Capstone-Project: **Create a Testing Framework for Sporty Shoes Website**

==========

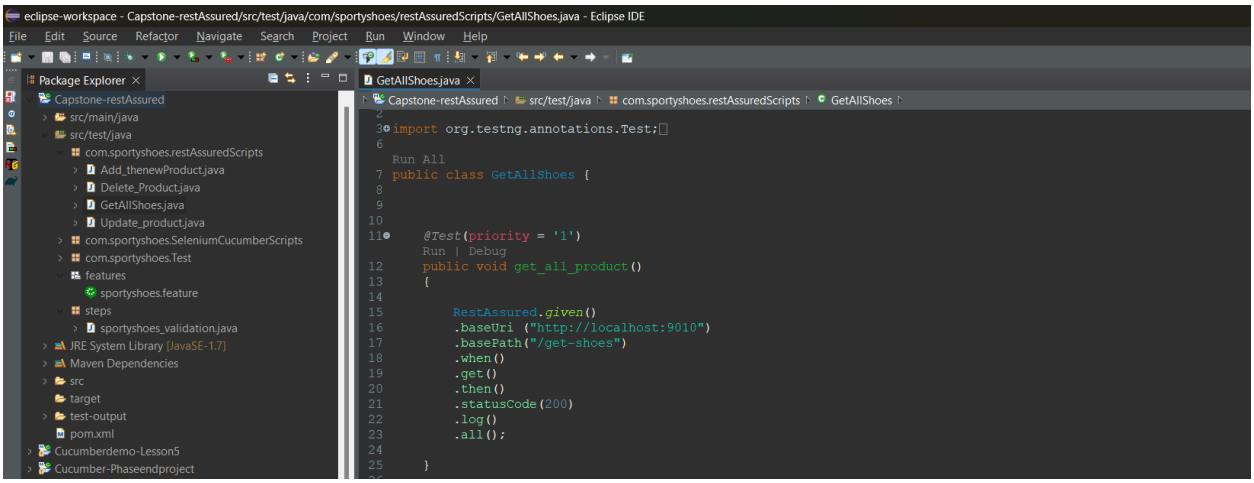
Screenshots:-

1. Automate the below API endpoints using Rest-Assured

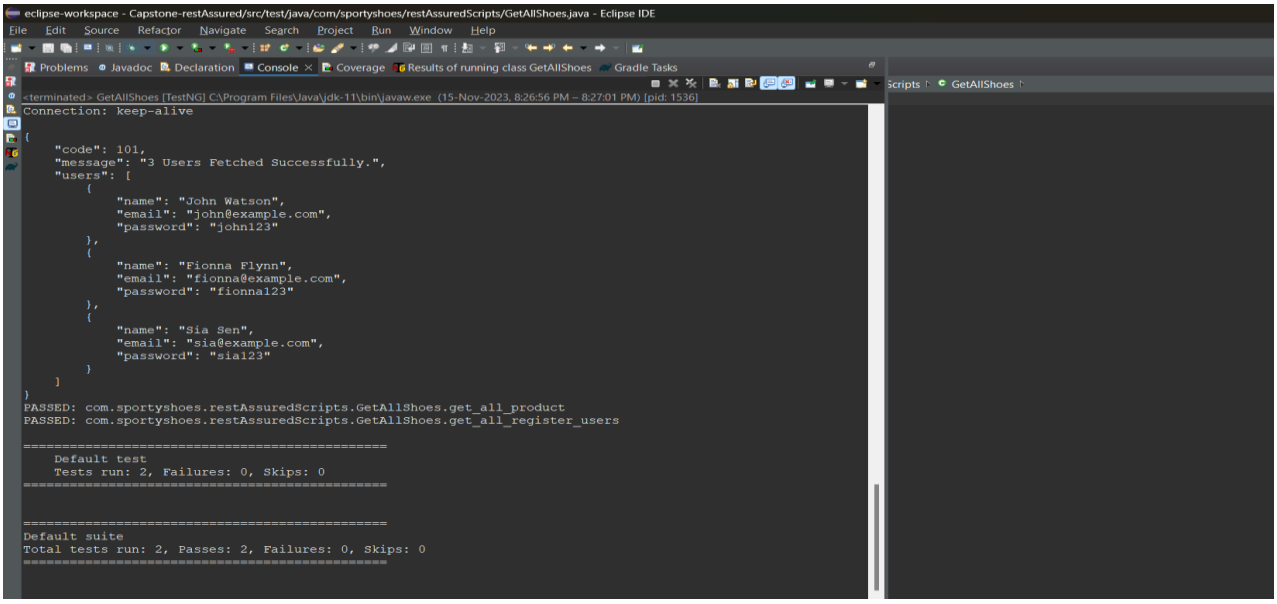
==========

- **Here below we are retrieving the list of products in the store using Rest-Assured**:

==========



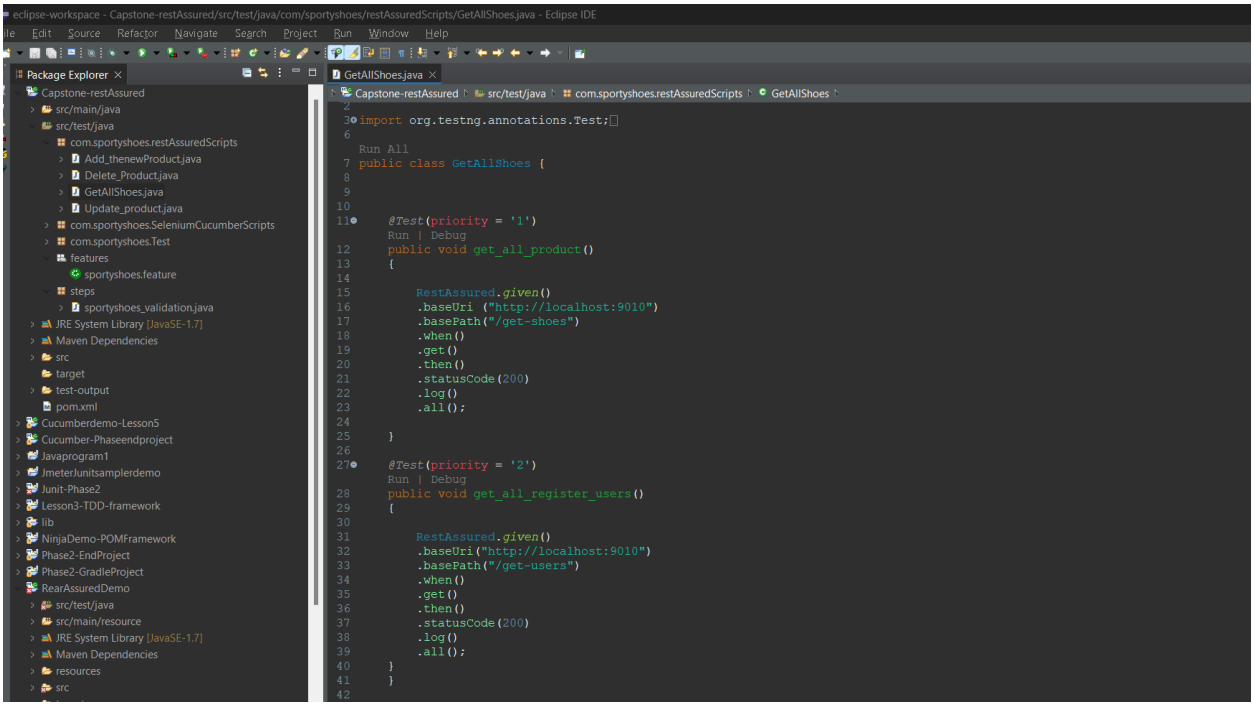
Output response:



==========

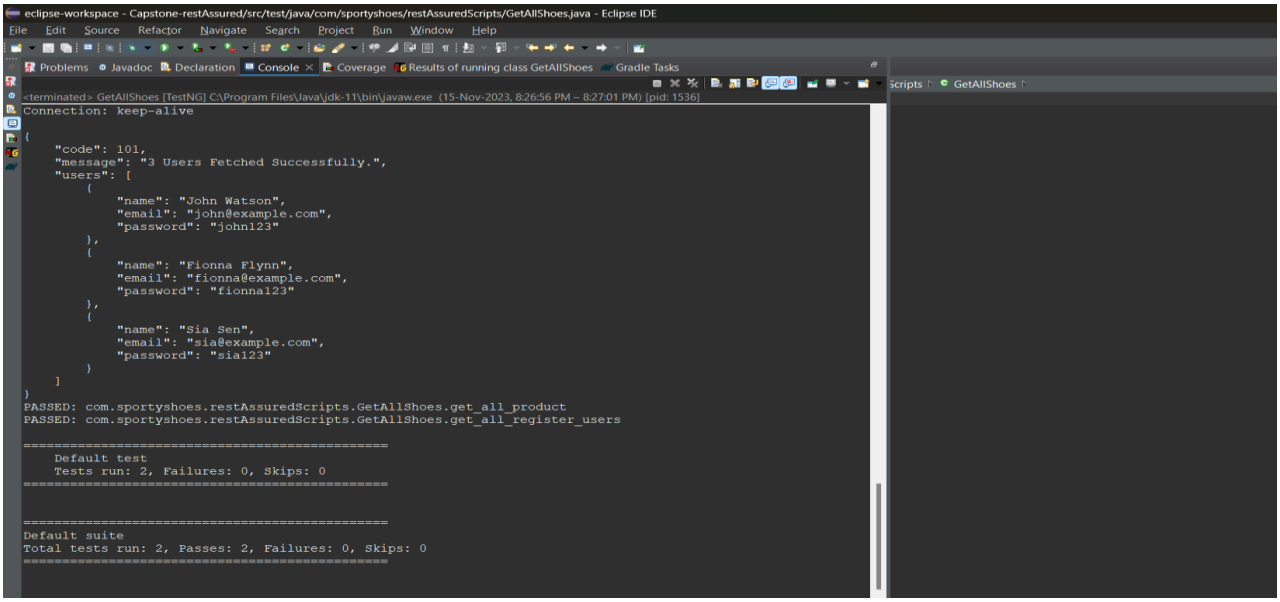
- **Retrieving the list of all registered users:**

==========



```
import org.testng.annotations.Test;

Run All
public class GetAllShoes {


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

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

    }

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

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

Output response:

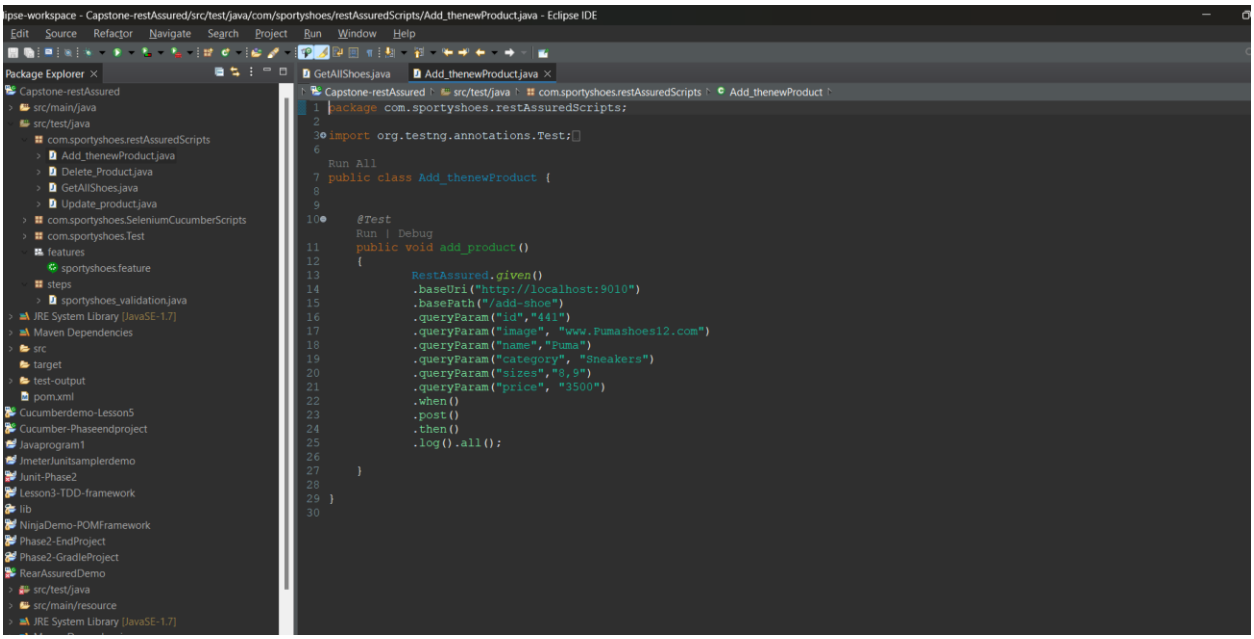


```
Connection: keep-alive

{
    "code": 101,
    "message": "3 Users Fetched Successfully.",
    "users": [
        {
            "name": "John Watson",
            "email": "john@example.com",
            "password": "john123"
        },
        {
            "name": "Fionna Flynn",
            "email": "fionna@example.com",
            "password": "fionna123"
        },
        {
            "name": "Sia Sen",
            "email": "sia@example.com",
            "password": "sia123"
        }
    ]
}
PASSED: com.sportyshoes.restAssuredScripts.GetAllShoes.get_all_product
PASSED: com.sportyshoes.restAssuredScripts.GetAllShoes.get_all_register_users

===============================================
    Default test
    Tests run: 2, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

