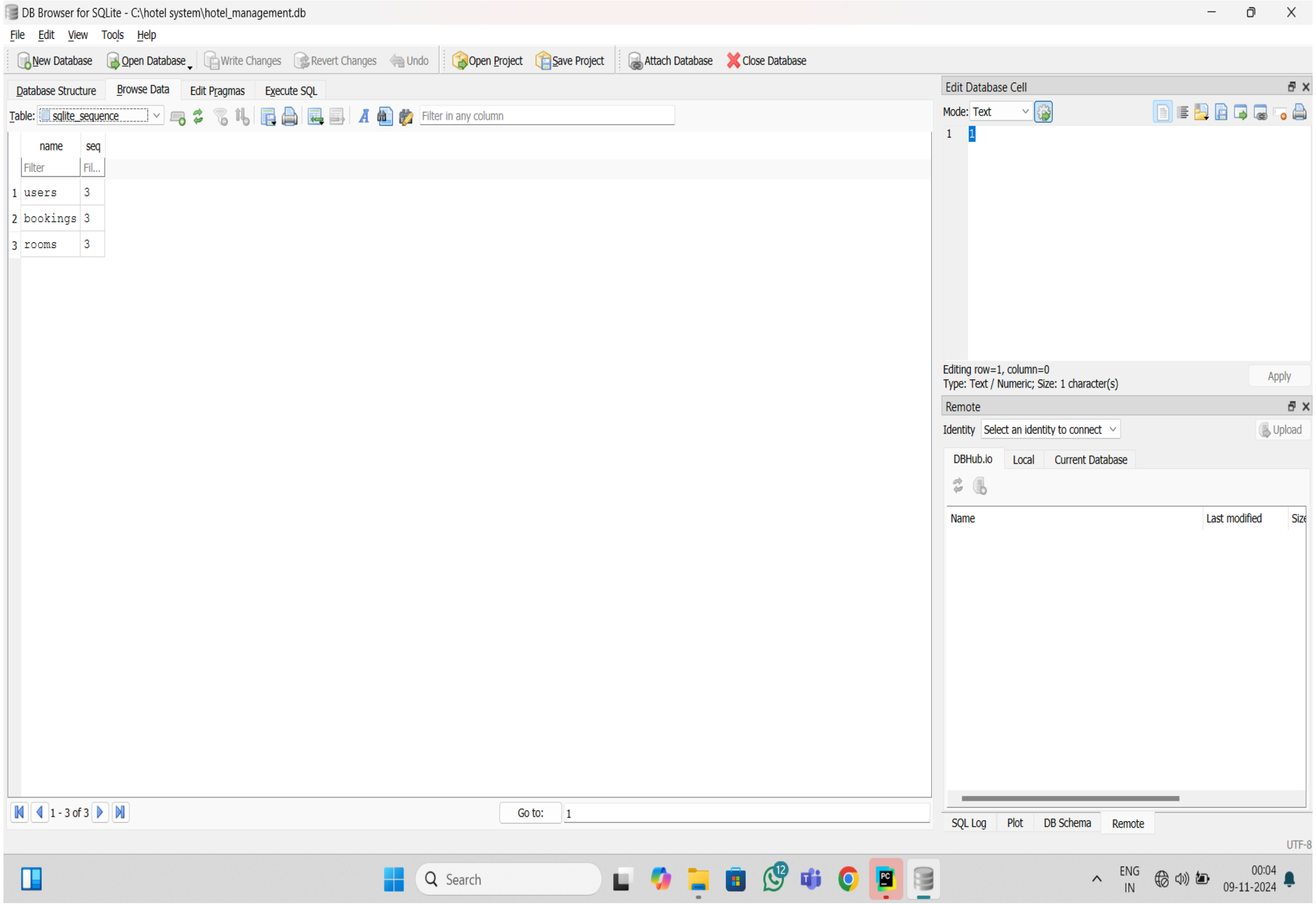


```
from db import create_db, check_user_credentials, register_user

create_db()

def register():  1 usage
    username = entry_username.get()
    password = entry_password.get()
    contact = entry_contact.get()

    if username and password and contact:
        try:
            register_user(username, password, contact)
            messagebox.showinfo( title: "Registration Successful",  message: "You can now log in.")
        except Exception as e:
                                              itle: "Error",  message: f"Registration failed: {e}")
        el
                                              "Error",  message: "Username, password, and contact are required.")

def lo
    username = entry_username.get()
```

```
C:\Users\                    12.exe "C:\hotel system\main.py"
```

