Module 3: Selenium WebDriver

Trainer Name: Ankush Vankore

Total Sessions: 19

* Selenium WebDriver
  + Automation Concepts
  + Components of Selenium
  + Introduction to Selenium WebDriver
  + Configuration of Selenium WebDriver
  + Basic methods of WebDriver
    - get()
    - getTitle()
    - getCurrentUrl()
    - getPageSource()
    - close()
  + Locators
    - Name
    - Id
    - ClassName
    - CssSelector
    - XPath
    - LinkText
    - PartialLinkText
    - TagName
    - Relative Locator
  + Handling control
    - Text Box
    - Button
    - Links
    - Checkbox
    - Radio Button
    - List box
    - Drop down List
  + Synchronization
    - Thread.Sleep()
    - ImplicitWait
    - ExplicitWait
    - FluentWait
    - PageLoadTimeout
  + Handle a table
  + Handle alert
  + Handling multiple windows
  + Handling File Upload
  + Robot Class
  + Actions class (For mouse actions)
    - Right click
    - Left click
    - Double click
    - Drag and drop
    - Hover the mouse
  + JavascriptExecutor
    - Scroll the page
    - Click on some control
  + Handling menus
  + Screenshot
* TestNG
  + Introduction
  + Configuration
  + Executing single test
  + Executing multiple tests
  + Annotations
    - @Test
    - @BeforeTest
    - @AfterTest
    - @BeforeMethod
    - @AfterMethod
    - @DataProvider
    - @Parameters
    - @Listeners
  + Data Driven Testing via @DataProvider
  + Modular Framework
    - Executing / skipping single / multiple test
    - Executing single / multiple classes
    - Executing / skipping groups
  + Keyword driven framework
  + Page Object Model (POM)
    - Utility class
    - Client class
  + Data Driven Framework
    - Read the data from Excel file
    - Write the data to Excel file
  + Hybrid Framework
  + Extent Report
* Introduction to Maven
  + Introduction
  + Configuration
  + Adding dependencies
  + Executing the script
  + Architecture
* Introduction to BDD (Cucumber)
  + Introduction
  + Configuration
  + Creating Feature file with help of keywords
    - Feature
    - Scenario
    - Given
    - When
    - Then
    - And
    - Background
    - Scenario Outline
    - Examples
  + Step Definition
  + Runner class
  + Reports in Cucumber
  + Tags in cucumber
  + Hooks in Cucumber
  + Data driven framework using Cucumber

Revision on Java

* Basics
  + Variables
  + Data Types
  + Checking the conditions
    - If
    - If-else
    - Nested if else
    - Switch case
  + Loops
    - For loop
    - Enhanced for loop (for each loop)
  + Functions
    - Writing the functions
    - Calling the functions
  + Arrays
    - 2 – D Array
* OOPS
  + Creating class
  + Creating object of class
  + Static members and static methods
  + Constructor
* Inheritance
  + **Interface**
* Exception Handling
  + throws keyword
* Collections
  + List
  + Set
  + Map

Software Testing

Process of checking the **C**orrectness, **C**ompleteness, **S**ecurity and **Q**uality of developed software application.

Manual Testing

* Enter some data in text box
* Click on button
* Selecting the value from drop down list, list box, check box, radio button etc.
* Navigation from one page to another
* Marking the test case as pass or fail

Here tester uses his/her hand-eye-brain co-ordination

**Automation testing** is performing all above actions via **automation testing tool.**

Every testing tool is a software.

Every tool understand some specific programming language like Java, C# .net, JavaScript, Python, Ruby etc.

**Advantages of Automation**

* Time saving
* Accuracy
* Reduces human error
* Cost Efficiency
* 100% test coverage
* Regression testing

**When to Automate?**

* Not in one time testing
* Large and complex projects
* Stable requirements
* Large amount of data to be tested
* Performance testing, Security testing
* Compatibility Testing

**Types of Automation tools**

* Unit Testing
  + Junit
  + Nunit
* GUI Testing / Functionality
  + Selenium
  + Tosca
  + Appium
  + QTP
* API Testing
  + Postman
  + RestFULL API

STLC

Requirement Analysis 🡪 Planning 🡪 Design 🡪 Implementation 🡪 Execution 🡪 Closure

**Process of Automation**

1. Planning
2. Selection of Tool
   1. Type of application
   2. Cost of the tool
   3. Support availability
   4. Testers availability
   5. Automated Reports
3. Creating/Writing the Script
4. Generate a data
5. Execute the script
6. Generate a Report
7. Maintance

**Selenium**

It is bundle / suite of test automation tools to test web based application. (Web Sites)

**Components of Selenium**

1. Selenium IDE (Record and playback)
2. Selenium Grid – Parallel Execution
3. Selenium RC
4. Selenium WebDriver

**Selenium WebDriver**

* WebDriver is a test automation tool for testing web based applications (Web Sites)
* It is an API
* It is an Interface in java.

**Create 2 Folders**

1. YourName\_SeleniumDemos (For storing all the demos)
2. Selenium Jar Files

**Pre-Requisite for Selenium WebDriver**

* Minimum Windows 10
* JDK should be installed on the system
  + Minimum 11 version
* Editor
  + Eclipse
  + Idea Intellij
* Any one updated browser
* Selenium Jar File

Download Selenium Jar File

* Launch <https://www.selenium.dev/>
* Click on Downloads link
* Click on Latest stable version [4.33.0](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.33.0/selenium-server-4.33.0.jar)
* Cut / Copy paste this file to 2nd folder (Selenium Jar Files)

Eclipse Configuration for Selenium WebDriver

1. Launch Eclipse
2. Click on Browse on Launch Workspace window
3. Select the 1st folder (YourName\_SeleniumDemos)
4. Click on Launch
5. Click on File Menu 🡪 New 🡪 Java Project
   1. Select Java version grater than or equal to JavaSE 11
   2. Uncheck **Create module-info.java file** checkbox
   3. Click on Finish
6. Create a Package
7. Create a class in the above package
8. Right click on the Project 🡪 Build Path 🡪 Configure Build Path
9. Go to Libraries Tab
10. Click on ClassPath
11. Click on Add External JARs… button
12. Go to the second folder which you created and select selenium-server-4.33.0 file
13. Click on Open
14. Click on Apply and Close

Methods of WebDriver

1. Launch a Browser 🡪 Create an object of WebDriver interface
2. get() 🡪 Launch the website. You have to pass the URL.