UNIVERSITI TEKNOLOGI MARA

(UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN INFORMATICS LIBRARY

(CDIM144)

PROGRAMMING FOR LIBRARIES [IML208]

PREPARED BY:

NAME	STUDENT ID
NUR AIN ZULAIKHA BINTI ABD RAHIM	2023879878

GROUP: KCDIM144 3E

PREPARED FOR: MOHD FIRDAUS BIN MOHD HELMI

SUBMISSION DATE:

18TH DECEMBER 2024

INDIVIDUAL ASSIGMENT:

HOTEL BOOKING SYSTEM

NUR AIN ZULAIKHA BINTI ABD RAHIM

2023879878

KCDIM144 3E

UNIVERSITI TEKNOLOGI MARA (UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS

18TH DECEMBER 2024

TABLE OF CONTENT

ACKNOWLEDGEMENT

Assalamualaikum w.b.t,

First of all, I would like to thank the Almighty (Allah SWT) for giving me the strength and patient to complete my first assignment in subject IML 208. Without His blessings, I wouldn't have gone this so far. In completing my assignment, I had to ask few people for guidance and hereby I consent my greatest gratitude to them.

I highly indebted to our great lecturer, Sir Mohd Firdaus Bin Mohd Helmi for his guidance and constant supervision as well as providing necessary information regarding this assignment.

I would like to expand my deepest gratitude to all who have directly and indirectly guided and help me in completing this assignment. This assignment report will not successfully complete without their helps and guidance. Thank you so much and may Allah grant us His blessings.

X_____

NAME: NUR AIN ZULAIKHA BINTI ABD RAHIM

STUDENT ID: 2023879878

CLASS: KCDIM144 3E

UNIVERSITI TEKNOLOGI MARA

STUDENT PLEDGE OF ACADEMIC INTEGRITY

As a student of Universiti Teknologi MARA (UiTM), it is my responsibility to act in accordance with UiTM's academic assessment and evaluation policy. I hereby pledge to act and uphold academic integrity and pursue scholarly activities in UiTM with honesty and responsible manner. I will not engage or tolerate acts of academic dishonesty, academic misconduct, or academic fraud including but not limited to:

a. **Cheating**: Using or attempt to use any unauthorized device, assistance, sources, practice or materials while completing academic assessments. This include but not limited to copying from another, allowing others to copy, unauthorized collaboration on an assignment or open book tests, or engaging in any act or conduct that can be construed as cheating.

b. **Plagiarism**: Using or attempts to use the work of others (ideas, design, words, art, music, etc.) without acknowledging the source; using or purchasing materials prepared by another person or agency or engaging in other behaviour that a reasonable person would consider as plagiarism.

c. **Fabrication**: Falsifying data, information, or citations in any academic assessment and evaluation.

d. **Deception**: Providing false information with intend to deceive an instructor concerning any academic assessment and evaluation.

e. **Furnishing false information**: Providing false information or false representation to any UiTM official, instructor, or office.

With this pledge, I am fully aware that I am obliged to conduct myself with utmost honesty and integrity. I fully understand that a disciplinary action can be taken against me if I, in any manner, violate this pledge.

X

_____

Name: NUR AIN ZULAIKHA BINTI ABD RAHIM

Matric Number: 2023879878

Course Code: IML208

Programme code: CDIM144

Faculty / Campus: COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

PROJECT NAME: HOTEL RESERVATION SYSTEM

File Name: hotel_booking.py

PROMPT DATA:

i. **Name**: The name of the guest making the booking.
ii. **Room ID**: A unique identifier for each room in the hotel.
iii. **Room Type**: Specifies the type of room (e.g., Single, Double, Suite).
iv. **Price per Night**: The cost of the room per night.
v. **Nights Stayed**: The number of nights the room will be occupied.