Enter Check-out Date (YYYY-MM-DD):

2025-07-04



DB Browser for SQLite - C:\hotel system\hotel_management.db

Table: sqlite_sequence

	name	seq
1	users	3
2	bookings	3
3	rooms	3

1 - 3 of 3

Go to: 1

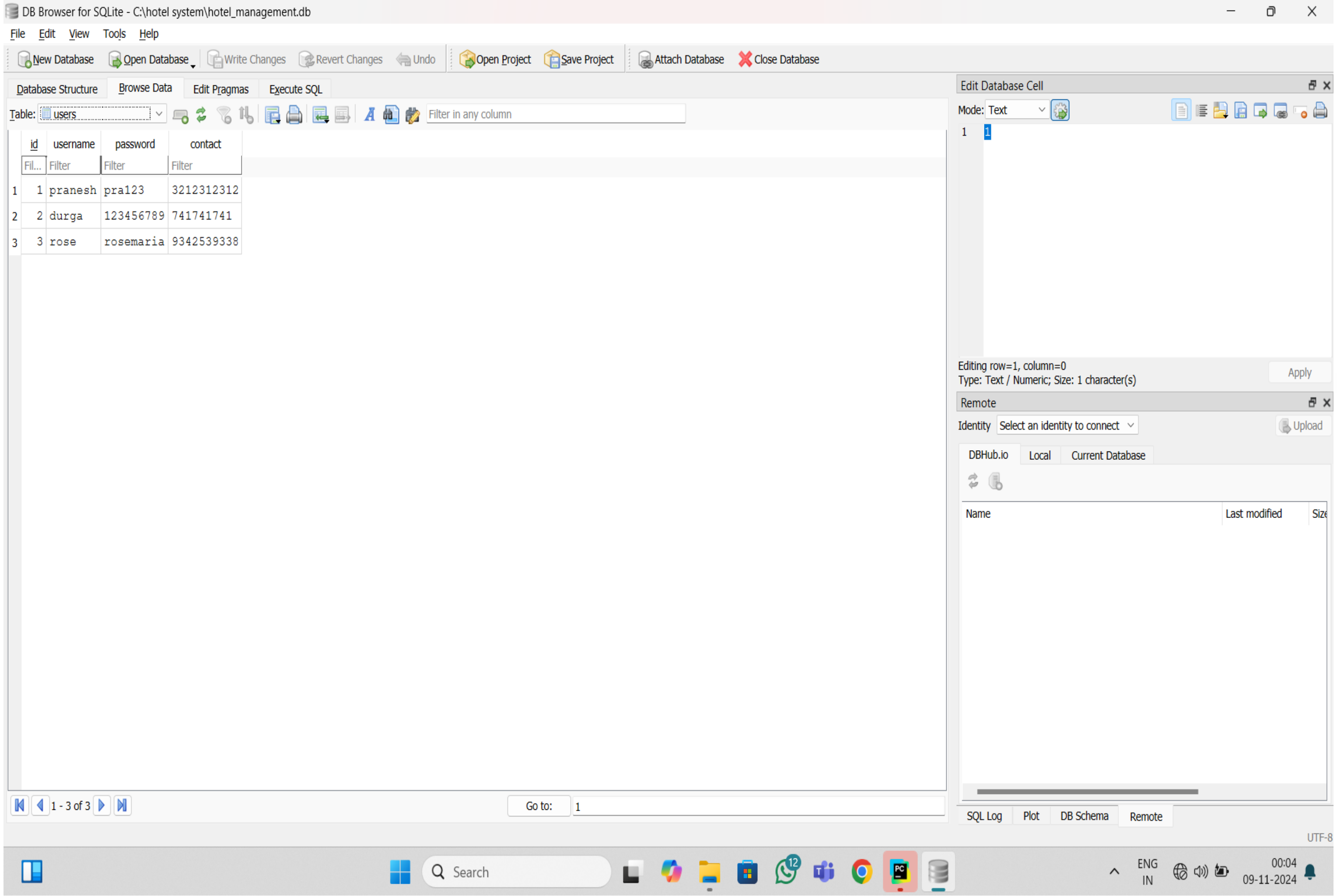
Mode: Text

1 1

Editing row=1, column=0
Type: Text / Numeric; Size: 1 character(s)

