Course Project Assignment #1 - Front End Requirements

Group 5: Kevin Hu, Nicholas Wang, Safowan Mostaque

UI url:
https://www.figma.com/proto/k0TUjh1LLW8VAj1PW0bS5R/Figma-basics?node-id=1669-162202&t=bEkIZG4xFNkvtW2Z-1

Project #2: Flight Ticket Booking System.

Functional Requirements

1. **Flight Search**
   Users will be able to search for available flights by entering from, to, and dates of travel.
2. **Seat Selection**
   Users will be able to select their preferred seat from a seat map.
3. **Flight Details Display**
   The system will display the flight details, including departure time, arrival time, duration, and price.
4. **Payment Processing**
   Users will be able to enter payment details and make a payment for a selected flight.
5. **Booking Confirmation**
   After successful payment, users will be provided with a booking confirmation.
6. **Account Creation**
   Users will be able to create a new account.
7. **Log in**
   Users will be able to log into an existing account.

Test Cases

Requirement 1: Flight Search

- **Test Case Name**: Search flights form validation
- **Objective**: Verify that users can enter the origin, destination, and travel date, and then validate these inputs before search.
- **Action**: Leave one or more input fields blank, then click the search button.
- **Assert:** The system should display an error message indicating that all fields must be filled in.
- **Test Case Name**: Successful flight search
- **Objective**: Verify that the search button functions properly after valid inputs.
- **Action**: Enter all fields valid then click the search button.
- **Assert:** The system should transition to the flight results page.

Requirement 2: Seat Selection

- **Test Case Name**: Seat map display interaction
- **Objective**: Verify that the seat map is displayed correctly and users can select a seat.
- **Action**: On the seat selection page, click on an available seat to select it.
- **Assert:** The selected seat should visually change, and a confirmation indicator should appear on the page.
- **Test Case Name**: Prevent selection of unavailable seat
- **Objective**: Ensure users cannot select seats marked as unavailable.
- **Action**: Try to click on a seat marked as unavailable.
- **Assert:** The seat should not change its state, and no confirmation should appear.

Requirement 3: Flight Details Display

- **Test Case Name**: Flight details display
- **Objective**: Searching for a flight displays details such as departure time, arrival time, duration, price, luggage, passenger, and the seat selector.
- **Action**: Perform a flight search and view the flight results.
- **Assert:** The system displays the details of the selected flight, are able to enter in passenger details and move to the seat map page.
- **Test Case Name**: Passenger details verification
- **Objective**: Make sure the passengers details are valid and work with the ticket provided
- **Action**: Enter invalid passenger details
- **Assert:** The system displays an error message indicating the format is incorrect and does not move to the next page
- **Test Case Name**: Successful passenger details
- **Objective**: Make sure the passengers details with valid inputs pass
- **Action**: Enter valid passenger details
- **Assert:** The system displays a confirmation message and allows navigation to the next page

Requirement 4: Input Validation for Payment Form

- **Test Case Name**: Payment information validation
- **Objective**: Verify that the payment inputs are valid, examples being credit card number, expiration date, and CVV.
- **Action**: Enter invalid credit card information and click pay.
- **Assert:** The system displays an error message indicating the format is incorrect.
- **Test Case Name**: Successful payment
- **Objective**: Make sure that the payment with valid inputs will pass.
- **Action**: Enter valid credit card information and click pay.
- **Assert:** The system displays a confirmation message stating that the payment was successful

Requirement 5: Booking Confirmation Page Display

- **Test Case Name**: Display booking confirmation
- **Objective**: Verify that a confirmation page is displayed with the correct booking reference after a payment is completed.
- **Action**: Complete the booking process and proceed to the confirmation page.
- **Assert:** The system should display a booking reference number, flight details, and a "Thank you for your booking" message.

Requirement 6: User Registration UI

- **Test Case Name**: User registration form validation
- **Objective**: Registration checks for valid inputs.
- **Action**: Leave any of the fields blank or enter an invalid format.
- **Assert:** The system should display an error message.
- **Test Case Name**: Password strength validation
- **Objective**: Ensure that the system checks for password strength during registration.
- **Action**: Enter a weak password and submit the form.
- **Assert:** The system should display a message indicating that the password is too weak.
- **Test Case Name**: Successful registration submission
- **Objective**: Users can submit with valid input.
- **Action**: Enter valid names, email, phone number, password, then click sign up.
- **Assert:** The system should redirect the user to a login page.

Requirement 7: Login UI

- **Test Case Name**: Login form validation
- **Objective**: Validates user inputs before submission.
- **Action**: Enter an invalid email format and/or incorrect password, then click login
- **Assert:** The system should display an error message.
- **Test Case Name**: Successful login submission
- **Objective**: Make sure users are able to log in.
- **Action**: Enter a valid email and password, then click login
- **Assert:** The user is redirected to the searching page.