**Test Automation Bootcamp**

Automation Scripting using WebDriver Exercises

1. **Automate the following manual test case**

|  |  |  |
| --- | --- | --- |
| **Step** | **Expected Result** | **Test Data** |
| Launch Browser & Navigate to [www.costco.com](http://www.costco.com) | Costco Home Page appears | URL: [www.costco.com](http://www.costco.com) |
| Click on ‘Beverages & Water’ found within the ‘Grocery’ menu | Set Delivery Zip Code Page appears, or Bakery & Desserts Page appears |  |
| Update Deliver Zip Code to 22031 if page appears | Beverages & Water Page appears | Zip Code: 22031 |
| Select ‘5-hour Energy’ from the Brand Filter | ‘5-Hour Energy Beverages & Water’ header appears |  |
| Add the 1st product to the shopping cart *(you may add the 2nd product if the 1st is sold out or unavailable)* | ‘1 in Cart’ message appears |  |
| From the Search bar at the top of the page, search for ‘Breakfast’ | ‘We found \_\_\_ results for “Breakfast”’ header is displayed | Search Item: Breakfast |
| Display the names of all products displayed on the 1st page of results | ‘Syrup’ must be a part of at least 1 product’s name that is displayed | Text: Syrup |
| Add the 1st Item to the shopping cart | Cart icon displays ‘2’ indicating that 2 items are in the cart  \*If the ‘Added to Cart’ alert appears, click on ‘Continue Shopping’ | Cart Items: 2 |
| Click on the Cart | ‘Cart’ page appears |  |
| Increase the quantity of the 1st product in the cart | The Subtotal amount is increased by the price of the 1st product (i.e. if subtotal is $10.00 & product price is $5.00, then new subtotal should be $15.00) |  |
| Remove the 2nd product from the cart | The Subtotal amount is decreased by the price of the 2nd product (i.e. if subtotal is $10.00 & product price is $8.00, then new subtotal should be $2.00) |  |

**NOTES:**

* The website has many sync issues that you will need to work around. You must use implicit and explicit waits as needed. You may use ‘Thread.sleep’ for a maximum of 8 times through the entire script.
* An adequate message(s) must be printed in the console for each Expected Result
* You must identify an opportunity to use the *JavaScriptExecutor’s* ***executeScript*** method at least once during this test script and implement it successfully. The method can be used to perform any action required by the test case (*highlighting an element is not something that is required by the test case*).
* There is no need to provide a pass/fail message, but you may implement it if you’d like
* You will not need any other APIs except for Selenium WebDriver to complete this exercise

1. **Cars.Com Exercise using Excel**

This exercise requires you to read test data from Excel, input it into the application under test, and then manipulate the application.

Steps to follow:

1. Navigate to [www.cars.com](http://www.cars.com)
2. Using the data provided in the relevant excel sheet, input the appropriate values to search for a car
3. On the Search Results page, capture all of the prices displayed on the 1st page of search results and print out the average car price to the console.
   1. Average = (price1 + price2 + … + priceN) / number of prices
   2. The print-out should read: **\_\_\_\_\_\_\_\_ is the average price of the “Used Cars” Car Search Results**
      1. The “Used Cars” is coming from Excel under the *Stock Type* column. It will change depending on the row of data being passed to the application.
      2. The console must show the quotation marks ( “ “ ) when printed
4. Capture the number of Models available from the filters panel for the specified Make along with their names. Display the total number of models as well as each model available as part of the filter.
   1. The print-out for the number of models available should read: **<Make> has \_\_\_ models to choose from**
   2. <Make> is coming from Excel under the *Make* column. It will change depending on the row of data being passed to the application

**NOTES:**

* Majority of the locators used must be CSS Selectors. You may use a maximum of 4 Xpath locators. All locators used to identify the elements on the 1st page must be CSS Selectors.
* We have seen multiple ways to implement reading data from excel. Implement the reading from excel action as you deem appropriate. There is no singularly correct way of doing this.
* The code to read from Excel must be in a separate class file from your test script.
* Implement Logging at each relevant step. The limit of the log file size must be 50,000kb.
* You will need the necessary APIs for Reading from Excel, along with Selenium WebDriver to complete this exercise. You will also need the necessary APIs required for logging.