**Web Automation with Selenium**

**Summary Report**

**Introduction:**

In this assignment, I worked on building and automating web applications using **Selenium WebDriver**, **TestNG**, **Apache POI**, and **Jenkins**. The main objectives were to develop automation scripts for different scenarios, implement data-driven testing, and integrate the project into a CI/CD pipeline using Jenkins.

**Implementation Overview:**

* **Basic Automation**: Created a Selenium script to open a browser, navigate to Wikipedia, print the page title, and interact with the search bar using multiple locator strategies (ID, Name, XPath, CSS). Implemented dropdown selection and applied both implicit and explicit waits.
* **Advanced Automation**: Automated interactions on the DEMOQA website, including checkboxes, radio buttons, handling alerts, and switching between windows and tabs.
* **Test Automation with TestNG**: Configured pom.xml to include TestNG, created tests with @BeforeMethod, @Test, and @AfterMethod, and generated HTML reports. Used Apache POI to read test data from an Excel file for filling out the DEMOQA Practice Form. Created a CommonMethods utility class for handling Excel data. Added supporting resources (Excel test data and image file) to enable automated form submissions.
* **Data Driven Test**: Performed data-driven testing with multiple iterations in Jenkins.
* **Jenkins Integration**: Set up Jenkins to pull the project from GitHub and automatically execute TestNG tests.

**Challenges and Solutions:**

* **Locator Issues**: Some elements required XPath instead of ID due to dynamic attributes. I solved this by combining different locator strategies.
* **Synchronization**: Elements sometimes loaded slowly; I overcame this by using **explicit waits (WebDriverWait)** for reliable execution. Since the website is a free app with advertisements, I had to add a Thread.sleep() at one point to manually handle an ad, as explicit wait could not be applied due to limitations.
* **Apache POI**: Initially I had trouble understanding Apache POI and how to read data from an excel file.
* **Jenkins Integration**: Configuring GitHub webhook was tricky; I resolved it by installing the correct Jenkins plugins and configuring maven correctly. Also learned that pom.xml file path is important during configuration.

**Key Learnings:**

* Gained hands-on experience with Selenium locators, waits, and browser interactions.
* Learned how to structure test automation using TestNG annotations.
* Gained practical knowledge of integrating external data sources for testing with Apache POI.
* Learned how to integrate automation into a CI/CD pipeline using Jenkins and GitHub webhooks.

**Screenshots:**

**Basic Automation:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Advanced Automation:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Test Automation:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**TestNG HTML Report:**

**A screenshot of a computer error

AI-generated content may be incorrect.**

**Jenkins Test:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Conclusion:**

This assignment helped me to test web automation from writing test scripts and organizing tests with TestNG to running them in a CI/CD pipeline using Jenkins. The process gave me confidence in building real-world automation frameworks and highlighted the importance of maintainable test structures and integration with CI tools.

**GitHub Repository Link:**

https://github.com/Komaldewan11/Module7\_Web\_Automation