|  |
| --- |
| package com.InternCrowed.MovieTicket;  import java.awt.\*;  import java.awt.event.\*;  import javax.swing.\*;  import java.util.\*;  import java.util.List;  import java.io.File;  import java.io.IOException;  import javax.imageio.ImageIO;  import java.awt.image.BufferedImage;  public class MovieTicketBookingSystem extends JFrame {      private JLabel movieLabel;      private JLabel ticketTypeLabel;      private JLabel dateLabel;      private JLabel timeLabel;      private JLabel mobileNumberLabel;      private JCheckBox[][] seatCheckBoxes;      private JComboBox<String> movieComboBox;      private JComboBox<String> ticketTypeComboBox;      private JComboBox<String> dateComboBox;      private JComboBox<String> timeComboBox;      private JTextField mobileNumberTextField;      private JButton bookButton;      private static Map<String, List<Ticket>> bookings = new HashMap<>();      private static Map<String, Integer> prices = new HashMap<>();      static {          prices.put("Adult", 100);          prices.put("Child", 50);          prices.put("Senior Citizen", 30);      }      public MovieTicketBookingSystem() {          super("Movie Ticket Booking System");          setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);          movieLabel = new JLabel("Movie:");          ticketTypeLabel = new JLabel("Ticket Type:");          dateLabel = new JLabel("Date:");          timeLabel = new JLabel("Time:");          mobileNumberLabel = new JLabel("Mobile Number:");          movieComboBox = new JComboBox<>(new String[]{"Movie 1", "Movie 2", "Movie 3"});          ticketTypeComboBox = new JComboBox<>(new String[]{"Adult", "Child", "Senior Citizen"});          dateComboBox = new JComboBox<>(new String[]{"1/1/2023", "2/1/2023", "3/1/2023"});          timeComboBox = new JComboBox<>(new String[]{"10:00 AM", "12:00 PM", "2:00 PM"});          mobileNumberTextField = new JTextField();          bookButton = new JButton("Book");          int numRows = 5; // Number of rows in the seat layout          int numCols = 10; // Number of columns in the seat layout          seatCheckBoxes = new JCheckBox[numRows][numCols]; // Initialize the 2D array          JPanel seatSelectionPanel = new JPanel(new GridLayout(numRows, numCols));          // Create the seat selection checkboxes and add them to the panel          for (int row = 0; row < numRows; row++) {              for (int col = 0; col < numCols; col++) {                  JCheckBox seatCheckBox = new JCheckBox((row + 1) + "-" + (col + 1));                  seatSelectionPanel.add(seatCheckBox);                  seatCheckBoxes[row][col] = seatCheckBox; // Store the checkbox in the array              }          }          bookButton.addActionListener(new ActionListener() {              public void actionPerformed(ActionEvent e) {                  String movie = (String) movieComboBox.getSelectedItem();                  String ticketType = (String) ticketTypeComboBox.getSelectedItem();                  String date = (String) dateComboBox.getSelectedItem();                  String time = (String) timeComboBox.getSelectedItem();                  String mobileNumber = mobileNumberTextField.getText();                  // Book the tickets                  int price = prices.get(ticketType);                  int totalAmount = 0;                  List<String> selectedSeats = new ArrayList<>();                  for (int row = 0; row < numRows; row++) {                      for (int col = 0; col < numCols; col++) {                          JCheckBox seatCheckBox = seatCheckBoxes[row][col];                          if (seatCheckBox.isSelected()) {                              String seatNumber = (row + 1) + "-" + (col + 1);                              selectedSeats.add(seatNumber);                              totalAmount += price;                          }                      }                  }                  if (selectedSeats.isEmpty()) {                      JOptionPane.showMessageDialog(null, "No seats selected.");                      return;                  }                  for (String seatNumber : selectedSeats) {                      if (bookings.containsKey(movie)) {                          List<Ticket> movieBookings = bookings.get(movie);                          for (Ticket ticket : movieBookings) {                              if (ticket.getSeatNumber().equals(seatNumber)) {                                  JOptionPane.showMessageDialog(null, "Seat " + seatNumber + " already booked.");                                  return;                              }                          }                          movieBookings.add(new Ticket(seatNumber, ticketType));                      } else {                          List<Ticket> movieBookings = new ArrayList<>();                          movieBookings.add(new Ticket(seatNumber, ticketType));                          bookings.put(movie, movieBookings);                      }                  }                  String confirmationMessage = "Booking Confirmation:\n\n";                  confirmationMessage += "Movie: " + movie + "\n";                  confirmationMessage += "Seats: " + String.join(", ", selectedSeats) + "\n";                  confirmationMessage += "Ticket Type: " + ticketType + "\n";                  confirmationMessage += "Date: " + date + "\n";                  confirmationMessage += "Time: " + time + "\n";                  confirmationMessage += "Total Amount: " + totalAmount + "\n";                  confirmationMessage += "Mobile Number: " + mobileNumber + "\n";                  JOptionPane.showMessageDialog(null, confirmationMessage);                  // Save the receipt as an image                  saveReceiptAsImage(movie, selectedSeats.toArray(new String[0]), ticketType, date, time, totalAmount);                  // Generate and send OTP to the user's mobile number                  String otp = generateOTP();                  sendOTP(mobileNumber, otp);                  // Display seat booking confirmation status                  displayBookingConfirmation(movie, selectedSeats.toArray(new String[0]), ticketType, date, time, totalAmount, mobileNumber);              }          });          setLayout(new GridLayout(8, 2, 10, 10));          add(movieLabel);          add(movieComboBox);          add(ticketTypeLabel);          add(ticketTypeComboBox);          add(dateLabel);          add(dateComboBox);          add(timeLabel);          add(timeComboBox);          add(mobileNumberLabel);          add(mobileNumberTextField);          add(new JLabel()); // Empty label for spacing          add(bookButton);          add(new JLabel()); // Empty label for spacing          add(seatSelectionPanel); // Add the seat selection panel          pack();          setVisible(true);      }      private void saveReceiptAsImage(String movie, String[] seatNumbers, String ticketType, String date, String time, int totalAmount) {          try {              Robot robot = new Robot();              Rectangle screenRect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());              BufferedImage screenshot = robot.createScreenCapture(screenRect);              // Set the output file name              String fileName = movie + "\_receipt.png";              // Save the screenshot as an image file              ImageIO.write(screenshot, "png", new File(fileName));              System.out.println("Receipt saved as: " + fileName);          } catch (AWTException | IOException ex) {              ex.printStackTrace();          }      }      private String generateOTP() {          // Generate a random 4-digit OTP          Random random = new Random();          int otp = 1000 + random.nextInt(9000);          return String.valueOf(otp);      }      private void sendOTP(String mobileNumber, String otp) {          // Simulate sending OTP to the user's mobile number          System.out.println("OTP sent to " + mobileNumber + ": " + otp);      }      private void displayBookingConfirmation(String movie, String[] seatNumbers, String ticketType, String date, String time, int totalAmount, String mobileNumber) {          String confirmationMessage = "Booking Confirmation:\n\n";          confirmationMessage += "Movie: " + movie + "\n";          confirmationMessage += "Seats: " + String.join(", ", seatNumbers) + "\n";          confirmationMessage += "Ticket Type: " + ticketType + "\n";          confirmationMessage += "Date: " + date + "\n";          confirmationMessage += "Time: " + time + "\n";          confirmationMessage += "Total Amount: " + totalAmount + "\n";          confirmationMessage += "Mobile Number: " + mobileNumber + "\n";          JOptionPane.showMessageDialog(null, confirmationMessage);      }      public static void main(String[] args) {          SwingUtilities.invokeLater(new Runnable() {              public void run() {                  new MovieTicketBookingSystem();              }          });      }  }  class Ticket {      private String seatNumber;      private String ticketType;      public Ticket(String seatNumber, String ticketType) {          this.seatNumber = seatNumber;          this.ticketType = ticketType;      }      public String getSeatNumber() {          return seatNumber;      }      public String getTicketType() {          return ticketType;      }  } |

Output:

