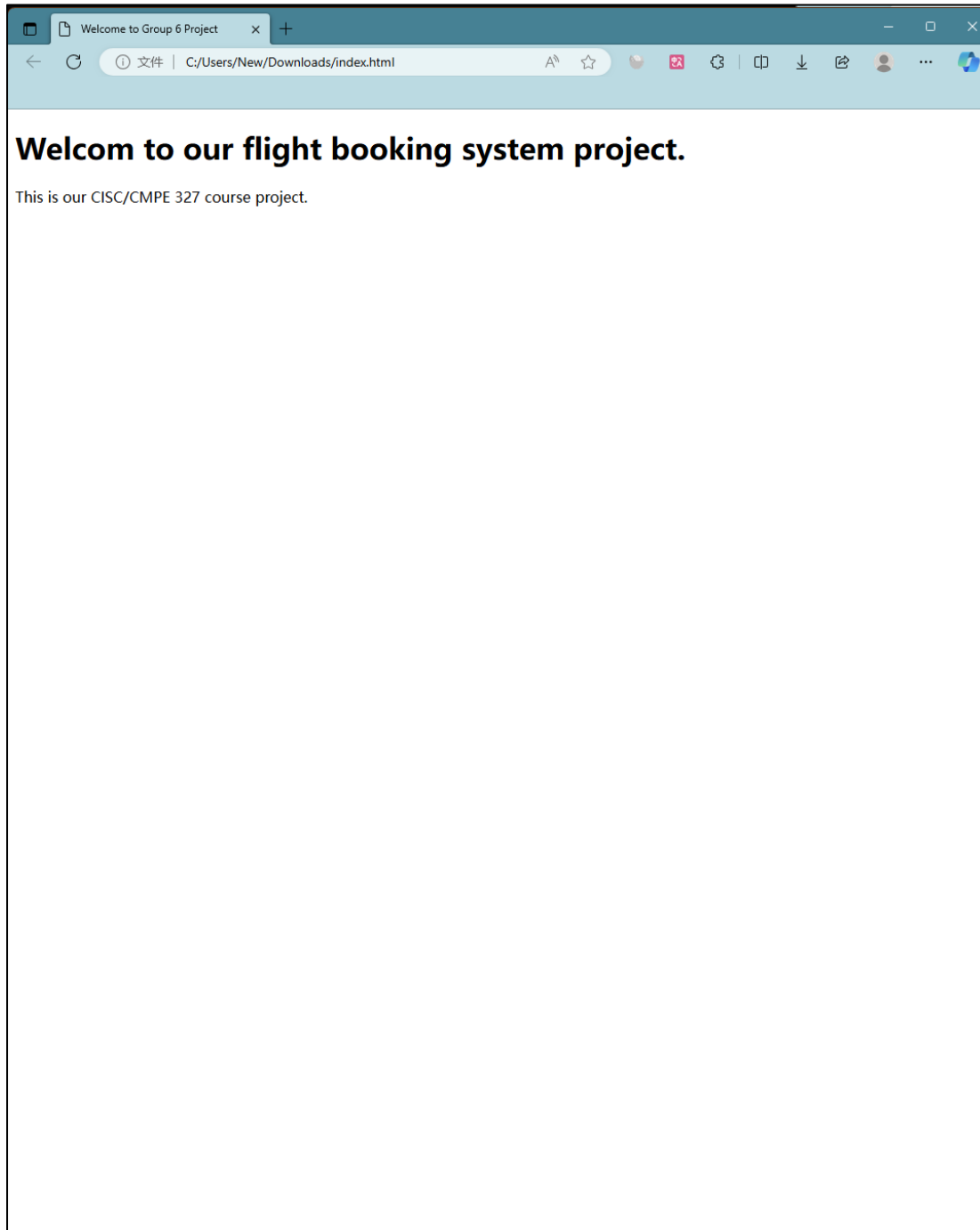


## Step 1: index.html



## Step 2: requirements list

- **Search Flights:** Users shall be able to search for available flights by entering the departure and destination cities along with the travel dates.

- **Select Seats:** Users shall be able to view available seats on selected flights and choose their preferred seats.
- **Booking Process:** Users shall be able to enter passenger details and proceed with booking.
- **Payment Gateway Integration:** Users shall be able to make payments through an integrated payment gateway.
- **Booking Confirmation:** After payment, users shall receive a booking confirmation, which includes ticket details and a unique booking reference number.
- **Booking History:** Users shall be able to view their past bookings and current reservations on their account dashboard.
- **User registration:** User should register first, then they can use the booking system
- **User Login:** User who already register can login to use booking system
- **Password Recovery:** User who forgot the password can reset account password.
- **Logout:** User who want to logout can logout and quit/login other account.

## Step 3: requirements

### 1. Test Case for Flight Search

- **Objective:** To verify that users can successfully search for flights based on their input.
- **Action:** Input departure city, destination city, and travel dates, then click on the "Search" button.
- **Assert:** Verify that the system returns a list of available flights matching the criteria.

## 2. Test Case for Select Seats

- **Objective:** To verify that users can view and select available seats on a chosen flight.
- **Action:** After selecting a flight, display the seating chart and choose an available seat by clicking on it.
- **Assert:** Verify that the selected seat is highlighted, and the selection is confirmed when the user proceeds to the next step.

## 3. Test Case for Booking Process

- **Objective:** To verify that users can enter passenger details and complete the booking process.
- **Action:** Input passenger details (name, age, passport number, etc.) on the booking form and click "Proceed to Payment".
- **Assert:** Verify that the system accepts the details and redirects the user to the payment gateway.

## 4. Test Case for Payment Gateway Integration

- **Objective:** To verify that users can make payments through the integrated payment gateway.
- **Action:** After entering passenger details, select a payment method (credit card, PayPal, etc.), input payment details, and click "Pay Now".
- **Assert:** Verify that the payment is processed, and the system redirects the user to the booking confirmation page.

## 5. Test Case for Booking Confirmation

- **Objective:** To verify that users receive a booking confirmation after successful payment.
- **Action:** Complete the payment process and check the booking confirmation screen for ticket details and a unique booking reference number.
- **Assert:** Verify that the booking confirmation displays the correct flight and passenger information, along with the booking reference number.

## 6. Test Case for User Registration

- **Objective:** To verify that users can register for a new account on the platform.
- **Action:** On the registration page, input the required information (name, email, password) and click "Register."
- **Assert:** Verify that the system creates the account and redirects the user to the login page with a success message.

## 7. Test Case for User Login

- **Objective:** To verify that registered users can log in to the system using their credentials.
- **Action:** On the login page, input the registered email and password, then click "Login."
- **Assert:** Verify that the system logs the user in and redirects them to the booking system's homepage or dashboard.

## 8. Test Case for Password Recovery

- **Objective:** To verify that users can reset their password if they forget it.

- **Action:** Click on "Forgot Password" on the login page, enter the registered email, and follow the instructions to reset the password.
- **Assert:** Verify that the system sends a password reset link to the user's email and allows the user to set a new password.

## 9. Test Case for Logout

- **Objective:** To verify that users can securely log out of their account.
- **Action:** Click on the "Logout" button from the user dashboard.
- **Assert:** Verify that the system logs the user out and redirects them to the login page or homepage.

## 10. Test Case for Booking History

- **Objective:** Verify that users can access their booking history.
- **Action:** Log in and navigate to the booking history page.
- **Assert:** Confirm that all past bookings are displayed with correct details.

## 11. Test Case for Passenger Information Addition

- **Objective:** To verify users can add multiple passenger details before booking.
- **Action:** In the booking process, click on the "+" button after passenger number and input additional passenger details (name, age, passport number, etc.).
- **Assert:** Verify that the system adds the passenger details to the booking and allows proceeding to the next step.

Figma URL: [https://www.figma.com/design/RDWfz7OuKiuGSB87vhXOnn/Flight-booking-wireframe-\(Community\)?node-id=40-302&t=2XoiG0Pk479cFh8V-1](https://www.figma.com/design/RDWfz7OuKiuGSB87vhXOnn/Flight-booking-wireframe-(Community)?node-id=40-302&t=2XoiG0Pk479cFh8V-1)