

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

AIM: Create a Testing Framework for Sporty Shoes Website

SOFTWARE: Eclipse, JMeter, Postman

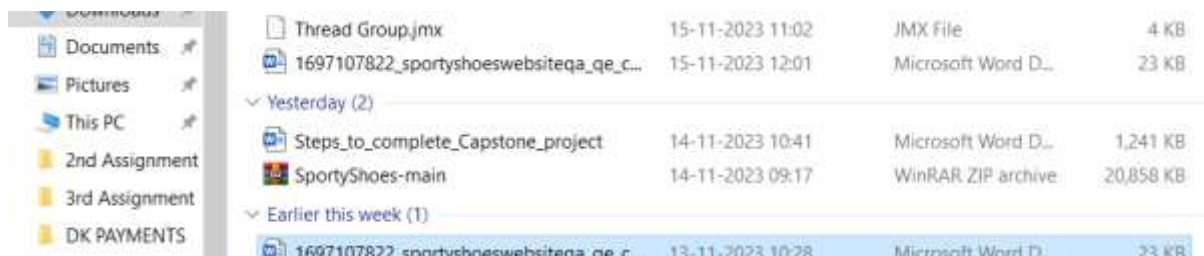
Step 1: Launch the Application under test.

Go to GitHub URL <https://github.com/Simplilearn-Edu/SportyShoes.git>

Click on Green button which has Code written on it. Click on Download Zip

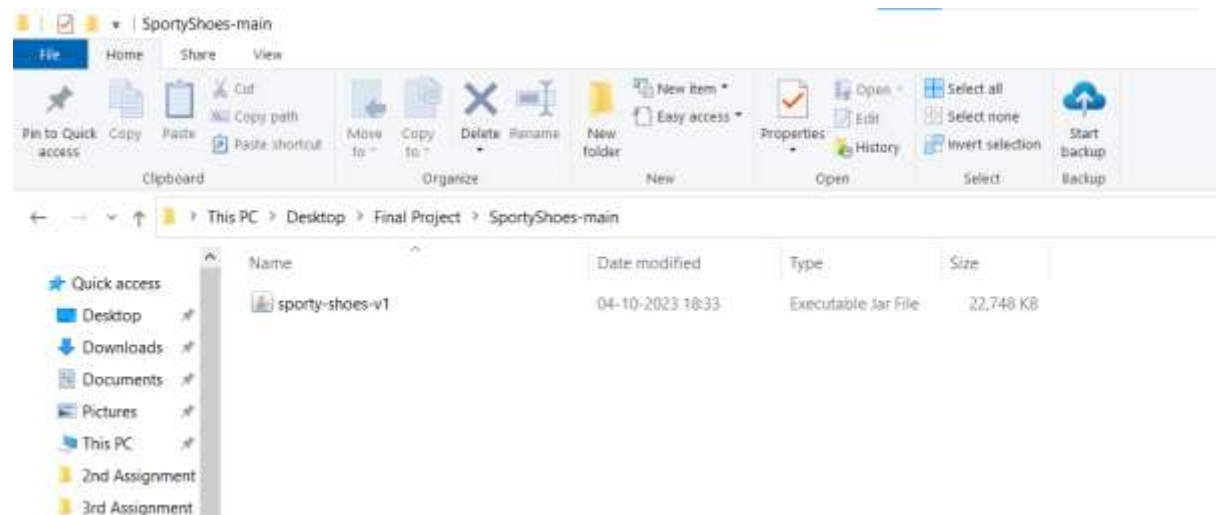
The download zip file will be in your download folder.

Extract all the files in downloads folder.



Go inside the folder Sporty Shoes-main and then again go in the folder Sporty Shoes-main.

Go inside the folder where you see the sporty-shoes-v1 jar file.

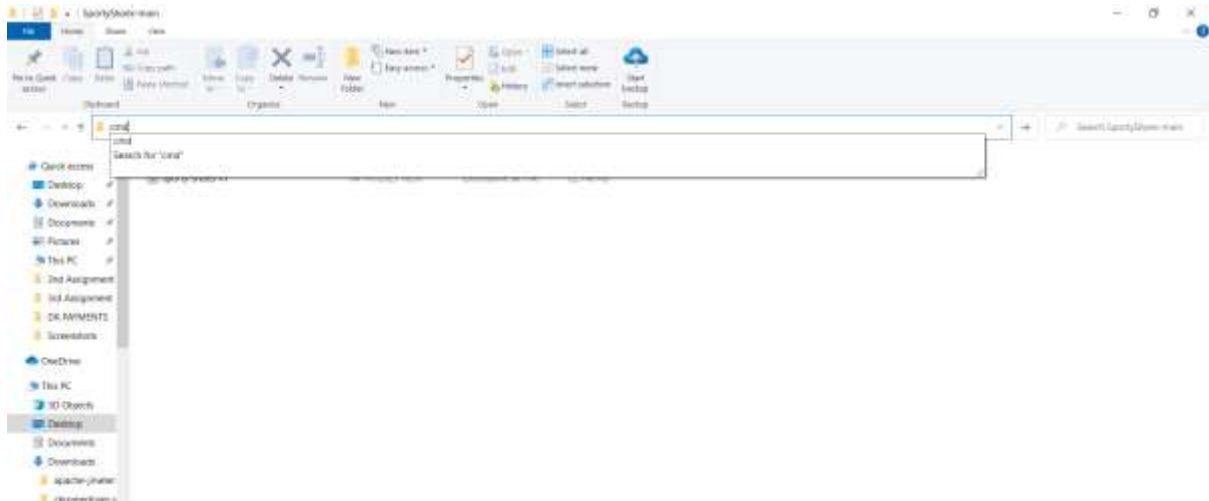


CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Start cmd prompt in the same folder.

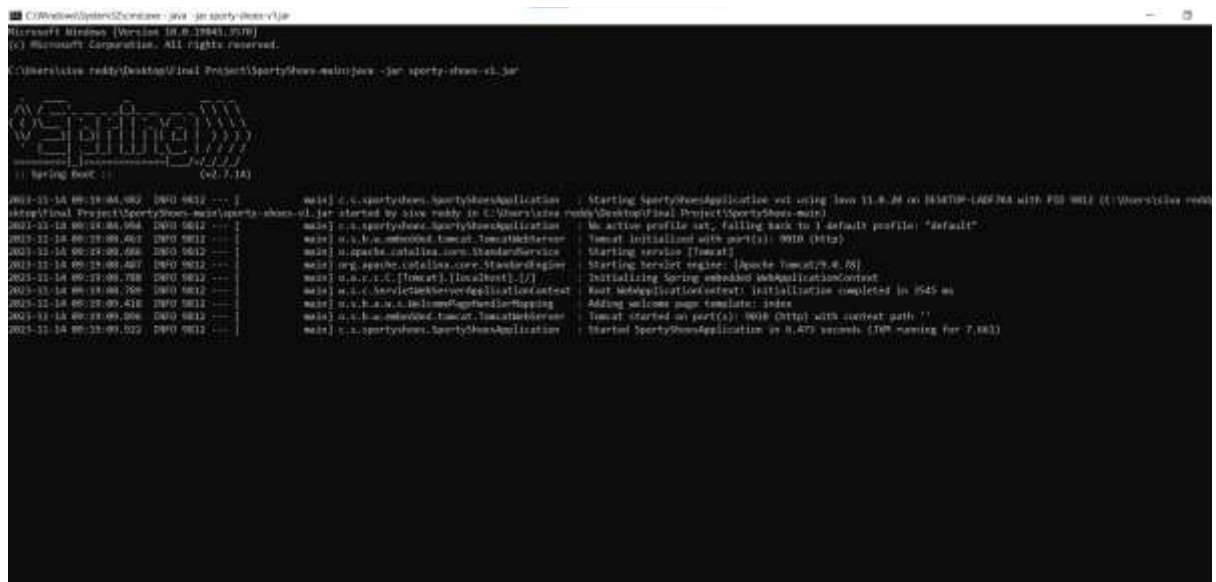


Execute the java command to run the jar file and deploy the application

`java -jar sporty-shoes-v1.jar`

OR

`java -jar -Dserver.port=8100 sporty-shoes-v1.jar`



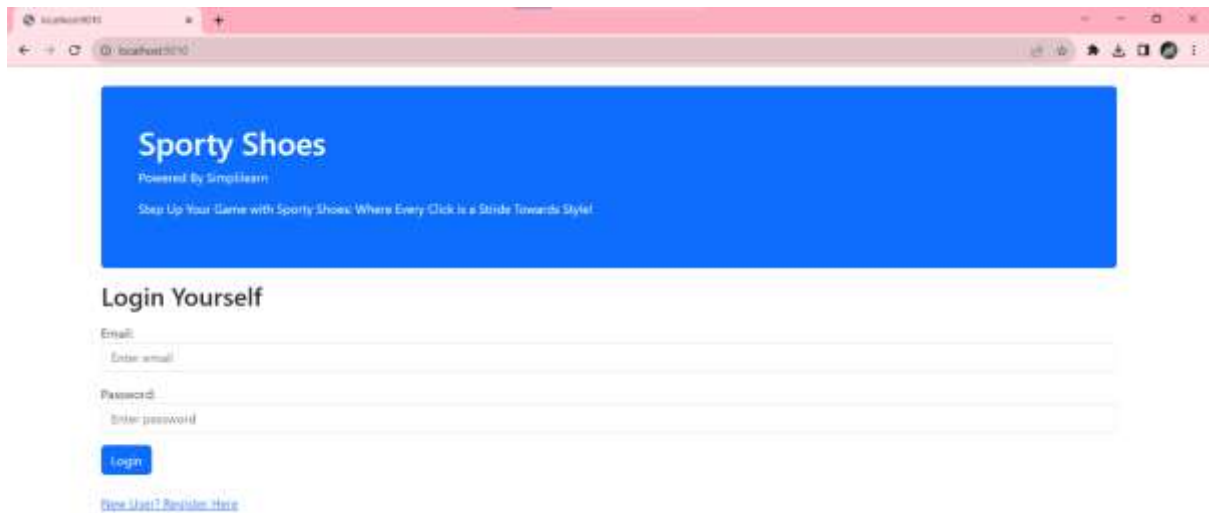
After Execute the commands in command prompt until it minimize the window to run the postman execution.

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Now we open chrome browser to run the command localhost:9010 command to open or not.



Implementing Postman steps:

We can open the postman to create the blank collection and its name is capstone project.

After we add a request.

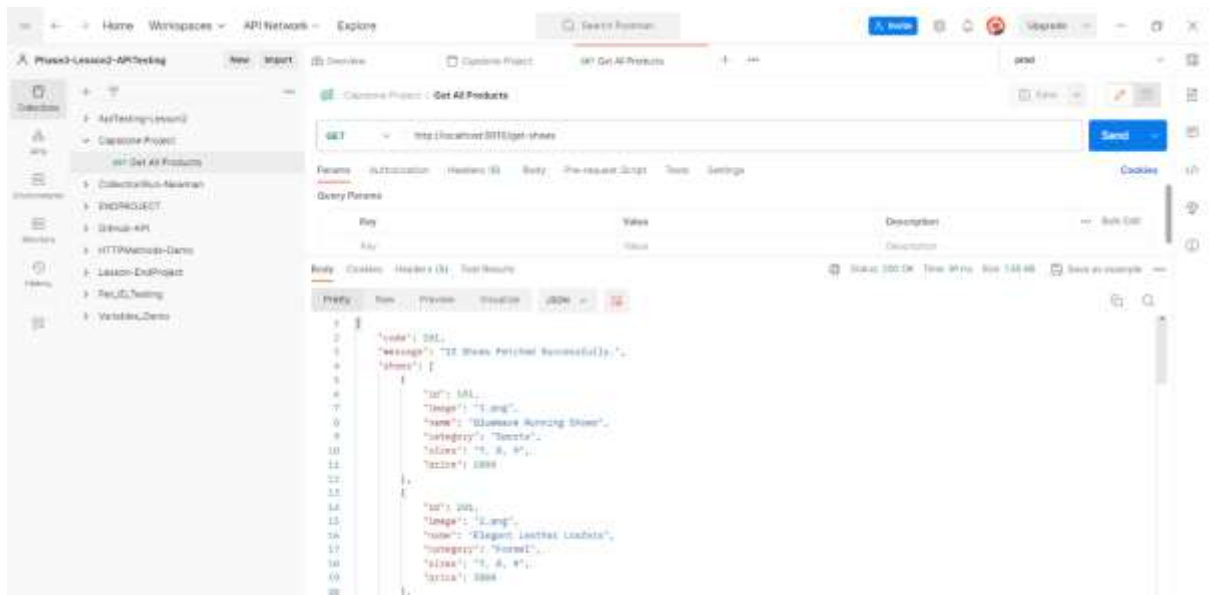
Step1: Retrieve the list of all products in the store.

We can use command in postman to use “<http://localhost:9010/get-shoes>”, to use ‘**GET**’ method and get all products of shoes.

CAPSTONE PROJECT

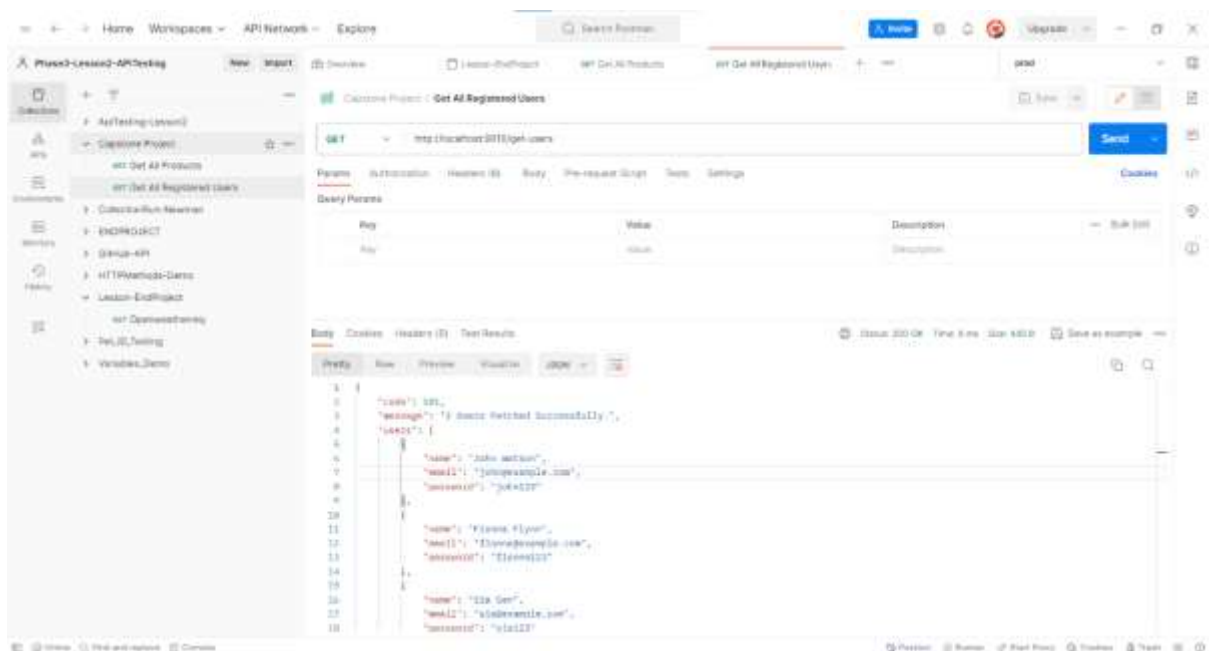
NAME: K SIVA REDDY

ID: 2576232



Step2: Retrieve the list of all registered users.

Again, we can add request to get all registered users. Now we can use command to get all registered users "<http://localhost:9010/get-users>" again we can use the "GET" method.



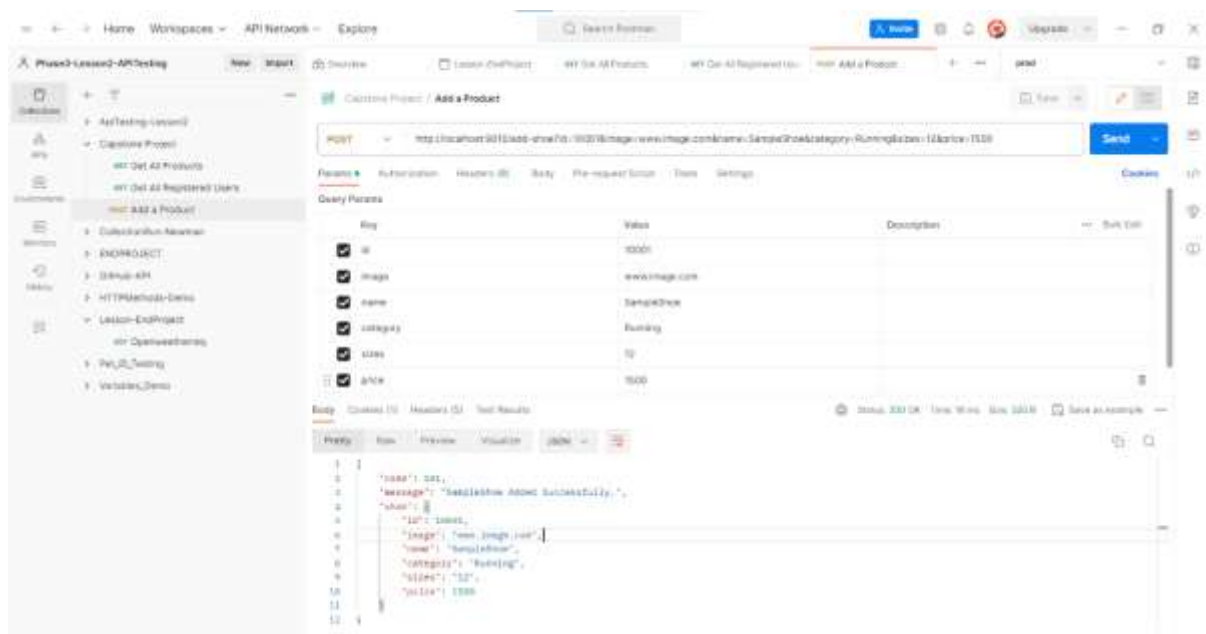
CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Step3: Add Product URL Sample with POST Method.

Again, we can add request to add the product. Now we can use command to add the product “<http://localhost:9010/add-shoe?id=10001&image=www.image.com&name=SampleShoe&category=Running&size=12&price=1500>” we can use the “**POST**” method.



Step4: Update Product URL Sample with PUT Method.

Again, we can add request to update the product. Now we can use command to update the product “<http://localhost:9010/update-shoe?id=101&image=www.image.com&name=SampleShoe&category=Running&size=6,7,9&price=5000>” we can use the “**PUT**” method.

We can add the body in raw type is JSON.

CAPSTONE PROJECT

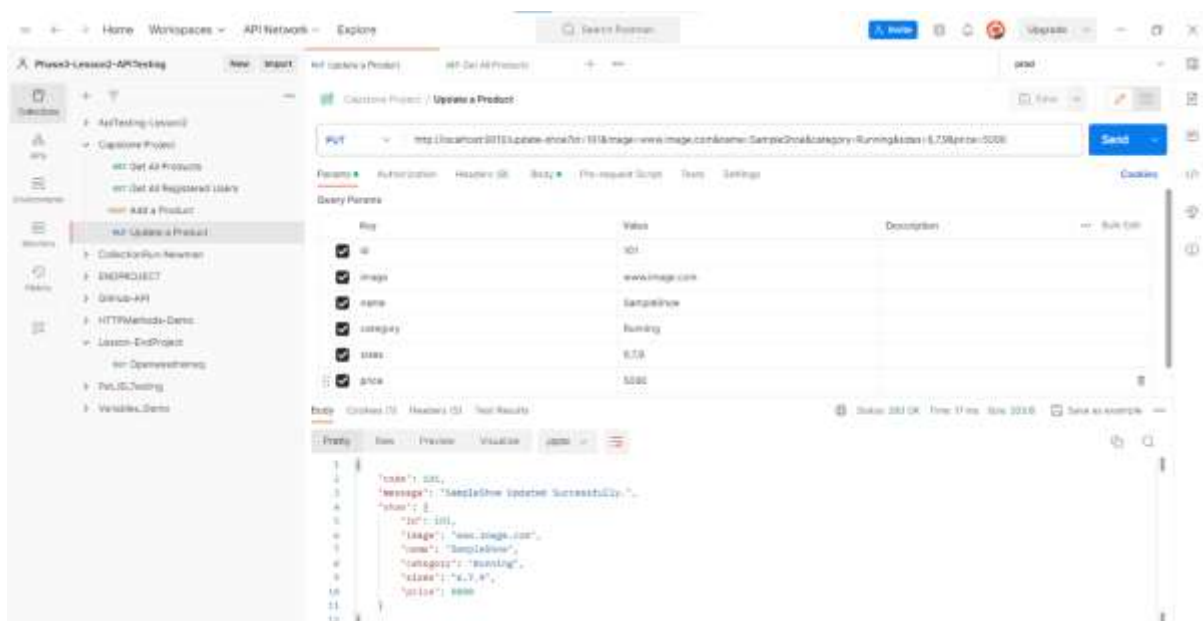
NAME: K SIVA REDDY

ID: 2576232

Request body (<http://localhost:9010/update-shoe>)

```
{  
  "id": 101,  
  "name": "Updated Shoe Name",  
  "category": "Updated Category",  
  "sizes": "8,9,10",  
  "price": 1500  
  "image": "updated_image_url",  
}
```

In this body cannot run in the postman its not execute the body so that we can make the change in given URL so now we can run the command it execute the command.

