UNIVERSITI TEKNOLOGI MARA

(UiTM) KEDAH, KAMPUS SUNGAI PETANI

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATICS (CDIM144)

PROGRAMMING FOR LIBRARIES (IML208)

INDIVIDUAL ASSIGNMENT : TICKET BUS BOOKING SYSTEM

PREPARED BY :

NUR FIDA FARHANA BINTI ROSLI (2023615074)

KCDIM1443B

PREPARED FOR :

MOHD FIRDAUS BIN MOHD HILMIE

SUBMISSION DATE :

18th DECEMBER 2024

ACKNOWLEDGEMENT

Assalamualaikum w.b.t,

Alhamdulillah, I want to thank God for making it easier for me to complete my assignment. With his help, I was able to complete this assignment successfully.

I, as a student from CDIM144 library informatics, would like to express my appreciation to my lecturer, Sir Mohd Firdaus bin Mohd Hilmie he always helps me complete this assignment by giving clear explanations. Without his attention and supervision, it would be difficult for me to complete this assignment well.

Last but not least, I also want to thank my classmates who were involved directly or indirectly in helping me while I completed this assignment. They helped me a lot by answering my questions in the group or in personal messages that could eliminate my confusion and doubts.

NAME : NUR FIDA FARHANA BINTI ROSLI
ID STUDENT : 2023615074
CLASS : KCDIM1443B

TABLE OF CONTENT

PROJECT NAME : Ticket Bus Booking System

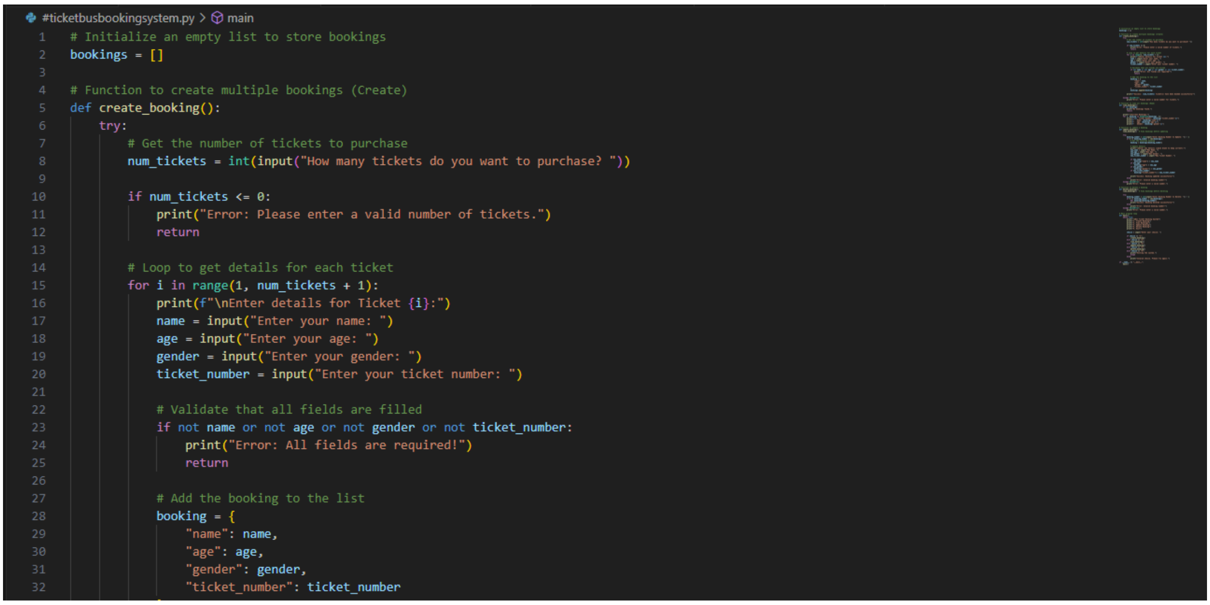
FILE NAME : ticketbusbookingsystem.py

PROMPT DATA :

