

Loyola Academy

Old Alwal, Secunderabad-500010

(An Autonomous Degree College Affiliated to Osmania University)

Accredited by NAAC with 'A' Grade (Third Cycle)

A College with Potential for Excellence by UGC



Department of BSc. Computer Science and Cognitive Systems

Practical Record

In

Software Testing Lab

Certificate

This is to certify that this is the bonafide record of the work done in Software Testing during the practical lab for the Fifth Semester of the academic year 2022-23

Name :

Class :

UID :

Internal

Principal

External

INDEX

S No.	Topic	Page No.	Signature
1	Configuring a Project in Selenium		
2	Test data in a Flat File		
3	Program to Login into a webpage/ Program a test case based on controls		
4	Test Case to Verify Student Grade		
5	Test case to select the number of students who have scored more than 60 in a subject		
6	Write and test a program to get the number of list items in a combo box		
7	Test a HTML file		
8	Test addition of 2 values in a program		

Configuring a Project with Selenium

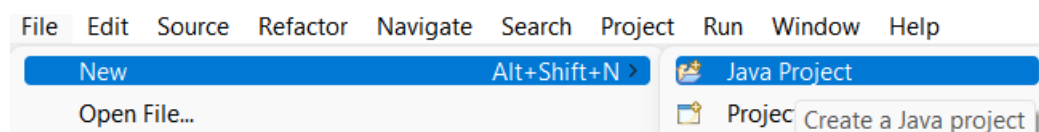
Prerequisites:

Download the Java JAR files for Selenium from www.selenium.dev/downloads

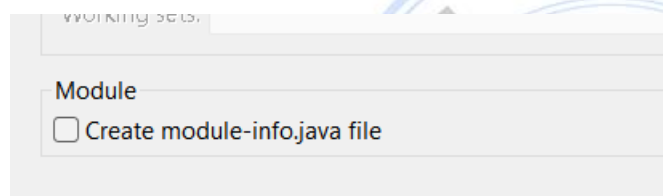
Steps:

Step 1: Open Eclipse IDE.

Step 2: Create a new Java Project (File -> New -> Java Project)



Enter the name,



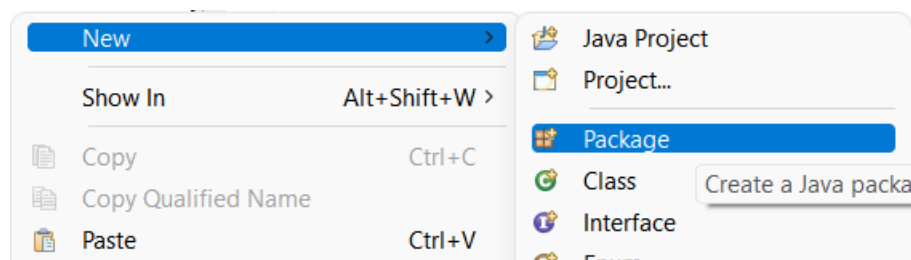
Uncheck '*Create module-info.java file*' and

Click on Finish.

Step 3: Now, from the package explorer, expand the newly created project.

The package explorer is available in (window>show view> package explorer)

Right click on the '*src*' folder and select '*New -> Package*'

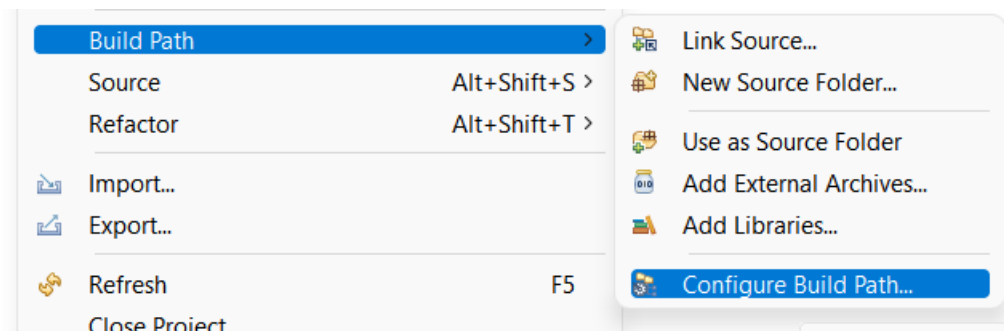


Step 4: Enter the package name and click on Finish.

