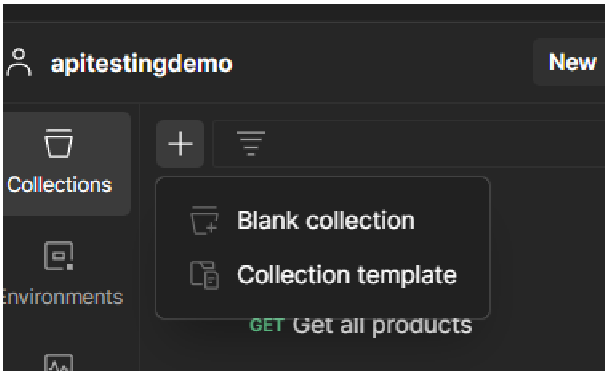
Source code for capstone Project:

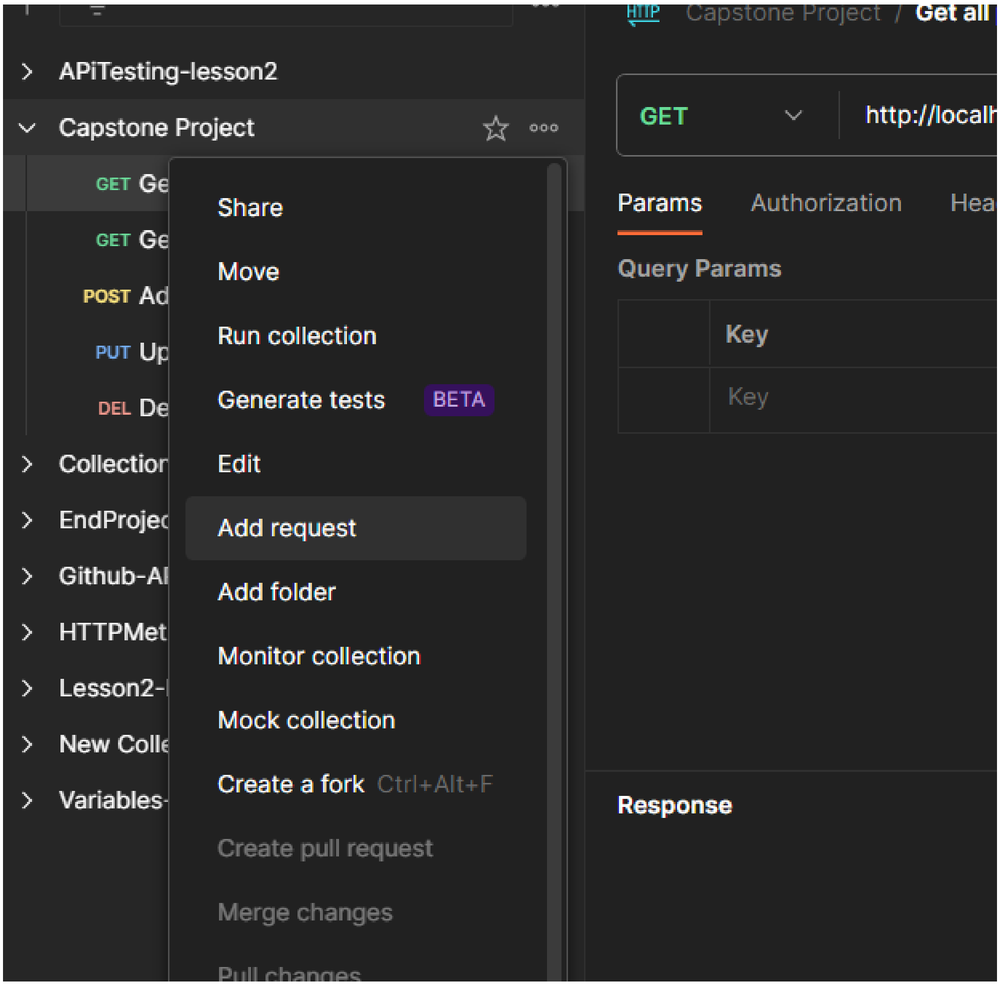
Creating a Testing Framework for sporty shoes website:

1. 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.

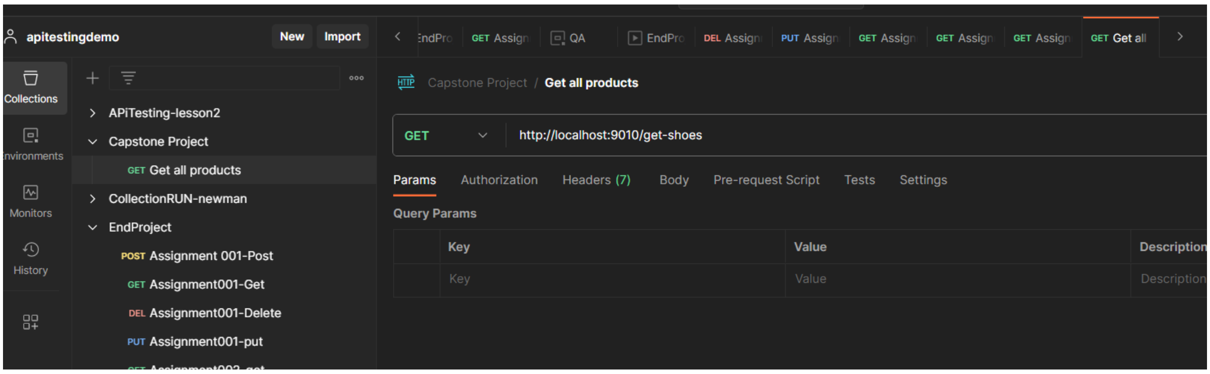
First go to postman application



Click on + symbol here show 2 options select blank collection and put name as capston e project.



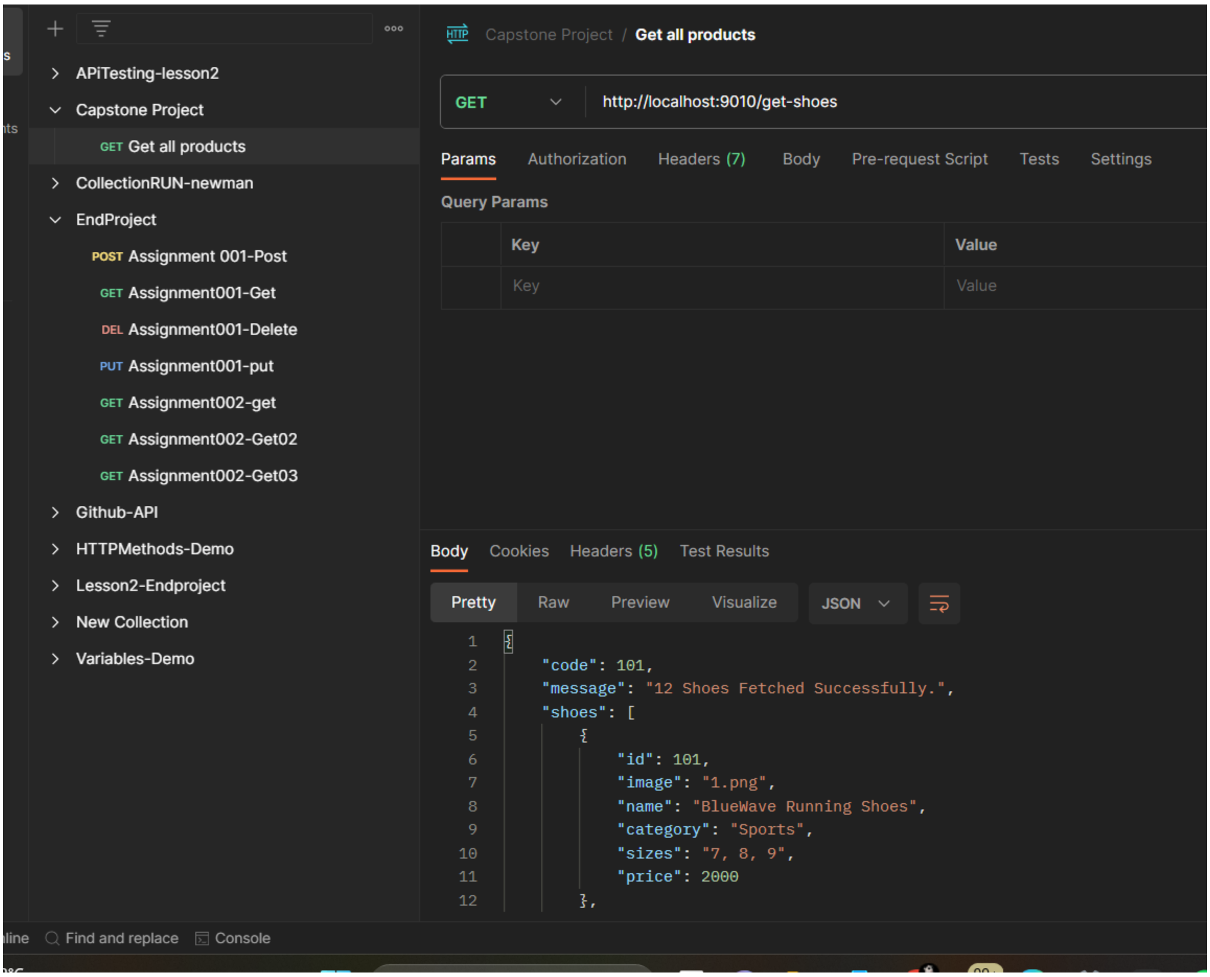
Next step capstone project have 3 dots click on 3 dots have option Add request Put name as Get all Products.



GET http://localhost:9010/get-shoes

Enter URL in GET method http://localhost:9010/get-shoes

After adding URL click on save button and run this URL



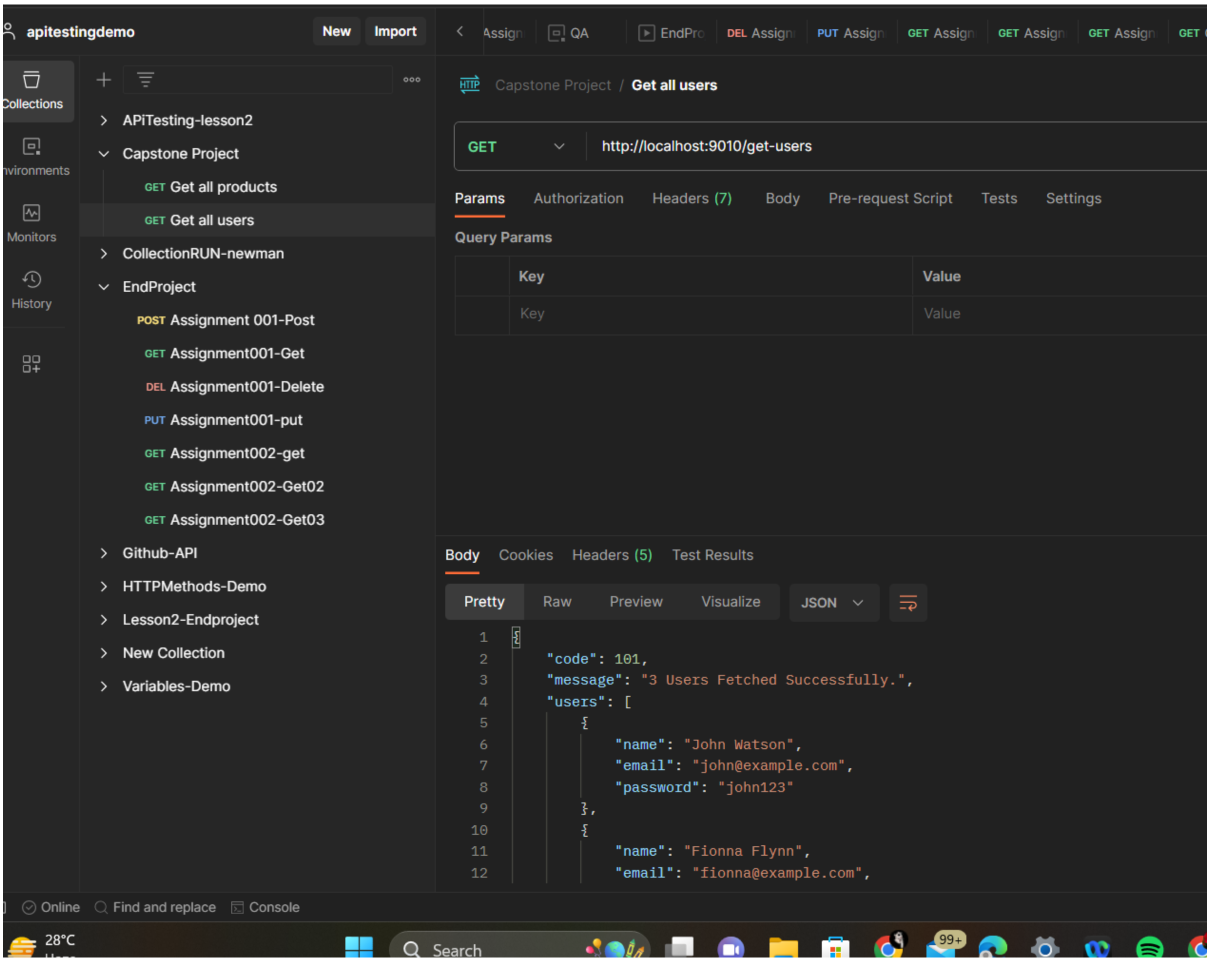
Capstone Project / Get all products

GET http://localhost:9010/get-shoes

```
{
    "code": 101,
    "message": "12 Shoes Fetched Successfully.",
    "shoes": [
        {
            "id": 101,
            "image": "1.png",
            "name": "BlueWave Running Shoes",
            "category": "Sports",
            "sizes": "7, 8, 9",
            "price": 2000
        },
```