i. Name
ii. Age
iii. Gender
iv. Ticket Number

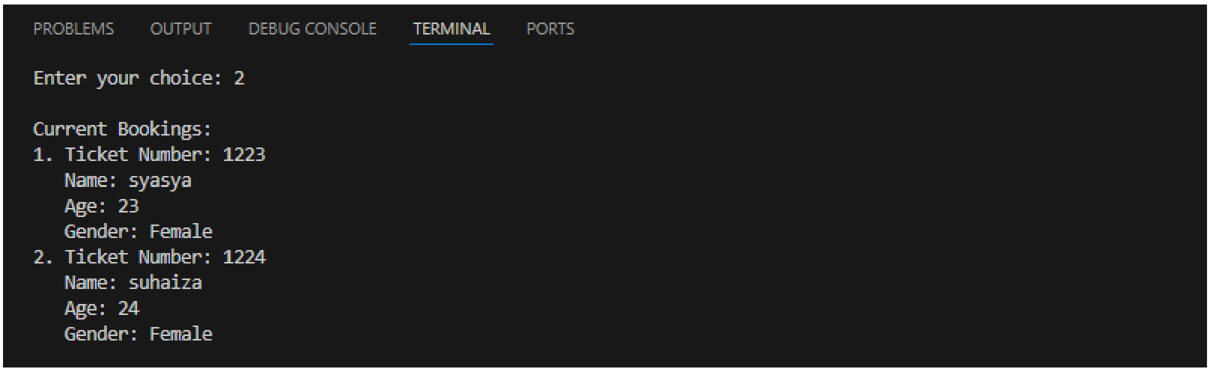
FUNCTION :

i. Create data



```python
# Initialize an empty list to store bookings
bookings = []

# Function to create multiple bookings (Create)
def create_booking():
    try:
        # Get the number of tickets to purchase
        num_tickets = int(input("How many tickets do you want to purchase? "))

        if num_tickets <= 0:
            print("Error: Please enter a valid number of tickets.")
            return

        # Loop to get details for each ticket
        for i in range(1, num_tickets + 1):
            print(f"\nEnter details for Ticket {i}:")
            name = input("Enter your name: ")
            age = input("Enter your age: ")
            gender = input("Enter your gender: ")
            ticket_number = input("Enter your ticket number: ")

            # Validate that all fields are filled
            if not name or not age or not gender or not ticket_number:
                print("Error: All fields are required!")
                return

            # Add the booking to the list
            booking = {
                "name": name,
                "age": age,
                "gender": gender,
                "ticket_number": ticket_number
```

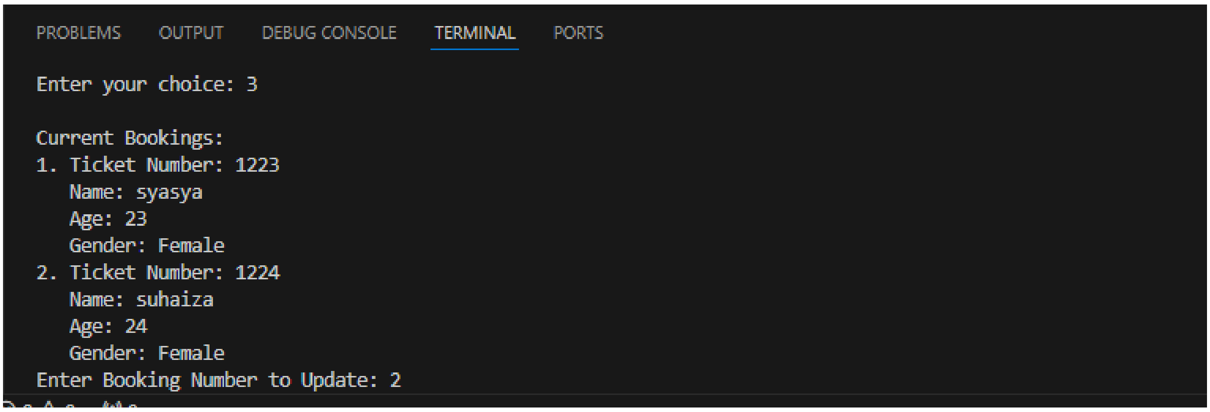
ii. Read data



```
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 24
   Gender: Female
```

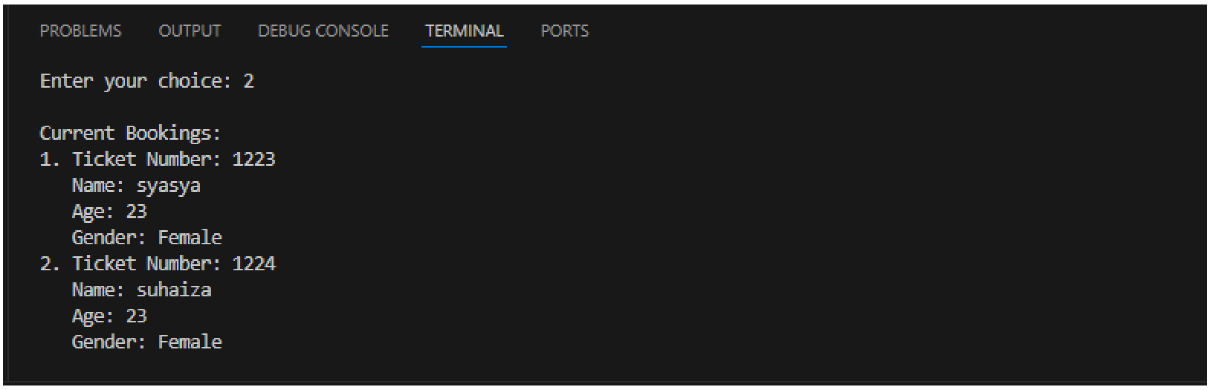
iii. Update data

Before :



```
Enter your choice: 3

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 24
   Gender: Female
Enter Booking Number to Update: 2
```

