FLIGHT TICKET BOOKING

A MINI-PROJECT BY:

SHRADHA.S 230701311

V.ROSHINI 230701270

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **“FLIGHT TICKET BOOKING”** is the bonafide work of **“V.ROSHINI , SHRADHA.S”** who carried out the project work under my supervision.

Submitted for the practical examination held on ____________________

SIGNATURE

Mr.G.SARAVANA GOKUL
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

SIGNATURE

Ms.V.JANANEE
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

This project presents a dynamic Java Swing application designed to revolutionize flight ticket booking by combining user-centric design with robust management capabilities. Integrated with a MySQL database, it offers an intuitive interface for users to search flights, choose seats, and make secure payments, while administrators gain tools for efficient flight, booking, and user management.

Key features include advanced search filters, dynamic seat selection with visual maps, trip type selection, add-on services like extra luggage and meal options, and secure payment integration. Users can enjoy personalized profiles to manage bookings and view history, while admins oversee operations through real-time updates and comprehensive dashboards.

The application employs responsive design for accessibility, robust error handling for smooth user experience, and state-of-the-art payment security. By blending sophisticated technology with usability, this solution aims to redefine the flight booking landscape, providing a seamless, engaging experience for both users and administrators.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

The Flight Ticket Booking App is a user-friendly platform designed to simplify travel planning. It allows users to search for flights based on destinations, dates, and preferences like seat types and trip types. With features like advanced search filters, personalized recommendations, and dynamic booking options, it streamlines the process of finding and booking flights. Users can manage bookings, choose add-on services, and make secure payments through the app. This app serves as a practical tool for saving time, enhancing convenience, and improving the overall travel booking experience.

1.2 IMPLEMENTATION

The **FLIGHT TICKET BOOKING** project discussed here is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.3 SCOPE OF THE PROJECT

The Flight Ticket Booking App offers a broad scope, including advanced flight search options, personalized booking recommendations, and travel planning support. It caters to diverse user needs with features like trip type selection, dynamic seat maps, and add-on services. By integrating secure payments, user profiles, and real-time management tools, the app serves as a versatile solution for both travelers and administrators, streamlining the booking process and enhancing the overall travel experience

1.4 WEBSITE FEATURES

1.4.1 Home page.

1.4.2 Flights to be searched and chosen.

1.4.3 Booking to be confirmed.

1.4.4 Payment Transaction to be done.

1.4.5 Booking Confirmation showing the details.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR	:	Intel i5
MEMORY SIZE	:	4GB(Minimum)
HARD DISK	:	500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE	:	Java, MySQL
FRONT-END	:	Java
BACK-END	:	MySQL
OPERATING SYSTEM	:	Windows 11

SAMPLE CODE

3.1 REGISTRATION PAGE DESIGN

```
import javax.swing.*;

public class A1App {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new A9LoginPage().setVisible(true);
        });
    }
}
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A10RegistrationPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JTextField emailField;
    private A6UserController userController;
    private BufferedImage backgroundImage;

    public A10RegistrationPage() {
        userController = new A6UserController();
        setTitle("Registration Page");
        setSize(728, 485);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/src/images/f3.jpg")); // Update with
your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        // Create a custom panel for the background image
        JPanel panel = new JPanel() {
```

```
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (backgroundImage != null) {
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
        }
    }
};

panel.setLayout(new GridBagLayout());
panel.setOpaque(false); // Allow transparency

// Create layout constraints for centering components
GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10); // Padding

// Add components to the panel
gbc.gridx = 0;
gbc.gridy = 0;

// Create a panel for the Username label
JPanel usernamePanel = new JPanel();
usernamePanel.setBackground(Color.LIGHT_GRAY); // Set background color for label
usernamePanel.add(new JLabel("Username:"));
panel.add(usernamePanel, gbc);

gbc.gridx = 1;
usernameField = new JTextField(15); // Fixed size for the text field
panel.add(usernameField, gbc);

gbc.gridx = 0;
gbc.gridy = 1;

// Create a panel for the Password label
JPanel passwordPanel = new JPanel();
passwordPanel.setBackground(Color.LIGHT_GRAY); // Set background color for label
passwordPanel.add(new JLabel("Password:"));
panel.add(passwordPanel, gbc);
gbc.gridx = 1;
passwordField = new JPasswordField(15); // Fixed size for the password field
panel.add(passwordField, gbc);
gbc.gridx = 0;
gbc.gridy = 2;

// Create a panel for the Email label
JPanel emailPanel = new JPanel();
emailPanel.setBackground(Color.LIGHT_GRAY); // Set background color for label
```

```
        emailPanel.add(new JLabel("Email:"));
        panel.add(emailPanel, gbc);
        gbc.gridx = 1;
        emailField = new JTextField(15); // Fixed size for the email field
        panel.add(emailField, gbc);
        gbc.gridx = 0;
        gbc.gridy = 3;
        JButton registerButton = new JButton("Register");
        registerButton.addActionListener(new RegisterAction());
        panel.add(registerButton, gbc);

        gbc.gridx = 1;
        JButton backButton = new JButton("Back to Login");
        backButton.addActionListener(e -> {
            dispose();
            new A9LoginPage().setVisible(true);
        });
        panel.add(backButton, gbc);

        // Add the panel to the frame
        setContentPane(panel);
    }

    private class RegisterAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            A3User user = new A3User();
            user.setUsername(usernameField.getText());
            user.setPassword(new String(passwordField.getPassword()));
            user.setEmail(emailField.getText());

            if (userController.registerUser(user)) {
                JOptionPane.showMessageDialog(A10RegistrationPage.this, "Registration
successful!");
                dispose();
                new A9LoginPage().setVisible(true);
            } else {
                JOptionPane.showMessageDialog(A10RegistrationPage.this, "Registration failed!",
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new A10RegistrationPage().setVisible(true));
    }
}
```

3.2 LOGIN PAGE DESIGN

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A9LoginPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private A6UserController userController;
    private BufferedImage backgroundImage;

    public A9LoginPage() {
        userController = new A6UserController();
        setTitle("Login Page");
        setSize(728, 485); // Smaller size
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/src/images/f4.jpg")); // Update with
your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        // Create a custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };

        panel.setLayout(new GridBagLayout());
        panel.setOpaque(false); // Allow transparency
        // Create layout constraints for centering components
        GridBagConstraints gbc = new GridBagConstraints();
      gbc.insets = new Insets(10, 10, 10, 10); // Padding

        // Username Label with Background
        gbc.gridx = 0;
        gbc.gridy = 0;
        JPanel usernamePanel = new JPanel();
        usernamePanel.setBackground(Color.LIGHT_GRAY); // Set background color for
label
        usernamePanel.add(new JLabel("Username:"));
        panel.add(usernamePanel, gbc);

        // Username Text Field
        gbc.gridx = 1;
        usernameField = new JTextField(15); // Fixed size for the text field
        panel.add(usernameField, gbc);