This above picture is the GET all products output

Next select another request put name as GET all users

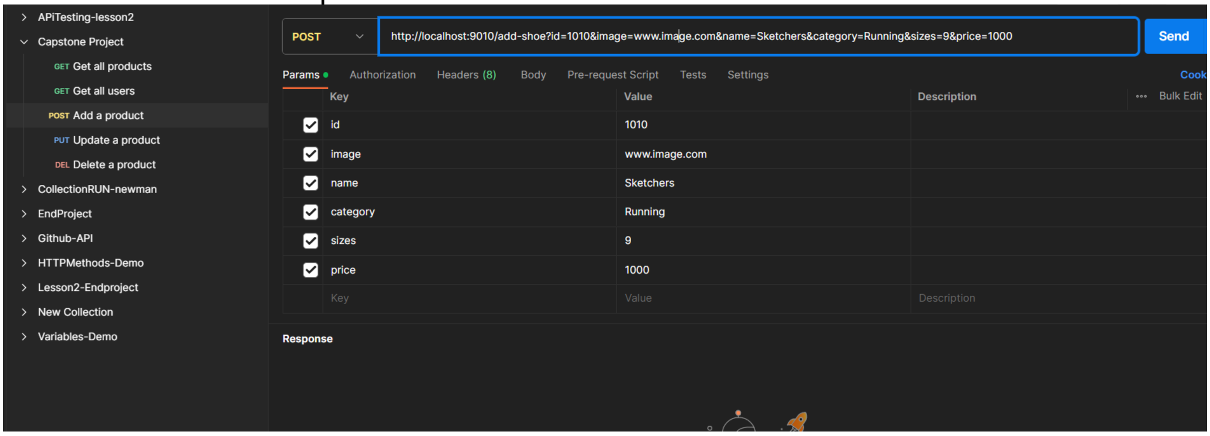


GET http://localhost:9010/get-users

```
{
    "code": 101,
    "message": "3 Users Fetched Successfully.",
    "users": [
        {
            "name": "John Watson",
            "email": "john@example.com",
            "password": "john123"
        },
        {
            "name": "Fionna Flynn",
            "email": "fionna@example.com",
```

In GET all users enter URL http://localhost:9010/get-users
And click on save and run this URL above picture is get all users output

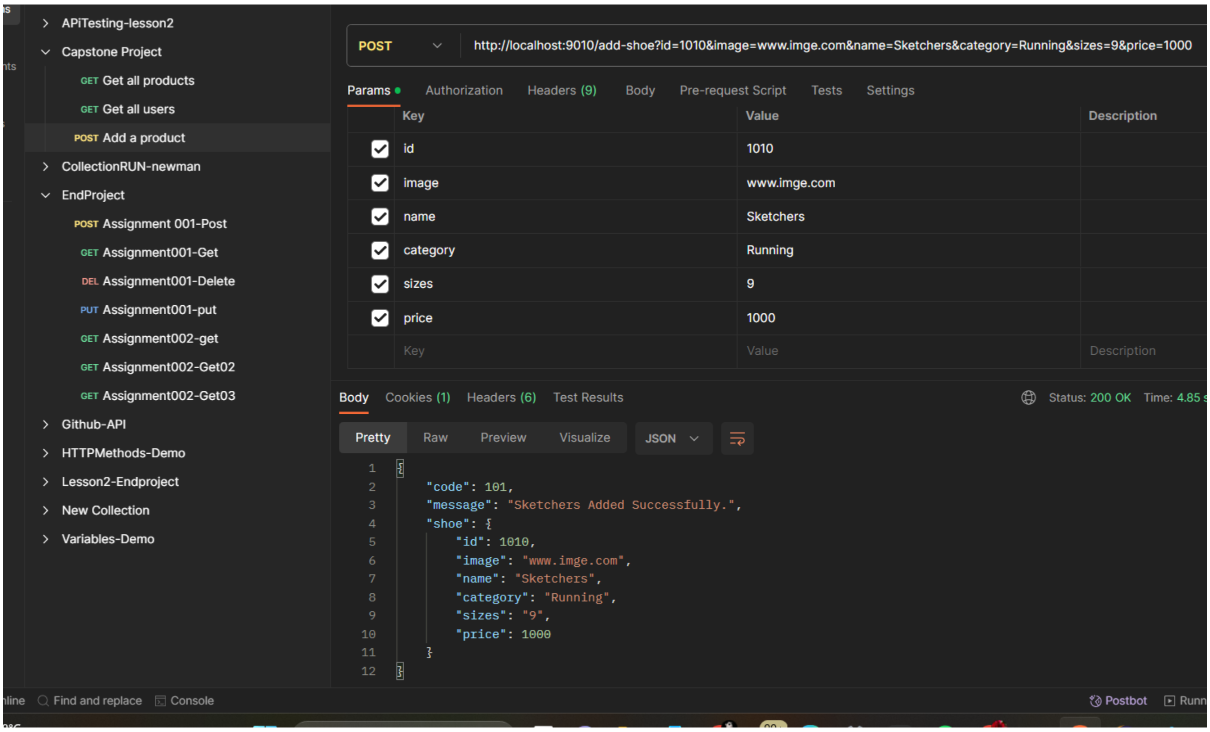
Next add another Request and enter name as Post add a Product



POST http://localhost:9010/add-shoe?id=1010&image=www.image.com&name=Sketchers&category=Running&sizes=9&price=1000

	Key	Value	Description
☑	id	1010	
☑	image	www.image.com	
☑	name	Sketchers	
☑	category	Running	
☑	sizes	9	
☑	price	1000	
	Key	Value	Description

Enter URL select method as POST method click on save and send the request



POST http://localhost:9010/add-shoe?id=1010&image=www.imge.com&name=Sketchers&category=Running&sizes=9&price=1000

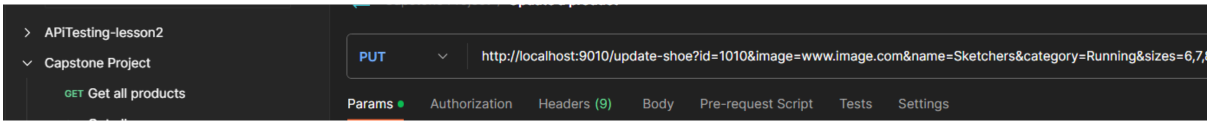
	Key	Value	Description
☑	id	1010	
☑	image	www.imge.com	
☑	name	Sketchers	
☑	category	Running	
☑	sizes	9	
☑	price	1000	

Status: 200 OK Time: 4.85

```
{
    "code": 101,
    "message": "Sketchers Added Successfully.",
    "shoe": {
        "id": 1010,
        "image": "www.imge.com",
        "name": "Sketchers",
        "category": "Running",
        "sizes": "9",
        "price": 1000
    }
}
```

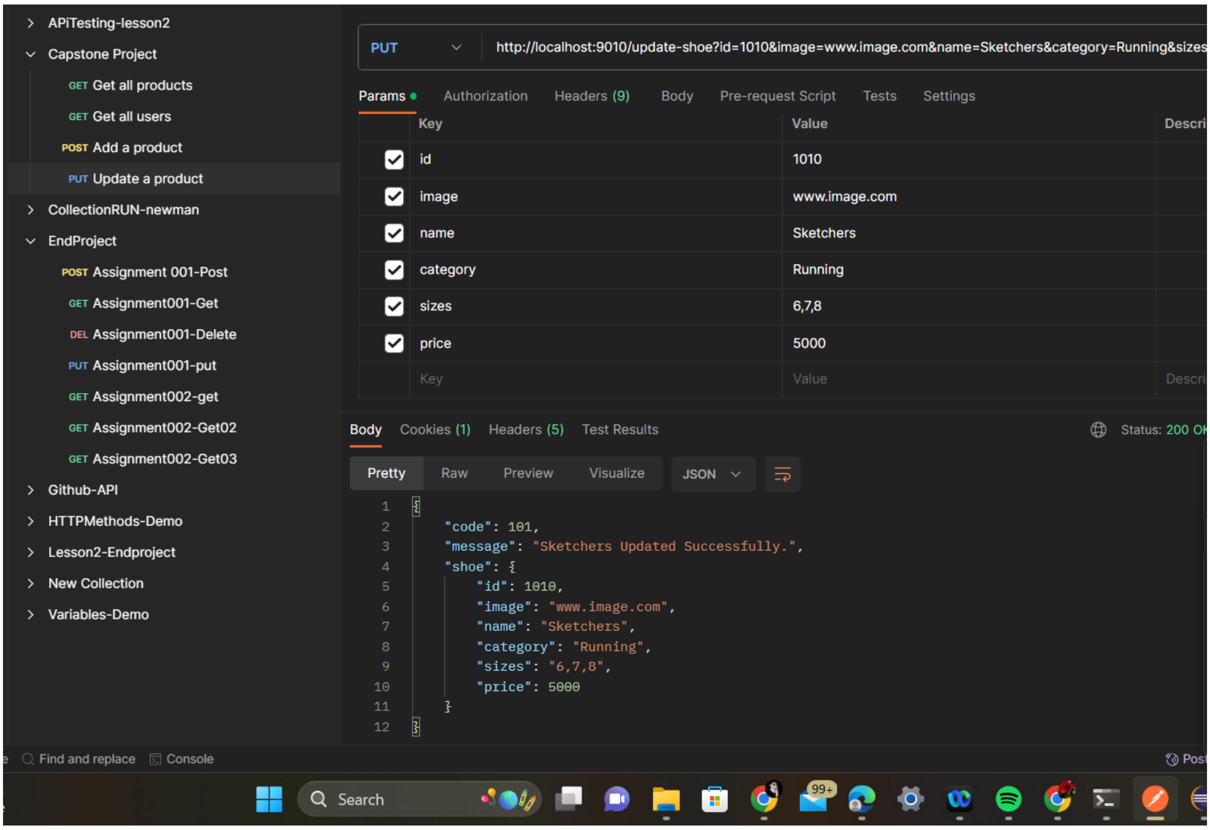
this is the output for POST add a Product.

