**Repo link:** https://github.com/hindujar/CAPSONE\_PROJECT.git

**CAPSTONE PROJECT**

**Aim:**

To create a testing framework for Sporty Shoes Website

**Description:**

**1.** **Browser-based end-user testing using Selenium WebDriver with TestNG Framework**

i. Create a test base class and import all the classes and add the url of the product website

**SOURCE CODE OF TEST BASE**

package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class TestBase {

public static WebDriver driver;

public static void OpenBrowser(String browser)

{

if(browser == "Chrome")

{

driver = new ChromeDriver();

}

if(browser == "FireFox")

{

driver = new FirefoxDriver();

}

driver.manage().window().maximize();

driver.manage().deleteAllCookies();

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

}

public static void closebrowser()

{

driver.close();

}

}

ii. **Sign Up and Log In**

Automate the sign-up process with valid and invalid credentials.

Test the login functionality with correct and incorrect credentials.

Check for error messages on failed login attempts.

**SOURCE CODE OF HOME PAGE AND HOME PAGE TEST:**

package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class HomePage extends TestBase {

@FindBy(xpath="//div[@class='container mt-3']/descendant::p[1]")

WebElement text;

@FindBy(linkText="New User? Register Here")

WebElement registerLink;

public HomePage(WebDriver driver) {

PageFactory.initElements(driver, this);

}

public String getURL\_page()

{

String URLnew = driver.getCurrentUrl();

return URLnew;

}

public String Validate\_Text\_On\_Page()

{

String pageText = text.getText();

System.out.println(pageText);

return pageText;

}

public void click\_register\_link()

{

registerLink.click();

}}

**HOME PAGE TEST**

package com.sportyshoes.Tests;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

import static org.testng.Assert.assertEquals;

import org.testng.Assert;

import org.testng.Assert.\*;

public class HomePageTest extends TestBase {

HomePage hp;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

}

@Test(priority='1')

public void test\_getTitle\_page()

{

String expected = "http://localhost:9010/";

String Actual = hp.getURL\_page();

Assert.assertEquals(Actual, expected);

}

@Test(priority='2')

public void Test\_Validate\_Text\_On\_Page()

{

String expected = "Powered By Simplilearn";

String actualText = hp.Validate\_Text\_On\_Page();

Assert.assertEquals(actualText, expected);

}

@Test(priority='3')

public void test\_click\_register\_link() throws InterruptedException

{

Thread.sleep(1500);

hp.click\_register\_link();

}}

**SOURCE CODE FOR REGISTER PAGE AND TEST**:

package com.sportyshoes.Tests;

import static org.testng.Assert.assertEquals;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

public class RegisterPageTest extends TestBase {

HomePage hp;

RegisterPage rp;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

rp = new RegisterPage(driver);

}

@Test(priority='1')

public void test\_click\_register\_link() throws InterruptedException

{

Thread.sleep(1500);

hp.click\_register\_link();

}

@Test(priority='2')

public void test\_getTitle\_page()

{

String expected = "http://localhost:9010/register";

String Actual = hp.getURL\_page();

Assert.assertEquals(Actual, expected);

}

@Test(priority='3')

public void Test\_register\_user()

{

rp.register\_user();

}

@Test(priority='4')

public void Test\_validate\_registration\_URL()

{

String expected = "http://localhost:9010/register-user";

String Actual = rp.validate\_registration\_URL();

assertEquals(Actual, expected);

}

@Test(priority='5')

public void Test\_validate\_registration\_Text()

{

String expected = "Hello hinduja!";

String actualText = rp.validate\_registration\_Text();

Assert.assertEquals(actualText, expected);

}

}

**SOURCE CODE OF REGISTER EST:**

package com.sportyshoes.Tests;

import static org.testng.Assert.assertEquals;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

public class RegisterPageTest extends TestBase {

HomePage hp;

RegisterPage rp;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

rp = new RegisterPage(driver);

}

@Test(priority='1')

public void test\_click\_register\_link() throws InterruptedException

{

Thread.sleep(1500);

hp.click\_register\_link();

}

@Test(priority='2')

public void test\_getTitle\_page()

{

String expected = "http://localhost:9010/register";

String Actual = hp.getURL\_page();

Assert.assertEquals(Actual, expected);

}

@Test(priority='3')

public void Test\_register\_user()

{

rp.register\_user();

}

@Test(priority='4')

public void Test\_validate\_registration\_URL()

{

String expected = "http://localhost:9010/register-user";

String Actual = rp.validate\_registration\_URL();

assertEquals(Actual, expected);

}

@Test(priority='5')

public void Test\_validate\_registration\_Text()

{

String expected = "Hello hinduja!";

String actualText = rp.validate\_registration\_Text();

Assert.assertEquals(actualText, expected);

}

}

**SOURCE CODE OF LOGINPAGE AND LOGIN TEST:**

package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class LoginPage {

@FindBy(xpath="//input[@id='email']")

WebElement loginEmail;

@FindBy(xpath="//input[@id='password']")

WebElement loginpassword;

@FindBy(xpath="//button[@type='submit']")

WebElement loginbtn;

@FindBy(linkText="Cart")

