Capstone Project

Creating a Testing Framework for Spotify Shoes Website

GITHUB URL : https://github.com/naveenmpc/JavaCodes1/tree/master/src

GITHUB – Capstone url - https://github.com/naveenmpc/CAPSTONE-PROJECT

The objective is to develop a comprehensive QA and test suite for the Sporty Shoes website. The QA effort will require the following:

1. Browser-based end-user testing using Selenium WebDriver with TestNG Framework.
2. Load Testing using JMeter.
3. API Testing with Cucumber.
4. API Testing with Postman and Rest Assured.

1.Rest Assure

1.1 Getting all Shoes using restassure ..having base uri and path , we can fetch the data.

RegisterPag... CapstonePro... LoginPage.java Delete_meth... GetAllAShoe... Post_Put_Me... 14

```java
package capstone.Sporty.shoes.RestAssureScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

Run All
public class GetAllAShoes {

    @Test
    Run | Debug
    public void get_all_shoes()

    {
        RestAssured.given()
        .baseUri("http://localhost:9010")
        .basePath("/get-shoes")
        .when()
        .get()
        .then()
        .statusCode(200)
        .log()
        .all();
    }

    @Test
    Run | Debug
    public void get_all_users()

    {
        RestAssured.given()
        .baseUri("http://localhost:9010")
        .basePath("/get-users")
        .when()
        .get()
        .then()
        .log()
        .all();
    }
}
```

Fetched 12 Shoes Successfully

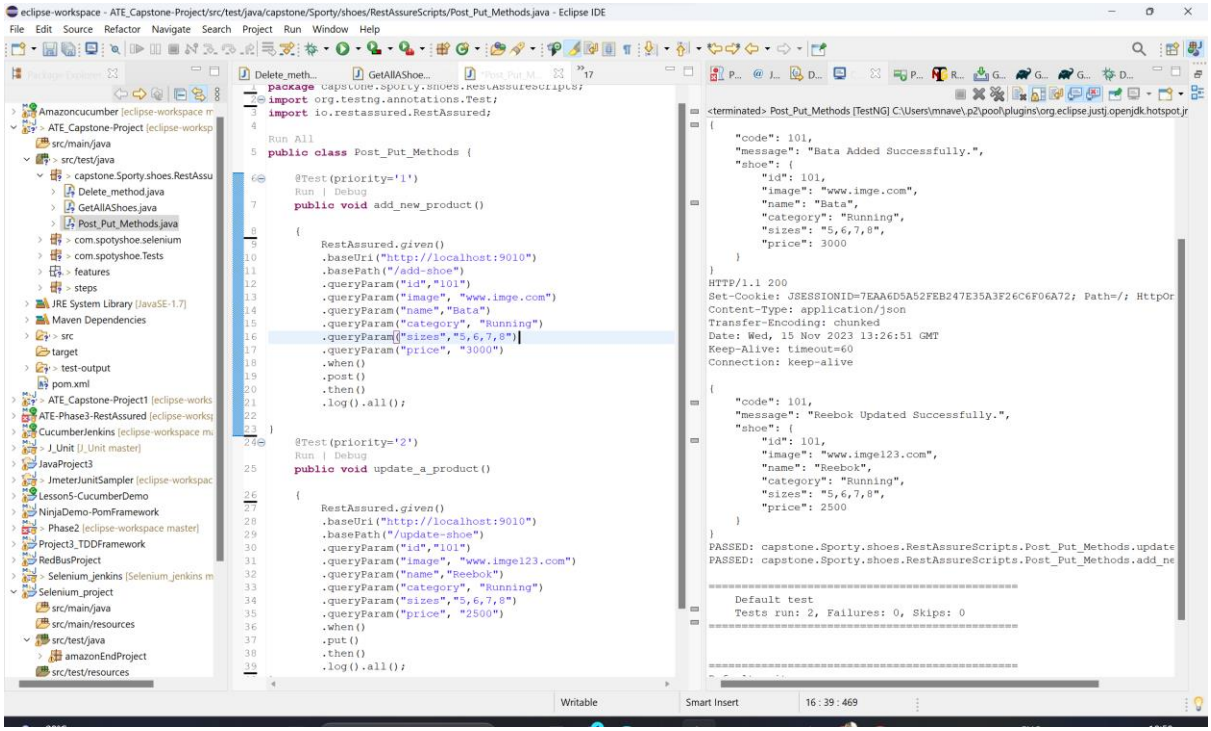
Delete_meth... GetAllAShoe... Post_Put_Me... 17

```java
package capstone.Sporty.shoes.RestAssureScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

Run All
public class GetAllAShoes {

    @Test
    Run | Debug
    public void get_all_shoes()

    {
        RestAssured.given()
        .baseUri("http://localhost:9010")
        .basePath("/get-shoes")
        .when()
        .get()
        .then()
        .statusCode(200)
        .log()
        .all();
    }

    @Test
    Run | Debug
    public void get_all_users()

    {
        RestAssured.given()
        .baseUri("http://localhost:9010")
        .basePath("/get-users")
        .when()
        .get()
        .then()
        .log()
        .all();
    }
}
```

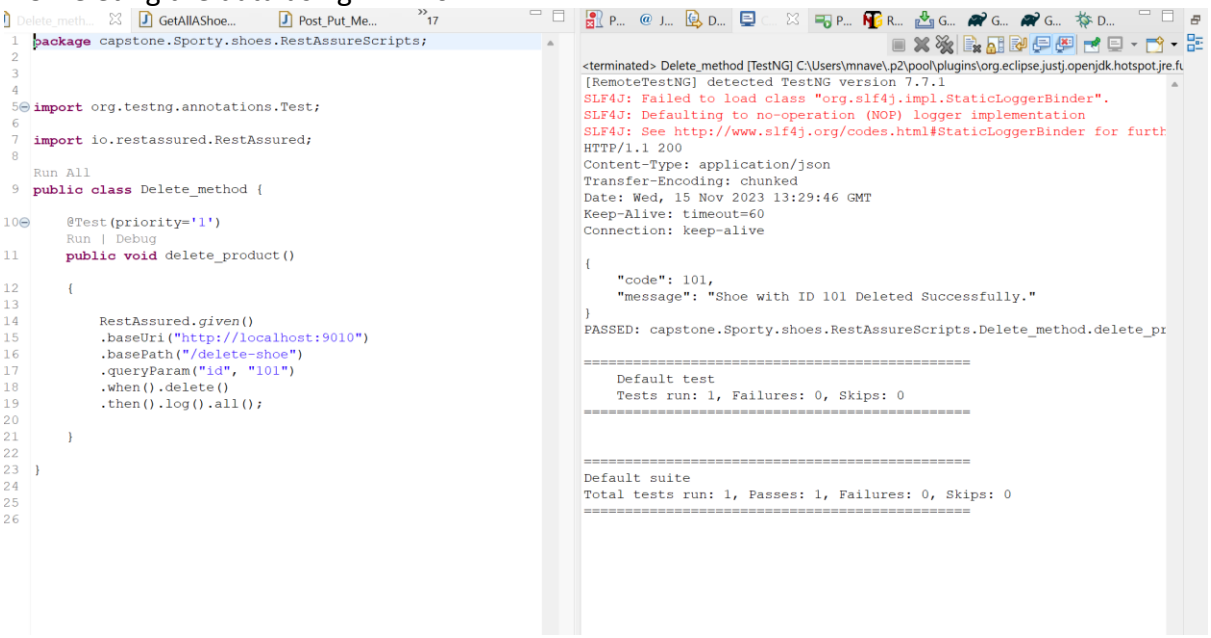
<terminated> GetAllAShoes [TestNG] C:\Users\mnave\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full

```
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for furth
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 15 Nov 2023 13:24:32 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
    "code": 101,
    "message": "12 Shoes Fetched Successfully.",
    "shoes": [
        {
            "id": 201,
            "image": "2.png",
            "name": "Elegant Leather Loafers",
            "category": "Formal",
            "sizes": "7, 8, 9",
            "price": 3000
        },
        {
            "id": 301,
            "image": "3.png",
            "name": "NeoFlex Athletic Shoes",
            "category": "Sports",
            "sizes": "7, 8, 9, 10",
            "price": 4500
        },
        {
            "id": 401,
            "image": "4.png",
            "name": "PowerStride Training Shoes",
            "category": "Sports",
            "sizes": "6, 7, 8, 9",
            "price": 6000
        },
        {
            "id": 501,
            "image": "5.png",
            "name": "StreetRunner Urban Sneakers",
            "category": "Sports",
            "sizes": "7, 9",
```

1.2 Updating and Adding the data to SpotyShoes using Put and Post methods in rest assure. Updation done via Query Parameters , the name and price has been updated and added.



