**RailwayManagementSystem**

**Aim:**

**The aim of a Railway Management System in Java is to simplify train ticket booking and manage schedules, allowing users to book tickets easily and view booking history, while helping administrators manage train resources effectively.**

**Coding:**

**import javax.swing.\*;**

**import javax.swing.table.DefaultTableModel;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.net.URL;**

**import java.util.ArrayList;**

**// Train class to store train details**

**class Train {**

**int trainNumber;**

**String name;**

**String source;**

**String destination;**

**String startTime;**

**String reachTime;**

**int totalSeats;**

**int balance;**

**public Train(int trainNumber, String name, String source, String destination, String startTime, String reachTime, int totalSeats, int balance) {**

**this.trainNumber = trainNumber;**

**this.name = name;**

**this.source = source;**

**this.destination = destination;**

**this.startTime = startTime;**

**this.reachTime = reachTime;**

**this.totalSeats = totalSeats;**

**this.balance = balance;**

**}**

**@Override**

**public String toString() {**

**return "Train No: " + trainNumber + ", Name: " + name + ", Source: " + source + " (Start: " + startTime + "), Destination: " + destination + " (Reach: " + reachTime + ")";**

**}**

**}**

**// Custom JPanel for background image**

**class BackgroundPanel extends JPanel {**

**private Image backgroundImage;**

**public BackgroundPanel(String imagePath) {**

**URL imageURL = getClass().getClassLoader().getResource(imagePath);**

**if (imageURL != null) {**

**backgroundImage = new ImageIcon(imageURL).getImage();**

**} else {**

**System.out.println("Background image not found at: " + imagePath);**

**}**

**}**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**if (backgroundImage != null) {**

**g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);**

**}**

**}**

**}**

**// Train Schedule Frame with search by source and destination**

**class TrainScheduleFrame extends JFrame {**

**private ArrayList<Train> trainList;**

**private JTable table;**

**private DefaultTableModel tableModel;**

**private RailwayManagementSystem system;**

**public TrainScheduleFrame(RailwayManagementSystem system, ArrayList<Train> trains) {**

**this.system = system;**

**this.trainList = trains;**

**setTitle("Train Schedule");**

**setExtendedState(JFrame.MAXIMIZED\_BOTH);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**JPanel searchPanel = new JPanel();**

**searchPanel.setLayout(new FlowLayout());**

**JLabel sourceLabel = new JLabel("Enter Source:");**

**JTextField sourceField = new JTextField(10);**

**JLabel destinationLabel = new JLabel("Enter Destination:");**

**JTextField destinationField = new JTextField(10);**

**JButton searchButton = new JButton("Search");**

**searchButton.setPreferredSize(new Dimension(100, 30));**

**searchPanel.add(sourceLabel);**

**searchPanel.add(sourceField);**

**searchPanel.add(destinationLabel);**

**searchPanel.add(destinationField);**

**searchPanel.add(searchButton);**

**String[] columnNames = {"Train Number", "Name", "Source", "Destination", "Start Time", "Reach Time", "Total Seats", "Balance"};**

**tableModel = new DefaultTableModel(columnNames, 0);**

**table = new JTable(tableModel);**

**JScrollPane scrollPane = new JScrollPane(table);**

**searchButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**String sourceText = sourceField.getText().trim();**

**String destinationText = destinationField.getText().trim();**

**searchTrainSchedule(sourceText, destinationText);**

**}**

**});**

**JButton backButton = new JButton("Back");**

**backButton.setPreferredSize(new Dimension(100, 30));**

**backButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**dispose();**

**system.showMainFrame();**

**}**

**});**

**setLayout(new BorderLayout());**

**add(searchPanel, BorderLayout.NORTH);**

**add(scrollPane, BorderLayout.CENTER);**

**add(backButton, BorderLayout.SOUTH);**

**populateTrainTable();**

**setVisible(true);**

**}**

**private void populateTrainTable() {**

**tableModel.setRowCount(0);**

**for (Train train : trainList) {**

**tableModel.addRow(new Object[]{**

**train.trainNumber,**

**train.name,**

**train.source,**

**train.destination,**

**train.startTime,**

**train.reachTime,**

**train.totalSeats,**

**train.balance**

**});**

**}**

**}**

**private void searchTrainSchedule(String sourceText, String destinationText) {**

**tableModel.setRowCount(0);**

**if (sourceText.isEmpty() && destinationText.isEmpty()) {**

**populateTrainTable();**

**return;**

**}**

**for (Train train : trainList) {**

**if ((sourceText.isEmpty() || train.source.toLowerCase().contains(sourceText.toLowerCase())) &&**

**(destinationText.isEmpty() || train.destination.toLowerCase().contains(destinationText.toLowerCase()))) {**

**tableModel.addRow(new Object[]{**

**train.trainNumber,**

**train.name,**

**train.source,**

**train.destination,**

**train.startTime,**

**train.reachTime,**

**train.totalSeats,**

**train.balance**

**});**

**}**

**}**

**}**

**}**

**// Admin Panel with Insert and Back button added**

**class AdminPanel extends JFrame {**

**private ArrayList<Train> trainList;**

**private DefaultTableModel tableModel;**

**private RailwayManagementSystem system;**

**public AdminPanel(RailwayManagementSystem system, ArrayList<Train> trains) {**

**this.system = system;**

**this.trainList = trains;**

**setTitle("Admin Panel");**

**setExtendedState(JFrame.MAXIMIZED\_BOTH);  // Fullscreen**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**String[] columnNames = {"Train Number", "Name", "Source", "Destination", "Start Time", "Reach Time", "Total Seats", "Balance"};**