WebElement clickCart;

public LoginPage(WebDriver driver) {

PageFactory.initElements(driver, this);

}

public void user\_login()

{

loginEmail.sendKeys("hinduja@gmail.com");

loginpassword.sendKeys("Hinduja@20");

loginbtn.click();

}

public void click\_cart()

{

clickCart.click();

}}

**LOGIN TEST :**

package com.sportyshoes.Tests;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.LoginPage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

public class LoginPageTest extends TestBase {

HomePage hp;

RegisterPage rp;

LoginPage lp;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

rp = new RegisterPage(driver);

lp = new LoginPage(driver);

}

@Test(priority='1')

public void test\_login()

{

lp.user\_login();

}

@Test(priority='2')

public void test\_getTitle\_page()

{

String expected = "http://localhost:9010/login";

String Actual = hp.getURL\_page();

Assert.assertEquals(Actual, expected);

}

@Test(priority='3')

public void Test\_validate\_registration\_Text()

{

String expected = "Hello hinduja!";

String actualText = rp.validate\_registration\_Text();

Assert.assertEquals(actualText, expected);

}

@Test(priority='4')

public void test\_click\_cart()

{

lp.click\_cart();

}

}

IV. Order Generation and History

Confirm that orders are generated after successful checkouts.

Validate that orders are stored accurately in the order history of the user.

**SOURCE CODE FOR ORDERPAGE AND TEST:**

ackage com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class OrderPage {

@FindBy(linkText="Orders")

WebElement orderlink;

public OrderPage(WebDriver driver) {

PageFactory.initElements(driver, this);

}

public void click\_orderPage()

{

orderlink.click();

}}

**ORDERTEST:**

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.LoginPage;

import com.sportyshoe.SeleniumCucumberScripts.OrderPage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

public class OrderpageTest extends TestBase {

HomePage hp;

RegisterPage rp;

LoginPage lp;

OrderPage op;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

rp = new RegisterPage(driver);

lp = new LoginPage(driver);

op = new OrderPage(driver);

}

@Test(priority='1')

public void test\_login()

{

lp.user\_login();

}

@Test(priority='2')

public void test\_click\_orders()

{

op.click\_orderPage();

}

@Test(priority='3')

public void test\_getTitle\_page()

{

String expected = "http://localhost:9010/orders";

String Actual = hp.getURL\_page();

Assert.assertEquals(Actual, expected);

}

}

V.Cart Operations

Test adding items to the cart and verify their presence.

Validate the ability to remove items from the cart.

**SOURCE CODE FOR ADD TO CART AND TEST:**

package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class AddtoCartPage {

@FindBy(xpath="//a[@id=\"shoe101\"]")

WebElement viewShoeBTN;

@FindBy(xpath = "//a[@id='cart101']")

WebElement addtocartBTN;

@FindBy(xpath="//div[@class='alert alert-success']/descendant::p[1]")

WebElement successText;

JavascriptExecutor executor;

public AddtoCartPage(WebDriver driver) {

PageFactory.initElements(driver, this);

executor = (JavascriptExecutor) driver;

}

public void add\_product\_to\_cart() throws InterruptedException

{

executor.executeScript("arguments[0].click();", addtocartBTN);

}

public String validate\_success\_message()

{

String successtext = successText.getText();

return successtext;

}}

**ADD O CART TEST:**

package com.sportyshoes.Tests;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.AddtoCartPage;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.LoginPage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

public class AddtoCartPageTest extends TestBase {

HomePage hp;

RegisterPage rp;

LoginPage lp;

AddtoCartPage ac;

@BeforeTest

public void start\_browser()

{

OpenBrowser("Chrome");

hp = new HomePage(driver);

rp = new RegisterPage(driver);

lp = new LoginPage(driver);

ac = new AddtoCartPage(driver);

}

@Test(priority='1')

public void test\_login()

{

lp.user\_login();

}

@Test(priority='2')

public void test\_add\_product\_to\_cart() throws InterruptedException

{

ac.add\_product\_to\_cart();

}

@Test(priority='3')

public void test\_validate\_success\_message()

{

String expected = "Message:Shoe BlueWave Running Shoes Added Successfully to Cart";

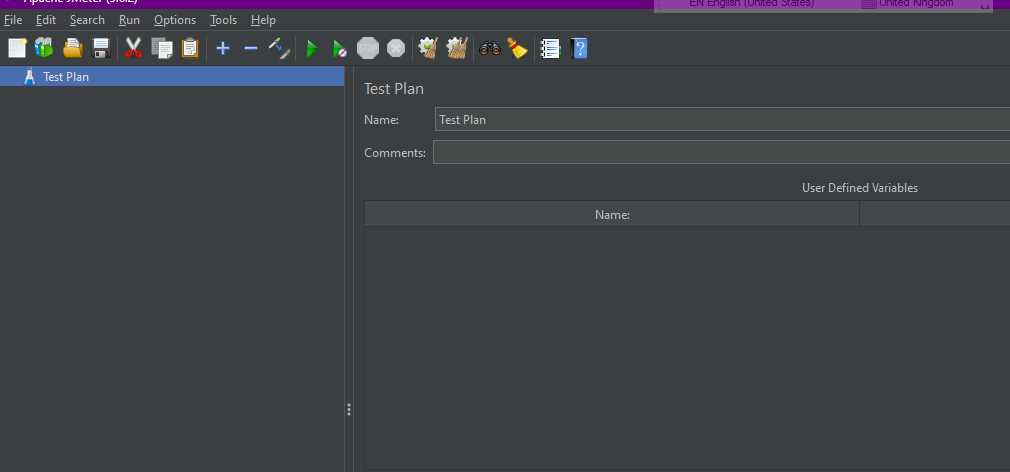
String actualText= ac.validate\_success\_message();

