**Case Study for Auto Project**

**Problem Statement**

**LINK:**

<https://lexus.roimotors.com/new/LEXUS/2018-LEXUS-GS+350-9478ae060a0a00f93a13f68a2e7fb412.htm>

**ASK:**

* With the link provided, create a test and upload your code to GitHub so that we can download your solution and run it locally.

**GUIDELINES:**

* Use the Page Object Model (if you are familiar with it) to write this test.
* Write the test in whatever language you want
* Make sure that the test runs against Chrome and Firefox
* Make sure that the code is easily downloadable and easy to run locally.

**TEST PROBLEMS:**

* As a customer you have $51,000 to offer a dealer for this vehicle.  Update the Offer Price field to $51,000.
* As a customer you only have $2500 as a down payment and your credit score is 700.  Update the Cash Down and Credit Score field to display the correct values.
* As a dealer I need the customer information in order to contact them regarding the quote.
* After clicking on Ready To Send you will need to do the following:
  + Enter a Zip Code
  + Create a simple method to randomly generate a 'n' character string to use in the next step.
  + Enter the First Name, Last Name and Message (using the method created above)
* Enter an Email and Phone Number
* As a customer I would like to confirm that the dealer is getting the correct summary of pricing from me.  Verify that the Cash Down value is correct on the right side of the screen
* As a customer I would like to confirm that my offer has been sent.  Click on Send My Offer and wait for the 'Protect Your Vehicle' page to be displayed

**Solution**

As per above Test Problem, below are 2 Test Cases for the Problem

Test Case 1: Verify if the User offer has corrected Down Payment value

Test Case 2: Once the User details are filled in successfully, verify if the User form is sent to Dealer successfully and ‘Protect your Browser’ page is displayed.

**Automation Project Details:**

The test automation is done using below tools and technologies:

1. Selenium Webdriver (version 3.11.0)
2. Chrome driver (For Chrome)
3. GeckoDriver (For Firefox)
4. Eclipse IDE
5. Java Coding language
6. TestNG for Reporting

**Packages and code Structure:**

1. BaseClass - This program has the basic method to open and close the browser with proper settings.
2. Pages – This package contains page factory for 2 pages involved in the Test cases. Classes for each page has Web elements and methods/activities on the elements.
3. Action – This package contains the methods pertaining to the test cases for each page involved in the test case.
4. TestCases – This package contains the actual test cases along with TestNG annotations.
5. Testng.xml – This is a executable file through eclipse which has browser settings and test execution settings for test cases.

**Below are the Setup instructions for execution of the Test Automation Script**

In order to execute the code below Setup is required:

1. Java (Current version)
2. Eclipse IDE for Java
3. TestNG (Install from Eclipse Marketplace - TestNG version 6.8.0)
4. Maven
5. Chrome Browser (Version 66.0.3359.139 or higher)
6. Quantum Firefox Browser (version 59.0.3 (64-bit))

To execute this Program, please follow below steps:

1. Download the project from Github in local folder.
2. Open the downloaded project in Eclipse IDE
3. Once the project is open, right click on the project and select Maven, select "Update Project". Select "Force Update of Snapshots/Releases" and click OK.
4. Once the project is built, right click on testng.xml and Run the code using “TestNG Suite”
5. Check the results in folder test-output/Cox\_Auto\_Assignment. Check 2 reports for 2 Test suites in html. Open HTML files in Web browser.