Documentation and Assumptions

**Challenge:**

**Go To URL:** http://automationpractice.com/index.php Create an automated smoke test suite for the below Acceptance Criteria

**Acceptance Criteria**: I want to be able to add 2 items to the cart and place an order

**Assumptions**: The above acceptance criteria is high level and can be further divided into 4 different user journeys and below are the details

User Journey 1: As a Registered user, I want to search for the products and place an order.

User Journey 2: As a Guest user, I want to search for products and place an order.

User Journey 3: As a Registered user, I want to select first product from mega menu and second product from search and place an order.

User Journey 4: As a Guest user, I want to select first product from mega menu and second product from search and place an order.

Please Note: The above user journey can be automated in two different flows

**Flow 1**:

Place Order

Add Products To Cart

Signin

**Flow 2**:

Place order

Signin

Add Products To Cart

**Conclusion**: To perform the above challenge, I have selected the User journey 1 and flow 2 from the above Assumptions.

**API Challenge**: Open Weather Map is free to use the website (and JSON API service) that can be used to query the weather forecast of major cities around the world. More information can be found

here: http://openweathermap.org/forecast16

Below Acceptance criteria:

**Scenario**: A happy holidaymaker

**Given** I like to holiday in Sydney

**And** I only like to holiday on Thursdays

**When** I look up the weather forecast

**Then** I receive the weather forecast

**And** the temperature is warmer than 10 degrees

**Assumptions**: The API services comes with the subscription and unfortunately, I couldn’t able to find the service request for above acceptance criteria even after subscribing to the weather API. So I have automated the below acceptance criteria to accept the API challenge. And below is the acceptance criteria.

Acceptance criteria:

**Scenario**: A happy holidaymaker

**Given** user is able to access weather api

**When** user send request for "city"

**Then** I receive the weather forecast as "status"

**And** user should see the response contains "city"

**Framework Documentation**:

**Description**: The framework is designed to execute both UI and API tests using Selenium WebDriver,BDD Cucumber and programming language as Java and Maven as build tool.

**Design Techniques**:

Automation Tool: Selenium WebDriver

Framework: POM(Page object Model)

Approach: BDD

Test Framework: Cucumber with JUnit

Api Test Framework: Rest Assured

Reporting: Extent, Pretty

Acceptance Criteria Execution:

* Execution starts from Home page
* It will search for the products
* Then it will look at the result page and click on a product
* Test will go to product details page
* Then product will be added to cart by clicking on add to cart button
* And above steps will be repeated again after clicking “Continuing shopping”
* Test will be proceed to checkout after adding two products to cart
* And order will be placed with registered user

**How to Execute the Test**:

1. Via Feature File: To run the test via feature file, go to src/test/resources/features and right click on the any feature file and select Run As>cucumber.feature file.
2. Via JUnit Test: To run the test via JUnit tests, go to src/test/java/junitTests and right click on the any JUnit test and select Run As>JUnit Test.
3. Via Command Line: To run the test via command line, go to root directory where pom.xml is saved and run the command “mvn test”.

**Conculsion**:

* I have implemented the best practices and approach using open source tools to solve the QA challenge. The methods designed and implemented in the frame work are reusable and robust.
* To deliver best quality code and results in short time frame, I have selected the above mentioned user case to automated the functionality, but there are other best practices I can introduce like Data Tables, independent Step Definition class,maven profiles for independent API,UI tests etc for more robust and reusable framework( if given more time).