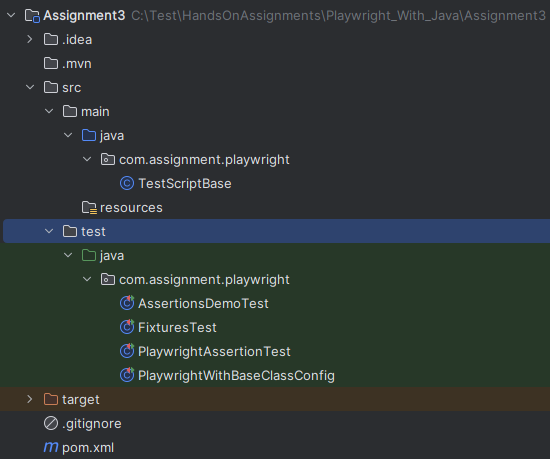
# Playwright for Java - Hands-On Assignments 3

# Project Structure



## **Problem Statement 1: Using Playwright Fixtures for Browser and Page Setup**

**Objective:**  
 Practice using Playwright JUnit 5 fixtures to inject Page and Browser objects for cleaner, reusable, and isolated test cases.

### **Steps:**

1. Use the @UsePlaywright annotation to enable Playwright’s built-in JUnit 5 fixture support.
2. Create two separate tests in the same class:

**Test 1: Using the Page Fixture**

* 1. Navigate to the URL: <https://example.com>
  2. Assert that the page title equals "Example Domain".

**Test 2: Using the Browser Fixture**

* 1. Create a new browser context from the injected Browser instance.
  2. Set the locale to "en-US" and the base URL to <https://example.com>.
  3. Open a new page from the context.
  4. Navigate to the base URL using page.navigate("").
  5. Print the page title.

### **Expected Output:**

* **Test 1:**

Test passes if the title equals "Example Domain"

* **Test 2:**

Printed page title: Example Domain

## **Problem Statement 2: Using JUnit and Playwright Assertions**

**Objective:**  
 Practice combining traditional JUnit assertions with Playwright’s built-in assertions for validating page content and element states.

### **Steps:**

1. Use the @UsePlaywright annotation to enable Playwright’s JUnit 5 fixture support.
2. Create a test method that:
   1. Navigates to <https://practicetestautomation.com/practice-test-login/>
   2. Fills in:
      1. **Username:** leave blank
      2. **Password:** leave blank
   3. Clicks the **Submit** button.
3. Locate the feedback message element with the selector .show.  
    *(This is the error message displayed when you attempt login without filling credentials.)*
4. Perform the following assertions:
   1. **JUnit assertions:**
      1. Assert that the page title is "Test Login | Practice Test Automation".
      2. Assert that the feedback element is visible.
   2. **Playwright assertions:**
      1. Assert that the page title is not "abcd".
      2. Assert that the feedback element is visible.

### **Expected Output:**

JUnit Assertions passed: Correct title and feedback visible.

Playwright Assertions passed: Title not 'abcd' and feedback visible.

## **Problem Statement 3: Extending a Test Base Class with Browser Context Options**

**Objective:**  
 Learn how to centralize Playwright setup/teardown in a base class while adding enhanced browser context configuration for all tests.

### **Steps:**

1. Create a base class TestScriptBase that:
   1. Starts Playwright and launches Chromium with:
      1. headless(false)
      2. slowMo(500) for visible step-by-step execution.
   2. Creates a new page before each test using browser.newContext() with:
      1. Viewport size: 1280×800
      2. Locale: "en-US"
      3. Base URL: <https://example.com>
   3. Closes the page after each test.
   4. Shuts down Playwright after all tests.
2. Create a subclass PlaywrightWithBaseClassConfig with **two test methods**:
   1. **Test 1:** Navigate to the base URL (empty string with page.navigate("")) and print the page title.
   2. **Test 2:** Navigate to /more (or any valid path under example.com) and print the page URL.

### **Expected Output:**

* **Test 1:**

Test 1: Example Domain

* **Test 2:**

Test 2: https://example.com/more

### **Problem Statement 4: Validating HTML Form Submission with Mixed Assertions**

**Objective:**  
 Use both JUnit and Playwright assertions to test a basic HTML form and validate its behavior when values are submitted.

**Steps:**

1. Use the @UsePlaywright annotation to enable Playwright’s JUnit 5 fixture support.
2. Create a test method that:
   1. Navigates to:  
       <https://testpages.herokuapp.com/styled/basic-html-form-test.html>
   2. Fills the Username input field using the selector:  
       input[name='username']
   3. Clicks the Submit button using role-based locator.
   4. Locates the feedback message with the text:  
       "No Value for username"
3. Perform the following **assertions**:
   1. **JUnit Assertions**:
      1. Assert that the page title is "Processed Form Details"
      2. Assert that the feedback message is **not visible**
   2. **Playwright Assertions**:
      1. Assert that the page title is **not** "abcd"
      2. Assert that the feedback message is **not visible**

### **Expected Output:**

JUnit Assertions passed: Page title verified, feedback message not visible.

Playwright Assertions passed: Title is not 'abcd', feedback message not visible.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*