Identity Select an identity to connect

00:04
09-11-2024

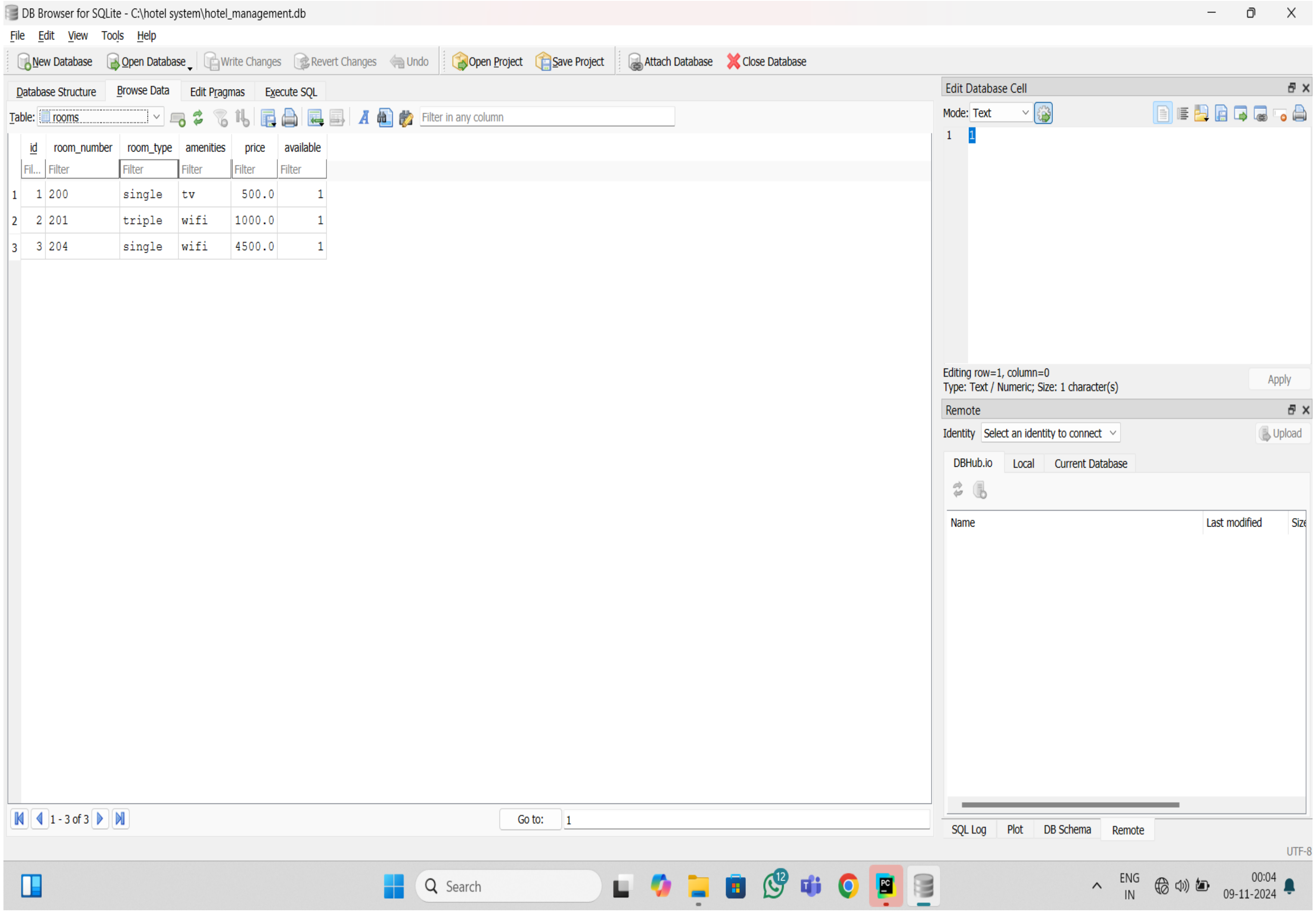


DB Browser for SQLite - C:\hotel system\hotel_management.db

Table: users

	id	username	password	contact
1	1	pranesh	pra123	3212312312
2	2	durga	123456789	741741741
3	3	rose	rosemaria	9342539338

Editing row=1, column=0
Type: Text / Numeric; Size: 1 character(s)



DB Browser for SQLite - C:\hotel system\hotel_management.db

Table: rooms

Filter in any column

	id	room_number	room_type	amenities	price	available
1	1	200	single	tv	500.0	1
2	2	201	triple	wifi	1000.0	1
3	3	204	single	wifi	4500.0	1

