Framework :- It is a standard rules and guidelines that should be followed by all the automation engineers while automating the test case of the application.



Framework Architecture:-

We always follow Maven Architecture in framework –

In Automation framework there are Three Phases

1. Development Phase

2. Implementation Phase

3. Execution Phase

We have following code structure in our framework –

1. Common Data :- • We take common data from JSON file such as url of the application, username, password and a browser. • It is called as common data because it is same for all the test scripts. • This file has an extension .json.

2. Test Data :- • We take test data from excel file. • Test data is a test script specific data. • This file has an extension .xlsx.

3. Resources :- • Generally, we store all the resources in src/test/resources directory. • The resources are the driver executables such as chromedriver.exe, geckodriver.exe,etc…

4. Generic Utils :- • Generic utility is common for all the tests. • It is present in a src/main/java directory where we create package generic utils.

• Our generic utilities are – ➢ IPath Constants: IPath Constants is an Interface where we store all the constant paths like JSON file path, Excel file path. By default, all the members are static and final so that no one can modify the path.

➢ Data Base Utility: Data Base Utility contains some generic methods to read the data from data base. Here we have generic methods for establishing the connection with the data base, we can execute the query and close the connection with data base.

➢ Excel Utility: In excel utility we store the test data. It’s a test script specific data. We are storing test data in excel file because 1 row will be allocated for the test script. In excel utility there are some generic methods to retrieve the data from excel sheet.

➢ File Utility: In file utility we store common data required for the test script. We have generic methods to read the data from JSON file or Property file. In our project we read the common data from JSON file. We have to store the data in a key, value pair.

➢ Java Utility: Java utility contains the java specific generic methods. In our project for the organization name, it will not accept duplicate organization name. So we have created generic java method which generates unique name every time, we use random number method inside generic java method. We use the current system date for taking the screenshots, we have created a generic java method for that as well.

➢ Web Driver Utility: In our project lots of repetitive action happens so for that we have created some generic methods for that actions. For Eg. Handle the drop down, Mouse Hover, Implicit wait, Explicit Wait, Handle the window, maximize the window, etc…

➢ Listener: Listner is a abstract class which implements ITestListner. It will monitor the script during runtime. Here we write the logic of taking the screenshot. If any test scripts failed it will take the screenshot.

➢ Retry Analysis: We execute multiple test scripts even if any test case gets failed it will continue the execution.

➢ Base Class: All the test scripts should extends Base class. Base class has basic configuration annotations like-. @Before Suite @Before Test @Before Class @Before Method @Test @After Method @After Class @After Test @After Suite All the above annotations are basically used to execute some preconditions or postconditions at different levels. ▪

@Before Suite: Before suite annotated method will execute once for one suite. Here the database connection is established. ▪

@After Suite: After suite annotated method will execute after the suite file. Here we close the database connection. ▪

@Before Test: Before test annotated method will execute before the execution of a test tag in a xml file. ▪

@After Test: After test annotated method will execute after the execution of a test tag in a xml file. ▪

@Before Class: Before annotated method will be executed before Testng class execution. Here we Launch the browser. ▪

@After Class: After annotated method will be executed after Testng class execution. Here we close the browser. ▪

@Before Method: Before annotated method will be executed before the execution of every test case i.e., before

@Test annotation. Here we Login to the application. ▪

@After Method: After annotated method will be executed after the execution of every test case i.e., after @Test annotation. Here we Logout from the application. ▪

@Test: It will execute the actual test script.

5. Object Repository :- • Collection of elements or web element in one place is called as object repository. • We store all the object repository in src/main/java. •

Here we identify all the elements for the specific page and store it in a specific class or page by using @FindBy Annotation or @FindBys Annotation or @FindAll Annotation. •

@FindBy annotation is also locate an element by the locator strategy and at the time of storage it will store element reference not address.

• We used to write our business login in POM class only.

Advantages of POM :

➢ Code reusability

➢ Code optimization

➢ Test script development is faster

➢ Avoid stale element reference exception

➢ Auto healing

6. Test Scripts :- • We write our test scripts in a src/test/java.

• We create a separate separate package according to the module and we write the test scripts.

• Every test script should extend the Base class because the execution is starting from the base class only.

• Here we create a object of our POM pages so that we can access the elements and the business logics in our test scripts. 7. Driver / Rerun / testng.xml:-

• After writing all the test scripts if we want to do batch execution, parallel execution, group execution, etc… we convert our test cases in Testng.xml file.

• So, in xml file as per our need we can execute our test scripts.

8. Reports :- • After execution via Testng.xml file we will get the reports. • In that reports we will get to know that how many test scripts are passed, failed and skipped.

9. Screenshots:- • If any test scripts are failed so that we will get the screenshots.

• We have to create a folder for those screenshots and we have to give the path where we have to store the screenshots.

• In selenium, screenshots always taken in .png format only.

• We have to check those screenshots and if any defect is there so we can raise a defect by attaching the screenshot of the defect.

10. Maven / Build Management Tool :-

• For our project maven architecture is follow to create a framework

. • In pom.xml we have to add all the dependencies and surefire plugins.

• We add dependencies like tentng dependency, selenium dependency, commons.io dependency, apache poi dependency, my sql dependency, Jackson dependency, etc…

• To execute Testng we should add surefire plugins.

11. Jenkins :-

• Jenkins is a CICD tool [Continuous Integration Continuous Deployment tool].

• It is used by developers and automation engineers to deploy [build] the application and run the automation scripts.

• We integrate our automation framework with Jenkins so that it is automatically executed whenever there is a changes in the framework.

• Whenever we get the build if we want to execute any suite we were running on Jenkins.

• Jenkins was pulling the data from the Git Hub repo and was executing on Jenkins and it was giving the reports there itself.

❖ This is the complete structure of my framework.

❖ So, it’s a Hybrid framework because lots of data we are using here so we are implementing the concepts of Data Driven Framework.

❖ We are performing repetitive actions for that we are using so many methods so we are implementing the concepts of Method Driven Framework.

❖ It’s a huge project and modules are also more so we are implementing the concepts of Modular Driven Framework.

❖ It is a combination of all the three frameworks that’s why it’s a Hybrid Framework.

Advantages of Framework :-

1. Maintenance is easy

2. Increased code reusability

3. Improved test efficiency

4. Low maintenance cost

5. Maximum test coverage

6. Minimal manual intervention