Assert.assertEquals(actualText, expected);

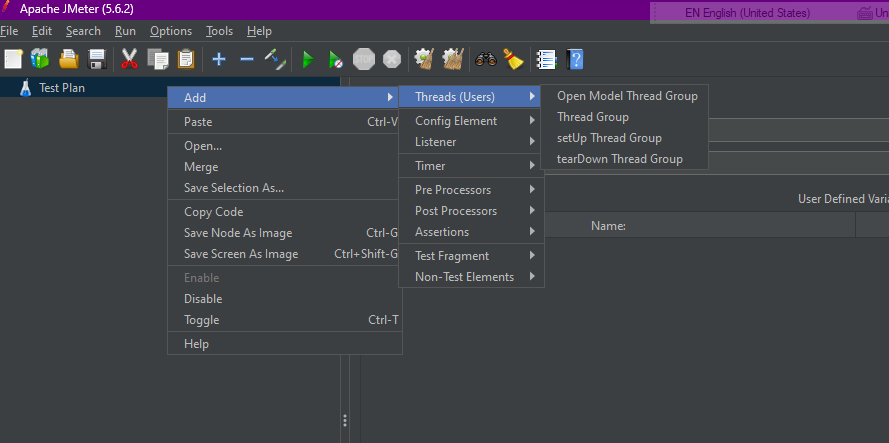
}}

**2.Load Testing using JMeter**

i.Crerate a Test plan by clicking on it

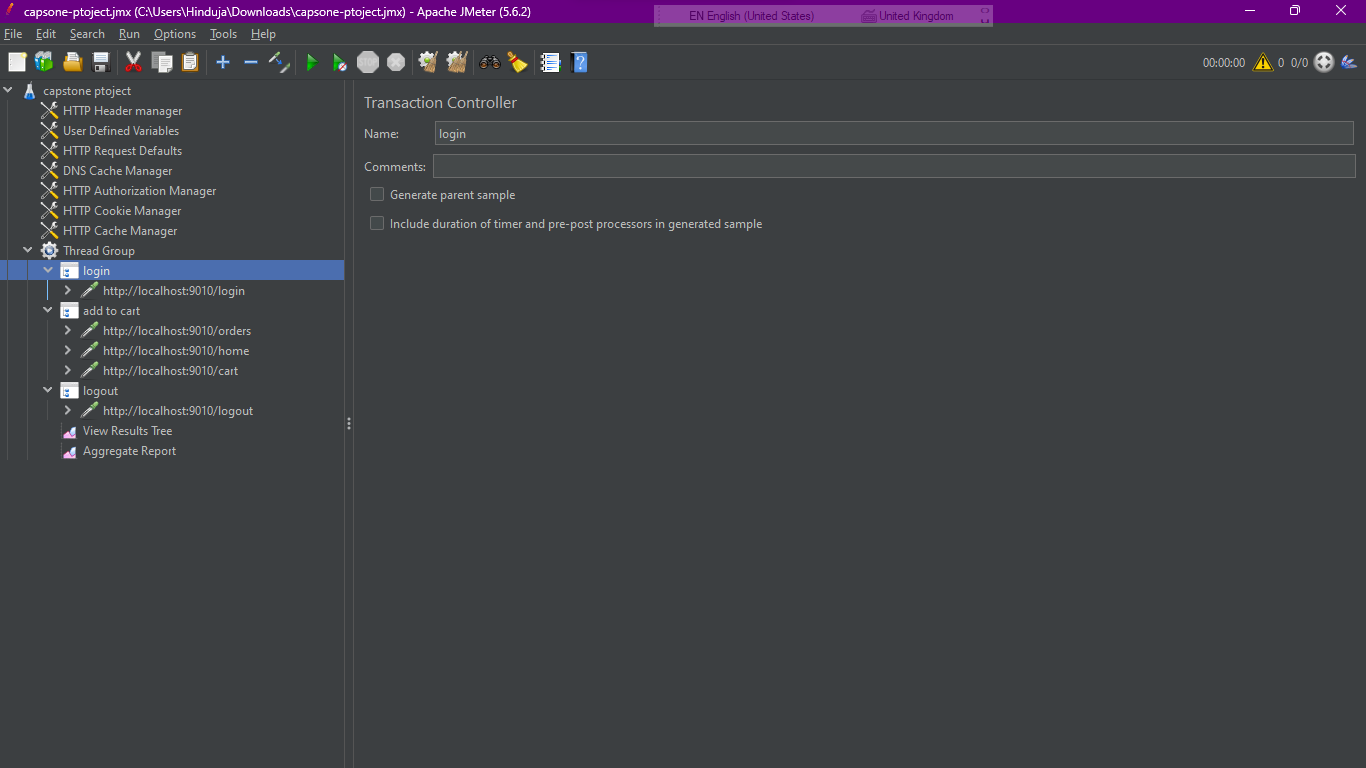


ii.Create a thread group by right click on the test plan

