**Name: Kaveri Reddy Veluri( Ph No: 573 730 4847)**

**Project setup:**

**Overview:**

This stage involves creating a new Node.js project and using Playwright, a potent browser automation tool, to automate testing.

**Steps Involved:**

1. Launch a fresh Node.js project:  
     
   A JavaScript runtime environment is Node.js. When a new project is started, its foundational files and structure are put up.

mkdir asana-automation # Create a new directory for your project

cd asana-automation # Navigate into the project directory

npm init -y # Initialize a new Node.js project with default settings (-y skips prompts)

Install Playwright and dependencies:

Playwright (@playwright/test) is a Node.js library for browser automation. We also install axios for making HTTP requests, which can be useful for fetching test data (optional in this case).

npm install @playwright/test axios

‘@playwright/test’ allows us to write and run tests using Playwright.

‘axios’ is used here to fetch test data from a JSON file (testCases.json).

Project Structure:

Create necessary folders and files for organizing your projec

test/: Directory to store test files.

asana.spec.js: This file will contain your Playwright tests.

testCases.json: JSON file where you define your test cases.

1. Implementing Login Automation

Overview:

Here, we write automated tests using Playwright to log in to Asana with provided credentials (ben+pose@workwithloop.com, Password123).

Detailed Steps:

Create asana.spec.js:

This file will contain our automated tests using Playwright's test framework (@playwright/test).

Explanation:

fetchTestCases(): Fetches test case data from testCases.json.

loginToAsana(page): Logs in to Asana using provided credentials.

navigateToProject(page, leftNav): Navigates to a specific project page within Asana.

verifyCardInColumn(page, columnName, cardTitle): Verifies if a card with cardTitle is present in the specified columnName column.

1. JSON Test Cases

Overview:

Define test cases in testCases.json that will drive the automated tests. Each test case specifies navigation details and expected card presence in Asana.

Detailed Steps:

Create testCases.json:

This file contains an array of test cases, each specifying leftNav, column, and card\_title

Each test case specifies:

* leftNav: Navigation path in Asana.
* column: Column name where the card should be verified.
* card\_title: Title of the card expected in the specified column.

1. Running Tests

Overview:

Add a script to package.json to execute the tests using the Playwright test runner.

Detailed Steps:

Update package.json:

Add a script named test that runs Playwright tests.

This script executes playwright test, which is a command provided by @playwright/test to run tests written using Playwright.

**Summary**

By following these steps, you have set up a structured Node.js project for automating tests using Playwright. You've created automated tests that log in to Asana, navigate to specific pages, and verify the presence of cards based on data-driven test cases defined in testCases.json.

* **Key Concepts Covered**:
  + Setting up a Node.js project (npm init).
  + Installing dependencies (npm install).
  + Writing automated tests using Playwright (@playwright/test).
  + Using JSON files for data-driven testing.
  + Running tests with Playwright (playwright test script in package.json).

This approach provides a foundation for automating browser interactions and performing structured testing in web applications like Asana using JavaScript and Playwright. Adjustments can be made to selectors and logic based on specific application requirements and UI changes.