***Introduction:***

The attached project is a Java-based framework I have created for functional test automation proof of concept of an application (bestbuy.com).

For the purposes of this assignment, it might be useful to go through the \src\supportingFiles for supporting functions I have coded in Java.

You can specifically take a look at FileUtility.java where I have functions to read test data from the excel sheet in framework, read the sheet which serves as an object repository, create step log for each step of the execution, etc.

This project is also available on github (public access for the time being)

***https://github.com/rtsuresh/BestBuy***

**More details on how to run the framework as a whole given below:**

***Pre-requisites:***

JDK version 1.6 or higher, Eclipse (Required JARs are referenced in the build path internally), Chrome Browser version 79

***About the framework:***

1. **Data** from a Test data sheet in excel format(.xlsx) in the <ProjectPath>->TestData->TestData.xlsx
2. **Test cases** are currently JUnit files under the package roofandFloorTestCases. The framework can accommodate more than one file, and we can organise this by functionality
3. **Objects** are defined in an Excel sheet in the <ProjectPath>->Objects->ORSheet.xlsx
4. **Report** / Step Log is in notepad format under <ProjectPath>->StepLog->StepLog <TimeStamp>.txt . This can be modified to a different format later.
5. **Supporting Files package** contains
   1. Business Functions – Contains business-logic functions (application specific)
   2. Utility functions -Contains s screen-level actions (non-application specific)
   3. FileUtility – Contains functions interacting with IO files. (Excel/notepad)
   4. Reporting-related functions
   5. Variables (for global usage)
6. **All the necessary JARS** are in the <ProjectPath>->jars and referenced
7. **Chrome Driver is also in the JARS folder**

***How to Run:***

1. Import the project into Eclipse
2. Open bestBuyTestCases package -> SearchFunctionality JUnit class file.
3. Run the JUnit file
4. Go to the StepLog folder for report logs. For every run, there will be a separate report.

***Future scope:***

1. **Reporting:** 
   1. There needs to be more flexibility provided here in terms of options for reports. Can include Excel and HTML
   2. Need to include screenshots for failure
   3. Can discuss options about a summary report.
2. **Test-driving engine:**
   1. Currently, I am using JUnit to drive the whole script. This entire approach can be changed and a hand-crafted driver script can be written with extensive reporting.
3. **Miscellaneous:** 
   1. More wrappers in the driver/JUnit file.
   2. Multi-browser capability
   3. A configuration sheet to give common details
   4. Currently, XPath is the only locator being used, this can be extended to any locator by a very small code change
   5. Need to create more functions to accommodate all the required scenarios. Currently, I have created only the bare basic required functions in both Business and Utility-level.