And then select another method as Put update a Product



PUT http://localhost:9010/update-shoe?id=1010&image=www.image.com&name=Sketchers&category=Running&sizes=6,7,

And insert the URL click on save and send Request



PUT http://localhost:9010/update-shoe?id=1010&image=www.image.com&name=Sketchers&category=Running&sizes

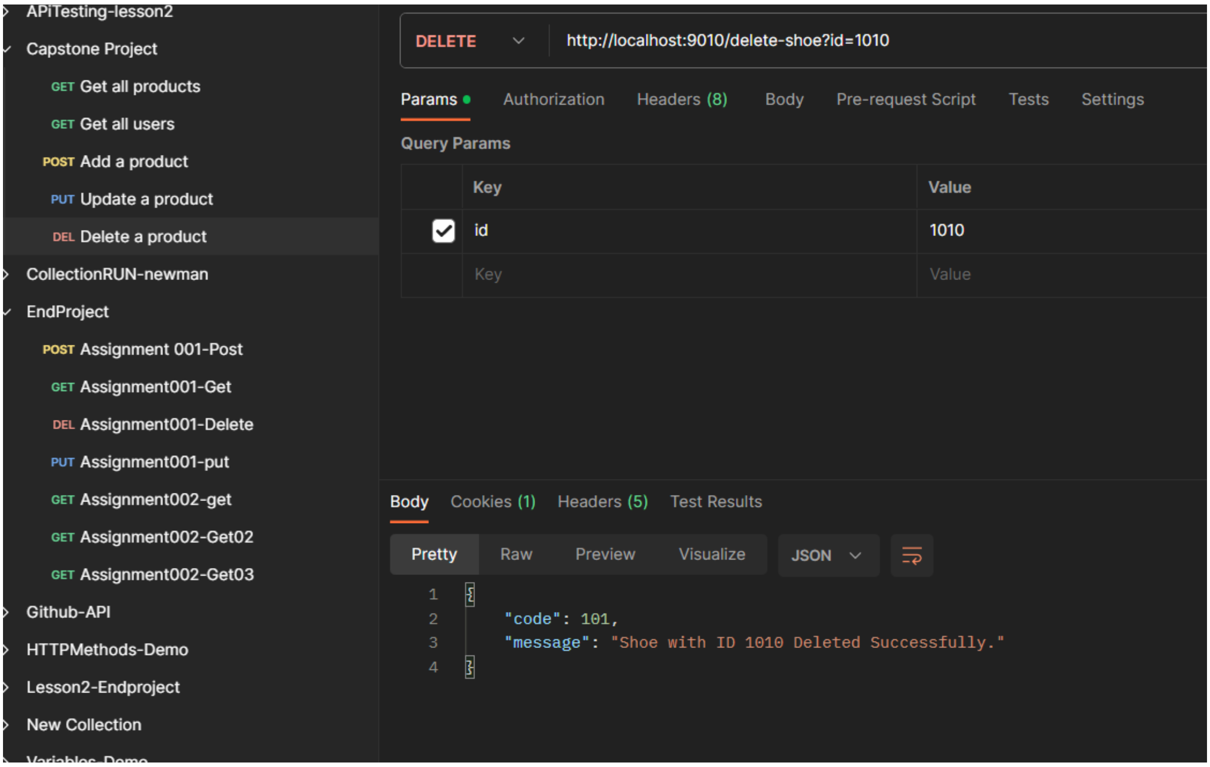
	Key	Value	Descri
☑	id	1010	
☑	image	www.image.com	
☑	name	Sketchers	
☑	category	Running	
☑	sizes	6,7,8	
☑	price	5000	
	Key	Value	Descri

Status: 200 OK

```
{
    "code": 101,
    "message": "Sketchers Updated Successfully.",
    "shoe": {
        "id": 1010,
        "image": "www.image.com",
        "name": "Sketchers",
        "category": "Running",
        "sizes": "6,7,8",
        "price": 5000
    }
}
```

This is the output for PUT Update a product

Next select another request Delete a Product



DELETE http://localhost:9010/delete-shoe?id=1010

Query Params

	Key	Value
☑	id	1010
	Key	Value

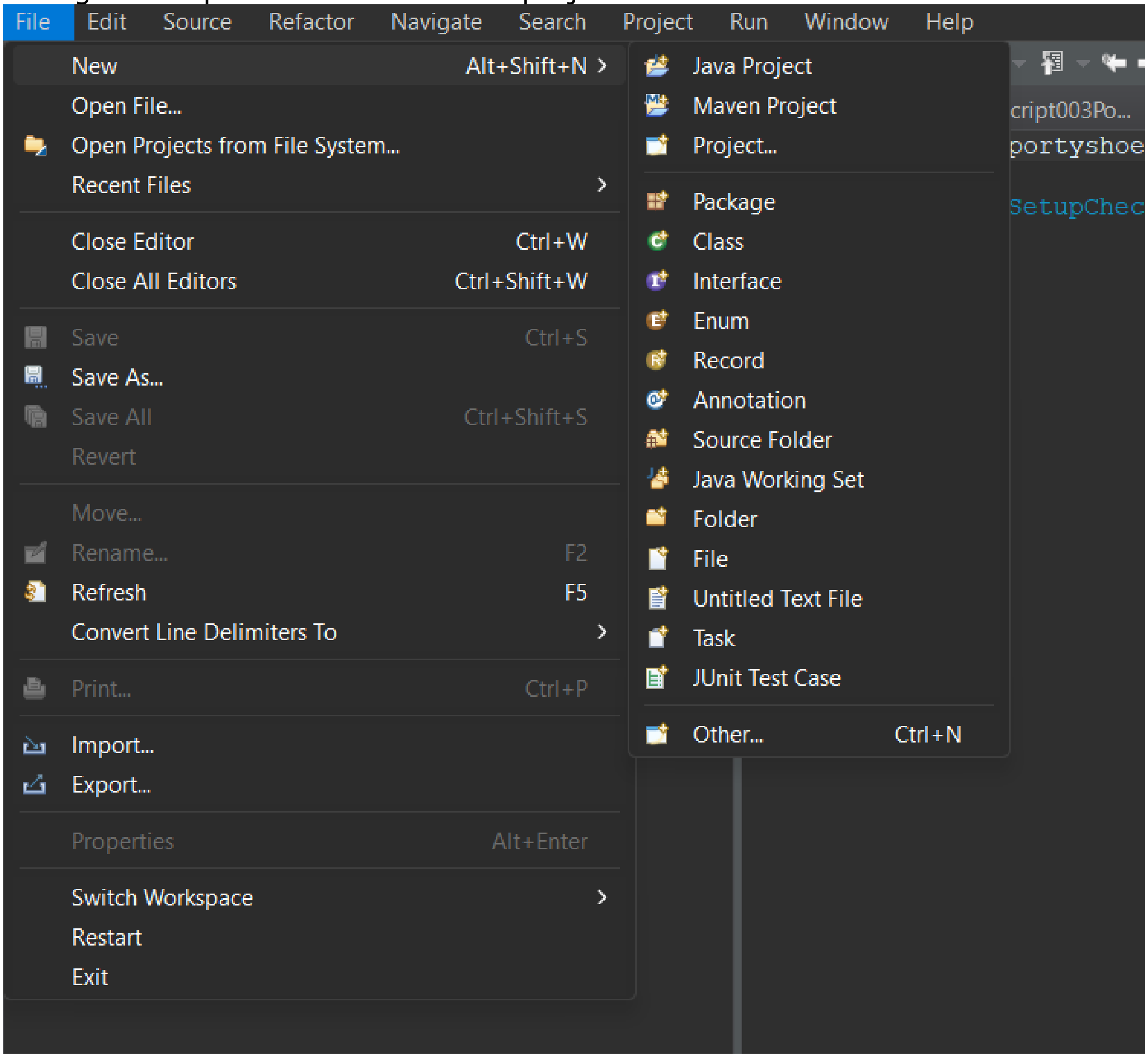
```
{
    "code": 101,
    "message": "Shoe with ID 1010 Deleted Successfully."
}
```

Enter link as http://localhost:9010/delete-shoe?id=1010

Click on save and run it

1. 2.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.

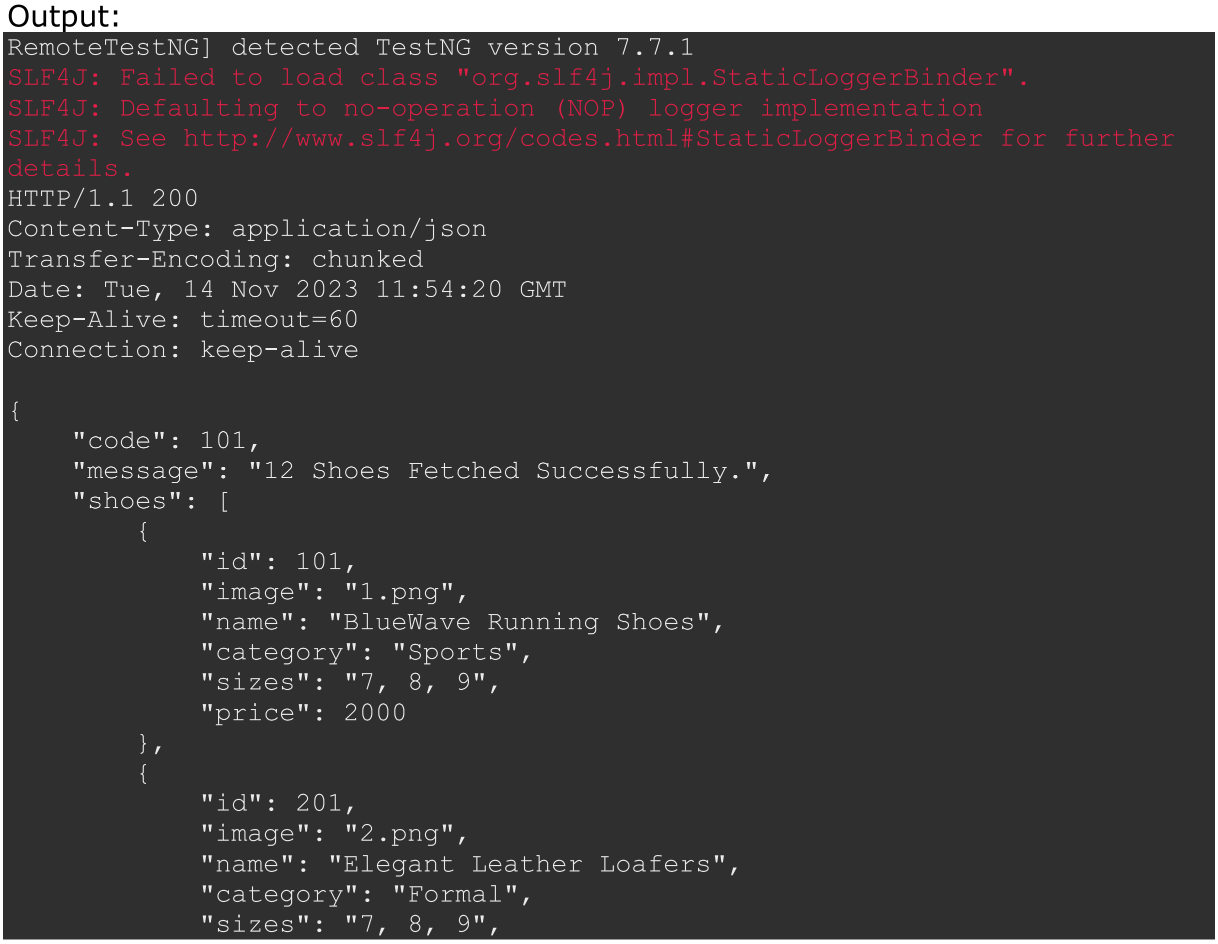
First go to Eclipse file >new>maven project



And enter name as capstone project
In capstone project add a package com.spotryshoe.restAssuredscripts
Put class name as getallShoes.java
Code:

```
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();
        }

}
```

