**Hybrid Frame Work**

**In our Project, We are following Hybrid Frame Work.**

**It is combination of Page Object Model design pattern, TestNG Frame work and Data driven framework**

**Coming to Page Object Model Design pattern, for every web page or entity specific, we will write separate java class file and corresponding methods will be implemented.**

**And for every page specific, we are declaring Web Elements and maintaining Page specific Object Repository like that.**

**And Coming to the main Test case development,**

**We are importing the Page class Files and**

**We are creating object for every page class file and**

**We are calling required methods as per the Test Case functional flow**

**Like that we are following Page Object Model Design Pattern.**

**Coming to Data Driven Frame work**

**In the TestNG class file, we will specify Excel File path and sheet name**

**To retrieve input data from the excel file.**

**And we are using Excel Util java class file,**

**Which contains excel related methods to read / write data from the excel.**

**And we have developed these Excel Util Java class files by using org.apache.poi supporting java class file libraries.**

**And once after retrieving input values from the Excel**

**And we are storing all the input data, in the collection objects like List, Set and Map objects**

**And after that, we are we passing those input data to the required Test Methods**

**So that Test Methods will be executed with the required input values**

**Like that, we are following data driven framework, in our project.**

**Coming to Reporting part,**

**We had used TestNG Emailable Reports and Extent Reports.**

**We had used com.aventstack 3rd party plug in to generate extent report.html**

**Once after executing TestNG batch Files, ExtentReport.html report will be generated for every test case.**

**In ExtentReport.html, We are capturing Actual results, Screen shots for passes and failed steps,**

**And covering logs for every verification check point and assertions results.**

**And we are uploading these Test Evidences in SharePoint as per the release specific Folders. So that every can able to access Test Reports from the SharePoint**

**Like that, we have used extent report.html in our project**

**Coming TestNG Frame work, we had used all features like**

**TestNG Features**

**1. We can generate TestNG.xml and can execute Batch File contains set of TestNG class files**

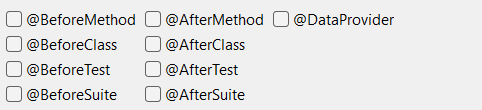
**2. By using TestNG, We can able to execute multiple batch files or suite Files**

**3. TestNG supports Parallel Test Execution of classes, Tests and methods**

**4. We can generate TestNG Emailable Report.html > we can monitor the execution progress**

**5. We can define TestNG Parameters in TestNG.xml . By Using TestNG Parameters > we can able to achieve Browser compatibility Testing .Based on TestNG parameter value scripts will be executed on corresponding browser combination**

**6. TestNG Framework provided so many annotations tags as**

****

**7. We can able to execute only Failed Test cases from TestNG Suite Folder created in Test output Folder**

**8. By using TestNG groups feature, We can able to executed specific group of Test cases**

**9. By using TestNG Priorities feature, We can able to execute methods as per the priority order specified**

**10. By using TestNG Listeners feature , We can able to customize TestNG Emailable Reports , We user ITest Listener to customize emailable reports**

**11. TestNG Data Provider feature provides input data to the test and Test will be executed multiple times for different sets of input data defined in two-dimensional object array**

**12. TestNG Depends on methods feature , executes prerequisite methods prior to execution of main @Test Method**

**13. @Test level , we can write invocation count feature .**

**If we can specify the invocation count value, so that test will be executed those many iterations**

**14. @Test level , we can write is enabled= true or false , if we write as false test case execution will be skipped and if we write as true test case execution will be continued**

15 .TestNG**provides an excellent feature as IRetryAnalyzer**

**By using this feature ,we can**retry**a test case multiple times before declaring it as Failed.**

**Advantages in Page Object model design pattern are:**

**Code reusability is very high:**

**Simply we can import the page class files and we can create the object and we can call the required methods. So, code reusability is very high.**

**Maintaining of object repository is very simple:**

**Adding, modifying and deleting of Web elements at page specific level object repository changes are very simple and easy**

**Understanding of technical flow is very easy:**

**Because we are calling the page specific methods as per the test case functional flows. So anyone can easily understand the technical flow of the functionality.**

**Applying Quick fixes is easy:**

**POM streamlines the process of fixing the code when a website or app is updated. The moment you update the "maps" page, everything clicks into place.**

**Reduced errors:**

**One of the major benefits of Page Object Model is also that it lessens the probability of errors. There is a lower likelihood of mistakes appearing because nothing is written twice.**

**Saves time:**

**Testing becomes quicker because you aren't beginning from scratch each time. For testers, this saves time and effort.**

**Improved maintenance:**

**POM simplifies updates and modifications. You only need to make one adjustment whenever a button moves or the page layout changes**

**Less repetition:**

**You can avoid repeatedly writing the same testing procedures when using POM. For each page or segment, you construct a "map" that you can use again and again. This makes it one of the most advantageous benefits of Page Object Model.**

**Better collaboration:**

**POM facilitates developers' and testers' collaborative work. While testers concentrate on testing, developers create the software. Each component's function is made obvious using this model.**