**Testing Engine**

**Building the Testing Engine Framework Integrate with Playwright**

**Epic** -1: As a user I want create the UI for Testing Engine.

**Epic** -2: As a user I want to add the Start Testing button.

**Epic** -3: As a user, I want to configure Playwright in the testing engine framework.

**Epic** -4: As a user, I want to integrate Playwright with the testing engine

**Epic** -5: As a tester, I want to define reusable test scripts using Playwright

**Epic** -6: As a user I want to add the report for complete the execute the resting.

**Epic** -7: As a tester, I want to integrate Playwright with CI/CD pipelines via the testing engine

**Epic -1: As a user I want create the UI for Testing Engine.**

**Description:**  
Create an intuitive and user-friendly UI for the Testing Engine that allows users to interact seamlessly with its features. The UI should include essential functionalities such as test case creation, test suite management, execution scheduling, and result visualization. Ensure the design aligns with the application's overall theme and supports responsive behavior for various devices. Include placeholders for integrating additional features such as Playwright integration and dashboard views.

**User Story:**

**US- 1:** As a user, I want to create the core UI components

* Implement the navigation system, including a main navigation bar and side menu.
* Create the Test Case Creation page with input fields for name, description, priority, and tags.

**US- 2:** As a user, I want to create and manage multiple test suites with configurable scheduling ptions

* Build the Test Suite Management page for listing, editing, and grouping test cases.
* Add a Test Execution Scheduler page to configure execution time, browser, and environment settings.

**US- 3:** As a user, I want a dedicated page where I can easily view the outcomes of my completed tests

* Design and develop a dashboard to display test results with graphs, logs, and status Indicators.
* Ensure placeholders for Playwright integration are added where required.

**US- 4:** As a user, I want a comprehensive final testing process, a fully responsive interface, and a clearly structured report UI

* Test the UI for responsiveness on mobile, tablet, and desktop devices.
* Perform user testing to gather feedback and make necessary improvements.
* Document the UI design, user flow, and provide onboarding materials.

**Epic -2: As a user I want to add the Start Testing button.**

**Description:**Implement a Start Testing button in the Testing Engine UI that enables users to initiate test execution seamlessly. The button should be prominently displayed on the Test Execution Scheduler page or relevant sections, ensuring easy access. Upon clicking the button, it should trigger the test execution process by integrating with the backend or testing engine API. The button must provide real-time feedback, such as a loading spinner or progress indicator, to enhance user experience. Additionally, it should handle errors gracefully, displaying appropriate messages and options for retrying or canceling failed attempts.

**User Story:**

**US 1: Design and Place the Start Testing Button**

* Create a visually appealing button aligned with the UI theme.
* Place the button on the Test Execution Scheduler page or relevant area.

**US 2: Integrate Button with Backend**

* Connect the button to the backend service or testing engine API.
* Trigger test execution workflows upon clicking the button.

**US 3: Add Feedback Mechanism**

* Implement a loading spinner or progress bar to indicate that tests are running.
* Display a success message when tests are initiated successfully.

**US 4: Handle Error Scenarios**

* Provide error messages if the test execution fails or encounters an issue.
* Include retry and cancel options for failed attempts.

**US 5: Test and Validate Functionality**

* Test the button for proper functionality, responsiveness, and usability.
* Validate that it works across different browsers and devices