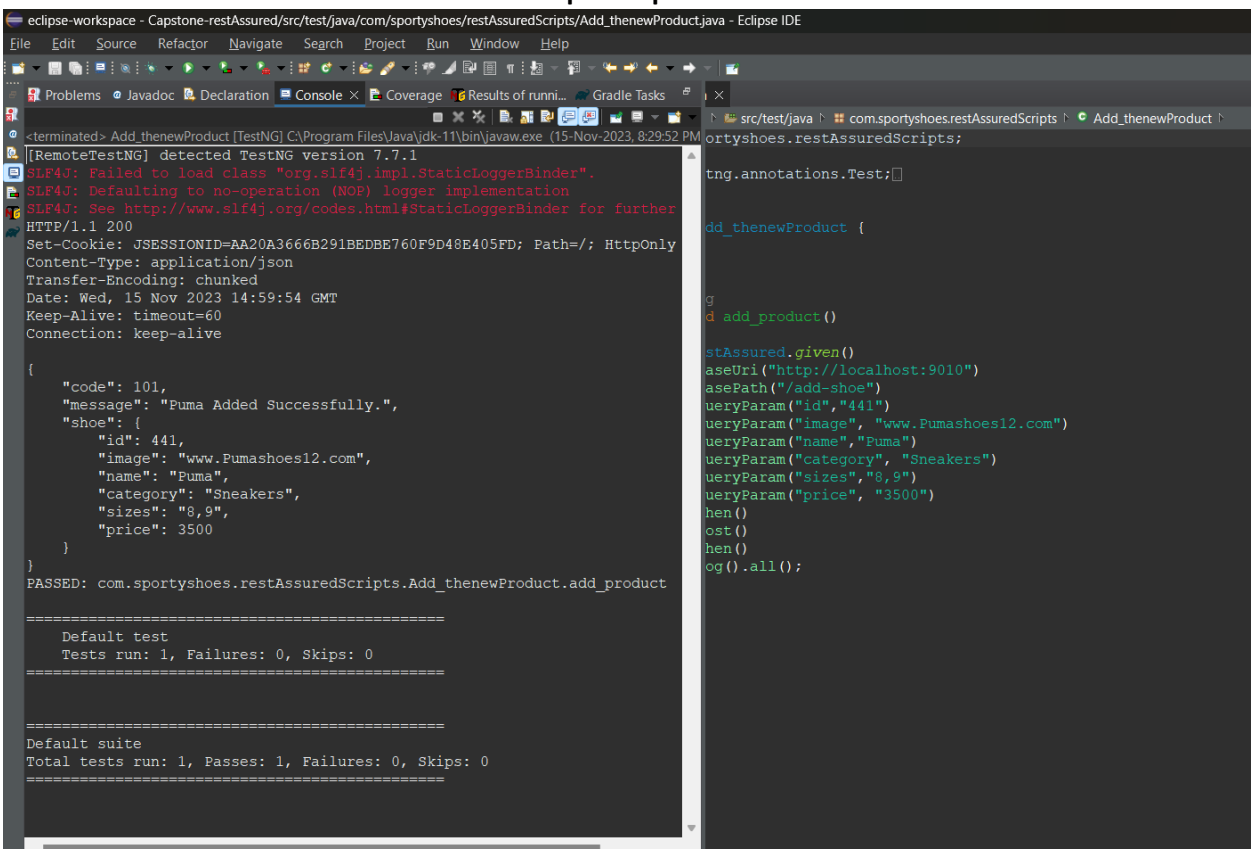
==========

- **Add the Products:**

  ==========



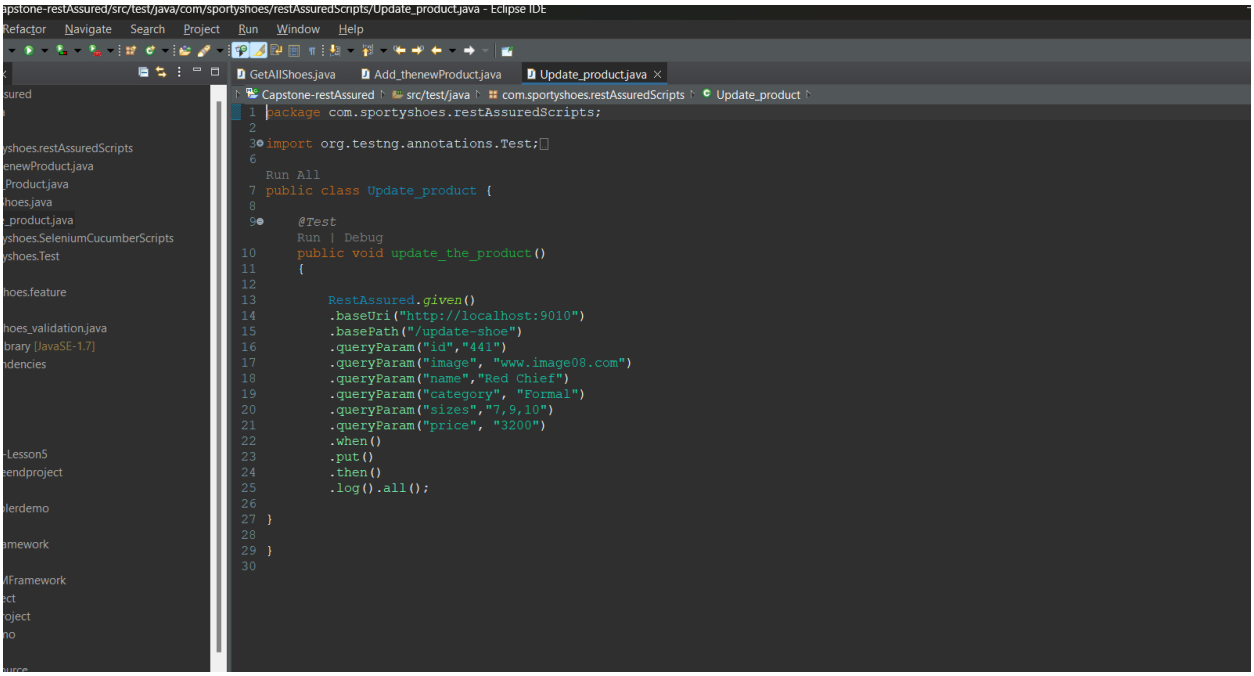
Output response:



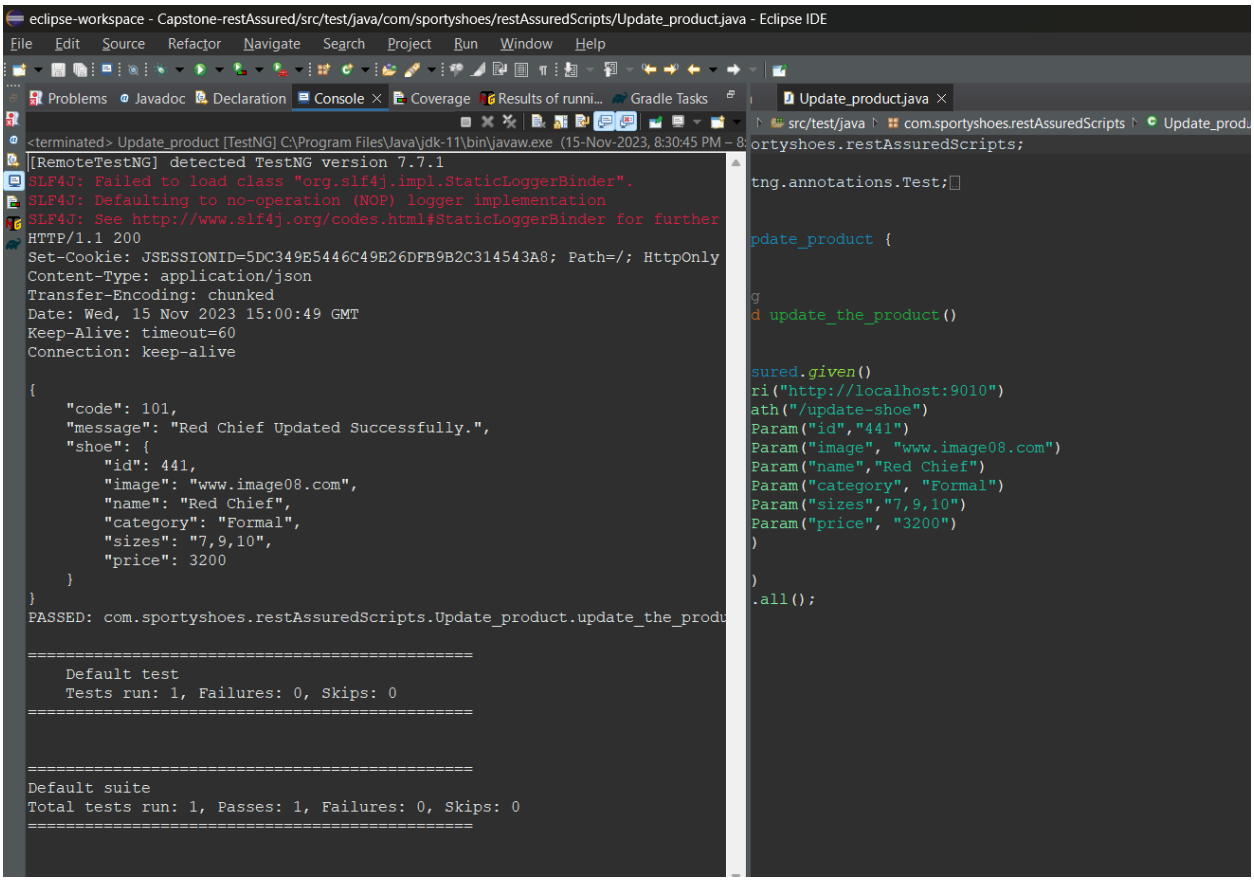
==========

- **Update the products:**

==========



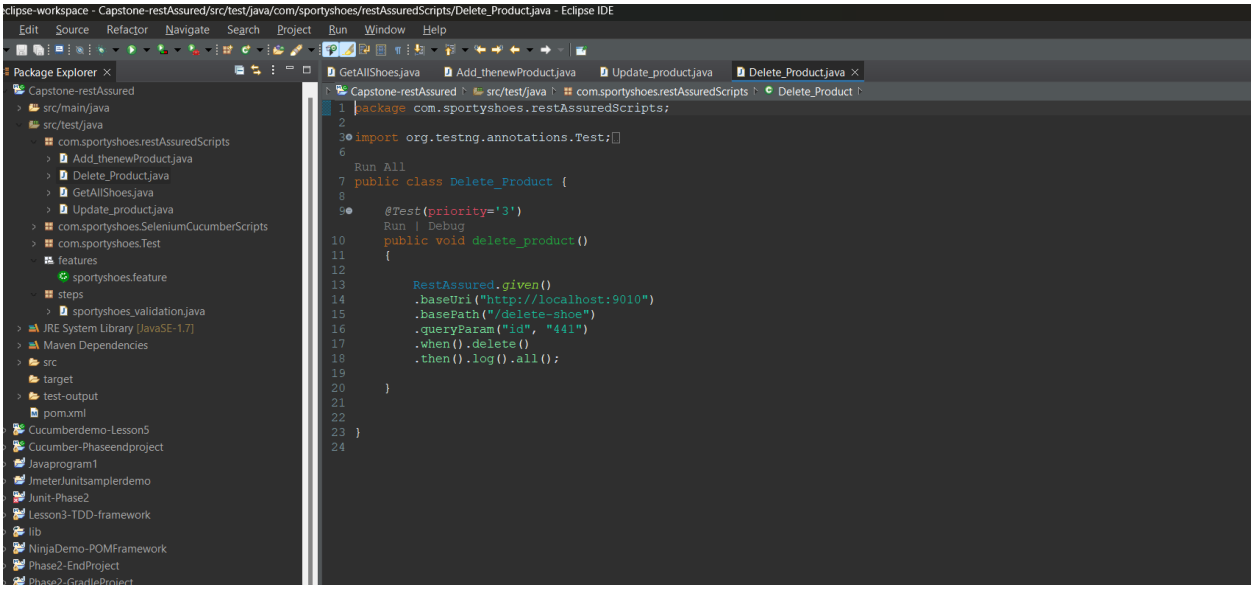
Output response:



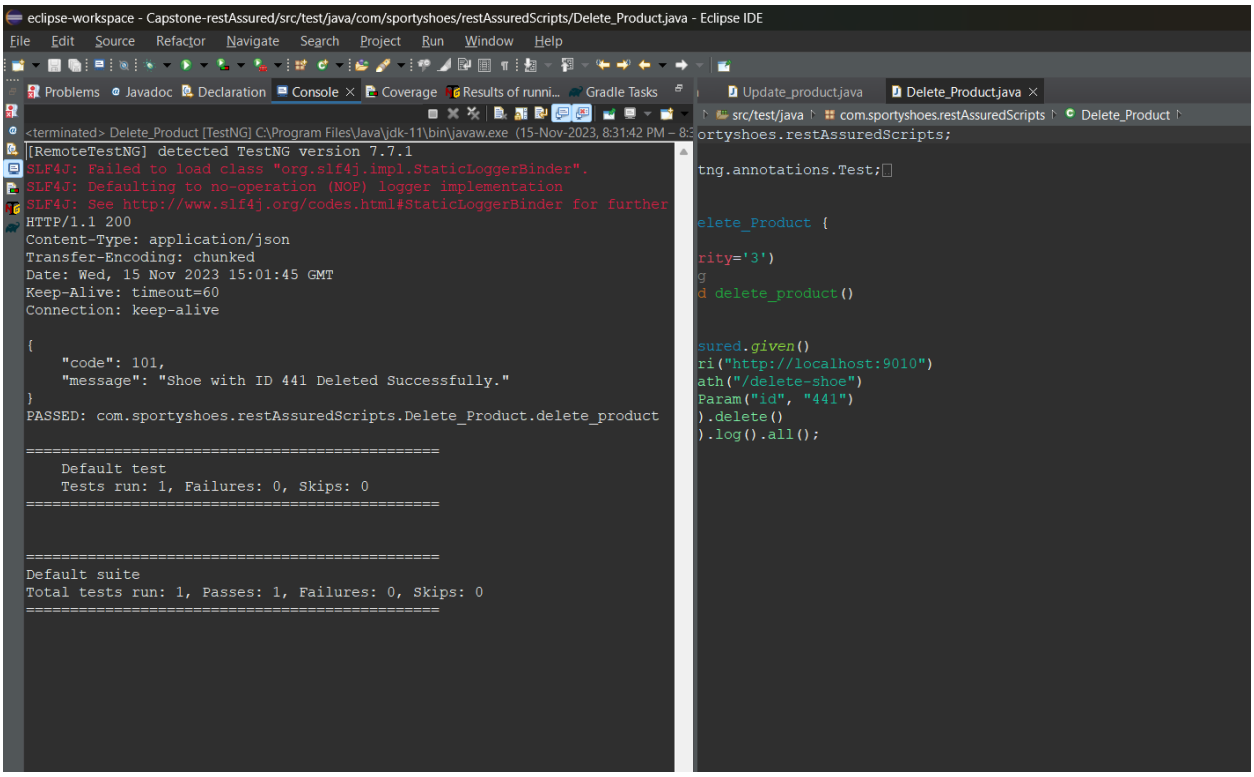
==========

- **Delete the product:**

  ==========



Output response:



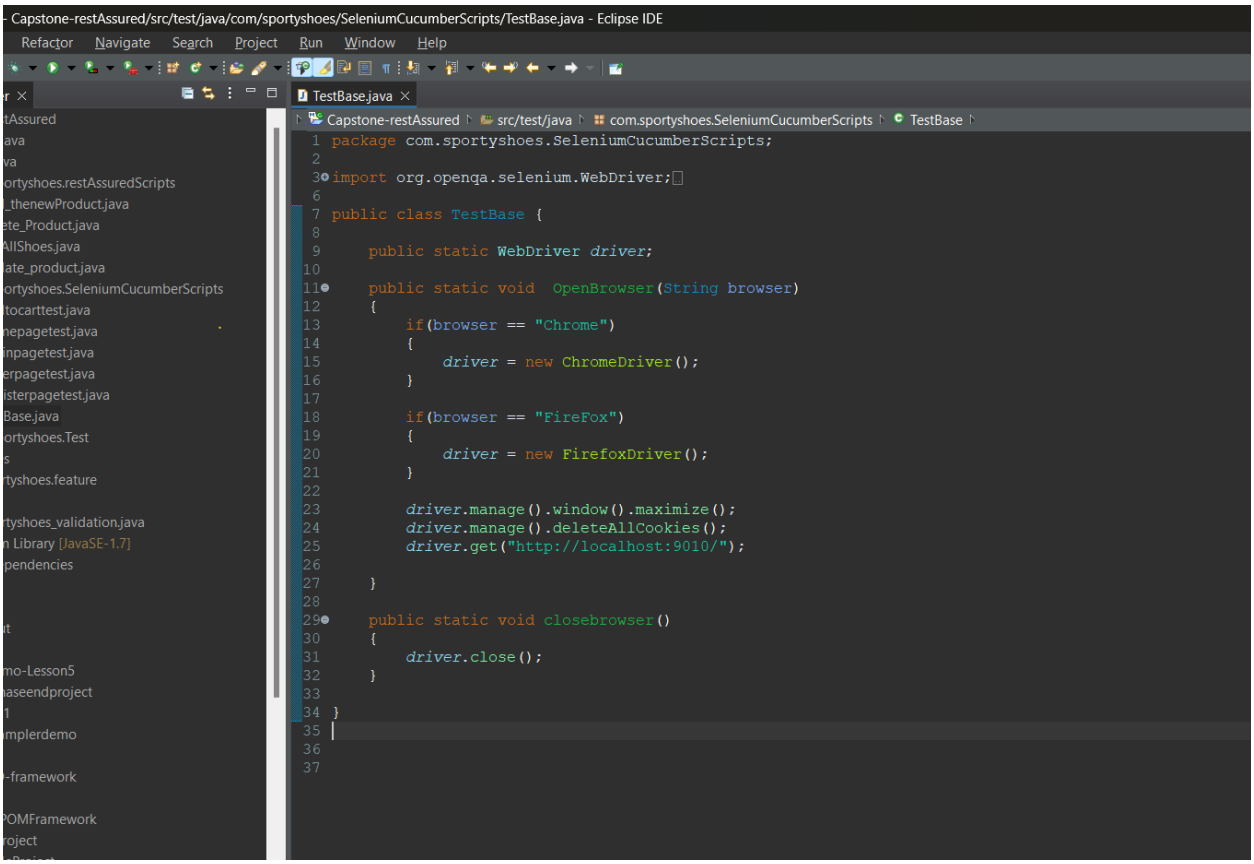
==========

2. **Selenium Scripts using TestNG to test all the pages with URL: http://localhost:9010/**

==========

TestBase.java class:

==========

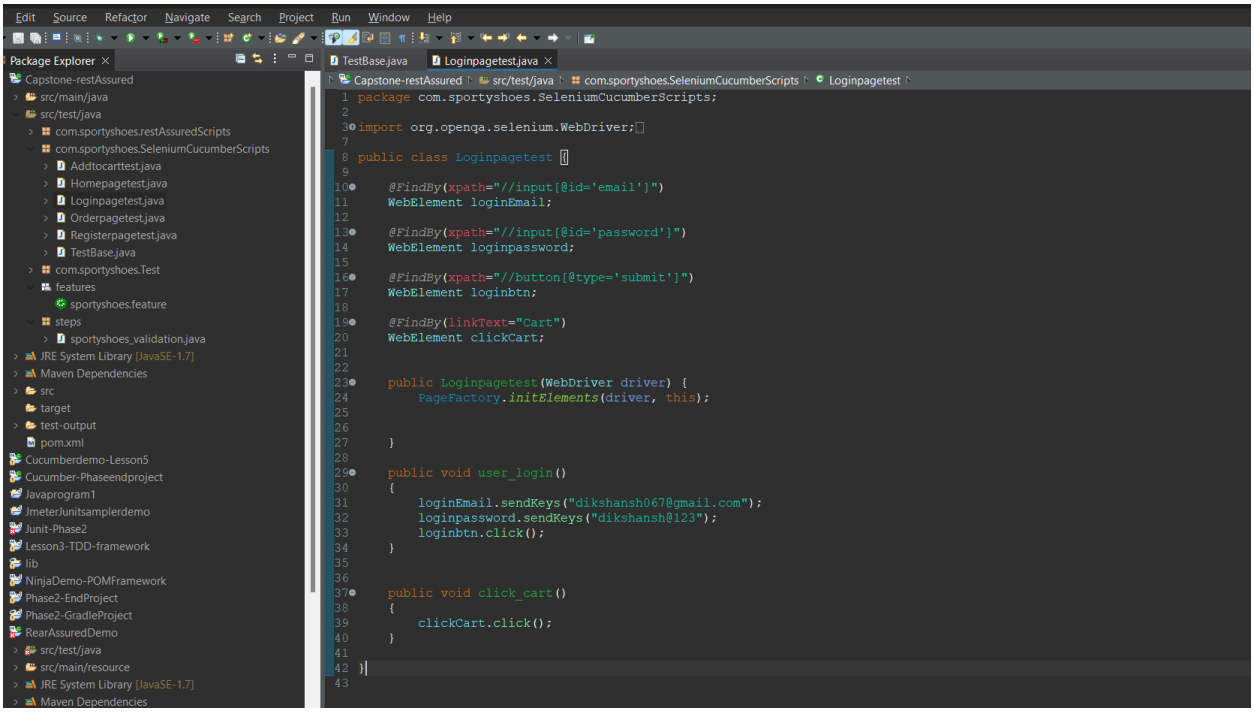


==========

- **Selenium script for login page**:

Loginpage.java class:

==========

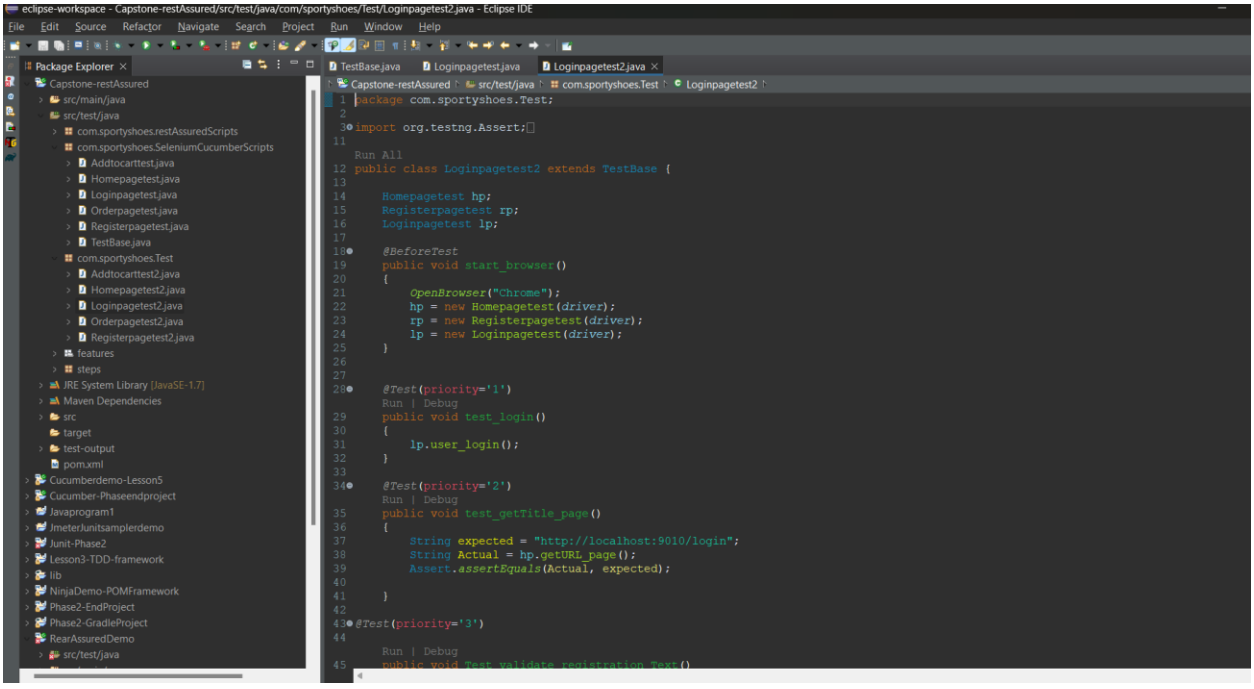


```
package com.sportyshoes.SeleniumCucumberScripts;

import org.openqa.selenium.WebDriver;

public class Loginpagetest {

    @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 Loginpagetest(WebDriver driver) {
        PageFactory.initElements(driver, this);


    }

    public void user_login()
    {
        loginEmail.sendKeys("dikshansh067@gmail.com");
        loginpassword.sendKeys("dikshansh@123");
        loginbtn.click();
    }


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

}
```

==========

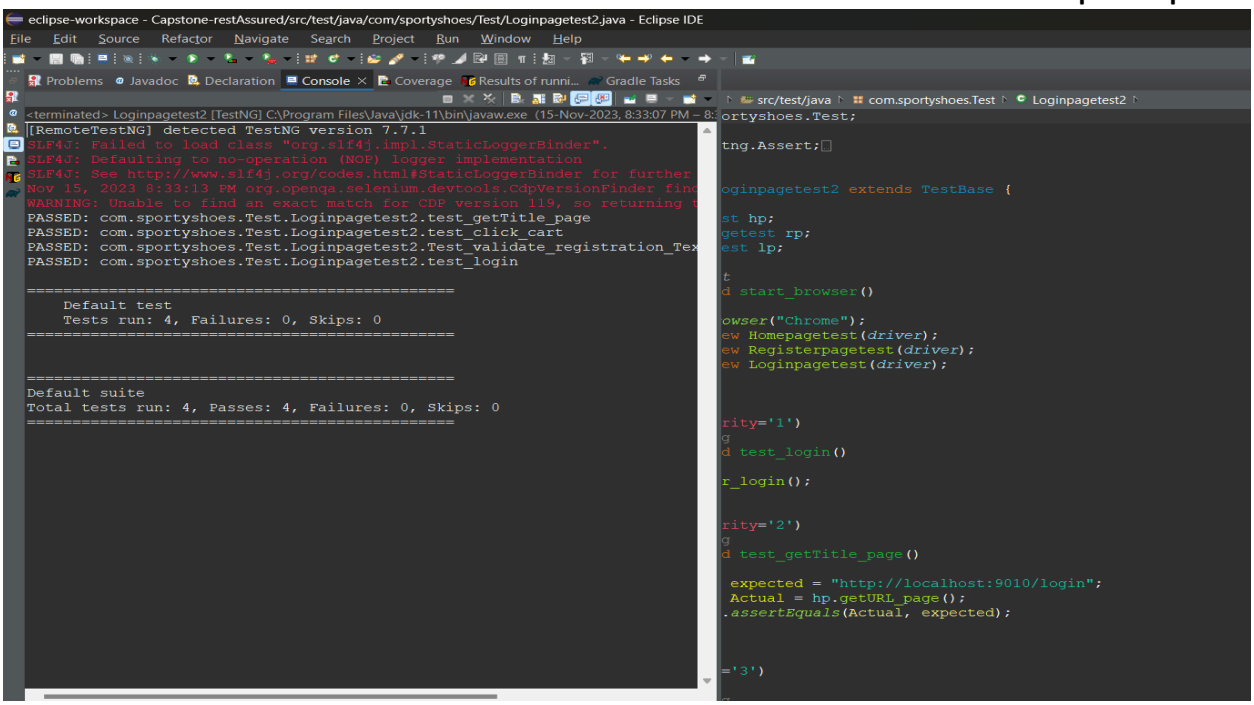
Loginpagetest.java

==========



```
package com.sportyshoes.Test;

import org.testng.Assert;

Run All
public class Loginpagetest2 extends TestBase {

    Homepagetest hp;
    Registerpagetest rp;
    Loginpagetest lp;

    @BeforeTest
    public void start_browser()
    {
        OpenBrowser("Chrome");
        hp = new Homepagetest(driver);
        rp = new Registerpagetest(driver);
        lp = new Loginpagetest(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()
```

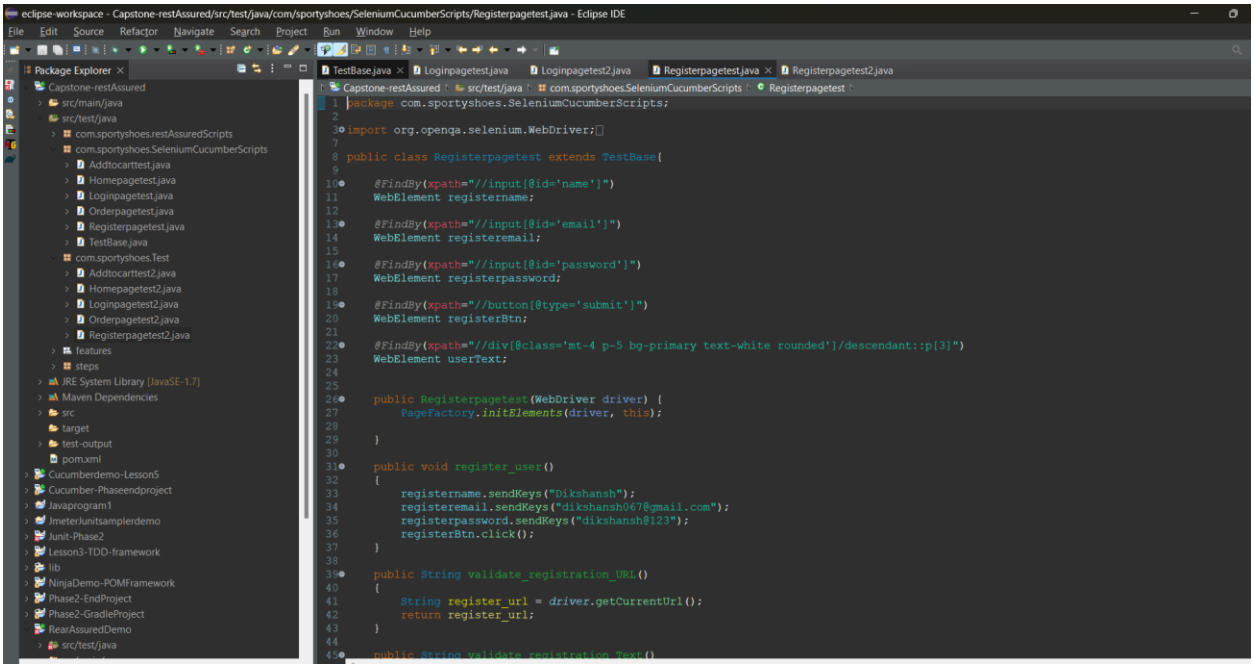
Output Response:



==========

Registrationpage.java class:

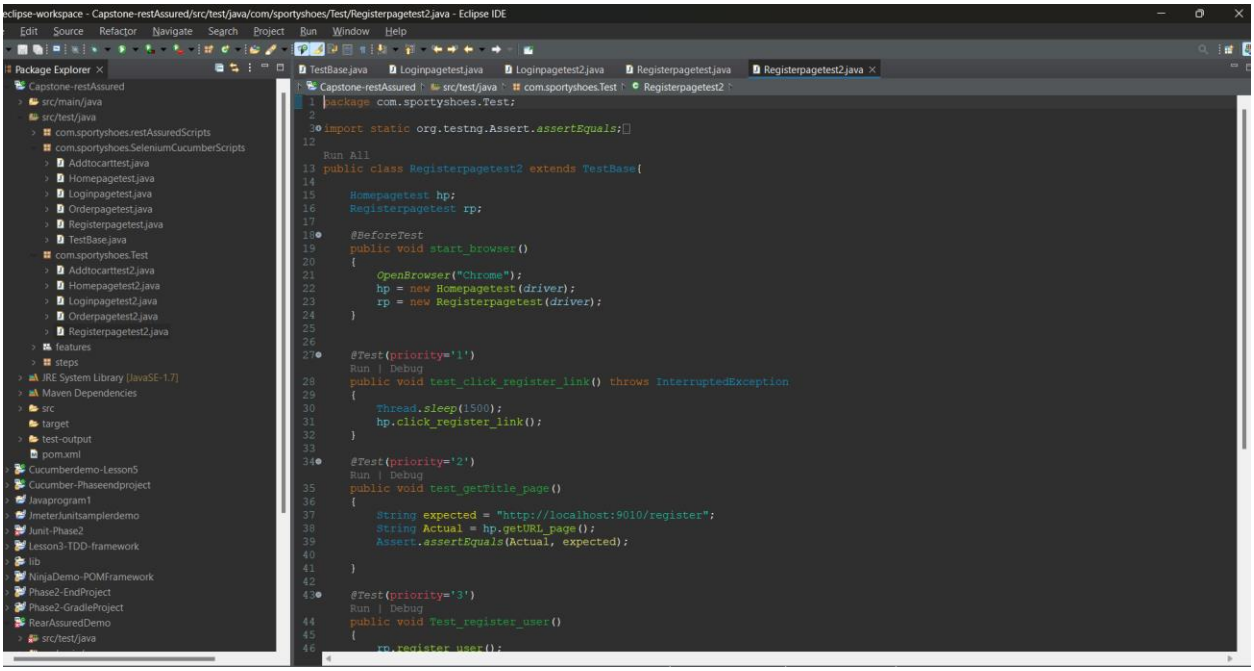
==========



==========

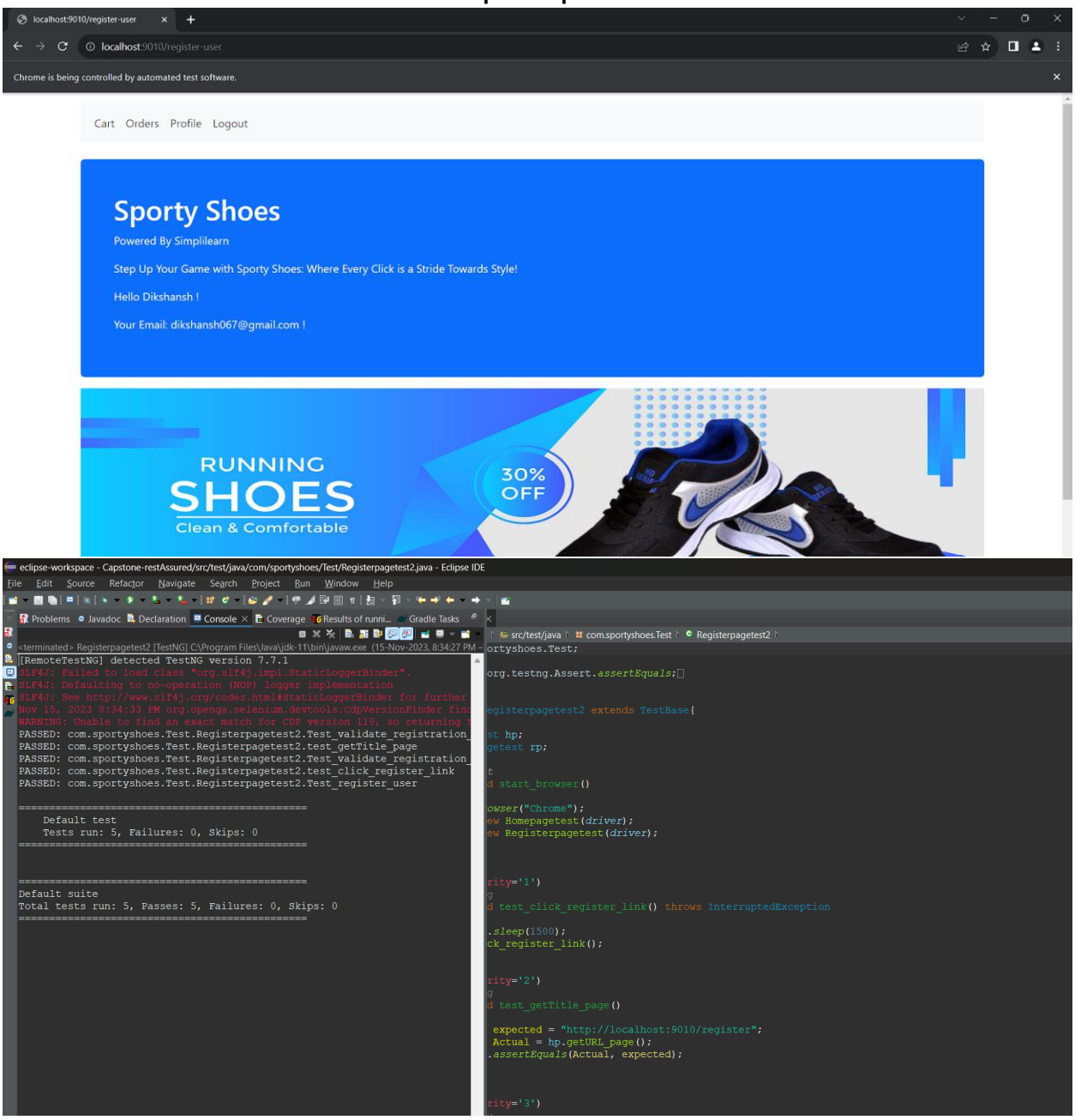
Registrationpagetest.java class:

==========




```
45     {
46         rp.register_user();
47     }
48
49     @Test(priority='4')
       Run | Debug
50     public void Test_validate_registration_URL()
51     {
52         String expected = "http://localhost:9010/register-user";
53         String Actual = rp.validate_registration_URL();
54         assertEquals(Actual, expected);
55     }
56
57 @Test(priority='5')
58
       Run | Debug
59     public void Test_validate_registration_Text()
60     {
61         String expected = "Hello Dikshansh !";
62         String actualText = rp.validate_registration_Text();
63         Assert.assertEquals(actualText, expected);
64     }
65
66 }
67
```

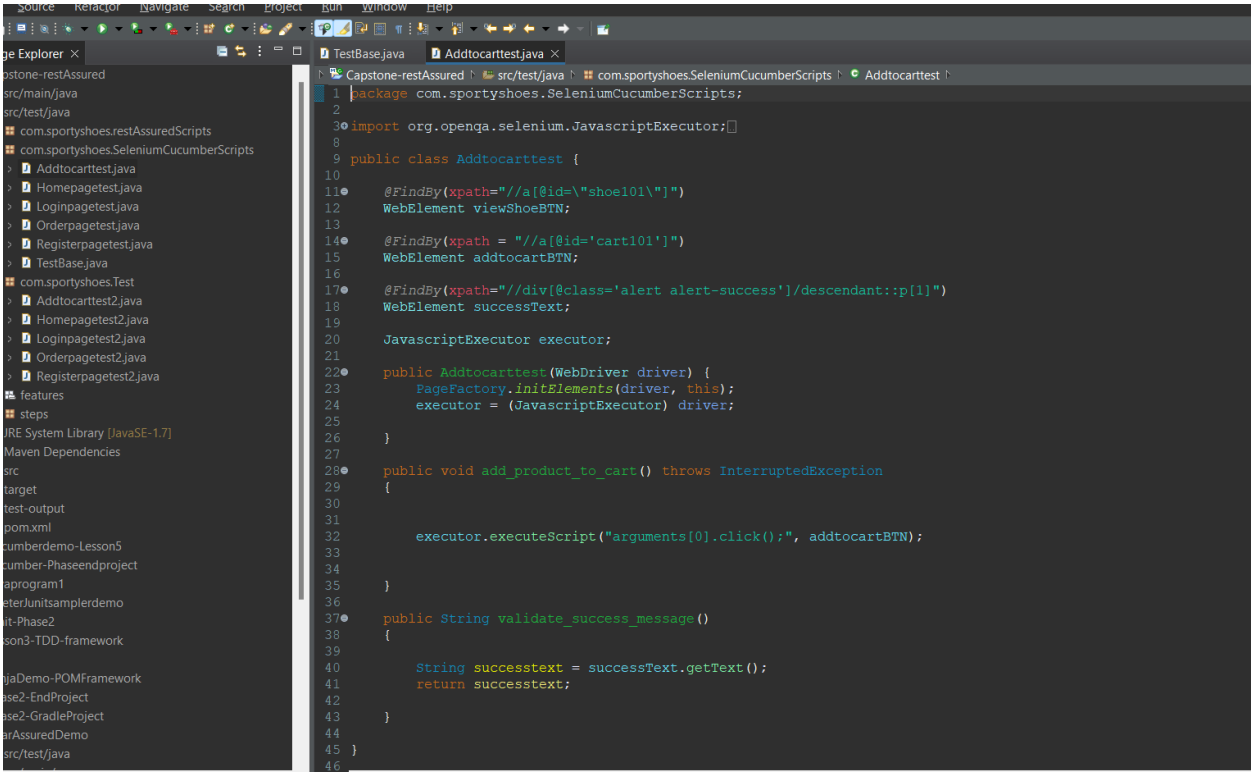
Output Response:



==========

Addproducttocart.java class:

==========

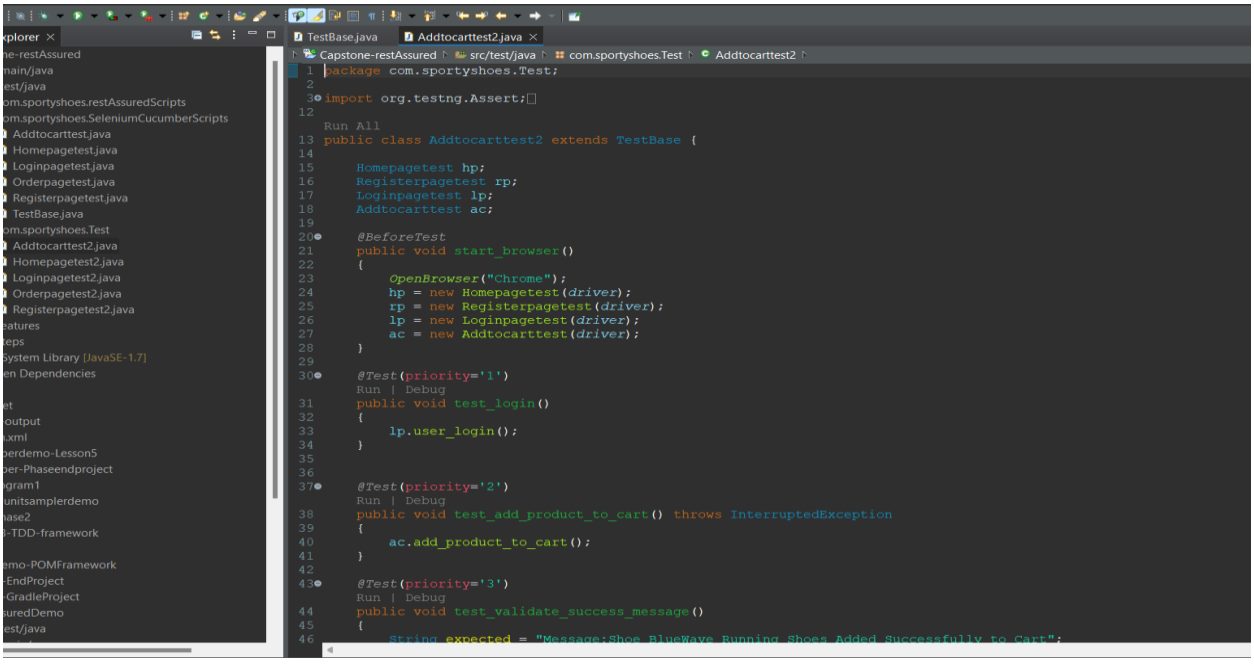


```
package com.sportyshoes.SeleniumCucumberScripts;

import org.openqa.selenium.JavascriptExecutor;

public class Addtocarttest {

    @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 Addtocarttest(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;

    }

}
```