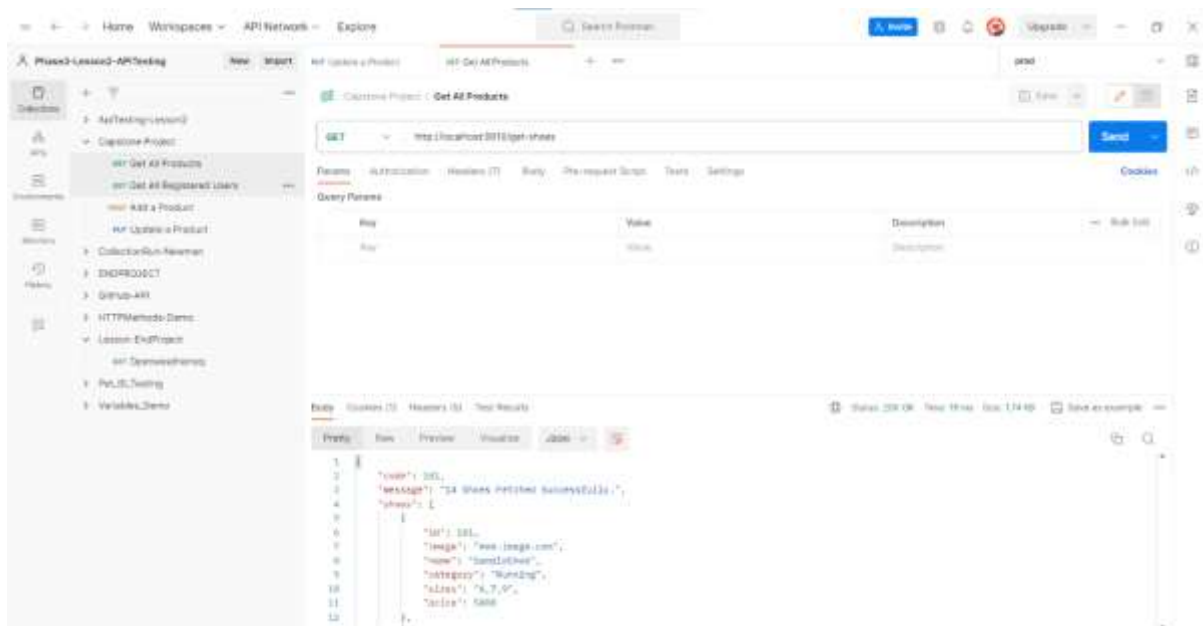


Now we can again run the get command it product details update or not check the run

CAPSTONE PROJECT

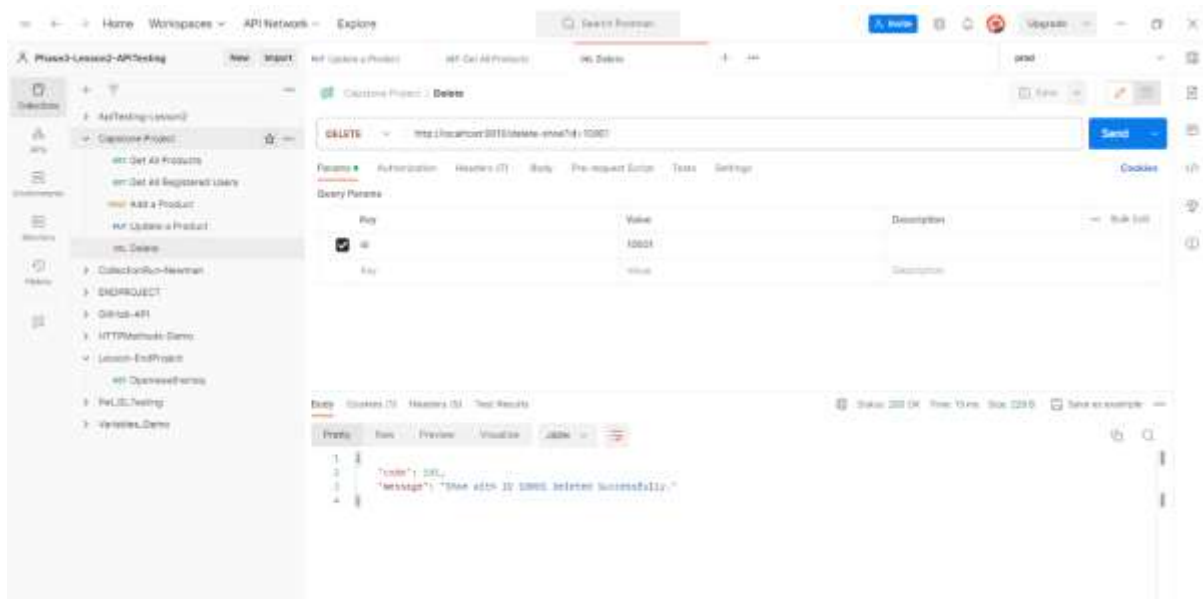
NAME: K SIVA REDDY

ID: 2576232



Step5: Delete Product URL Sample with DELETE Method

Again, we can add request to delete the product. Now we can use command to delete the product “<http://localhost:9010/delete-shoe?id=10001>” we can use the “**DELETE**” method.