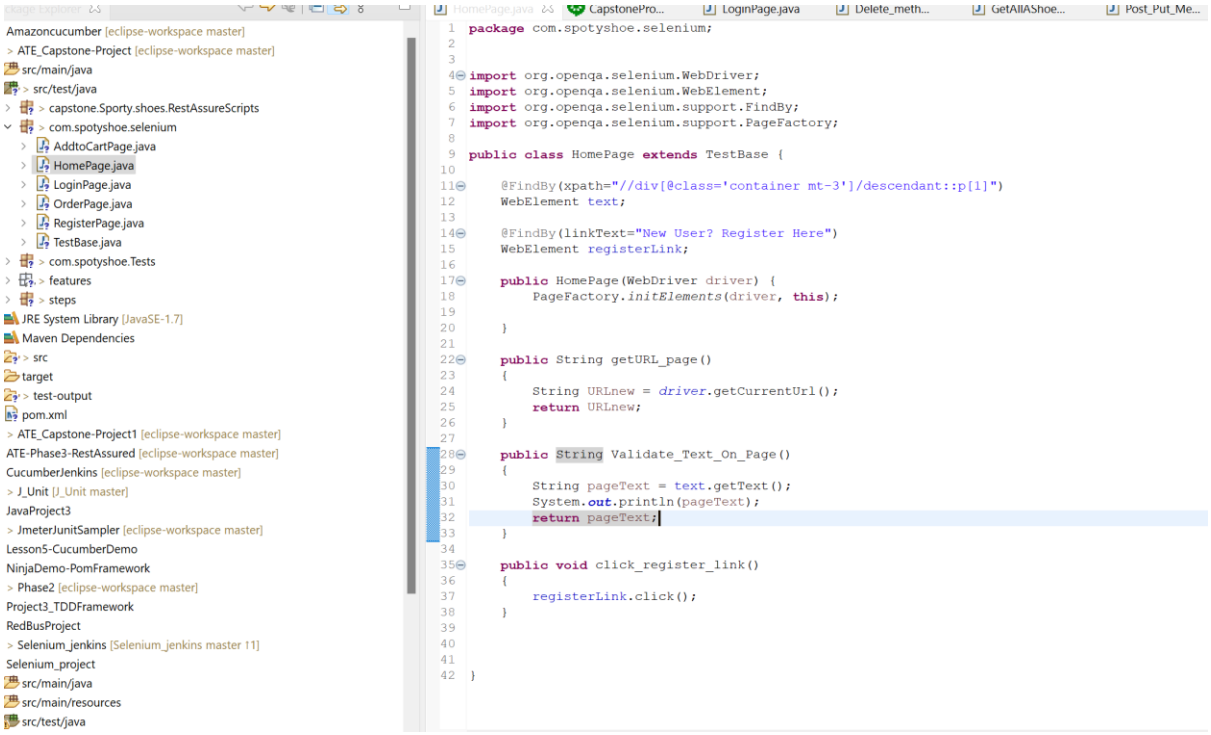
1.3 Deleting the data using ID = 101



2 . Creating the Browser-based end-user testing using Selenium WebDriver with TestNG Framework

Lets us Quickly add the selenium scripts of

Homepage:

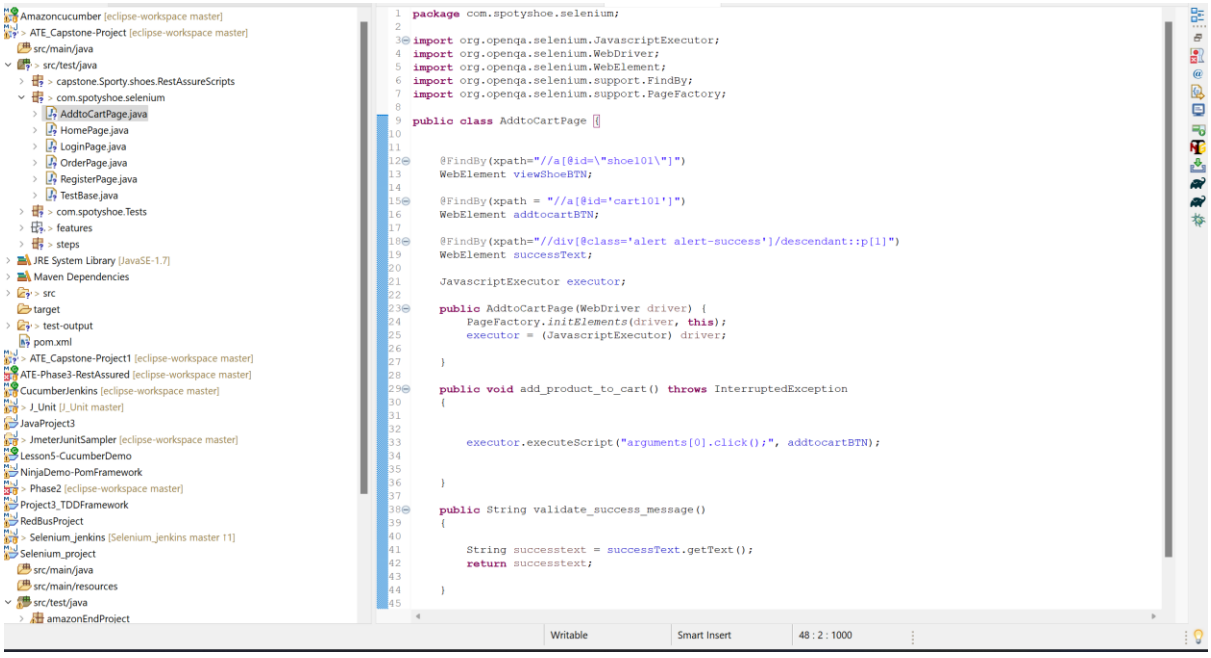


Login Page:

```java
package com.spotyshoe.selenium;



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("mnaveenkumar.mpc@gmail.com");
        loginpassword.sendKeys("Naveen@123");
        loginbtn.click();
    }


    public void click_cart()
    {
        clickCart.click();
    }
```

Writable | Smart Insert | 39 : 29 : 790

Added to cart:

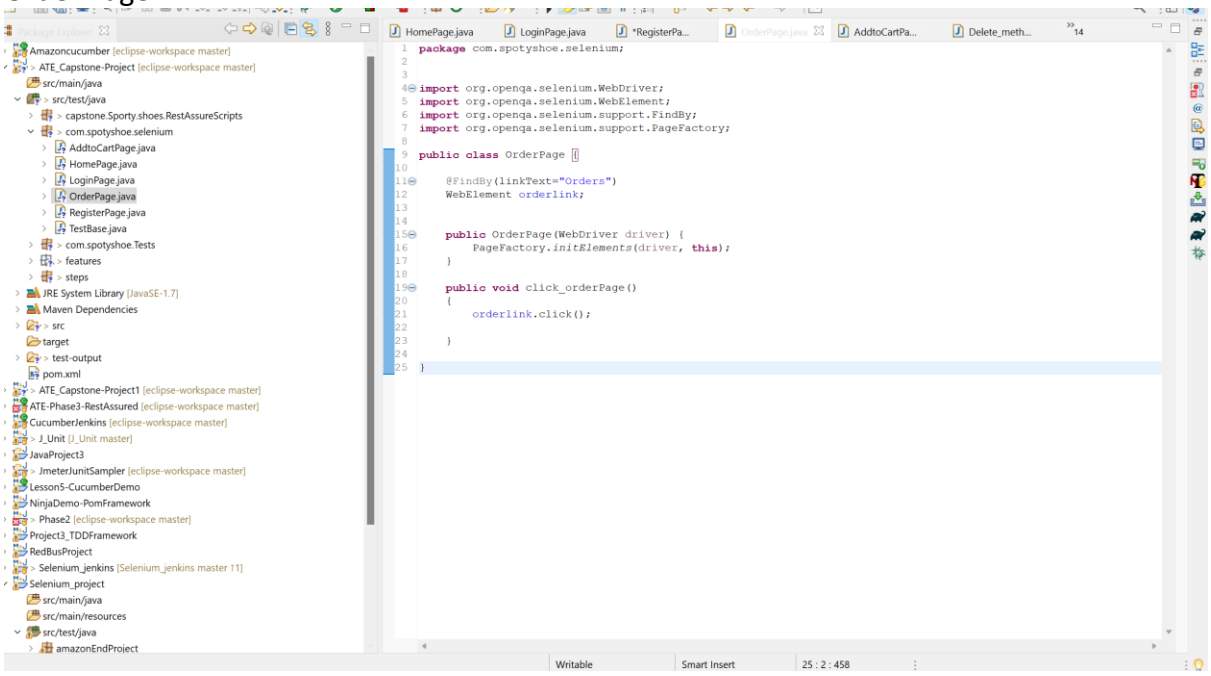


Register Page:

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class RegisterPage extends TestBase{
    @FindBy(xpath="//input[@id='name']")
    WebElement registername;

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

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

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

    @FindBy(xpath="//div[@class='mt-4 p-5 bg-primary text-white rounded']/descendant::p[3]")
    WebElement userText;
    public RegisterPage(WebDriver driver) {
        PageFactory.initElements(driver, this);