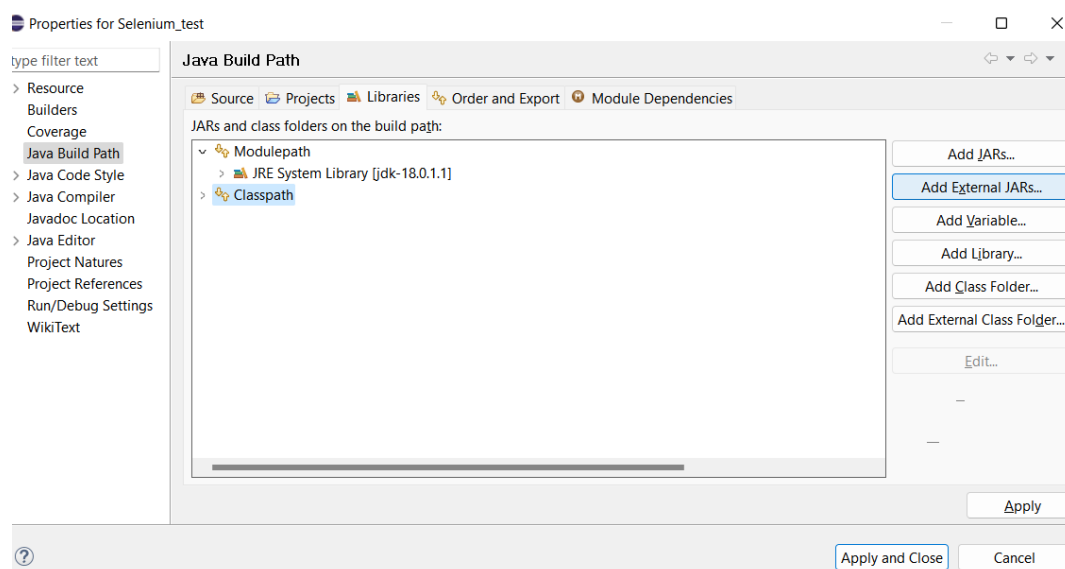
Step 5: Right click on the newly created package and select '*New -> Class*'

Step 6: Enter the class name and click on finish.

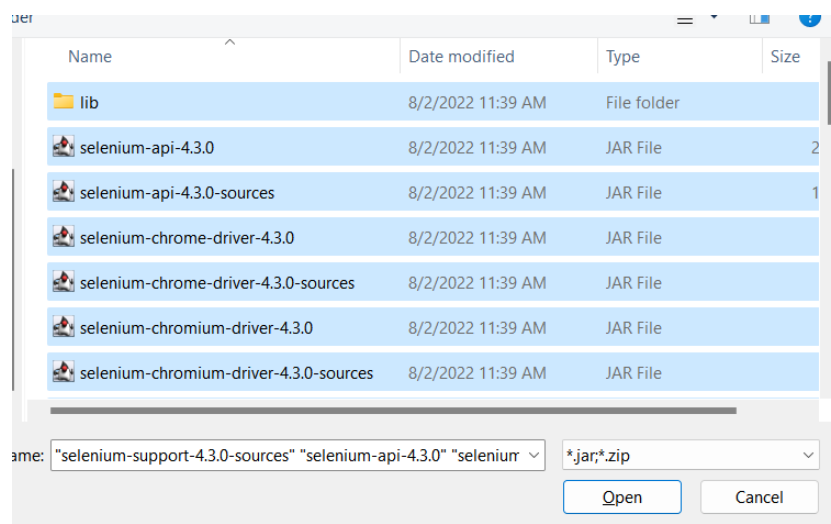
Step 7: Now, right-click on the project and select 'Build Path -> Configure Build Path', select the 'Libraries' tab.



