**Framework type – TDD**

**Language/tools used – Java, TestNG, Maven**

**Folder/Package Structure/ Framework Details:**

* InputData(Stored excel file to provide data to scripts)
* PageElements(Interfaces to store xpaths as per page)
* TestCases(Actual Test cases as per page)
* UtilityClasses(Common functions has been written)

**Input Data Reading –**

* Browser and respective driver path can be configure as per requirement through **Data.xlsx** file.
* Test Cases inputs can be customized through same excel.
* Both Browser and Test Case data configuration are separated in two different sheets. i.e. **‘BrowserSettings’** and **‘FilterData’**

**PageElements->IStatisticsPage.java**

* String type of Final variables are declared and assigned with xpath
* Designed interface, keeping in my mind that if in future any other page class need to use already exist elements from various pages then they can implement multiple interfaces.

**UtilityClasses->CommonFunctions.java**

* Class to write some reusable codes
* Currently wrote some utilities such as ‘getElement’ – to get WebElement, ‘getRowData’- to read excel data just by passing sheet names

**TestCases->TestSortFilter.java**

* Class to write actual test cases which can utilize utilities and interfaces to perform test steps.

**TestNG->testng.xml** – Test cases configuration for execution and report

* You can get ‘index.html’ for detail report in ‘/test-output’ folder

**Pom.xml** – All maven dependencies has been mentioned

**Limitations-**

Due to time constraints below are some limitation in this code-

* Log file integration is not available however, console will have all appropriate information during execution
* Screenshot Utility is not available
* Excel path had to configure due to some local machine issue

**Use Cases use for Automation**