FUNCTIONS:

i. **Create Data:**
- **Add a room to the hotel**: New rooms can be created by specifying room type, price, and availability.
- **Create a user to make a booking**: The system allows adding user details for the purpose of reserving a room.
- **Book a room and store user details**: When a room is booked, the user's information (name, room details, dates) is stored for future reference.

ii. **Read Data:**
- **View available rooms**: The system shows rooms that are currently available for booking.
- **View current reservations**: Displays details of all existing booking (user details, room types, duration).
- **View the total revenue and average room price**: Tracks the total revenue from bookings and calculates the average price per night for rooms.

iii. **Update Data:**
- **Update a reservation by cancelling or modifying room bookings**: Allows users to change booking details or cancel reservation altogether.

iv. **Delete Data:**
- **Cancel an existing reservation:** Enables users to remove a reservation if it is no longer needed.

CONDITIONAL STATEMENT:

Yes, the code includes conditional statements using **if**, **elif**, and **else**:

- **If** is used for checking whether a room is available or if a booking is successful.

- **elif** is used to manage more complex checks, such as determining if the room is already booked for the chosen dates.

- **else** is used to handle cases when no condition is met (e.g., when a room is not found or when no reservation exists).

GUI:

Yes. the system has a graphical user interface (GUI) created using **tkinter.** The interface includes input fields for the user's name, room details, booking dates, and buttons for booking and cancelling reservations.

- **Input fields:**
  - **Name:** For entering the user's name.
  - **Room ID:** To specify which room the user wants to book or cancel.
  - **Nights Stayed:** To specify the number of nights the user intends to stay.

- **Buttons:**
  - **Book Room:** Allows the user to book a room after entering the necessary details.

  - **Cancel Reservation:** Allows the user to cancel an existing reservation.
  - **View Available Rooms:** Displays the lists of rooms that are still available.
  - **View Reservations:** Shows a list of all current reservations.
  - **View Revenue:** Displays the total revenue and average room price for current bookings.

