AUTOMATION TESTING FOR A WEB APPLICATION

FINAL PROJECT REPORT

Submitted by,

ARJUN M

DIGITAL ACADEMY – CERTIFICATION PROGRAM IN SOFTWARE TEST ENGINEER

OCTOBER 2023

TABLE OF CONTENTS

ABSTRACT

In this project, the results and findings of our Cucumber with Selenium automation project is presented. The project aimed to automate the testing of demo nopCommerce website which is the Application Under Test (AUT). Through the use of Cucumber for behavior-driven development (BDD) and Selenium for web automation, and successfully achieved the automation tests.

INTRODUCTION

OVERVIEW

This project is about testing a web application using automation scripts using various selenium commands the project is built as a Maven project and this project uses Cucumber framework.

SIGNIFICANCE

This project also includes Data Driven Testing in which a Test data is read from Excel and imported into the project to check whether the login application works based on the inputs from Excel sheet. This is done using Apache POI, Cucumber framework.

SCOPE OF THE PROJECT

The scope of the automation project includes the functionalities as follows purchasing a product from the ecommerce website this includes the complete process of purchase, and checking the login functionality of a different web application.

OUT OF SCOPE

The Login functionality of invalid data is not tested and not covered in this automation scripts.

METHODOLOGY

TOOLS AND TECHNOLOGIES USED

Cucumber for BDD

Selenium WebDriver

Programming Language: Java

Eclipse IDE

Apache POI

OVERVIEW OF TEST FRAMEWORK

Objective

The primary objective of this framework is to automate testing of a website using Cucumber for Behavior Driven Development and Selenium for web automation. It allows write readable scenarios, automate web interactions and generate comprehensive test reports.

Components

1.Cucumber

Feature Files: Written in Gherkin, feature files describe test scenarios in a readable format

StepDefinition: These files contain the automation code that interprets Gherkin steps and interacts with the web application using Selenium.

Runner Class: Configure test execution and specifies which feature to run.

2.Selenium WebDriver

Selenium WebDriver is used for automating web interactions, it provides APIs to interact with web elements, navigate through webpages and perform actions like clicks, input and verification.

3.TestData

Input data, test data or test parameters required for test scenarios. This data can be stored in separate files or directly in feature files.

4.Reporting

Reporting tools like Cucumber's built in reports or external tools like Extent Reports for generating detailed test reports with pass/fail status, screenshots and logs.

Test Workflow

Test Scenario Definition:

- QA/Testers write feature files using Gherkin language to define test scenarios.
- Each scenario consists of Given, When, Then, And steps that describes the preconditions, actions, and expected outcomes.

Step Definitions:

- Step definition classes is created to map Gherkin steps to code.
- These classes use Selenium WebDriver to automate interactions with the website.
- Data Driven tests may involve fetching test data from external sources.

Test Execution

- A test runner class configures test execution.
- It specifies which files to run and may set up the test Environments.
- Cucumber executes the scenarios, invoking corresponding step definitions.
- Selenium interacts with the website, performing, actions and verifications.

Reporting

- After test execution the framework generates detailed reports.
- Reports include information about passed and failed scenarios, step by step logs, screenshots.

Test Scenarios

Scenario 1:

Check URL

User enters the URL the homepage is displayed.

Scenario 2:

Search for a Product in the search box

- User clicks on the search box
- Enters the product name
- Clicks Search button

Scenario 3:

Select the product to purchase

- User selects the product
- User then selects the quantity of the product
- Clicks add to cart button

Scenario 4:

Checking the shopping cart

- The notification of adding the product to the cart is checked
- Selects the shopping cart from menu
- Select wrapping option for the product
- Select Terms of service

Scenario 5:

Checkout details:

- User selects checkout as guest option
- User adds the necessary details into the form
- User selects the shipping method and payment process
- And order is confirmed

TEST ENVIRONMENT

Hardware and Software requirements

Hardware

1. Computer:

Processor: i5 core
RAM: 4GB or more
Storage: Sufficient free disk space for your project and test data.

2. Web Browser

Chrome or Firefox or any browser

3. Internet Connectivity

Software

1. Operating System: Windows
2. Java Development Kit
3. Integrated Development Environment (IDE): Eclipse
4. Selenium WebDriver
5. Cucumber
6. Dependencies

Configuration setup

- Add Selenium WebDriver - Selenium- java dependencies
- Add Cucumber Dependencies
- Create Cucumber feature files
- Create Step definitions
- Configure Test runner

TEST EXECUTION

- A test runner class configures test execution.
- It specifies which files to run and may set up the test Environments.
- Cucumber executes the scenarios, invoking corresponding step definitions.
- Selenium interacts with the website, performing, actions and verifications.