    }

    public void register_user()
    {
        registername.sendKeys("Naveen");
        registeremail.sendKeys("mnaveenkumar.mpc@gmail.com");
        registerpassword.sendKeys("Naveen@123");
        registerBtn.click();
    }

    public String validate_registration_URL()
    {
        String register_url = driver.getCurrentUrl();
        return register_url;
    }

    public String validate_registration_Text()
    {
        String user_name = userText.getText();
        return user_name;
    }
```

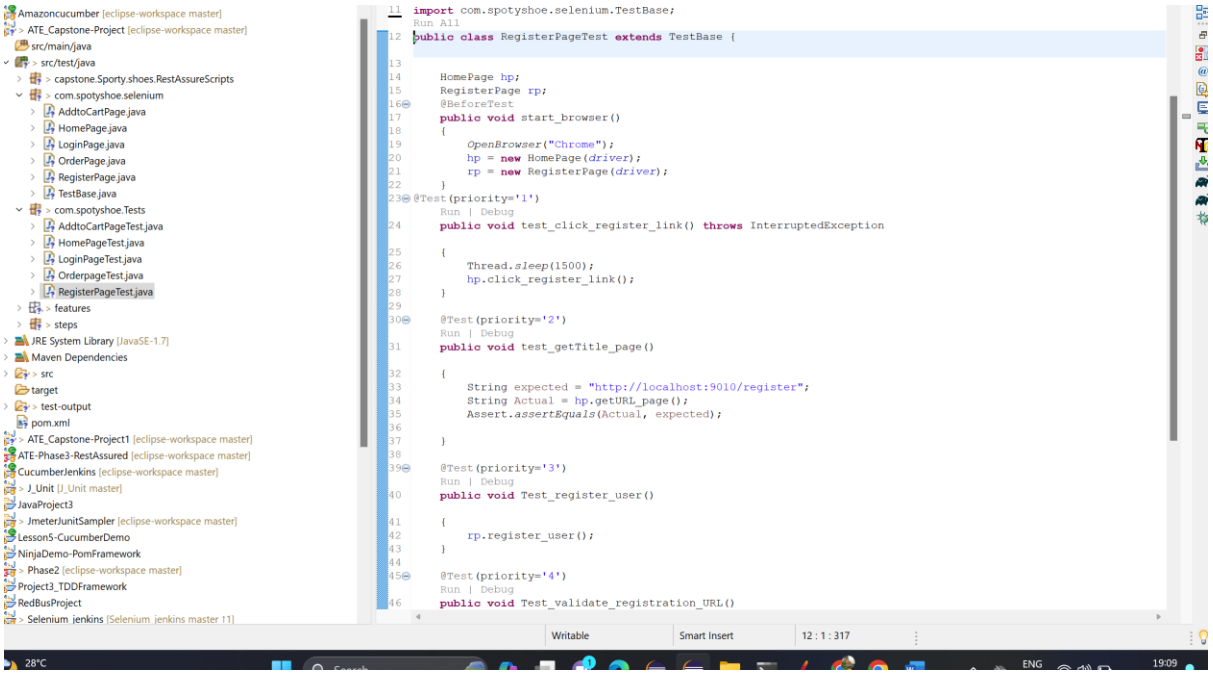
Writable | Smart Insert | 9 : 5 : 255

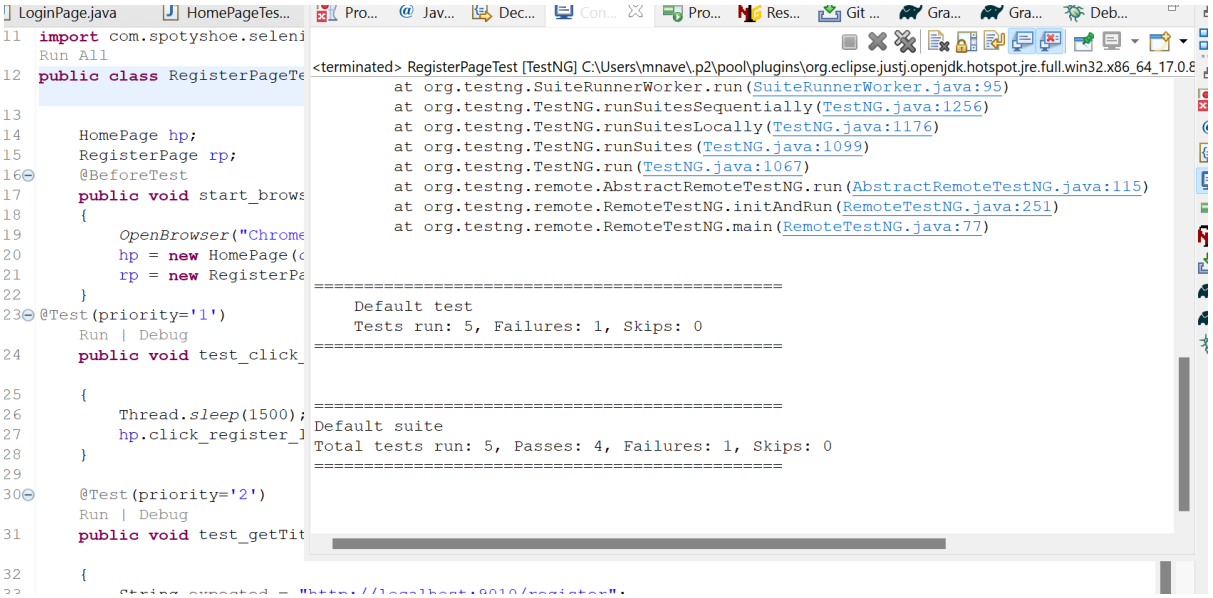
Order Page:



Selenium Webdriver with TestNg:

Register Page


Amazoncucumber [eclipse-workspace master]
ATE_Capstone-Project [eclipse-workspace master]
src/main/java
src/test/java
capstone.Sporty.shoes.RestAssureScripts
com.spotyshoe.selenium
AddtoCartPage.java
HomePage.java
LoginPage.java
OrderPage.java
RegisterPage.java
TestBase.java
com.spotyshoe.Tests
AddtoCartPageTest.java
HomePageTest.java
LoginPageTest.java
OrderpageTest.java
RegisterPageTest.java
features
steps
JRE System Library [JavaSE-1.7]
Maven Dependencies
src
target
test-output
pom.xml
ATE_Capstone-Project1 [eclipse-workspace master]
ATE-Phase3-RestAssured [eclipse-workspace master]
CucumberJenkins [eclipse-workspace master]
J_Unit [J_Unit master]
JavaProject3
JmeterJunitSampler [eclipse-workspace master]
Lesson5-CucumberDemo
NinjaDemo-PomFramework
Phase2 [eclipse-workspace master]
Project3_TDDFramework
RedBusProject
Selenium_jenkins [Selenium_jenkins master !1]
import com.spotyshoe.selenium.TestBase;
Run All
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')
Run | Debug
public void test_click_register_link() throws InterruptedException
{
Thread.sleep(1500);
hp.click_register_link();
}
@Test(priority='2')
Run | Debug
public void test_getTitle_page()
{
String expected = "http://localhost:9010/register";
String Actual = hp.getURL_page();
Assert.assertEquals(Actual, expected);
}
@Test(priority='3')
Run | Debug
public void Test_register_user()
{
rp.register_user();
}
@Test(priority='4')
Run | Debug
public void Test_validate_registration_URL()
Writable
Smart Insert
12 : 1 : 317
28°C
ENG
19:09

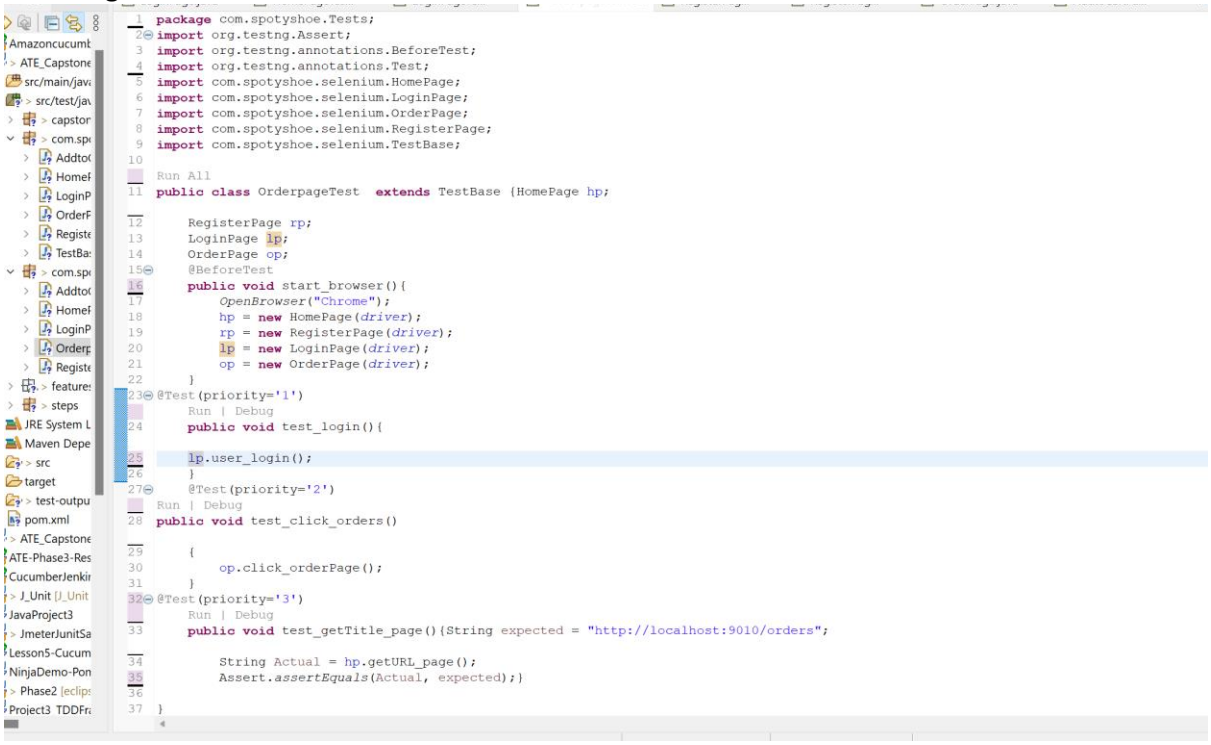

LoginPage.java
HomePageTes...
Pro...
Jav...
Dec...
Con...
Pro...
Res...
Git ...
Gra...
Gra...
Deb...
<terminated> RegisterPageTest [TestNG] C:\Users\mnave\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8
at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)
at org.testng.TestNG.runSuitesSequentially(TestNG.java:1256)
at org.testng.TestNG.runSuitesLocally(TestNG.java:1176)
at org.testng.TestNG.runSuites(TestNG.java:1099)
at org.testng.TestNG.run(TestNG.java:1067)
at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)
at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:251)
at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:77)
Default test
Tests run: 5, Failures: 1, Skips: 0
Default suite
Total tests run: 5, Passes: 4, Failures: 1, Skips: 0

LoginPage :

```java
import com.spotyshoe.selenium.HomePage;
import com.spotyshoe.selenium.LoginPage;
import com.spotyshoe.selenium.RegisterPage;
import com.spotyshoe.selenium.TestBase;
Run All
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')
    Run | Debug
    public void test_login()

    {
        lp.user_login();
    }


    @Test(priority='2')
    Run | Debug
    public void test_getTitle_page()

    {
        String expected = "http://localhost:9010/login";
        String Actual = hp.getURL_page();
        Assert.assertEquals(Actual, expected);

    }

@Test(priority='3')

    Run | Debug
    public void Test_validate_registration_Text()

    {
```

```
<terminated> LoginPageTest [TestNG] C:\Users\mnave\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Nov 15, 2023 7:12:25 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 119, so returning the closest v
PASSED: com.spotyshoe.Tests.LoginPageTest.test_login
PASSED: com.spotyshoe.Tests.LoginPageTest.test_click_cart
PASSED: com.spotyshoe.Tests.LoginPageTest.test_getTitle_page
PASSED: com.spotyshoe.Tests.LoginPageTest.Test_validate_registration_Text

===============================================
    Default test
    Tests run: 4, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```

OrderPage:

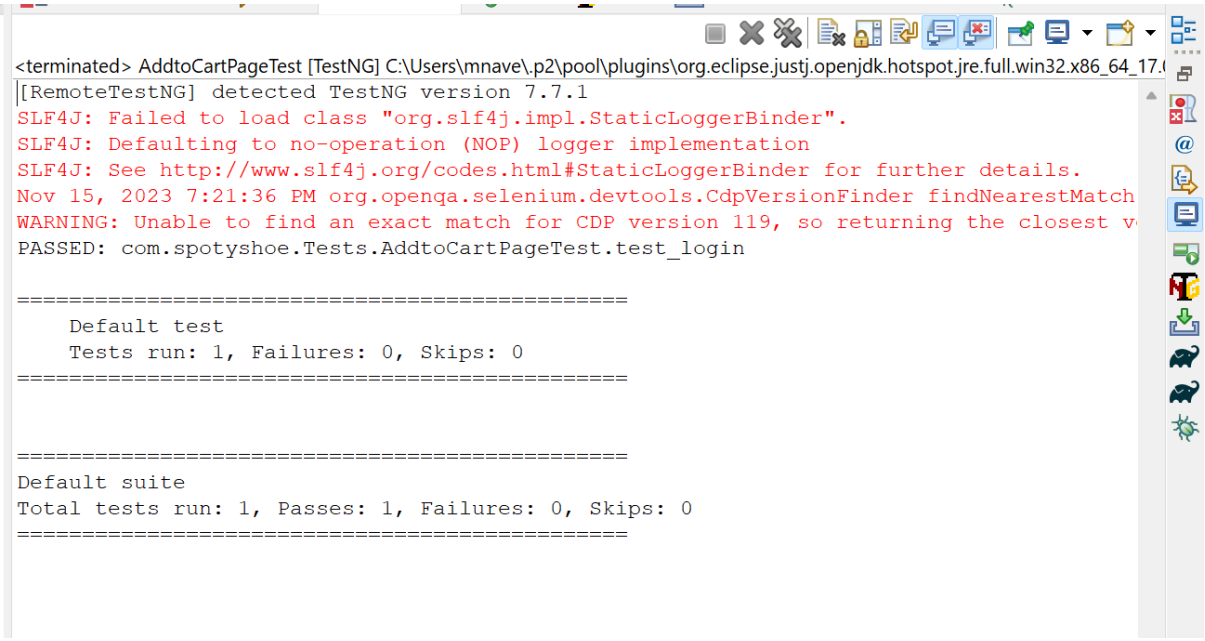


```java
package com.spotyshoe.Tests;
import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import com.spotyshoe.selenium.HomePage;
import com.spotyshoe.selenium.LoginPage;
import com.spotyshoe.selenium.OrderPage;
import com.spotyshoe.selenium.RegisterPage;
import com.spotyshoe.selenium.TestBase;

Run All
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')
    Run | Debug
    public void test_login(){

    lp.user_login();
    }
    @Test(priority='2')
Run | Debug
public void test_click_orders()

    {
        op.click_orderPage();
    }
@Test(priority='3')
    Run | Debug
    public void test_getTitle_page(){String expected = "http://localhost:9010/orders";

        String Actual = hp.getURL_page();
        Assert.assertEquals(Actual, expected);}


}
```

```
<terminated> OrderpageTest [TestNG] C:\Users\mnave\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Nov 15, 2023 7:15:34 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 119, so returning the closest v
PASSED: com.spotyshoe.Tests.OrderpageTest.test_login
PASSED: com.spotyshoe.Tests.OrderpageTest.test_click_orders
PASSED: com.spotyshoe.Tests.OrderpageTest.test_getTitle_page

===============================================
    Default test
    Tests run: 3, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
===============================================
```

Added to Cart Test:

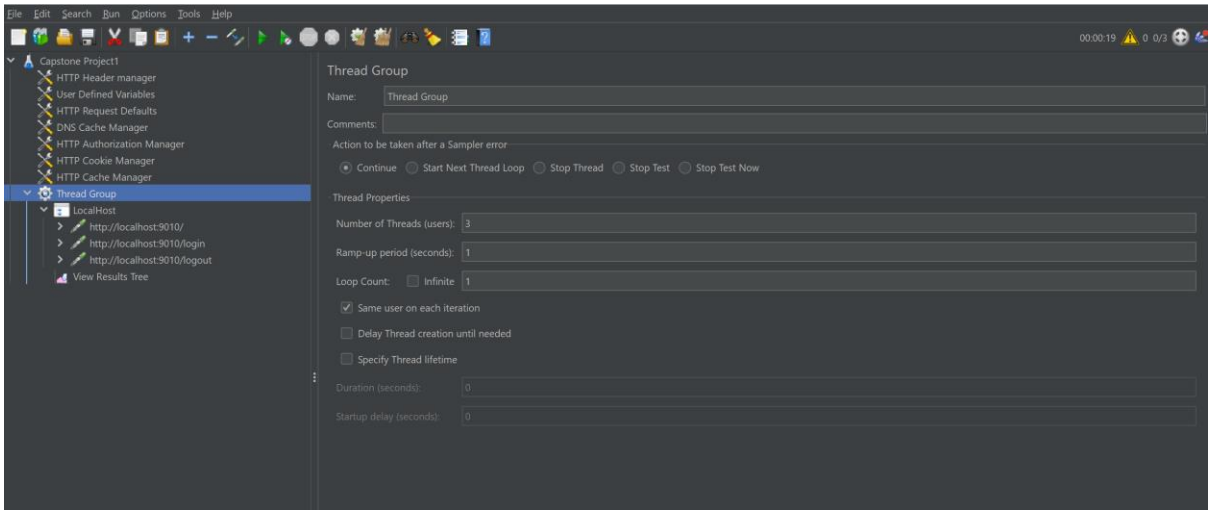
```java
import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import com.spotyshoe.selenium.AddtoCartPage;
import com.spotyshoe.selenium.HomePage;
import com.spotyshoe.selenium.LoginPage;
import com.spotyshoe.selenium.RegisterPage;
import com.spotyshoe.selenium.TestBase;

Run All
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);}

    Run | Debug
    @Test(priority='1')public void test_login()

    {
        lp.user_login();
    }


    @Test(priority='2')
    Run | Debug
    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);}
    }
```