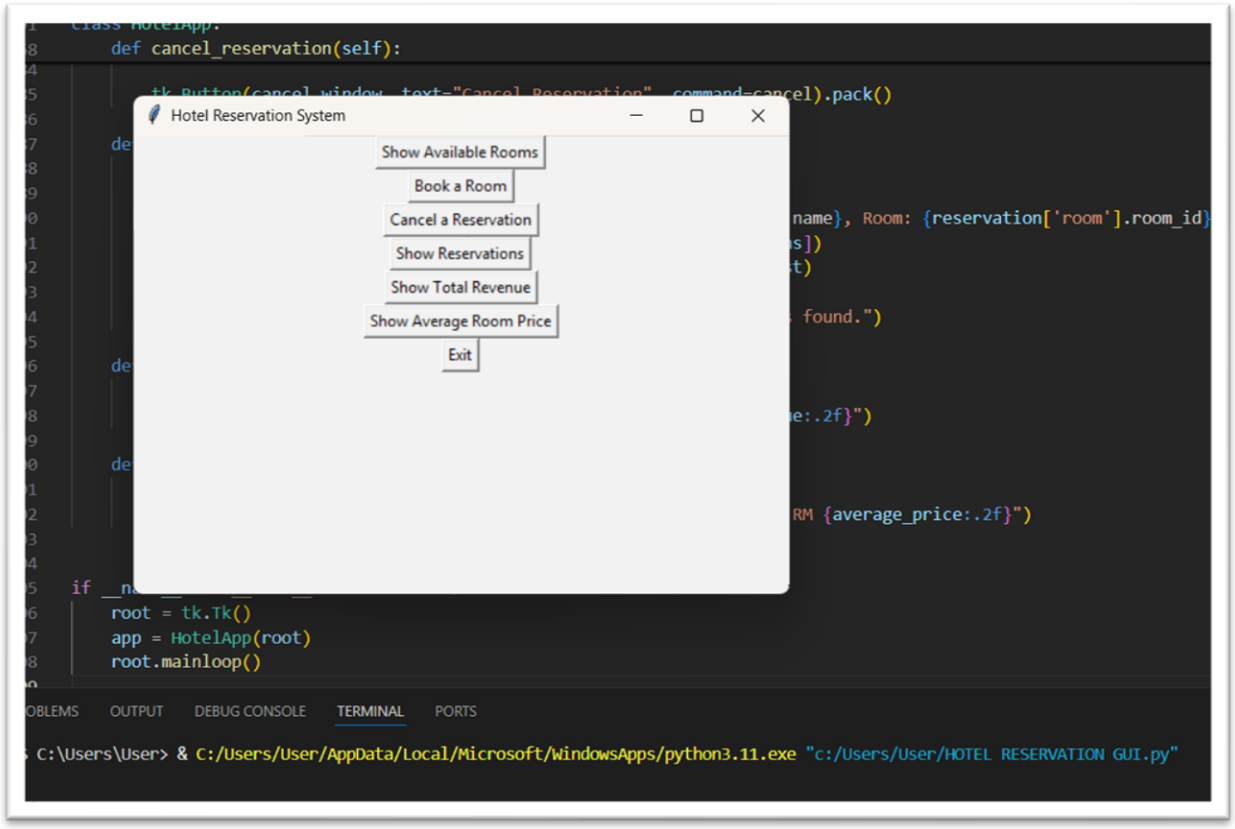

Figure 1.0

RESULT:

- Output:

i. Show Available Room
   - Users can easily view the rooms that are available for booking, along with their price and type.

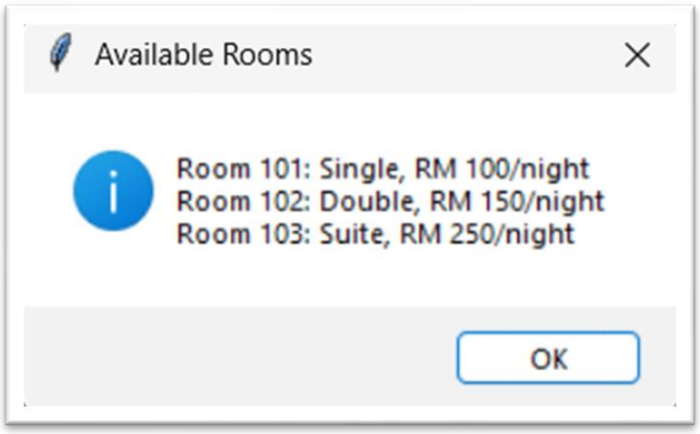

Figure 2.0

ii. Book a Room
   - Users can enter their name, choose a room, and specify the number of nights they wish to stay. The system checks availability and processes the booking if the room is available.

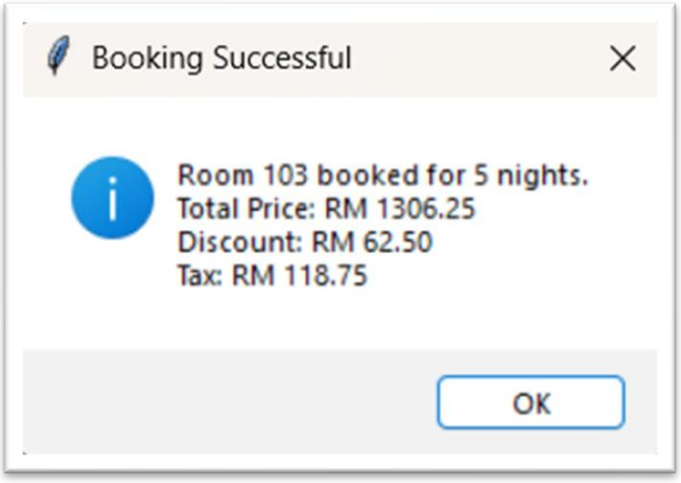

Figure 3.0

iii. Cancel a Reservation

- Users can cancel their booking, which makes the room available for others.