==========

Addproducttocarttest.java class:

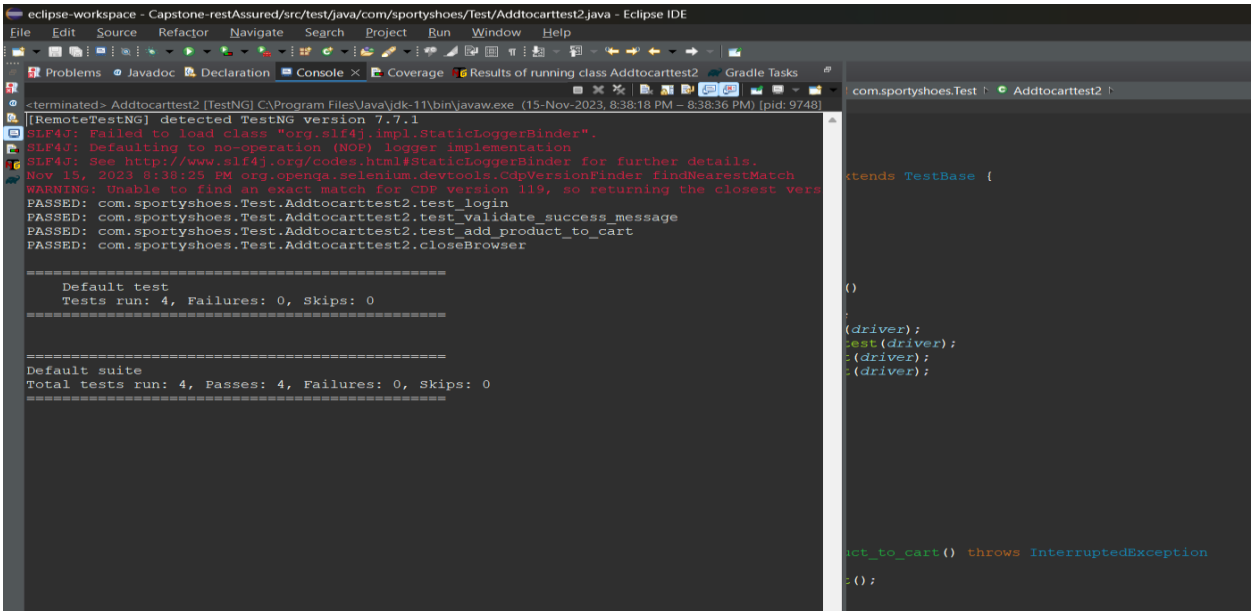
==========



```
package com.sportyshoes.Test;

import org.testng.Assert;

Run All
public class Addtocarttest2 extends TestBase {

    Homepagetest hp;
    Registerpagetest rp;
    Loginpagetest lp;
    Addtocarttest ac;

    @BeforeTest
    public void start_browser()
    {
        OpenBrowser("Chrome");
        hp = new Homepagetest(driver);
        rp = new Registerpagetest(driver);
        lp = new Loginpagetest(driver);
        ac = new Addtocarttest(driver);
    }

    @Test(priority='1')
    Run | Debug
    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')
    Run | Debug
    public void test_validate_success_message()
    {
        String expected = "Message:Shoe BlueWave Running Shoes Added Successfully to Cart";
```

```
46          String expected = "Message:Shoe BlueWave Running Shoes Added Successfully to Cart";
47      String actualText=  ac.validate_success_message();
48      Assert.assertEquals(actualText, expected);
49      }
50
51      @Test (priority='4')
        Run | Debug
52      public void closeBrowser()
53      {
54
55          driver.close();
56      }
57  }
58
59
```

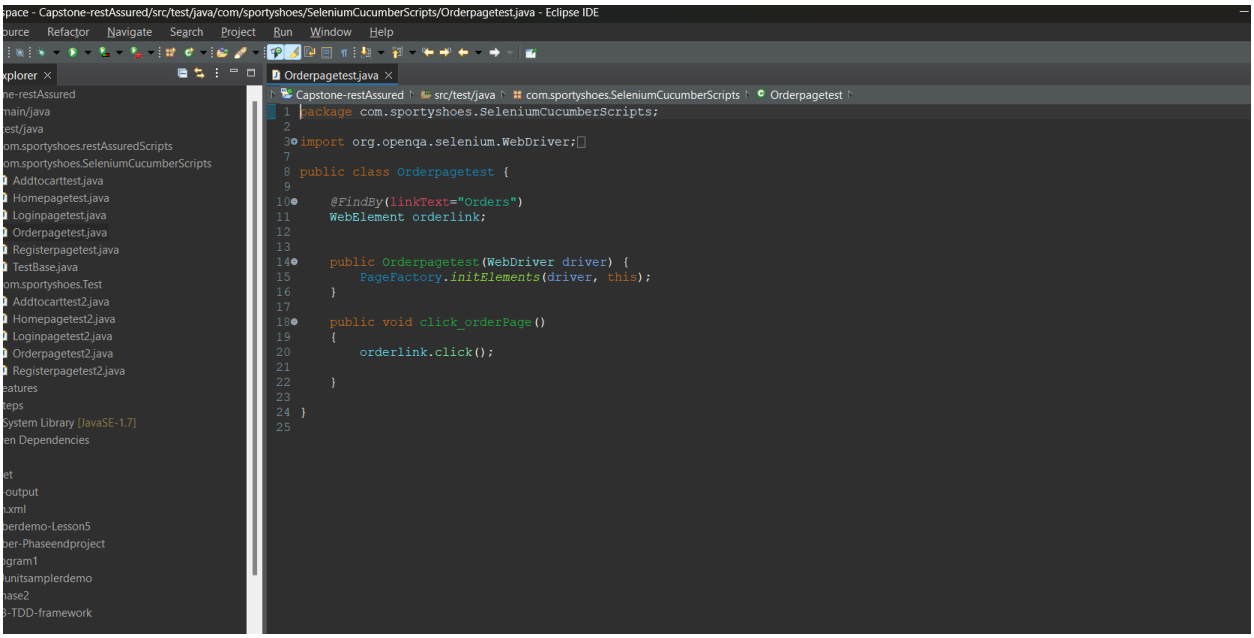
Output Response:



==========

Orderpage.java class:

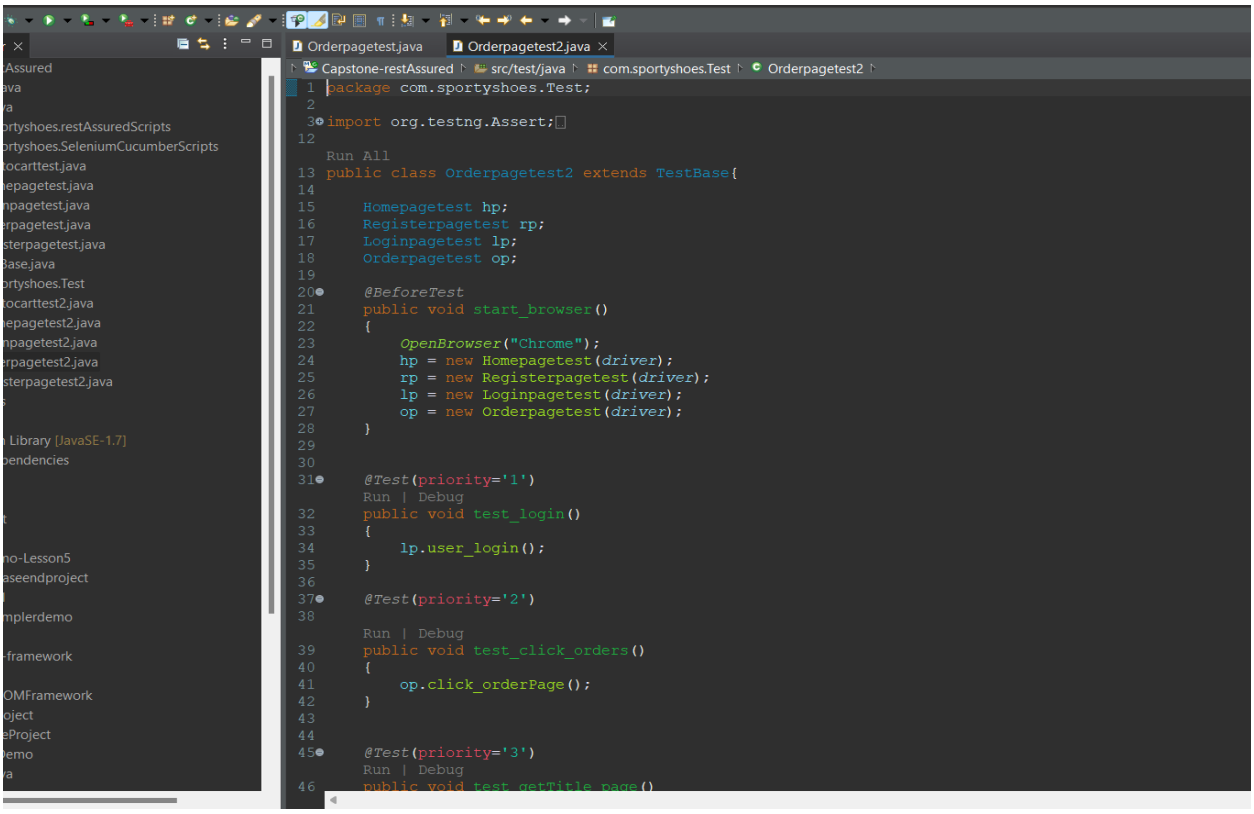
==========



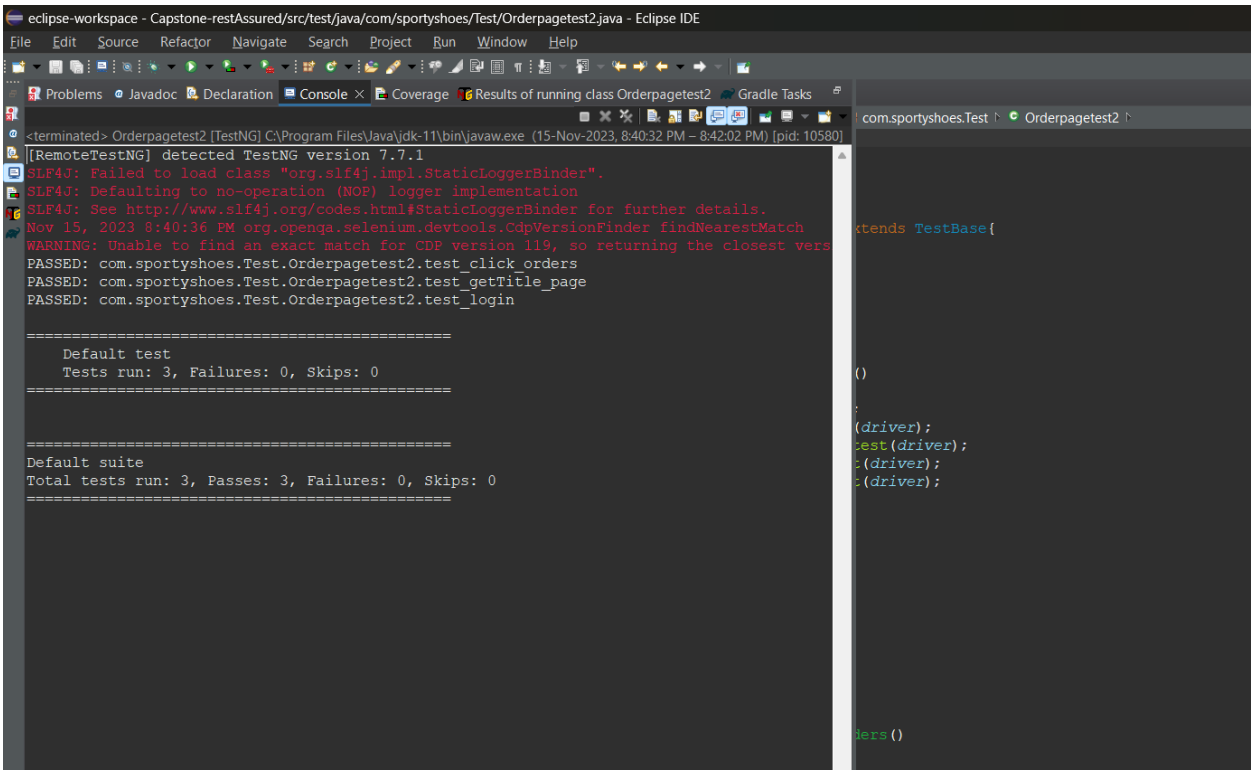
==========

Orderpagetest.java class:

==========



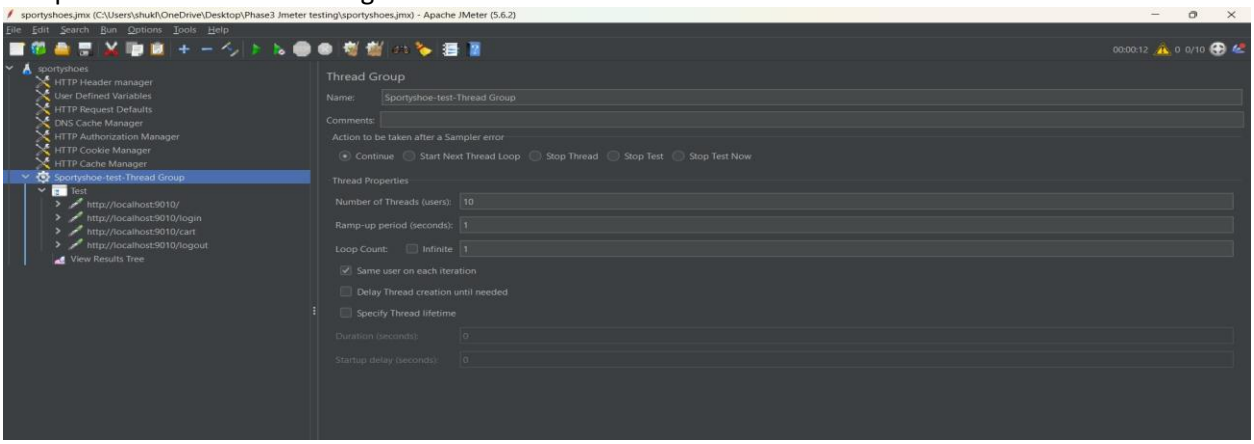
Output:

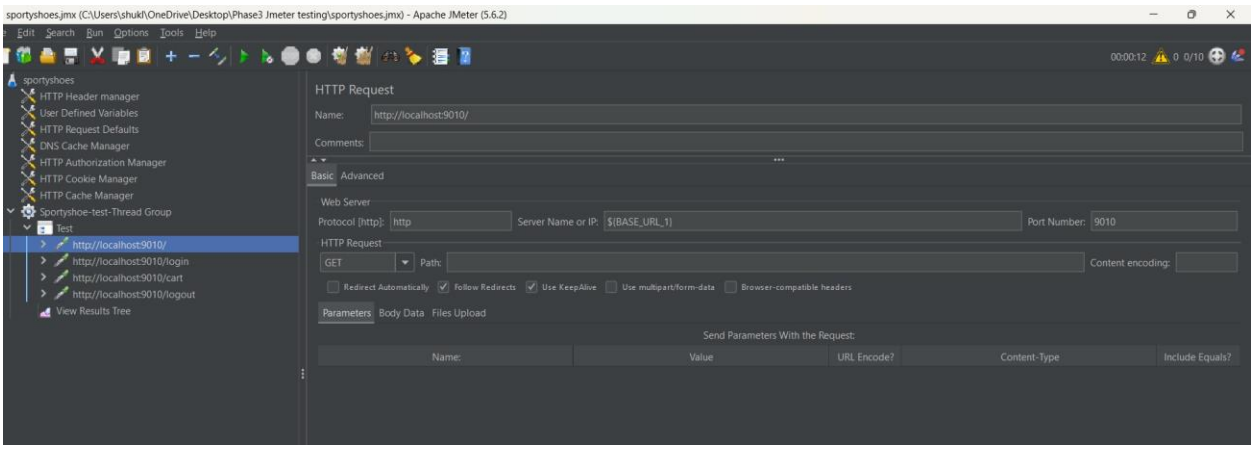


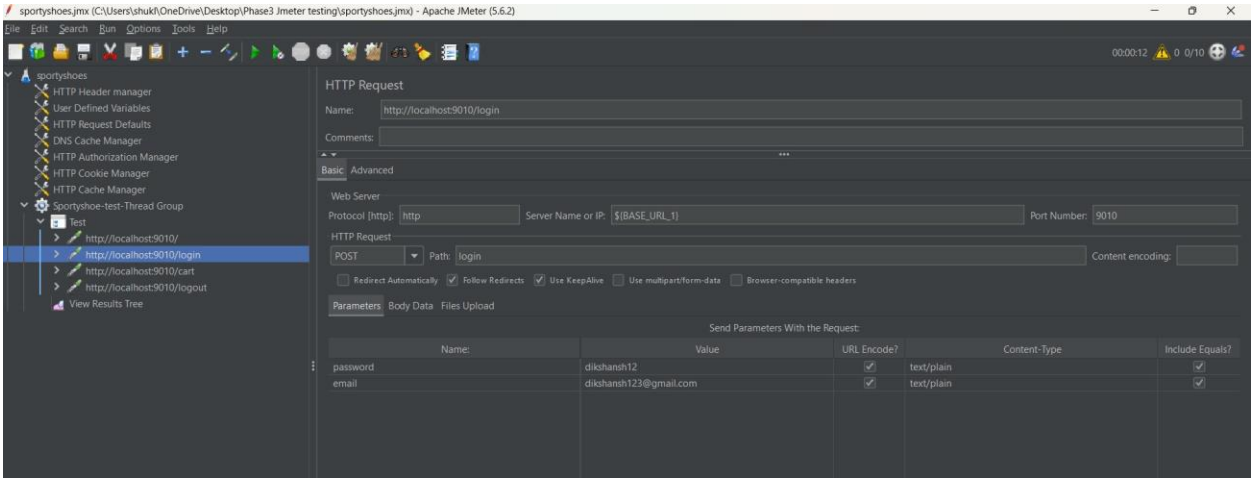
---

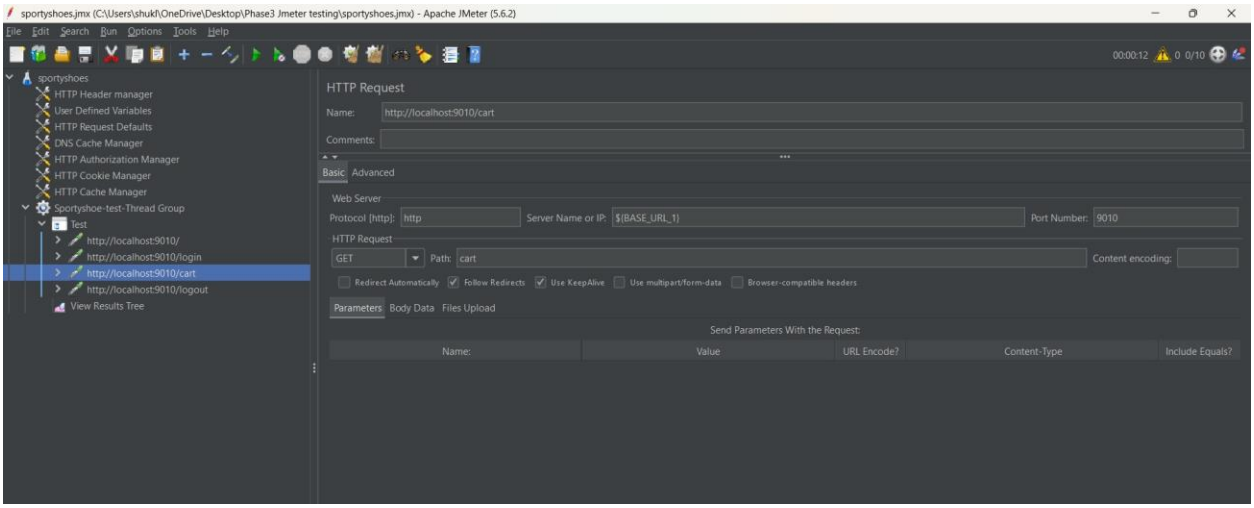
3. Create JMeter scripts to do load testing of the homepage and the product detail page.

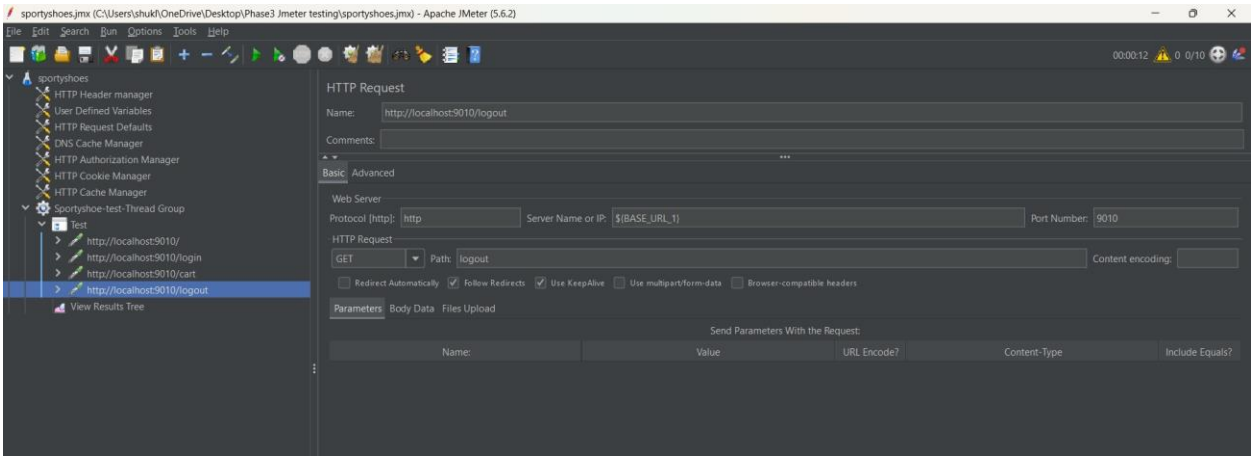
The below are the scripts which had been recorded on http://localhost:9010/ -> login -> cart -> logout and performed the load testing with threads = 10



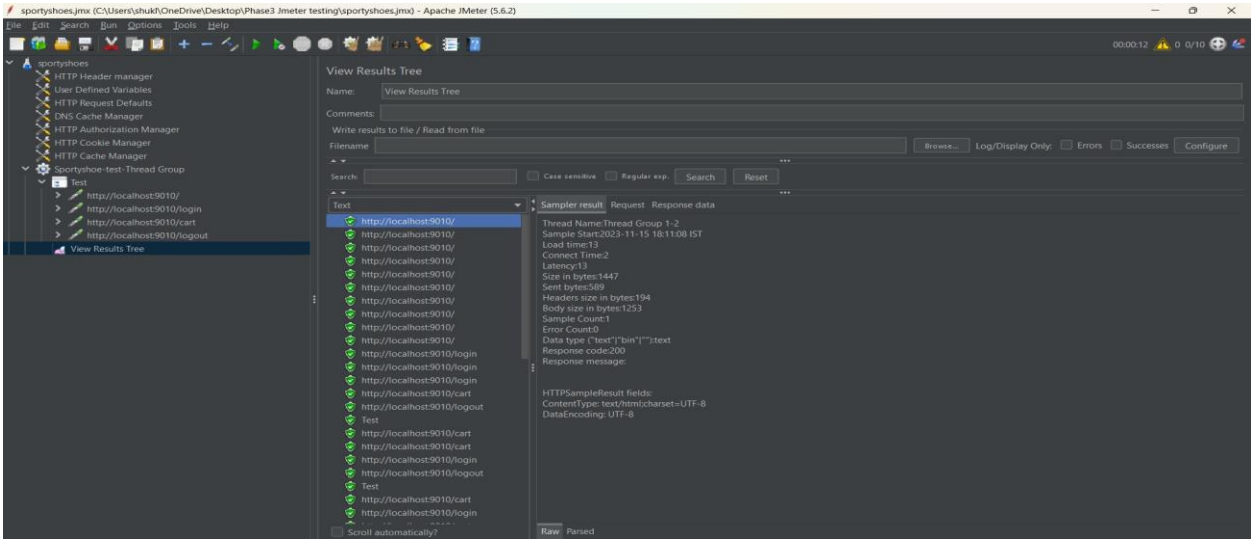

sportyshoes.jmx (C:\Users\shukl\OneDrive\Desktop\Phase3 Jmeter testing\sportyshoes.jmx) - Apache JMeter (5.6.2)
Edit Search Run Options Tools Help
00:00:12 0 0/10
sportyshoes
HTTP Header manager
User Defined Variables
HTTP Request Defaults
DNS Cache Manager
HTTP Authorization Manager
HTTP Cookie Manager
HTTP Cache Manager
Sportyshoe-test-Thread Group
Test
http://localhost:9010/
http://localhost:9010/login
http://localhost:9010/cart
http://localhost:9010/logout
View Results Tree
HTTP Request
Name: http://localhost:9010/
Comments:
Basic Advanced
Web Server
Protocol [http]: http
Server Name or IP: ${BASE_URL_1}
Port Number: 9010
HTTP Request
GET
Path:
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?


sportyshoes.jmx (C:\Users\shukl\OneDrive\Desktop\Phase3 Jmeter testing\sportyshoes.jmx) - Apache JMeter (5.6.2)
File Edit Search Run Options Tools Help
00:00:12 0 0/10
sportyshoes
HTTP Header manager
User Defined Variables
HTTP Request Defaults
DNS Cache Manager
HTTP Authorization Manager
HTTP Cookie Manager
HTTP Cache Manager
Sportyshoe-test-Thread Group
Test
http://localhost:9010/
http://localhost:9010/login
http://localhost:9010/cart
http://localhost:9010/logout
View Results Tree
HTTP Request
Name: http://localhost:9010/login
Comments:
Basic Advanced
Web Server
Protocol [http]: http
Server Name or IP: ${BASE_URL_1}
Port Number: 9010
HTTP Request
POST
Path: login
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?
password
dikshansh12
text/plain
email
dikshansh123@gmail.com
text/plain


sportyshoes.jmx (C:\Users\shukl\OneDrive\Desktop\Phase3 Jmeter testing\sportyshoes.jmx) - Apache JMeter (5.6.2)
File Edit Search Run Options Tools Help
00:00:12 0 0/10
sportyshoes
HTTP Header manager
User Defined Variables
HTTP Request Defaults
DNS Cache Manager
HTTP Authorization Manager
HTTP Cookie Manager
HTTP Cache Manager
Sportyshoe-test-Thread Group
Test
http://localhost:9010/
http://localhost:9010/login
http://localhost:9010/cart
http://localhost:9010/logout
View Results Tree
HTTP Request
Name: http://localhost:9010/cart
Comments:
Basic Advanced
Web Server
Protocol [http]: http
Server Name or IP: ${BASE_URL_1}
Port Number: 9010
HTTP Request
GET
Path: cart
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?