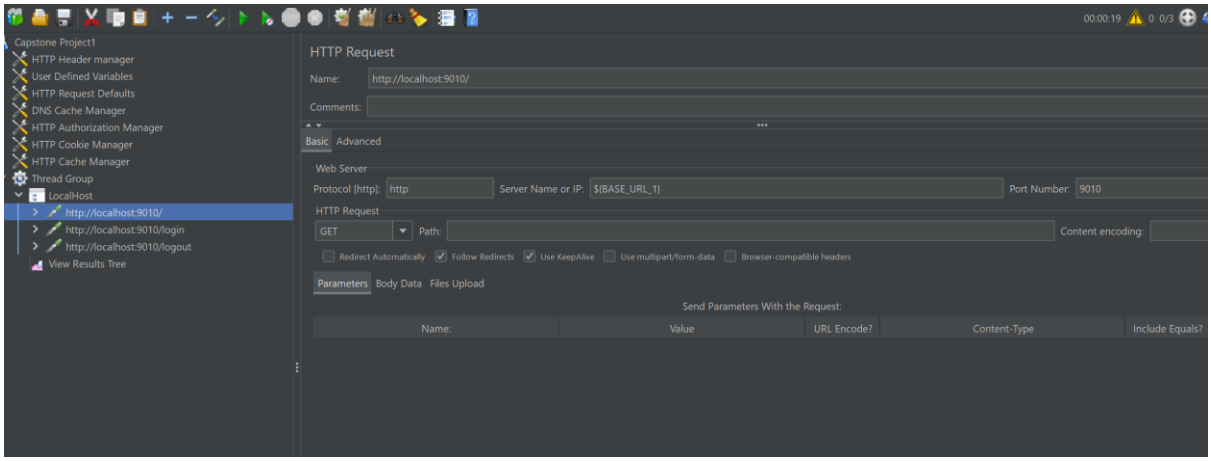
3.JMETER SCRIPT

Recording of script has been done via blazemeter of local host 9010. Login and logout .

Script has been recorded and we create a thread group , to load the Thread users to run on server.

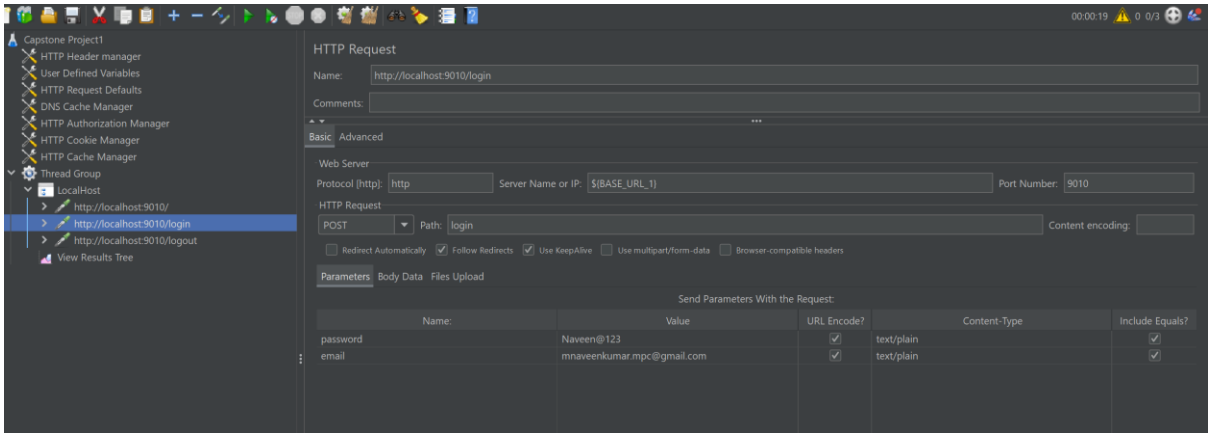


Added three requests - home page, login page , logoutpage.

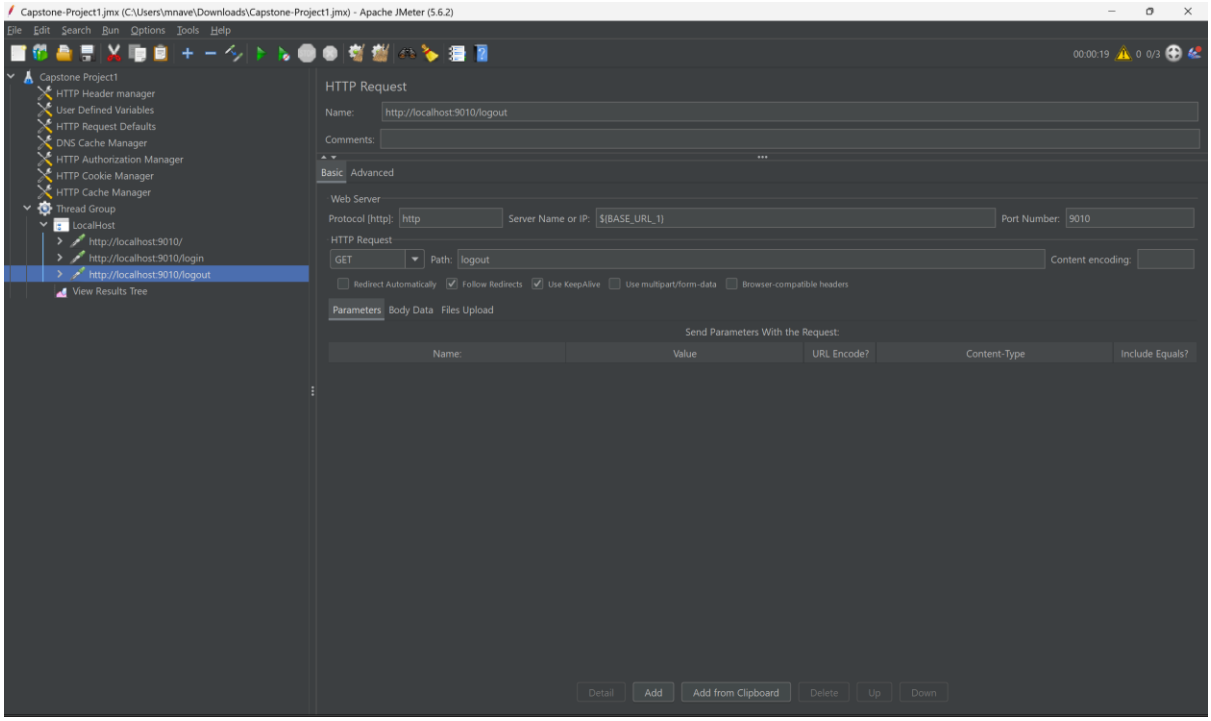


LoginPage :

User Email
Password

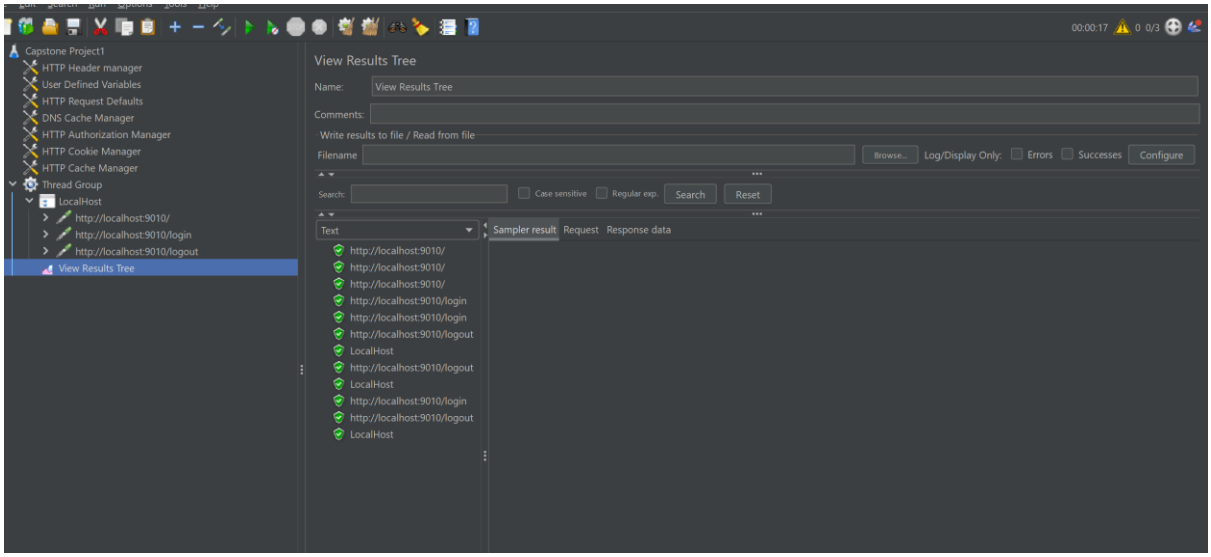


Logout :



Output:

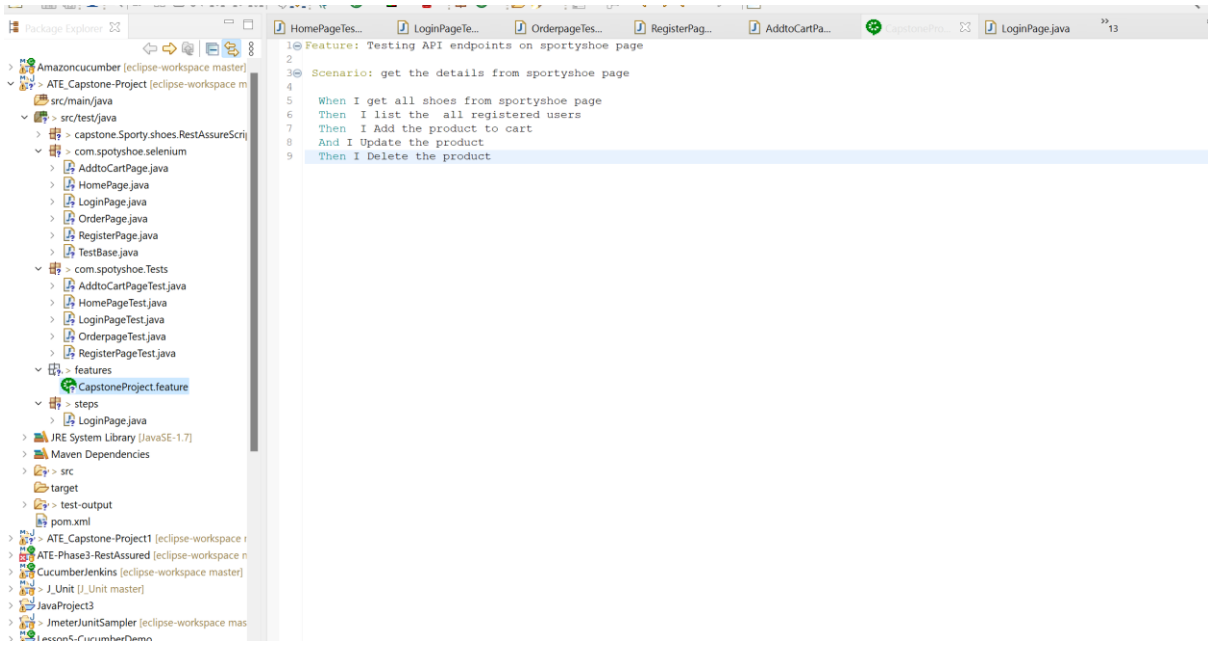
Thread Load given with 3 users , so each request executes 3 times overally 3 requests with 9 executions.



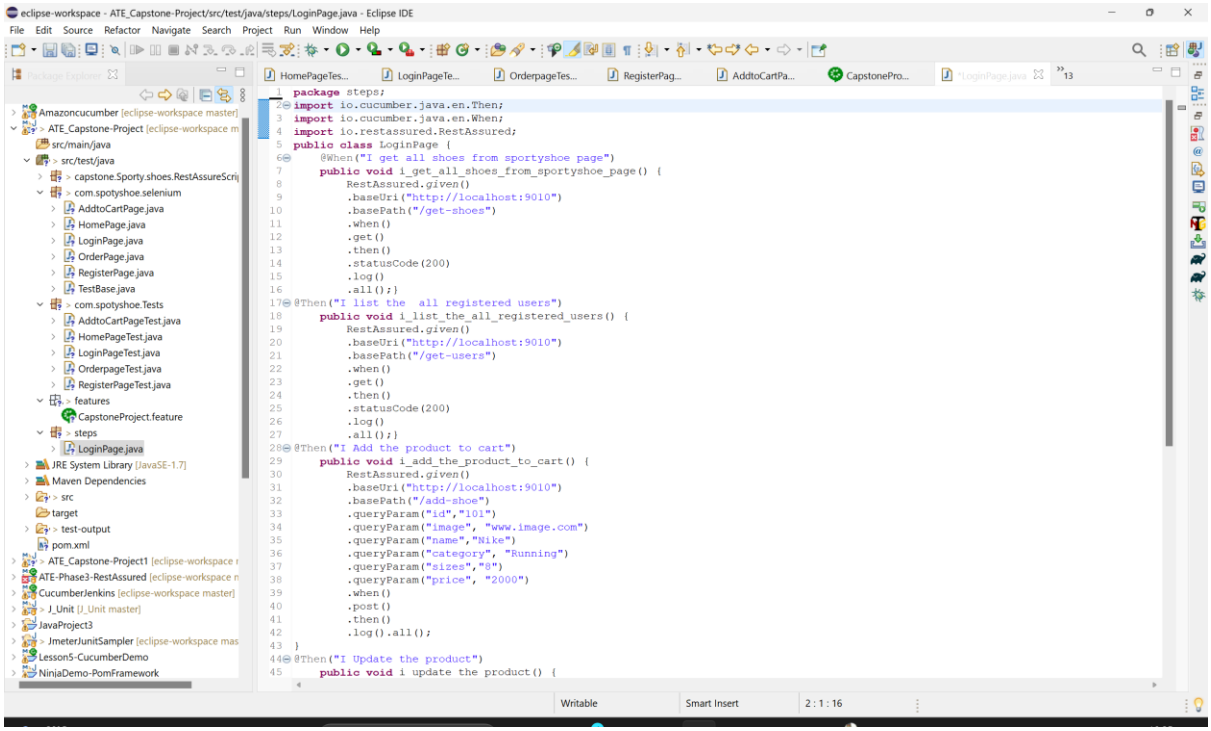
4 . CUCUMBER

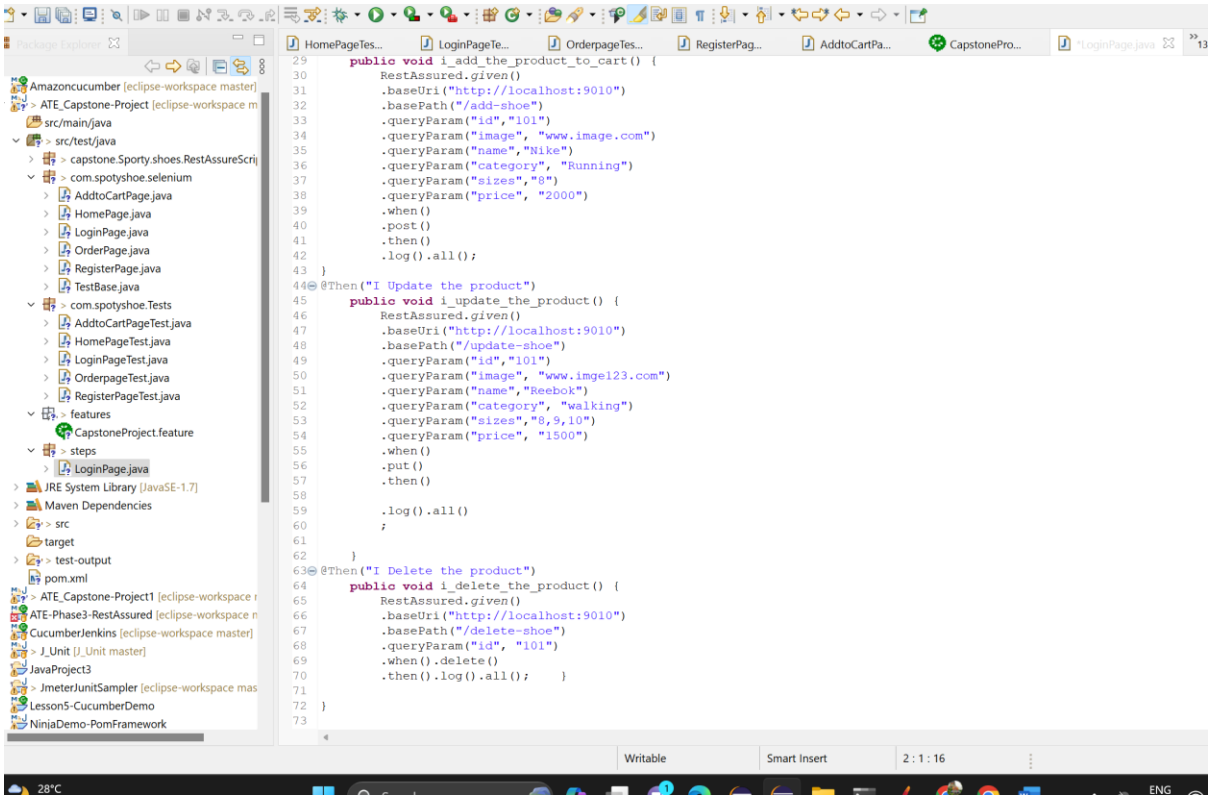
Feature File:

Cucumber file with feature file and scenario.



