Technical Evaluation Instructions

Objective: Write a data-driven test using Playwright by driving the test scenarios from a JSON object. We've provided you with the necessary shell; you simply need to write the Playwright code.

Expected Length: 1 Hour

Submission Instructions:
Please publish your code repo to a public GitHub repository. Post-completion, provide a write-up for the "client" detailing:

Challenges and Solutions: Mention any obstacles encountered and your solutions.
Recommendations: Offer any suggestions for either the tested features or the testing process.

Please email both the GitHub link & write-up document to stacia@workwithloop.com🙂
Feel free to respond on LinkedIn as well so we know to expect your evaluation!

Resources:
YouTube on Playwright Data-Driven

Acceptance Criteria:

Setup and Preparation:

- Create a new project or workspace specifically for this task.
- Ensure Playwright and the necessary dependencies are installed and set up.

Login Automation:

- Automate the login process to Asana using the provided credentials.
  - https://app.asana.com/-/login
  - Email: ben+pose@workwithloop.com
  - Password: Password123
- The script should be capable of inputting the email address and password and submit the form successfully.

Data-Driven Testing Using JSON:

- Your tests should be driven from the `testCases` JSON object.

- Implement a mechanism to read test case data from this JSON object to drive your tests.
- Each test case will comprise of 3 test.step:
    - Navigation: Navigate to the correct `leftNav` item as mentioned in the JSON object.
    - Column Verification: Confirm that the specified `card_title` is present in the designated `column`.

JavaScript

```
const { test, expect } = require('@playwright/test');

const testCases = [
  {
    "id": 1,
    "name": "Test Case 1",
    "leftNav": "Cross-functional project plan, Project",
    "column": "To do",
    "card_title": "Draft project brief",
  },
  {
    "id": 2,
    "name": "Test Case 2",
    "leftNav": "Cross-functional project plan, Project",
    "column": "To do",
    "card_title": "Schedule kickoff meeting",
  },
  {
    "id": 3,
    "name": "Test Case 3",
    "leftNav": "Cross-functional project plan, Project",
    "column": "To do",
    "card_title": "Share timeline with teammates",
  },
  {
    "id": 4,
    "name": "Test Case 4",
    "leftNav": "Work Requests",
    "column": "New Requests",
    "card_title": "[Example] Laptop setup for new hire",
  },
  {
    "id": 5,
    "name": "Test Case 5",
```

```
    "leftNav": "Work Requests",
    "column": "In Progress",
    "card_title": "[Example] Password not working",
  },
  {
    "id": 6,
    "name": "Test Case 6",
    "leftNav": "Work Requests",
    "column": "Completed",
    "card_title": "[Example] New keycard for Daniela V",
  }
];

test.describe('Asana Data-Driven Tests', () => {
  testCases.forEach((data) => {
    test('', async ({ page }) => {
      await test.step('Login to Asana', async () => {
        // Login to Asana
      });

      await test.step('Navigate to the project page', async () => {
        // Navigate to the project page
      });

      await test.step('Verify the card is within the right column', async () =>
{
        // Verify the card is within the right column
      });

    });
  });
});
```

Expected Implementation Techniques:

Dynamic Locators: The test dynamically generates locators based on the current test case's data, such as data.leftNav, data.column, and data.card_title. This demonstrates Playwright's flexibility in creating locators at runtime.