Feature: Amazon product search

As a user

I want to search for a product on Amazon

So that I can find the product's details and price

Scenario: Search for a product on Amazon and retrieve its details and price

Given I am on the Amazon homepage

When I search for "Samsung Galaxy S21"

Then I see a list of search results

When I click on the first search result

Then I see the product details page

When I navigate to the product reviews section

Then I see the dynamic webtable in the following column:

| Reviewer name | Review title | Review date | Rating | Comment |

And I retrieve the data from 5 rows of WebTable

import { defineSupportCode } from 'cucumber';

import { Builder, Capabilities, WebDriver } from 'selenium-webdriver';

import { Options } from 'selenium-webdriver/chrome';

import assert from 'assert';

import fs from 'fs';

import { After, Before, Status } from 'cucumber';

let driver: WebDriver;

//step definition initializes the Selenium WebDriver instance, opens the Amazon homepage in the Chrome browser

Given('I am on the Amazon homepage', async function () {

driver = await new Builder().forBrowser('chrome').build();

await driver.get('https://www.amazon.com/');

});

// this step definition searches for the “searchQuery” parameter on the Amazon homepage by finding the search box element and simulating the user's input

When('I search for {string}', async function (searchQuery: string) {

const searchBox = await driver.findElement(By.id('twotabsearchtextbox'));

await searchBox.sendKeys(searchQuery);

await searchBox.sendKeys('\uE007'); // simulate pressing Enter key

});

//this step definition checks if a list of search results is displayed on the search results page of the Amazon website by finding the search results list element and checking if it is displayed using an assertion

Then('I see a list of search results', async function () {

const searchResults = await driver.findElement(By.css('.s-result-list'));

await assert(searchResults.isDisplayed());

});

//clicks on the first search result on the search results page of the Amazon website by finding the first search result element and simulating a user click action

When('I click on the first search result', async function () {

const firstSearchResult = await driver.findElement(By.css('.s-result-item:first-child h2 a'));

await firstSearchResult.click();

});

//checks if the user has been redirected to the product details page on the Amazon website by getting the current URL of the page and checking if it includes the string '/dp/' using an assertion

Then('I see the product details page', async function () {

await assert(driver.getCurrentUrl().includes('/dp/'));

});

// navigates to the product reviews section on the product details page of the Amazon website by finding the link to the product reviews section and simulating a user click action

When('I navigate to the product reviews section', async function () {

const reviewsLink = await driver.findElement(By.id('reviewsMedley'));

await reviewsLink.click();

});

//checks for the presence of a dynamic webtable on the product reviews section of the Amazon website by finding the dynamic webtable and comparing its column names to the expected column names using the assert module

Then('I see the dynamic webtable in the following column:', async function (dataTable: any) {

const expectedColumns = dataTable.raw()[0];

const table = await driver.findElement(By.css('#cm-cr-dp-review-list'));

const actualColumns = await table.findElements(By.css('thead th'))

.then(headers => Promise.all(headers.map(header => header.getText())));

await assert.deepStrictEqual(expectedColumns, actualColumns);

});

//This step retrieves the data from the dynamic web table on the product reviews page of Amazon's website. It takes an integer argument numRows that specifies the number of rows of data to retrieve from the web table

Then('I retrieve the data from {int} rows of WebTable', async function (numRows: number) {

const table = await driver.findElement(By.css('#cm-cr-dp-review-list'));

const rows = await table.findElements(By.css('tbody tr')).slice(0, numRows);

for (let row of rows) {

const cells = await row.findElements(By.css('td'));

const reviewerName = await cells[0].getText();

const reviewerTitle = await cells[1].getText();

const reviewerDate = await cells[2].getText();

const rating = await cells[3].getText();

const comment = await cells[4].getText();

}

});

// terminate the WebDriver instance and close the browser window

After(async function () {

await driver.quit();

});