**tableModel = new DefaultTableModel(columnNames, 0);**

**JTable table = new JTable(tableModel);**

**JScrollPane scrollPane = new JScrollPane(table);**

**// Populate the table with train data**

**populateTrainTable();**

**// Delete Train Button**

**JButton deleteButton = new JButton("Delete Train");**

**deleteButton.setPreferredSize(new Dimension(150, 30));**

**deleteButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**int selectedRow = table.getSelectedRow();**

**if (selectedRow != -1) {**

**int trainNumber = (int) tableModel.getValueAt(selectedRow, 0);**

**trainList.removeIf(train -> train.trainNumber == trainNumber);**

**tableModel.removeRow(selectedRow);**

**JOptionPane.showMessageDialog(null, "Train deleted successfully!");**

**} else {**

**JOptionPane.showMessageDialog(null, "Please select a train to delete.");**

**}**

**}**

**});**

**// Insert Train Button**

**JButton insertButton = new JButton("Insert Train");**

**insertButton.setPreferredSize(new Dimension(150, 30));**

**insertButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**String trainNumberStr = JOptionPane.showInputDialog("Enter Train Number:");**

**String name = JOptionPane.showInputDialog("Enter Train Name:");**

**String source = JOptionPane.showInputDialog("Enter Source:");**

**String destination = JOptionPane.showInputDialog("Enter Destination:");**

**String startTime = JOptionPane.showInputDialog("Enter Start Time:");**

**String reachTime = JOptionPane.showInputDialog("Enter Reach Time:");**

**String totalSeatsStr = JOptionPane.showInputDialog("Enter Total Seats:");**

**String balanceStr = JOptionPane.showInputDialog("Enter Balance:");**

**int trainNumber = Integer.parseInt(trainNumberStr);**

**int totalSeats = Integer.parseInt(totalSeatsStr);**

**int balance = Integer.parseInt(balanceStr);**

**Train newTrain = new Train(trainNumber, name, source, destination, startTime, reachTime, totalSeats, balance);**

**trainList.add(newTrain);**

**tableModel.addRow(new Object[]{trainNumber, name, source, destination, startTime, reachTime, totalSeats, balance});**

**JOptionPane.showMessageDialog(null, "Train inserted successfully!");**

**}**

**});**

**// Back Button**

**JButton backButton = new JButton("Back");**

**backButton.setPreferredSize(new Dimension(150, 30));**

**backButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**dispose();**

**system.showMainFrame();**

**}**

**});**

**JPanel buttonPanel = new JPanel(new FlowLayout());**

**buttonPanel.add(deleteButton);**

**buttonPanel.add(insertButton);**

**buttonPanel.add(backButton);**

**// Layout for admin panel**

**setLayout(new BorderLayout());**

**add(scrollPane, BorderLayout.CENTER);**

**add(buttonPanel, BorderLayout.SOUTH);**

**setVisible(true);**

**}**

**private void populateTrainTable() {**

**tableModel.setRowCount(0);**

**for (Train train : trainList) {**

**tableModel.addRow(new Object[]{**

**train.trainNumber,**

**train.name,**

**train.source,**

**train.destination,**

**train.startTime,**

**train.reachTime,**

**train.totalSeats,**

**train.balance**

**});**

**}**

**}**

**}**

**class HistoryFrame extends JFrame {**

**public HistoryFrame(RailwayManagementSystem system, ArrayList<String> bookingHistory) {**

**setTitle("Booking History");**

**setExtendedState(JFrame.MAXIMIZED\_BOTH); // Fullscreen**

**setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);**

**JTextArea historyTextArea = new JTextArea();**

**historyTextArea.setEditable(false);**

**historyTextArea.setFont(new Font("Arial", Font.PLAIN, 16));**

**for (String record : bookingHistory) {**

**historyTextArea.append(record + "\n\n");**

**}**

**JScrollPane scrollPane = new JScrollPane(historyTextArea);**

**scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED);**

**JPanel buttonPanel = new JPanel();**

**JButton backButton = new JButton("Back");**

**backButton.setPreferredSize(new Dimension(100, 30));**

**backButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**dispose(); // Close the HistoryFrame**

**}**

**});**

**buttonPanel.add(backButton);**

**// Add components to the frame**

**setLayout(new BorderLayout());**

**add(scrollPane, BorderLayout.CENTER);**

**add(buttonPanel, BorderLayout.SOUTH);**

**setVisible(true);**

**}**

**}**

**// Booking Frame for ticket booking**

**class BookTicketFrame extends JFrame {**

**private RailwayManagementSystem system;**

**private ArrayList<Train> trainList;**

**private ArrayList<String> bookingHistory;**

**public BookTicketFrame(RailwayManagementSystem system, ArrayList<Train> trains, ArrayList<String> history) {**

**this.system = system;**

**this.trainList = trains;**

**this.bookingHistory = history;**

**setTitle("Book Ticket");**

**setExtendedState(JFrame.MAXIMIZED\_BOTH); // Fullscreen**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**BackgroundPanel backgroundPanel = new BackgroundPanel("trainn.jpg"); // Ensure this image file is available in the resources folder**

