5. Demonstrate how test reports are exported to Excel.

Step 1 Creating a project with test cases with multiple annotations

- Open the Eclipse and create a Java project.
- Create multiple test case classes(Say Test_01, Test02).
- Create a Base class to extend the test cases.

Step 2 Adding AT Excel report jars

- Extent Reports jar file is already present in your practice lab in /home/ubuntu/libs directory.
- Add the Extent Reports jar file to your project: Right-click on project->Build path->Configure build path->Add external Jars.
- Click on Apply and then click OK.

Step 3 Executing the test suites to see the generated report in Excel sheet

- Write the test script in the Test_01 class.

```
package Testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import base.Baseclass;

public class TEST_01 extends Baseclass {

        @Test
        public void t_001()
        {

driver.findElement(By.xpath("(//*[contains(text(),'Categories')])[1]")).cli
ck();

driver.findElement(By.xpath("//*[contains(text(),'Central')]")).click();
                System.out.println("Test_01 executed successfully");
        }


}
```

- Write the test script in the Test_02 class.

```java
package Testcases;

import org.openqa.selenium.By;
import org.testng.annotations.Test;

import base.Baseclass;

public class TEST_02 extends Baseclass {
	@Test
	public void t_002() {


	driver.findElement(By.xpath("(//*[contains(text(),'Popular')])[1]")
).click();
		System.out.println("Test_02 executed successfully");
	}

}
```

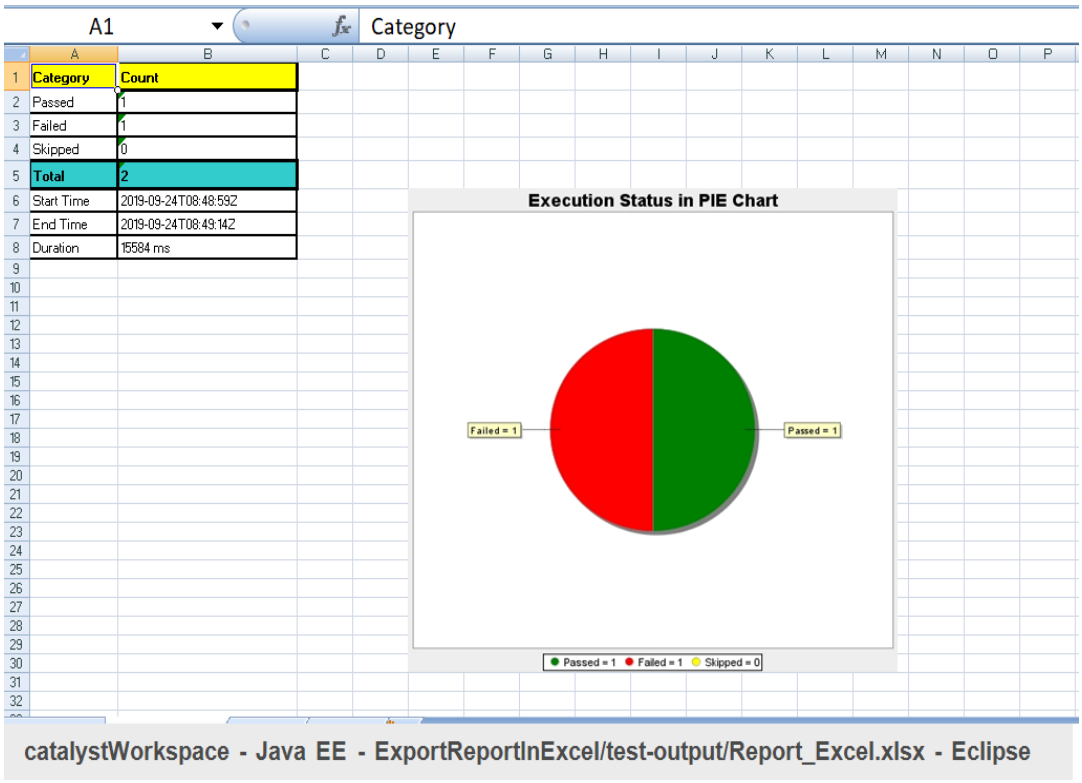
- Write the test script for the extended Base class, where all annotations are declared here.

```java
package base;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.BeforeMethod;
import org.automationtesting.excelreport.Xl;

public class Baseclass {

	public WebDriver driver;

	@BeforeMethod
	public void baseclass1()
	{
		System.setProperty("webdriver.gecko.driver",
"/home/ubuntu/Downloads/gechodriver");
		driver = new geckodriver();
		driver.get("https://mvnrepository.com/");
	}

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

	@AfterSuite
	public void generateReport() throws Exception {
		Xl.generateReport("Report_Excel.xlsx");
	}

}
```

- Execute the test suite with multiple test cases and the testng.xml file will look like :

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="ExportReport">


        <test name="TEST1">

                <classes>
                        <class name="Testcases.TEST_01"></class>
                </classes>
</test>

 <test name="TEST2">

                <classes>
                        <class name="Testcases.TEST_02"></class>
                </classes>
        </test>
</suite>
```

Finally, the executed script can generate the report in Excel and the graph will look like :



Step 4: Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master