        // Password Label with Background
        gbc.gridx = 0;
        gbc.gridy = 1;
        JPanel passwordPanel = new JPanel();
        passwordPanel.setBackground(Color.LIGHT_GRAY); // Set background color for
label
        passwordPanel.add(new JLabel("Password:"));
        panel.add(passwordPanel, gbc);

        // Password Text Field
        gbc.gridx = 1;
        passwordField = new JPasswordField(15); // Fixed size for the password field
        panel.add(passwordField, gbc);

        gbc.gridx = 0;
              gbc.gridy = 2;
        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(new LoginAction());
        panel.add(loginButton, gbc);

        gbc.gridx = 1;
        JButton registerButton = new JButton("Register");
        registerButton.addActionListener(e -> {
           dispose();
           new A10RegistrationPage().setVisible(true);
        });
        panel.add(registerButton, gbc);

        // Add the panel to the frame
```

```
            setContentPane(panel);
        }

        private class LoginAction implements ActionListener {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                A3User user = userController.loginUser(username, password);
                if (user != null) {
                    JOptionPane.showMessageDialog(A9LoginPage.this, "Login Successful!");
                    dispose();
                 new A11FlightSearchPage(user).setVisible(true);
                } else {
                 JOptionPane.showMessageDialog(A9LoginPage.this, "Invalid credentials!",
"Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        }

        public static void main(String[] args) {
            SwingUtilities.invokeLater(() -> new A9LoginPage().setVisible(true));
        }
    }
```

3.3 FLIGHTS SEARCH PAGE DESIGN:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;
import java.util.Date;
import java.util.List;

public class A11FlightSearchPage extends JFrame {
    private JTextField originField;
    private JTextField destinationField;
    private JSpinner dateSpinner;
    private A3User user;
    private A7FlightController flightController;
    private BufferedImage backgroundImage;
    public A11FlightSearchPage(A3User user) {
        this.user = user;
        flightController = new A7FlightController();
        setTitle("Flight Search Page");
        setSize(728, 455);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/bin/images/f2.jpg")); // Update
with your image path
        } catch (IOException e) {
            e.printStackTrace();
        }
        // Create a custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
        panel.setLayout(new GridBagLayout());
        panel.setOpaque(false); // Allow transparency

        // Create layout constraints for centering components
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10); // Padding
        gbc.anchor = GridBagConstraints.CENTER; // Center components
        // Add components to the panel
        gbc.gridx = 0;
        gbc.gridy = 0;

        // Create a panel for the Origin label
        JPanel originPanel = new JPanel();
        originPanel.setBackground(Color.LIGHT_GRAY); // Set background color for
label
        originPanel.add(new JLabel("Origin:"));
        panel.add(originPanel, gbc);
        gbc.gridx = 1;
        originField = new JTextField(15);
        panel.add(originField, gbc);
        gbc.gridx = 0;
        gbc.gridy = 1;

        // Create a panel for the Destination label
        JPanel destinationPanel = new JPanel();
        destinationPanel.setBackground(Color.LIGHT_GRAY); // Set background color
for label
        destinationPanel.add(new JLabel("Destination:"));
        panel.add(destinationPanel, gbc);
        gbc.gridx = 1;
        destinationField = new JTextField(15);
        panel.add(destinationField, gbc);
        gbc.gridx = 0;
        gbc.gridy = 2;

        // Create a panel for the Departure Date label
        JPanel departureDatePanel = new JPanel();
        departureDatePanel.setBackground(Color.LIGHT_GRAY); // Set background
color for label
        departureDatePanel.add(new JLabel("Departure Date:"));
        panel.add(departureDatePanel, gbc);
        gbc.gridx = 1;
        dateSpinner = new JSpinner(new SpinnerDateModel());
        JSpinner.DateEditor dateEditor = new JSpinner.DateEditor(dateSpinner, "yyyy-
MM-dd");
        dateSpinner.setEditor(dateEditor);
        dateSpinner.setValue(new Date()); // Set current date as default
        panel.add(dateSpinner, gbc);
        gbc.gridx = 0;
        gbc.gridy = 3;
        JButton searchButton = new JButton("Search Flights");
        searchButton.addActionListener(new SearchAction());
        panel.add(searchButton, gbc);
        gbc.gridx = 1;
        JButton backButton = new JButton("Back to User Profile");
        backButton.addActionListener(e -> {
            dispose();
            new A16UserProfilePage(user).setVisible(true);
        });
        panel.add(backButton, gbc);

        // Add the panel to the frame
        setContentPane(panel);
    }

    private class SearchAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String origin = originField.getText().trim();
            String destination = destinationField.getText().trim();
            Date departureDate = (Date) dateSpinner.getValue();

            if (origin.isEmpty() || destination.isEmpty() || departureDate == null) {
                JOptionPane.showMessageDialog(A11FlightSearchPage.this, "Please fill all fields!", "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }
            List<A4Flight> flights = flightController.searchFlights(origin, destination, departureDate);
            if (flights.isEmpty()) {
                JOptionPane.showMessageDialog(A11FlightSearchPage.this, "No flights found!", "Info", JOptionPane.INFORMATION_MESSAGE);
            } else {
                dispose();
                new A12FlightDetailsPage(user, flights).setVisible(true);
            }
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new A11FlightSearchPage(new A3User()).setVisible(true));
    }
}
```

3.4 FLIGHT DETAILS PAGE DESIGN

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A12FlightDetailsPage extends JFrame {
    private JTable flightTable;
    private DefaultTableModel tableModel;
    private A3User user;
    private List<A4Flight> flights;
    private BufferedImage backgroundImage;

    public A12FlightDetailsPage(A3User user, List<A4Flight> flights) {
        this.user = user;
        this.flights = flights;

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("path/to/your/background_image.jpg")); // Update with your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        setTitle("Flight Details");
        setSize(600, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Create a custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
        panel.setLayout(new BorderLayout());
        panel.setOpaque(false); // Allow transparency

        // Create table model and table
        tableModel = new DefaultTableModel(new String[]{"Flight Number", "Origin",
"Destination", "Departure", "Arrival", "Price", "Available Seats"}, 0);
        flightTable = new JTable(tableModel) {
            @Override
            public boolean isCellEditable(int row, int column) {
                return false; // Make cells non-editable
            }
        };

        // Set table properties for transparency
        flightTable.setOpaque(false);
        flightTable.setBackground(new Color(0, 0, 0, 0)); // Fully transparent background
        flightTable.setSelectionBackground(new Color(135, 206, 250, 150)); // Light blue
selection color with some transparency
        flightTable.setSelectionForeground(Color.BLACK); // Black text on selection

        // Add flight data to the table
        for (A4Flight flight : flights) {
            tableModel.addRow(new Object[]{
                    flight.getFlightNumber(),
                    flight.getOrigin(),
                    flight.getDestination(),
                    flight.getDepartureDate(),
                    flight.getArrivalDate(),
                    flight.getPrice(),
                    flight.getAvailableSeats()
            });
        }

        // Add components to the panel
        panel.add(new JScrollPane(flightTable), BorderLayout.CENTER);
        JScrollPane scrollPane = new JScrollPane(flightTable);
        scrollPane.setOpaque(false);
        scrollPane.getViewport().setOpaque(false); // Make the viewport transparent
        panel.add(scrollPane, BorderLayout.CENTER);

        // Create a button panel for booking
        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
        JButton bookButton = new JButton("Book Flight");
        bookButton.addActionListener(new BookAction());
        buttonPanel.add(bookButton);
        panel.add(buttonPanel, BorderLayout.SOUTH);

        // Add the panel to the frame
        setContentPane(panel);
    }

    private class BookAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            int selectedRow = flightTable.getSelectedRow();
            if (selectedRow != -1) {
                A4Flight selectedFlight = flights.get(selectedRow);
                dispose();
                new A13BookingPage(user, selectedFlight).setVisible(true);
            } else {
                JOptionPane.showMessageDialog(A12FlightDetailsPage.this, "Please select a
flight!", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new A12FlightDetailsPage(new A3User(),
List.of()).setVisible(true));
    }
}
```

3.5 BOOKING PAGE DESIGN

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A13BookingPage extends JFrame {
    private A3User user;
    private A4Flight flight;
    private JTextField passengerNameField;
    private JTextField passengerEmailField; // Added email field
    private BufferedImage backgroundImage;

    public A13BookingPage(A3User user, A4Flight flight) {
        this.user = user;
        this.flight = flight;

        // Load background image
        try {
            backgroundImage = ImageIO.read(new File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/src/images/f5.jpg")); // Update with your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        setTitle("Booking Page");
        setSize(700, 500);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Create a custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
    panel.setLayout(new GridBagLayout());
    panel.setOpaque(false); // Allow transparency

    // Create layout constraints for centering components
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(10, 10, 10, 10); // Padding
    gbc.anchor = GridBagConstraints.CENTER; // Center components

    // Passenger Name Field
gbc.gridx = 0;
gbc.gridy = 0;

// Create a panel for the label
JPanel namePanel = new JPanel();
namePanel.setBackground(Color.LIGHT_GRAY); // Set background color for label
namePanel.add(new JLabel("Passenger Name:"));
panel.add(namePanel, gbc);

gbc.gridx = 1;
passengerNameField = new JTextField(15);
panel.add(passengerNameField, gbc);

// Passenger Email Field
gbc.gridx = 0;
gbc.gridy = 1;

// Create a panel for the label
JPanel emailPanel = new JPanel();
emailPanel.setBackground(Color.LIGHT_GRAY); // Set background color for label
emailPanel.add(new JLabel("Passenger Email:"));
panel.add(emailPanel, gbc);

gbc.gridx = 1;
passengerEmailField = new JTextField(15);
panel.add(passengerEmailField, gbc);

    // Confirm Booking Button
    gbc.gridx = 0;
    gbc.gridy = 2;
    JButton confirmButton = new JButton("Confirm Booking");
    confirmButton.addActionListener(new ConfirmBookingAction());
    panel.add(confirmButton, gbc);

    // Cancel Button
    gbc.gridx = 1;
    JButton cancelButton = new JButton("Cancel");
```

```
        cancelButton.addActionListener(e -> {
            dispose();
            new A11FlightSearchPage(user).setVisible(true);
        });
        panel.add(cancelButton, gbc);

        setContentPane(panel);
    }
    private class ConfirmBookingAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String email = passengerEmailField.getText(); // Get email from the text field
            String name = passengerNameField.getText(); // Get name from the text field

            if (email.isEmpty() || name.isEmpty()) {
                JOptionPane.showMessageDialog(A13BookingPage.this, "Please fill in all fields!", "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            // Proceed with booking
            A5Booking booking = new A5Booking();
            booking.setUserId(user.getId());
            booking.setFlightId(flight.getId());

            A8BookingController bookingController = new A8BookingController();
            if (bookingController.bookFlight(booking)) {
                JOptionPane.showMessageDialog(A13BookingPage.this, "Booking successful!");
                dispose();
                new A14PaymentPage(user, flight).setVisible(true);
            } else {
                JOptionPane.showMessageDialog(A13BookingPage.this, "Booking failed!", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            A3User user = new A3User(); // Dummy user for testing
            user.setEmail("test_user@example.com"); // Set a dummy email for testing
            A4Flight flight = new A4Flight(); // Dummy flight for testing
            new A13BookingPage(user, flight).setVisible(true);
        });
    }
}
```

3.6 PAYMENT PAGE DESIGN

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A14PaymentPage extends JFrame {
    private A3User user;
    private A4Flight flight;
    private BufferedImage backgroundImage;

    public A14PaymentPage(A3User user, A4Flight flight) {
        this.user = user;
        this.flight = flight;

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/src/images/f1.jpg")); // Update
with your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        setTitle("Payment Page");
        setSize(500, 333);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
        panel.setLayout(new GridBagLayout());
        panel.setOpaque(false); // Allow transparency

        // Create layout constraints for centering components
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10); // Padding
        gbc.anchor = GridBagConstraints.CENTER; // Center components

        // Add flight payment information
        JLabel paymentInfoLabel = new JLabel("Payment for Flight: " +
flight.getFlightNumber(), SwingConstants.CENTER);
        paymentInfoLabel.setForeground(Color.WHITE);
        gbc.gridx = 0;
        gbc.gridy = 0;
        panel.add(paymentInfoLabel, gbc);

        // Pay Now Button
        JButton payButton = new JButton("Pay Now");
        payButton.addActionListener(new PayAction());
        gbc.gridy = 1;
        panel.add(payButton, gbc);

        setContentPane(panel);
    }

    private class PayAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Here you would typically integrate with a payment gateway
            JOptionPane.showMessageDialog(A14PaymentPage.this, "Payment successful!
Booking confirmed!");
```

```
            dispose();
            new A15BookingConfirmationPage(user, flight).setVisible(true);
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            A3User user = new A3User(); // Dummy user for testing
            A4Flight flight = new A4Flight(); // Dummy flight for testing
            new A14PaymentPage(user, flight).setVisible(true);
        });
    }
}
```

3.7 BOOKING CONFIRMATION PAGE DESIGN

```
import javax.swing.*;
import java.awt.*;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A15BookingConfirmationPage extends JFrame {
    private A3User user;
    private A4Flight flight;
    private BufferedImage backgroundImage;

    public A15BookingConfirmationPage(A3User user, A4Flight flight) {
        this.user = user;
        this.flight = flight;

        // Load background image
        try {
            backgroundImage = ImageIO.read(new
File("C:/Users/aaral/Desktop/Javaprojects/Newproject2/src/images/f7.jpg")); // Update
with your image path
```

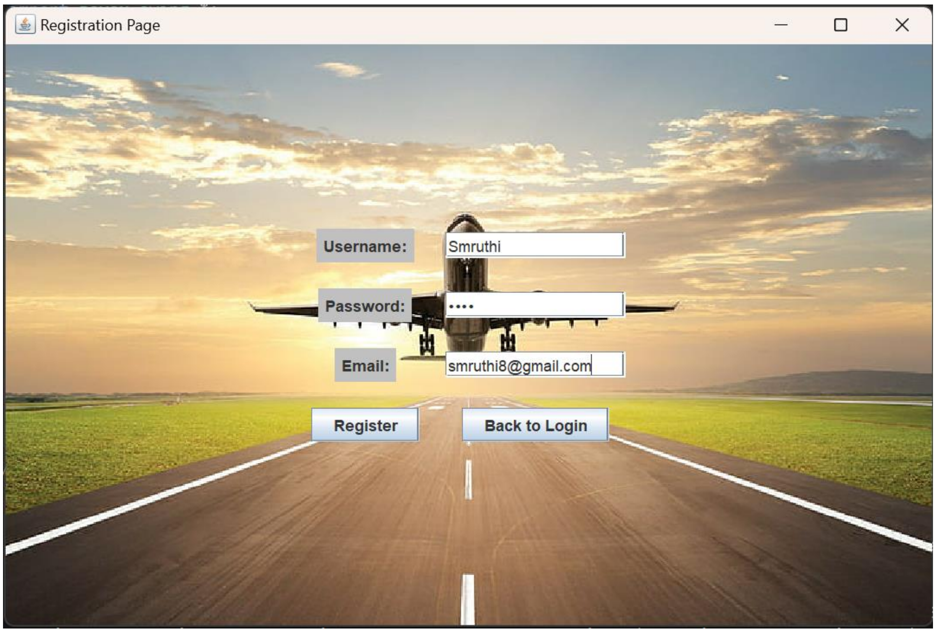
```
        } catch (IOException e) {
            e.printStackTrace();
        }

        setTitle("Booking Confirmation");
        setSize(480, 360); // Smaller size for the content
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
        panel.setLayout(new GridBagLayout());
        panel.setOpaque(false); // Allow transparency

        // Create layout constraints for centering components
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10); // Padding
        gbc.anchor = GridBagConstraints.CENTER; // Center components
        // Add labels for booking confirmation with dark gray background and white text
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.weighty = 1; // Allow components to grow vertically
        JPanel flightNumberPanel = new JPanel();
        flightNumberPanel.setBackground(Color.DARK_GRAY); // Set to dark gray
        JLabel flightNumberLabel = new JLabel("Booking Confirmed for Flight: " +
flight.getFlightNumber(), SwingConstants.CENTER);
        flightNumberLabel.setForeground(Color.WHITE); // Set text color to white
        flightNumberPanel.add(flightNumberLabel);
        panel.add(flightNumberPanel, gbc);
        gbc.gridy = 1;
        JPanel originPanel = new JPanel();
        originPanel.setBackground(Color.DARK_GRAY); // Set to dark gray
        JLabel originLabel = new JLabel("Origin: " + flight.getOrigin(),
SwingConstants.CENTER);
        originLabel.setForeground(Color.WHITE); // Set text color to white
        originPanel.add(originLabel);
        panel.add(originPanel, gbc);

        gbc.gridy = 2;
        JPanel destinationPanel = new JPanel();
        destinationPanel.setBackground(Color.DARK_GRAY); // Set to dark gray
        JLabel destinationLabel = new JLabel("Destination: " + flight.getDestination(),
SwingConstants.CENTER);
        destinationLabel.setForeground(Color.WHITE); // Set text color to white
        destinationPanel.add(destinationLabel);
        panel.add(destinationPanel, gbc);

        // Back to User Profile Button
        JButton backButton = new JButton("Back to User Profile");
        backButton.addActionListener(e -> {
            dispose();
            new A16UserProfilePage(user).setVisible(true);
        });
        gbc.gridy = 3;
        gbc.weighty = 0; // Do not grow vertically
        panel.add(backButton, gbc);
        setContentPane(panel);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            A3User user = new A3User(); // Dummy user for testing
            A4Flight flight = new A4Flight(); // Dummy flight for testing
            new A15BookingConfirmationPage(user, flight).setVisible(true);
        });
    }
}
```

3.8 USER PROFILE PAGE DESIGN

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class A16UserProfilePage extends JFrame {
    private A3User user;
    private BufferedImage backgroundImage;

    public A16UserProfilePage(A3User user) {
        this.user = user;

        // Load background image
        try {
            backgroundImage = ImageIO.read(new File("C:/Users/aaral/Downloads/f8.jpg"));
      // Update with your image path
        } catch (IOException e) {
            e.printStackTrace();
        }

        setTitle("User Profile");
        setSize(300, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Custom panel for the background image
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
                }
            }
        };
        panel.setLayout(new GridBagLayout());
        panel.setOpaque(false); // Allow transparency

        // Create layout constraints for centering components
        GridBagConstraints gbc = new GridBagConstraints();
}
        gbc.insets = new Insets(10, 10, 10, 10); // Padding
        gbc.anchor = GridBagConstraints.CENTER; // Center components

        // Add welcome message
        JLabel welcomeLabel = new JLabel("Welcome, " + user.getUsername(),
      SwingConstants.CENTER);
        welcomeLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Change font style
        gbc.gridy = 0;
        panel.add(welcomeLabel, gbc);

        // Add email label
        JLabel emailLabel = new JLabel("Email: " + user.getEmail(),
      SwingConstants.CENTER);
        emailLabel.setFont(new Font("Arial", Font.PLAIN, 14)); // Change font style
        gbc.gridy = 1;
        panel.add(emailLabel, gbc);

        // Create "Search Flights" button
        JButton searchFlightsButton = new JButton("Search Flights");
        searchFlightsButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Navigate to Flight Search Page
                new A11FlightSearchPage(user).setVisible(true);
                dispose(); // Close User Profile Page
            }
        });
        gbc.gridy = 2;
        panel.add(searchFlightsButton, gbc);

        // Set content pane
        setContentPane(panel);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            A3User user = new A3User(); // Dummy user for testing
            new A16UserProfilePage(user).setVisible(true);
        });
    }
}
```