**backgroundPanel.setLayout(new BorderLayout());**

**JPanel formPanel = new JPanel();**

**formPanel.setOpaque(false); // Make the form panel transparent**

**formPanel.setLayout(new GridBagLayout());**

**GridBagConstraints gbc = new GridBagConstraints();**

**gbc.insets = new Insets(10, 10, 10, 10);**

**gbc.fill = GridBagConstraints.HORIZONTAL;**

**JLabel sourceLabel = new JLabel("Source:");**

**JTextField sourceField = new JTextField(10);**

**JLabel destinationLabel = new JLabel("Destination:");**

**JTextField destinationField = new JTextField(10);**

**JLabel seatsLabel = new JLabel("Number of Seats:");**

**JTextField seatsField = new JTextField(5);**

**JLabel divisionLabel = new JLabel("Division:");**

**String[] divisions = {"1st Class", "2nd Class", "3rd Class"};**

**JComboBox<String> divisionCombo = new JComboBox<>(divisions);**

**JPanel dynamicFieldsPanel = new JPanel();**

**dynamicFieldsPanel.setOpaque(false);**

**dynamicFieldsPanel.setLayout(new BoxLayout(dynamicFieldsPanel, BoxLayout.Y\_AXIS));**

**// Adding components to the form panel**

**gbc.gridx = 0; gbc.gridy = 0; formPanel.add(sourceLabel, gbc);**

**gbc.gridx = 1; formPanel.add(sourceField, gbc);**

**gbc.gridx = 0; gbc.gridy = 1; formPanel.add(destinationLabel, gbc);**

**gbc.gridx = 1; formPanel.add(destinationField, gbc);**

**gbc.gridx = 0; gbc.gridy = 2; formPanel.add(seatsLabel, gbc);**

**gbc.gridx = 1; formPanel.add(seatsField, gbc);**

**gbc.gridx = 0; gbc.gridy = 3; formPanel.add(divisionLabel, gbc);**

**gbc.gridx = 1; formPanel.add(divisionCombo, gbc);**

**gbc.gridx = 0; gbc.gridy = 4; gbc.gridwidth = 2; formPanel.add(dynamicFieldsPanel, gbc);**

**// Seats Input Listener**

**seatsField.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**dynamicFieldsPanel.removeAll(); // Clear previous fields**

**int numberOfSeats;**

**try {**

**numberOfSeats = Integer.parseInt(seatsField.getText().trim());**

**} catch (NumberFormatException ex) {**

**numberOfSeats = 0;**

**}**

**for (int i = 0; i < numberOfSeats; i++) {**

**JPanel seatPanel = new JPanel(new FlowLayout());**

**seatPanel.setOpaque(false); // Make the seat panel transparent**

**JTextField nameField = new JTextField(10);**

**JTextField ageField = new JTextField(5);**

**seatPanel.add(new JLabel("Name " + (i + 1) + ":"));**

**seatPanel.add(nameField);**

**seatPanel.add(new JLabel("Age " + (i + 1) + ":"));**

**seatPanel.add(ageField);**

**dynamicFieldsPanel.add(seatPanel);**

**}**

**dynamicFieldsPanel.revalidate();**

**dynamicFieldsPanel.repaint();**

**}**

**});**

**// Booking Button**

**JButton bookButton = new JButton("Book Ticket");**

**bookButton.setPreferredSize(new Dimension(150, 30));**

**bookButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**String source = sourceField.getText().trim();**

**String destination = destinationField.getText().trim();**

**Train selectedTrain = null;**

**for (Train train : trainList) {**

**if (train.source.equalsIgnoreCase(source) && train.destination.equalsIgnoreCase(destination)) {**

**selectedTrain = train;**

**break;**

**}**

**}**

**if (selectedTrain == null) {**

**JOptionPane.showMessageDialog(null, "Invalid source or destination. Please select from available trains.");**

**return;**

**}**

**String division = (String) divisionCombo.getSelectedItem();**

**String seatsText = seatsField.getText().trim();**

**int seats = Integer.parseInt(seatsText);**

**if (selectedTrain.balance < seats) {**

**JOptionPane.showMessageDialog(null, "Insufficient seats available.");**

**return;**

**}**

**StringBuilder bookingDetails = new StringBuilder();**

**// Collecting names and ages**

**for (Component component : dynamicFieldsPanel.getComponents()) {**

**if (component instanceof JPanel) {**

**JPanel seatPanel = (JPanel) component;**

**JTextField nameField = (JTextField) seatPanel.getComponent(1);**

**JTextField ageField = (JTextField) seatPanel.getComponent(3);**

**String name = nameField.getText().trim();**

**String ageText = ageField.getText().trim();**

**if (name.isEmpty() || ageText.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Please fill in all fields.");**

**return;**

**}**

**bookingDetails.append("Name: ").append(name)**

**.append(", Age: ").append(ageText).append(", ");**

**}**

**}**

**// Record booking and update the train's balance**

**selectedTrain.balance -= seats; // Update the balance of seats**

**String booking = bookingDetails.toString() + "Source: " + source + ", Destination: " + destination + ", Division: " + division + ", Seats: " + seats;**

