

Name: Dev Sapariya

Roll Number: 21BCP293

Division & Group: Div 5 and G9

Subject: Advance Python Programming

Assignment: 2

Problem Statement

Railway Ticket Reservation System

You are tasked with developing a railway ticket reservation system for a busy rail network. The system should handle ticket booking, seat availability, and generate reports for the railway administration. Your task is to implement a Python program that provides the following functionalities:

Load Train Data: The program should read the train data from a CSV file named "trains.csv." Each row in the CSV file represents a train with the following information:

- Train ID (a unique alphanumeric code)
- Train Name
- Source Station
- Destination Station
- Total Seats (total number of seats available on the train)

Load Passenger Data: The program should read the passenger data from a CSV file named "passengers.csv." Each row in the CSV file represents a passenger with the following information:

- Passenger Name
- Train ID (the ID of the train the passenger wants to book a ticket on)
- Number of Tickets (the number of tickets the passenger wants to book)

Check Seat Availability: Given the train ID and the number of tickets requested by a passenger, the program should check if there are enough seats available on the specified train for booking. If seats are available, the booking should be confirmed, and the total fare for the booking should be calculated as per the fare rules (you can define fare rules based on distance, class, etc.).

Update Seat Availability: After confirming the booking, the program should update the seat availability for the corresponding train.

Generate Reports:

Report 1: The program should generate a report showing the details of all the trains, including their names, source stations, destination stations, and the total number of seats available on each train.

Report 2: The program should generate a report showing the total revenue earned from each train based on the total number of confirmed bookings and their respective fares.

Handle Errors: The program should handle various types of errors gracefully, such as invalid train IDs, invalid passenger names, insufficient seats, etc., and provide appropriate error messages.

Note:

You can assume that the passenger data in "passengers.csv" will not exceed the available seats on any train.

You can design the fare rules based on your preference and mention them clearly in the program.

Write the Python program to implement the above functionalities for the railway ticket reservation system. Use comments to explain each step of your implementation and provide sample CSV files ("trains.csv" and "passengers.csv") for testing the program.

Code

```
import csv

def load_train_data():
    trains = {}
    with open('trains.csv', mode='r') as file:
```

```

        reader = csv.DictReader(file)
        for row in reader:
            train_id = row['Train ID']
            train_name = row['Train Name']
            source_station = row['Source Station']
            destination_station = row['Destination Station']
            total_seats = int(row['Total Seats'])
            fare_per_seat = int(row['fareperseat'])
            booked_seats = int(row['Booked Seats']) if 'Booked Seats' in row
    else 0

    trains[train_id] = {
        'Train Name': train_name,
        'Source Station': source_station,
        'Destination Station': destination_station,
        'Total Seats': total_seats,
        'fare_per_seat': fare_per_seat,
        'Booked Seats': booked_seats
    }
    return trains

def load_passenger_data():
    passengers = []
    with open('passengers.csv', mode='r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            passenger_name = row['Passenger Name']
            train_id = row['Train ID']
            num_tickets = int(row['Number of Tickets'])
            passengers.append({
                'Passenger Name': passenger_name,
                'Train ID': train_id,
                'Number of Tickets': num_tickets
            })
    return passengers

def check_seat_availability(trains, train_id, no_of_passenger):
    if train_id not in trains:
        print(f"Invalid Train ID: {train_id}")
        return False

    available_seats = trains[train_id]['Total Seats'] -
trains[train_id]['Booked Seats']
    if no_of_passenger > available_seats:
        print(f"Insufficient Seats on Train {train_id}. Available Seats:
{available_seats}")
        print("Please try again with a different train or lower number of
passengers")

```

```

        print("-----")
        return False

    return True

def update_seat_availability(trains, train_id, no_of_passenger):
    trains[train_id]['Booked Seats'] += no_of_passenger

def menu():
    print("Welcome to the Railway Reservation System")
    print("1. Book Ticket")
    print("2. Cancel Ticket")
    print("3. Show All Train Details")
    print("4. Exit")
    print("5. Revenue")
    print("6. Get Detailed Summary in text file")
    choice = int(input("Enter your choice: "))
    return choice

def add_new_passenger(passenger_name, train_id, no_of_passenger):
    with open('passengers.csv', mode='a', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=['Passenger Name', 'Train ID', 'Number of Tickets'])
        writer.writerow({
            'Passenger Name': passenger_name,
            'Train ID': train_id,
            'Number of Tickets': no_of_passenger
        })

def book_ticket(trains):
    train_id = input("Enter the Train ID: ")
    train = trains.get(train_id)
    if train is None:
        print(f"Invalid Train ID: {train_id}")
        return

    no_of_passenger = int(input("Enter the number of passengers: "))
    if check_seat_availability(trains, train_id, no_of_passenger):
        print("Seats are available")
        name_of_passenger = input("Enter the name of the passenger: ")
        print("Summary")
        print(f"Train ID: {train_id}")
        print(f"Train Name: {train['Train Name']}")
        print(f"Number of Passengers: {no_of_passenger}")