Output:



```
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.
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 14 Nov 2023 11:54:20 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
    "code": 101,
    "message": "12 Shoes Fetched Successfully.",
    "shoes": [
        {
            "id": 101,
            "image": "1.png",
            "name": "BlueWave Running Shoes",
            "category": "Sports",
            "sizes": "7, 8, 9",
            "price": 2000
        },
        {
            "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",
            "price": 4000
        },
        {
            "id": 601,
            "image": "6.png",
            "name": "VentureHike Trail Shoes",
            "category": "Sports",
            "sizes": "5, 8, 9",
            "price": 2500
        },
        {
            "id": 701,
            "image": "7.png",
            "name": "EnduraGrip Sports Sneakers",
            "category": "Sports",
            "sizes": "4, 6, 9",
            "price": 5000
        },
        {
            "id": 801,
            "image": "8.png",
            "name": "LightStride Performance Shoes",
            "category": "Sports",
            "sizes": "7, 8",
            "price": 1200
        },
        {
            "id": 901,
            "image": "9.png",
            "name": "MaxFit Pro Sports Shoes",
            "category": "Sports",
            "sizes": "7, 8, 9, 10",
            "price": 4700
        },
        {
            "id": 111,
            "image": "10.png",
            "name": "RapidFlex Running Shoes",
            "category": "Sports",
            "sizes": "4, 5, 6",
            "price": 2500
        },
        {
            "id": 211,
            "image": "11.png",
            "name": "ZenFlex Sports Sneakers",
            "category": "Sports",
            "sizes": "7, 8, 9",
            "price": 6000
        },
        {
            "id": 311,
            "image": "12.png",
            "name": "TrailBlaze Adventure Shoes",
            "category": "Sports",
            "sizes": "5, 6, 9",
            "price": 7500
        }
    ]
}
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 14 Nov 2023 11:54:20 GMT
Keep-Alive: timeout=60
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.sportyshoe.restAssuredScripts.GetAllshoes.get_all_users
PASSED: com.sportyshoe.restAssuredScripts.GetAllshoes.get_all_shoes

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



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

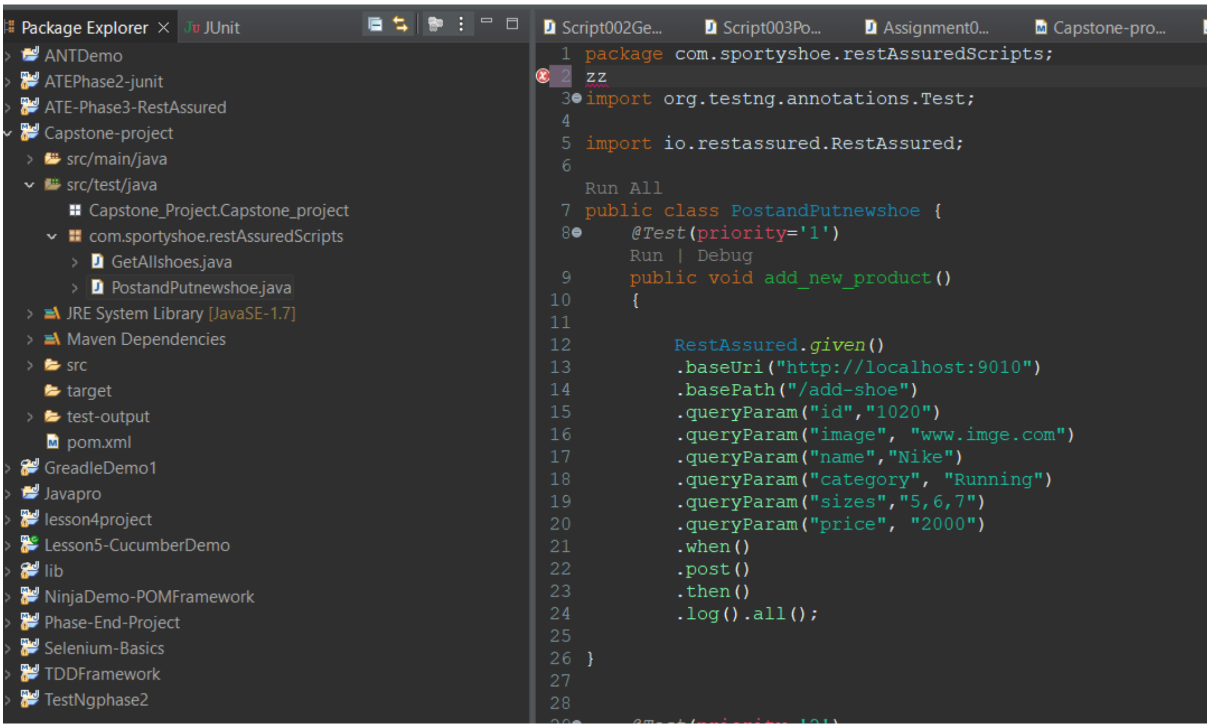
And Create another class as postandputnew shoe

Code:

```
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();


}
```

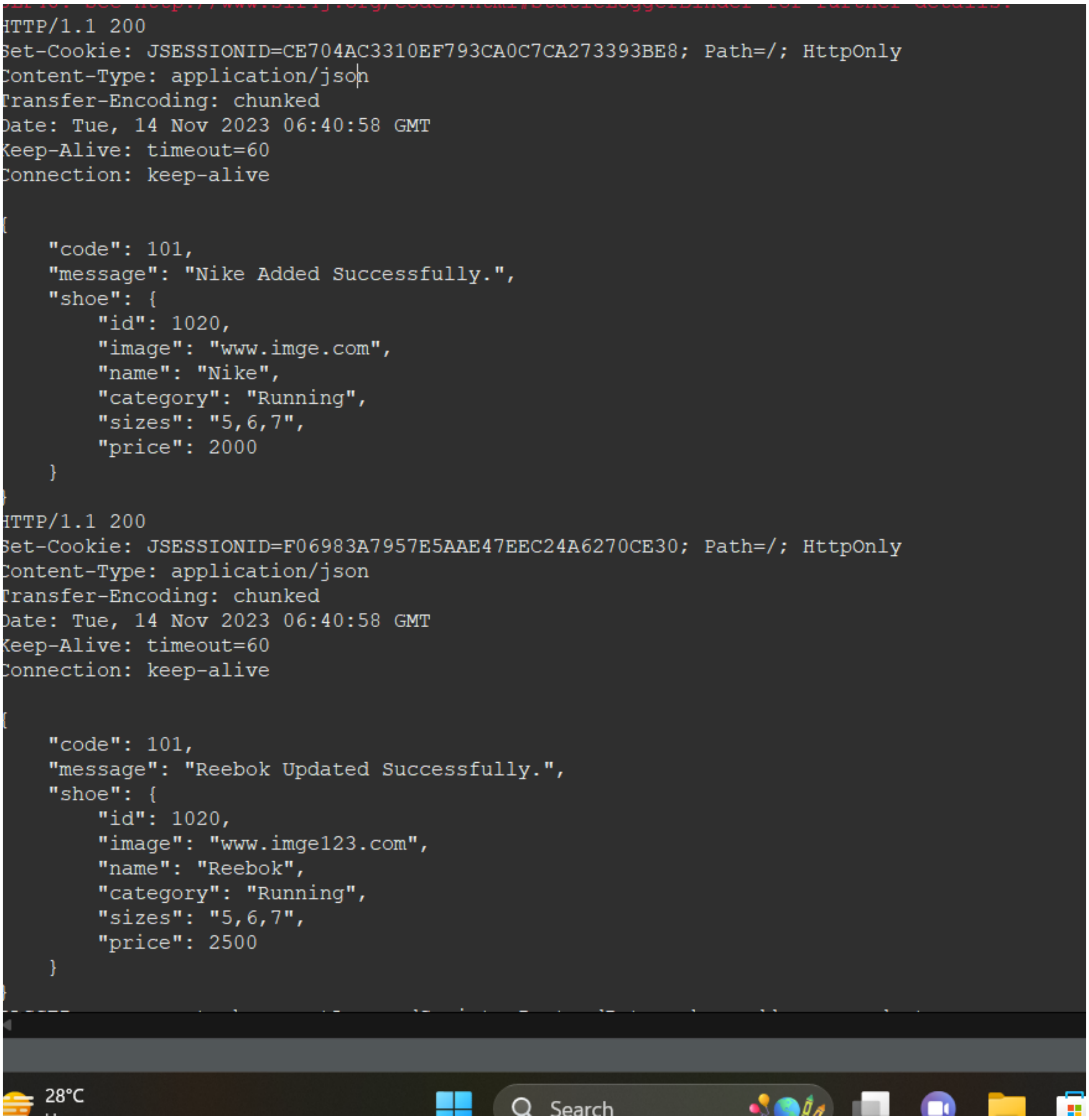
```
        @Test(priority='2')

        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();


    }


    }
```

Output:



```
package com.sportyshoe.restAssuredScripts;
zz
import org.testng.annotations.Test;

import io.restassured.RestAssured;

Run All
public class PostandPutnewshoe {
    @Test(priority='1')
    Run | Debug
    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();

}


```

Output:



```
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
HTTP/1.1 200
Set-Cookie: JSESSIONID=CE704AC3310EF793CA0C7CA273393BE8; Path=/; HttpOnly
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 14 Nov 2023 06:40:58 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
    "code": 101,
    "message": "Nike Added Successfully.",
    "shoe": {
        "id": 1020,
        "image": "www.imge.com",
        "name": "Nike",
        "category": "Running",
        "sizes": "5,6,7",
        "price": 2000
    }
}
HTTP/1.1 200
Set-Cookie: JSESSIONID=F06983A7957E5AAE47EEC24A6270CE30; Path=/; HttpOnly
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 14 Nov 2023 06:40:58 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
    "code": 101,
    "message": "Reebok Updated Successfully.",
    "shoe": {
        "id": 1020,
        "image": "www.imge123.com",
        "name": "Reebok",
        "category": "Running",
        "sizes": "5,6,7",
        "price": 2500
    }
}
```

Next add delete Rest assured code

Code:

package com.sportyshoe.restAssuredScripts;

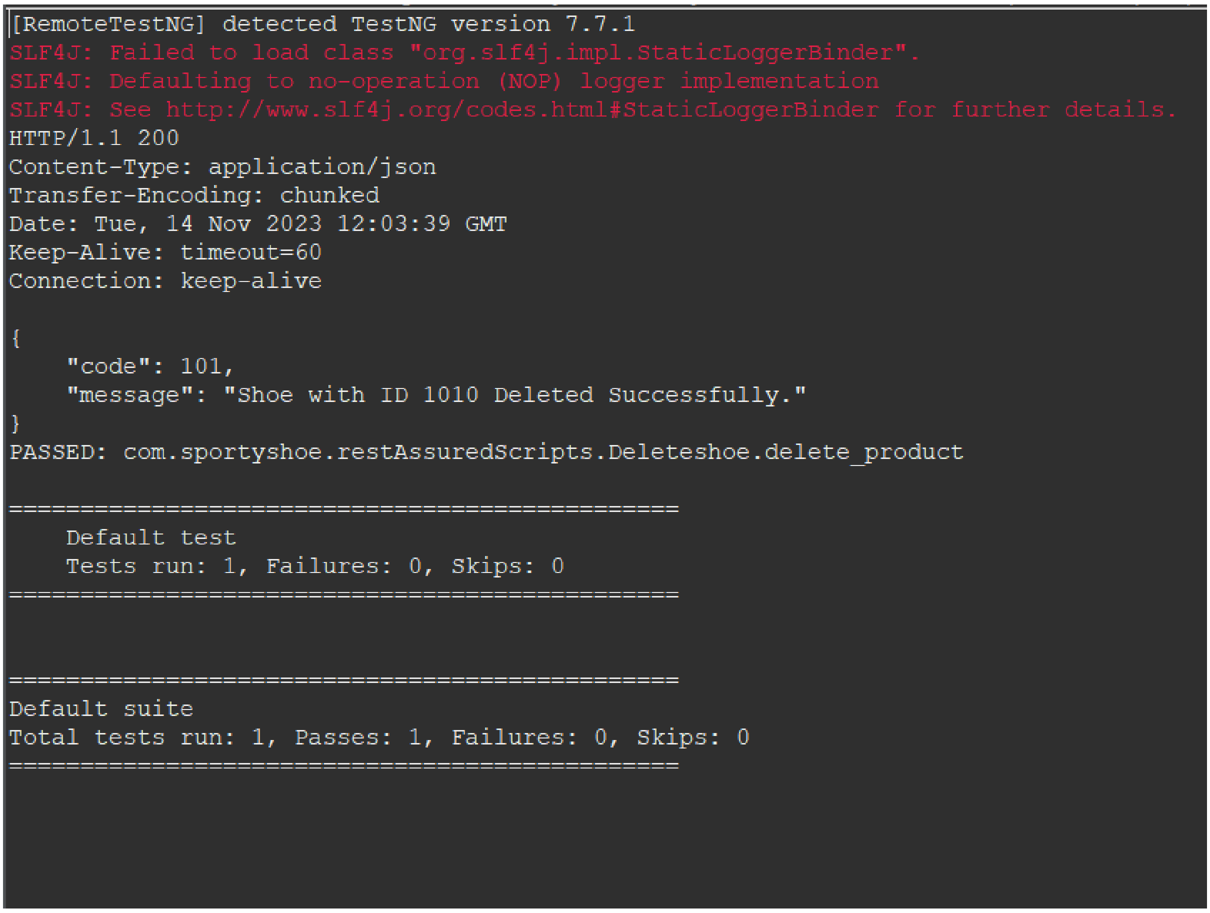
import org.testng.annotations.Test;

import io.restassured.RestAssured;

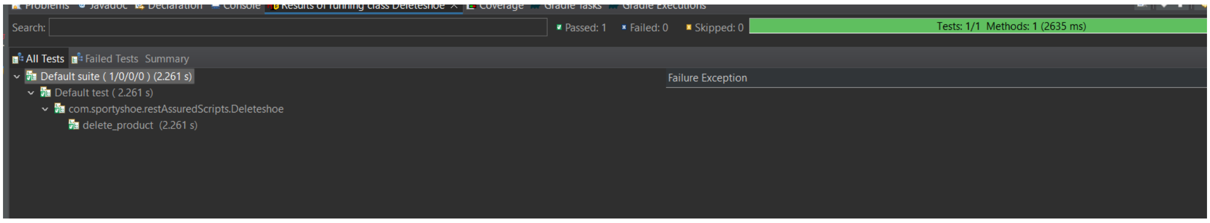
public class Deleteshoe {

