**Project – FlinkAutomation**

* Automation Tool - Selenium
* Programming Language –Java 8.0
* Framework Approach – TestNg with Page Object Model
* Chrome Browser version – Version 96.0.4664.45 (Official Build) (64-bit)
* Edge Browser version – Version 96.0.1054.43 (Official build) (64-bit)
* Chrome Driver version - 95.0.4638.69
* Edge Driver version - 96.0.1054.43

**Information about Different Packages**

1. **weatherShopperPages**
2. **WeatherShopperHomePage** – This is a page class for Current temperature.
3. **ProductsPage** – This is a page class for Moisturizer and Sunscreen pages.
4. **CheckOutPage** – This is a page class for Cart Items page
5. **PaymentPage** – This is a page class for Stripe.com Example charge
6. **ConfirmationPage** – This is a page class for Confirmation page
7. **weatherShopperTest**
8. **AssignmentTC1** – In the Test Case, I am performing the end to end automation from the step of

* Checking current temperature
* Selecting product based on temperature
* Adding the least expensive product in cart
* Verifying product in cart with those added in it
* Filling Credit Card details and proceeding to payment page
* Verifying Order success or failure

1. **AssignmentTC2** – This test case is same as above. The purpose of creating this test case to show the capability of my framework to execute test cases in parallel.
2. **BaseTest** – This test case is created to help in generating the Extent report.
3. **weatherShopperUtil**
4. **BaseSetup** – This class is used to

* Instantiate the respective browser driver
* Instantiate the log4j.xml
* Opening the URL and validating it

1. **ExtentReporterNG** – This class is used to configure the Extent Report as per user need.
2. **Log** – This class is used to configure the Logger as per user need.
3. **Utility** – The purpose of creating this class to accumulate all the utilities developer/tester needs while performing automation. Different methods created in this class are –

* **waitForEle** – Wait for Element till user defined timeframe
* **sortByValue** – Sorting of hash map based on values

**Different XMLs used in Project**

* log4j.xml
* testing.xml
* pom.xml

**Different Browsers used in Project**

* Chrome
* Microsoft Edge