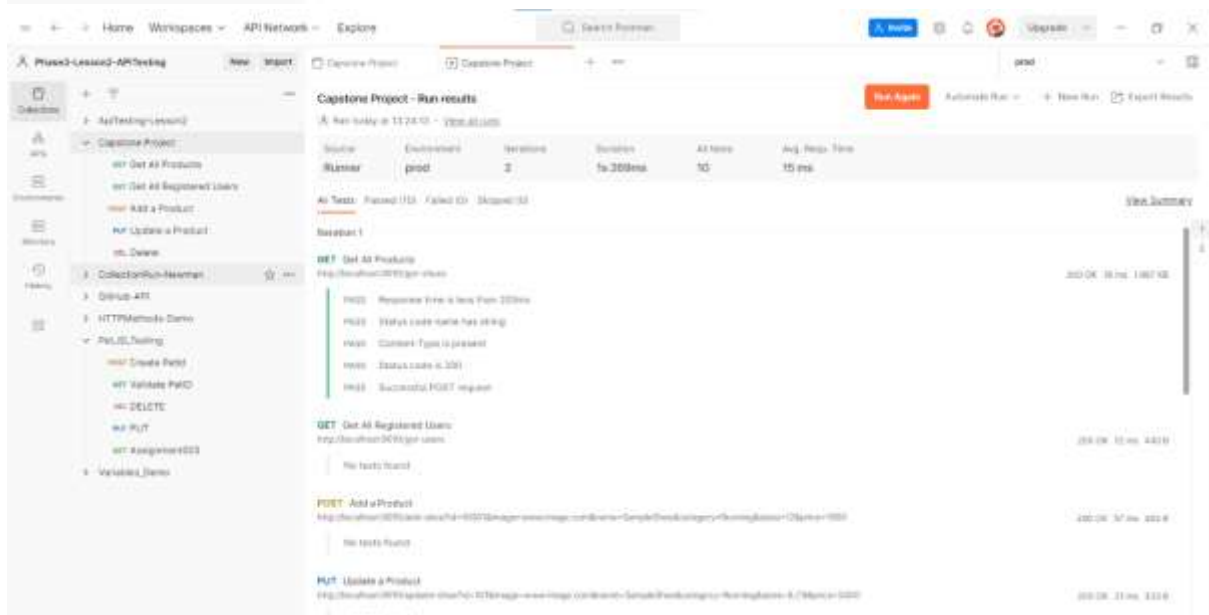
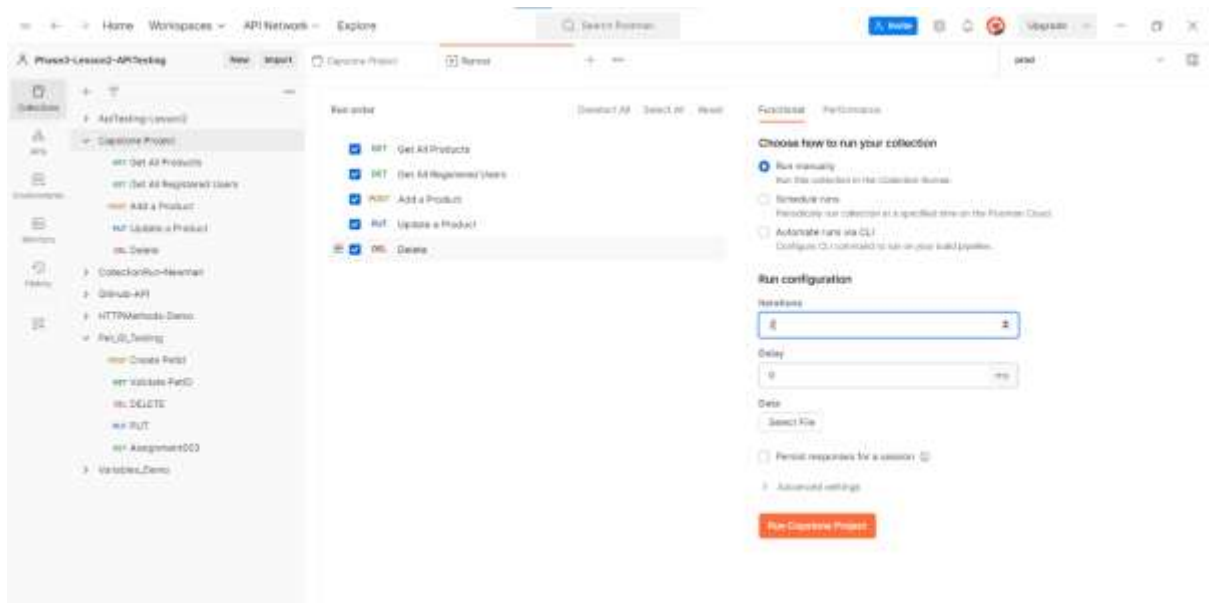


CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

In postman we can run all commands now we create, add, update, delete, get all details products of shoes.

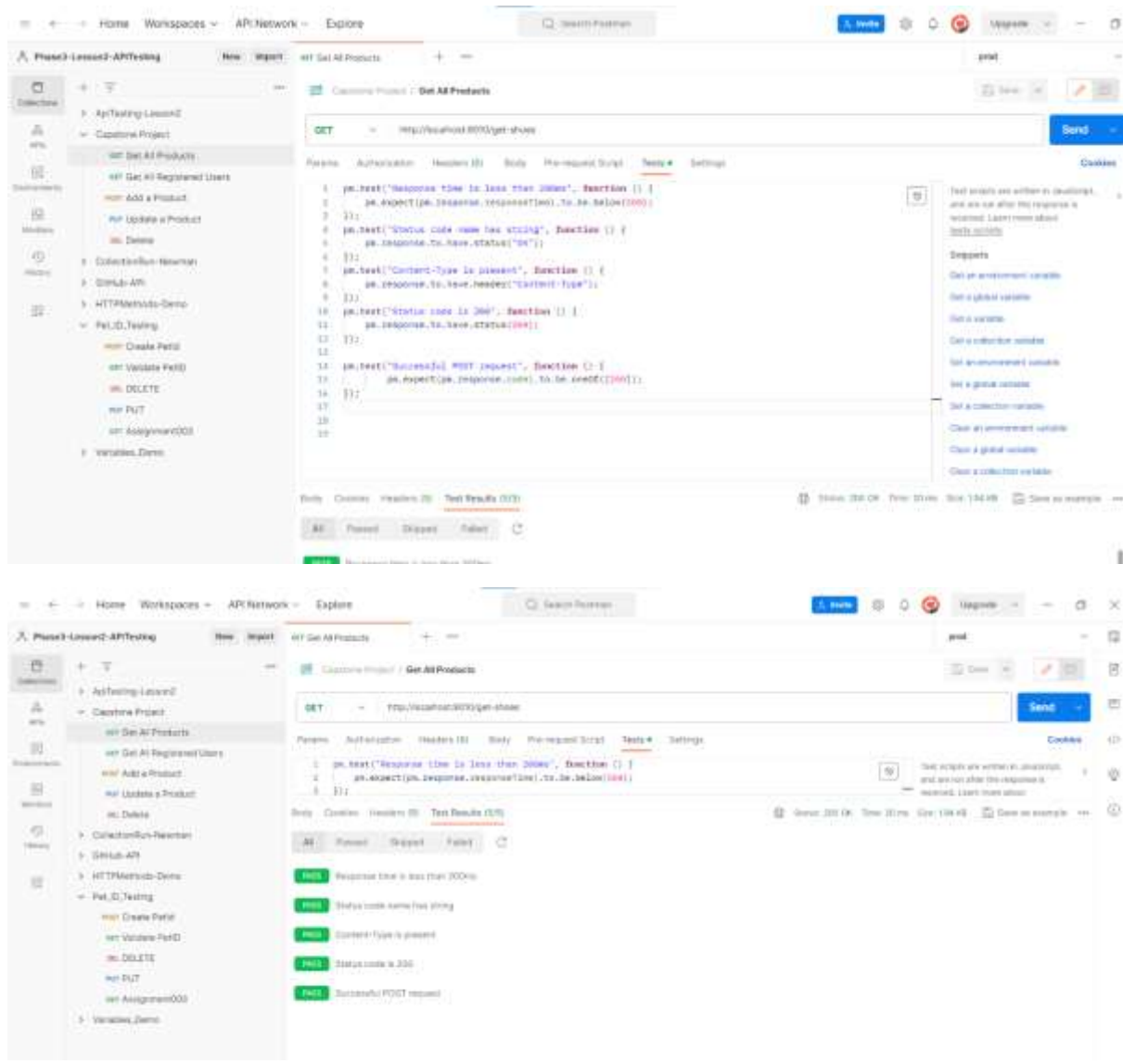


And we can run different test scripts in the postman.