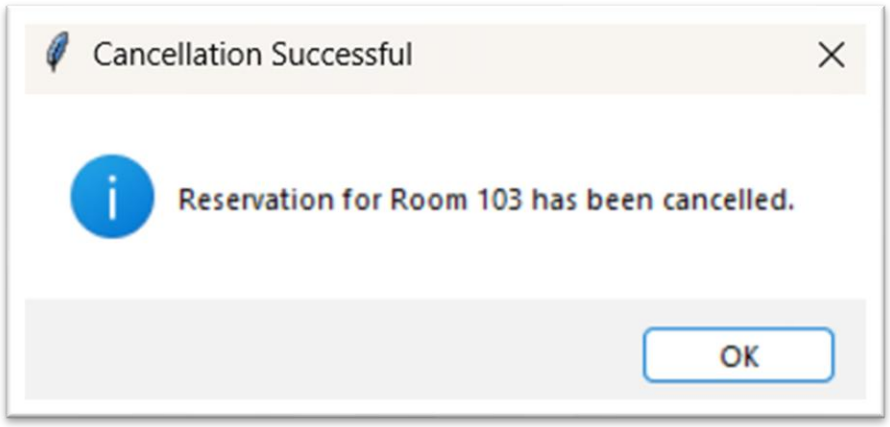

Figure 4.0

iv. Show Reservations

- The system shows the list of active reservations, so the user can keep track of bookings.

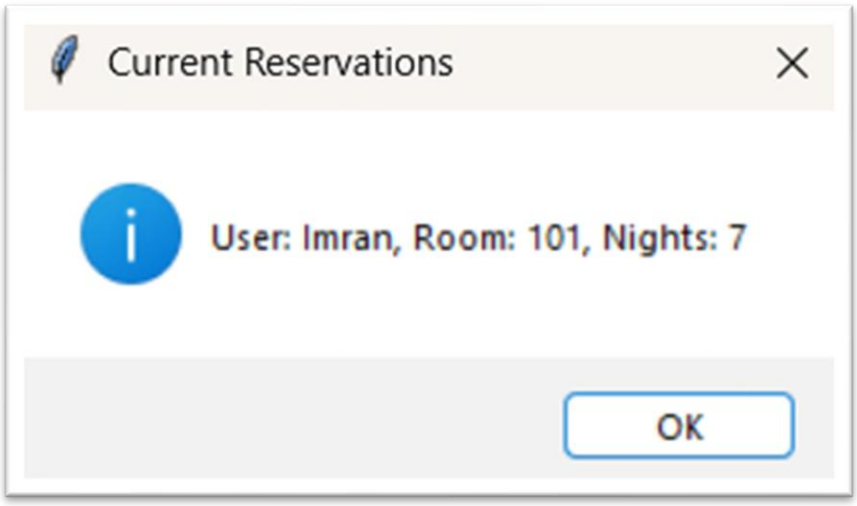

Figure 5.0

v. Show Total Revenue

- The system calculates the total revenue generated from bookings.

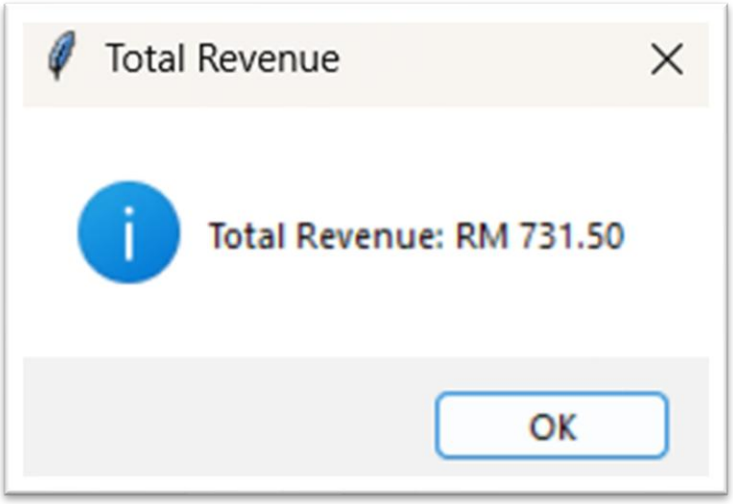

Figure 6.0

vi. Show Average Room Price

- Displays the average price for the rooms.

