Experiment No: 7

AIM: a) To setup an Eclipse IDE and Run Selenium Tests Using Maven.

b) To Setup and Run Selenium Tests in Jenkins Using Maven.

THEORY: Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

//Introduction to Selenium and TestNG

Add the description here.

a) To setup an Eclipse IDE and Run Selenium Tests Using Maven.

Steps for Writing Selenium Test Cases in Eclipse using Maven:

1. Download and install any latest version of Eclipse IDE from https://www.eclipse.org/downloads/
2. Create your first maven project.
3. Add a new java AppTest class in your \src\test\java. Make sure to that filename should have a keyword 'Test'.
4. Add the TestNG dependency under dependencies tag in pom.xml <dependencies> visit https://mvnrepository.com/artifact/org.testng/testng/7.4.0 to copy the foll. dependency tag.

```
<dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>7.4.0</version>
        <scope>test</scope>
</dependency>
```

5. Write a sample test case using TestNG to print "Hello world" by annotating @Test and left click on red cross when shown as error to import package import org.testng.annotations.Test and similarly write some valid / suitable test cases.

```
public class TestHelloworld {
        @Test
        public void testhelloworld()
        {
                System.out.println("Hello world test 1 ");
        }
}
```

6. Save and Run as Maven test.
7. Verify if the build is successful or not in the console output.
8. Now add Selenium dependency under dependencies tag in your pom.xml. Visit https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver to copy the foll dependency tag.

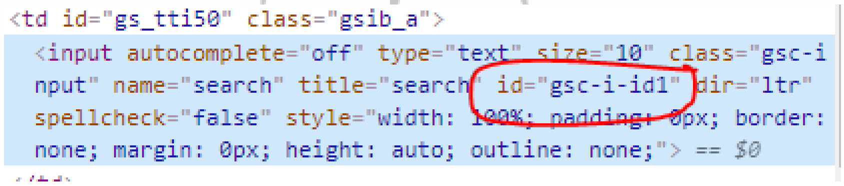
```
<dependency>
   <groupId>org.seleniumhq.selenium</groupId>
   <artifactId>selenium-chrome-driver</artifactId>
```

```
            <version>4.0.0-alpha-6</version>
        </dependency>
```

9. Download a web driver for chrome similar to the version of your chrome browser and extract chrome driver on some suitable location in your drive. https://sites.google.com/a/chromium.org/chromedriver/downloads. *Ensure that it should be of same version as that of your Chrome browser. To Check chrome version => Go in Help - > About Google chrome*
10. Add foll. test case using selenium chrome web drivers : Under this test, we will automate the following scenarios:
    a. Invoke Google Chrome browser.
    b. Open URL: http://www.javatpoint.com
    c. Click on the Search text box.
    d. Type the value "javatpoint tutorials"
    e. Click on the Search button.

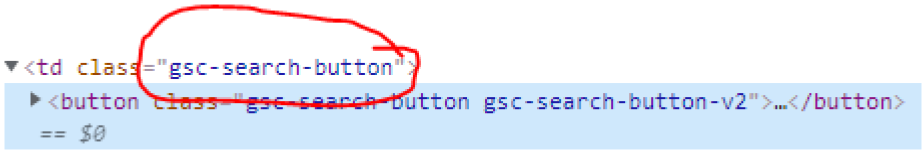
```
@Test
public void testhelloworld2()
{
System.setProperty("webdriver.chrome.driver","C:\\Users\\app\\Downloads\\chromedriver\\chromedriver.exe" );
    // Instantiate a ChromeDriver class.
      WebDriver driver=new ChromeDriver();

     // Launch Website
      driver.navigate().to("http://www.javatpoint.com/");

       //Maximize the browser
        driver.manage().window().maximize();

       //Scroll down the webpage by 5000 pixels
      JavascriptExecutor js = (JavascriptExecutor)driver;
      js.executeScript("scrollBy(0, 5000)");

    // Click on the search text box and send value
      driver.findElement(By.id("gsc-i-id1")).sendKeys("core Java");

      // Click on the search button
      driver.findElement(By.className("gsc-search-button")).click();
    }
```

11. To Find the id of search input box ,visit the given url http://www.javatpoint.com and right click on search box and select inspect element to find a html below like this:



```
<td id="gs_tti50" class="gsib_a">
  <input autocomplete="off" type="text" size="10" class="gsc-i
  nput" name="search" title="search" id="gsc-i-id1" dir="ltr"
  spellcheck="false" style="width: 100%; padding: 0px; border:
  none; margin: 0px; height: auto; outline: none;"> == $0
```

12. Similarly Find the class / id of search button



```
▼<td class="gsc-search-button">
  ▶<button class="gsc-search-button gsc-search-button-v2">…</button>
   == $0
```

13. Run as Maven test and observe the output.

b) To Setup and Run Selenium Tests in Jenkins Using Maven

1. Create a maven job as Maven Project
2. In Configure section **-** >go to **Source Code Management** -> Git - > paste this repo url =>https://github.com/bushsk/SimpleMavenProject.git (Assuming our maven project is on Github repository)
   Note: Ensure that the GitHub Plugin is installed at this point. If it has not been installed, then do install it from Manage Jenkins-> Manage Plugins
3. If your project is on local machine then go to Build section -> Root POM -> C:\Users\app\git\Eclipse_TestNG_Project\firstmaven\pom.xml Write the goals and options as => **test or clean test**. Make sure Root POM is set to pom.xml. -> save and apply
4. Go to build now -> console output and verify whether build as successful.

Conclusion: - Hence we have successfully created and ran Selenium Tests in Jenkins Using Maven.