1 - 3 of 3

Go to: 1

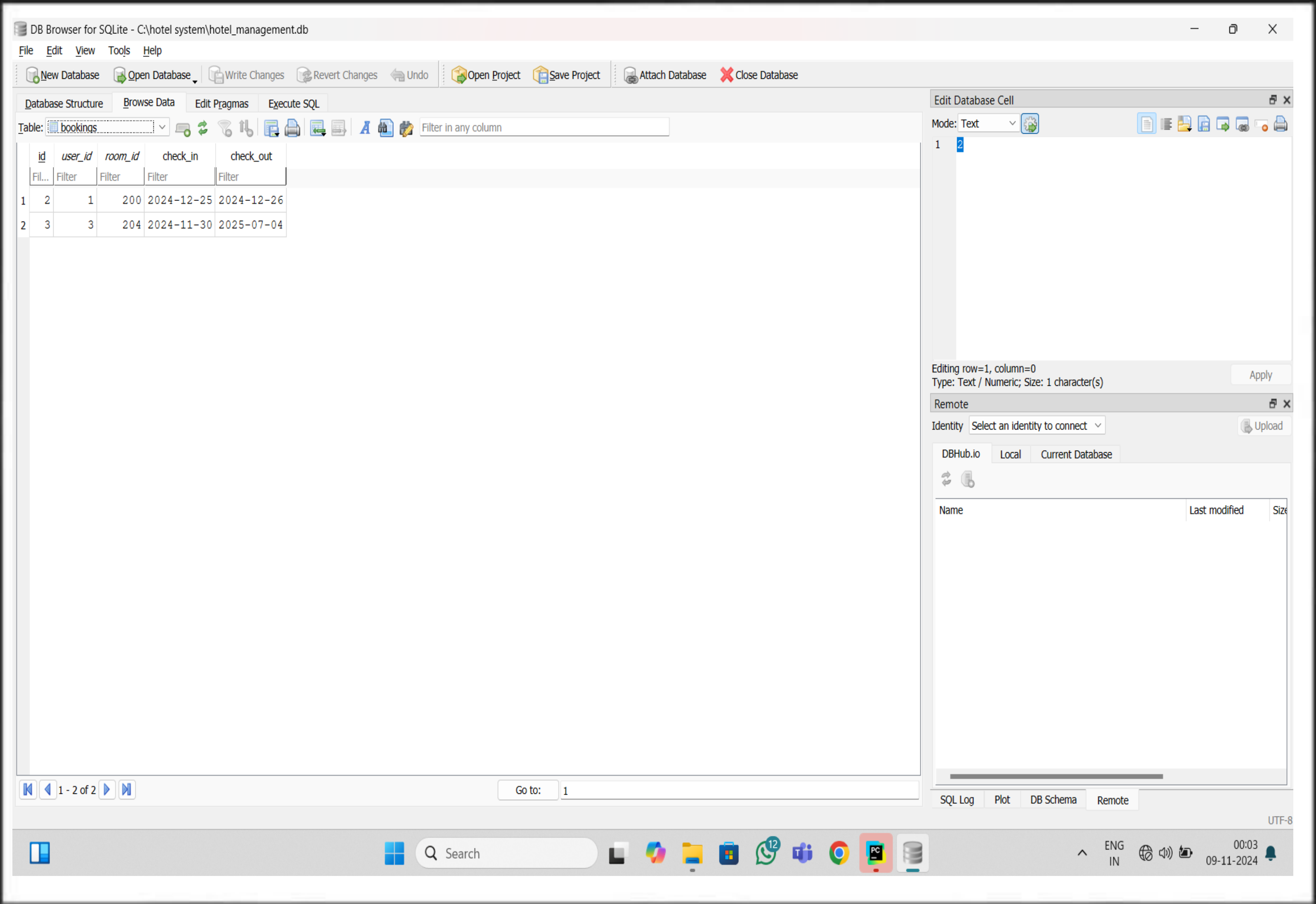
Mode: Text

1 1

Editing row=1, column=0
Type: Text / Numeric; Size: 1 character(s)

Identity: Select an identity to connect

00:04
09-11-2024



DB Browser for SQLite - C:\hotel system\hotel_management.db

Table: bookings

	id	user_id	room_id	check_in	check_out
1	2	1	200	2024-12-25	2024-12-26
2	3	3	204	2024-11-30	2025-07-04

1 - 2 of 2

Go to: 1

Mode: Text

1 2

Editing row=1, column=0
Type: Text / Numeric; Size: 1 character(s)

Identity: Select an identity to connect

00:03
09-11-2024

CONCLUSION

The Python-based Hotel Management System automates room booking and availability, manages user access securely, and simplifies hotel operations with separate admin and user dashboards.

Thank You !