**Project Title:**

**Railway Reservation System using Java**

**Objective:**

To develop a basic railway reservation system in Java, which allows users to view available trains, check seat availability, book tickets, cancel bookings, and display booked tickets. The system will manage reservations and offer simple functionality to help users interact with a train reservation platform.

**Features:**

1. **Display Available Trains**:
   Users can see a list of available trains, including train names, departure times, passenger capacity, and train numbers.
2. **Check Seat Availability**:
   Users can input a train number to check how many seats are available for booking in that particular train.
3. **Book Tickets**:
   Users can book a ticket by providing their name and selecting the train they wish to travel on, as long as seats are available.
4. **Cancel Tickets**:
   Users can cancel their ticket by providing their name. The system will remove their name from the list of booked passengers.
5. **Display Booked Tickets**:
   The system allows users to view the list of all passengers who have booked seats.
6. **Exit**:
   The system allows users to exit the program when finished.

**Implementation:**

- The project will be implemented in **Java** using classes and objects.
- It will utilize a simple **console-based interface** to interact with users.
- **ArrayLists** will be used to manage the available trains and booked tickets.

**Key Classes:**

1. **Train Class**:
   Stores information about each train, such as train name, departure time, passenger capacity, and train number.
2. **ReservationSystem Class**:
   Handles the main operations like booking tickets, checking availability, and cancelling bookings.

3. **Main Class**:
   Provides a user-friendly interface to interact with the system, offering options through a menu-driven approach.

**Tools & Technologies:**

- **Java** as the programming language.
- **ArrayList** for storing train and booking information.
- **Scanner** for user input.

**Conclusion:**

This railway reservation system will provide users with essential features to manage train bookings efficiently. It is an ideal system for demonstrating basic object-oriented programming concepts and user interaction in Java.

# Source code

```
import java.util.ArrayList;

import java.util.Scanner;



class Train {

    String name;

    String time;

    int passengerStrength;

    int trainNumber;



    public Train(String name, String time, int passengerStrength, int trainNumber) {

        this.name = name;

        this.time = time;

        this.passengerStrength = passengerStrength;

        this.trainNumber = trainNumber;
```

```
    }
}

class ReservationSystem {
    private ArrayList<Train> availableTrains = new ArrayList<>();
    private ArrayList<String> bookedSeats = new ArrayList<>();

    public ReservationSystem() {
        availableTrains.add(new Train("Mumbai - Delhi", "13:05", 50, 1010));
        availableTrains.add(new Train("Delhi - Jaipur", "7:00", 50, 2013));
        availableTrains.add(new Train("Prayagraj - Delhi", "10:00", 50, 3045));
    }

    public void displayAvailableTrains() {
        System.out.println("Available Trains:");
        System.out.println("Train Name\tTime\tPassenger Strength\tTrain Number");
        for (Train train : availableTrains) {
            System.out.println(train.name + "\t" + train.time + "\t" + train.passengerStrength + "\t" +
train.trainNumber);
        }
    }

    public void checkSeatAvailability(int trainNumber) {
        for (Train train : availableTrains) {
```

```
            if (train.trainNumber == trainNumber) {
                int availableSeats = train.passengerStrength - bookedSeats.size();
                System.out.println("Available seats on Train " + train.trainNumber + ": " +
availableSeats);
                return;
            }
        }
        System.out.println("Train not found.");
    }

    public void bookTicket(int trainNumber, String passengerName) {
        for (Train train : availableTrains) {
            if (train.trainNumber == trainNumber) {
                if (bookedSeats.size() < train.passengerStrength) {
                    bookedSeats.add(passengerName);
                    System.out.println("Ticket booked successfully for " + passengerName);
                } else {
                    System.out.println("Sorry, the train is fully booked.");
                }
                return;
            }
        }
        System.out.println("Train not found.");
    }
```

```
    public void cancelTicket(String passengerName) {
        if (bookedSeats.remove(passengerName)) {
            System.out.println("Ticket canceled successfully for " + passengerName);
        } else {
            System.out.println("Passenger not found or no booking exists for this passenger.");
        }
    }

    public void displayBookedTickets() {
        System.out.println("Booked Tickets:");
        for (String passenger : bookedSeats) {
            System.out.println(passenger);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        ReservationSystem reservationSystem = new ReservationSystem();
        Scanner scanner = new Scanner(System.in);

        while (true) {
```

```
System.out.println("\nRailway Reservation System Menu:");
System.out.println("1. Display Available Trains");
System.out.println("2. Check Seat Availability");
System.out.println("3. Book a Ticket");
System.out.println("4. Cancel a Ticket");
System.out.println("5. Display Booked Tickets");
System.out.println("6. Exit");
System.out.print("Enter your choice: ");
int choice = scanner.nextInt();
scanner.nextLine();  // Consume the newline character

switch (choice) {
    case 1:
        reservationSystem.displayAvailableTrains();
        break;
    case 2:
        System.out.print("Enter Train Number: ");
        int trainNumber = scanner.nextInt();
        reservationSystem.checkSeatAvailability(trainNumber);
        break;
    case 3:
        System.out.print("Enter Train Number: ");
        trainNumber = scanner.nextInt();
```

```
                scanner.nextLine();  // Consume the newline character
                System.out.print("Enter Passenger Name: ");
                String passengerName = scanner.nextLine();
                reservationSystem.bookTicket(trainNumber, passengerName);
                break;
            case 4:
                System.out.print("Enter Passenger Name to Cancel: ");
                passengerName = scanner.nextLine();
                reservationSystem.cancelTicket(passengerName);
                break;
            case 5:
                reservationSystem.displayBookedTickets();
                break;
            case 6:
                System.out.println("Exiting Railway Reservation System. Thank you!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
}
}
```