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



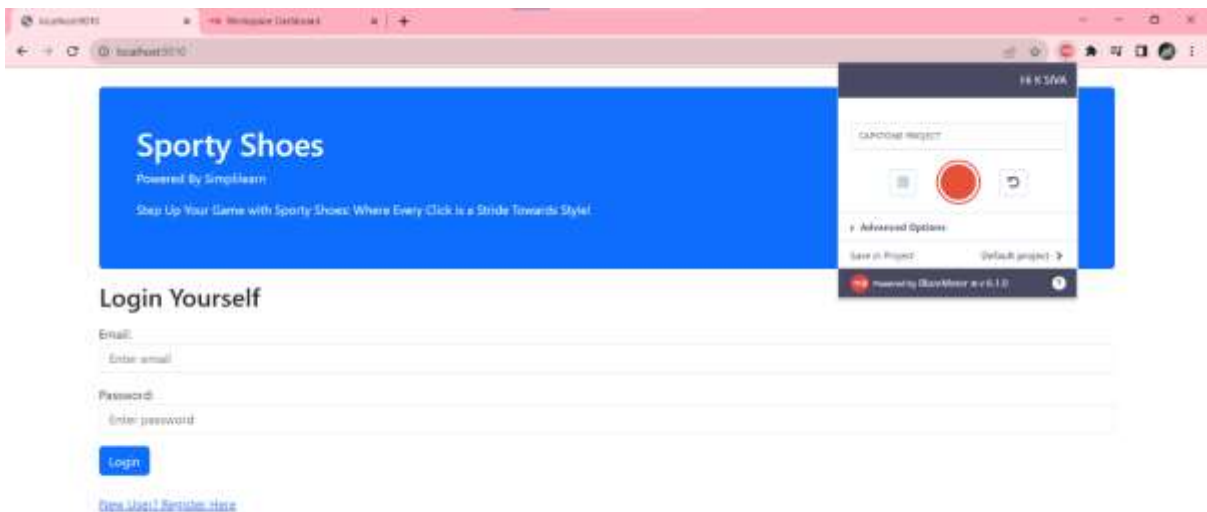
CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

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

Now we can run the home page login and logout user to run with help of blaze meter.

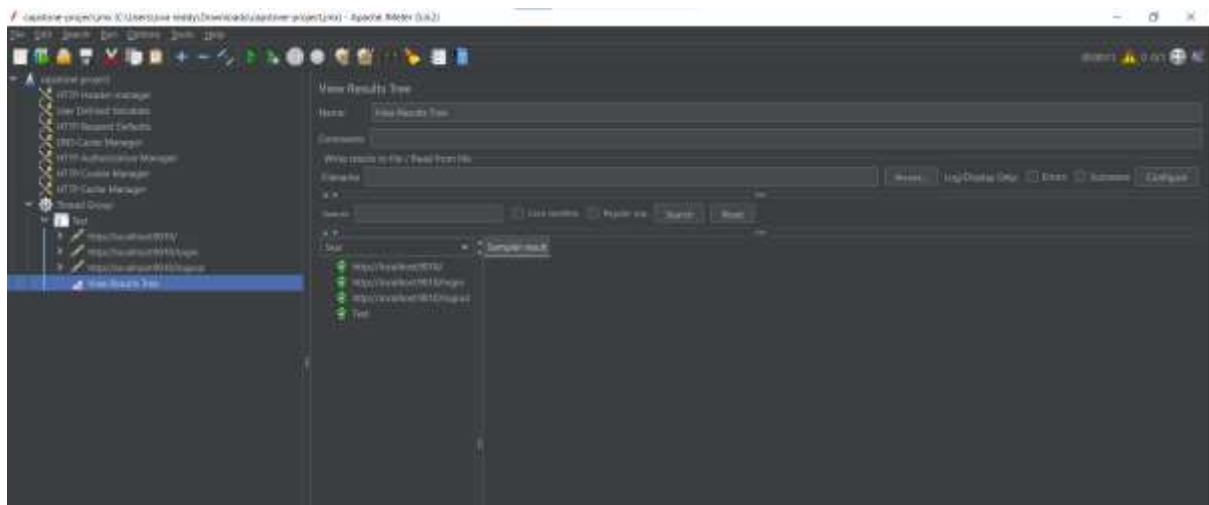


Now we can add the run file and run it.

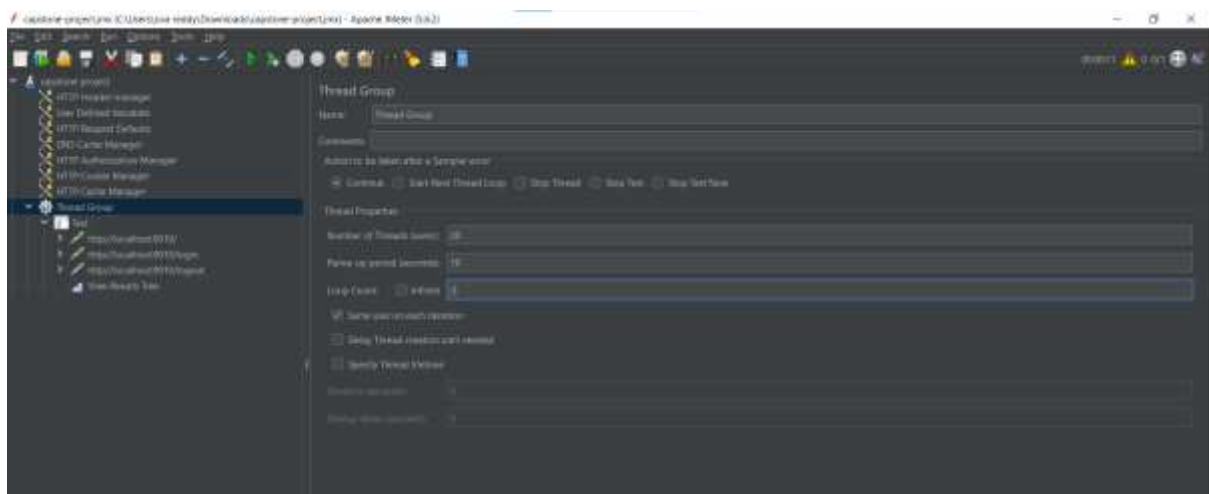
CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