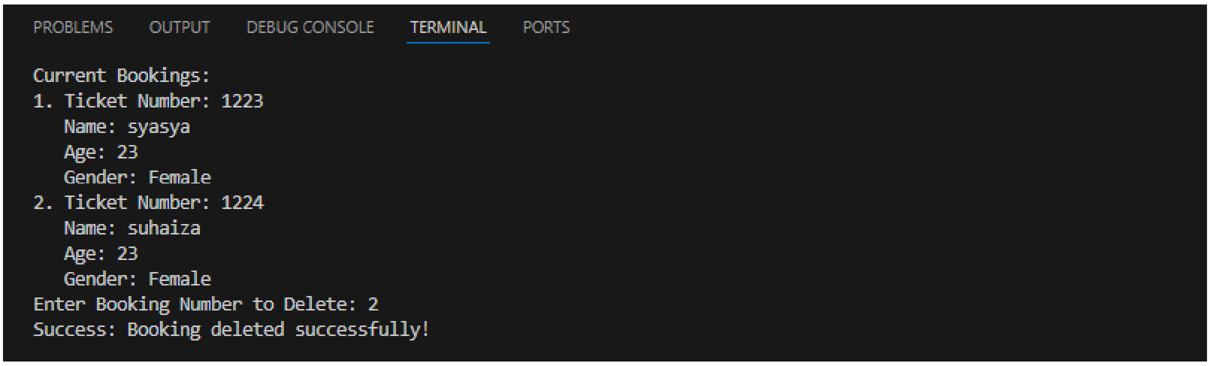
After :



```
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 23
   Gender: Female
```

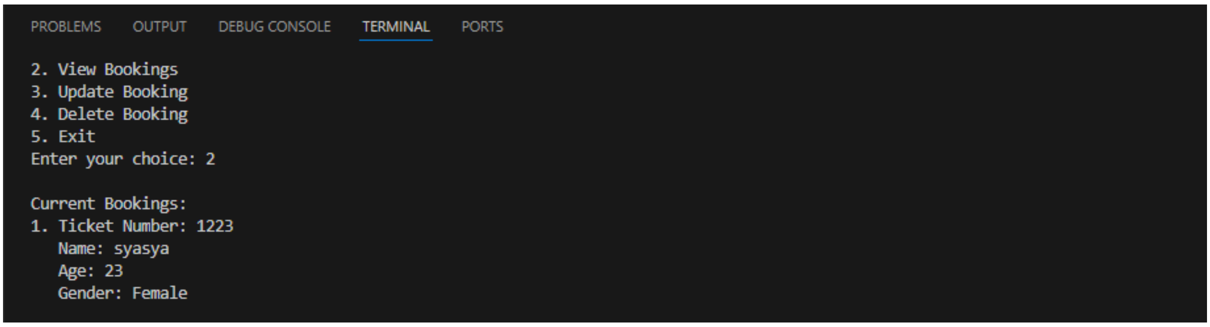
iv. Delete existing data

Before :



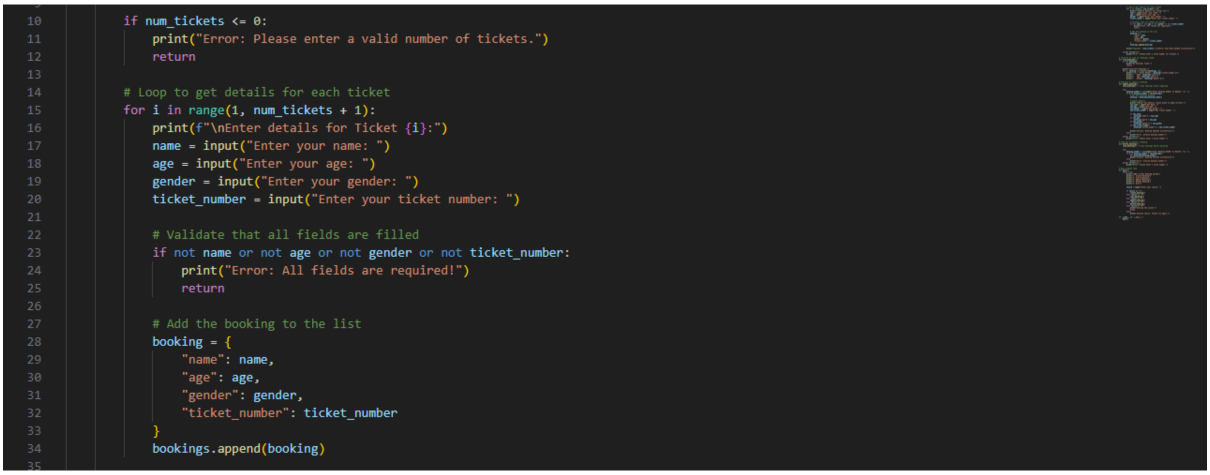
```
Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 23
   Gender: Female
Enter Booking Number to Delete: 2
Success: Booking deleted successfully!
```

