Source Code naveen n k

**GITHUB URL:** [**https://github.com/nknaveengowda/Capstone-Project-Sporty-shoes**](https://github.com/nknaveengowda/Capstone-Project-Sporty-shoes)

1. **Create Selenium scripts using TestNG to test all the pages in the web app that will automate:**

**● Login page**

**● Registration Page**

**● Add Product to cart page.**

**● Place Order Page**

**Login:**

package com.sportyshoes.com;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class LoginTest {

private WebDriver driver;

@BeforeClass

public void setUp() {

driver = new ChromeDriver();

driver.get("http://localhost:9010/");

}

@Test

public void testLogin() {

WebElement emailInput = driver.findElement(By.id("email"));

WebElement passwordInput = driver.findElement(By.id("password"));

WebElement loginButton = driver.findElement(By.cssSelector("button.btn.btn-primary"));

// Provide login information

emailInput.sendKeys("naveen@gmail.com");

passwordInput.sendKeys("Password1234");

// Click on the Login button

loginButton.click();

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

//driver.quit();

}

}

}

**Registration:**

package com.sportyshoes.com;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class RegistrationTest {

private WebDriver driver;

@BeforeClass

public void setUp() {

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to the registration page

driver.get("http://localhost:9010/register");

}

@Test

public void testRegistration() {

WebElement registerButton = driver.findElement(By.xpath("//button[@type='submit']"));

// Provide registration information

WebElement nameInput = driver.findElement(By.id("name"));

WebElement emailInput = driver.findElement(By.id("email"));

WebElement passwordInput = driver.findElement(By.id("password"));

nameInput.sendKeys("Naveen");

emailInput.sendKeys("naveen@gmail.com");

passwordInput.sendKeys("Password1234");

// Click on the Register button

registerButton.click();

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

//driver.quit();

}

}

}

**Add to Cart:**

package com.sportyshoes.com;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class AddToCartTest {

private WebDriver driver;

@BeforeClass

public void setUp() {

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to the login page

driver.get("http://localhost:9010/");

// Login

WebElement emailInput = driver.findElement(By.id("email"));

WebElement passwordInput = driver.findElement(By.id("password"));

WebElement loginButton = driver.findElement(By.cssSelector("button.btn.btn-primary"));

emailInput.sendKeys("naveen@gmail.com");

passwordInput.sendKeys("Password1234");

loginButton.click();

}

@Test

public void testAddToCart() {

// Navigate to the product page

//driver.get("http://localhost:9010/login");

// Click on the "Add to Cart" button for product

WebElement addToCartButton = driver.findElement(By.id("cart201")); // ID or selector

addToCartButton.click();

// Introduce a short delay (Not recommended in real-world scenarios)

try {

Thread.sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

// Print successful addition message

WebElement successMessage = driver.findElement(By.id("successMessage"));

System.out.println(successMessage.getText());

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

// driver.quit();

}

}

}

**PlaceOrder:**

package com.sportyshoes.com;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class PlaceOrderTest {

private WebDriver driver;

@BeforeClass

public void setUp() {

// Initialize WebDriver

driver = new ChromeDriver();

// Navigate to the login page

driver.get("http://localhost:9010/");

// Login

WebElement emailInput = driver.findElement(By.id("email"));

WebElement passwordInput = driver.findElement(By.id("password"));

WebElement loginButton = driver.findElement(By.cssSelector("button.btn.btn-primary"));

emailInput.sendKeys("naveen@gmail.com");

passwordInput.sendKeys("Password1234");

loginButton.click();

}

@Test

public void testPlaceOrder() {

// Click on the "Cart" button

WebElement cartButton = driver.findElement(By.linkText("Cart"));

cartButton.click();

// Introduce a short delay (Not recommended in real-world scenarios)

try {

Thread.sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

// Click on the "Place Order" button

WebElement placeOrderButton = driver.findElement(By.id("place-order")); // Replace with actual ID or selector

placeOrderButton.click();

// Introduce a short delay (Not recommended in real-world scenarios)

try {

Thread.sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

// Print a success message

WebElement successMessage = driver.findElement(By.id("successMessage"));

System.out.println(successMessage.getText());

}

@AfterClass

public void tearDown() {

// Close the browser

if (driver != null) {

//driver.quit();

