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: Varad Amol Pisale **PRN : 124B1B043**

Department: Computer Engineering **Division : A**

Course Name : Data Structures Laboratory **Code:BCE23PC02**

Completion Date : 15/10/2025

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 <bits/stdc++.h>
using namespace std;

class Queue
{
   int ID;
   string name;
   Queue *next = nullptr;

public:
   Queue() {}
   Queue(string n, int id)
   {
      name = n;
      ID = id;
   }
   friend class Ticket_system;
};

class Ticket_system
{
   static const int MAX = 5; // total confirmed seats
   string booked[MAX];
```

```
    int bookedCount = 0;

    int identy = 1;
    Queue *front = nullptr, *rear = nullptr;

public:
    void add_ticket(string name)
    {
        if (bookedCount < MAX)
        {
            booked[bookedCount++] = name;
            cout << "Ticket confirmed for " << name << " (ID " << identy++ << ")\n";
        }
        else
        {
            Queue *temp = new Queue(name, identy++);
            if (rear == nullptr)
            {
                front = rear = temp;
            }
            else
            {
                rear->next = temp;
                rear = temp;
            }
            cout << "All seats full! Added to waiting list: " << name << endl;
        }
    }

    void remove_ticket()
    {
        if (bookedCount == 0)
        {
            cout << "No confirmed bookings to cancel!\n";
            return;
        }

        cout << "Cancelled ticket of " << booked[0] << endl;

        // shift all bookings left
        for (int i = 1; i < bookedCount; i++)
            booked[i - 1] = booked[i];
        bookedCount--;

        // if waiting queue not empty, move front person to booked list
        if (front != nullptr)
```

```
        {
            booked[bookedCount++] = front->name;
            cout << "Moved from waiting list to confirmed: " << front->name << endl;

            Queue *temp = front;
            front = front->next;
            delete temp;
            if (front == nullptr)
                rear = nullptr;
        }
    }

    void print_all()
    {
        cout << "\n--- Confirmed Tickets ---\n";
        if (bookedCount == 0)
            cout << "None\n";
        else
            for (int i = 0; i < bookedCount; i++)
                cout << i + 1 << ". " << booked[i] << endl;

        cout << "\n--- Waiting List ---\n";
        if (front == nullptr)
            cout << "None\n";
        else
        {
            Queue *temp = front;
            while (temp != nullptr)
            {
                cout << temp->ID << ". " << temp->name << endl;
                temp = temp->next;
            }
        }
        cout << endl;
    }
};

int main()
{
    Ticket_system obj;
    cout << "=== Railway Reservation System ===\n";
    cout << "1. Book Ticket\n";
    cout << "2. Cancel Ticket\n";
    cout << "3. View All Tickets\n";
    cout << "4. Exit\n";
```

```
    while (true)
    {
        int op;
        cout << "\nEnter Option: ";
        cin >> op;
        cin.ignore();

        switch (op)
        {
        case 1:
        {
            string name;
            cout << "Enter Passenger Name: ";
            getline(cin, name);
            obj.add_ticket(name);
            break;
        }
        case 2:
            obj.remove_ticket();
            break;
        case 3:
            obj.print_all();
            break;
        case 4:
            return 0;
        default:
            cout << "Invalid option!\n";
            break;
        }
    }
}
```

Screen Shot of Output :

```
=== Railway Reservation System ===
1. Book Ticket
2. Cancel Ticket
3. View All Tickets
4. Exit

Enter Option: 1
Enter Passenger Name: Varad
Ticket confirmed for Varad (ID 1)

Enter Option: 1
Enter Passenger Name: OM
Ticket confirmed for OM (ID 2)

Enter Option: 1
Enter Passenger Name: Hariom
Ticket confirmed for Hariom (ID 3)

Enter Option: 1
Enter Passenger Name: Shubhang
Ticket confirmed for Shubhang (ID 4)

Enter Option: 1
Enter Passenger Name: Aniruddha
Ticket confirmed for Aniruddha (ID 5)

Enter Option: 1
Enter Passenger Name: Kamlesh
All seats full! Added to waiting list: Kamlesh
```

```
Enter Option: 3

--- Confirmed Tickets ---
1. Varad
2. OM
3. Hariom
4. Shubhang
5. Aniruddha

--- Waiting List ---
6. Kamlesh


Enter Option: 2
Cancelled ticket of Varad
Moved from waiting list to confirmed: Kamlesh

Enter Option: 3

--- Confirmed Tickets ---
1. OM
2. Hariom
3. Shubhang
4. Aniruddha
5. Kamlesh

--- Waiting List ---
None


Enter Option: 4
```

Conclusion: Hence we have implemented a Railway Reservation system using array and queue.