Diagram

Pom.xml
Selenium 4.9.0
Apache poi 5.3.1
Apache TestNG 7.7.0
Apache commons.io <dependency>

Jenkins

CI/CD Tool

TestNG.xml
<suit>
<groups>
<include>
<classes>
<class>

testNGreport.html
Passed failed skipped

TestScripts

```
public class ClassName{
@Test
public void testcase(){
------------------
------------------
}
}
```

CommonUtility
BaseClass
ExcelUtility
FileUtility
WebdriverUtility

TestData
commondata.properties
testcasespecificdata.xlsx

Resources
Chromedriver.exe
Firefoxdriver.exe
Edgedriver.exe

pom/object repository

Login{ @findBy }	HomePage{ @findBy }	LoginToContinue{ @findBy }

1. CommonUtility

- In Common utility package we created the different classes likes Base Class, Excel Utlity, File Utility and Web driver Utility, because of that reusability, synchronisation can be achieved.
- One person build the code but used by the another person with the help of method calling so the reusability of code will be happened.
- In the frame work the data can not be hardcoded so because of that the data can be stored in property and excel files.
- Both in excel and file utility class the file input stream object is created for reading the byte oriented data from both the classes and the data is secured because only specific data can retrieved.
- In web driver utility class the reusability of parameterised methods as been created to reduce the number of scripts and exception can be controlled as much as possible.
- One person build the code but used by the another person with the help of method calling so the reusability of code will be happened.
- During runtime which browser should be selected is based on the data written in property and excel file from this we can achieved run time polymorphism.
- In base class post and pre-condition as written with the help of before and after method annotations and object is created for file, excel and web driver utility class in base class and base class will extend to test scripts class.

2. Test Data

- In the test data package data will be stored with the help of the excel and properties file.

3. Page object model (object repository)

- In this pom the address will be written for all the web element and web elements with the help of locators by using find by annotations and we should call getter method because of that encapsulation can be achieved.
- Here we modify the web element as private because of that we can't access that web element in the different class so with the help of getter method we make that web element as public and from that references we can access that web element in different class.

➢ One webpage contains one class i.e. for sign in page a class will be created like that for each and every webpage we should create a class.
➢ Both common utility & pom repository packages will be stored in the main folder of the project.

4. Test Scripts

➢ All the test scripts class should be extended to base class with the help of is a relationship from this we can achieve single level inheritance.
➢ All manual test cases will be converted into automation test scripts & one test case contain one non static method and all the methods should be annoted with test annotation and test annotation acts like a main method.
➢ For the test scripts we can give the priority based on the navigational steps & also we can pass different arguments like enabled, invocation count, groups, depends on methods etc ...
➢ All the test scripts can be run in the suit file with help of testNG.xml.

5. testNG.xml

➢ Data driven testing can also be done with the help of xml file.
➢ It contains so many tags like suit, groups, include, classes, class, parameters and so on.
➢ Include & exclude tag will be used whenever we have a depends on methods present in the test script and we need to run the particular type of test script we use these tags along with suit, parameters, groups, run, test, class and classes tags will be used.
➢ With the help of suit file, we can also do parallel execution, batch or group execution & full regression testing can be achieved.
➢ Once we run the suit and the execution is completed, we will get a testNG report. From that we will get to know what are the test script are pass, fail and skipped based on that we can re-run the failed and skipped test script & it will gives the report only the test scripts which are re-run.
➢ TestNG report will acts like a proof document for the customer & it contains what are all the types of testing done by the engineers, from this document customer will believe with the engineers because each test script contains a unique Id and it cannot be editable or modifiable.

6. Pom.xml

➢ pom.xml is a project configuration file and in this file all the dependencies are injected into maven repository like selenium, Apache poi, testNG, commons.io.