}

}

}

1. **Automate the below API endpoints using Rest-Assured**

* **Retrieve the list of all products in the store.**
* **Retrieve the list of all registered users.**
* **Add the product.**
* **Delete the product.**
* **Update the product.**

**Get List:**

package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class GetAllshoes {

@Test(priority='1')

public void get\_all\_shoes() {

RestAssured.given()

.baseUri("http://localhost:9010")

.basePath("/get-shoes")

.when()

.get()

.then()

.statusCode(200)

.log().all();

}

@Test(priority='2')

public void get\_all\_users() {

RestAssured.given()

.baseUri("http://localhost:9010")

.basePath("/get-users")

.when()

.get()

.then()

.statusCode(200)

.log().all();

}

}

**POST and PUT:**

package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class PostandPutnewshoe { @Test(priority='1')

public void add\_new\_product(){

RestAssured.given()

.baseUri("http://localhost:9010")

.basePath("/add-shoe")

.queryParam("id","1020")

.queryParam("image", "www.imge.com")

.queryParam("name","Nike")

.queryParam("category", "Running")

.queryParam("sizes","5,6,7")

.queryParam("price", "2000")

.when()

.post()

.then()

.log().all();

}

@Test(priority='2')

public void update\_a\_product(){

RestAssured.given()

.baseUri("http://localhost:9010")

.basePath("/update-shoe")

.queryParam("id","1020")

.queryParam("image", "www.imge123.com")

.queryParam("name","Reebok")

.queryParam("category", "Running")

.queryParam("sizes","5,6,7")

.queryParam("price", "2500")

.when()

.put()

.then()

.log().all();

}

}

**DELETE Shoe:**

package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class Deleteshoe {

@Test(priority='1')

public void delete\_product(){

RestAssured.given()

.baseUri("http://localhost:9010")

.basePath("/delete-shoe")

.queryParam("id", "1010")

.when().delete()

.then().log().all();

}

}

1. **Setup Cucumber in Java Project and write Feature Files using Gherkin to test the API endpoints**

**TestRunner:**

package com.simplilearn.cucumber.testrunner;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

@CucumberOptions(features = "src/test/resources/resource/features",

glue = {"com/simplilearn/cucumber/stepdefinitions" })

public class TestRunner {

}

**Feature:**

* **GetProduct:**

Feature: Retrieve all products from the sports shoe store

As a user,

I want to retrieve the list of all available shoes from the sports shoe store,

So that I can browse and choose the desired products.

@tag @GetProducts

Scenario: Retrieve all products successfully

Given I am interested in viewing the available shoes

When I send a request to retrieve the list of products

Then I should receive a successful response with status code 200

And the response should contain a valid list of sports shoes

* **CreateShoe:**

@tag

Feature: Create a new shoe in the store

I want to test the functionality of creating a new shoe

@tag1

Scenario: Create a new shoe on the website

Given Open the API base URL

And Enter the shoe details

When Hit the POST request to create a new shoe

Then The response code should be 200

And Print the result of adding new shoe

* **Update:**

@tag

Feature: Update a shoe in the store

I want to test the functionality of updating a shoe in the store

@tag1 @UpdateRequest

Scenario: Update a shoe on the website

Given open the update product base url

When hit the PUT request to update the shoe with ID 101

Then the response code should be equal to 200

And the response should indicate successful update

* **Delete:**

@tag

Feature: Delete a shoe from the store

I want to test the functionality of deleting a shoe from the store

@tag1

Scenario: Delete a shoe from the website

Given Open the API base URL

And there is an existing shoe with ID 101

When hit the DELETE request for the shoe with ID 101

Then the response code should be 200

And the response should indicate successful deletion

1. **There are no any source code for JMETER and POSTMAN Frameworks to write there in this documentation. I have uploaded the screenshots of execution in other Document, please check that.**

**THANK YOU FOR THE PROJECT. I FELT REALLY GREAT DURING TRAINING PERIOD. I WILL UTILIZE MY KNOWLEDGE THROUGHT MY CREAR.**

**(NAVEEN N K)**

**GITHUB URL:** [**https://github.com/nknaveengowda/Capstone-Project-Sporty-shoes**](https://github.com/nknaveengowda/Capstone-Project-Sporty-shoes)