Challenges and Considerations

Setup and Configuration: Initial setup can be time consuming.
Maintenance Overhead: As the project grows, maintaining step definitions and handling changes in the website's structure may require effort.
Test Data Management: Managing test data, especially for data driven testing can be complex

IMPLEMENTATIONS AND RESULTS

Feature File
Test.feature

```
Feature: Automated Tests

  Scenario: User place an order of an item from search
    Given User is on Home page
    Then Getting the title of webpage
    And Getting the current url of the website
    When User searches for Speakers
    Then Selects the item to buy
    And The product is added to cart
    And Checkout from cart
    And Enters details on checkout as guest page
```

Test Runner

runners (package name)
FinalTestRunner.java

```
package runners;
import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
@RunWith(Cucumber.class)
@CucumberOptions(
                        features =
{"src/main/java/Feature/Test.feature"},
                        //publish=true,
                        glue = {"stepDefinitionsSteps"},
                        plugin = {"pretty",
"com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapt
er:"},
                        //monochrome=true)
                                publish=true)
public class FinalTestRunner{}
```

Step Definition

stepDefinitionsSteps (package name)
FinalTestSteps.java

```
package stepDefinitionsSteps;
import  org.junit.Assert;
import org.testng.asserts.SoftAssert;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.And;
```

```
public class FinalTestSteps {


	WebDriver driver;
	@Given("User is on Home page")
	public void user_is_on_Home_Page() throws InterruptedException{

	System.setProperty("webdriver.chrome.driver","C:/Users/Arjun/Downloads/ch
romedriver-win64/chromedriver-win64/chromedriver.exe");
		driver = new ChromeDriver();
		driver.manage().window().maximize();
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
		driver.get("https://demo.nopcommerce.com/");
	}
	@Then("Getting the title of webpage")
	public void getting_the_title_of_webpage() throws InterruptedException{
		String actualTitle = driver.getTitle();
		String expectedTitle = "nopCommerce demo store";
		Assert.assertEquals(actualTitle, expectedTitle);
		System.out.println("Assert passed");
		Thread.sleep(2000);
		}
	@And("Getting the current url of the website")
	public void getting_url() throws InterruptedException{
		SoftAssert softAssert = new SoftAssert();
		String actUrl = driver.getCurrentUrl();
		String expUrl = "https://demo.nopcommerce.com/";
		softAssert.assertEquals(actUrl, expUrl);
		WebElement web =
driver.findElement(By.xpath("/html/body/div[6]/div[3]/div/div/div/div/div[4]/di
v[2]/div[3]/div/div[2]/h2/a"));
		String actText = web.getText();
		String expText = "HTC One M8 Android L 5.0 Lollipop";
		softAssert.assertEquals(actText,expText);
		softAssert.assertAll();
	}
	@When("User searches for Speakers")
	public void searches() throws InterruptedException {
		driver.findElement(By.id("small-searchterms")).sendKeys("Portable
Sound Speakers");
		driver.findElement(By.xpath("//*[@id=\"small-search-box-
form\"]/button")).click();
		Thread.sleep(2000);
		}
	@Then("Selects the item to buy")
	public void choose_to_buy_the_first_item() throws InterruptedException {
		WebElement items = driver.findElement(By.linkText("Portable Sound
Speakers"));
		items.click();
		Thread.sleep(2000);
```

```
	driver.findElement(By.xpath("//*[@id=\"product_enteredQuantity_23\"]")).s
endKeys("");

		WebElement addToCart = driver.findElement(By.xpath("//*[@id=\"add-
to-cart-button-23\"]"));
		addToCart.click();
		Thread.sleep(2000);
		}
	@And("The product is added to cart")
	public void added() throws InterruptedException{
		Boolean Display = driver.findElement(By.xpath("//*[@id=\"bar-
notification\"]")).isDisplayed();
		System.out.println("Notification displayed is : " + Display);
		Thread.sleep(2000);
	}
	@And("Checkout from cart")
	public void checkout_cart() throws InterruptedException {
		System.out.println("Product is added to cart");
		WebElement cart = driver.findElement(By.cssSelector("#topcartlink >
a"));
		cart.click();
		Thread.sleep(2000);

		Select wrapping = new
