***All requirements below were formulated by all three group members.**

Figma Link: https://www.figma.com/design/TUrb6uVBoVXS5UzSBdiK4U/327-Assignment-1?node-id=0-1&t=nvvozDS7r2rbclxS-1

Requirement 1: Index Page To Authentication Redirect

Test Case: Redirect to Login

Objective:
To verify that the "Login" button on the `/index` page correctly redirects to the `/login` page.

Action:

1. Navigate to base app URL.
2. Click the "Login" button.
3. Wait to be redirected to the `/login` page.

Assert:

- Verify that the user is redirected to the `/login` page.
- Verify that the user login form is properly displayed.
- Verify that the "Login" button is properly displayed.

Test Case: Redirect to Register

Objective:
To verify that the "Sign up" button on the `/index` page correctly redirects to the `/sign-up` page.

Action:

1. Navigate to base app URL.
2. Click the "Sign up" button.
3. Wait to be redirected to the `/sign-up` page.

Assert:

- Verify that the user is redirected to the `/sign-up` page.
- Verify that the user registration form is properly displayed.
- Verify that the "Sign up" button is properly displayed.

Requirement 2: User Registration

Note A valid email is defined as the following:

- In the form of: {locale}@{domain}.{tld}.
- Maximum total length of 320, minimum total length of 6
- Maximum locale length of 64, minimum length of 1
- Maximum domain length of 191, minimum length of 1
- Maximum TLD length of 63, minimum length of 2

A valid password is defined as the following:

- Between 6-320 characters (inclusive.)

- Any character type.

Test Case: Invalid Email Form Error

Objective:
Verify form is being verified on and displaying correct form errors on the invalid input.

Action:

1. Navigate to base app URL, click the "Sign up" button.
2. Wait to be redirected to the `/sign-up` page.
3. Input an invalid (falls out of any part of the definition) email.
4. Input a valid password.
5. Press "Sign up" button.

Assert:

- Verify that the user is redirected to the `/sign-up` page.
- No form error present on the password field.
- An "Invalid email" form error is present on the email field.
- No redirect occurs.

Test Case: Password Too Short Form Error

Objective:
Verify form is being verified on and displaying correct form errors on the invalid input.

Action:

1. Navigate to base app URL, click the "Sign up" button.
2. Wait to be redirected to the `/sign-up` page.
3. Input a valid email.
4. Input an invalid (5 or fewer characters) password.
5. Press "Sign up" button.

Assert:

- Verify that the user is redirected to the `/sign-up` page.
- No form error present on the email field.
- A "Too short" form error is present on the password field.
- No redirect occurs.

Test Case: Password Too Long Form Error

Objective:
Verify form is being verified on and displaying correct form errors on the invalid input.

Action:

1. Navigate to base app URL, click the "Sign up" button.
2. Wait to be redirected to the `/sign-up` page.

3. Input a valid email.
4. Input an invalid (321 or more characters) password.
5. Press "Sign up" button.

Assert:

- Verify that the user is redirected to the `/sign-up` page.
- No form error present on the email field.
- A "Too long" form error is present on the password field.
- No redirect occurs.

Test Case: Redirect on Register

Objective:
Verify that you are redirected on a valid registration.

Action:

1. Navigate to base app URL, click the "Sign up" button.
2. Wait to be redirected to the `/sign-up` page.
3. Input a valid email.
4. Input a valid password.
5. Press "Sign up" button.
6. Wait to be redirected to the `/login` page.

Assert:

- Verify that the user is redirected to the `/sign-up` page.
- No form errors are present.
- Verify that the user is correctly redirected to the `/login` page.
- Verify that the user login form is properly displayed.
- Verify that the "Login" button is properly displayed.

Requirement 3: Flight Search

Note An available flight is defined as the following:

- Departure time is in the future
- Has available seats

Test Case: Flight Index Page

Objective:
Verify available flights are displayed on the flights index page.

Action:

1. Navigate to base app URL, click the "Flights" button.
2. Wait to be redirected to the `/flights` page.

Assert:

- Verify that the user is redirected to the `/flights` page.
- All flights displayed have at least one available seat in their total seat capacity.
- All flights displayed have a departure date at least one minute from the current time.
- All flights have an arrival time greater than the departure time.

Requirement 4: Flight View

Test Case: Flight View Page

Objective:
Verify correct information is displayed on a specific flight view page.

Action:

1. Navigate to base app URL, click the "Flights" button.
2. Wait to be redirected to the `/flights` page.
3. Click on a flight.
4. Wait to be redirected to a `/flights/{id}` page.

Assert:

- Verify that the user is redirected to the `/flights` page.
- Verify that the user is redirected to a `/flights/{id}` page.
- Seat availability is no less than what was displayed on the `/flights` page for this specific flight.
- Departure date matches what was displayed on the `/flights` page for this specific flight.
- Arrival date matches what was displayed on the `/flights` page for this specific flight.
- Verify that the "Purchase" button is displayed.

Requirement 5: Ticket Purchase Steps

Note A valid name is defined as the following:

- In the form of: `{firstname} {lastname}`.
- Maximum total length of 61, minimum total length of 3
- Maximum firstname length of 60, minimum length of 1
- Maximum lastname length of 60, minimum length of 1
- One space, and one space only, required.

Test Case: Seat Selection

Objective:
Verify the correct seats are displayed on the first step of the ticket purchase form.

Action:

1. Navigate to base app URL, click the "Flights" button.
2. Wait to be redirected to the `/flights` page.

3. Click on a flight.
4. Wait to be redirected to a `/flights/{id}` page.
5. Click the "Purchase" button.
6. Wait to be redirected to a `/flights/{id}/purchase?step=1` page.

Assert:

- Verify that the user is redirected to the `/flights/{id}/purchase?step=1` page.
- Seat availability is no less than what was displayed on the `/flights/{id}` page for this specific flight.
- Specific seat availability matches internal records.
- User is able to select a seat from a dropdown.
- Verify a "Select Seat" label is displayed near the seat select dropdown.
- Verify that the "Next" button is displayed.

Test Case: Too Short Firstname Personal Details Form

Objective:
Verify the correct error is displayed when entering a too short firstname.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (too short firstname) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify a "Firstname too short" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Too Short Lastname Personal Details Form

Objective:
Verify the correct error is displayed when entering a too short lastname.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (too short lastname) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify a "Lastname too short" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Too Long Firstname Personal Details Form

Objective:
Verify the correct error is displayed when entering a too long firstname.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (too long firstname) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify a "Firstname too long" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Too Long Lastname Personal Details Form

Objective:
Verify the correct error is displayed when entering a too long lastname.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (too long lastname) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify a "Lastname too long" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Missing Space Name Personal Details Form

Objective:
Verify the correct error is displayed when entering a name without a space.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (correct length but missing space) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify a "Lastname required" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Multiple Space Name Personal Details Form

Objective:
Verify the correct error is displayed when entering a name with multiple spaces.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input an invalid (correct length but multiple spaces) name.
5. Click the "Next" button.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- Verify an "Invalid name" error is displayed under the "Full name" input.
- No redirect occurs.

Test Case: Valid Personal Details Form

Objective:
Verify the correct redirection occurs when entering valid personal details.

Action:

1. Complete the "Seat Selection" test case steps.
2. Click the "Next" button.
3. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
4. Input a valid name.

5. Click the "Next" button.
6. Wait to be redirected to a Stripe hosted checkout session.

Assert:

- Verify the correct "Personal Details" form is displayed.
- Verify that the "Next" button is displayed.
- Verify that the "Previous" button is displayed.
- No form errors are present.
- Verify that the user is redirected to a Stripe hosted checkout session.

Requirement 6: Stripe Ticket Purchase Payment

Test Case: Redirect Checkout Session

Objective:
Verify the correct redirection to the Stripe hosted checkout session.

Action:

1. Navigate to base app URL, click the "Flights" button.
2. Wait to be redirected to the `/flights` page.
3. Click on a flight.
4. Wait to be redirected to a `/flights/{id}` page.
5. Click the "Purchase" button.
6. Wait to be redirected to a `/flights/{id}/purchase?step=1` page.
7. Complete seat selection.
8. Click the "Next" button.
9. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
10. Input a valid name.
11. Click the "Next" button.
12. Wait to be redirected to a Stripe hosted checkout session.

Assert:

- Verify that the user is redirected to the `/flights` page.
- Verify that the user is redirected to a `/flights/{id}` page.
- Verify that the user is redirected to a `/flights/{id}/purchase?step=1` page.
- Verify that the user is redirected to a `/flights/{id}/purchase?step=2` page.
- Verify that the user is redirected to a Stripe hosted checkout session.

Test Case: Checkout Session Return

Objective:
Verify the correct redirection after completing the Stripe checkout.

Action:

1. Navigate to base app URL, click the "Flights" button.

2. Wait to be redirected to the `/flights` page.
3. Click on a flight.
4. Wait to be redirected to a `/flights/{id}` page.
5. Click the "Purchase" button.
6. Wait to be redirected to a `/flights/{id}/purchase?step=1` page.
7. Complete seat selection.
8. Click the "Next" button.
9. Wait to be redirected to a `/flights/{id}/purchase?step=2` page.
10. Input a valid name.
11. Click the "Next" button.
12. Wait to be redirected to a Stripe hosted checkout session.
13. Complete Stripe form.
14. Wait to be redirected back to the `/flights` page.

Assert:

- Verify that the user is redirected to the `/flights` page.
- Verify that the user is redirected to a `/flights/{id}` page.
- Verify that the user is redirected to a `/flights/{id}/purchase?step=1` page.
- Verify that the user is redirected to a `/flights/{id}/purchase?step=2` page.
- Verify that the user is redirected to a Stripe hosted checkout session.
- Verify that the user is redirected back to the `/flights` page after completing the Stripe form.