```
    @Test(priority='1')

    public void delete_product()

    {


        RestAssured.given()

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

        .basePath("/delete-shoe")

        .queryParam("id", "1010")

        .when().delete()

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


    }


}
```

Output:



```
[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.
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 14 Nov 2023 12:03:39 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
    "code": 101,
    "message": "Shoe with ID 1010 Deleted Successfully."
}
PASSED: com.sportyshoe.restAssuredScripts.Deleteshoe.delete_product

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```



```
Passed: 1   Failed: 0   Skipped: 0   Tests: 1/1  Methods: 1 (2635 ms)

Default suite ( 1/0/0/0 ) (2.261 s)
  Default test ( 2.261 s)
    com.sportyshoe.restAssuredScripts.Deleteshoe
      delete_product  (2.261 s)
```

Selenium Scripts:

3.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

First go to eclipse create a package com.sportyshoe.seleniumcucumberscripts

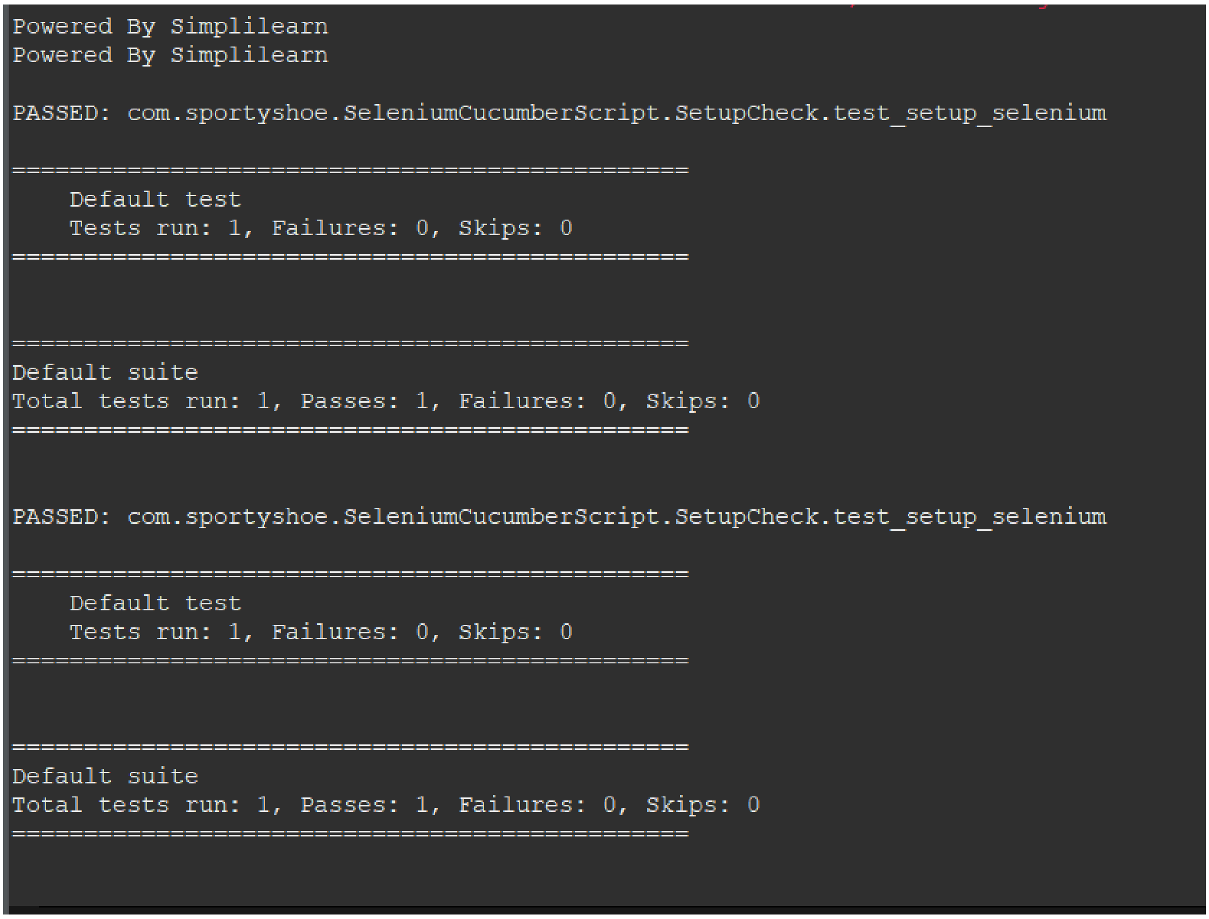
Under class setupcheck.java

Code:

```
package com.sportyshoe.SeleniumCucumberScripts;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class SetupCheck {


	@Test
	public void test_setup_selenium()
	{
		WebDriver driver = new ChromeDriver();

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

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

                String text = driver.findElement(By.xpath("//div[@class='container mt-3']/descendant::p[1]")).getText();

                System.out.println(text);

                System.out.println(driver.getTitle());
        }

}
```

Output:



```
Powered By Simplilearn
Powered By Simplilearn

PASSED: com.sportyshoe.SeleniumCucumberScript.SetupCheck.test_setup_selenium

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================


PASSED: com.sportyshoe.SeleniumCucumberScript.SetupCheck.test_setup_selenium

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

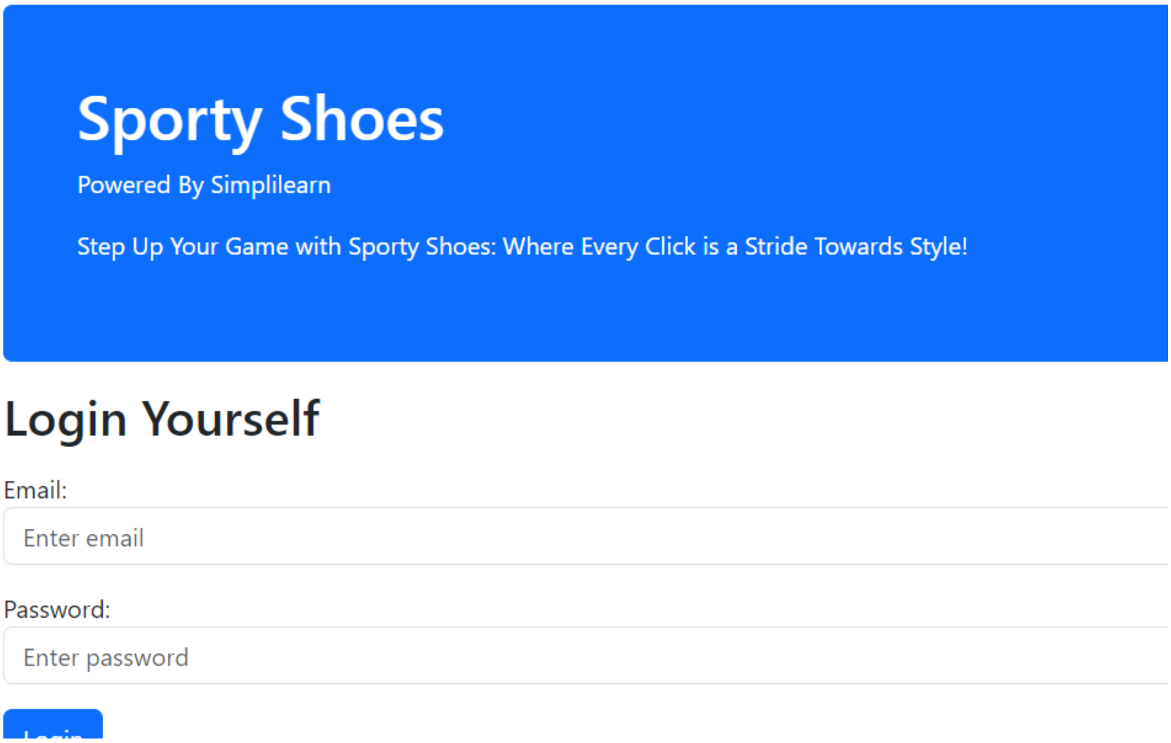
Next go to another class name as TestBase

Code:

```
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();
	}



}
```

Output:



Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Login Yourself

Email:

Enter email

Password:

Enter password

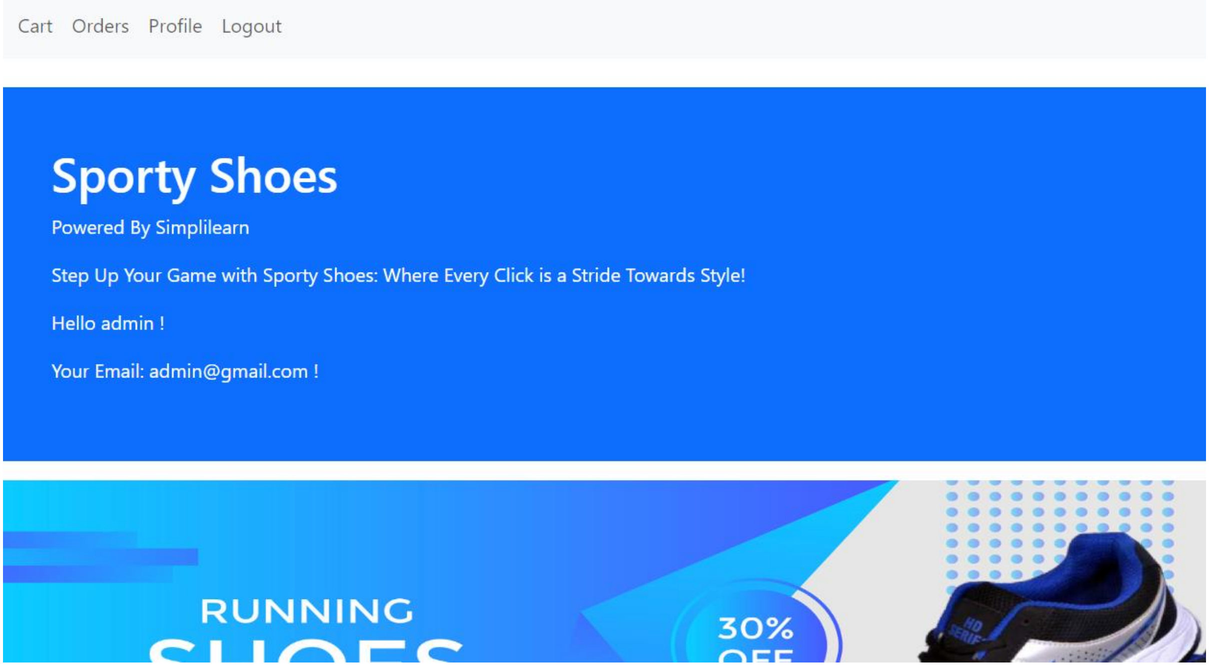
Next go to another class name as HomePage

Code:

```
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();
        }


}
```

Output:



Cart Orders Profile Logout

Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Hello admin !

Your Email: admin@gmail.com !

Next add another class Login Page under same package:

Code:

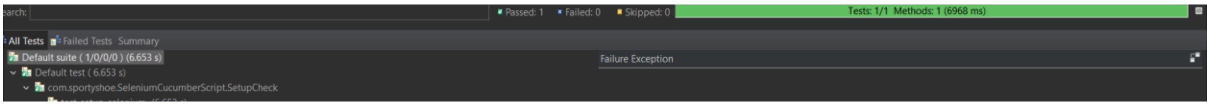
```
package com.sportyshoe.SeleniumCucumberScripts;


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("sonal@gmail.com");

        loginpassword.sendKeys("sonal@123");

        loginbtn.click();

}



public void click_cart()

{

        clickCart.click();