After :

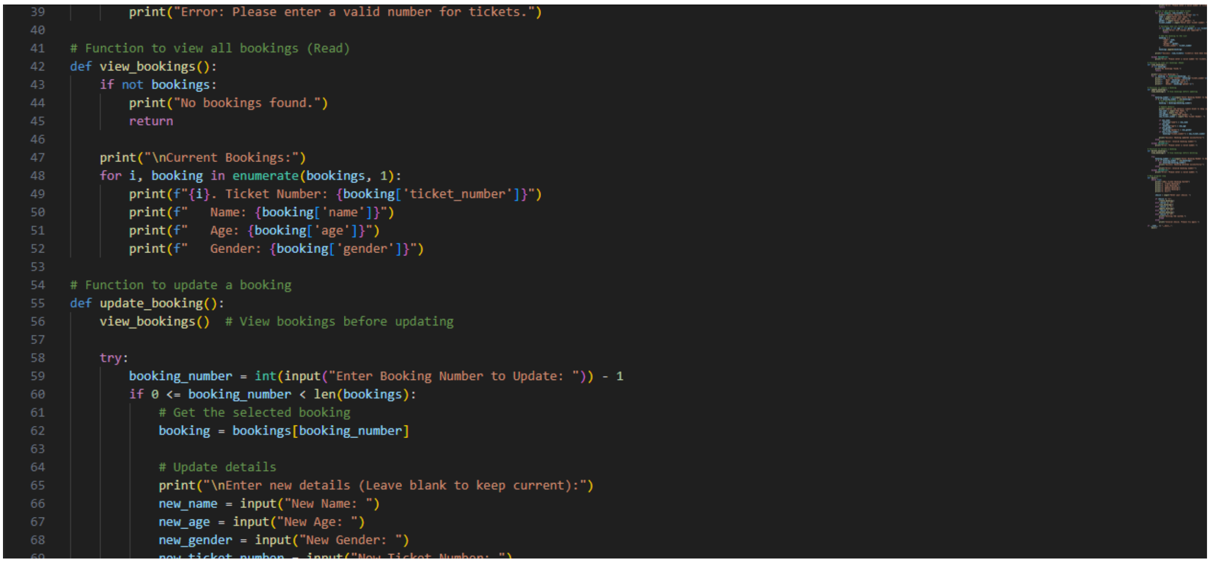


```
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
```

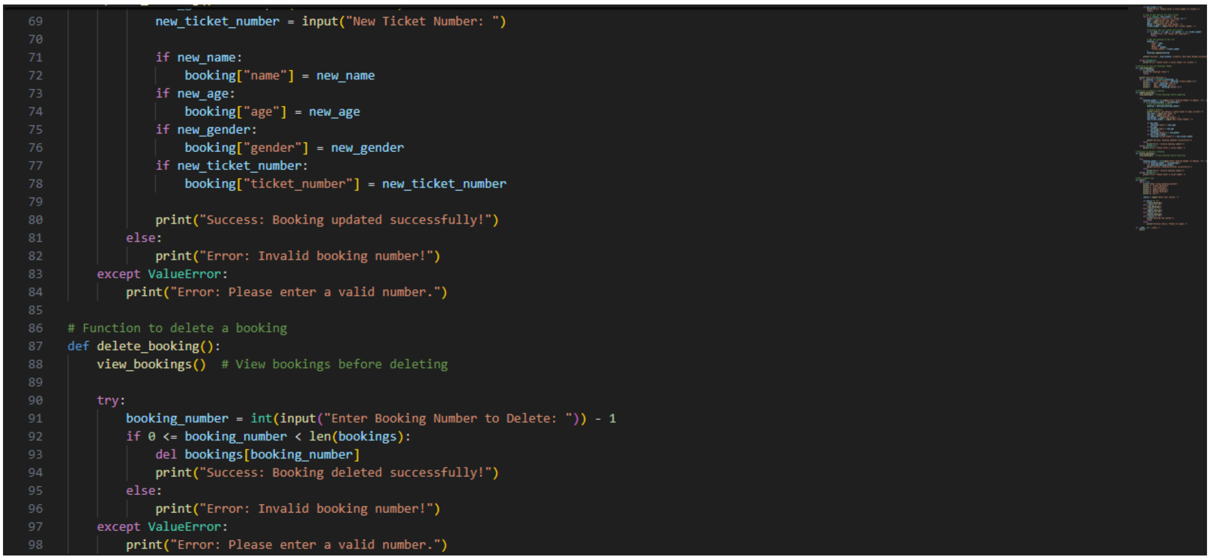
CONDITIONAL STATEMENT : Yes



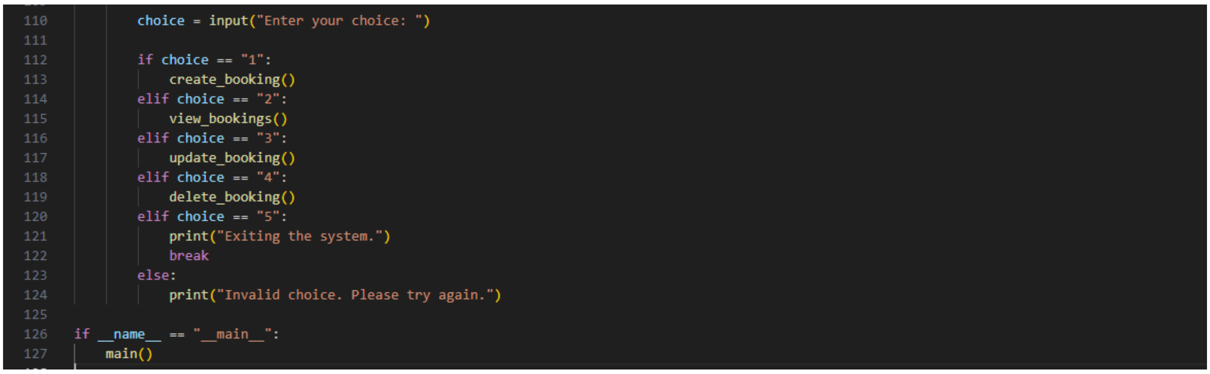
```
        if num_tickets <= 0:
            print("Error: Please enter a valid number of tickets.")
            return

        # Loop to get details for each ticket
        for i in range(1, num_tickets + 1):
            print(f"\nEnter details for Ticket {i}:")
            name = input("Enter your name: ")
            age = input("Enter your age: ")
            gender = input("Enter your gender: ")
            ticket_number = input("Enter your ticket number: ")

            # Validate that all fields are filled
            if not name or not age or not gender or not ticket_number:
                print("Error: All fields are required!")
                return

            # Add the booking to the list
            booking = {
                "name": name,
                "age": age,
                "gender": gender,
                "ticket_number": ticket_number
            }
            bookings.append(booking)

```



```
        print("Error: Please enter a valid number for tickets.")

# Function to view all bookings (Read)
def view_bookings():
    if not bookings:
        print("No bookings found.")
        return

    print("\nCurrent Bookings:")
    for i, booking in enumerate(bookings, 1):
        print(f"{i}. Ticket Number: {booking['ticket_number']}")
        print(f"   Name: {booking['name']}")
        print(f"   Age: {booking['age']}")
        print(f"   Gender: {booking['gender']}")

# Function to update a booking
def update_booking():
    view_bookings()  # View bookings before updating

    try:
        booking_number = int(input("Enter Booking Number to Update: ")) - 1
        if 0 <= booking_number < len(bookings):
            # Get the selected booking
            booking = bookings[booking_number]

            # Update details
            print("\nEnter new details (Leave blank to keep current):")
            new_name = input("New Name: ")
            new_age = input("New Age: ")
            new_gender = input("New Gender: ")
            new_ticket_number = input("New Ticket Number: ")
```