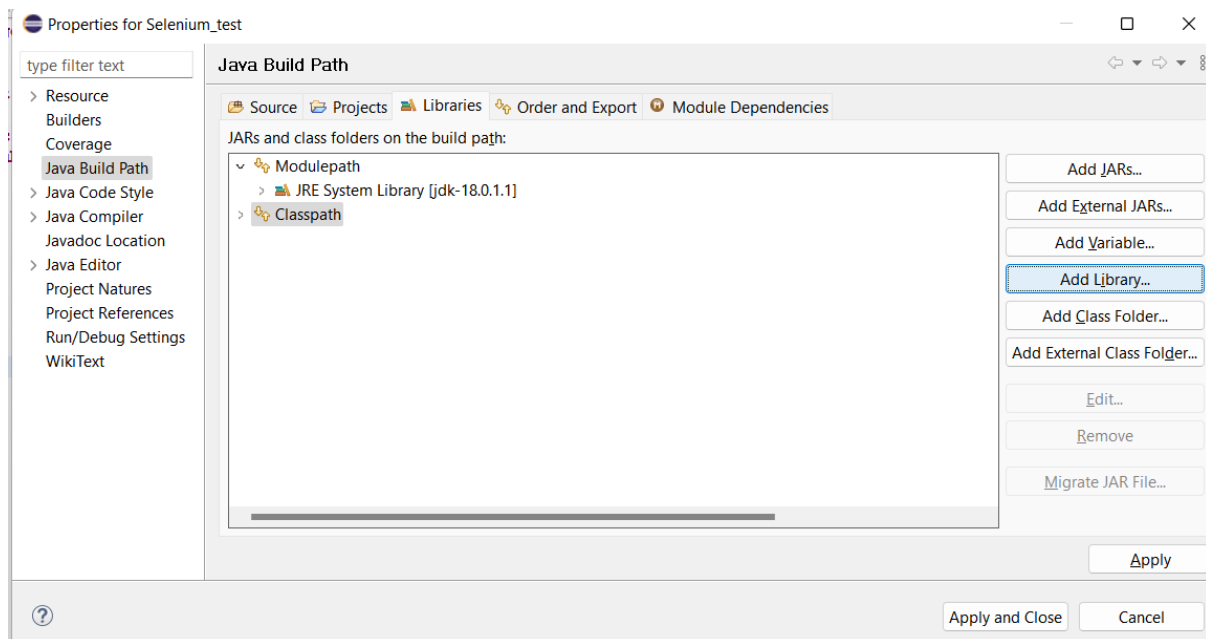
Step 8: Click on 'Classpath' and from the options available on the right, select



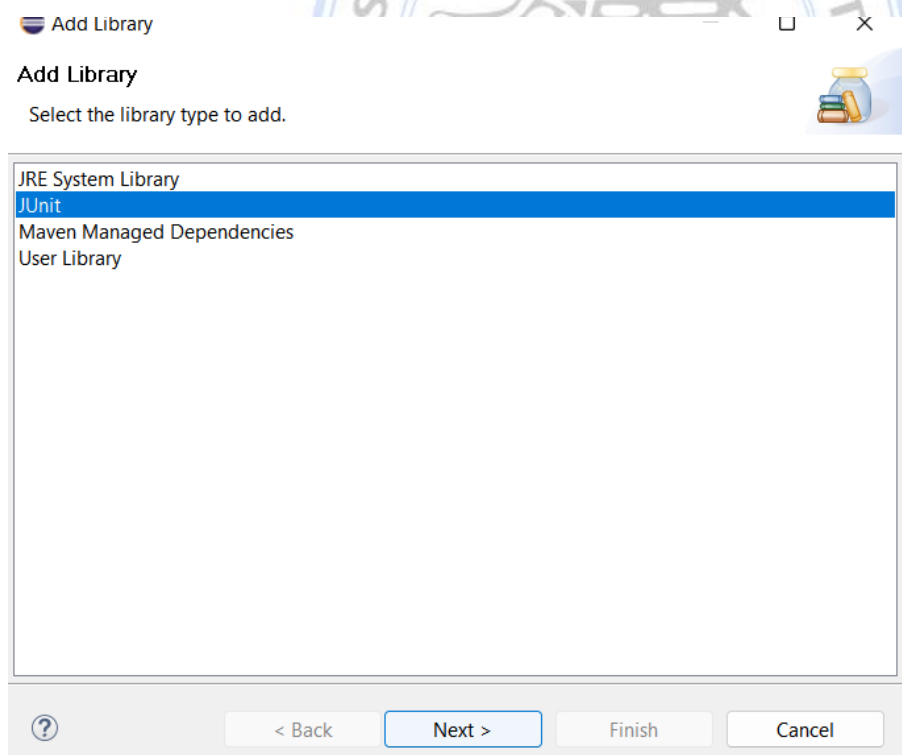
Step 9: 'Add External JARs', then browse to the selenium JARs folder and upload all the JAR files. (Even the ones in 'lib' folder)



Step 10: After that, again select '*Classpath*' and then select '*Add Library*'.



Step 11: Select '*JUnit*', select next and select the JUnit version and click Finish.w



The project is configured to work with Selenium.

Program to display a Flat File

Prerequisites:

Pre-configuration of Selenium in Eclipse

A CSV file to display (can be downloaded from Kaggle.com)

Steps:

Step 1: Open Eclipse IDE.

Step 2: Create a new Java Project and configure it to work with Selenium.

Step 3: Double-click on the class file to open it up.

In the class, type in the following code:

```
package practicals;

import java.io.*;
import java.util.Scanner;

public class FlatFiletest {
    public static void main(String args[]) throws IOException {
        Scanner sc = new Scanner(new File("C:\\\\testFile.csv"));
        sc.useDelimiter(",");
        while(sc.hasNext()) {
            System.out.println(sc.next());
        }
        sc.close();
    }
}
```

Things to replace:

- Replace (*C:\\\\testFile.csv*) with the path of the CSV file.

The program should look like:

```

Flatfiletest.java x
1 package practicals;
2
3 import java.io.*;
4
5
6 public class Flatfiletest {
7     public static void main(String args[]) throws IOException {
8         Scanner sc = new Scanner(new File("D:\\testFile.csv"));
9         sc.useDelimiter(",");
10        while(sc.hasNext()) {
11            System.out.println(sc.next());
12        }
13        sc.close();
14    }
15 }
16
17

```

Note: If any warnings occur, click on them and import the required packages.

Step 4: Clean by selecting Project -> Clean

Step 5: Run the program by clicking on Run -> Run (or) the Run icon.

OUTPUT

```

<terminated> FFtest [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin'
Name
Age
John
18
Jason
19
Jonathan
21

```

Program to Login to a Webpage/ Program a test case based on Controls

Prerequisites:

An account on the website in which we going to login.

Steps:

(We are going to login into Adactin Hotel App's page in the steps below)

Step 1: Open Eclipse IDE.

Step 2: Create a new Java Project and configure it to work with Selenium.

Step 3: Double click on the class file to open it up.

In the class, type in the following code:

```
@Test
public void test() {
    //Locating the WebDriver

    System.setProperty("webdriver.chrome.driver", "(chromedriver
_path)");

    //Creating WebDriver Object
    ChromeDriver driver = new ChromeDriver();

    //Opening a page
    driver.get("https://adactinhotelapp.com");

    //Finding the username and password elements using
IDs

    WebElement u = driver.findElement(By.id("username"));
    u.sendKeys("(your_username)");
```



```

WebElement p = driver.findElement(By.id("password"));
p.sendKeys("(your_password)");

//Clicking the login button

driver.findElement(By.id("login")).click();
}

```

Things to replace:

- Replace (*chromedriver_path*) with the path of ChromeDriver on your system.
- Replace (*your_username*) with your username on the website.
- Replace (*your_password*) with your password on the website.

The program should look like:



```

1 package newpack;
2
3 import org.junit.Test;
4
5 public class LoginTest {
6
7     @Test
8     public void test() {
9         //Locating the WebDriver
10        System.setProperty("webdriver.chrome.driver", "E:\\Subjects\\Sem 5\\ST\\Selenium Projects\\chromedriver.exe");
11        //Creating WebDriver Object
12        ChromeDriver driver = new ChromeDriver();
13
14        //Opening a page
15        driver.get("https://adactinhotelapp.com");
16
17        //Finding the username and password elements using IDs
18
19        WebElement u = driver.findElement(By.id("username"));
20        u.sendKeys("AshSmith");
21
22        WebElement p = driver.findElement(By.id("password"));
23        p.sendKeys("wow123");
24
25        //Clicking the login button
26
27        driver.findElement(By.id("login")).click();
28    }
29 }
30
31
32
33

```

Note: If any warnings occur, click on them and import the required packages.

Step 4: Clean by selecting Project -> Clean

Step 5: Run the program by clicking on Run -> Run (or) the Run icon.

Test case to verify student grade

Prerequisites:

Selenium IDE must be downloaded on the system.

Steps:

Step 1: Open Notepad (or) any Text Editor.

Step 2: Type in HTML code to verify grade.

Code:

```
<<!DOCTYPE html>
<html lang="en">
<head>
    <title>Practical 1</title>
</head>
<body>
    <form>
        <label><h2>Enter Mark : </h2></label>

        <input type = "text" id = "markInput">
        <hr>
        <label><h2>Grade Point : </h2></label>
        <button type = "button" onClick = "getResult()">Get
Result</button>
        <input type = "text" id = "resultLabel">

    </form>
    <script>
        function getResult(){
            var mark;
            var gradePoint = "";
            mark =
parseInt(document.getElementById("markInput").value);
```

```

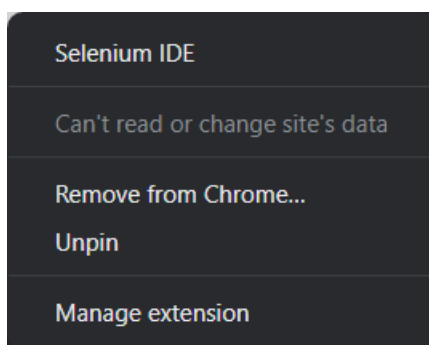
        if(mark >= 60 ){
            gradePoint = "First Division";
        }
        else if(mark >= 40 ){
            gradePoint = "Second Division";
        }
        else if(mark >= 33){
            gradePoint = "Third Division";
        }
        else{
            gradePoint = "Fail";
        }
        console.log("student grade : " + gradePoint);
        document.getElementById("resultLabel").value = gradePoint;
    }
</script>
</body>
</html>

```

Step 3: Save the file with .html extension

Step 4: Open Chrome where the selenium IDE is installed.

Right click on the selenium extension> manage extension



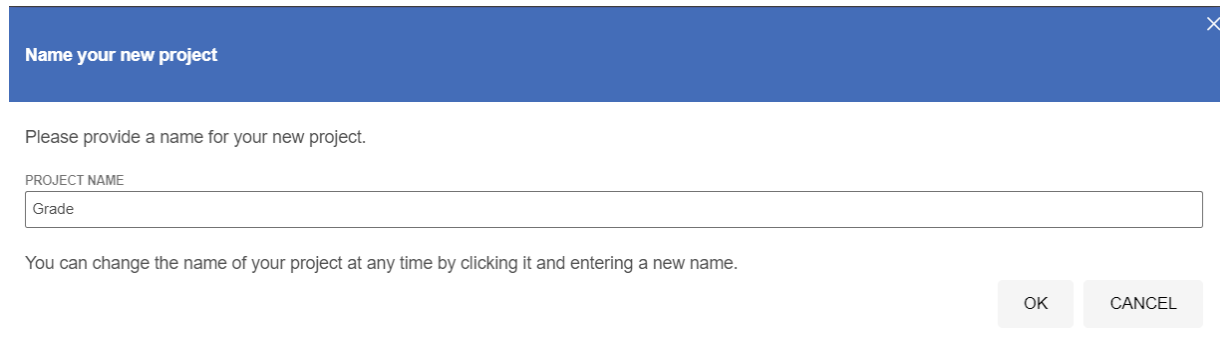
Check the option to allow access to file URLs.



Step 5: Open Selenium IDE.

Step 6: Click on the option “Record a new test in a new project”

Enter the project file and continue



Press cancel when given the prompt to set the projects base url

Step 7: In the Playback Base URL box, paste the copied path of the HTML file

Click on the record button on the right-hand side of the screen

Step 8: It will open the HTML page with the “selenium is recording” overlay


Type the marks and click on Get result

	Command	Target	Value
1	open	file:///C:/Users/danie/Documents/grade.html	
2	set window size	628x816	
3	click	id=markInput	
4	type	id=markInput	60
5	click	css=button	

Enter Mark :

Grade Point :

Press on the stop recording option

Step 9: Press the  button to run the Selenium program to verify the grade

Test case to select the number of students who have scored more than 60 in a subject

Prerequisites:

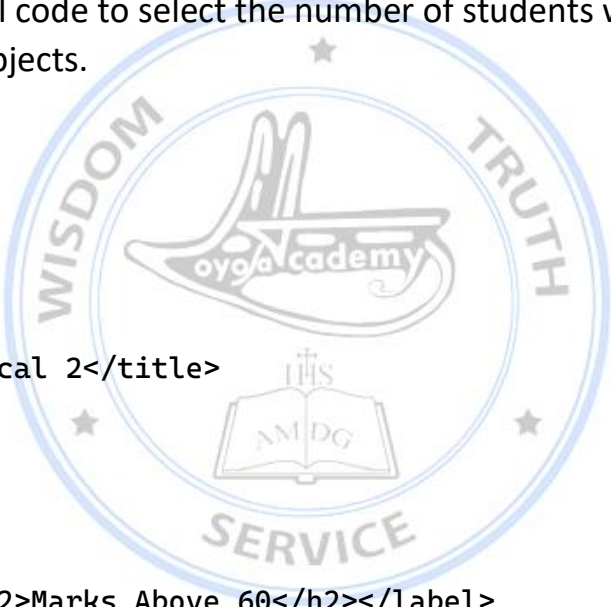
Chromedriver must be downloaded on the system.

Steps:

Step 1: Open Notepad (or) any Text Editor.

Step 2: Type in html code to select the number of students who have scored more than 60 in subjects.

Code:



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Practical 2</title>
</head>
<body>
  <form>
    <label><h2>Marks Above 60</h2></label>
    <input type = "text" id = "output">
    <button type = "button" onClick = "result()">Total
Students</button>
  </form>
  <script>
    function result(){
      var count = 0;
<!--Array of all the listed marks -->
      var markArr = [30,45,50,68,99,79,89,53,41,50];
      for(var i = 0;i<markArr.length;i++){
```

```

        if(markArr[i] >= 60){
            count++;
        }
        else{
            continue;
        }
    }

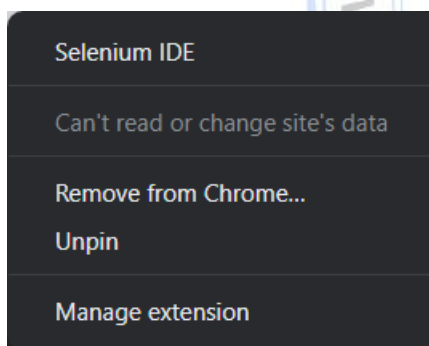
    document.getElementById("output").value = count + "
students scored above 60";
}
</script>
</body>
</html>

```

Step 3: Save the file with .html extension

Step 4: Open Chrome where the selenium IDE is installed.

Right click on the selenium extension> manage extension



Check the option to allow access to file URLs.



Step 5: Open Selenium IDE.

Step 6: Click on the option "Record a new test in a new project"

Enter the project file and continue

Name your new project

Please provide a name for your new project.

PROJECT NAME

You can change the name of your project at any time by clicking it and entering a new name.

OK

CANCEL

Press cancel when given the prompt to set the projects base url

Step 7: In the Playback Base URL box, paste the copied path of the HTML file

Click on the record button on the right-hand side of the screen

Step 8: It will open the HTML page with the “selenium is recording” overlay

Type the marks and click on Get result


	Command	Target	Value
1	open	file:///C:/Users/daniel/Documents/grade.html	
2	set window size	628x816	
3	click	id=markInput	
4	type	id=markInput	60
5	click	css=button	

Enter Mark :

Grade Point :

Get Result

Press on the stop recording option

Step 9: Press the  button to run the Selenium program to calculate the number of students

Program to count the number of items in a Combo Box

Prerequisites:

Pre-configuration of Selenium in Eclipse

Steps to create the HTML file:

Step 1: Open Notepad

Step 2: Type the following code (to create a combobox)

```
<html>
<head>
    <title>combobox read test</title>
</head>
<body>
    <form>
        <select name="cars" id="cars">
            <option value="volvo">Volvo</option>
            <option value="saab">Saab</option>
            <option value="mercedes">Mercedes</option>
            <option value="audi">Audi</option>
        </select>
    </form>
</body>
</html>
```

Step 3: Save it as a .html file

Steps(for selenium):

Step 1: Open Eclipse IDE.

Step 2: Create a new Java Project and configure it to work with Selenium.

Step 3: Double-click on the class file to open it up.

In the class, type in the following code:

```
package practicals;

import java.util.List;

import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

class ComboTest {
    @Test
    void test() {

        System.setProperty("webdriver.chrome.driver", "(chromedriver-
        path)");

        ChromeDriver driver = new ChromeDriver();
        driver.get(combobox-html-file);

        Select dropDown = new
        Select(driver.findElement(By.id("cars")));

        List<WebElement> elementCount = dropDown.getOptions();

        System.out.println("Number of items : " +
        elementCount.size());
    }
}
```

Things to replace:

- Replace (*chromedriver_path*) with the path of ChromeDriver on your system.
- Replace (*combobox-html-file*) with the path to the HTML file.

The program should look like:

```

1 package practicals;
2
3
4 import java.util.List;
11
12 class ComboTest {
13     @Test
14     void test() {
15         System.setProperty("webdriver.chrome.driver", "C:\\Users\\danie\\Documents\\chromedriver.exe ");
16         ChromeDriver driver = new ChromeDriver();
17         driver.get("C:\\\\Users\\\\danie\\\\Documents\\\\Combobox.html");
18
19         Select dropDown = new Select(driver.findElement(By.id("cars")));
20         List<WebElement> elementCount = dropDown.getOptions();
21         System.out.println("Number of items : " + elementCount.size());
22     }
23
24 }

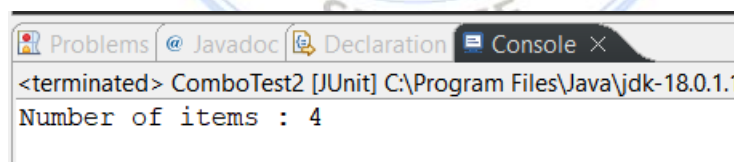
```

Note: If any warnings occur, click on them and import the required packages.

Step 4: Clean by selecting Project -> Clean

Step 5: Run the program by clicking on Run -> Run (or) the Run icon.

OUTPUT



Testing a HTML file

Prerequisites:

Chromedriver must be downloaded on the system.

Steps:

Step 1: Open Notepad (or) any Text Editor.

Step 2: Type in basic html code to make a webpage.

Example code:

```
<html>
  <head>
    <title> Test Page </title>
    <style> body {background-color: rgb(241, 135, 135); color:
greenyellow;} </style>
  </head>
  <body>
    <center>
      <h1> A Cool Title </h1>
      <p> This is a paragraph, this page is boring cus of no css
</p>
      <h2> One more cool title </h2>
      <p> This is another paragraph </p>
    </center>
  </body>
</html>
```

Step 3: Save the file with .html extension.

Step 4: Open Eclipse IDE.

Step 5: Create a new Java Project and configure it to work with Selenium.

Before moving to the next step, make sure you copy the path of
ChromeDriver and your html file.

Step 6: Double click on the class file to open it up.

In the class, type in the following code:

```
@Test

Public void test() {

    System.setProperty("webdriver.chrome.driver", "(chromedriver_path)");

    ChromeDriver driver = new ChromeDriver();

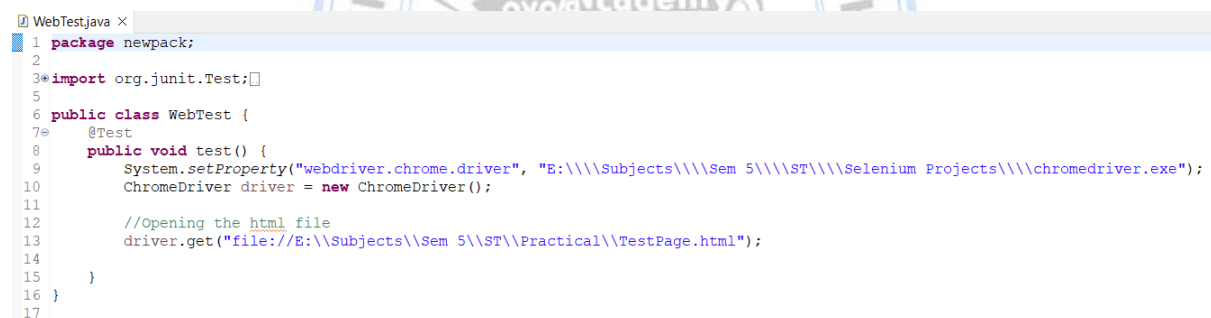
    //Opening the html file
    driver.get("file://(html_file_path)");

}
```

Paste the ChromeDriver location in place of *(chromedriver_path)*

And the html file path in place of *(html_file_path)*

It should look like



```
WebTest.java x
1 package newpack;
2
3 import org.junit.Test;
4
5
6 public class WebTest {
7     @Test
8     public void test() {
9         System.setProperty("webdriver.chrome.driver", "E:\\\\Subjects\\\\Sem 5\\\\ST\\\\Selenium Projects\\\\chromedriver.exe");
10        ChromeDriver driver = new ChromeDriver();
11
12        //Opening the html file
13        driver.get("file://E:\\\\Subjects\\\\Sem 5\\\\ST\\\\Practical\\\\TestPage.html");
14
15    }
16 }
17
```

Note: If any warnings occur, click on them and import the required packages

Step 7: Clean by selecting Project -> Clean

Step 8: Run the program by clicking on Run -> Run (or) the Run icon

Test a Program to add two numbers

Prerequisites:

Selenium IDE must be downloaded on the system.

Steps:

Step 1: Open Notepad (or) any Text Editor.

Step 2: Type in html code to add two numbers.

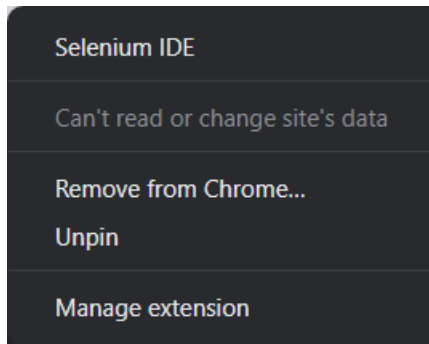
Example code:

```
<html>
<body>
  <form>
    <label><h2>Enter First Number : </h2></label>
    <input type = "text" id = "firstNum">
    <label><h2>Enter Second Number : </h2></label>
    <input type = "text" id = "secondNum">
    <br> <br>
    <button type = "button" onclick = "result()">Add!</button>
    <input type = "text" id = "output">
    <br>
  </form>
  <script>
    function result(){
      var num1 = parseInt(document.getElementById("firstNum").value);
      var num2 = parseInt(document.getElementById("secondNum").value);
      var sum = num1+num2;
      document.getElementById("output").value = sum;
    }
  </script>
</body>
</html>
```

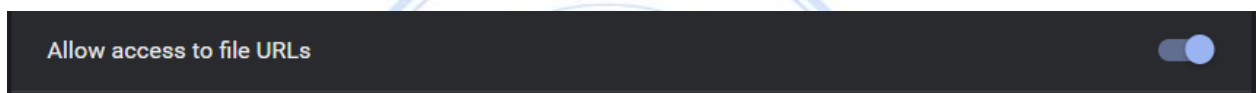
Step 3: Save the file with .html extension

Step 4: Open Chrome where the selenium IDE is installed.

Right click on the selenium extension> manage extension



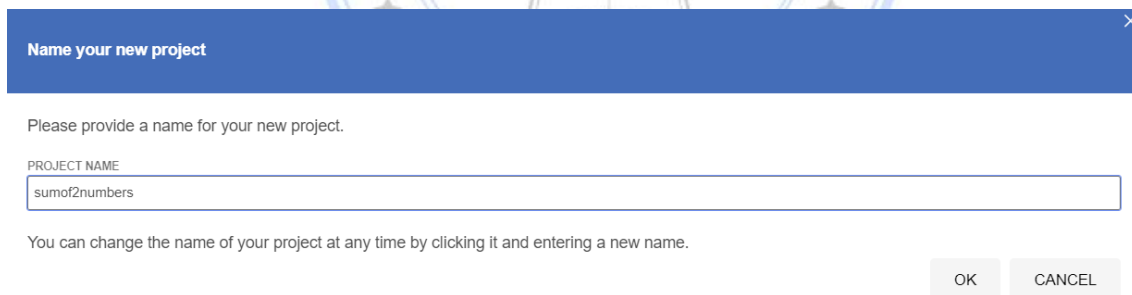
Check the option to allow access to file URLs.



Step 5: Open Selenium IDE.

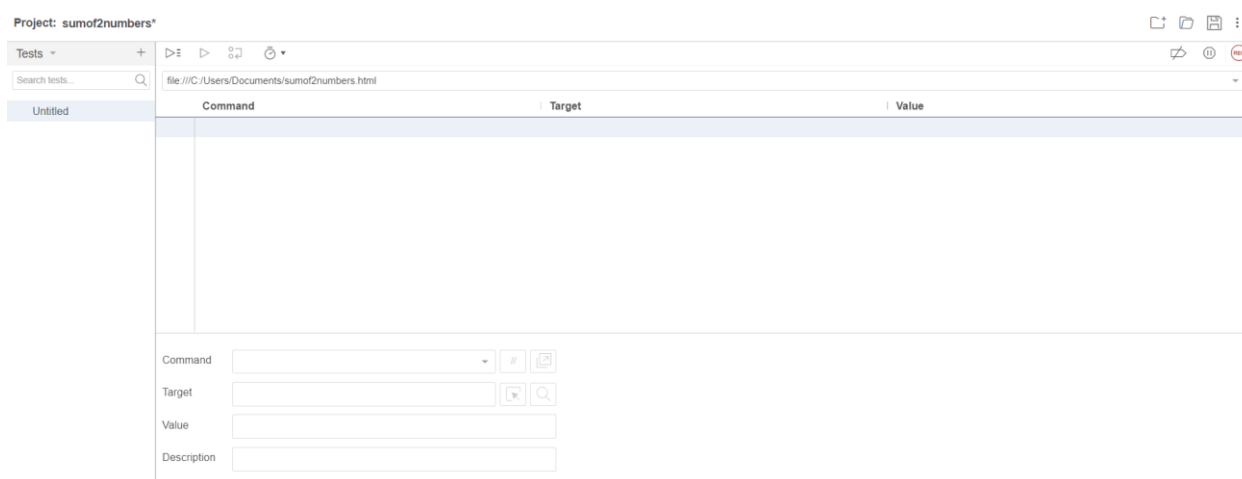
Step 6: Click on the option “Record a new test in a new project”

Enter the project file and click OK



Press cancel on the next pop-up when given the prompt to set the projects base url

Step 7: In the Playback Base URL box, paste the copied path of the HTML file



Click on the record button on the right-hand side of the screen

Step 8: It will open the HTML page with the “selenium is recording” overlay

Type the numbers in the following boxes and press add

Enter First Number :

10

Enter Second Number :

20

Add!

30

Press on the stop recording option

Step 9: Press the  button to run the Selenium program to add two numbers.