

Framework and Approach

Problem Statement:

This framework provides basic utilities for setting up “Amazon end-to-end test scenario” to cover a customer journey on Amazon website. This automation framework has been developed, tested using Selenium Framework with Cucumber BDD. This document gives an overview of implementation and approach for the solution.

Framework Overview:

This framework is based on Page Object Model pattern of Automation design with clear separation of responsibilities spread among three layers. The trigger point of the automation scripts is the feature file which accommodates all the tests in the gherkin standard format which is the - 'Given', 'When' and 'Then' statements e.g.

Scenario Outline: As a customer I should be able to Login in Amazon, search for an item, add to cart, proceed to checkout and provide my personal and dummy payment details to purchase an item.

Given I want to open "<Browser>" and launch Amazon URL

And I click on Sign-In button

And I enter Username with Password

When I search for an item

And I select the item

And I click on Add to Cart button

Then I validate the Title "<Shopping Cart Title>"

And I verify item is added to cart

And I click on Proceed to Checkout button

And I Enter the Card Details

Examples:

| Browser | Shopping Cart Title |

| chrome | Amazon.com.au Shopping Cart |

Framework & Approach for Selenium Cucumber Solution- Author: Anupam Mishra

Three layers of the framework are –

Feature Definition Layer -

At this layer, all the features/keywords used in the feature file are implemented. This layer does not have any business/test logic. This layer makes sure that to implement a particular keyword, which functions of the “function layer” need to be called and in which sequence.

Keyword Functions Layer -

This layer implements all the business logic or test logic. The methods in this layer serve different calls from Feature Definition layer and perform activities which are needed to be done as per test logic. But still no screen interactions are performed in this layer. Whenever any screen interaction is needed in this layer, call is made to third layer to serve the purpose.

Page Object Layer -

This layer handles the actual screen interactions and provides convenient methods which serve to the calls from Keyword Functions layer. This layer does not have any test logic implementation, but only screen interaction methods. This is the place which stores all the X-Paths or CSS Selectors for any particular page/screen. So, if during the process of development of the page, if there is any change in the xpath or css selectors, then this is the only layer which needs to be modified, without any disturbance to the other two layers.

Automation framework provides utility methods at all of the three layers. At the first layer, there are few background tasks for automating any project - like opening the browser (any particular browser as per need), setting up URL (as per environment requirements), logging in the application if needed and setting up the webdriver etc .