SNAPSHOTS

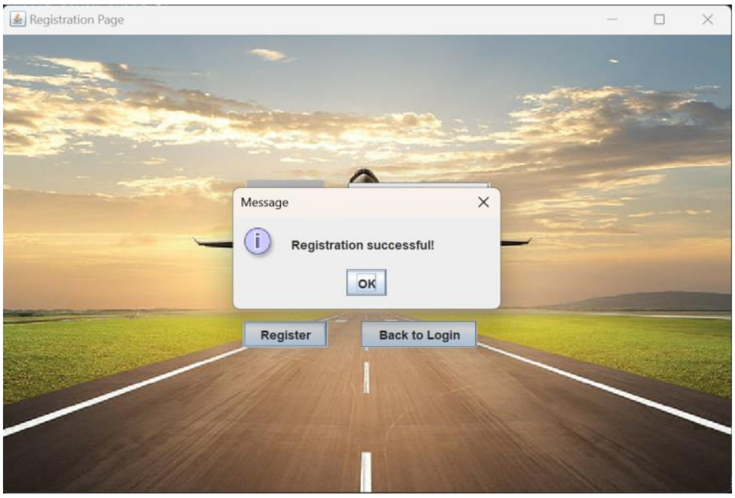
4.1 REGISTRATION PAGE



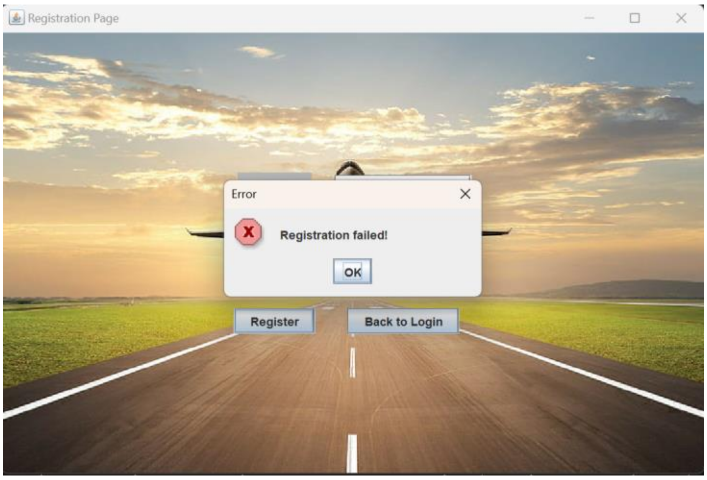
Username: Smruthi

Password: ••••

Email: smruthi8@gmail.com

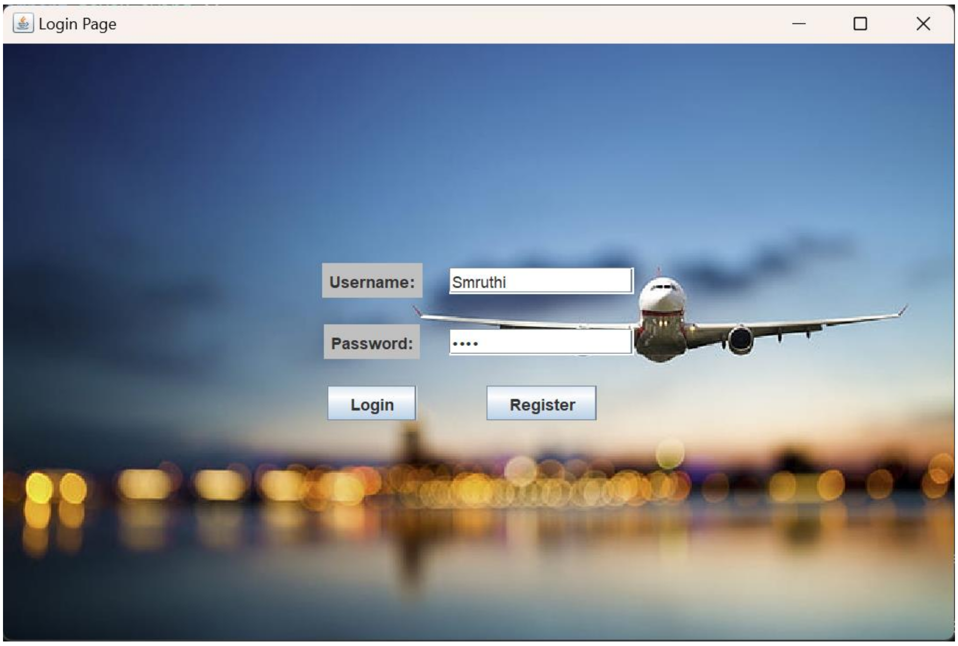


Registration successful!



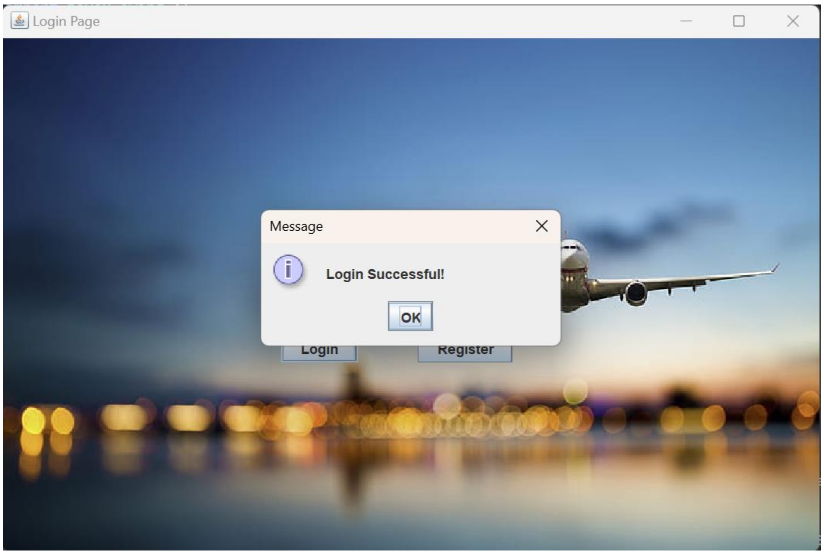
Registration failed!

4.2 LOGIN PAGE

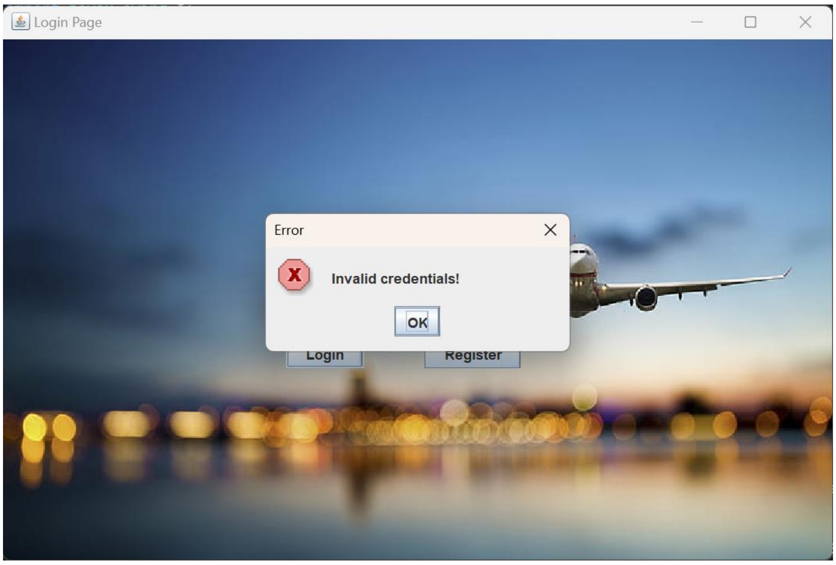


Username: Smruthi

Password: ••••

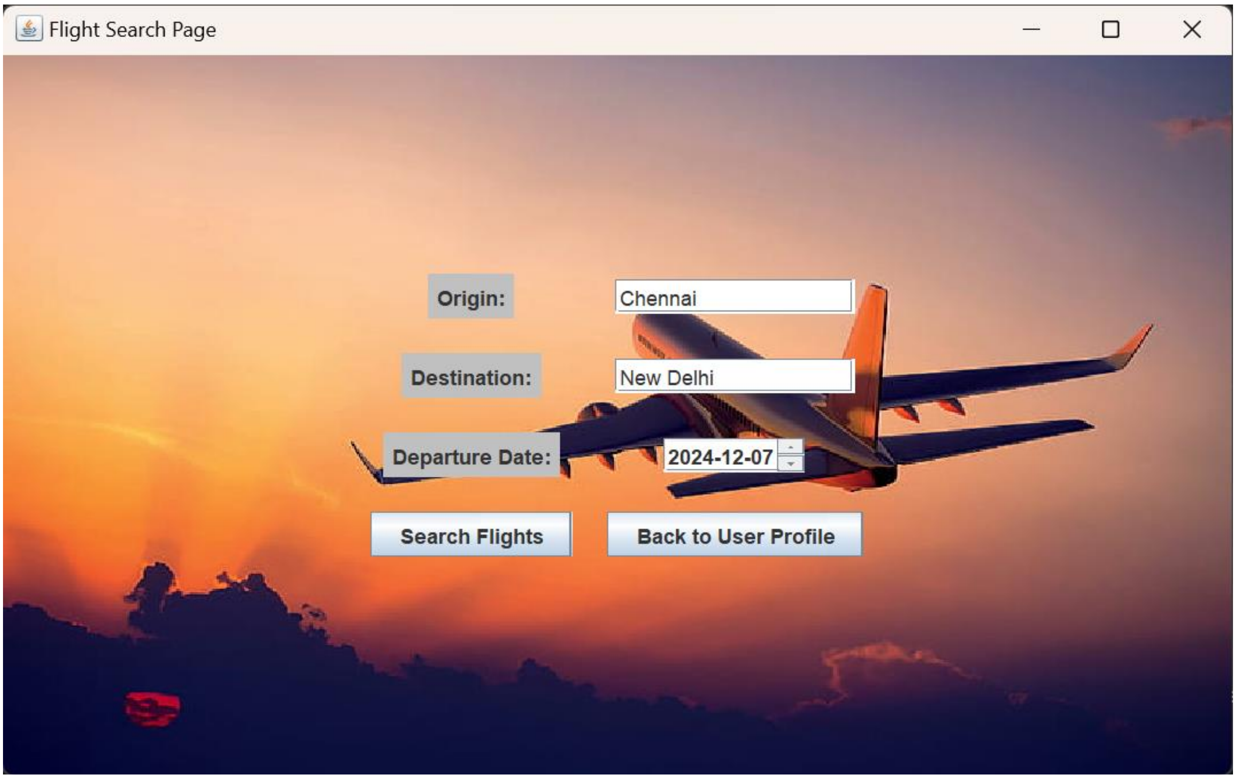


Login Successful!



Invalid credentials!

4.3 FLIGHTS SEARCH PAGE



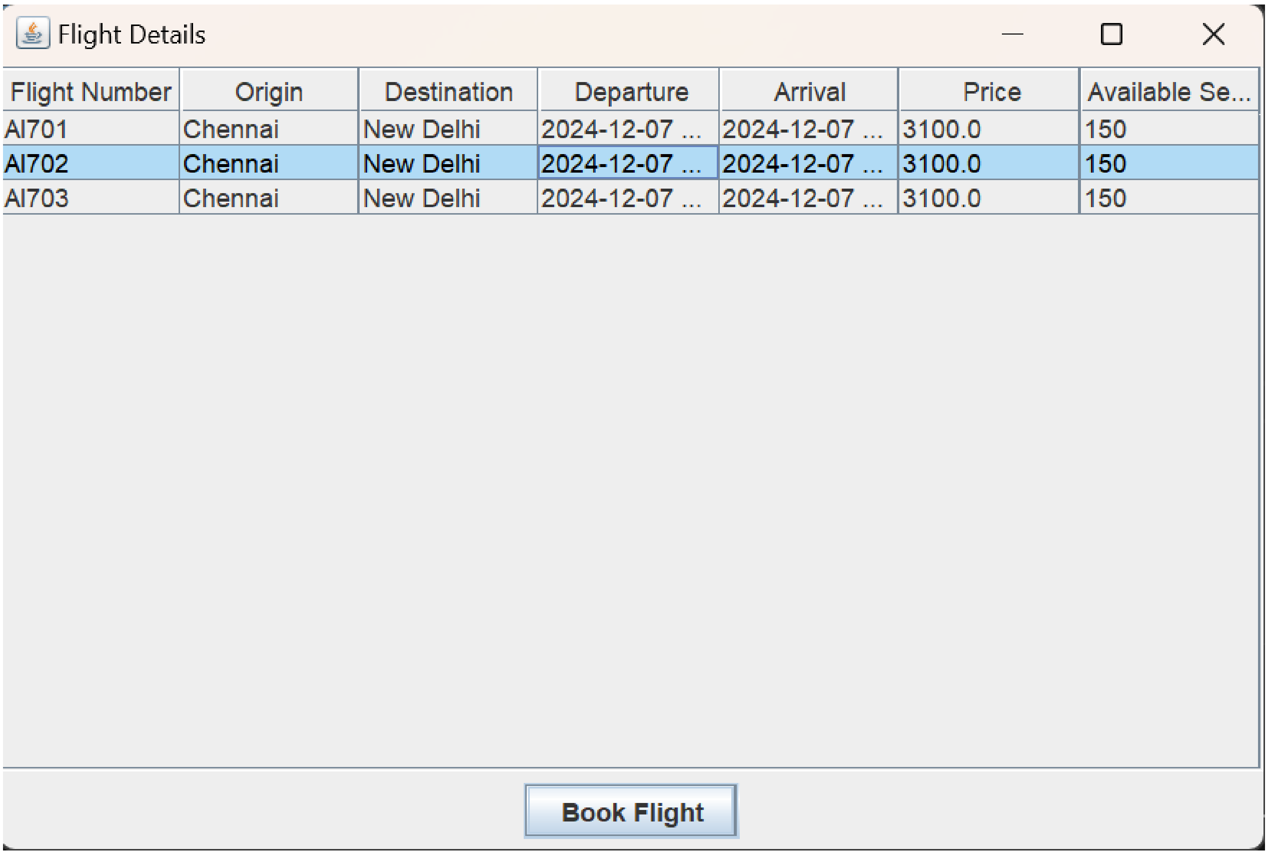
Flight Search Page

Origin: Chennai

Destination: New Delhi

Departure Date: 2024-12-07

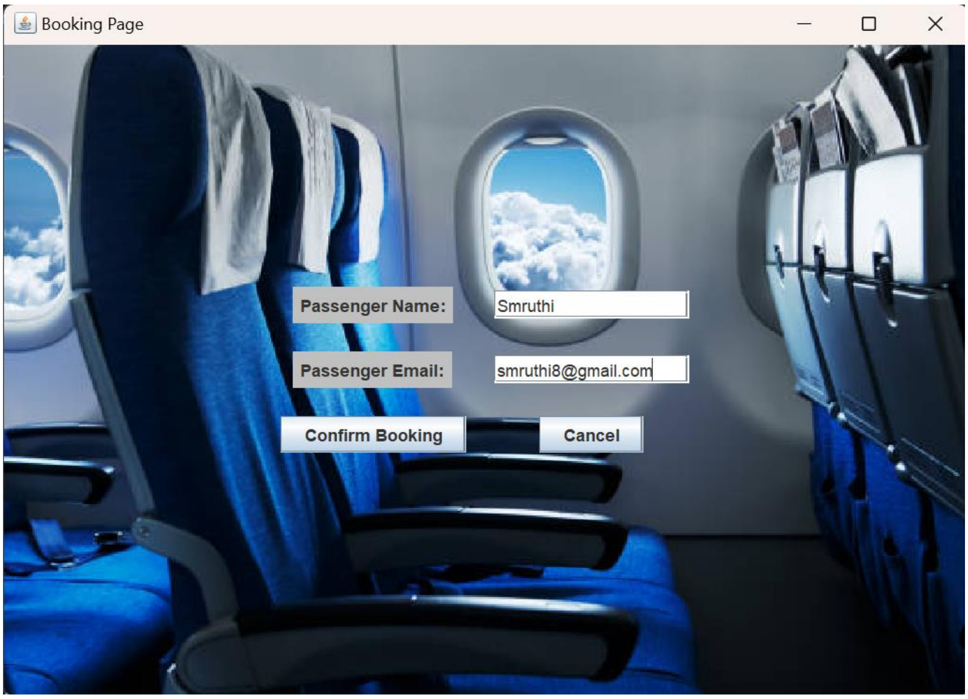
4.4 FLIGHT DETAILS PAGE



Flight Details

Flight Number	Origin	Destination	Departure	Arrival	Price	Available Se...
AI701	Chennai	New Delhi	2024-12-07 ...	2024-12-07 ...	3100.0	150
AI702	Chennai	New Delhi	2024-12-07 ...	2024-12-07 ...	3100.0	150
AI703	Chennai	New Delhi	2024-12-07 ...	2024-12-07 ...	3100.0	150

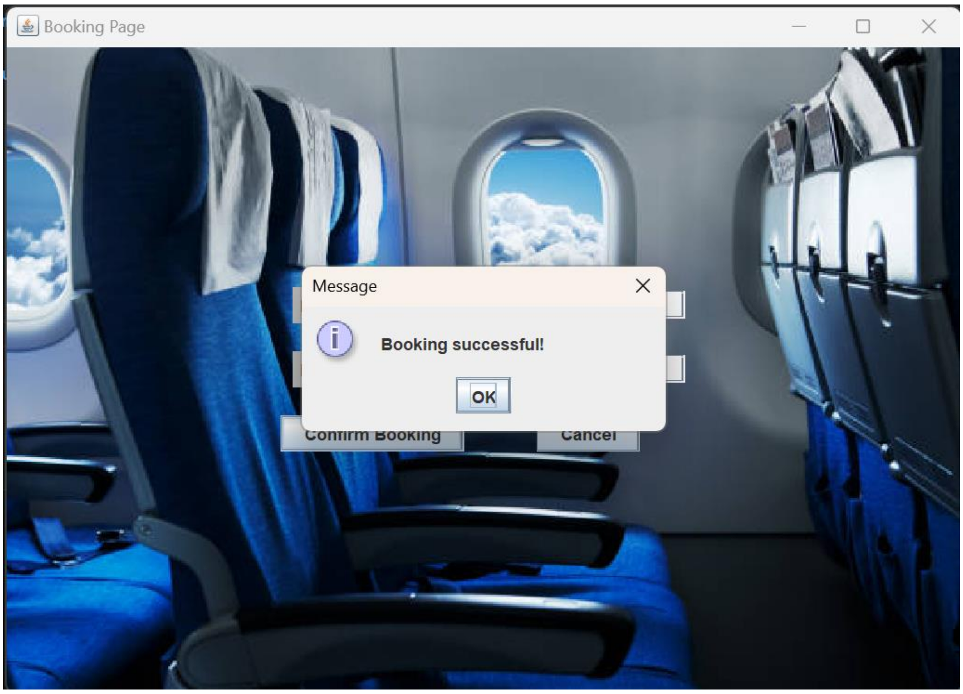
4.5 BOOKING PAGE



Booking Page

Passenger Name: Smruthi

Passenger Email: smruthi8@gmail.com

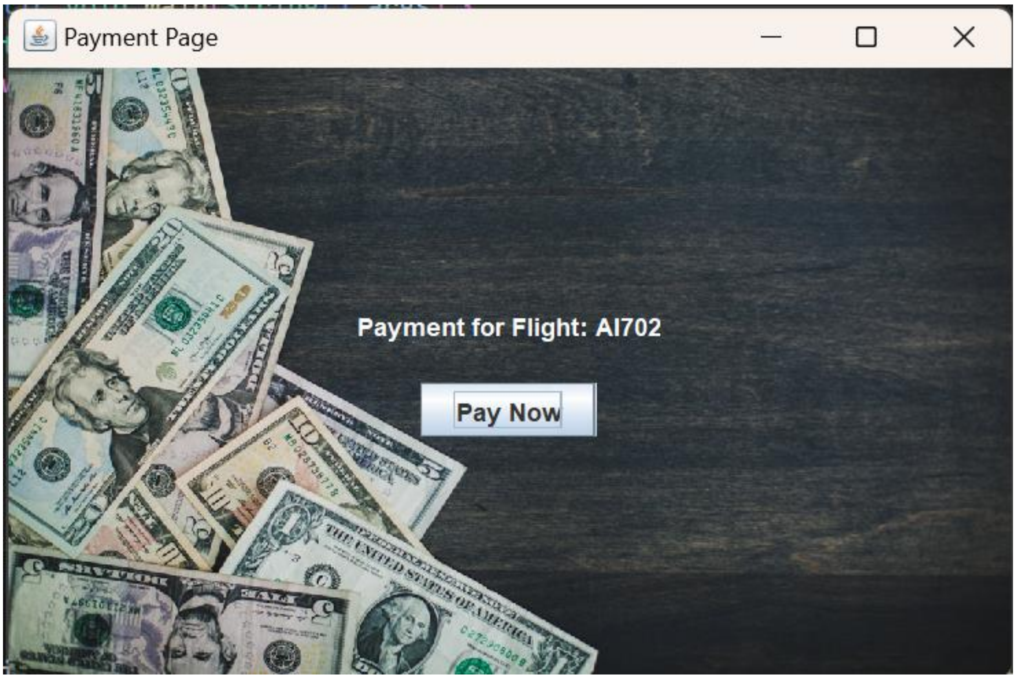


Booking Page

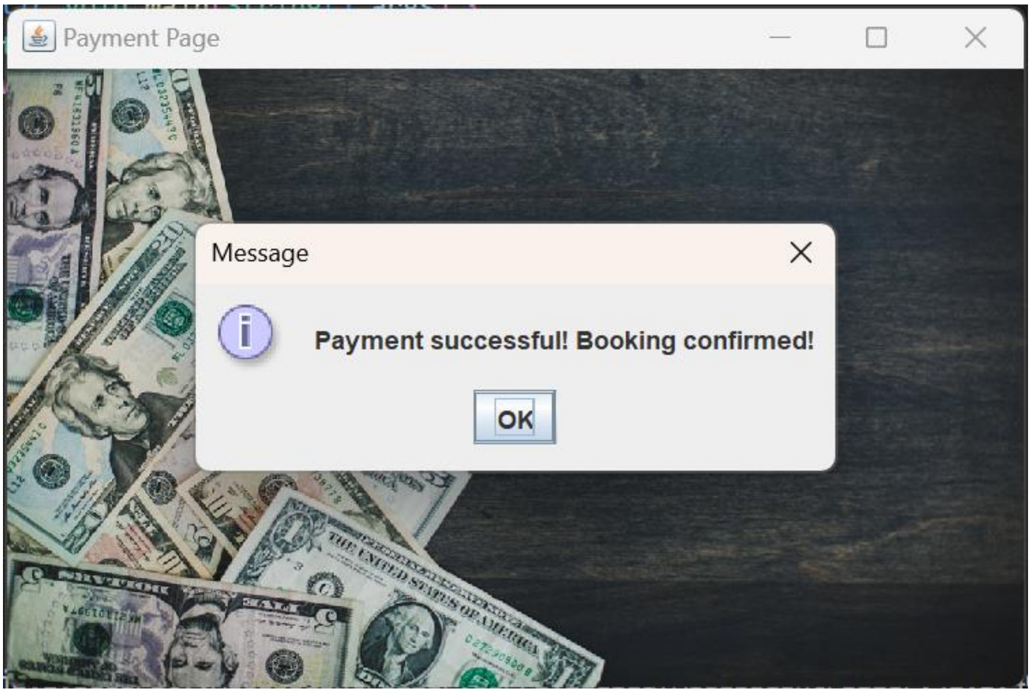
Message

Booking successful!

4.6 PAYMENT PAGE

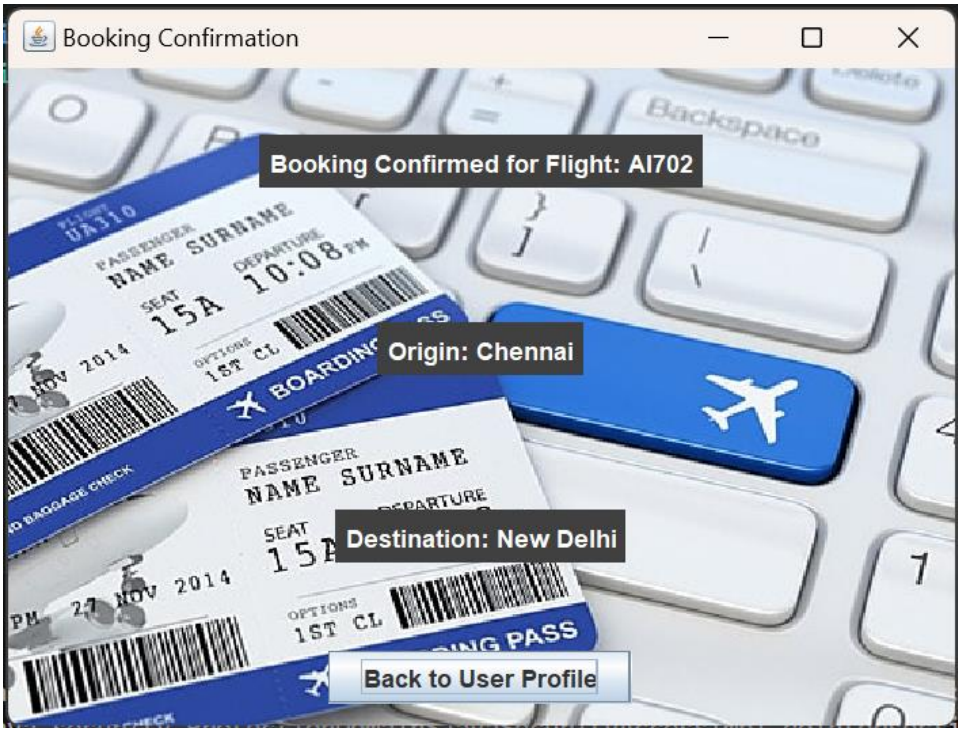


Payment for Flight: AI702



Payment successful! Booking confirmed!

4.7 BOOKING CONFIRMATION PAGE

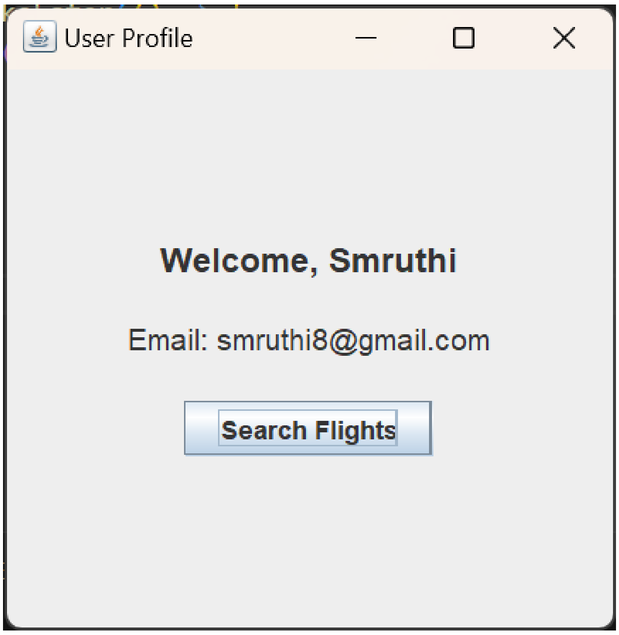


Booking Confirmed for Flight: AI702

Origin: Chennai

Destination: New Delhi

4.8 USER PROFILE PAGE



Welcome, Smruthi

Email: smruthi8@gmail.com

CONCLUSION

The Flight Ticket Booking App transforms the travel booking process by offering a seamless, personalized, and efficient platform for users and administrators. With features like advanced search filters, dynamic seat selection, add-on services, and secure payment options, the app enhances convenience and user satisfaction while streamlining operations for administrators.

In the future, the app can be enhanced with features such as **real-time flight tracking**, **multi-currency payment options**, **AI-based fare predictions**, and **loyalty programs for frequent users**. These additions will further elevate the user experience by providing more value and convenience.

Ultimately, the app redefines the travel booking experience, making it more accessible, user-friendly, and enjoyable for everyone, while continually evolving to meet the changing needs of travelers and administrators alike.

REFERENCES

4.7.1 https://www.javatpoint.com/java-tutorial

4.7.2 https://www.wikipedia.org/

4.7.3 https://www.w3schools.com/sql/

4.7.4 SQL | Codecademy