Student Management System

In my project, I have converting my manual script into the Automation test cases.

For each and every test script I have developed my script into the different-different components.

In our application, there are 3 main modules :- 1. Admin module

2. Teacher module

3. Parent module

Every modules are developed with the combinations of different pages such as Login page, then after we get the Dash Board page, then after we get the Teacher module, Student module, parent module as well as Time-Table, Exam, Events, Friends, Fees, Salary, Subjects, Sub-Routing.

For each and every page of the application, our senior developers and test lead has created separate-separate POM pages, and each and every pages contains numbers of test cases, with different methods, which comes under the Element Utility classes.

Each module contained more than 10 sub-modules and for each and every module we have separate classes, which contains different operations in it. These all modules developed by our Test Lead or senior Developers.

Our framework development has the 3 stages, which are :- 1. Design

2. Implementation 3. Execution

DESIGN : In our 1st sprint, our Test Lead or senior Developers creates our Generic utility packages and Object Utility packages for our framework development and parallely we the test engineers were busy in Testing manually. This was our Design stage of framework.

IMPLEMENTATION : In our 2nd sprint, we have implemented the Utilities like Element Utility class, Generic Utilities and POM pages, which were designed by the our Test Lead or senior Developers.

Generic Utility Class : in the generic utility Package, we have to maintain the Generic Usable classes for Base Class, WebDriver, Database, Excel file, Property file and some interfaces like IPATHCONSTANT.

In WebDriver utility class, we used to maintain the generic utility methods for Select class, Actions class, Frames and Explicit conditions by achieving METHOD OVERLOADING.

With the help of Excel Utilities we were maintaining all the logics to read and write the data from excel sheet.

whare as in File Utilities, we were maintaining all the read and write logics from Property file, because in Property file contains all the Static and Final values which were constant.

Base Class used to maintain the separate logic to Connecting and Dis-Connecting the Database, Launching and Closing the browser, Parallel execution, Launching the Url and Dis-Connecting the Url.

We were creating the objects for the remaining utilities inside the base class, because while writing the script we used to extend our test class, which we got the indirect access to all the methods inside the utilities.

In the OBJECT REPOSOTORY, we used to maintain the web elements to related web pages in separate class, by making these PRIVATE to achieve the ENCAPSULATION.

* To initialize those web element we have to pass the Driver reference variable inside the constructor of the current class.
* As per the situation, sometimes we have to create our Business login insight the Test Script itself or sometimes inside the Test Script itself.

Script Development : when it comes to script development, we EXTENED our Test class to Base class for INHERITING the property of base class in Test class to inherite the property and used our script by @test annotation.

For each @test annotation we used to create Non-static method inside, which we used to write the script by utilizing Base Class and utility classes.

When it comes to perform some actions we used to create the Object of the specific page present inside the object repository by passing DRIVER reference variable in it, by that it can invoke the constructor present inside each page and help to initialize the web elements to perform actions as per the business logic.

When it comes to script execution, 1st we create the suite xml file and then we used to perform the Batch execution.

At the time of full Regression execution, at that time when we have the group of smoke test case, then we don’t have sufficient time then we had to perform Parallel execution for compatibility testing.

To track the each and every instance of test execution we have the @listeners at Test level and Suite level.

So to get the Test Execution Report, then we have to implement the @listeners using implementation class, where we used to override the method.

So at each and every instance before starting of the execution of test script, till it sent, it will invoke the particular method according to the instance.

When we have a failed test script at that time “Test Failed OverRide method” will get invoke after this TakesScreenShots method will comes in the picture, which has the property to capture the screen shot of the Page at which our method got failed.

After done with the script execution, we used to generate Extend report for that we used to maintain separate class to write code and that we maintain in the Generic utility method.

We also implemented TestNG framework in our test script for the script optimization, for that purpose we had created and extended Base class to our Test scripts.

With the help of IRetryAnalyser, we can rerun our scripts for desired number of times, when our test script gets failed due to internet issue.