```

```

print(f"Name of Passenger: {name_of_passenger}")
print(f"Total Fare: {no_of_passenger * train['fare_per_seat']}")
confirm = input("Confirm Booking (y/n): ")
if confirm.lower() == 'y':
    update_seat_availability(trains, train_id, no_of_passenger)
    add_new_passenger(name_of_passenger, train_id, no_of_passenger)
    print("Ticket Booked Successfully")
else:
    print("Ticket Booking Cancelled")

def cancel_ticket(trains):
    print("Cancel Ticket")
    passenger_name = input("Enter the name of the passenger: ")
    train_id = input("Enter the Train ID: ")

    with open('passengers.csv', mode='r') as file:
        reader = csv.DictReader(file)
        passengers = []
        found = False
        for row in reader:
            if row['Passenger Name'] == passenger_name and row['Train ID'] ==
train_id:
                found = True
                num_tickets = int(row['Number of Tickets'])
                continue

            passengers.append(row)

        if not found:
            print("Invalid Passenger Name or Train ID")
            return

    with open('passengers.csv', mode='w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=['Passenger Name', 'Train
ID', 'Number of Tickets'])
        writer.writeheader()
        for passenger in passengers:
            writer.writerow(passenger)
    trains[train_id]['Booked Seats'] -= num_tickets
    print("Ticket Cancelled Successfully")

def show_all_train_details(trains):
    print("Train Details")
    print("-----")
    print("-----")
    for train_id, train in trains.items():
        print(f"Train ID: {train_id}")

```

```

        print(f"Train Name: {train['Train Name']}")
        print(f"Source Station: {train['Source Station']}")
        print(f"Destination Station: {train['Destination Station']}")
        print(f"Total Seats: {train['Total Seats']}")
        print(f"Seats Booked: {train['Booked Seats']}")
        print(f"Fare per Seat: {train['fare_per_seat']}")
        print("-----")
    -----")

def revenue(trains):
    print("Revenue")
    print("-----")
    -----")
    total_revenue = 0
    for train_id, train in trains.items():
        revenue = train['Booked Seats'] * train['fare_per_seat']
        total_revenue += revenue
        print(f"Train ID: {train_id}")
        print(f"Train Name: {train['Train Name']}")
        print(f"Seats Booked: {train['Booked Seats']}")
        print(f"Revenue: {revenue}")
        print("-----")
    -----")
    print(f"Total Revenue: {total_revenue}")

def get_detailed_summary(trains):
    with open("summary.txt", "w") as file:
        file.write("All trains\n")
        for train_id, train in trains.items():
            file.write(f"Train ID: {train_id}\n")
            file.write(f"Train Name: {train['Train Name']}\n")
            file.write(f"Source Station: {train['Source Station']}\n")
            file.write(f"Destination Station: {train['Destination
Station']}\n")
            file.write(f"Total Seats: {train['Total Seats']}\n")
            file.write(f"Seats Booked: {train['Booked Seats']}\n")
            file.write(f"Fare per Seat: {train['fare_per_seat']}\n")
            file.write(f"Total revenue: {train['Booked Seats'] *
train['fare_per_seat']}\n")
            file.write("-----\n")

    print("Detailed summary written to summary.txt")

```

```

def main_menu():
    trains = load_train_data()
    passengers = load_passenger_data()
    while True:
        choice = menu()
        if choice == 1:
            book_ticket(trains)
        elif choice == 2:
            cancel_ticket(trains)
        elif choice == 3:
            show_all_train_details(trains)
        elif choice == 4:
            break
        elif choice == 5:
            revenue(trains)
        elif choice == 6:
            get_detailed_summary(trains)
        else:
            print("Invalid Choice")
            print("-----")
            print("-----")
            get_detailed_summary(trains)
            print("Thank you for using the Railway Reservation System")

main_menu()

```

trains.csv

Train ID,Train Name,Source Station,Destination Station,Total Seats,fareperseat
 123,Duranto,Mumbai,Ahmedabad,200,500
 456,Ashram,Ahmedabad,Jaipur,150,1200
 789,Shatabdi,Kolkata,Dehli,100,1400
 1011,Intercity,Surat,Rajkot,180,200
 1213,Rajdhani,Dehli,Bengluru,250,1500

passengers.csv

Passenger Name,Train ID,Number of Tickets

Output:

```
E:\College\Sem 5\Labs\Advance Python\Lab 2 04-08>python myfile.py
```

```
Welcome to the Railway Reservation System
```

1. Book Ticket
2. Cancel Ticket
3. Show All Train Details
4. Exit
5. Revenue
6. Get Detailed Summary in text file

```
Enter your choice: 3
```

```
Train Details
```

```
-----  
Train ID: 123  
Train Name: Duranto  
Source Station: Mumbai  
Destination Station: Ahmedabad  
Total Seats: 200  
Seats Booked: 0  
Fare per Seat: 500  
-----
```

```
Train ID: 456  
Train Name: Ashram  
Source Station: Ahmedabad  
Destination Station: Jaipur  
Total Seats: 150  
Seats Booked: 0  
Fare per Seat: 1200  
-----
```

```
Train ID: 789  
Train Name: Shatabdi  
Source Station: Kolkata  
Destination Station: Dehli  
Total Seats: 100  
Seats Booked: 0  
-----
```

```
-----  
Train ID: 1011  
Train Name: Intercity  
Source Station: Surat  
Destination Station: Rajkot  
Total Seats: 180  
Seats Booked: 0  
Fare per Seat: 200  
-----
```

```
Train ID: 1213  
Train Name: Rajdhani  
Source Station: Dehli  
Destination Station: Bengluru  
Total Seats: 250  
Seats Booked: 0  
Fare per Seat: 1500  
-----  
-----
```



```
Welcome to the Railway Reservation System
```

1. Book Ticket
2. Cancel Ticket
3. Show All Train Details
4. Exit
5. Revenue
6. Get Detailed Summary in text file

```
Enter your choice: 1
```

```
Enter the Train ID: 123
```

```
Enter the number of passengers: 20
```

```
Seats are available
```

```
Enter the name of the passenger: Dev
```

```
Summary
```

```
Train ID: 123
```

```
Train Name: Duranto
```

```
Number of Passengers: 20
```

```
Name of Passenger: Dev
```

```
Total Fare: 10000
```

```
Confirm Booking (y/n): y
```

```
Ticket Booked Successfully
```

```
-----
```

```
-----  
Welcome to the Railway Reservation System
```

1. Book Ticket
2. Cancel Ticket
3. Show All Train Details
4. Exit
5. Revenue
6. Get Detailed Summary in text file

```
Enter your choice: 2
```

```
Cancel Ticket
```

```
Enter the name of the passenger: Dev
```

```
Enter the Train ID: 123
```

```
Ticket Cancelled Successfully
```

```
-----
```

```
3. Show All Train Details
4. Exit
5. Revenue
6. Get Detailed Summary in text file
Enter your choice: 5
Revenue
```

```
-----
Train ID: 123
Train Name: Duranto
Seats Booked: 0
Revenue: 0
-----
```

```
-----
Train ID: 456
Train Name: Ashram
Seats Booked: 30
Revenue: 36000
-----
```

```
-----
Train ID: 789
Train Name: Shatabdi
Seats Booked: 0
Revenue: 0
-----
```

```
-----
Train ID: 1011
Train Name: Intercity
Seats Booked: 0
Revenue: 0
-----
```

```
-----
Train ID: 1213
Train Name: Rajdhani
Seats Booked: 5
Revenue: 7500
-----
```

```
-----
Total Revenue: 43500
-----
```

```
Welcome to the Railway Reservation System
1. Book Ticket
2. Cancel Ticket
3. Show All Train Details
4. Exit
5. Revenue
6. Get Detailed Summary in text file
Enter your choice: 4
Detailed summary written to summary.txt
Thank you for using the Railway Reservation System
```

passengers.csv

Passenger Name,Train ID,Number of Tickets
Khush,456,30
Ronak,1213,5

summary.txt

All trains
Train ID: 123

Train Name: Duranto
Source Station: Mumbai
Destination Station: Ahmedabad
Total Seats: 200
Seats Booked: 0
Fare per Seat: 500
Total revenue: 0

Train ID: 456
Train Name: Ashram
Source Station: Ahmedabad
Destination Station: Jaipur
Total Seats: 150
Seats Booked: 30
Fare per Seat: 1200
Total revenue: 36000

Train ID: 789
Train Name: Shatabdi
Source Station: Kolkota
Destination Station: Dehli
Total Seats: 100
Seats Booked: 0
Fare per Seat: 1400
Total revenue: 0

Train ID: 1011
Train Name: Intercity
Source Station: Surat
Destination Station: Rajkot
Total Seats: 180
Seats Booked: 0
Fare per Seat: 200
Total revenue: 0

Train ID: 1213
Train Name: Rajdhani
Source Station: Dehli
Destination Station: Bengluru
Total Seats: 250
Seats Booked: 5
Fare per Seat: 1500
Total revenue: 7500
