**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("---------------------------------------------------------------------------------------")

    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("---------------------------------------------------------------------------------------")

    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,Kolkota,Dehli,100,1400

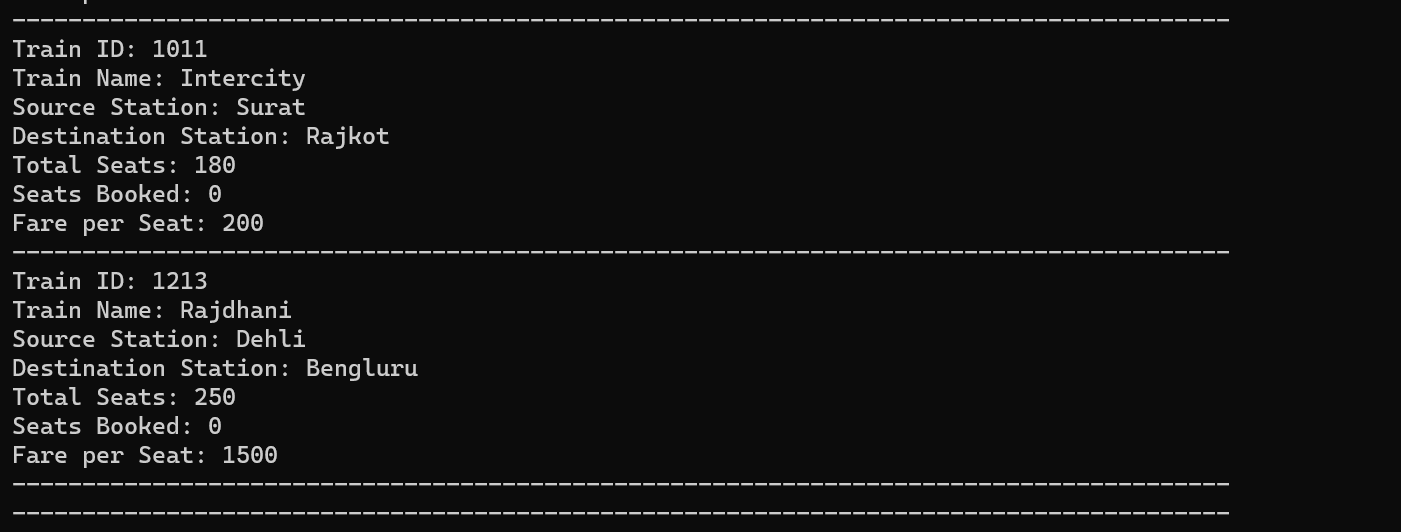
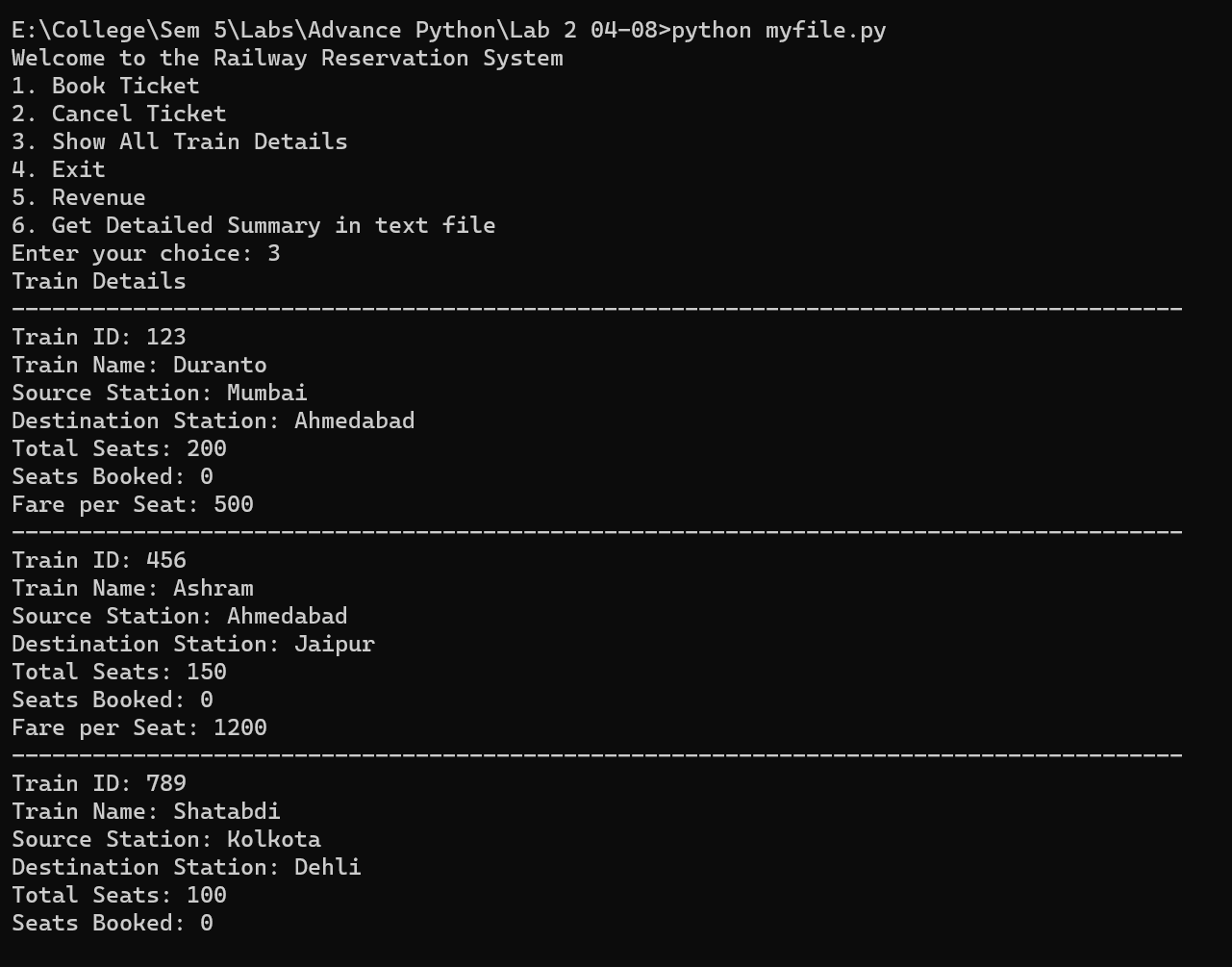
1011,Intercity,Surat,Rajkot,180,200