}
```

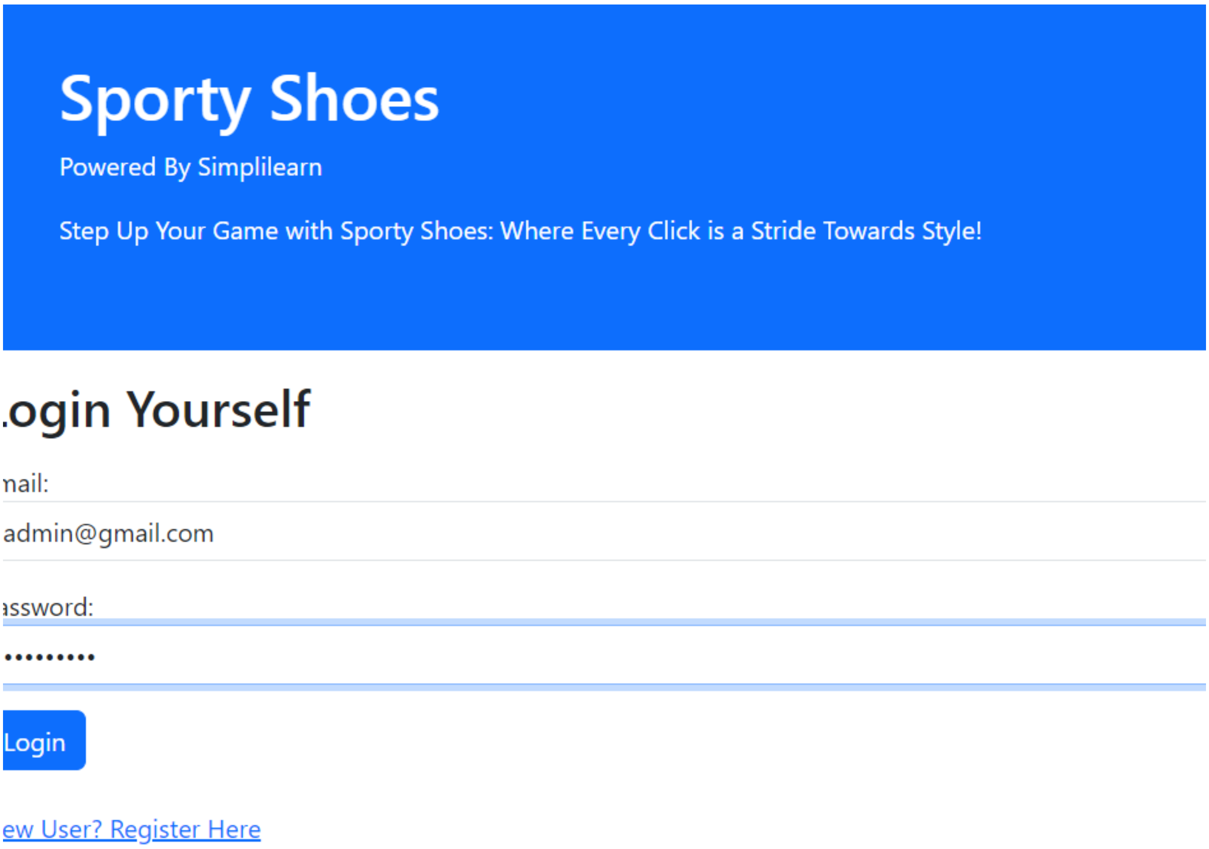
```
}
```

```
}
```

Output:



```
earch:                    ■ Passed: 1   ■ Failed: 0   ■ Skipped: 0   Tests: 1/1  Methods: 1 (6968 ms)
All Tests  Failed Tests  Summary
Default suite ( 1/0/0/0 ) (6.653 s)                        Failure Exception
  Default test ( 6.653 s)
    com.sportyshoe.SeleniumCucumberScript.SetupCheck
```



Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

.ogin Yourself

nail:

admin@gmail.com

assword:

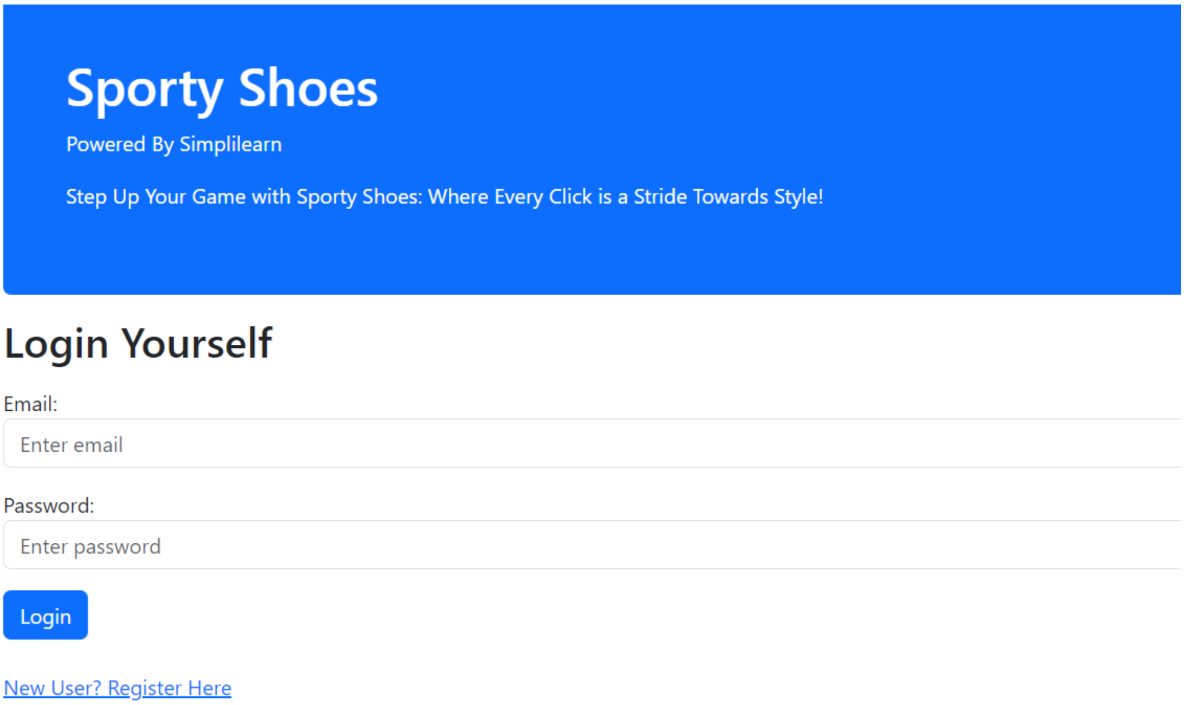
•••••••••

ew User? Register Here

Next add another class under same package name as Register Page

```
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 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("sonal");
                registeremail.sendKeys("sonal@gmail.com");
                registerpassword.sendKeys("sonal@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;
        }
```

Output:



Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Login Yourself

Email:

Enter email

Password:

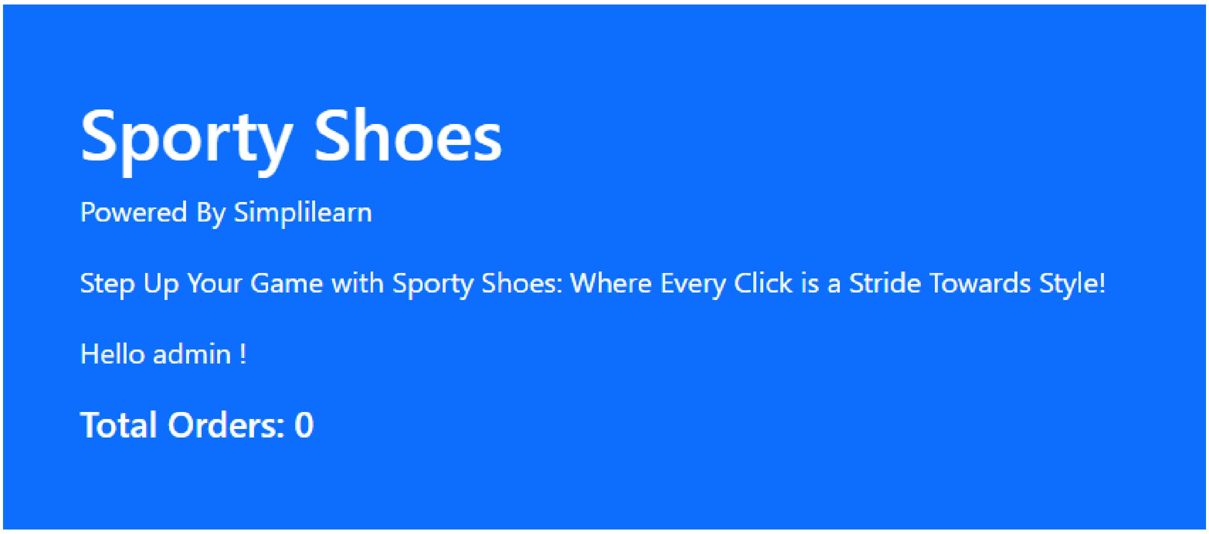
Enter password

New User? Register Here

Next add another class name as orderPage under the same package

```
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 OrderPage {


	@FindBy(linkText="Orders")

	WebElement orderlink;
	public OrderPage(WebDriver driver) {

		PageFactory.initElements(driver, this);

	}


	public void click_orderPage()

	{

		orderlink.click();


	}


}
```

Output:



Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Hello admin !

Total Orders: 0

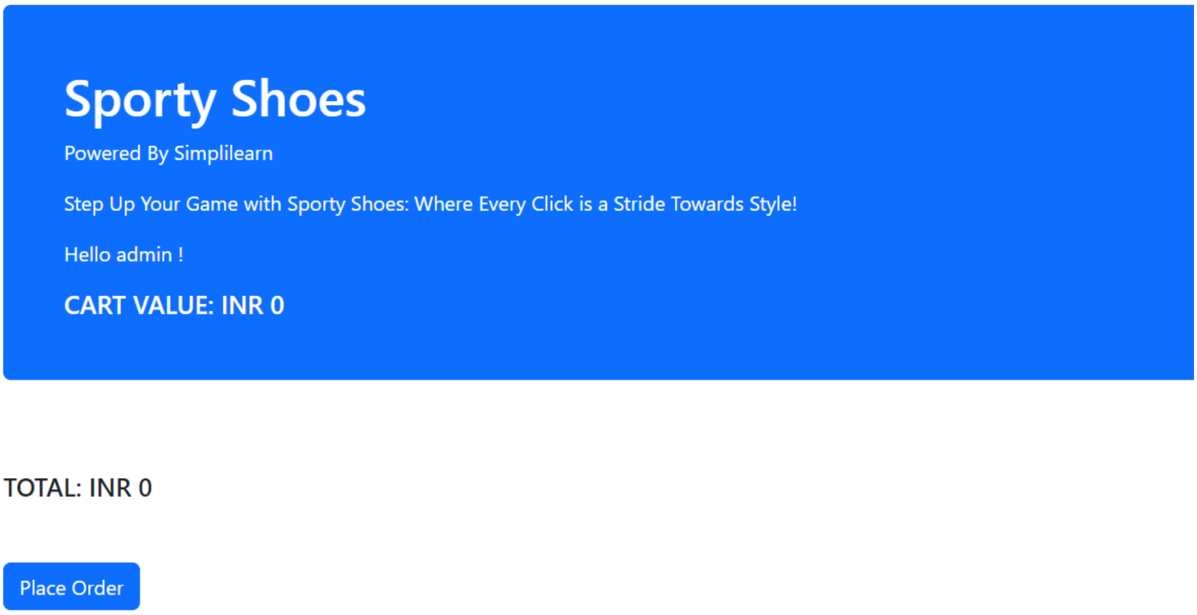
Next step add another class name as AddCart Page under same package

```
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;

          }



}
```

Output:



Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Hello admin !

CART VALUE: INR 0

TOTAL: INR 0

Next go to create another package name com.sportyshoe.Tests

Under class name as orderPageTests

Code:

```
package com.sportyshoe.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.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);


	}

}
```

Next class AddCartPage under same package

```
package com.sportyshoe.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);
    }




}
```

Next go to another class name HomePageTests

Code:

```
package com.sportyshoe.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();
	}
	}
```

Next go to add another class LoginPage Test under same Package

Code:

```
package com.sportyshoe.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 sonal !";
        String actualText = rp.validate_registration_Text();
        Assert.assertEquals(actualText, expected);
    }


@Test(priority='4')

public void test_click_cart()
```

```
{
    lp.click_cart();
}



}
```

Next go to another class add registerPagetests

Code:

```
package com.sportyshoe.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);
```

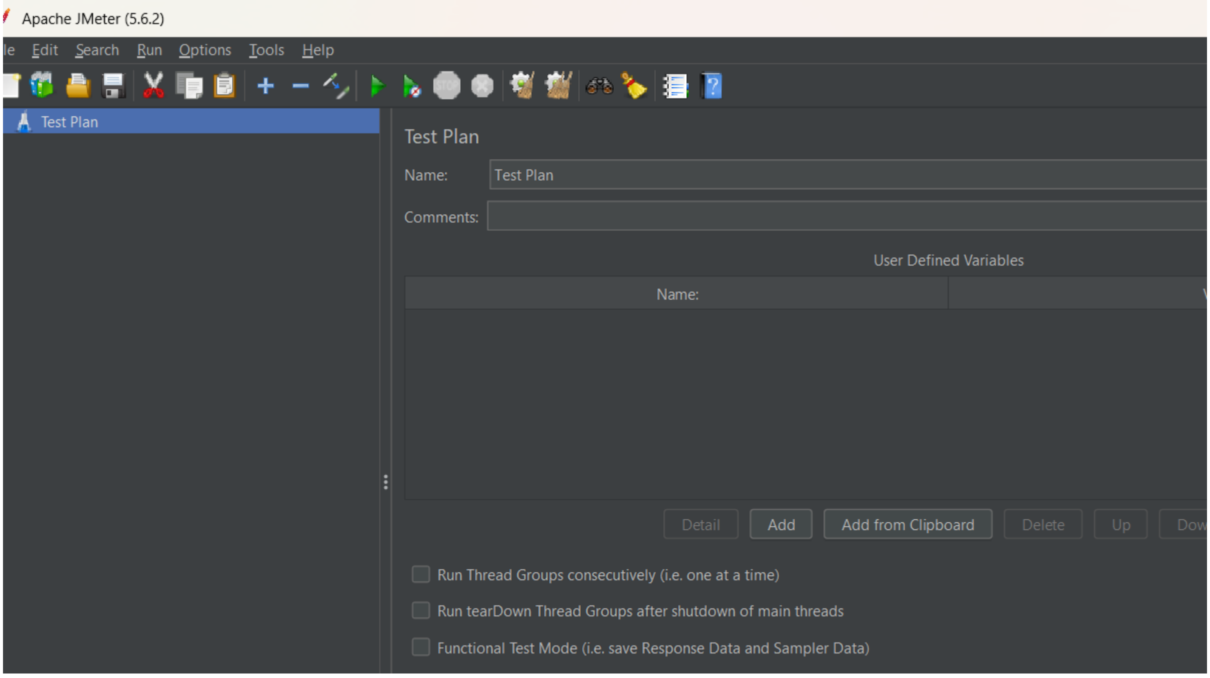
```
}



@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 sonal !";
                    String actualText = rp.validate_registration_Text();
                    Assert.assertEquals(actualText, expected);
          }




