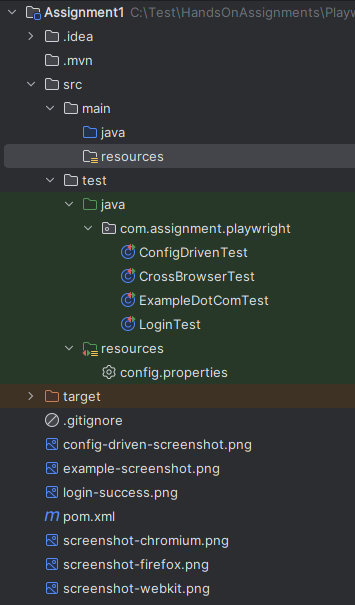
# Playwright for Java - Hands-On Assignments 1

# Project Structure



## Problem Statement 1: Set Up and Run First Playwright Script

Objective: Ensure the setup of Playwright correctly and execute a basic script.

Steps:

* Set up a Maven Playwright Java project.
* Install browser binaries.
* Create a test class - ExampleDotComTest .java - using JUnit5.
* Launch a Chromium browser, go to https://example.com, and take a screenshot.

Deliverable: A runnable Playwright test class and screenshot saved locally.

## Problem Statement 2: Write a Cross-Browser Test

Objective: Practice using multiple browser types with Playwright.

Steps:

* Modify the previous script to run on Chromium, Firefox, and WebKit. (CrossBrowserTest.java)
* Navigate to https://playwright.dev/ and assert the page title.
* Take a screenshot in each browser and store them with unique names.

Deliverable: A script showing browser loop execution and 3 screenshots.

## Problem Statement 3: Debugging & Troubleshooting Setup

Objective: Simulate troubleshooting common setup issues.

Steps:

* Edit the provided broken pom.xml (e.g., missing dependencies).
* Identify and fix the issues.

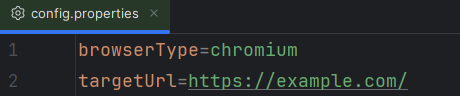
Deliverable: Corrected pom.xml and documented steps of the fix.

## Problem Statement 4: Parameterize URL and Browser Type

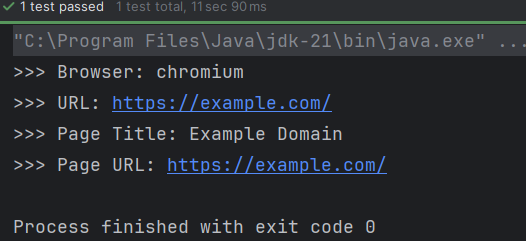
Objective: Introduce config-driven test execution.

Steps:

* Create a test script – ConfigDrivenTest.java
* Accept browserType and targetUrl( as parameters from the config file.



* Launch the browser accordingly and validate title and URL of the site.
* Print results in the console.



Deliverable: A config-enabled script with command-line or properties file support.

## Problem Statement 5: Create a Login Test Script

Objective: Practice form interaction with Playwright.

Steps:

* Create a test script – LoginTest.java
* Go to https://practicetestautomation.com/practice-test-login/.
* Fill in username (student) and password (Password123).
* Click 'Submit'.
* Assert the login success message or URL change. Take the page screenshot on success.

Deliverable: A working script with login verification and screenshot of the logged-in page.

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