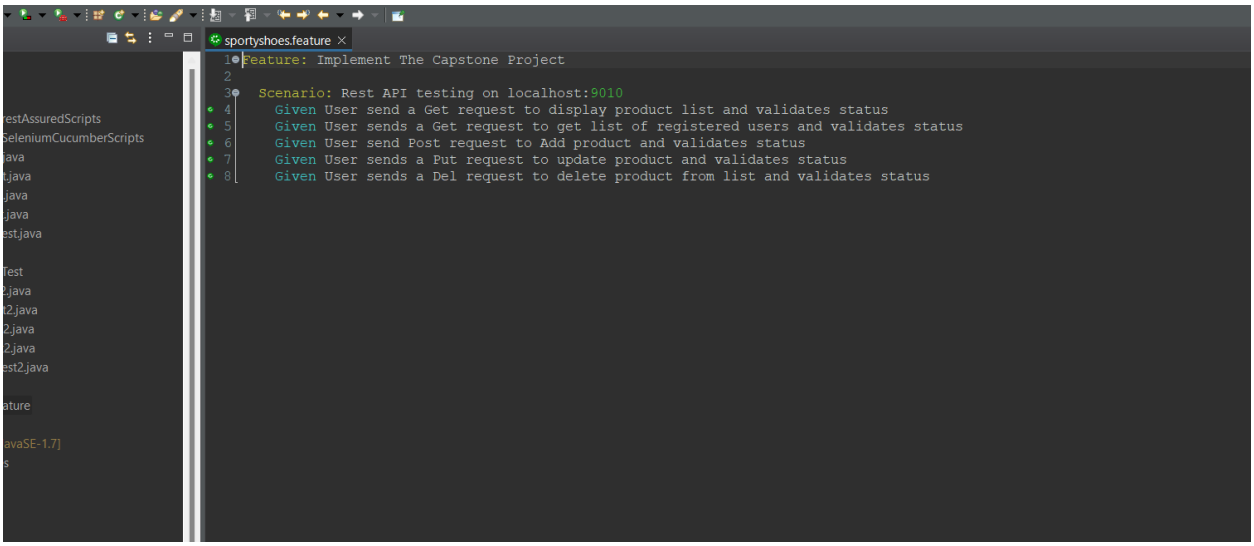


Output Response:

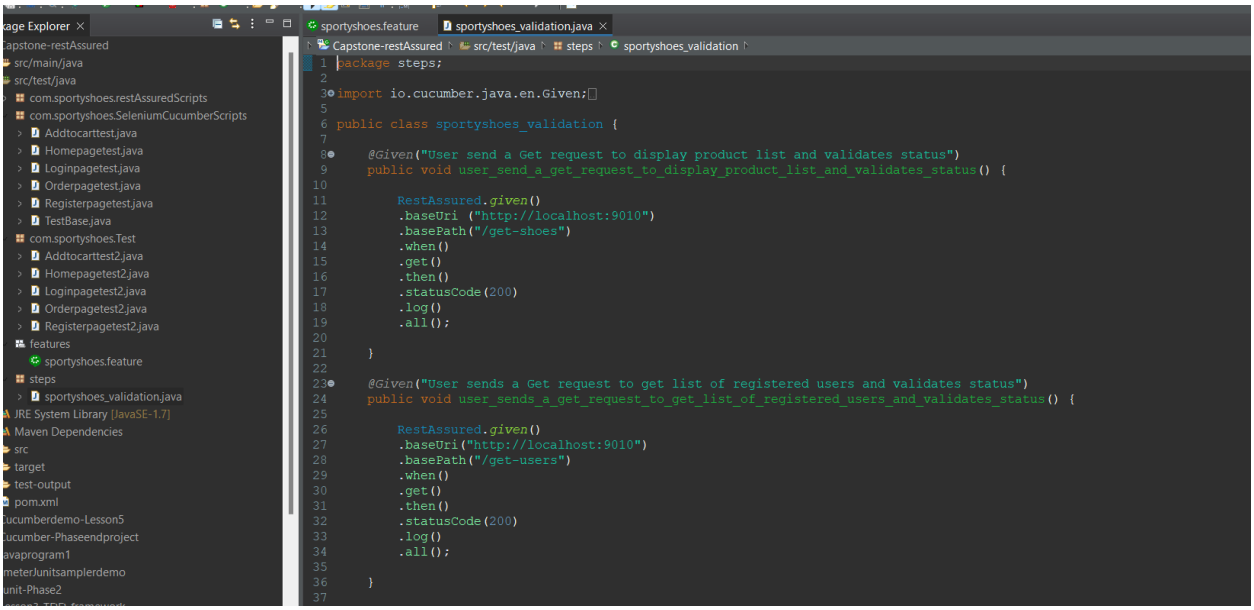


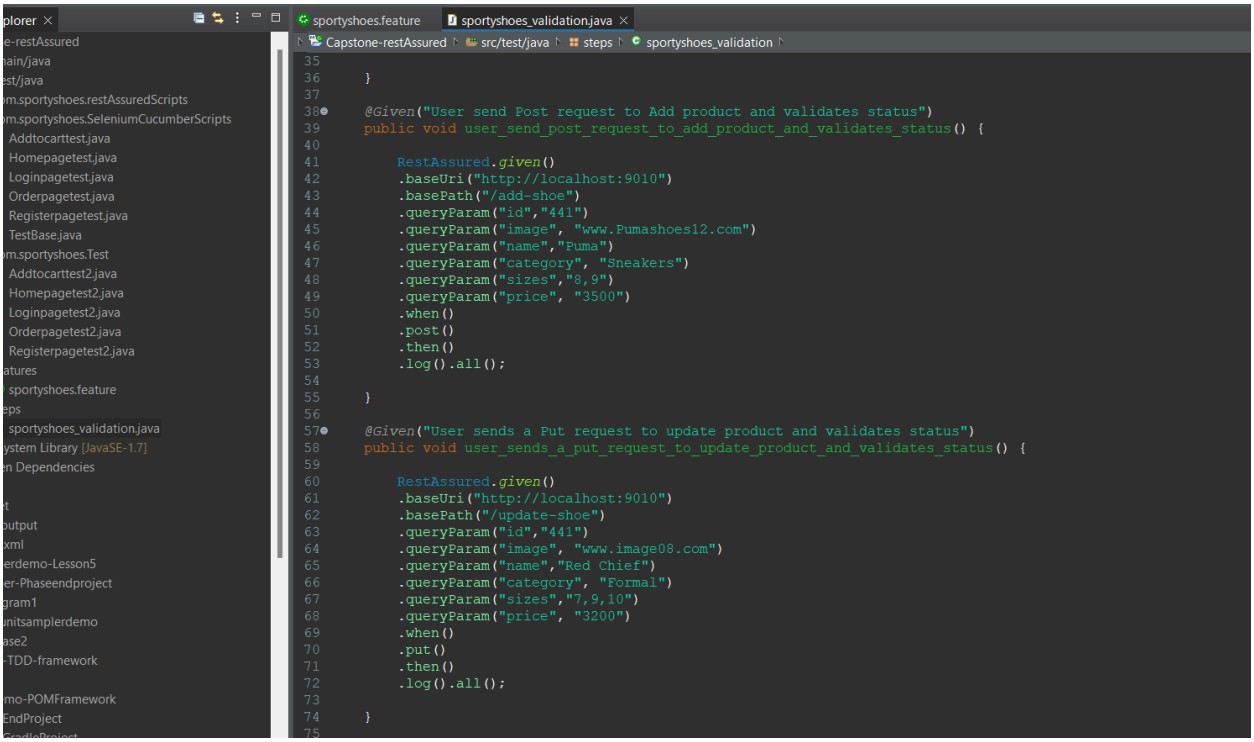
4. Setup Cucumber in Java Project and write Feature Files using Gherkin to test the API endpoints mentioned in point 1 above.

Sportyshoe.feature file:
==========



Steps Package:
(sportyshoes_validate.java)
==========



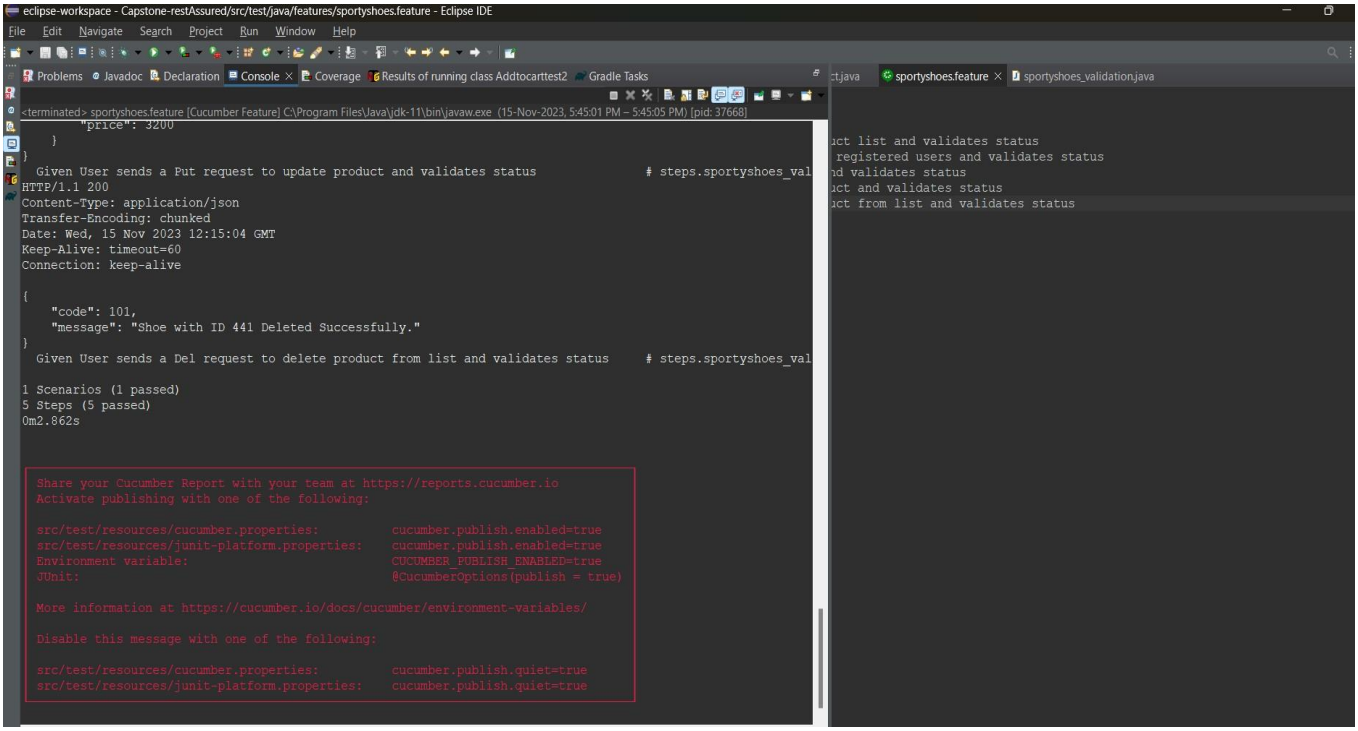



```
35
36      }
37
38      @Given("User send Post request to Add product and validates status")
39      public void user_send_post_request_to_add_product_and_validates_status() {
40
41          RestAssured.given()
42          .baseUri("http://localhost:9010")
43          .basePath("/add-shoe")
44          .queryParam("id","441")
45          .queryParam("image", "www.Pumashoes12.com")
46          .queryParam("name","Puma")
47          .queryParam("category", "Sneakers")
48          .queryParam("sizes","8,9")
49          .queryParam("price", "3500")
50          .when()
51          .post()
52          .then()
53          .log().all();
54
55      }
56
57      @Given("User sends a Put request to update product and validates status")
58      public void user_sends_a_put_request_to_update_product_and_validates_status() {
59
60          RestAssured.given()
61          .baseUri("http://localhost:9010")
62          .basePath("/update-shoe")
63          .queryParam("id","441")
64          .queryParam("image", "www.image08.com")
65          .queryParam("name","Red Chief")
66          .queryParam("category", "Formal")
67          .queryParam("sizes","7,9,10")
68          .queryParam("price", "3200")
69          .when()
70          .put()
71          .then()
72          .log().all();
73
74      }
75
```

```
73
74      }
75
76      @Given("User sends a Del request to delete product from list and validates status")
77      public void user_sends_a_del_request_to_delete_product_from_list_and_validates_status() {
78
79          RestAssured.given()
80          .baseUri("http://localhost:9010")
81          .basePath("/delete-shoe")
82          .queryParam("id", "441")
83          .when().delete()
84          .then().log().all();
85
86      }
87
88  }
89
```

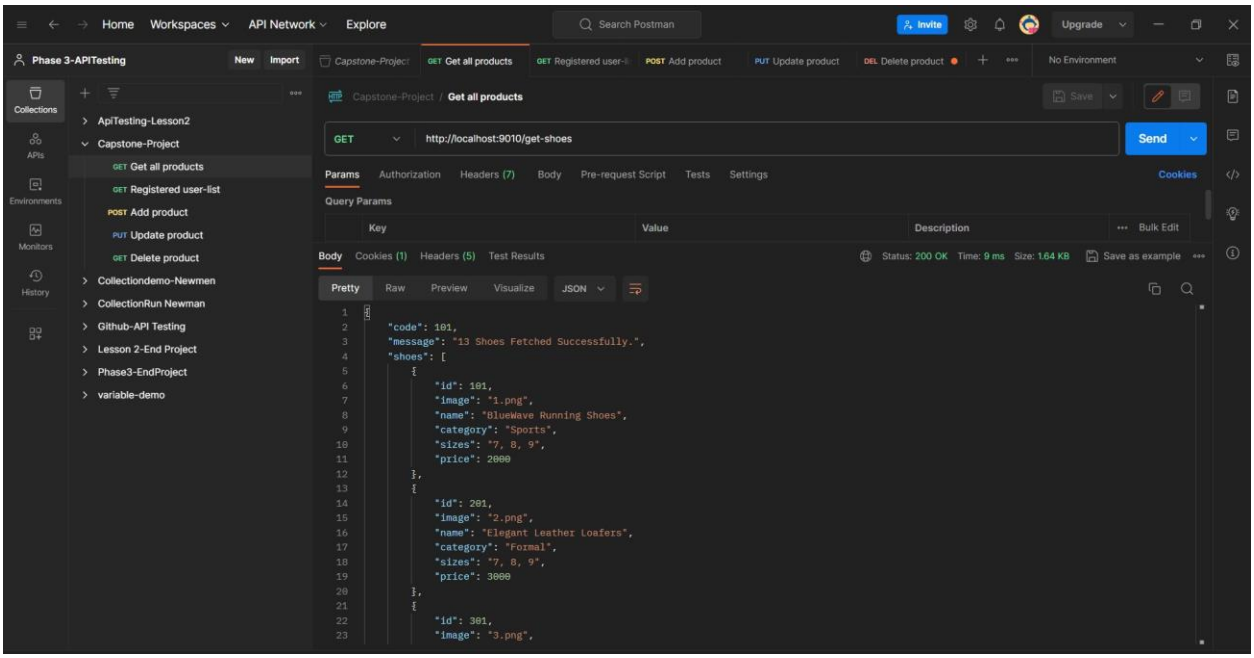
==========

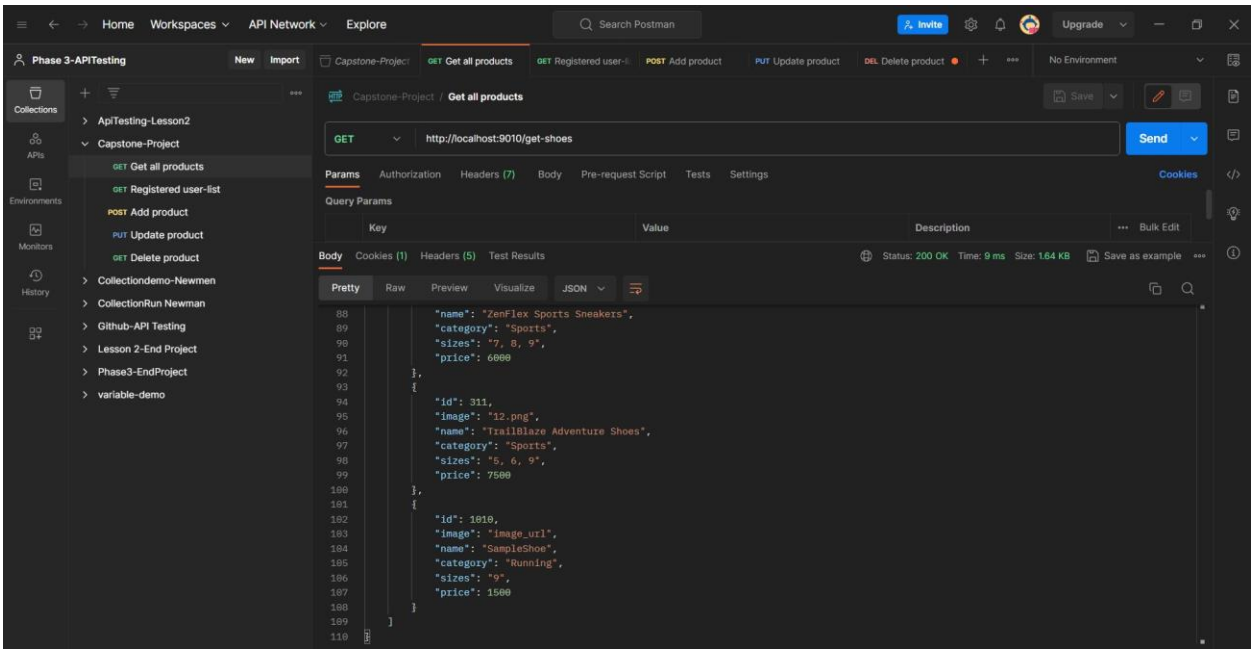
Cucumber Output response:



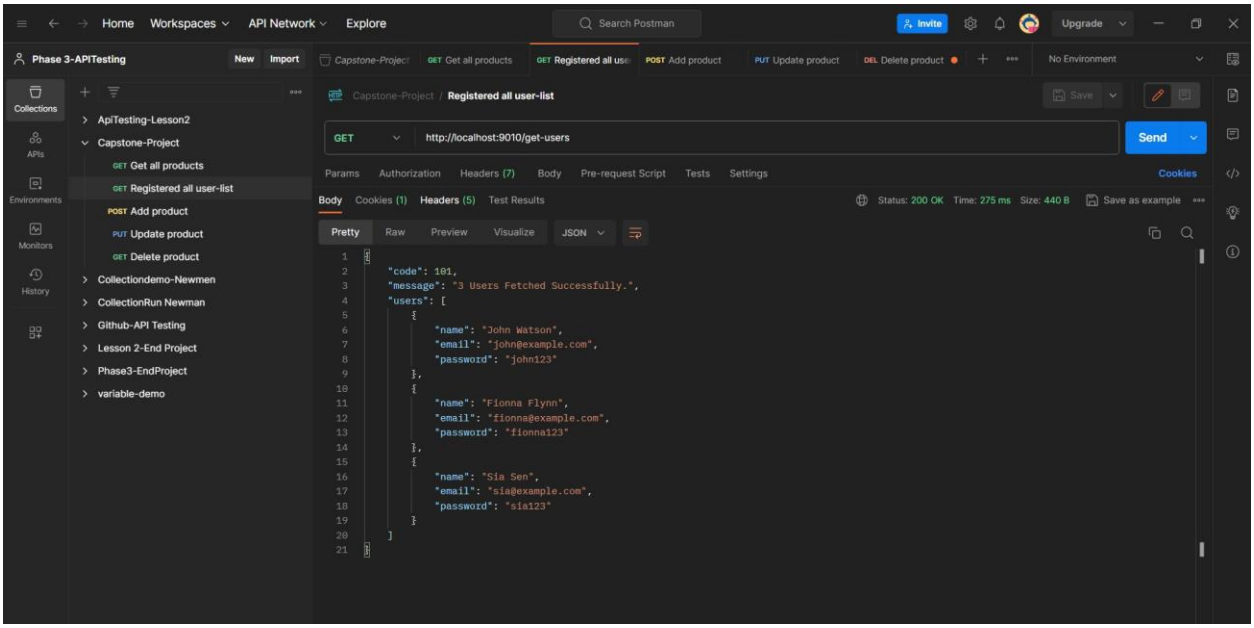
5. Create Postman scripts to test the following API endpoints:

Display all products in the store:

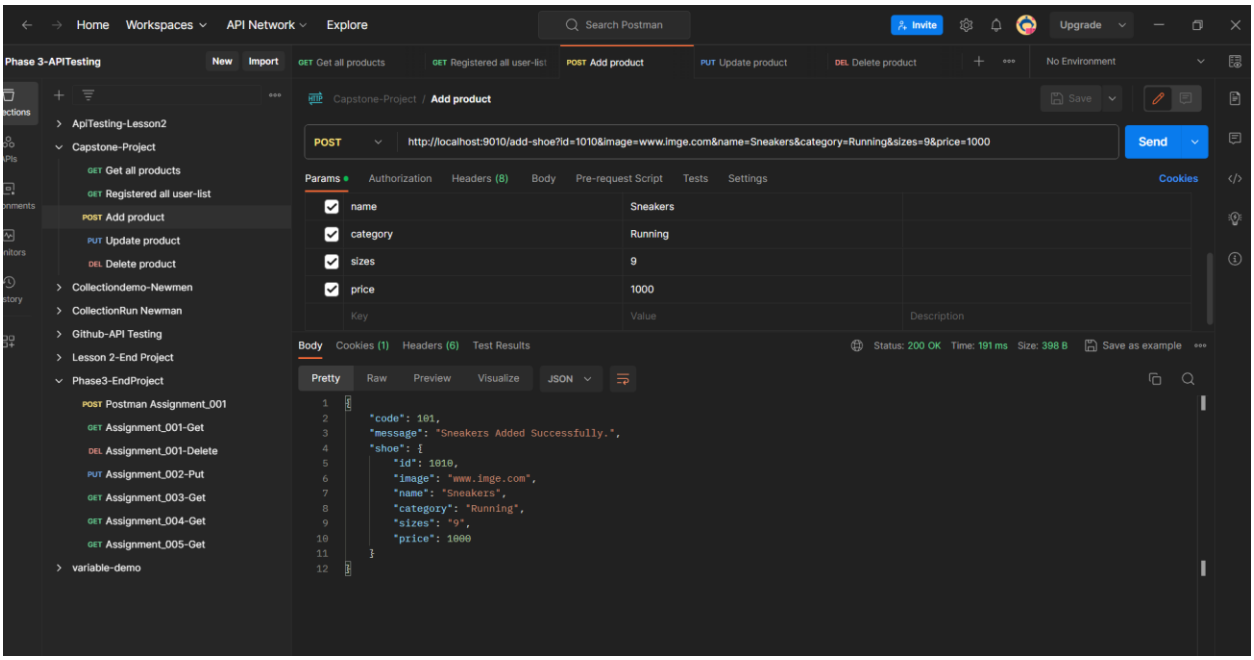




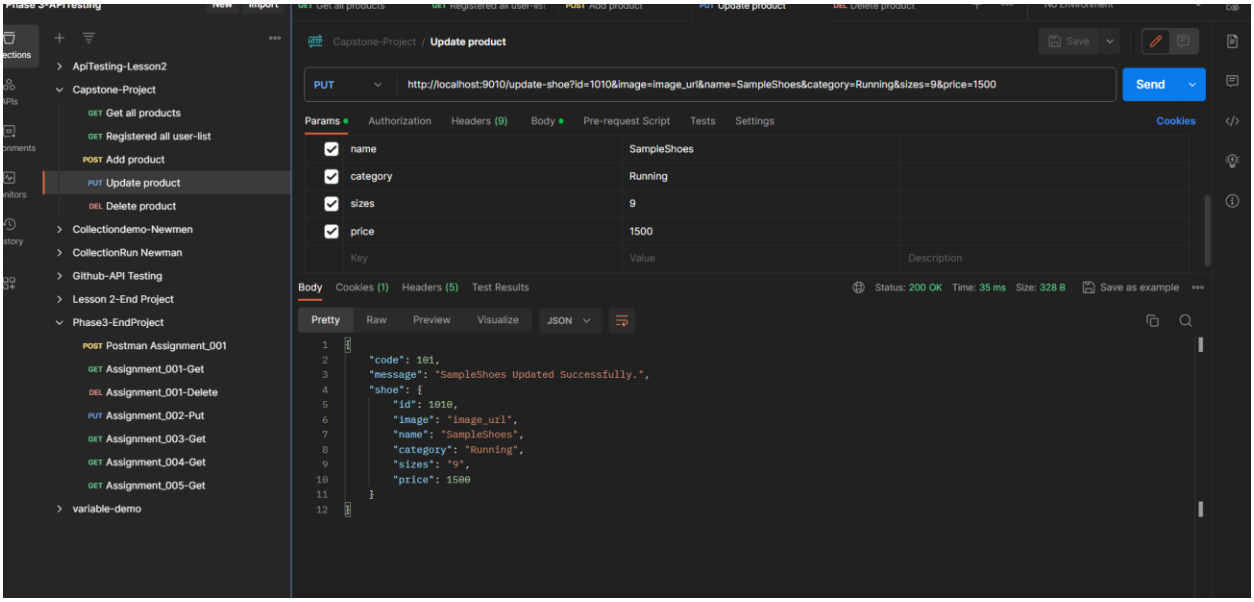
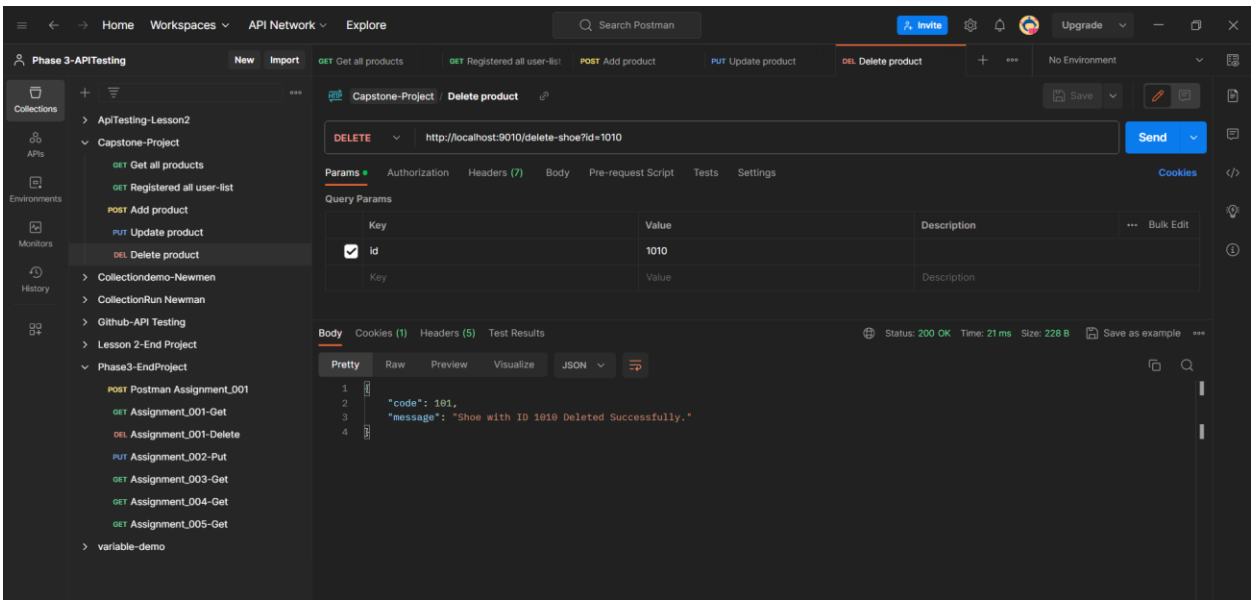
Display list of all registered users:



Add the Products:



Update Products:



Delete Product