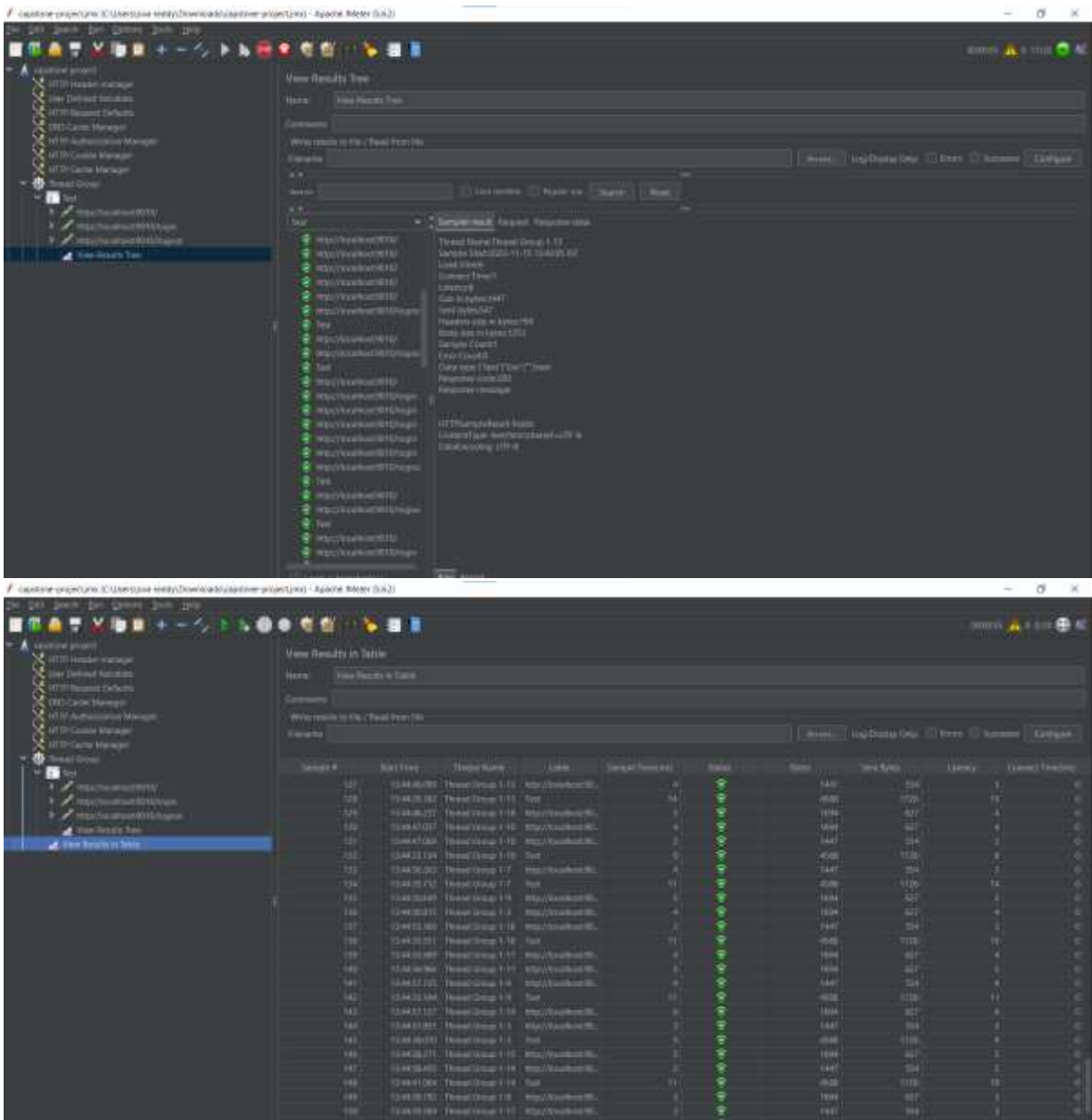
Now we can add the load in file increase the users to run the file.



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

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, we can add all selenium and cucumber drivers.

1. HOME PAGE:

Code:

Code for Home 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 HomePage extends TestBaseClass {
```

```
    @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()  
{
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
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();
}

}
```

Code for 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.TestBaseClass;

import static org.testng.Assert.assertEquals;

import org.testng.Assert;
import org.testng.Assert.*;

public class HomePageTest extends TestBaseClass {

    HomePage hp;
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@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);
}

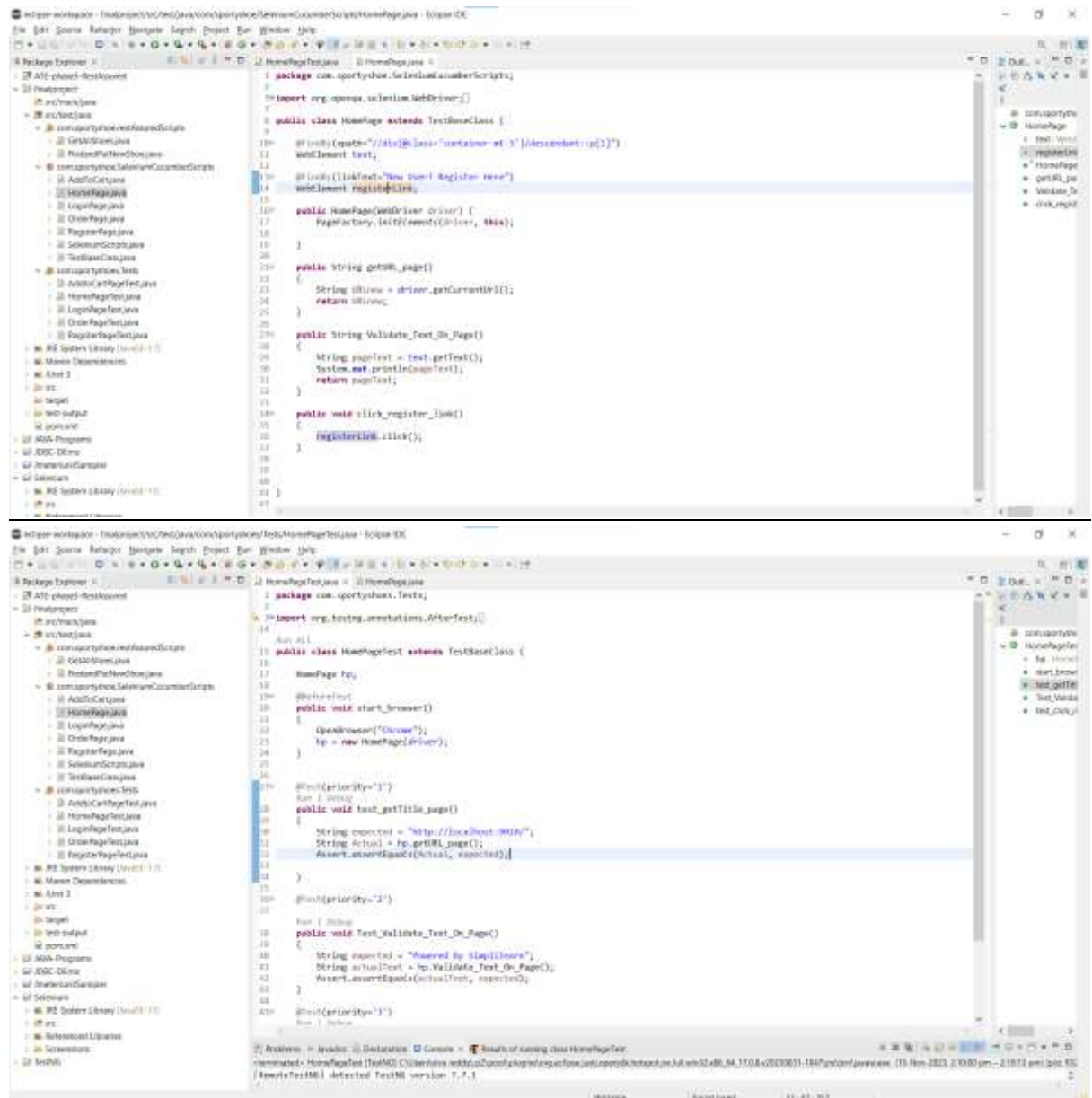
@Test(priority='3')
public void test_click_register_link() throws InterruptedException
{
    Thread.sleep(1500);
    hp.click_register_link();
}
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

OUTPUTS:



The image displays two screenshots of the Eclipse IDE, showing the implementation of Selenium tests for a web application.

Top Screenshot: The main editor shows the `HomePageTest.java` file. The code defines a `HomePage` class that extends `WebDriver` and implements methods for navigating to the home page, clicking the register button, and validating the page content. The `HomePageTest` class extends `TestBase` and implements the `testHomePage` method, which uses `WebDriver` to interact with the application.