# Outlook

```java
import java.util.ArrayList;
import java.util.Scanner;

class Train {
    String name;
    String time;
    int passengerStrength;
    int trainNumber;

    public Train(String name, String time, int passengerStrength, int trainNumber) {
        this.name = name;
        this.time = time;
        this.passengerStrength = passengerStrength;
        this.trainNumber = trainNumber;
    }
}

class ReservationSystem {
    private ArrayList<Train> availableTrains = new ArrayList<>();
    private ArrayList<String> bookedSeats = new ArrayList<>();

    public ReservationSystem() {
        availableTrains.add(new Train("Mumbai - Delhi", "13:05", 50, 1010));
        availableTrains.add(new Train("Delhi - Jaipur", "7:00", 50, 2013));
        availableTrains.add(new Train("Prayagraj - Delhi", "10:00", 50, 3045));
    }

    public void displayAvailableTrains() {
        System.out.println("Available Trains:");
        System.out.println("Train Name\tTime\tPassenger Strength\tTrain Number");
        for (Train train : availableTrains) {
            System.out.println(train.name + "\t" + train.time + "\t" + train.passengerStrength + "\t" + train.trainNumber);
        }
    }

    public void checkSeatAvailability(int trainNumber) {
        for (Train train : availableTrains) {
            if (train.trainNumber == trainNumber) {
                int availableSeats = train.passengerStrength - bookedSeats.size();
                System.out.println("Available seats on Train " + train.trainNumber + ": " + availableSeats);
                return;
            }
        }
        System.out.println("Train not found.");
    }

    public void bookTicket(int trainNumber, String passengerName) {
        for (Train train : availableTrains) {
            if (train.trainNumber == trainNumber) {
                if (bookedSeats.size() < train.passengerStrength) {
                    bookedSeats.add(passengerName);
                    System.out.println("Ticket booked successfully for " + passengerName);
```

```
java -cp /tmp/mufgKpil98/Main

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 1
Available Trains:
Train Name  Time    Passenger Strength  Train Number
Mumbai - Delhi  13:05   50  1010
Delhi - Jaipur  7:00    50  2013
Prayagraj - Delhi   10:00   50  3045

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
Mumbai - Delhi  13:05   50  1010
Delhi - Jaipur  7:00    50  2013
Prayagraj - Delhi   10:00   50  3045

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 2
Enter Train Number: 1010
Available seats on Train 1010: 50

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
    public void bookTicket(int trainNumber, String
        passengerName) {
        for (Train train : availableTrains) {
            if (train.trainNumber == trainNumber) {
                if (bookedSeats.size() < train
                    .passengerStrength) {
                    bookedSeats.add(passengerName);
                    System.out.println("Ticket booked
                        successfully for " + passengerName);
                } else {
                    System.out.println("Sorry, the train is
                        fully booked.");
                }
                return;
            }
        }
        System.out.println("Train not found.");
    }

    public void cancelTicket(String passengerName) {
        if (bookedSeats.remove(passengerName)) {
            System.out.println("Ticket canceled successfully
                for " + passengerName);
```

```
Enter Train Number: 1010
Available seats on Train 1010: 50

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 3
Enter Train Number: 1010
Enter Passenger Name: biva
Ticket booked successfully for biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
    public void cancelTicket(String passengerName) {
        if (bookedSeats.remove(passengerName)) {
            System.out.println("Ticket canceled successfully
                for " + passengerName);
        } else {
            System.out.println("Passenger not found or no
                booking exists for this passenger.");
        }
    }

    public void displayBookedTickets() {
        System.out.println("Booked Tickets:");
        for (String passenger : bookedSeats) {
            System.out.println(passenger);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        ReservationSystem reservationSystem = new
            ReservationSystem();
        Scanner scanner = new Scanner(System.in);
```

```
Enter Train Number: 1010
Enter Passenger Name: biva
Ticket booked successfully for biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 4
Enter Passenger Name to Cancel: biva
Ticket canceled successfully for biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
    }

    public void displayBookedTickets() {
        System.out.println("Booked Tickets:");
        for (String passenger : bookedSeats) {
            System.out.println(passenger);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        ReservationSystem reservationSystem = new
            ReservationSystem();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nRailway Reservation System
                Menu:");
            System.out.println("1. Display Available Trains"
                );
            System.out.println("2. Check Seat Availability"
                );
```

```
Enter Train Number: 1010
Enter Passenger Name: biva
Ticket booked successfully for biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 5
Booked Tickets:
biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice:
```

```
 99                 case 2:
100                     System.out.print("Enter Train Number: "
                            );
101                     int trainNumber = scanner.nextInt();
102                     reservationSystem.checkSeatAvailability
                            (trainNumber);
103                     break;
104                 case 3:
105                     System.out.print("Enter Train Number: "
                            );
106                     trainNumber = scanner.nextInt();
107                     scanner.nextLine();  // Consume the
                            newline character
108                     System.out.print("Enter Passenger Name:
                            ");
109                     String passengerName = scanner.nextLine
                            ();
110                     reservationSystem.bookTicket(trainNumber
                            , passengerName);
111                     break;
112                 case 4:
113                     System.out.print("Enter Passenger Name
                            to Cancel: ");
```

```
Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 5
Booked Tickets:
biva

Railway Reservation System Menu:
1. Display Available Trains
2. Check Seat Availability
3. Book a Ticket
4. Cancel a Ticket
5. Display Booked Tickets
6. Exit
Enter your choice: 6
Exiting Railway Reservation System. Thank you!

=== Code Execution Successful ===
```