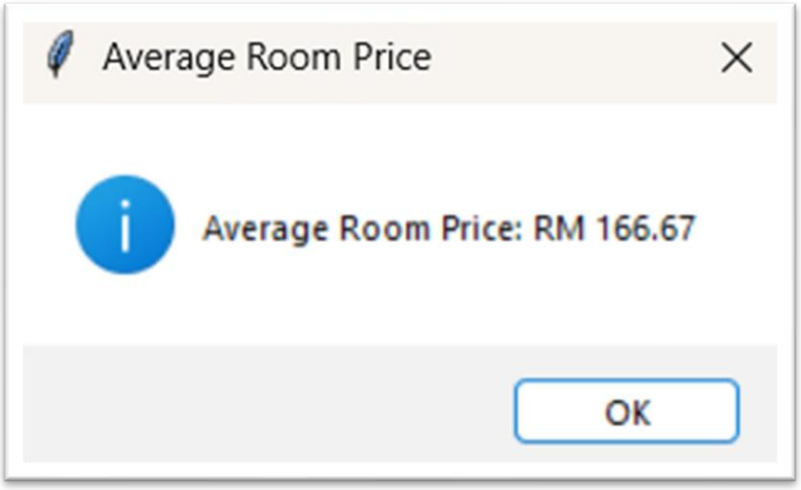

Figure 7.0

- The GUI present the user with options such as booking a room, showing available rooms, viewing reservations, and more.

STRENGHT:

- **Seamless Management:** The system allows users to easily book and cancel rooms, with automatic updates on room availability after each action. Once a room is booked or canceled, its status is immediately updated to reflect its availability. This seamless management ensures that double bookings are prevented, keeping the system efficient and preventing user confusion.

- **User-Friendly GUI:** The graphical user interface is designed to be intuitive, making it easy for users to navigate through the booking process. The input fields and buttons are clearly labeled, guiding users step-by-step. Real-time feedback, such as success or error messages, keeps users informed about the outcome of the actions, which helps maintain a smooth user experience throughout.

- **Modular Design:** The system is built with a modular approach, making it both scalable and maintainable. This modular design allows for easy future enhancements, such as adding new features enhancements, such as adding new features like payment integrations or discounts without disrupting the core functionality. Additionally, the use of reusable code reduces redundancy and simplifies updates to the system, making it easier to manage long-term.

- **Efficient Data Handling:** The system efficiently handles data related to reservations, room availability, and financial tracking. It stores key information such as room IDs, user names, and the numbers of nights booked. Real-time updates ensure that room availability is accurate and that users can always view up-to-date reservation details, making the system reliable for both customers and hotel management.

KAIZEN (Room for Improvement):

- **Room Availability by Date:** Currently, the system does not account for specify check-in and check-out dates. To prevent double booking, the system can be enhanced to track rooms' availability by date, not just by availability status.

- **Advanced Error Handling:** While the system handles basic input errors (e.g., non-numeric values for room ID or nights), more robust error handling could be added, such as validating the phone number format or confirming that entered dates fall within a valid range.

- **Enhanced User Interface:** The design could be improved by adding icons for the buttons, introducing a color scheme for better visual appeal, and reorganizing the layout for better flow. Additional features like a calendar widget for selecting check-in/check-out dates could also improve the user experience.

- **Data Persistence:** Currently, the system does not retain any data after the application is closed. Implementing file-based storage (e.g., CSV, JSON) or integrating a simple database would allow data to persist across sessions, ensuring that room availability and reservations are not lost when the program restarts.