Select(driver.findElement(By.id("checkout_attribute_1")));
		wrapping.selectByValue("2");

		WebElement CheckBox = driver.findElement(By.id("termsofservice"));
		boolean isSelected = CheckBox.isSelected();
		if(isSelected == false) {
			CheckBox.click();
		}
		WebElement continueToCheckout =
driver.findElement(By.id("checkout"));
		continueToCheckout.click();
		}
	@And("Enters details on checkout as guest page")
	public void details() throws InterruptedException {

	driver.findElement(By.xpath("/html/body/div[6]/div[3]/div/div/div/div[2]/
div[1]/div[1]/div[3]/button[1]")).click();
		Thread.sleep(5000);

		WebElement firstName =
driver.findElement(By.xpath("//*[@id=\"BillingNewAddress_FirstName\"]"));
		firstName.sendKeys("Arjun");

		WebElement lastName =
driver.findElement(By.cssSelector("#BillingNewAddress_LastName"));
		lastName.sendKeys("M");
```

```
            WebElement emailAddress =
driver.findElement(By.id("BillingNewAddress_Email"));
            emailAddress.sendKeys("abc@gmail.com");

            Select country = new
Select(driver.findElement(By.xpath("//*[@id=\"BillingNewAddress_CountryId\"]"))
);
            country.selectByValue("133");
            Thread.sleep(2000);

            WebElement city =
driver.findElement(By.xpath("//*[@id=\"BillingNewAddress_City\"]"));
            city.sendKeys("Chennai");

            WebElement address =
driver.findElement(By.xpath("/html/body/div[6]/div[3]/div/div/div/div[2]/ol/li[
1]/div[2]/form/div/div/div[2]/div/div/div[8]/input"));
            address.sendKeys("No.20,20th street,Besant nagar");

            WebElement pincode =
driver.findElement(By.cssSelector("#BillingNewAddress_ZipPostalCode"));
            pincode.sendKeys("600090");

            Thread.sleep(2000);
            WebElement phone =
driver.findElement(By.xpath("//*[@id=\"BillingNewAddress_PhoneNumber\"]"));
            phone.sendKeys("9876543210");

            WebElement contbtn = driver.findElement(By.xpath("//*[@id=\"billing-
buttons-container\"]/button[4]"));
            contbtn.click();

            WebElement shipping =
driver.findElement(By.xpath("//*[@id=\"shippingoption_2\"]"));
            shipping.isEnabled();
            System.out.println("Radio button is enabled for air shipping");

            }
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>Cucumber_Project</groupId>
  <artifactId>Final</artifactId>
  <version>0.0.1-SNAPSHOT</version>
```

```
<packaging>jar</packaging>

<name>Final</name>
<url>http://maven.apache.org</url>


<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

   <!--Selenium Dependencies-->
  <dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.12.0</version>
  </dependency>

  <!--Cucumber Dependencies-->
  <dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>7.13.0</version>
  </dependency>

  <dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-junit</artifactId>
  <version>7.13.0</version>
  <scope>test</scope>
  </dependency>

  <!--JUnit Dependencies-->
  <dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
  </dependency>

  <!--Apache POI Dependencies for Excel-->
  <dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>5.2.3</version>
  </dependency>

  <dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
```

```
    <version>5.2.3</version>
    </dependency>


    <!--TestNG Dependency-->
    <dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.8.0</version>
    <scope>test</scope>
    </dependency>

      <!-- Extent Reports -->
   <dependency>
    <groupId>com.aventstack</groupId>
    <artifactId>extentreports</artifactId>
    <version>5.0.9</version>
</dependency>


   <dependency>
     <groupId>Cucumber_Project</groupId>
     <artifactId>Check</artifactId>
     <version>0.0.1-SNAPSHOT</version>
   </dependency>
   <dependency>
     <groupId>tech.grasshopper</groupId>
      <artifactId>extentreports-cucumber6-adapter</artifactId>
      <version>1.2.0</version>
      <scope>test</scope>
 </dependency>

  </dependencies>



</project>
```

CONCLUSION

In conclusion, the automation project using Cucumber and Selenium has been a significant success, transforming our testing process. Throughout this project, several key objectives have been achieved and we can confidently assert that these goals have been met.