

DAY 7

CE 1:

Task 01: Maven Configuration

Create the Maven Project.

Add the Dependencies of Selenium Web Driver and TestNG in POM.Xml

Perform the following Steps:

1. Open the browser
2. Launch the <https://demo.wpeverest.com/user-registration/guest-registration-form/>
3. Assert the title of the Page.

CODING :

```
package day6;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeMethod;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;

public class ce1 {
    WebDriver driver;
    @Test
    public void check() {
        driver.get("https://demo.wpeverest.com/user-registration/guest-registration-form/");
        String exp="Guest Registration Form – User Registration";
        String act=driver.getTitle();
        driver.manage().window().maximize();
        Assert.assertEquals(exp, act);

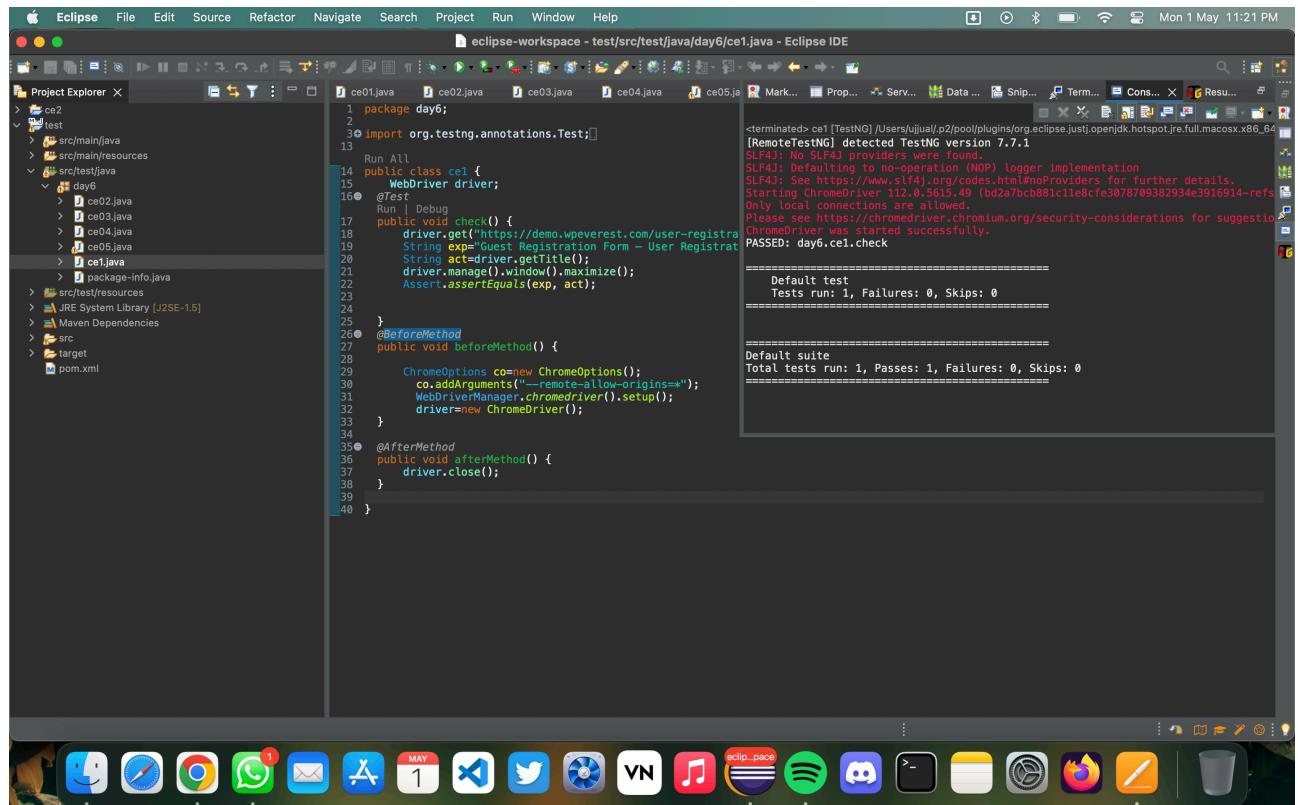
    }
    @BeforeMethod
    public void beforeMethod() {

        ChromeOptions co=new ChromeOptions();
        co.addArguments("--remote-allow-origins=*");
        WebDriverManager.chromedriver().setup();
        driver=new ChromeDriver();
    }
}
```

```
@AfterMethod
public void afterMethod() {
    driver.close();
}

}
```

OUTPUT :



The screenshot shows the Eclipse IDE interface on a Mac OS X desktop. The Project Explorer view on the left lists several Java files: ce01.java, ce02.java, ce03.java, ce04.java, ce05.java, ce1.java, package-info.java, and pom.xml. The ce1.java file is selected and shown in the editor pane. The code implements a TestNG test named 'ce1' that checks if the title of the browser window matches the expected value. It uses WebDriverManager to setup the ChromeDriver and adds a command-line argument to disable remote origins. The code concludes with an @AfterMethod annotation that closes the driver.

The right-hand pane displays the execution output of the test. It starts with a log message from TestNG indicating it's using version 7.7.1. It then shows the test being run, with the command 'java -jar target/testng-reports-1.0.jar'. The output indicates that the test passed successfully, with 1 test run, 0 failures, and 0 skips. Finally, it shows the total suite results with 1 test run, 1 pass, 0 failures, and 0 skips.

```
terminated> java -jar target/testng-reports-1.0.jar
[Ran 1 TestNG test]
[TestNG] detected TestNG version 7.7.1
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 112.0.5615.49 (bd2a7bcbb81c1e8cf3078709382934e3916914-refs
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestio
ChromeDriver was started successfully.
PASSED: day6.ce1.check
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====
```

CE 2 :

Task 02: TestNg Annotation

Design and execute a calculator logic which checks all the mathematical operations.

1. Create a Calculator class containing mathematical methods like add, subtract, multiply, and divide methods.
2. Write the TestNg method to test the Calculator Class.
3. Use appropriate Assertion to validate the results.
4. Set the priority of test cases.
5. Execute the TestNG File.

CODING :

```
package day6;

import org.testng.annotations.Test;
import org.testng.Assert;

public class ce02 {
    @Test(priority=2)
    public void add() {
        int a=10,b=20,res=0;
        res=a+b;
        System.out.println(res);
        Assert.assertEquals(res, 30);
    }
    @Test(priority=3)
    public void sub() {
        int a=20,b=1,res=0;
        res=a-b;
        System.out.println(res);
        Assert.assertEquals(res,19);
    }
    @Test(priority=1)
    public void div() {
        int a=20,b=2,res=0;
        res=a/b;
        System.out.println(res);
        Assert.assertEquals(res, 10);
    }
    @Test(priority=4)
    public void mul() {
```

```
int a=10,b=5,res=0;  
res=a*b;  
System.out.println(res);  
Assert.assertEquals(res, 50);  
}  
}
```

OUTPUT :

The screenshot shows the Eclipse IDE interface on a Mac OS X desktop. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The title bar indicates the current workspace is 'eclipse-workspace - test/src/test/java/day6/ce02.java - Eclipse IDE'. The bottom dock contains various application icons.

The left side features the 'Project Explorer' view, which lists several Java files: ce01.java, ce02.java, ce03.java, ce04.java, ce05.java, ce06.java, ce07.java, and package-info.java. It also shows a 'test' folder containing sub-folders like 'src/main/java', 'src/main/resources', and 'src/test/java/day6'. Other items listed include 'JRE System Library [J2SE-1.5]', 'Maven Dependencies', and 'target'.

The main editor area displays the Java code for 'ce02.java'. The code defines a class 'ce02' with methods for addition, subtraction, division, and multiplication, each annotated with @Test and @Test(priority=i). The code uses System.out.println to print the results and Assert.assertEquals to verify them against expected values (10, 30, 50).

The right side of the interface shows the 'Console' tab, which displays the execution output. It starts with a message about SLF4J defaulting to no-operation (NOP) logger implementation. The output then shows four test cases: add(), sub(), div(), and mul() all passing with results 10, 30, 50 respectively. A summary at the end of the console output states 'Default suite' and 'Total tests run: 4, Passes: 4, Failures: 0, Skips: 0'.

CE 3 :

Task 03: TestNg Annotation

- Create the TestNg File.
- Keep the below commands under the BeforeMethod
 - Open the Page: <https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>
 - Maximize the Screen.
- Keep the below commands under the Test.
 - Enter the Username and Password.
 - Click Login.
 - Validate the User is successfully logged in.
- Keep the below commands in AfterMethod
 - Close the browser.

CODING :

```
package day6;

import org.testng.annotations.Test;
import org.testng.annotations.BeforeMethod;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;

public class ce03 {
    WebDriver driver;
    @Test
    public void test() throws InterruptedException {
        WebElement name = driver.findElement(By.name("username"));
        name.sendKeys("Admin");
        Thread.sleep(4000);
        WebElement pass = driver.findElement(By.name("password"));
        pass.sendKeys("admin123");
        Thread.sleep(4000);
```

```

        driver.findElement(By.xpath("//html/body/div/div[1]/div/
div[1]/div/div[2]/div[2]/form/div[3]")).click();
        Thread.sleep(4000);
        String txt = "Dashboard";
        String login = driver.findElement(By.xpath("//html/body/div/
div[1]/div[1]/header/div[1]/div[1]/span/h6")).getText();
        Assert.assertEquals(txt, login);

    }

    @BeforeMethod
    public void beforeMethod() throws InterruptedException {
        driver = new ChromeDriver();
        String url = "https://opensource-demo.orangehrmlive.com/
web/index.php/auth/login";
        driver.get(url);
        driver.manage().window().maximize();
        Thread.sleep(4000);
    }

    @AfterMethod
    public void afterMethod() {
        driver.close();
    }
}

```

OUTPUT :

```

eclipse-workspace - test/src/test/java/day6/ce03.java - Eclipse IDE
Tue 2 May 9:28 AM

1 package day6;
2
3 import org.testng.annotations.Test;
4 import org.testng.annotations.BeforeMethod;
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.WebElement;
8 import org.openqa.selenium.chrome.ChromeDriver;
9 import org.testng.Assert;
10 import org.testng.annotations.AfterMethod;
11
12 Run All
13 public class ce03 {
14     WebDriver driver;
15     @Test
16     public void test() throws InterruptedException {
17         WebElement name = driver.findElement(By.name("username"));
18         name.sendKeys("Admin");
19         Thread.sleep(4000);
20         WebElement pass = driver.findElement(By.name("password"));
21         pass.sendKeys("admin123");
22         Thread.sleep(4000);
23         driver.findElement(By.xpath("//html/body/div/div[1]/d
24         String txt = "Dashboard";
25         String login = driver.findElement(By.xpath("//html/b
26         Assert.assertEquals(txt, login);
27     }
28
29     @BeforeMethod
30     public void beforeMethod() throws InterruptedException {
31         driver = new ChromeDriver();
32         String url = "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login";
33         driver.get(url);
34         driver.manage().window().maximize();
35         Thread.sleep(4000);
36     }
37
38     @AfterMethod
39     public void afterMethod() {
40         driver.close();
41     }
42 }

```

terminated: ce03 [TestNG] /Users/ujjual/p2/pool/plugins/org.eclipse.jdt/openjdk.hotspot.jre.full.macosx.x86_64 [RemoteTestNG] detected TestNG version 7.7.1
SLF4J: No SLF4J providers were found.
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 112.0.5615.49 (bd2a7bcbb81c1e0fcfe3078709382934e3916914-refs
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions
ChromeDriver was started successfully.
PASSED: ce03.test

=====
Default test
Tests run: 1, Failures: 0, Skips: 0

=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

CE 4:

Task 04: TestNg Annotation

- Create the TestNg File.
- URL: <https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>
- Create the below testcases under the Test annotation
 - TestCase1: Check Login Page is loaded Successfully.
 - TestCase2: Check Logout is working fine.
- Use depends on attribute to make the Test case2 is depends on the TestCase1.

CODING :

```
package day6;
import org.testng.annotations.Test;
import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.Assert;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;

public class ce04 {

    WebDriver driver;

    @Test
    public void login() throws InterruptedException {
        Thread.sleep(3000);
        driver.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/form/div[1]/div/div[2]/input")).sendKeys("Admin");
        driver.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/form/div[2]/div/div[2]/input")).sendKeys("admin123");
        driver.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
        Thread.sleep(3000);
        String str=driver.findElement(By.xpath("//html/body/div/div[1]/div[1]/header/div[1]/div[1]/span/h6")).getText();
        Assert.assertEquals(str, "Dashboard");
    }
}
```

```

@Test (dependsOnMethods="login")
public void logout() throws InterruptedException {
//    driver.switchTo().alert().accept();
    driver.findElement(By.xpath("/html/body/div/div[1]/div[1]/header/div[1]/div[2]/ul/li/span/p")).click();
    driver.findElement(By.xpath("/html/body/div/div[1]/div[1]/header/div[1]/div[2]/ul/li/ul/li[4]/a")).click();
    Thread.sleep(3000);
    String str=driver.findElement(By.xpath("/html/body/div/div[1]/div/div[1]/div/div[2]/h5")).getText();
    Assert.assertEquals(str, "Login");
}

@BeforeClass
public void beforeMethod() {
    ChromeOptions co = new ChromeOptions();
    co.addArguments("--remote-allow-origins=*");
    WebDriverManager.chromedriver().setup();
    driver = new ChromeDriver();

    driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
    driver.manage().window().maximize();
}

@AfterClass
public void afterMethod() {
    driver.close();
}
}

```

OUTPUT :

```

eclipse-workspace - test/src/test/java/day6/ce04.java - Eclipse IDE
=====
terminated: ce04 [TestNG] /Users/ujjua/p2/pool/plugins/org.eclipse.jst/openjdk.hotspot.re.full.macosx.x86_64
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 89.0.4389.49 (bd2a9bc081c1e8cf307870938293463916914-refs
Only local connections are allowed
Please see https://chromedriver.chromium.org/security-considerations for suggestions
ChromeDriver was started successfully.
=====
PASSED: day6.ce04.logout
=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====
=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====

=====

```

CE 5:

Task 05: TestNg Annotation-Group By

- Create the TestNg File.
- **Test Case 01** - Open Godaddy.com and Validate it's Page title.
- Steps to Automate:
 1. Launch browser of your choice say., Firefox, chrome etc.
 2. Open this URL - <https://www.godaddy.com/>
 3. Maximize or set size of browser window.
 4. Get Title of page and validate it with expected value.
 4. Get URL of current page and validate it with expected value.
 5. Close browser.
- **Test Case 02**- Open Godaddy.com and click on 'Domain Name Search'
- Steps to Automate:
 1. Launch browser of your choice like, firefox, chrome etc., using selenium webdriver.
 2. Open website url - <https://godaddy.com/>

-
- Steps to Automate:
 1. Launch browser of your choice like, firefox, chrome etc., using selenium webdriver.
 2. Open website url - <https://godaddy.com/>
 3. Maximize browser window.
 4. Set timeout using implicit wait command of Selenium Webdriver.
 5. Click on the first menu link, which is 'Domains'. It will open up a sub-menu, click on the 'Domain Name Search' link from the sub-menu.
 - **Test Case 03**- Open Godaddy.com and click on 'Domain Name Search'
 - Steps to Automate:
 - 1.Launch browser of your choice like, firefox, chrome etc., using selenium webdriver.
 2. Open website url - <https://godaddy.com/>
 3. Maximize browser window.
 4. Set timeout using implicit wait command of Selenium Webdriver.

CODING :

```
package com.testNG;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeClass;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.Assert;
import org.testng.annotations.AfterClass;

public class ce5 {
    WebDriver driver;
    @Test(groups="SmokeTest")
    public void check() {
        String text = driver.getTitle();
        String expected="Domain Names, Websites, Hosting &
Online Marketing Tools - GoDaddy IN";
        Assert.assertEquals(text,expected);
    }

    @SuppressWarnings("deprecation")
    @Test(groups="RegressionTest")
    public void testcase2() throws InterruptedException
    {
        driver.findElement(By.xpath("//
*[@id=\"id-631b049a-e9c0-4d24-8710-c504745206dd\"]/
div[2]/div[1]/ul/li[1]/button")).click();

        driver.manage().timeouts().implicitlyWait(4000, TimeUnit.S
ECONDS);
```

7. We'll match value fetched in step 7 with expected value, if it's matched then proceed to next step either failed the test case.
8. Verify that search box is present on the page and it's enabled by using selenium commands.
9. Verify that "Buy It" button is available along with search box.
10. Enter some test value in the search box, like "mydomain" and click on "Buy It" button.
11. If domain is available then verify that "Add to Cart" button is present alongside domain name.
12. Verify that price of the domain is also displaying alongside domain name.

Requirements to met:

1. Create the three tests
2. Group the Testcase01 as SmokeTest
3. Group TestCase02 and Testcase 03 as the Regression TestCase
4. Execute as the TestNG.xml File

```
driver.findElement(By.xpath("//  
*[@id=\"id-631b049a-e9c0-4d24-8710-c504745206dd\"]/  
div[2]/div[1]/ul/li[1]/div/div[2]/div[1]/ul/li[2]/  
a")).click();  
  
driver.manage().timeouts().implicitlyWait(4000, TimeUnit.S  
ECONDS);  
  
}  
@BeforeClass  
public void beforeClass() throws InterruptedException {  
    ChromeOptions co=new ChromeOptions();  
    co.addArguments("--remote-allow-origins=*");  
    WebDriverManager.chromedriver().setup();  
    driver=new ChromeDriver();  
    driver.get("https://www.godaddy.com/");  
    driver.manage().window().maximize();  
    Thread.sleep(6000);  
}  
  
@AfterClass  
public void afterClass() throws InterruptedException {
```

```
Thread.sleep(6000);
```

```
driver.close();
```

```
}
```

```
}
```

OUTPUT :

```
<terminated> ce5 [TestNG] C:\Program Files\Java\jdk-19\bin\javaw.exe (01-May-2023, 7:36:13 pm – 7:36:32 pm) [pid: 30784]
|[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 112.0.5615.49 (bd2a7bcb881c11e8cfe3078709382934e3916914-refs/branch-heads/5615@(#936)) on port 5623
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
PASSED: com.testng.ce5.testcase2
PASSED: com.testng.ce5.check

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====
```

727721eucs172
Ujjual