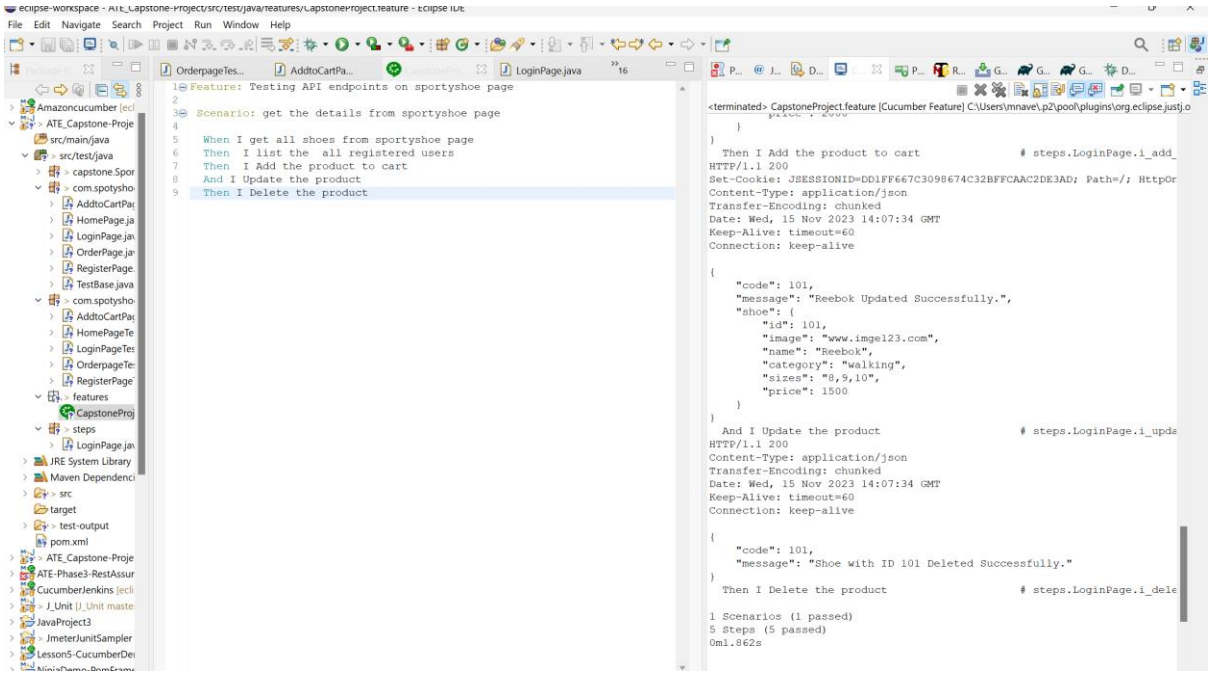
STEPS:





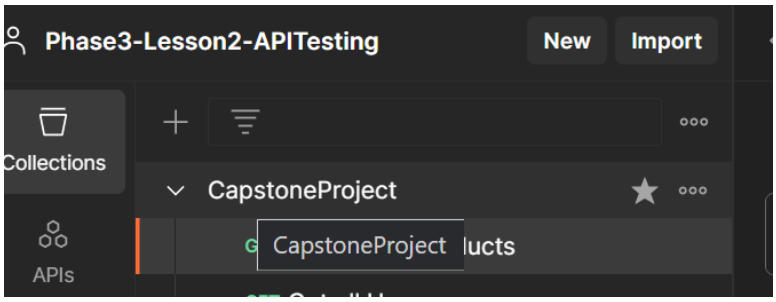

```
29      public void i_add_the_product_to_cart() {
30          RestAssured.given()
31          .baseUri("http://localhost:9010")
32          .basePath("/add-shoe")
33          .queryParam("id","101")
34          .queryParam("image", "www.image.com")
35          .queryParam("name","Nike")
36          .queryParam("category", "Running")
37          .queryParam("sizes","8")
38          .queryParam("price", "2000")
39          .when()
40          .post()
41          .then()
42          .log().all();
43  }
44  @Then("I Update the product")
45      public void i_update_the_product() {
46          RestAssured.given()
47          .baseUri("http://localhost:9010")
48          .basePath("/update-shoe")
49          .queryParam("id","101")
50          .queryParam("image", "www.imge123.com")
51          .queryParam("name","Reebok")
52          .queryParam("category", "walking")
53          .queryParam("sizes","8,9,10")
54          .queryParam("price", "1500")
55          .when()
56          .put()
57          .then()
58
59          .log().all()
60          ;
61
62      }
63  @Then("I Delete the product")
64      public void i_delete_the_product() {
65          RestAssured.given()
66          .baseUri("http://localhost:9010")
67          .basePath("/delete-shoe")
68          .queryParam("id", "101")
69          .when().delete()
70          .then().log().all();    }
71
72  }
73
```

OUTPUT:



5 .API Testing With Postman And Rest Assured

In PostMan First we need to create Collection and name it as Capstone Project and followed with some methods to test.



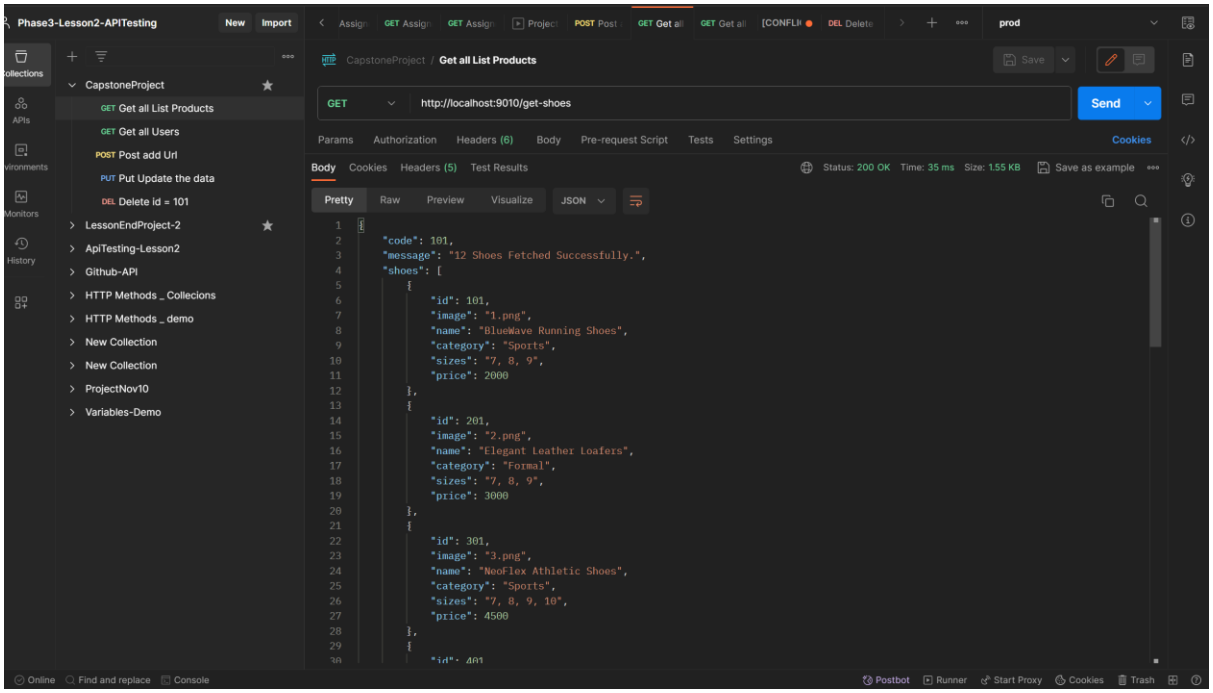
Methods Used:

Get
Post
Put
Delete

1.Getting the list of all products in the store :

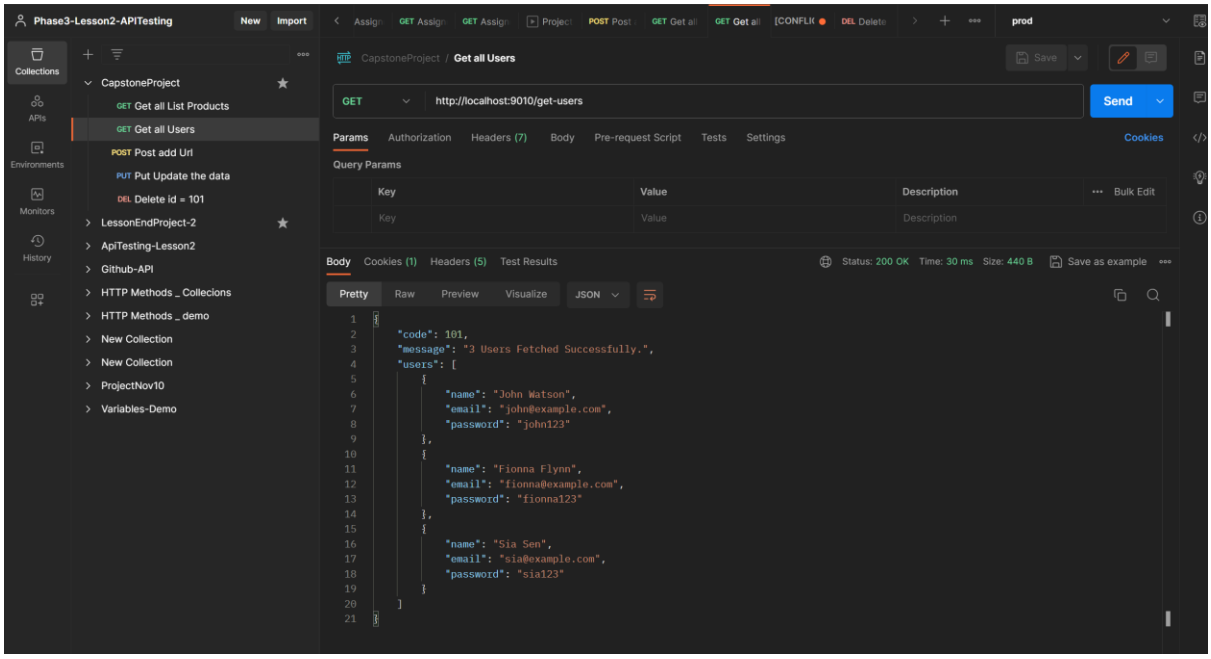
- Now Getting list of all products has done with GET Method And Url is (http://localhost:9010/get-shoes)

