PIMPRI CHINCHWAD EDUCATION TRUST's.
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
(An Autonomous Institute)

Class : SY BTech	**Acad. Yr. 2025-26**	**Semester : I**
Name of the student: Hariom Shrikrishna Gundale		**PRN : 124B1B036**
Department: Computer Engineering		**Division : A**
Course Name : Data Structures Laboratory		**Course Code:BCE23PC02**
Completion Date : 12/10/2025		

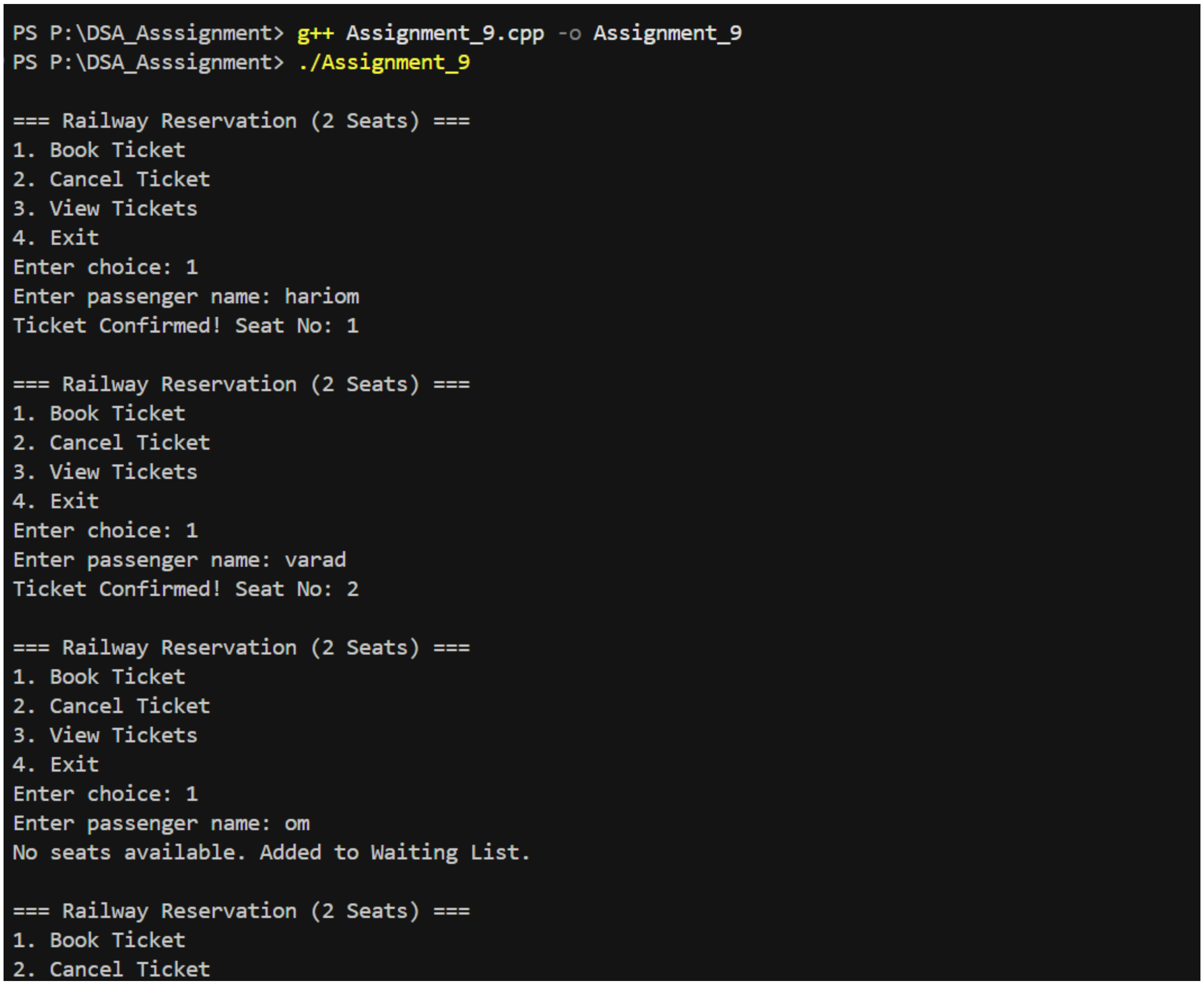
Assignment No. 9

Problem Statement: Design a simplified railway reservation system where users can book, cancel, and view tickets. Use an array to store booking details and a queue to manage the waiting list.