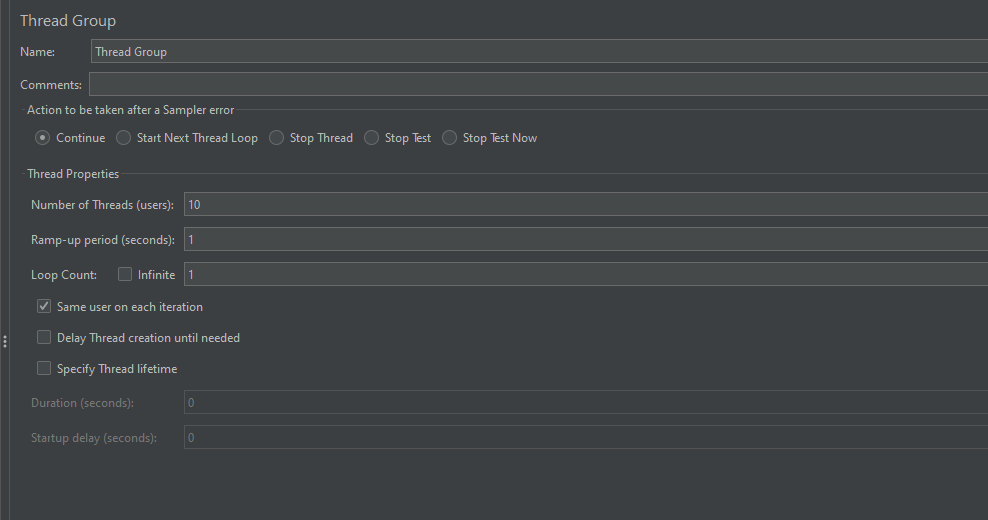


**iii.record the website by using blasemeter for**

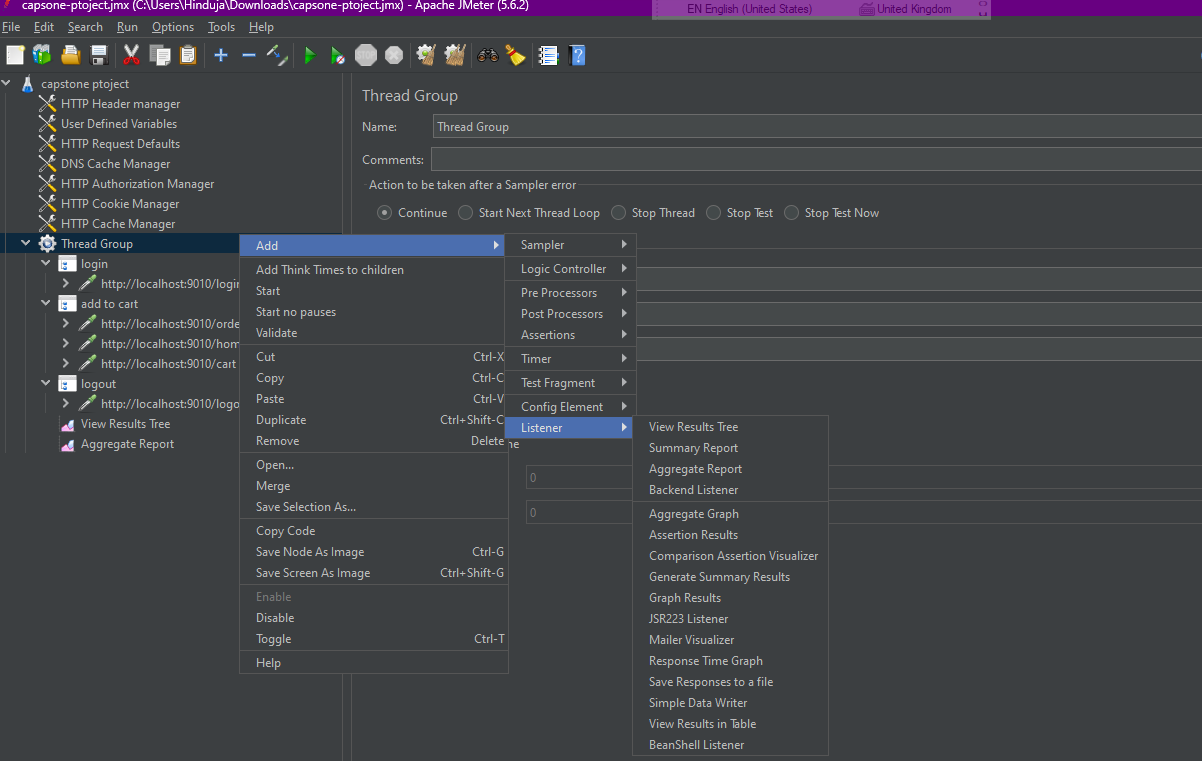
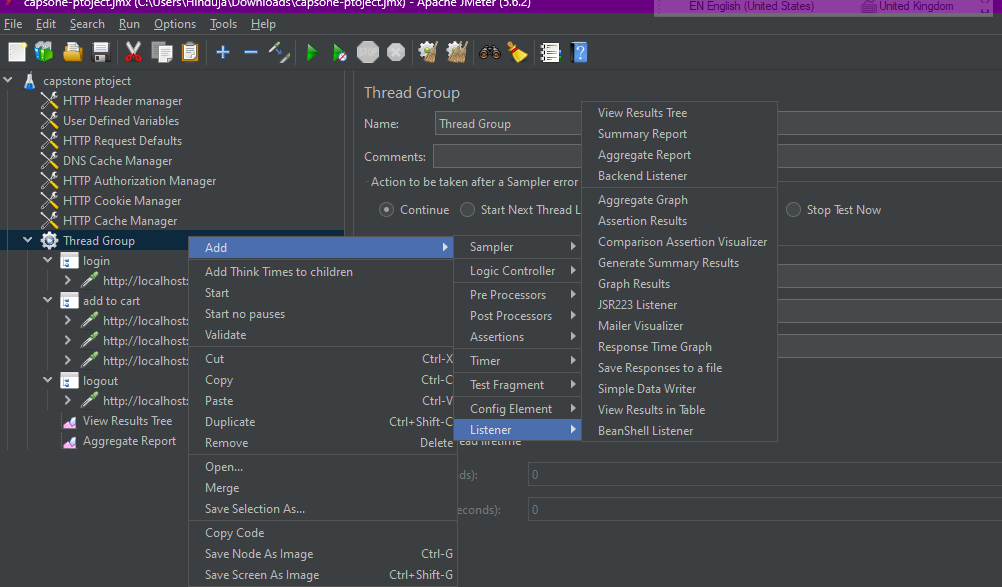
* login page
* register page
* add to car

