**Rallly Test Automation Suite**

This repository contains a comprehensive test automation suite for the Rallly application, leveraging **Playwright**, **Cucumber**, and **JMeter** to ensure robust functional, acceptance, and performance testing. The suite is designed with best practices in mind, including the **Page Object Model (POM)** design pattern, **Gherkin-style acceptance tests**, and **data-driven performance testing**.

**1. Playwright Tests (Functional Testing)**

The Playwright tests are written using the **Page Object Model (POM)** design pattern, ensuring maintainability, reusability, and scalability. The tests are located in the rallly-test/tests directory.

**1.1 Page Objects**

The page objects are stored in rallly-test/tests/pages and include:

* home.page.js: Contains locators and methods for the home page.
* create-poll.page.js: Contains locators and methods for the poll creation page.

**1.2 Test Cases**

The test file rallly-test/tests/create-poll.test.js contains **5 test cases** to validate the poll creation functionality:

1. **Create Poll Page Tests › should successfully create a poll with all fields filled**
2. **Create Poll Page Tests › should successfully create a poll with title and date only (location is optional)**
3. **Create Poll Page Tests › should successfully create a poll with title and date only (description is optional)**
4. **Create Poll Page Tests › should display error when title is not filled**
5. **Create Poll Page Tests › should display error when date is not filled**

**1.3 Running Playwright Tests**

To execute the Playwright tests, run the following command:

npx playwright test tests/create-poll.test.js

**2. Cucumber Tests (Acceptance Testing)**

Cucumber is implemented on top of the Playwright tests to expose **Gherkin-style acceptance tests**, making them easily accessible to non-technical stakeholders. The feature files and step definitions are located in the rallly-test/features directory.

**2.1 Feature File**

The feature file rallly-test/features/create-poll.feature is written in **Gherkin syntax** and describes the poll creation functionality in a human-readable format.

**2.2 Step Definitions**

The step definitions for the feature file are located in rallly-test/features/step-definitions/step-definitions.js

**2.3 Running Cucumber Tests**

To execute the Cucumber tests and generate a JSON report, run the following commands:

1. Run the tests:

npx cucumber-js --format json:report/cucumber\_report.json

1. Generate the HTML report:

node generate-report.js

**3. JMeter Tests (Performance Testing)**

JMeter is used to perform **data-driven performance testing**, ensuring the application meets non-functional requirements. The JMeter test plan consumes an external data source, adhering to best practices for scalability and maintainability.

**3.1 Running JMeter Tests**

To execute the JMeter performance tests, follow these steps:

1. **Clean Up**: Delete the /result and /report folders (if they exist) from rallly-test/jmeter to ensure a clean slate.
2. **Run the Test**:

jmeter -n -t /rallly-test/jmeter/performance-test.jmx -l /rallly-test/jmeter/result/results.jtl

1. **Generate the Report**:

jmeter -g /rallly-test/jmeter/result/results.jtl -o /rallly-test/jmeter/report

**4. Directory Structure**

rallly-test/

├── tests/

│ ├── pages/

│ │ ├── home.page.js

│ │ └── create-poll.page.js

│ └── create-poll.test.js

├── features/

│ ├── create-poll.feature

│ └── step-definitions/

│ └── step-definitions.js

├── jmeter/

│ ├── performance-test.jmx

│ ├── result/

│ └── report/

└── report/

└── cucumber\_report.json

**5. Prerequisites**

* **Node.js**: Ensure Node.js is installed to run Playwright and Cucumber tests.
* **Java and JMeter**: Ensure Java is installed to run JMeter