**bookingHistory.add(booking);**

**JOptionPane.showMessageDialog(null, "Booking Successful!");**

**dispose();**

**system.showMainFrame();**

**}**

**});**

**// Back Button**

**JButton backButton = new JButton("Back");**

**backButton.setPreferredSize(new Dimension(150, 30));**

**backButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**dispose();**

**system.showMainFrame();**

**}**

**});**

**gbc.gridy = 5; formPanel.add(bookButton, gbc);**

**gbc.gridy = 6; formPanel.add(backButton, gbc);**

**backgroundPanel.add(formPanel, BorderLayout.CENTER);**

**add(backgroundPanel);**

**setVisible(true);**

**}**

**class BackgroundPanel extends JPanel {**

**private Image backgroundImage;**

**public BackgroundPanel(String imagePath) {**

**URL imageURL = getClass().getClassLoader().getResource(imagePath);**

**if (imageURL != null) {**

**backgroundImage = new ImageIcon(imageURL).getImage();**

**} else {**

**System.out.println("Background image not found at: " + imagePath);**

**}**

**}**

**@Override**

**protected void paintComponent(Graphics g) {**

**super.paintComponent(g);**

**if (backgroundImage != null) {**

**g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);**

**}**

**}**

**}**

**}**

**public class RailwayManagementSystem {**

**private ArrayList<Train> trainList;**

**private ArrayList<String> bookingHistory;**

**public RailwayManagementSystem() {**

**trainList = new ArrayList<>();**

**bookingHistory = new ArrayList<>();**

**trainList.add(new Train(12951, "Rajdhani Express", "Mumbai Central", "New Delhi", "16:35", "08:35", 1200, 1200));**

**trainList.add(new Train(12001, "Shatabdi Express", "New Delhi", "Kalka", "06:00", "11:10", 600, 6000));**

**trainList.add(new Train(12245, "Duronto Express", "Howrah", "Yesvantpur", "20:00", "04:30", 800, 800));**

**trainList.add(new Train(12909, "Garib Rath Express", "Bandra Terminus", "Nizamuddin", "16:55", "09:40", 700, 700));**

**trainList.add(new Train(22438, "Humsafar Express", "Allahabad", "Anand Vihar", "22:10", "04:00", 1000, 1000));**

**trainList.add(new Train(12050, "Gatimaan Express", "New Delhi", "Jhansi", "08:10", "10:48", 500, 500));**

**trainList.add(new Train(22672, "Tejas Express", "Madurai", "Chennai Egmore", "15:00", "21:15", 500, 500));**

**trainList.add(new Train(12072, "Jan Shatabdi Express", "Jalna", "Mumbai CST", "05:50", "12:00", 300, 300));**

**trainList.add(new Train(12651, "Sampark Kranti Express", "Madurai", "Nizamuddin", "05:15", "10:40", 700, 700));**

**trainList.add(new Train(12081, "Jan Shatabdi Express", "Kannur", "Thiruvananthapuram", "14:55", "20:45", 400, 40));**

**trainList.add(new Train(11020, "Konark Express", "Bhubaneswar", "Mumbai CST", "15:25", "03:45", 800, 800));**

**trainList.add(new Train(12634, "Kanyakumari Express", "Hazrat Nizamuddin", "Kanyakumari", "17:20", "03:10", 700, 720));**

**trainList.add(new Train(11078, "Jhelum Express", "Jammu Tawi", "Pune", "21:45", "08:05", 600, 600));**

**trainList.add(new Train(11057, "Amritsar Express", "Mumbai CST", "Amritsar", "23:30", "08:30", 400, 400));**

**trainList.add(new Train(12436, "Dibrugarh Rajdhani Express", "New Delhi", "Dibrugarh", "16:25", "05:35", 1000, 1000));**

**trainList.add(new Train(12301, "Howrah Rajdhani Express", "Howrah", "New Delhi", "16:55", "10:00", 1200, 1200));**

**trainList.add(new Train(12015, "Ajmer Shatabdi Express", "New Delhi", "Ajmer", "06:05", "12:45", 600, 600));**

**trainList.add(new Train(12626, "Kerala Express", "New Delhi", "Trivandrum", "13:15", "19:15", 900, 900));**

**trainList.add(new Train(12401, "Nanda Devi Express", "Dehradun", "New Delhi", "22:50", "05:15", 500, 500));**

**trainList.add(new Train(14801, "Bhagat Ki Kothi Express", "Jodhpur", "Bhagat Ki Kothi", "23:45", "08:30", 600, 600));**

**trainList.add(new Train(11301, "Udyan Express", "Mumbai CST", "Bangalore", "20:30", "08:50", 800, 800));**

**trainList.add(new Train(12311, "Kalka Mail", "Howrah", "Kalka", "19:40", "04:30", 700, 700));**

**trainList.add(new Train(15906, "Vivek Express", "Dibrugarh", "Kanyakumari", "23:45", "22:00", 900, 900));**

**trainList.add(new Train(12260, "Sealdah Duronto Express", "Sealdah", "Bikaner", "12:50", "11:10", 800, 800));**

**trainList.add(new Train(22921, "Bandra Terminus - Gorakhpur Humsafar Exp", "Bandra Terminus", "Gorakhpur", "23:25", "07:25", 1000, 1000));**

**trainList.add(new Train(11043, "Lokmanya Tilak - Madurai Express", "Lokmanya Tilak", "Madurai", "00:15", "05:30", 700, 700));**

**trainList.add(new Train(22501, "KSR Bengaluru - New Tinsukia Express", "Bangalore", "New Tinsukia", "03:10", "19:15", 800, 800));**

**trainList.add(new Train(11078, "Jammu Tawi - Pune Jhelum Express", "Jammu Tawi", "Pune", "21:45", "08:15", 500, 500));**

**trainList.add(new Train(16317, "Kochuveli - Amritsar Express", "Kochuveli", "Amritsar", "04:50", "21:25", 600, 600));**

**trainList.add(new Train(52541, "New Jalpaiguri - Darjeeling Express", "New Jalpaiguri", "Darjeeling", "09:10", "16:30", 400, 400));**

**trainList.add(new Train(12682, "Pandian Express", "Madurai", "Chennai Egmore", "20:00", "04:30", 800, 800));**

**trainList.add(new Train(16127, "Kanyakumari Express", "Kanyakumari", "Chennai Egmore", "14:55", "22:45", 900, 900));**

**trainList.add(new Train(12660, "Vaigai Express", "Madurai", "Chennai Egmore", "05:00", "09:45", 700, 700));**

**trainList.add(new Train(16859, "Mysuru - Chennai Express", "Mysuru", "Chennai Egmore", "18:30", "22:30", 600, 600));**

**}**

**public void showMainFrame() {**

**JFrame mainFrame = new JFrame("Railway Management System");**

**mainFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**mainFrame.setExtendedState(JFrame.MAXIMIZED\_BOTH);  // Fullscreen**

**JLabel heading = new JLabel("Welcome to the Railway Management System", JLabel.CENTER);**

**heading.setFont(new Font("Arial", Font.BOLD, 36)); // Set font size and style**

**heading.setForeground(Color.black);**

**JPanel buttonPanel = new JPanel();**

**buttonPanel.setLayout(new FlowLayout());**

**// Create a BackgroundPanel with the path to your image**

**BackgroundPanel backgroundPanel = new BackgroundPanel("trai.jpg");**

**backgroundPanel.setLayout(new FlowLayout()); // Use FlowLayout or any layout you prefer**

**// Create buttons and add them to the background panel**

**JButton viewScheduleButton = new JButton("View Train Schedule");**

**viewScheduleButton.setPreferredSize(new Dimension(200, 30));**

**viewScheduleButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**new TrainScheduleFrame(RailwayManagementSystem.this, trainList);**

**}**

**});**

**JButton adminButton = new JButton("Admin Panel");**

**adminButton.setPreferredSize(new Dimension(200, 30));**

**adminButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**new AdminPanel(RailwayManagementSystem.this, trainList);**

**}**

**});**

**JButton bookTicketButton = new JButton("Book Ticket");**

**bookTicketButton.setPreferredSize(new Dimension(200, 30));**

**bookTicketButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**new BookTicketFrame(RailwayManagementSystem.this, trainList, bookingHistory);**

**}**

**});**

**JButton historyButton = new JButton("History");**

**historyButton.setPreferredSize(new Dimension(200, 30));**

**historyButton.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**new HistoryFrame(RailwayManagementSystem.this, bookingHistory);**