****

**iv. Increase the number of users to ‘10’ In ThreadGroup:**

****

**v. In order to view the result add listeners by right click on the thread group:**

** **

**3.** **API Testing with Cucumber**

**Source code of feature files of retrieving shoe**

Feature: Retrieving shoe and user information

Scenario: Retrieving all shoes

Given the base URI is http://localhost:9010

And the base path is /get-shoes

When a GET request is sent for shoes

Then the response status code should be 200

And the details should be logged

Scenario: Retrieving all users

Given the base URI is http://localhost:9010

And the base path is /get-users

When a GET request is sent for users

Then the response status code should be 200

And the details should be logged

**Source code of feature files of adding new product,deleting the product and updating the product**

Feature: Managing shoe products

Scenario: Adding a new product

Given the base URI is http://localhost:9010

And the base path is /add-shoe

When a POST request is sent to add a new shoe product

And parameters id=1020, image=www.imge.com, name=Nike, category=Running, sizes=5,6,7, price=2000 are provided

Then the response status code should be 200

And the details should be logged

Scenario: Updating a product

Given the base URI is http://localhost:9010

And the base path is /update-shoe

When a PUT request is sent to update a shoe product with id=1020

And parameters image=www.imge123.com, name=Reebok, category=Running, sizes=5,6,7, price=2500 are provided

Then the response status code should be 200

And the details should be logged

Scenario: Deleting a product

Given the base URI is http://localhost:9010

And the base path is /delete-shoe

When a DELETE request is sent to delete a shoe product with id=1020

Then the response status code should be 200

And the details should be logged

**SOURCE CODE OF RETRIVING THE ALL PRODUCS:**

package com.sportyshoe.re;

import org.testng.annotations.Test;

import io.cucumber.java.en.And;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

import io.restassured.RestAssured;

public class GetAllshoes {

@Test (priority='1')

@Given("the base URI is http://localhost:9010")

@And("the base path is /get-shoes")

@When("a GET request is sent for shoes")

@Then("the response status code should be 200")

@And("the details should be logged")

public void get\_all\_shoes()throws InterruptedException {

{

RestAssured.given()

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

.basePath("/get-shoes")

.when()

.get()

.then()

.statusCode(200)

.log()

.all();

}}

**SOURCE CODE OF RETRIVING ALL USERS**:

public class GetAllshoes {

@Test (priority='1')

@Given("the base URI is http://localhost:9010")

@And("the base path is /get-shoes")

@When("a GET request is sent for shoes")

@Then("the response status code should be 200")

@And("the details should be logged")

public void get\_all\_shoes()throws InterruptedException {

{

RestAssured.given()

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

.basePath("/get-shoes")

.when()

.get()

.then()

.statusCode(200)

.log()

.all();

}}

@Test(priority='2')

@Given("the base URI is http://localhost:9010")

@And("the base path is /get-users")

@When("a GET request is sent for users")

public void get\_all\_users()

{

RestAssured.given()

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

.basePath("/get-users")

.when()

.get()

.then()

.statusCode(200)

.log()

.all();