OutPut :



2 . Getting all Users

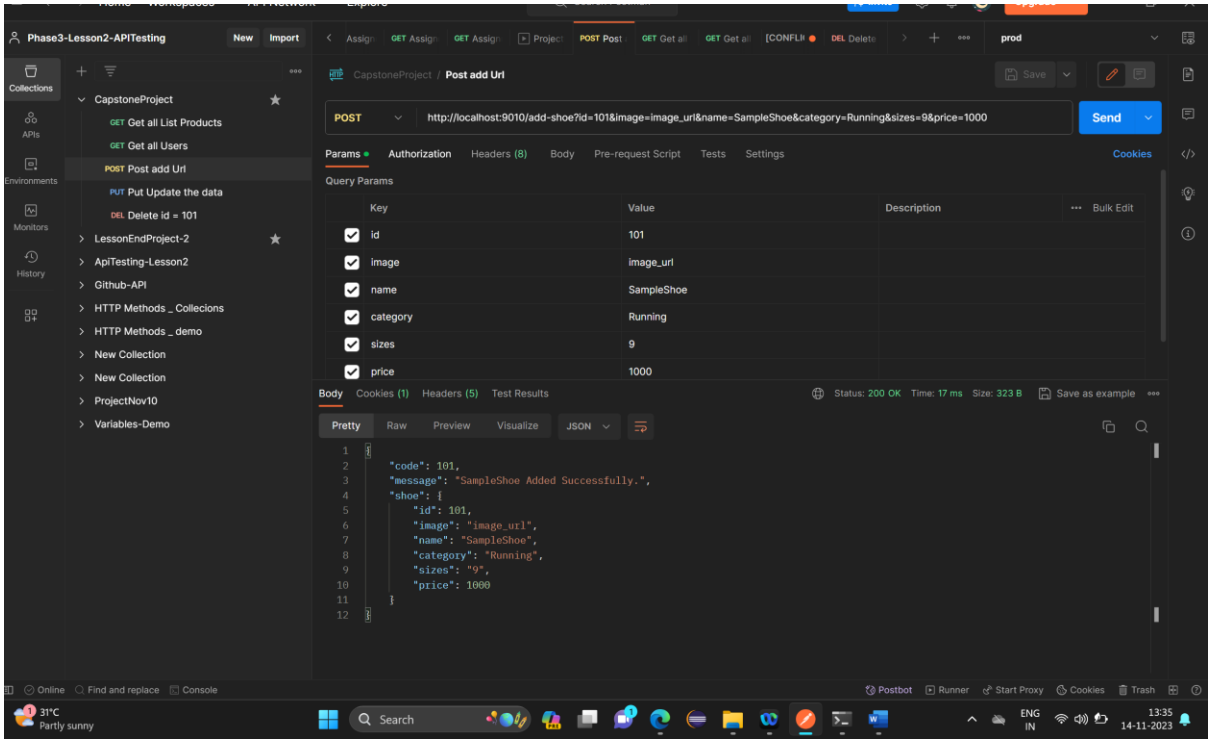
- Get method used to get all users from sporty shoes
- Status code Ok with 200.



3 . Add the Product

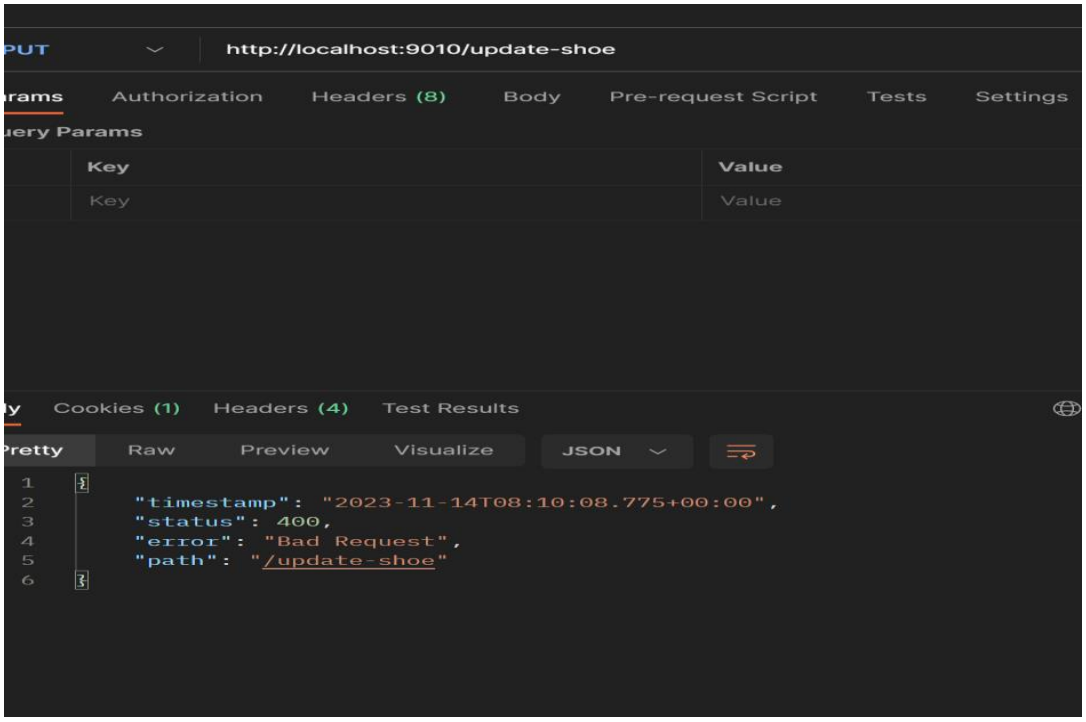
- Product has been added to list using method POST.
- Product id=101 and name =”SampleShoe” has been added.
- These Post method adds the data to the server or given url database.
- Post method Defaulty gives Query Parameters to add the data to sporty shoes.

OutPut:



Bugs :

- Updating data using put method is not fine ..gives error with given url shown below.
- So to overcome I used Post Method url and Post method output data as body in put method.



4 . Update the Data

- Data has been updated to Spoty shoes using Put method.
- Body given as below to update spoty shoes

{

"id": 101,

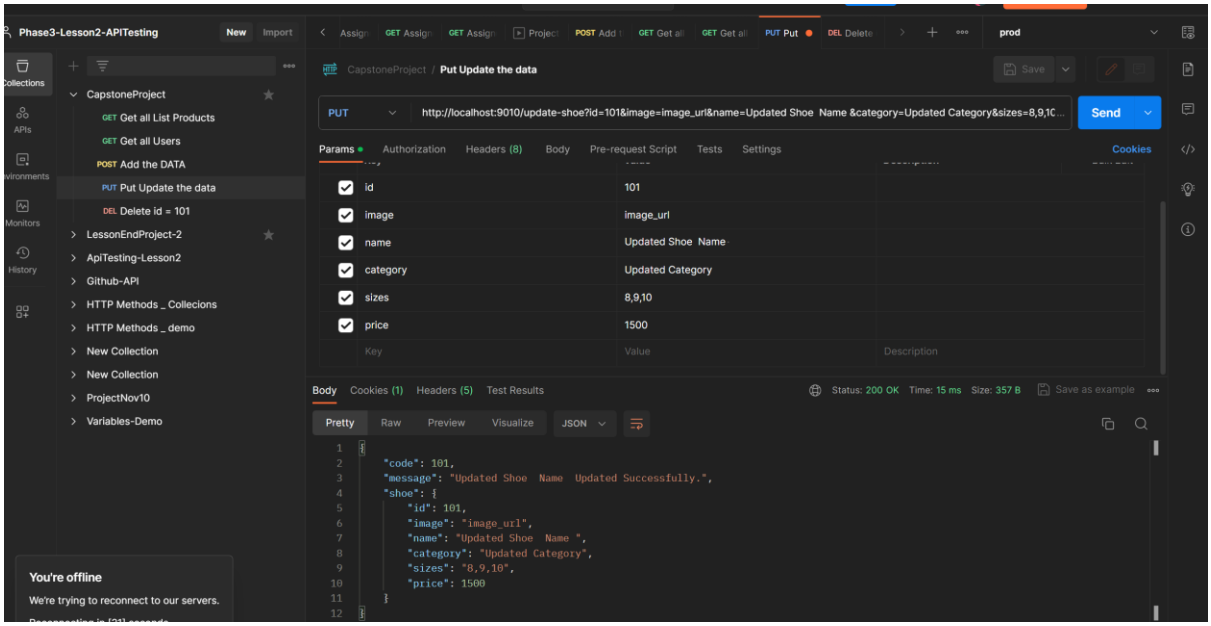
"name": "Updated Shoe Name",

"category": "Updated Category",

"sizes": "8,9,10",

"price": 1500

}



5. Delete ID

- Id has been Deleted using DELETE Method having ID number .
- Now deleting id with number ID = 101

Above ID is deleted successfully: