****

**AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH**

**Faculty of Science and Technology (FST)**

**SOFTWARE QUALITY AND TESTING [E]**

**2021-2022, Summer**

***Final-Term***

***Assignment-01***

**Submitted To:**

**COURSE TEACHER:** ABHIJIT BHOWMIK

Associate Professor & Special Assistant [OSA], Computer Science

**Group Members:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| 1. Saha, Arnab | 19-39836-1 |

*Table of Contents*

[**Part-I: Installation Process of Selenium Webdriver for Web Application Automation Testing.** 3](#_Toc108517398)

[**Flowchart of Installation Process** 3](#_Toc108517399)

[Visual Representation and Description of Each Step of the Flowchart 4](#_Toc108517400)

[Download and Install Java Development Kit (JDK) 4](#_Toc108517401)

[Configure Environment Variables for JDK 6](#_Toc108517402)

[Download and Install Eclipse IDE for Java Developers 9](#_Toc108517403)

[Download Selenium 12](#_Toc108517404)

[Download Chromedriver for the Chrome Browser 13](#_Toc108517405)

[Configure Selenium on the Eclipse 14](#_Toc108517406)

[**Part-II: Test a Web Application Module Using Selenium Webdriver.** 21](#_Toc108517407)

[**Flowchart of Testing Process** 21](#_Toc108517408)

[Visual Representation and Description of Each Step of the Flowchart 22](#_Toc108517409)

[Create a New Project in Eclipse IDE 22](#_Toc108517410)

[Select a Web Application Module Which Will be Tested 22](#_Toc108517411)

[Prepare a Test Case 22](#_Toc108517412)

[Write Required Code 23](#_Toc108517413)

[Run and Observe the Outcomes 26](#_Toc108517414)

# **About Selenium:** Selenium is basically used to automate the testing across various web browsers. It supports various browsers like Chrome, Mozilla, Firefox, Safari, and IE, and we can very easily automate browser testing across these browsers using Selenium WebDriver.

# **Part-I: Installation Process of Selenium Webdriver for Web Application Automation Testing.**

## **Flowchart of Installation Process**

Download and Install Java Development Kit (JDK)

Configure Environment Variables for JDK

Download and Install Eclipse IDE for Java Developers

Download Selenium

Download Chromedriver for the Chrome Browser

Configure Selenium on the Eclipse

### Visual Representation and Description of Each Step of the Flowchart

### Download and Install Java Development Kit (JDK)

1. For downloading the Java Development Kit (JDK) you need to click on that link. This is only for windows users. For macOS and Linux there are also available links on the website.

(<https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe>)

1. Then the required file will start downloading.

|  |
| --- |
|  |

1. After successful downloading open the file.

|  |
| --- |
|  |

1. After open this file we can able to see this interface and click on the **Next** option.

|  |
| --- |
|  |

1. Again, click on the Next option.

|  |
| --- |
|  |

1. Wait some moment for further process.

|  |
| --- |
|  |

1. After all processes click on **Close** option.

|  |
| --- |
|  |

1. So far, our installation process is finished.

### Configure Environment Variables for JDK

1. After installation we can find our required Java file in this location.

|  |
| --- |
|  |

1. Then we need to go to **Settings → System → About** and Click on **Advanced system settings** option.

|  |
| --- |
|  |

1. Then we can see this interface and click on the **Environment Variables** option.

|  |
| --- |
|  |

1. Click on **New** option.

|  |
| --- |
|  |

1. Then set the Variable Name as JAVA\_HOME or what you want. And paste location path of the installed JDK folder. (C:\Program Files\Java\jdk-18.0.1.1). Then click on **OK** in all the tabs.

|  |
| --- |
|  |

1. Now we check that our JDK environment variable properly set or not using the Command Prompt. Open the Command Prompt and write **java -version**. If we can see this kind of text after execute the command then our process is successful.

|  |
| --- |
|  |

### Download and Install Eclipse IDE for Java Developers

1. For downloading the Eclipse IDE for Java Developers, you need to click on that link. This is only for windows users. For macOS and Linux there are also available links on the website.

(<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2022-06/R/eclipse-inst-jre-win64.exe>)

1. Then the required file will start downloading.

|  |
| --- |
|  |

1. After successful downloading open the file.

|  |
| --- |
|  |

1. After that we can see this tab and click on the first option.

|  |
| --- |
|  |

1. Click on the **Install** option.

|  |
| --- |
|  |

1. Wait some moment for further process.

|  |
| --- |
|  |

1. Click on the **LAUNCH** option.

|  |
| --- |
|  |

1. Again, select the **Launch** option here to open the Eclipse IDE.

|  |
| --- |
|  |

1. After that, we can able to see this window. That’s mean our Eclipse IDE is ready for work and our configuration procedure finished.

|  |
| --- |
|  |

### Download Selenium

1. For downloading the Selenium Webdriver you need to click on that link.

(<https://www.selenium.dev/downloads/>). Select **Latest stable version 4.3.0 option**. Required file start downloading.

|  |
| --- |
|  |

### Download Chromedriver for the Chrome Browser

1. For downloading the Chromedriver you need to click on this link.

(<https://chromedriver.chromium.org/downloads>). Then find the version which will match with your Chrome Browser latest version.

|  |
| --- |
|  |

1. According to my Chrome Browser version I click on this option. Sometimes, the version is exactly matched, or if not then you click on the nearest below version.

|  |
| --- |
|  |

1. Then you can see this page. Click on the link which is according to your Operating System. I click on **chromedriver\_win32.zip**. Because I use Windows Operating System. Then the required file will start downloading.

|  |
| --- |
|  |

**Note:** If you use different browser then you need to download the driver according to your browser.

### Configure Selenium on the Eclipse

1. For our working convenience, we will keep the Selenium and Chromedriver file in the same folder which we already downloaded in our previous steps. In that case, I created a folder as Selenium in **C Drive**. Then keep those files inside that folder.

|  |
| --- |
|  |

1. Now we run the Eclipse IDE for configure the Selenium.

|  |
| --- |
|  |

1. For create a new java project. Click on **File → New → Java Project**.

|  |
| --- |
|  |

1. Then this tab will appear. Give the **Project Name** as your choice. Then click on that dropdown option and select **JavaSE-1.8** and click on **Finish**.

|  |
| --- |
|  |

1. Then expand the Project option which we recently created.

|  |
| --- |
|  |

1. Then right click on the project name. Click on **Bulid Path → Configure Build Path** option.

|  |
| --- |
|  |

1. Then this tab will appear to us. Then go to **Libraries → Add External JARs…** option.

|  |
| --- |
|  |

1. Then go to that path here we stored our Selenium file then select the file, click on **Open** then click on **Apply and Close**.

|  |
| --- |
|  |

1. Now right click on **src** option then click on **New → Package**.

|  |
| --- |
|  |

1. Then this tab will appear. Give a name and click on **Finish**.

|  |
| --- |
|  |

1. After that, click on **Test\_Script** option then **New → Class**.

|  |
| --- |
|  |

1. Give a name of **Package section**. Then **check the public static void main(String[]args)** option. Click on **Finish**.

|  |
| --- |
|  |

1. Ultimately this window will visible to us. That’s mean our Selenium configuration on Eclipse IDE is done.

|  |
| --- |
|  |

# **Part-II: Test a Web Application Module Using Selenium Webdriver.**

## **Flowchart of Testing Process**

Run and Observe the Outcomes

Write Required Code

Prepare a Test Case

Select a Web Application Module Which Will be Tested

Create a New Project in Eclipse IDE

### Visual Representation and Description of Each Step of the Flowchart

### Create a New Project in Eclipse IDE

1. In **Part-I**, **Configure Selenium on the Eclipse** section’s (ii-xiii) points already explained this procedure. Kindly look over that section then you can easily create a new project in Eclipse IDE.

### Select a Web Application Module Which Will be Tested

1. For testing purposes, I select a web application **“Modern Medical Management System”** that I did in my **“Web Technology”** course. I test the **Login module** of that application.

|  |
| --- |
|  |

### Prepare a Test Case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: Modern Medical Management System | | | Test Designed by: Mr. Kamal | |
| Test Case ID: 01 | | | **Test Designed date:** 11. July. 2022 | |
| Test Priority (Low, Medium, High): Medium | | | **Test Executed by:** Arnab Saha | |
| Module Name: Login Session | | | **Test Execution date:** 12. July. 2022 | |
| Test Title: Verify login with valid username and password | | |  | |
| Description: Test the website login page | | |  | |
| Precondition: User has valid username and password  Dependencies: If any | | | | |
| Test Steps | **Test Data** | **Expected Results** | **Actual Results** | **Status (Pass/Fail)** |
| 1.Go to the site  2.Enter username  3.Enter password  4.Click submit | Username:  John Kabir  Password:  John1234@ | User should login  Into the application | As expected | Pass |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | |

### Write Required Code

1. In that stage, we write the required code for test execution. Each line of the Code's properly explained with comments.
2. This is the code for login module automation testing.

package Test\_Script;

//import required packages

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Test {

public static void main(String[] args) {

// Set the path of Executable browser driver

System.setProperty("webdriver.chrome.driver",

"C:\\Selenium\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver(); //Parent p = new child (WebDriver is parent and Chrome driver is child))

driver.get("http://localhost/Web%20Tech%20Project%20Doctor/login2.php"); //opening url

driver.findElement(By.id("doctor")).sendKeys("Doctor");

//login module's doctor category field id

driver.findElement(By.id("name")).sendKeys("John Kabir");

//login module's username field id

driver.findElement(By.id("password")).sendKeys("john1234@");

//login module's password field id

driver.findElement(By.xpath("/html/body/div/div/div/form/div/input[4]")).click();

//login module's submit button's xpath

System.out.println("Test Successful"); //Print that text after successsful code execution

driver.quit(); //closing browser

}

}

1. Code in Eclipse IDE.

|  |
| --- |
|  |

**Note:** In code’s 18, 20, 22, 24 lines. Id and xpath come from webpage source code. To get that **id** and **xpath** we need to follow the below steps.

1. First, we need to go to the web page. Then select the field which id we needed (Category field, Doctor option) and right-click on that field. Go to Inspect then the respective field code block appears to us and we can easily find the id and copy that. **Same process goes for username and password field.**

|  |
| --- |
|  |

1. This is for username field.

|  |
| --- |
|  |

1. This is for password field.

|  |
| --- |
|  |

1. For submit button xpath. Right click on **Submit** button then go to **Inspect**. Then **code block** appears then **right click on that code block** then go to **Copy and Copy XPath.**

|  |
| --- |
|  |

By following these steps, we can easily find out the **id** and **xpath**.

### Run and Observe the Outcomes

1. After code written click on the run button.

|  |
| --- |
|  |

1. For very short time login page appear to us. Username, password field will fill-up with respective username and password which defined in testcase and code.

|  |
| --- |
|  |

1. Lastly submit button automatically clicked and successfully login into the website.

|  |
| --- |
|  |

1. In the **Console** section of Eclipse IDE **“Test Successful”** text will be appeared. That's mean our testing process is successful.

|  |
| --- |
|  |