}}

**SOURCE CODE OF ADDING NEW PRODUCT**

package com.sportyshoe.re

import org.testng.annotations.Test;

import io.cucumber.java.en.And;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

import io.restassured.RestAssured;

public class PostandPut {

@Test(priority='1')

@Given("the base URI is http://localhost:9010")

@And("the base path is /add-shoe")

@When("a POST request is sent to add a new shoe product")

@And("parameters id=1020, image=www.imge.com, name=Nike, category=Running, sizes=5,6,7, price=2000 are provided")

@Then("the response status code should be 200")

@And("the details should be logged")

public void add\_new\_product()throws InterruptedException {

{

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().statusCode(200)

.log().all();

}

}

**SOURCE CODE OF UPDATING THE PRODUCT:**

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()

.statusCode(200)

.log().all();

}

**SOURCE CODE OF DELETING THE PRODUCT:**

@Test(priority='3')

@Given("the base URI is http://localhost:9010")

@And("the base path is /delete-shoe")

@When("a DELETE request is sent to delete a shoe product with id=1020")

public void delete\_product()

{

RestAssured.given()

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

.basePath("/delete-shoe")

.queryParam("id", "1020")

.when().delete()

.then().statusCode(200).log().all();

}

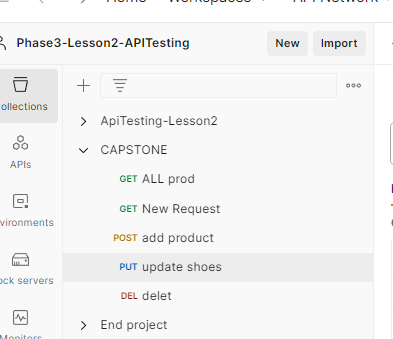
**4.API Testing with Postman and Rest Assured.**

Create Postman scripts to test the following API endpoints:

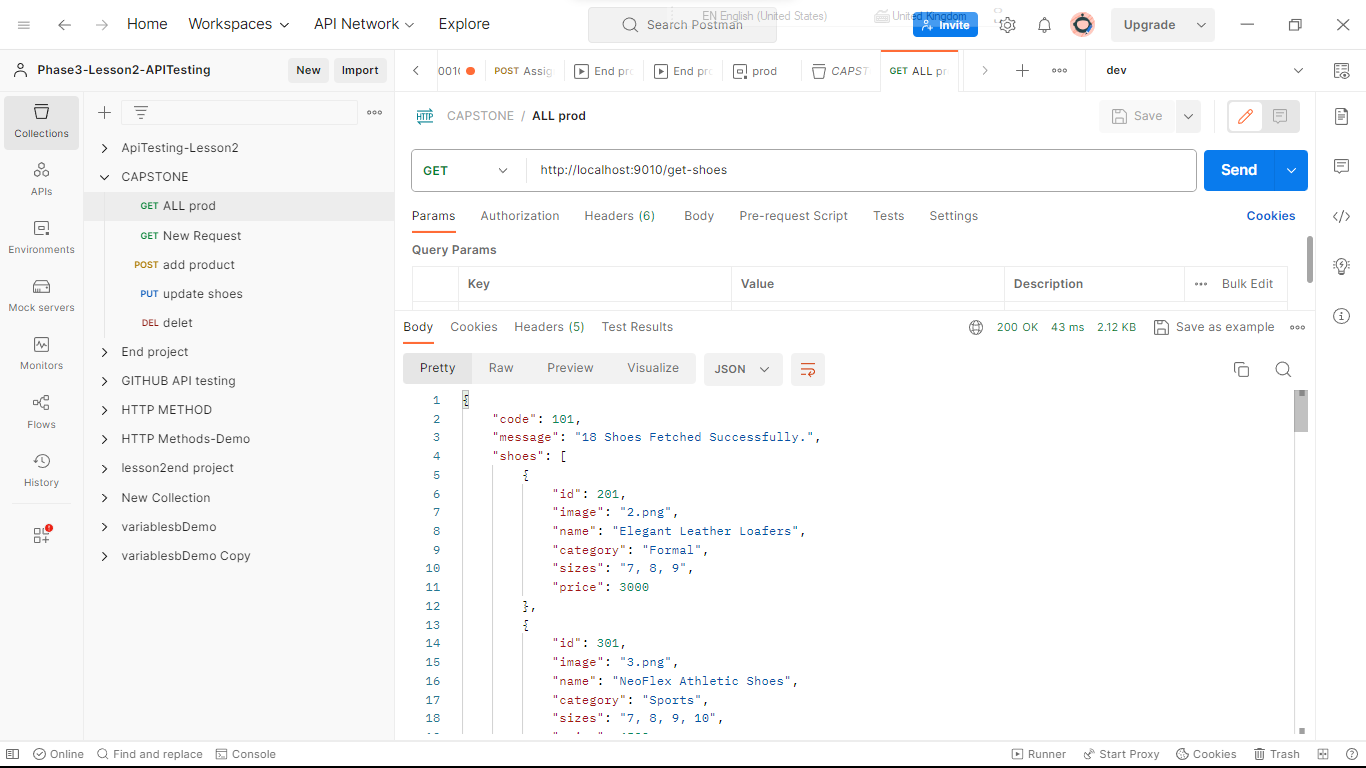
* Retrieve the list of all products in the store.
* Retrieve the list of all registered users.
* Add the product.
* Delete the product.
* Update the product.

**i.To** **Retrieve the list of all products in the store:**

* Create a new collection in postman

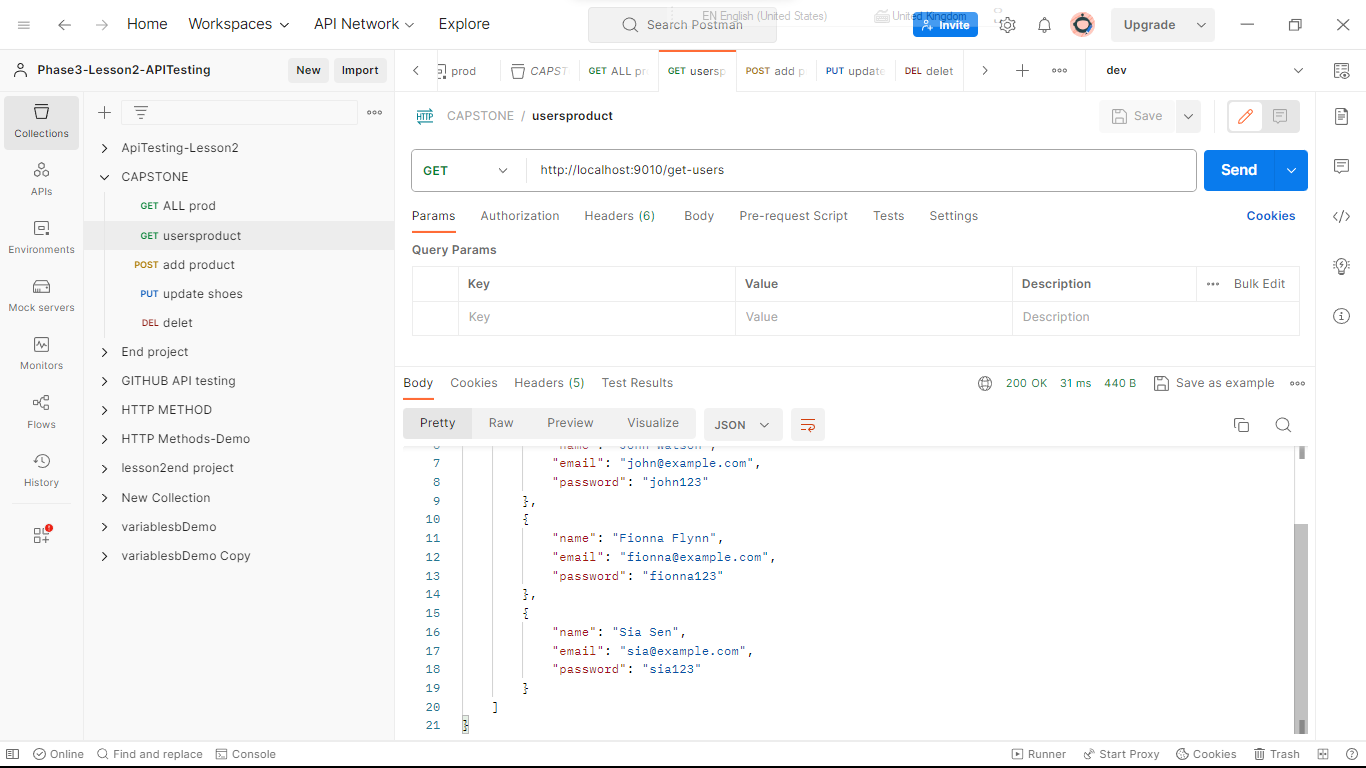


* Create a new request to retrieve all the products in the store by using GET method
* Paste the given URL of store



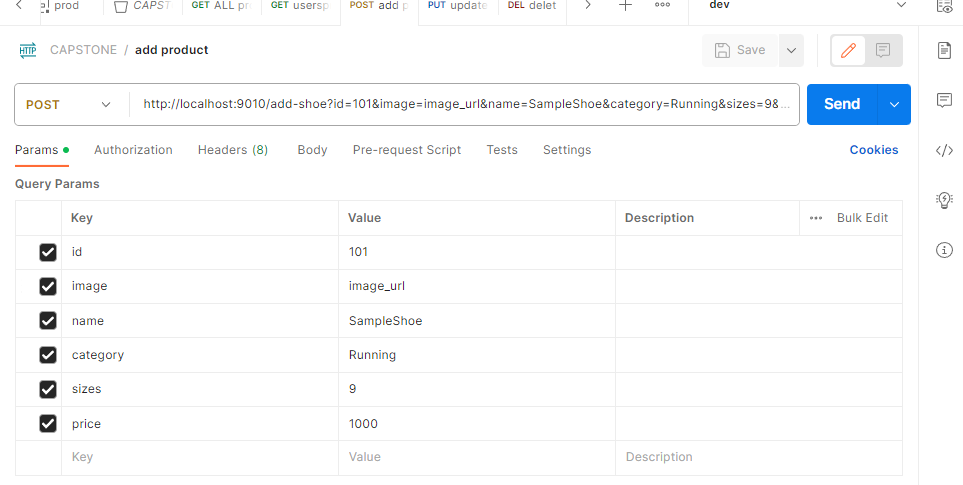
**ii. Retrieve the list of all registered users.**

Create a new GET request to retrieve the list of all registered users



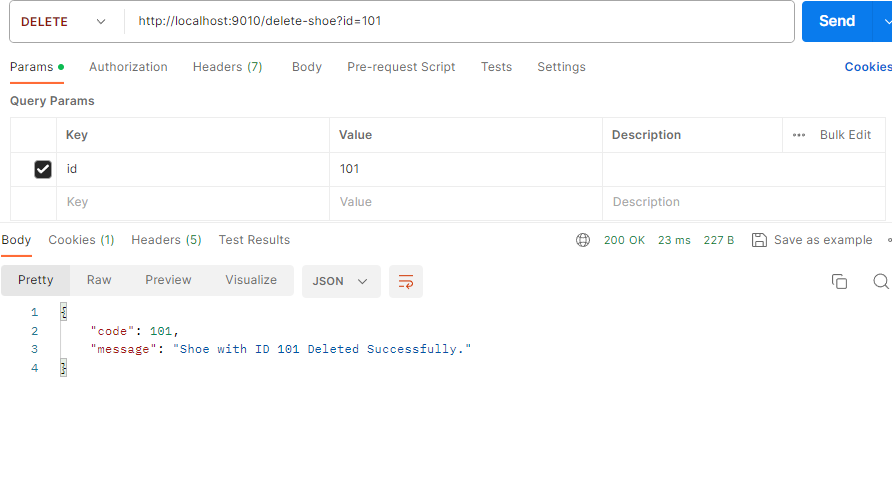
**iii.Add the product**

* Create a new request to add the new product in the list byb using POST method
* Add the keys and values in params
* Add the URL in POST method



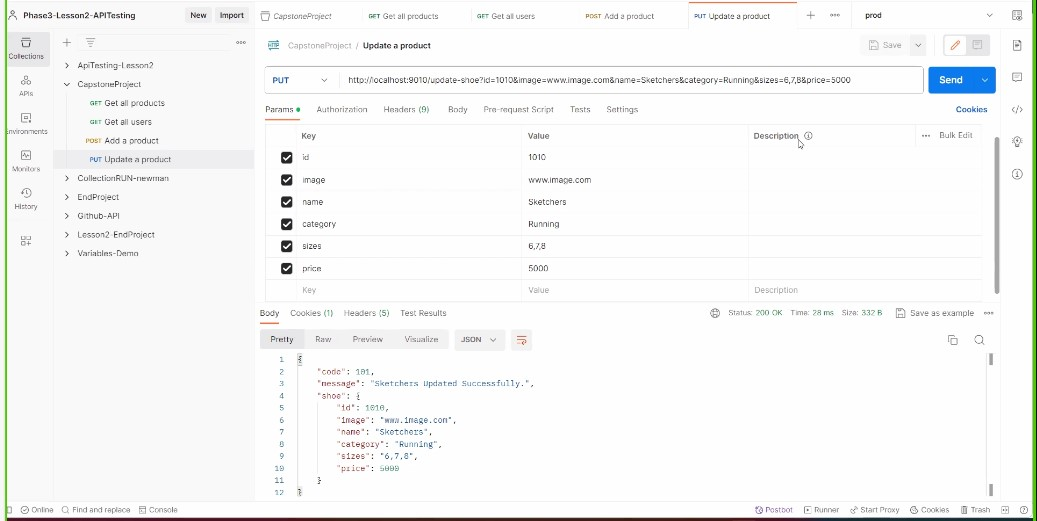
**iv.Delete the product**

* Add the delete request inorder to delete he specific product
* Place the URL to delete the id of the list of products.



**v.Update the product**

* Create the new request to modify the list
* Create the PUT method to update the product list



**Testing with Rest Assured**

**i.Retrieve the list of all products in the store**.

**SOURCE CODE:**

package com.sportyshoe.re;

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();

}

**ii.Retrieve the list of all registered users**

**SOURCE CODE**:

package com.sportyshoe.re;

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(401)

.log()

.all();

}}

**iii.Add the product**

**SOURCE CODE:**

package com.sportyshoe.re;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class PostandPut {

@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().statusCode(200)

.log().all();

}

**iv.Delete the product**

**SOURCE CODE:**

@Test(priority='3')

public void delete\_product()

{

RestAssured.given()

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

.basePath("/delete-shoe")

.queryParam("id", "1020")

.when().delete()

.then().statusCode(200).log().all();

}

**v. Update the product.**

**SOURCE CODE:**

package com.sportyshoe.re;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class PostandPut {

@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().statusCode(200)

.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()

.statusCode(200)

.log().all();

}

@Test(priority='3')

public void delete\_product()

{

RestAssured.given()

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

.basePath("/delete-shoe")

.queryParam("id", "1020")

.when().delete()

.then().statusCode(200).log().all();

}

}