}
```

Jmeter:

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

First go to Apache Jmeter



Apache JMeter (5.6.2)

Test Plan

Name: Test Plan

Comments:

User Defined Variables

Name:

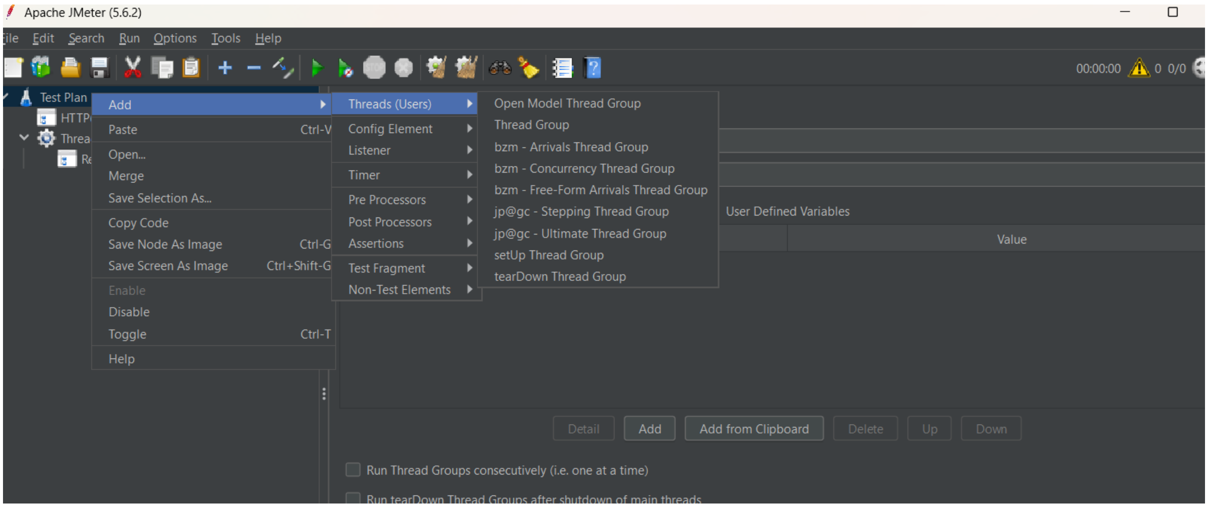
☐ Run Thread Groups consecutively (i.e. one at a time)

☐ Run tearDown Thread Groups after shutdown of main threads

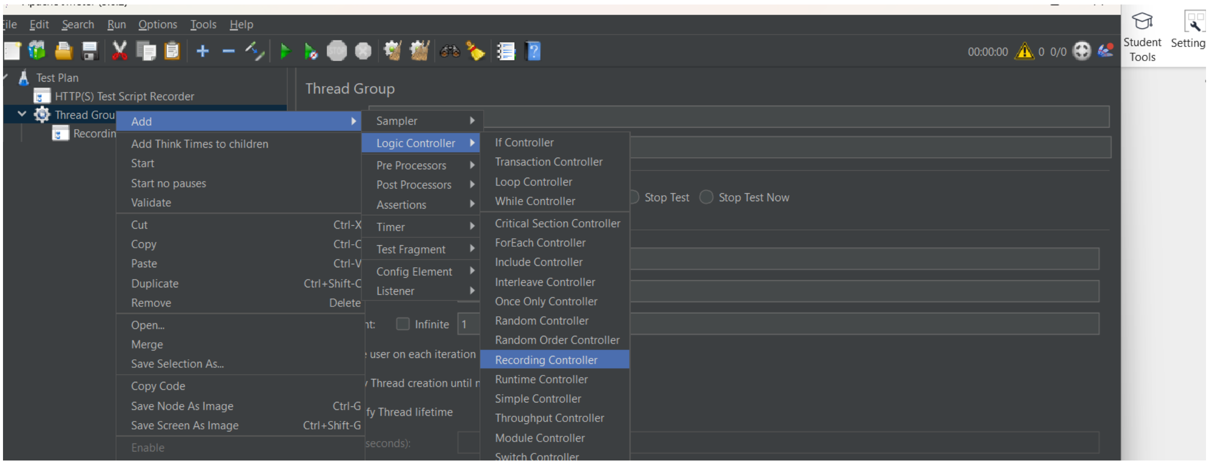
☐ Functional Test Mode (i.e. save Response Data and Sampler Data)

In this test plan add HTTP(s) Test Script Recorder

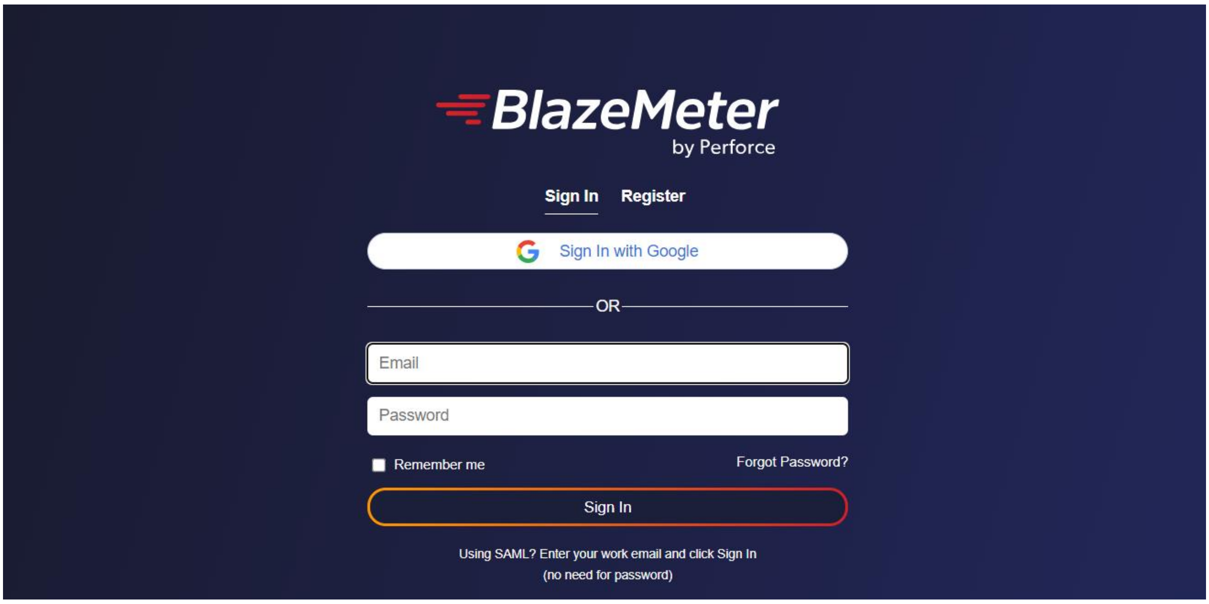
Next Add thread group

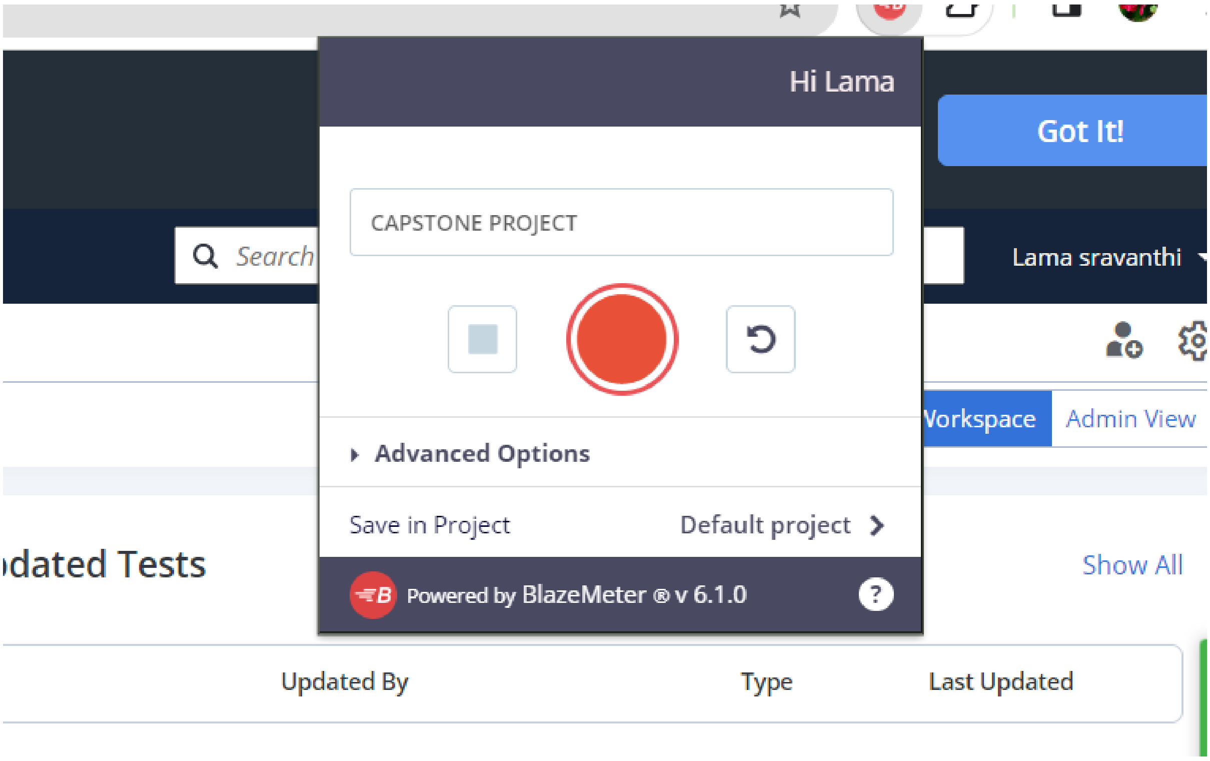


In thread group add controller >logic controller>Recording controller

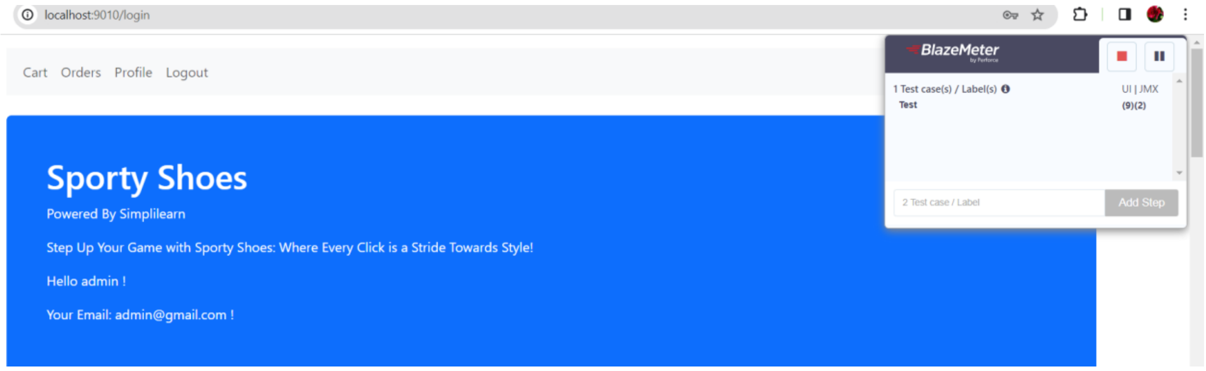


Next go to blazemeter for recording Scripts





This is login page



Sporty Shoes

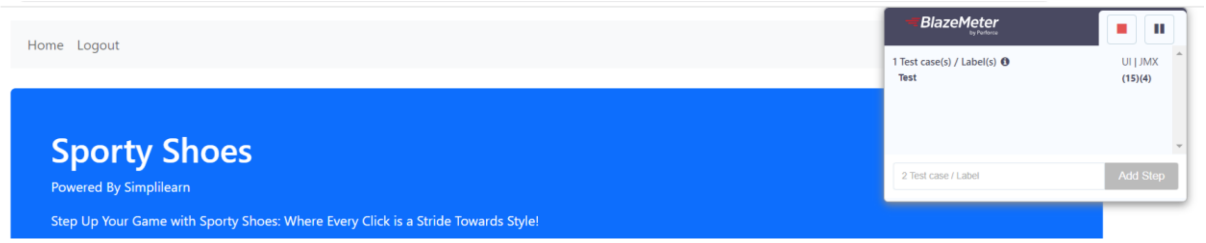
Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

Hello admin !

Your Email: admin@gmail.com !

Logout Page

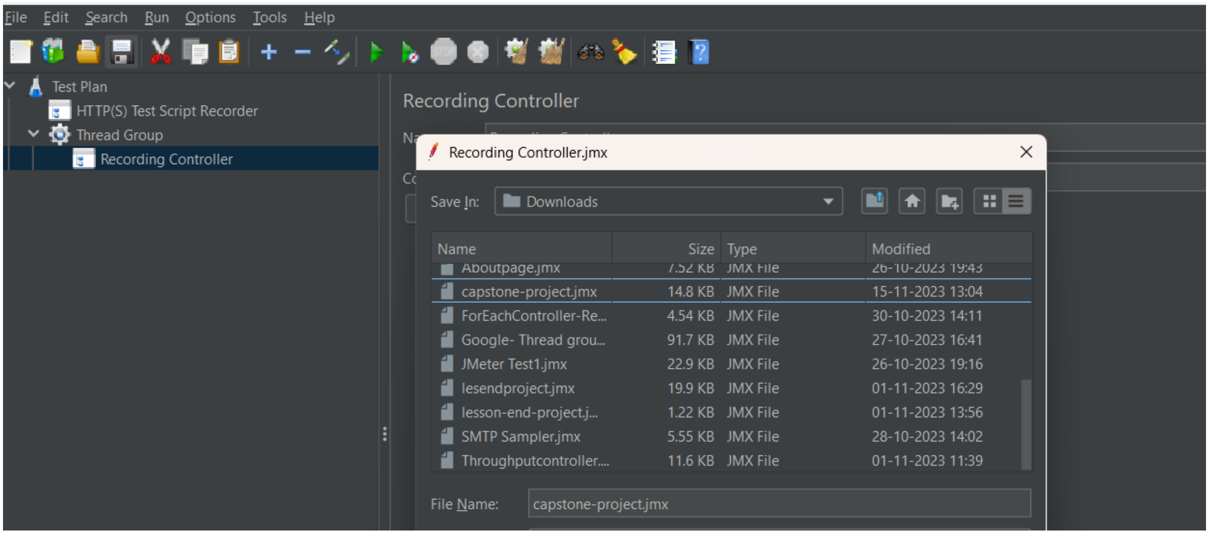


Sporty Shoes

Powered By Simplilearn

Step Up Your Game with Sporty Shoes: Where Every Click is a Stride Towards Style!

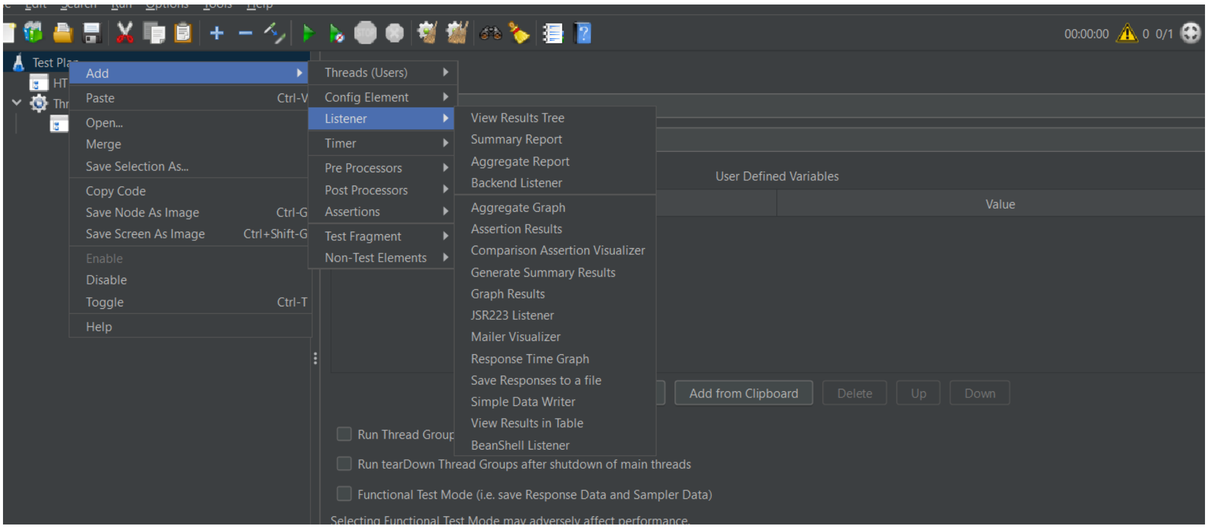
And save the recorder controller in jmx format



Name	Size	Type	Modified
Aboutpage.jmx	7.52 KB	JMX File	26-10-2023 19:43
capstone-project.jmx	14.8 KB	JMX File	15-11-2023 13:04
ForEachController-Re...	4.54 KB	JMX File	30-10-2023 14:11
Google- Thread grou...	91.7 KB	JMX File	27-10-2023 16:41
JMeter Test1.jmx	22.9 KB	JMX File	26-10-2023 19:16
lesendproject.jmx	19.9 KB	JMX File	01-11-2023 16:29
lesson-end-project.j...	1.22 KB	JMX File	01-11-2023 13:56
SMTP Sampler.jmx	5.55 KB	JMX File	28-10-2023 14:02
Throughputcontroller....	11.6 KB	JMX File	01-11-2023 11:39

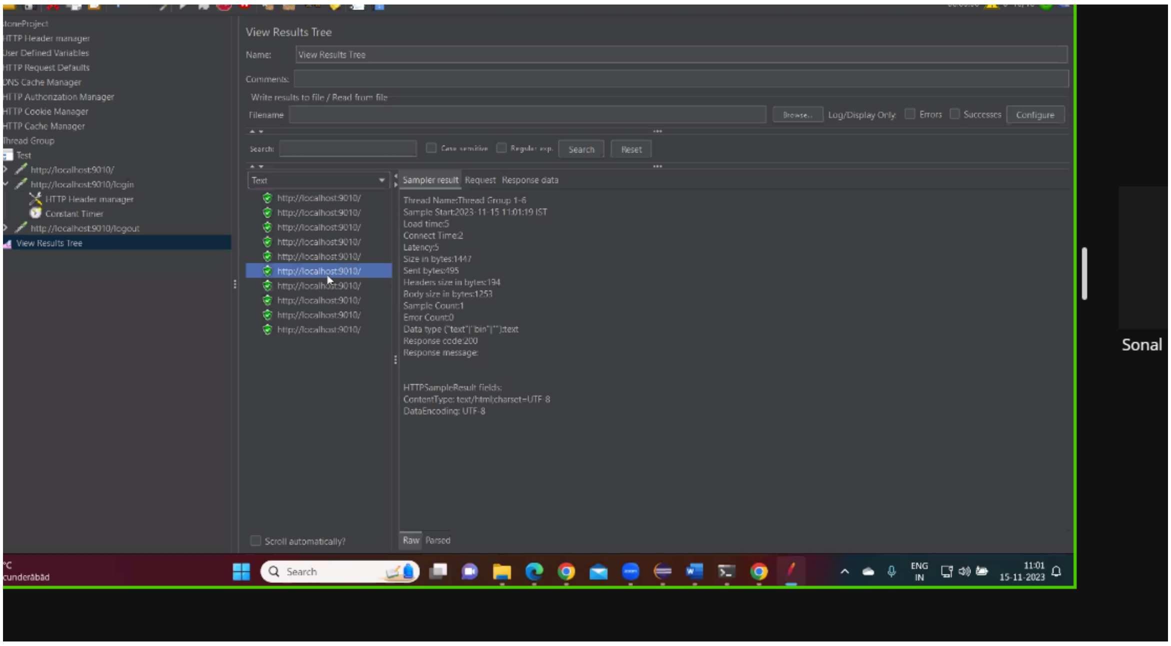
File Name: capstone-project.jmx

Add Listener to see results



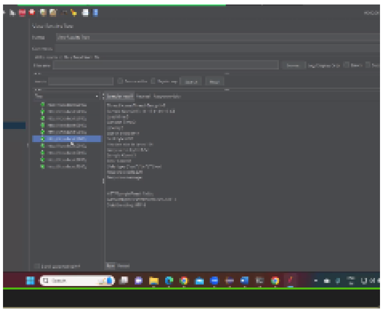
□ Run Thread Group
□ Run tearDown Thread Groups after shutdown of main threads
□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.



```
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
http://localhost:9010/
```

```
Thread Name:Thread Group 1-6
Sample Start:2023-11-15 11:01:19 IST
Load time:5
Connect Time:2
Latency:5
Size in bytes:1447
Sent bytes:495
Headers size in bytes:194
Body size in bytes:1253
Sample Count:1
Error Count:0
Data type ("text"|"bin"|""):text
Response code:200
Response message:


HTTPSampleResult fields:
ContentType: text/html;charset=UTF-8
DataEncoding: UTF-8
```



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

First go to eclipse create a java project add feature file convert into step defintion file

```
# This is used to write comments
# this is like your test plan
# This is my first feature file
Feature: first login to the spotyshoe. login page using local host:9010

  # In the scenario if there are stpes that are common to be executed in the begining
  # as a pre-requisite then we use the keyword Background
  # The below statements demonstrates the use of backgroud keyword
  # these statements will be executed before every scenario in the current feature file **

  Background:
    Given I open the browser and enter URL
    Then I capture the title of the Page
    When I enter username and password
    Then I click on login button


  # we are using Gherkin keywords
  # each step should have a matching Code
  # for every stpes we have to write code in step definition file

  Scenario: User has to test if login on sportyshoe.login is successful or not
    And I should see an Error message
    Then I click on Click Here link
    And I close the browser

  Scenario:
    Then I click on Create a sportyshoe. account
    Then I capture the title of the Page
    And I close the browser
```

Steps:

```
package com.sportyshoe.steps;

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("sonal@gmail.com");
                loginpassword.sendKeys("sonal@123");
                loginbtn.click();
        }
```

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




}
```