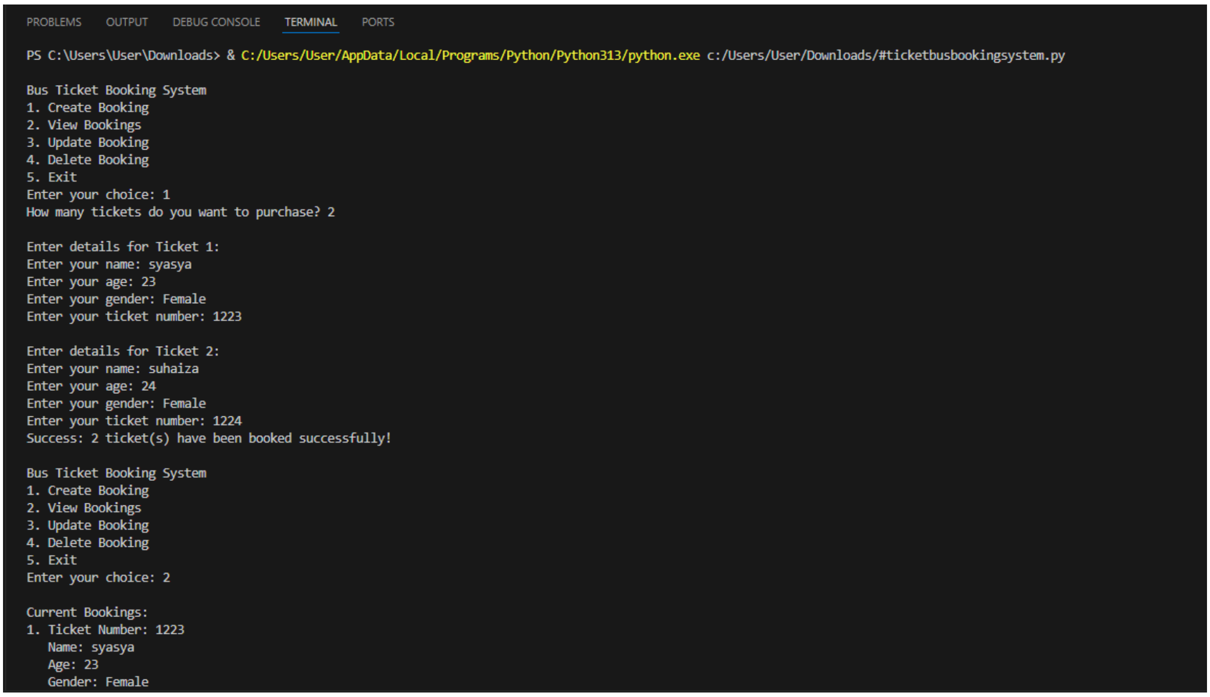
```
            new_ticket_number = input("New Ticket Number: ")

            if new_name:
                booking["name"] = new_name
            if new_age:
                booking["age"] = new_age
            if new_gender:
                booking["gender"] = new_gender
            if new_ticket_number:
                booking["ticket_number"] = new_ticket_number

            print("Success: Booking updated successfully!")
        else:
            print("Error: Invalid booking number!")
    except ValueError:
        print("Error: Please enter a valid number.")

# Function to delete a booking
def delete_booking():
    view_bookings()  # View bookings before deleting

    try:
        booking_number = int(input("Enter Booking Number to Delete: ")) - 1
        if 0 <= booking_number < len(bookings):
            del bookings[booking_number]
            print("Success: Booking deleted successfully!")
        else:
            print("Error: Invalid booking number!")
    except ValueError:
        print("Error: Please enter a valid number.")
```



