Author: Hemant Mahalpure

Date of creation: 27-03-2022

**Introduction to the Application:**

This is the ‘Be Cognizant’ application which is basically a Journal of the Cognizant. It has various insights about things which are going around cognizant. It has all the apps useful for various requirements.

**The testcase to be automated is:**

1. Launch any browser ex: Chrome/Firefox.
2. Open the application by invoking the following URL: <https://be.cognizant.com/>
3. Complete the sign in process to access the page by using valid credentials.
4. Get the user information from top right corner on the home page of the application.
5. Get the current stock price of the Cognizant which is displayed in header section on the home page of the application.
6. By opening new window, get the stock price of the Cognizant from Google.
7. Compare the two values of the stock price and display the message accordingly.
8. Click on ‘APPS AND TOOLS’ logo present in header section on the home page of the application.
9. Click on ‘OFFICE 365 APPS’ link present on the ‘APPS AND TOOLS’ page.
10. Display the Office 365 apps on the console.
11. Click on ‘COMPANY APPS’ link present on ‘APPS AND TOOLS’ page.
12. Display the Company apps on the console.
13. Close the browser.

**Framework used:**

We have used Selenium web driver 3.141.59, Java, Maven Project to complete the automation for the case study. We have used POM framework to create method libraries. We have implemented the testcases using TestNG suite.

**Project Framework:**

Graphical user interface, text, application

Description automatically generated

**Folder Structure in Framework:**

The maven project is “Main Project”

It is having different repositories for different components in framework.

1. src/main/java > FunctionLibraries is the package.

It contains three classes for 3 different scenarios.

2. src/test/java > ‘Scripts’ is the package.

It contains a single test class used for calling methods.

3.List of source folders in the framework is:

Drivers: This folder contains chrome and firefox driver.

ScreenShots: This folder contains the automated screenshot.

ExcelSheet: This folder contains excel sheet created as a result of

writing data to excel sheet.

ExtentReport: This folder contains the html report created after

execution of the project.

ObjectRepository: This folder contains the URL and locator

Information in properties file for 3 test scenarios.

Testcase: This folder contains the test case in excel file.

4.testng.xml > This file contains the class which we need to run as

TestNG suit.

**Steps to execute the script inside framework:**

Below are the steps which we need to follow while execution.

Step-1:

Open pom.xml, check the required dependencies.

Text

Description automatically generated with medium confidence

Clean and build the maven project from project menu of eclipse.

Crosscheck the content of the jars in “Maven dependencies” folder.

Text

Description automatically generated

Step-2:

Refresh the entire project.

Step-3:

Execute the entire project by running testing.xml file as a suite

**The console report is attached as below:**

Text

Description automatically generated

**The results of running suite is attached as below:**

Graphical user interface, text, application

Description automatically generated