**Task 1 - UI Automated test using Java, Selenium WebDriver**

**A Java project is created with TestNG framework for automating test cases using Selenium WebDriver. Please find the details below.**

1. **Project name: Sample project**
2. **Framework: TestNG**
3. **Language: Java**
4. **Automation tool: Selenum WebDriver**
5. **Reporting structure: TestNG report, Extent Reports**
6. **File source: Microsoft Excel**
7. **Parallel execution: TestNG**

**Features implemented**

1. **Java project with TestNG framework**
2. **Page object design pattern – A class file is created for each page in the application which consists of all the objects identified in that particular page along with methods**
3. **Common Function class – A class file is created which contains basic functions containing selenium commands like click(), sendKeys() etc. The purpose of this class file is to customize the selenium commands to functions. If there are any changes in any of the basic selenium commands, then we can directly edit those in this class file rather than going to each test case and changing it.**
4. **Extent Reports – The reporting is implemented using a third-party tool called Extent Reports. HTML reports are generated for each execution. There will be pass, fail, info status and screenshots are included whenever there is a failed case.**
5. **External file source – Data can be read from excel files**
6. **Parallel execution – TestNG supports parallel execution. TestNG.xml file is created for implementing this. Also, usage of @priority annotations decide the priority in which test cases are to be executed.**
7. **Cross browser testing – The testcases can be executed in different browser (IE, Chrome and Firefox). The corresponding driver files are included in the project.**
8. **Different environment (URLs) – URLs can be read from excel files, so test cases can be executed in different environments.**
9. **Automation of 3 selenium test cases using Selenium WebDriver**

**Project folder**

**Name of project folder: Sample project**

**Packages:**

1. **sample.drivers – This package consists of the .exe files for drivers. Chrome driver, Internet Explorer driver and Gecko driver is present**
2. **sample.pages – This package consists of class files representing the different pages in application. Since three test cases are given there are three files corresponding to each named as Sign In.java, LoginAccount.java and Checkout.java**
3. **sample.testcases – This package consists of one class file named as Testcase.java which contains the test cases for executing the Sign in, Login and Buy a tshirt flow**
4. **sample.testdata – This package contains the test data file. Since excel file is used ,it is present inside this package with the name Testdata.xlsx**
5. **sample.testhandlers – This package consists of two class files.**

**Utilities.java is a file that is considered as the common class library which includes customized reusable functions.**

**Reports.java is a file that contains code for implementing the reporting structure using Extent Reports. Reports are saved in the location C://HelloFresh//Reports**

**JRE System Library – This library consists of jar files that are required to run a Java program which is by default added by Eclipse**

**Referenced Libraries:**

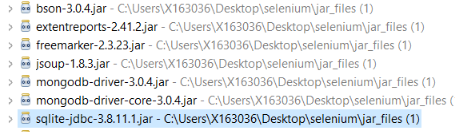
1. **Selenium jars- The following lists out the jars used in Selenium**



1. **org.apache.commons.io – This package is used for working with streams, writers, readers and files**



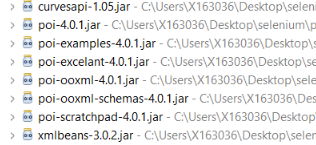
1. **Extent Reports – The following set of jars are used for Extent Reports support**



1. **The Apache poi jars helps in creating modifying and displaying MS Office files.**

**The following are the jar files included in it**





**Test output folder – This folder contains the TestNG reports**

**TestNG.xml – This xml file is used for implementing parallel execution**

**Task 2 - API Automated test using Java**

**For the webservice automation, three class files are created inside the package sample.api. The classes are named as Function1, Function2 and Function3.**

1. **Function1 – This class file contains the program to fetch all the responses from the given webservice link and then it checks the existence of the countries US, DE and GB**
2. **Function2 – This class file contains the program to validate each country separately for US, DE and GB and their response codes are printed in console. Also, it checks the existence of an invalid country and its response code**
3. **Function 3- This class file contains post method to post a json body for a new country code**