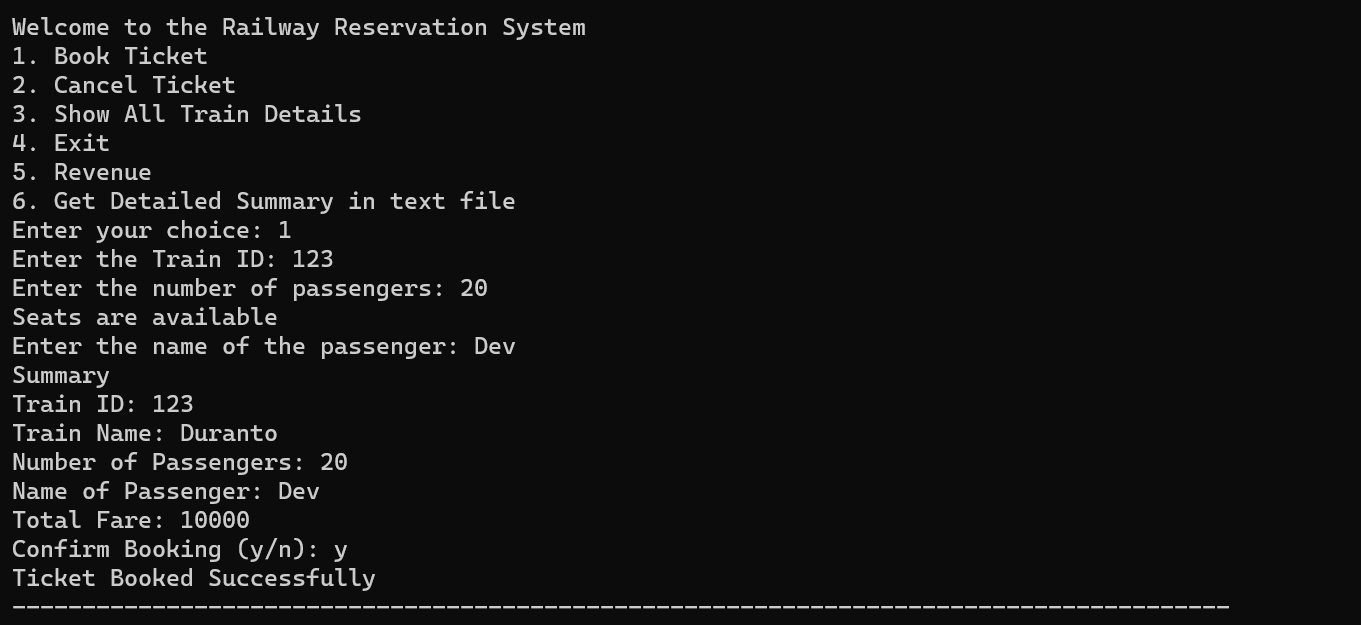
1213,Rajdhani,Dehli,Bengluru,250,1500

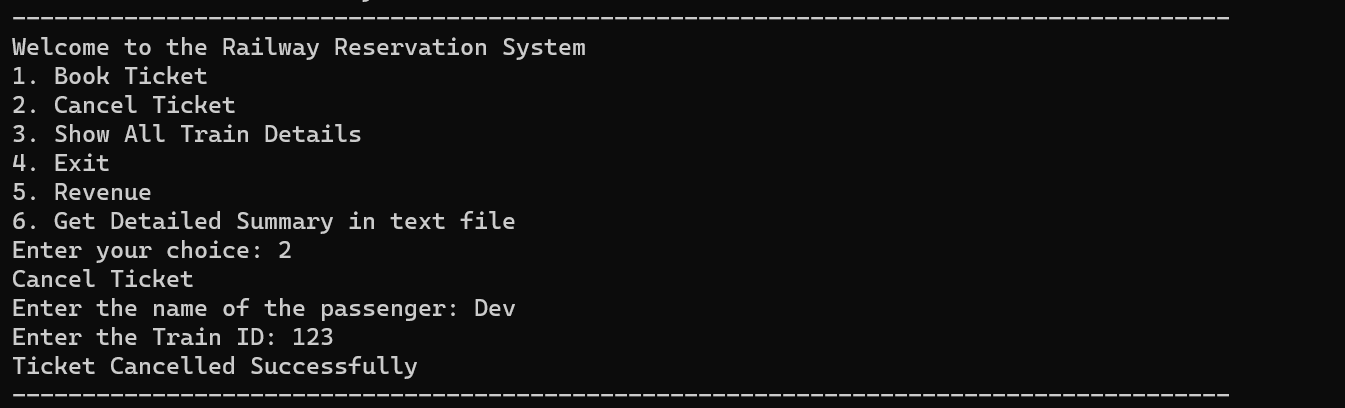
**passengers.csv**

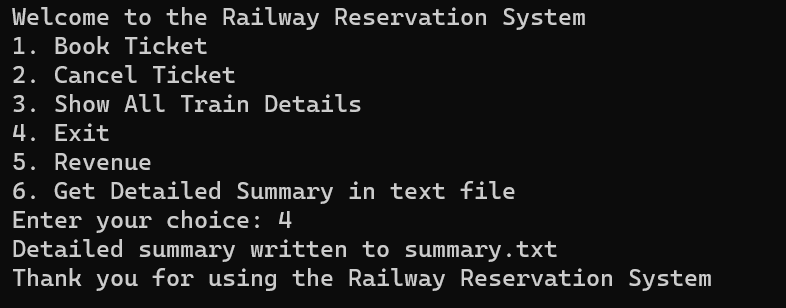
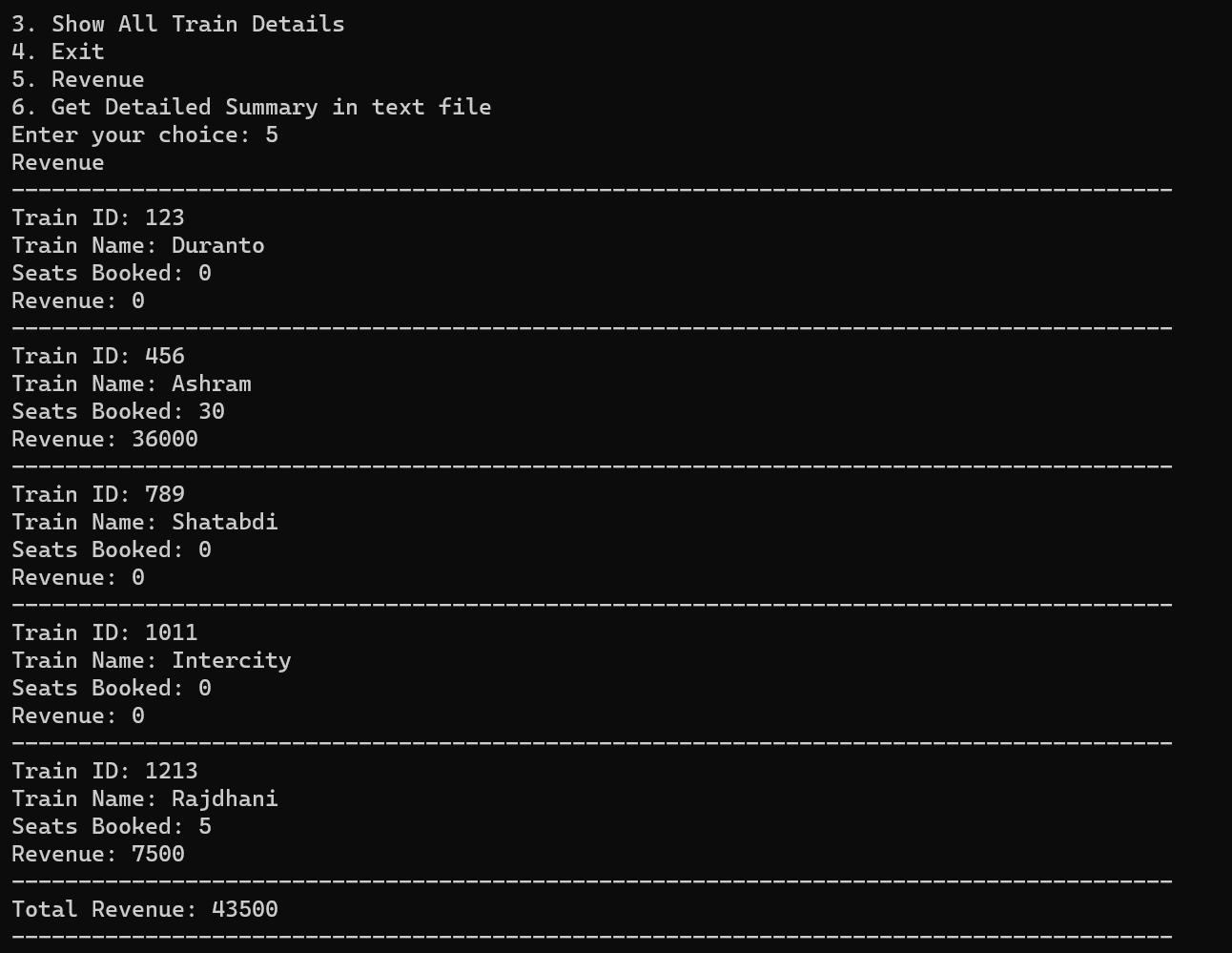
Passenger Name,Train ID,Number of Tickets

**Output:**

****

****

****



**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

--------------------------------------------------