Source Code :

```
#include <iostream>
#include <string>
#include <limits>
using namespace std;

// Node class for linked list queue (Waiting List)
class Node {
public:
    string name;
    Node* next;
    Node(string n) : name(n), next(nullptr) {}
};

class Queue {
private:
    Node *front, *rear;
public:
    Queue() : front(nullptr), rear(nullptr) {}
    bool isEmpty() { return front == nullptr; }

    void enqueue(string name) {
        Node* temp = new Node(name);
        if (rear == nullptr) front = rear = temp;
        else { rear->next = temp; rear = temp; }
    }

    string dequeue() {
        if (isEmpty()) return "";
        Node* temp = front;
        string name = front->name;
        front = front->next;
        delete temp;
        if (front == nullptr) rear = nullptr;
        return name;
    }

    void display() {
        if (isEmpty()) cout << "Waiting List is empty.\n";
        else {
            cout << "Waiting List:\n";
            Node* temp = front;
            while (temp) { cout << temp->name << endl; temp = temp->next; }
        }
    }
};

class ReservationSystem {
private:
    string confirmed[2];  // ✅ Only 2 seats
    int seatCount;
    Queue waitingList;

public:
    ReservationSystem() : seatCount(0) {}

    void bookTicket(string name) {
        if (seatCount < 2) {
            confirmed[seatCount++] = name;
            cout << "Ticket Confirmed! Seat No: " << seatCount << endl;
        } else {
            waitingList.enqueue(name);
            cout << "No seats available. Added to Waiting List.\n";
        }
    }

    void cancelTicket(int seatNo) {
        if (seatNo < 1 || seatNo > seatCount) {
            cout << "Invalid seat number!\n";
            return;
        }
        cout << "Ticket canceled for " << confirmed[seatNo - 1] << endl;
        for (int i = seatNo - 1; i < seatCount - 1; i++) {
            confirmed[i] = confirmed[i + 1];
        }
        seatCount--;

        if (!waitingList.isEmpty()) {
            string nextPassenger = waitingList.dequeue();
            confirmed[seatCount++] = nextPassenger;
            cout << "Seat assigned to waiting list passenger: " << nextPassenger << endl;
        }
    }

    void viewTickets() {
        if (seatCount == 0) cout << "No confirmed bookings.\n";
        else {
            cout << "Confirmed Bookings:\n";
            for (int i = 0; i < seatCount; i++)
                cout << "Seat " << i + 1 << ": " << confirmed[i] << endl;
        }
        waitingList.display();
    }
};

int safeInput() {
    int x;
    while (!(cin >> x)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Invalid input! Enter a number: ";
    }
    return x;
}

int main() {
    ReservationSystem rs;
    int choice, seatNo;
    string name;

    while (true) {
        cout << "\n=== Railway Reservation (2 Seats) ===\n";
        cout << "1. Book Ticket\n2. Cancel Ticket\n3. View Tickets\n4. Exit\nEnter choice: ";
        choice = safeInput();

        switch (choice) {
            case 1:
                cout << "Enter passenger name: ";
                cin >> name;
                rs.bookTicket(name);
                break;
            case 2:
                cout << "Enter seat number to cancel: ";
                seatNo = safeInput();
                rs.cancelTicket(seatNo);
                break;
            case 3:
                rs.viewTickets();
                break;
            case 4:
                cout << "Exiting...\n";
                return 0;
            default:
                cout << "Invalid Choice!\n";
        }
    }
}
```

Screen Shot of Output :



```
PS P:\DSA_Asssignment> g++ Assignment_9.cpp -o Assignment_9
PS P:\DSA_Asssignment> ./Assignment_9

=== Railway Reservation (2 Seats) ===
1. Book Ticket
2. Cancel Ticket
3. View Tickets
4. Exit
Enter choice: 1
Enter passenger name: hariom
Ticket Confirmed! Seat No: 1

=== Railway Reservation (2 Seats) ===
1. Book Ticket
2. Cancel Ticket
3. View Tickets
4. Exit
Enter choice: 1
Enter passenger name: varad
Ticket Confirmed! Seat No: 2

=== Railway Reservation (2 Seats) ===
1. Book Ticket
2. Cancel Ticket
3. View Tickets
4. Exit
Enter choice: 1
Enter passenger name: om
No seats available. Added to Waiting List.

=== Railway Reservation (2 Seats) ===
1. Book Ticket
2. Cancel Ticket
```

Conclusion:

Thus, we have successfully implemented the C++ program to design a simplified railway reservation system where users can book, cancel, and view tickets. Use an array to store booking details and a queue to manage the waiting list.