***Automating Facebook Login with JUnit***

Githublink- https://github.com/Utkarsh9991/Testing.git

***Introduction:***

In the landscape of software testing, the imperative role of automated testing in ensuring application reliability and functionality cannot be overstated.

JUnit, a distinguished testing framework for Java, is instrumental in streamlining the creation and execution of test cases.

***Test Script Overview:***

The exemplar at hand revolves around the automation of the Facebook login process through the adept use of JUnit.

Selenium WebDriver serves as the cornerstone for seamlessly interacting with the intricate web elements constituting the Facebook login page.

***Test Scenario:***

Commencing with the establishment of a ChromeDriver, meticulous attention is given to configuring the test environment.

Navigational prowess is then applied to gracefully traverse to the Facebook login page, ensuring a calibrated starting point.

Programmatic entry of a username and password, facilitated by Selenium WebDriver, sets the stage for subsequent actions.

The orchestration of a click on the login button propels the sequence forward.

Validation of a successful login transpires through a judicious inspection of the displayed user's name.

***WebDriver Initialization:***

The test class initializes a WebDriver instance for Chrome in the setUp() method using new ChromeDriver().

***Facebook Login Test:***

The testFacebookLogin() method performs the actual Facebook login automation.

It maximizes the browser window, navigates to the Facebook login page, and identifies the username and password input fields using XPath.

User cedentials (email and password) are then programmatically entered into the respective input fields.

After a brief pause (Thread.sleep(2000)) to allow for visibility, the login button is located and clicked.

The script then locates an element containing the user's name and retrieves the displayed text.

The actual and expected text are compared using JUnit's assertEquals() to verify a successful login.

Tear Down:

The tearDown() method is responsible for closing the browser via driver.quit() after each test, ensuring a clean test environment.

***POM.xml Dependencies:***

The pom.xml file manages project dependencies through Maven.

JUnit (version 5.9.3), Selenium (version 4.11.0), and WebDriverManager (version 4.4.3) are specified as dependencies.

These dependencies are crucial for effective test execution, providing the necessary tools for testing and automating browser interactions.

***Summary:***

The code exemplifies a basic automation script using Selenium WebDriver and JUnit for Facebook login. It follows best practices by utilizing JUnit assertions for verification and incorporates WebDriverManager to handle the WebDriver setup seamlessly. This script can serve as a foundation for more comprehensive testing suites and highlights the importance of automation in ensuring the robustness of web applications.