**}**

**});**

**backgroundPanel.add(viewScheduleButton);**

**backgroundPanel.add(adminButton);**

**backgroundPanel.add(bookTicketButton);**

**backgroundPanel.add(historyButton);**

**mainFrame.setLayout(new BorderLayout());**

**mainFrame.add(heading, BorderLayout.NORTH);**

**mainFrame.add(backgroundPanel);**

**mainFrame.setVisible(true);**

**}**

**public static void main(String[] args) {**

**SwingUtilities.invokeLater(new Runnable() {**

**@Override**

**public void run() {**

**new RailwayManagementSystem().showMainFrame();**

**}**

**});**

**}**

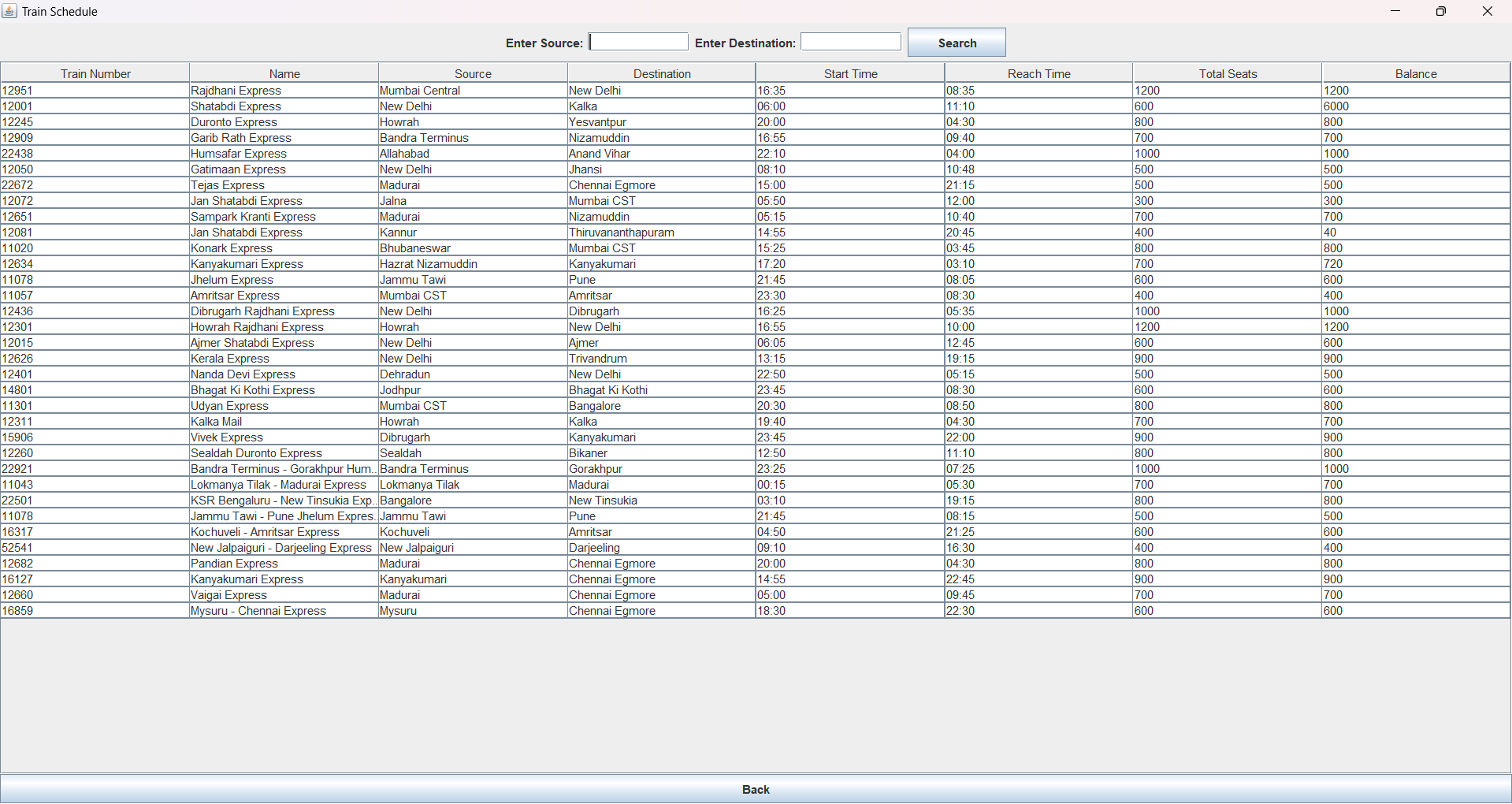
