**Lab 1: Lab 2: Simple Web Automation Test**

Objective:*Write a basic Selenium test to open a browser and navigate to a website*

Tasks:

1. Create a simple Selenium script.

2. Open a browser and navigate to AWS's official website.

3. Capture a screenshot of the webpage.

Documentation:

- Basics of Selenium.

- Opening webpages using Selenium.

- Taking screenshots.

Prerequisites:

1- An AWS account with administrative access.

2- Python Automation Course

3- Python Selenium Course

4- Bash Script Deep Dive Course

5- Previous Lab completed

Implementation Documentation:

**Basics of Selenium**

Selenium is a powerful tool for automating web browsers. It provides a way for developers to write scripts in various programming languages that can interact with web elements, fill out forms, click buttons, and perform other web-related tasks automatically.

**Step 1: Create a Simple Selenium Script in pycharm**

You can write Selenium scripts in Python, Java, C#, and other supported programming languages. In this lab, we'll use Python in Pycharm we setup in last lab.

| # Import the Selenium webdriver and time from selenium import webdriver Import time  # Step 2: Initialize the WebDriver (Choose your browser) # For example, if you're using Chrome: driver = webdriver.Chrome()  # Step 3: Perform actions on the webpage # Example: Open the AWS website driver.get("https://aws.amazon.com/")  # Step 4: Capture a Screenshot # You can use the save\_screenshot() method to capture a screenshot driver.save\_screenshot("aws\_website.png") time.sleep(5)  # Step 5: Close the browser driver.quit() |
| --- |

**STEP 2: Running the Script**

Choose the **current file** from the top bar and then click the **Run** button to run the script.

**Explanation of above Selenium Script**

In the script above, we use webdriver.Chrome() to initialize a Chrome browser driver. You can use webdriver.Firefox() for Firefox, and similar methods for other browsers.

**driver.get("URL")** is used to open a webpage.

Taking Screenshots

Selenium provides the **save\_screenshot("FileName.png")** method to capture a screenshot of the current webpage.

The screenshot will be saved in the current working directory with the provided filename.

**Conclusion**

By following these steps, you can create a basic Selenium script to open a browser, navigate to a website (in this case, AWS's official website), and capture a screenshot of the webpage. This lab is a fundamental building block for more complex web automation tasks.