## WRITEUP:

**GITHUB LINK:** <https://github.com/SweathaRJ/JAVA-FSD-PHASE5/tree/main/TestNgExample>

Step 1: Create a maven project and add dependencies in pom.xml file.

Step 2: Create a testng class and add the code.

**package** com.simplilearn.testing;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeMethod;

**import** org.testng.annotations.Test;

**import** org.testng.annotations.Test;

**public** **class** FacebookLogin {

WebDriver driver;

@Test

**public** **void** facebook() {

driver.get("https://www.facebook.com/");

driver.findElement(By.*id*("email")).sendKeys("sweathasureshf1@gmail.com");

driver.findElement(By.*id*("pass")).sendKeys("swea@123");

driver.findElement(By.*name*("login")).submit();

}

@BeforeMethod

**public** **void** beforeMethod() {

System.*setProperty*("webdriver.chrome.driver", "D:\\Phase 5\\chromedriver\_win32\\chromedriver.exe");

driver= **new** ChromeDriver();

}

@AfterMethod

**public** **void** afterMethod() {

//driver.close();

driver=**null**;

}

}

Step 3: Add dependencies in pom.xml

<!--ReportNg 1.1.4-->

<dependency>

<groupId>org.uncommons</groupId>

<artifactId>reportng</artifactId>

<version>1.1.4</version>

<scope>test</scope>

</dependency>

<!--Velocity-dep 1.4-->

<dependency>

<groupId>velocity</groupId>

<artifactId>velocity-dep</artifactId>

<version>1.4</version>

</dependency>

<!--Guice 3.0-->

<dependency>

<groupId>com.google.inject</groupId>

<artifactId>guice</artifactId>

<version>3.0</version>

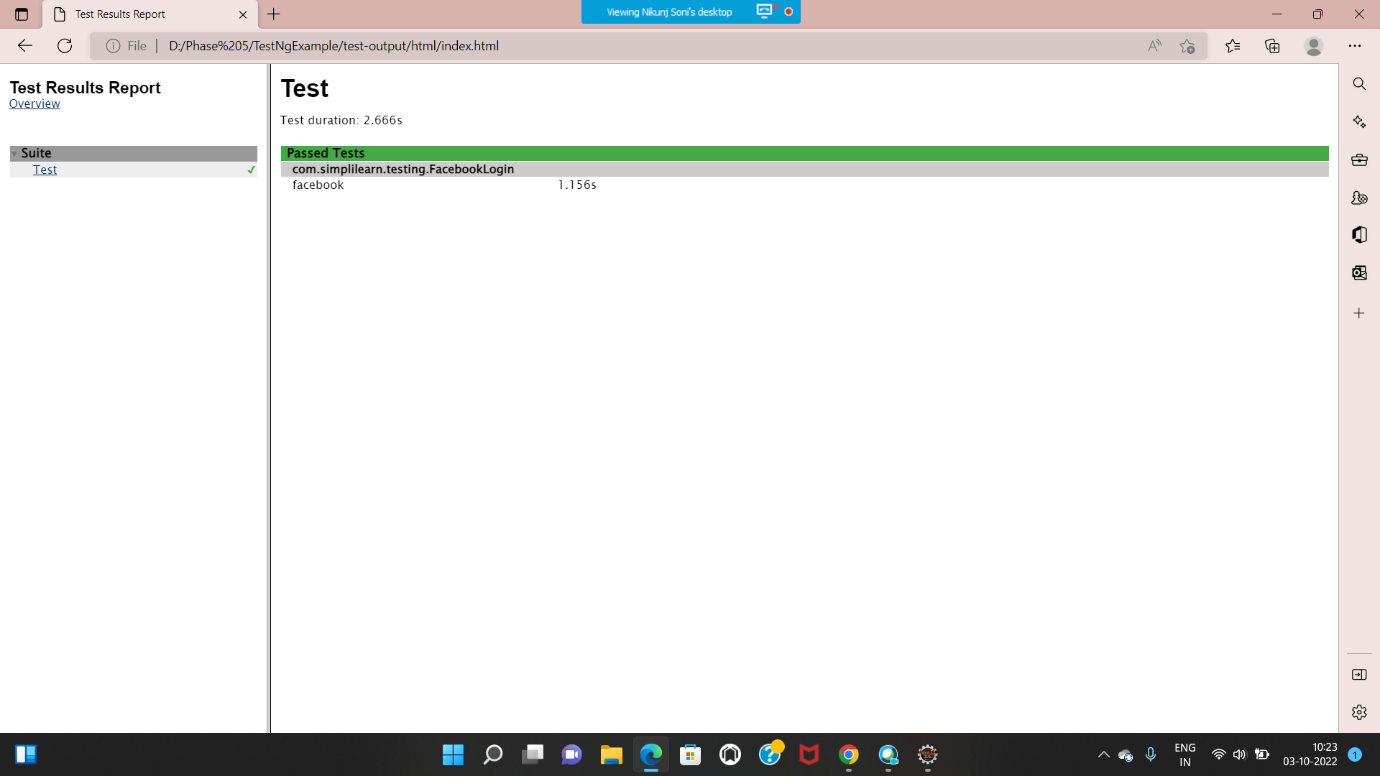
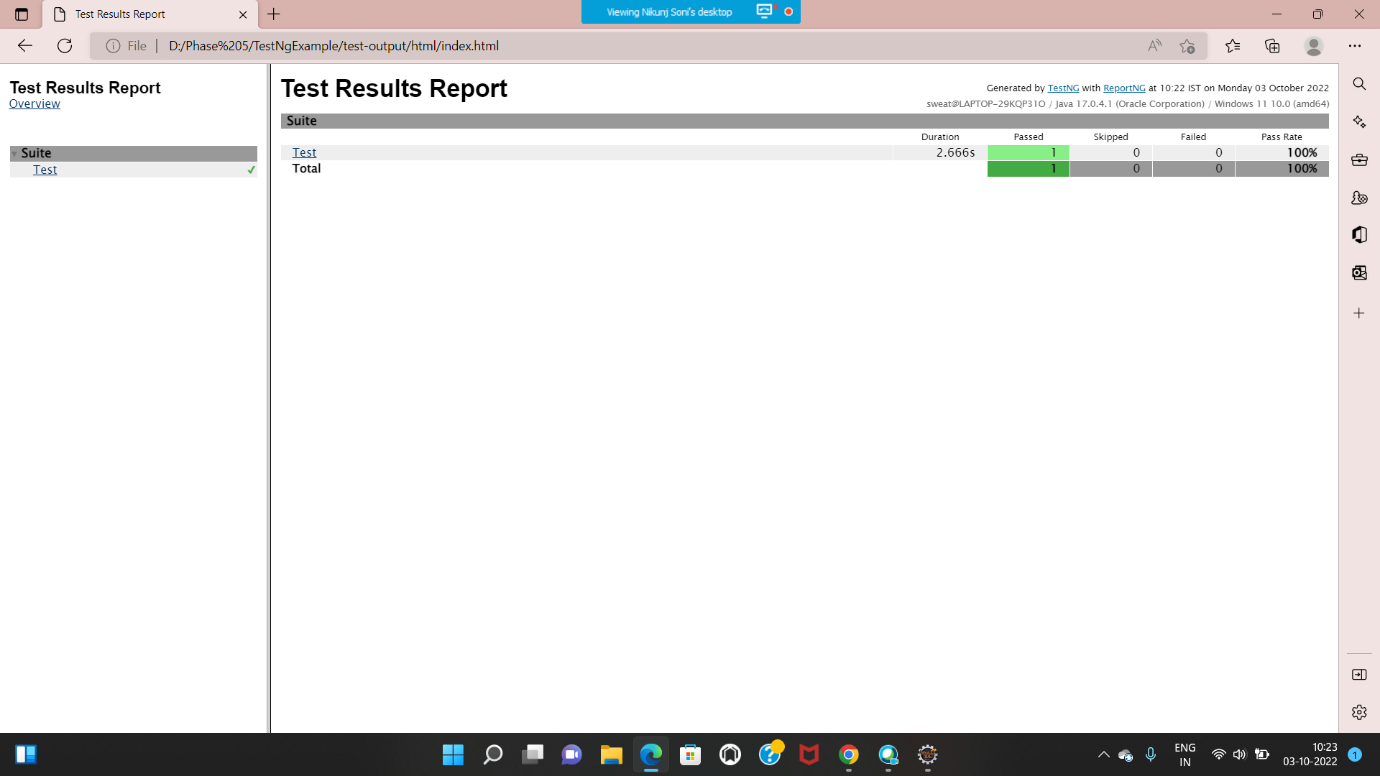
</dependency>

Step 4: Now go to project>right click>properties>testng

* Disable the default listener.
* Click on apply and close.
* Add listener in testng.xml

Step 5: Refresh the project.

* Html folder will be created
* Right click the index.html and open with web browser.
* Reportng will be prepared.



Output:

