Test plan for the American Airlines Flight Booking (Playwright Test Suite)

1. Test Plan Objective

- This test plan is designed to validate and automate the process of booking a one-way flight on the American Airlines website using Playwright. Each test step includes assertions, logging, and screenshot capture to ensure a reliable and auditable test execution process.

2. Scope of Testing

- **Website**: American Airlines booking page
- **Browser**: Chromium (or specified in `playwright.config.ts`)
- **Testing Tool**: Playwright
- **Test Scenarios**: Flight search and booking workflow from origin to destination on a specific date, capturing results at each significant step.

3. Assumptions and Preconditions

- The American Airlines website layout and identifiers remain consistent during the testing period.
- Date format is fixed as MM/DD/YYYY.
- No login or session authentication is required for this scenario.
- **Prerequisites**:
  - Node.js and Playwright are installed.
  - Playwright browsers are installed.
  - `.env` or `playwright.config.ts` files are correctly configured for headless or headed execution.

4. Test Environment

- **Operating System**: Windows/Mac/Linux
- **Browser**: Chromium (with configurations as per `playwright.config.ts`)
- **Tools and Frameworks**: Node.js, Playwright, TypeScript
- **Test Data**:
  - Origin: ATL
  - Destination: BUF
  - Departure Date: Pre-defined in MM/DD/YYYY format

5. Test Scenarios and Steps

5.1 Setup and Teardown

Objective: Prepare and clean up the testing environment before and after each test run.

- **Test Setup**:
    1. Launch the headless or headed Chromium browser as configured.
    2. Create a new browser context and a new page for a fresh session.
- **Test Teardown**:
    1. Close all browser contexts and browser instances after each test completion.
    2. Delete or archive trace files, screenshots, and reports as configured.

5.2 Detailed Test Case Breakdown

Each test step captures a screenshot for clarity and documentation.

Test Case 1: Launch the American Airlines Website

- **Description**: Verify that the website loads completely.
- **Steps**:
    1. Open the American Airlines website.
    2. Wait until the content is fully loaded.
    3. Take a screenshot to confirm successful load.
- **Expected Result**: The website loads fully, and the homepage is displayed.

Test Case 2: Select Flight Type

- **Description**: Set the flight type to “One way”.
- **Steps**:
    1. Locate the flight type selection option.
    2. Select the “One way” option.
    3. Validate that the option is selected.
    4. Capture a screenshot to confirm the selection.
- **Expected Result**: “One way” option is selected and visible.

Test Case 3: Enter Airports

- **Description**: Input the origin (ATL) and destination (BUF) airports.
- **Steps**:
    1. Input “ATL” as the origin airport.
    2. Log the input for validation.
    3. Capture a screenshot.
    4. Input “BUF” as the destination airport.
    5. Log the input for validation.
    6. Capture a screenshot.
- **Expected Result**: Origin and destination airports are entered correctly and logged.

Test Case 4: Select Departure Date

- **Description**: Set a specific departure date.
- **Steps**:
    1. Select a date input field.
    2. Enter the pre-defined departure date in MM/DD/YYYY format.
    3. Log confirmation of the date selection.
    4. Capture a screenshot of the date input field.
- **Expected Result**: The departure date is entered correctly.

Test Case 5: Submit Flight Search

- **Description**: Initiate the flight search.
- **Steps**:
    1. Check for the visibility of the search button.
    2. Click the search button.
    3. Log the status of the button.
    4. Capture a screenshot upon clicking.
- **Expected Result**: Flight search is initiated successfully.

Test Case 6: Validate Search Results and Capture Data

- **Description**: Verify that flight search results are displayed.
- **Steps**:
    1. Confirm that the flight results are displayed on the results page.
    2. Log the flight count from the UI and display the count in the console.
    3. Capture a screenshot of the results page.
- **Expected Result**: Flights are displayed with a count greater than zero.

Test Case 7: Select First Flight

- **Description**: Choose the first available flight.
- **Steps**:
    1. Locate the first flight option in the results.
    2. Select the flight option.
    3. Verify selection success.
    4. Capture a screenshot after the selection.
- **Expected Result**: First available flight option is selected successfully.

Test Case 8: Choose Fare Option

- **Description**: Select the “Basic Economy” fare.
- **Steps**:
    1. Locate the “Basic Economy” fare option.
    2. Select the option.
    3. Verify that the option is selected and visible.
    4. Capture a screenshot.
- **Expected Result**: “Basic Economy” fare option is selected successfully.

Test Case 9: Skip Upgrade

- **Description**: Skip the flight upgrade option.
- **Steps**:
    1. Locate the “No Upgrade” option.
    2. Click the option.
    3. Log the confirmation of the choice.
    4. Capture a screenshot.
- **Expected Result**: Upgrade option is skipped successfully.

Test Case 10: Verify Trip Summary

- **Description**: Ensure the trip summary is displayed with accurate details.
- **Steps**:
    1. Locate the trip summary section.
    2. Validate that it includes the correct details for the selected trip.
    3. Capture a final screenshot of the trip summary.

- **Expected Result**: Trip summary is displayed with accurate details, confirming the booking flow.

---

6. Test Reporting and Logging

- **Report Type**: HTML report generated in `test-results` directory.
- **Screenshot Log**: Each test step has associated screenshots saved in the designated screenshots folder.
- **Trace Files**: Enabled for every test run, allowing review and playback for debugging.

---

7. Execution Instructions

1. **Initialize and Run Tests**:
   - Ensure prerequisites are met.

Run commands:

```
npx tsc
npx playwright test tests/flight-reservation.spec.ts
```

2. **Review Results**:
   - View HTML report and screenshots in the `test-results` directory.
   - Use trace files for playback in case of failures.