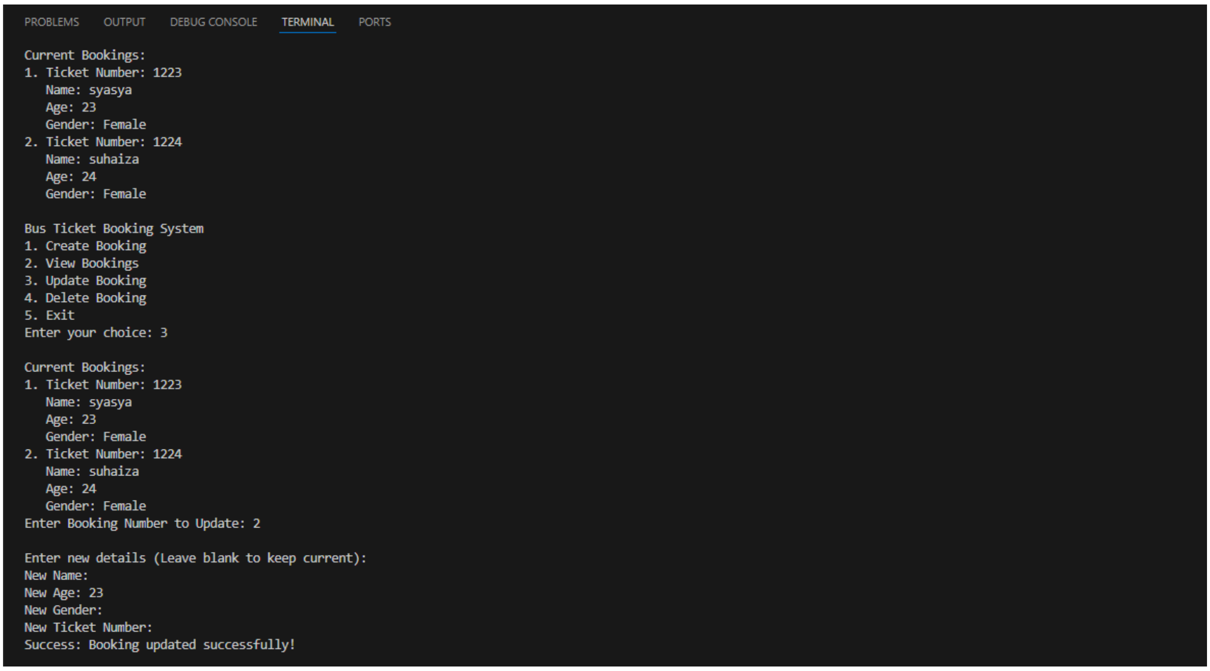
```
        choice = input("Enter your choice: ")

        if choice == "1":
            create_booking()
        elif choice == "2":
            view_bookings()
        elif choice == "3":
            update_booking()
        elif choice == "4":
            delete_booking()
        elif choice == "5":
            print("Exiting the system.")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

GUI : No

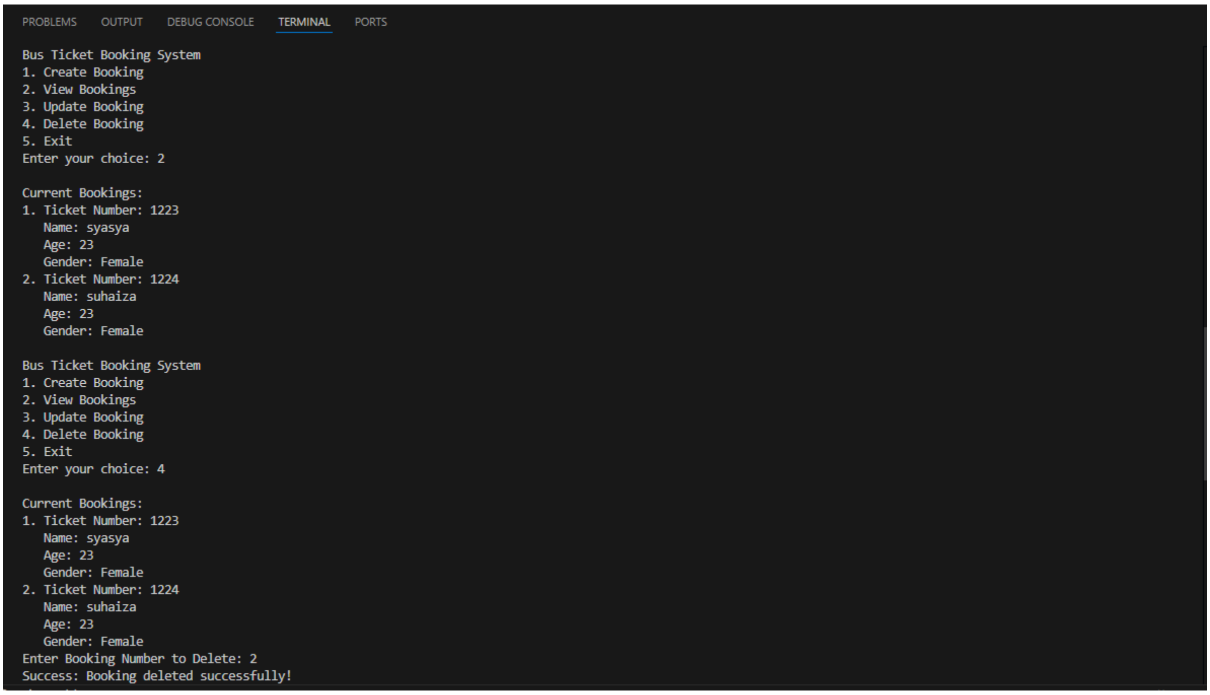
Result :



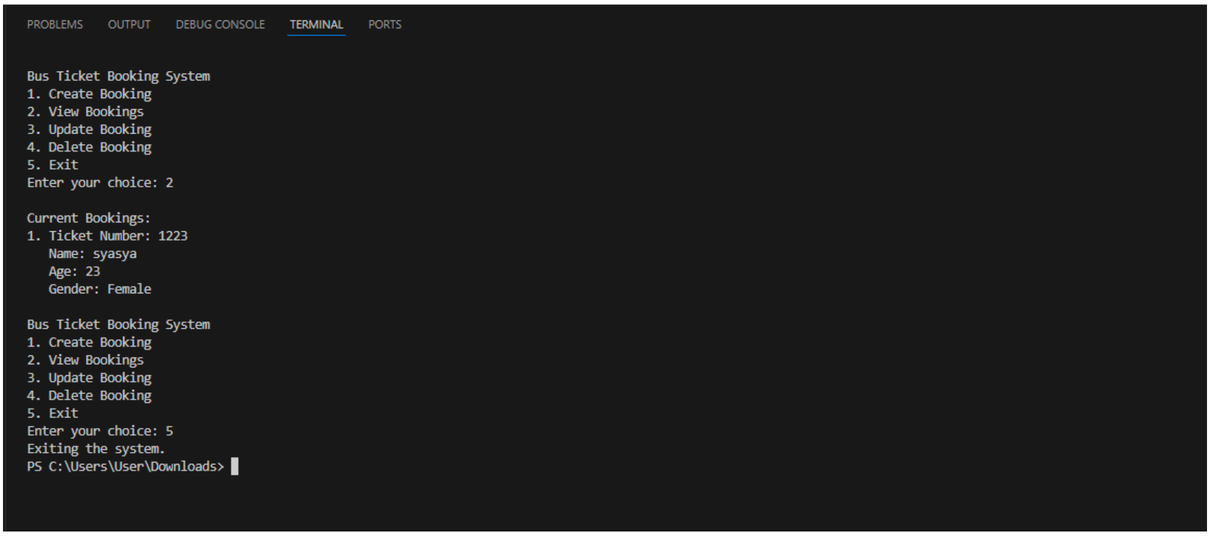
```
PS C:\Users\User\Downloads> & C:/Users/User/AppData/Local/Programs/Python/Python313/python.exe c:/Users/User/Downloads/#ticketbusbookingsystem.py

Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 1
How many tickets do you want to purchase? 2

Enter details for Ticket 1:
Enter your name: syasya
Enter your age: 23
Enter your gender: Female
Enter your ticket number: 1223

Enter details for Ticket 2:
Enter your name: suhaiza
Enter your age: 24
Enter your gender: Female
Enter your ticket number: 1224
Success: 2 ticket(s) have been booked successfully!

Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
```



```
Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 24
   Gender: Female

Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 3

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 24
   Gender: Female
Enter Booking Number to Update: 2

Enter new details (Leave blank to keep current):
New Name:
New Age: 23
New Gender:
New Ticket Number:
Success: Booking updated successfully!
```



```
Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 23
   Gender: Female

Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 4

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female
2. Ticket Number: 1224
   Name: suhaiza
   Age: 23
   Gender: Female
Enter Booking Number to Delete: 2
Success: Booking deleted successfully!
```



```
Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 2

Current Bookings:
1. Ticket Number: 1223
   Name: syasya
   Age: 23
   Gender: Female

Bus Ticket Booking System
1. Create Booking
2. View Bookings
3. Update Booking
4. Delete Booking
5. Exit
Enter your choice: 5
Exiting the system.
PS C:\Users\User\Downloads>
```

Strength:

1. Easy to manage and view records at any time by use the Create, Read, Update, and Delete (CRUD) bookings and making it easier to the users to booking the ticket smoothly.
2. The systems is user friendly interface which making it easy to navigate the program that uses a clean menu-driven approach while the prompts are clear and ensure smooth user input in the systems.
3. The systems handled the errors for invalid input with gracefully with appropriate messages and it also can handle the multiple ticket booked at once with efficient and user-friendly.

Kaizen (Room for improvement) :

1. I will suggest to add the gender, whether male or female on every seat that people have booked to make it more easier for them to choose wherever they want to seat and much more comfortable.
2. I will add the seat number and type of bus to make it easier to the users so that they do not miss the bus and can easily find a seat without getting mixed up.
3. Add the date and time of the ticket bus so that the users can set the time and date that has been provided on the ticket their schedule and can avoid missing the bus.

REFERENCE

Acsany, P. (2024, May 29). *What Are CRUD Operations?* Retrieved from RealPython: https://realpython.com/crud-operations/

Movie tickets Booking management system in Python. (2021, November 12). Retrieved from Geeksforgeeks: https://www.geeksforgeeks.org/movie-tickets-booking-management-system-in-python/

Suarez, a. j. (2020, September 1). *Ticket Booking System In Python With Source Code*. Retrieved from ITSOURCECODE: https://itsourcecode.com/free-projects/python-projects/ticket-booking-system-in-python-with-source-code/#google_vignette