**}**

**Output:**

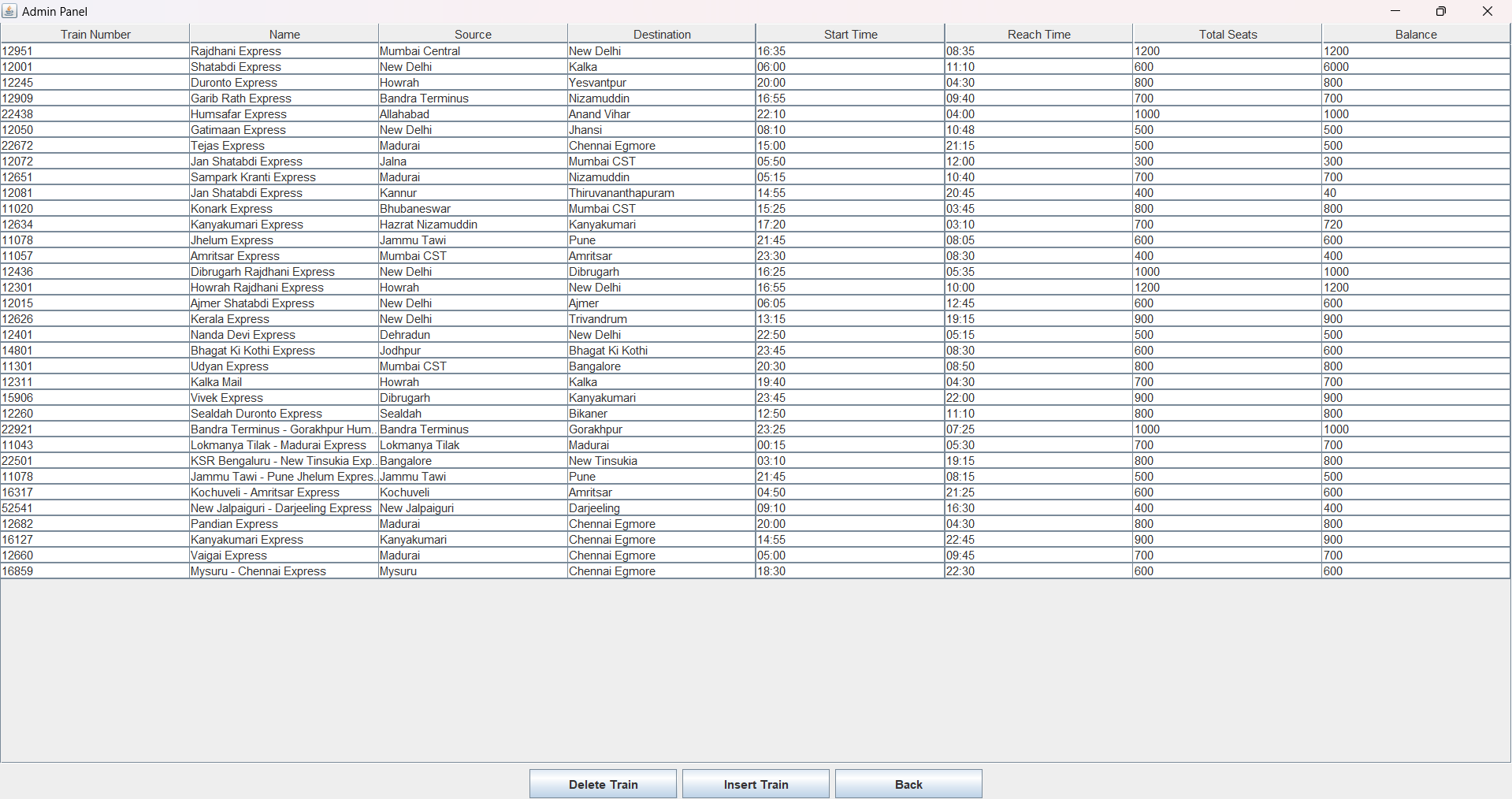
**Main page:**

****

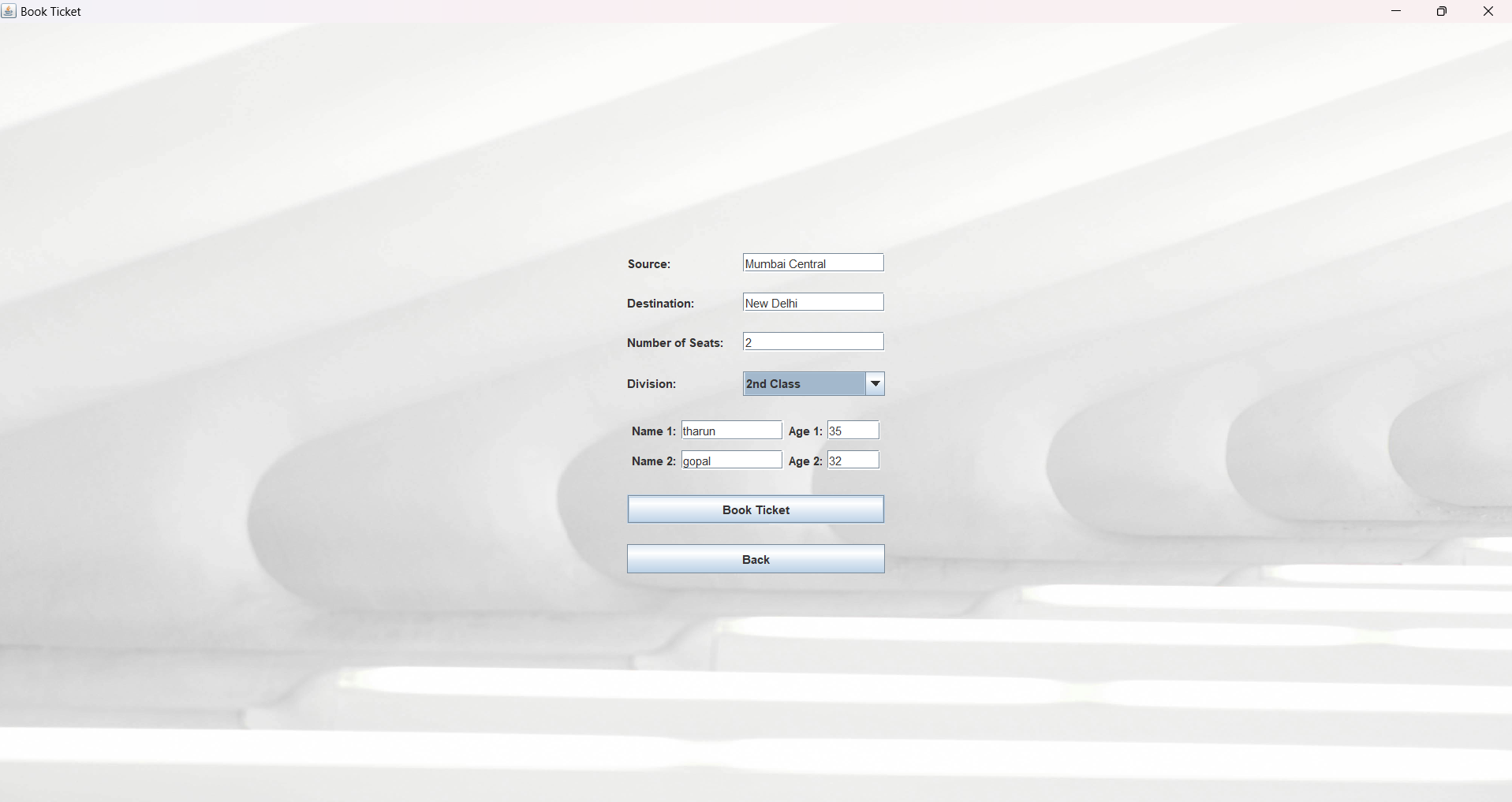
**Schedule panel:**

****

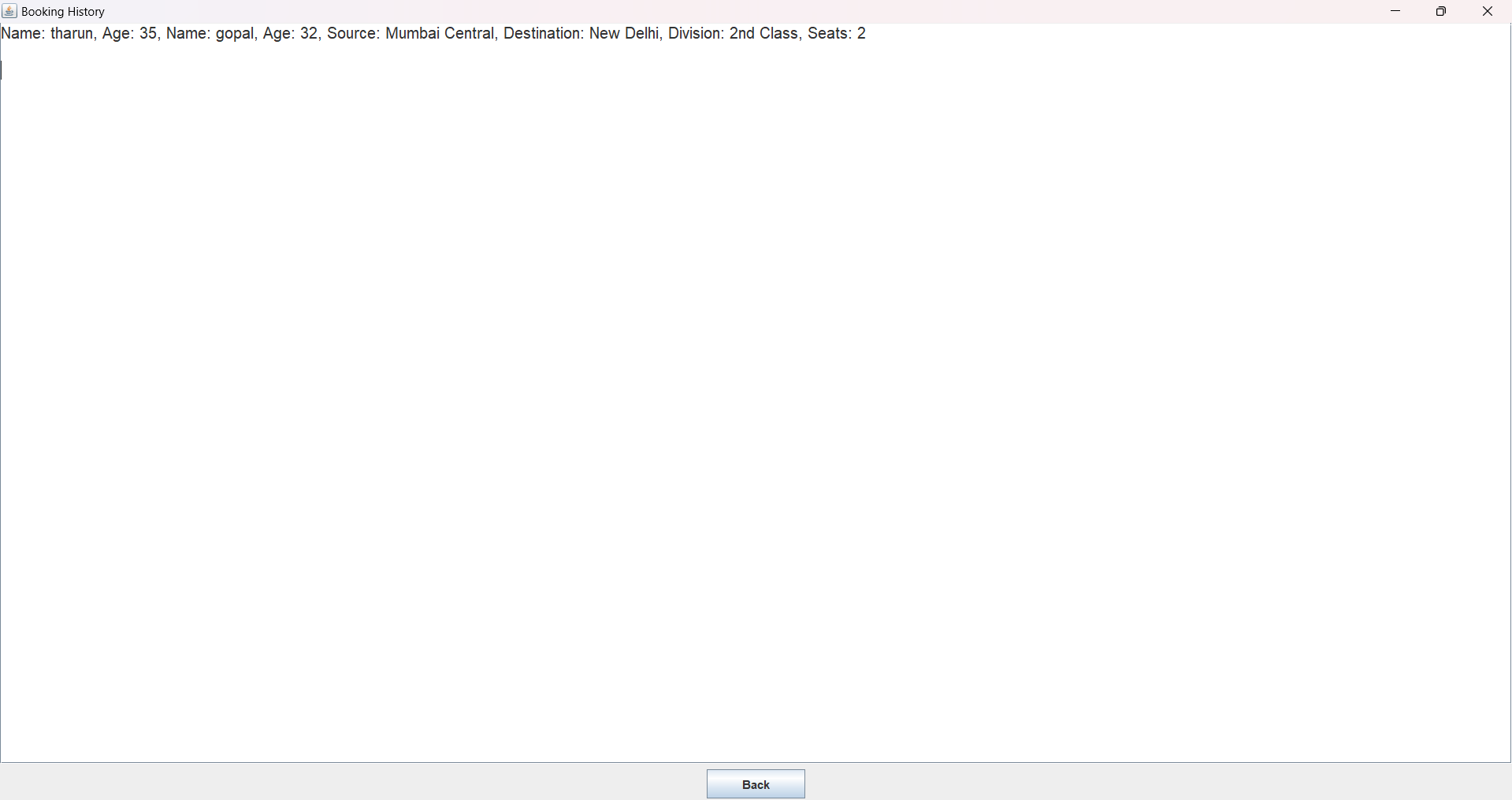
**Admin panel:**

****

**Ticket Booking panel:**

****

**History Panel:**

****