```
package com.aporttychou.seleniumwebdriver;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class HomePageTest extends TestBase {

    @Test(priority=1)
    public void testHomePage() {
        // Navigate to the home page
        WebDriver driver = new ChromeDriver();
        driver.get("http://localhost:8080/");

        // Click the register button
        WebElement registerButton = driver.findElement(By.id("register"));
        registerButton.click();

        // Validate the page content
        String pageText = driver.getPageSource();
        Assert.assertEquals("http://localhost:8080/", pageText);
    }
}
```

Bottom Screenshot: The main editor shows the `HomePageTest.java` file. The code defines a `HomePageTest` class that extends `TestBase` and implements the `testHomePage` method, which uses `WebDriver` to interact with the application. The `HomePageTest` class also includes a `testHomePageTest` method that uses `WebDriver` to interact with the application.

```
package com.aporttychou.seleniumwebdriver;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class HomePageTest extends TestBase {

    @Test(priority=1)
    public void testHomePage() {
        // Navigate to the home page
        WebDriver driver = new ChromeDriver();
        driver.get("http://localhost:8080/");

        // Click the register button
        WebElement registerButton = driver.findElement(By.id("register"));
        registerButton.click();

        // Validate the page content
        String pageText = driver.getPageSource();
        Assert.assertEquals("http://localhost:8080/", pageText);
    }

    @Test(priority=2)
    public void testHomePageTest() {
        // Navigate to the home page
        WebDriver driver = new ChromeDriver();
        driver.get("http://localhost:8080/");

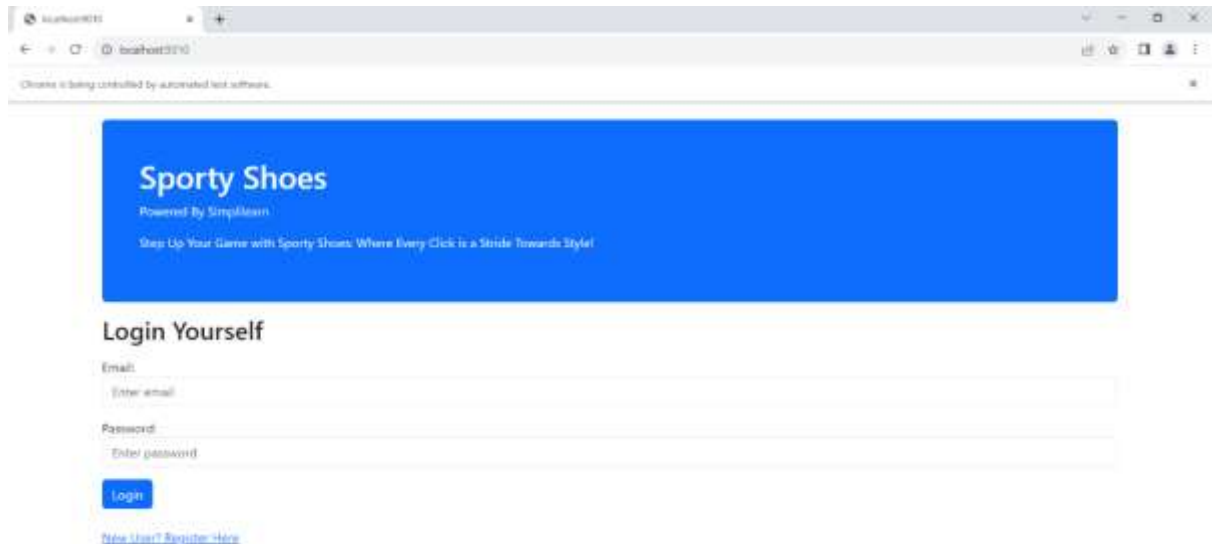
        // Click the register button
        WebElement registerButton = driver.findElement(By.id("register"));
        registerButton.click();

        // Validate the page content
        String pageText = driver.getPageSource();
        Assert.assertEquals("http://localhost:8080/", pageText);
    }
}
```

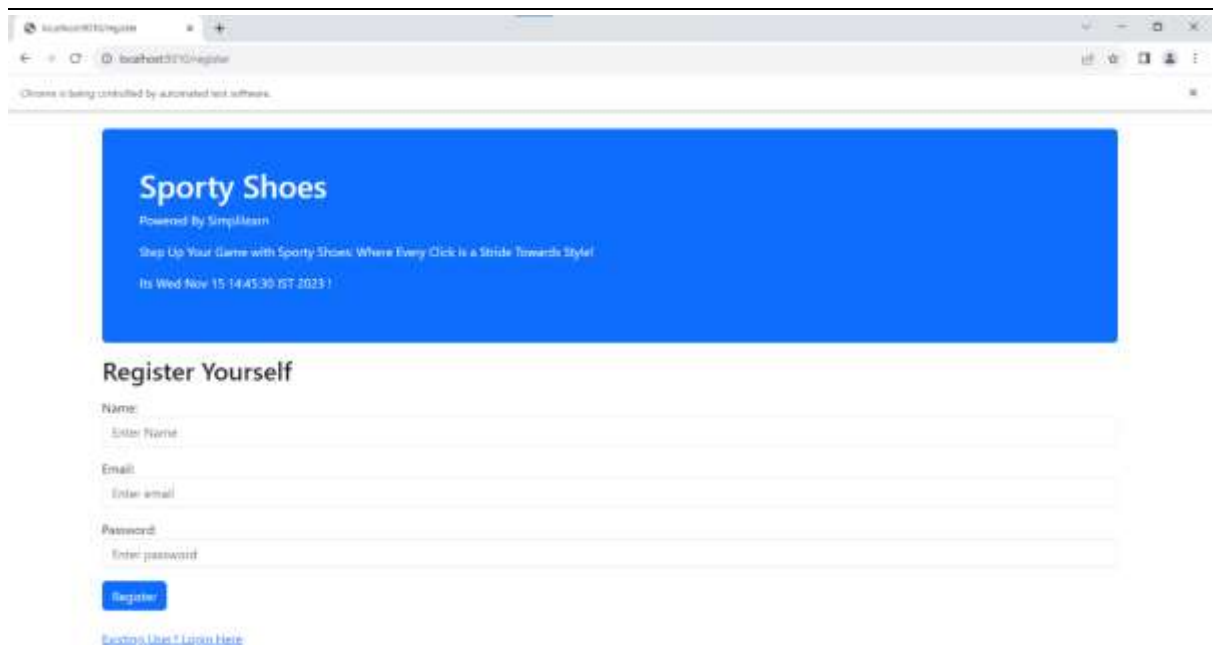

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



A screenshot of a web browser displaying the 'Sporty Shoes' login page. The browser's address bar shows 'localhost:3000'. The page has a blue header with the text 'Sporty Shoes', 'Powered By Simplilearn', and 'Step Up Your Game with Sporty Shoes. Where Every Click is a Stride Towards Style!'. Below the header, the section is titled 'Login Yourself'. It contains two input fields: 'Email' with placeholder text 'Enter email' and 'Password' with placeholder text 'Enter password'. A blue 'Login' button is positioned below the password field. At the bottom of the login section, there is a link that reads 'New User? Register Here'.

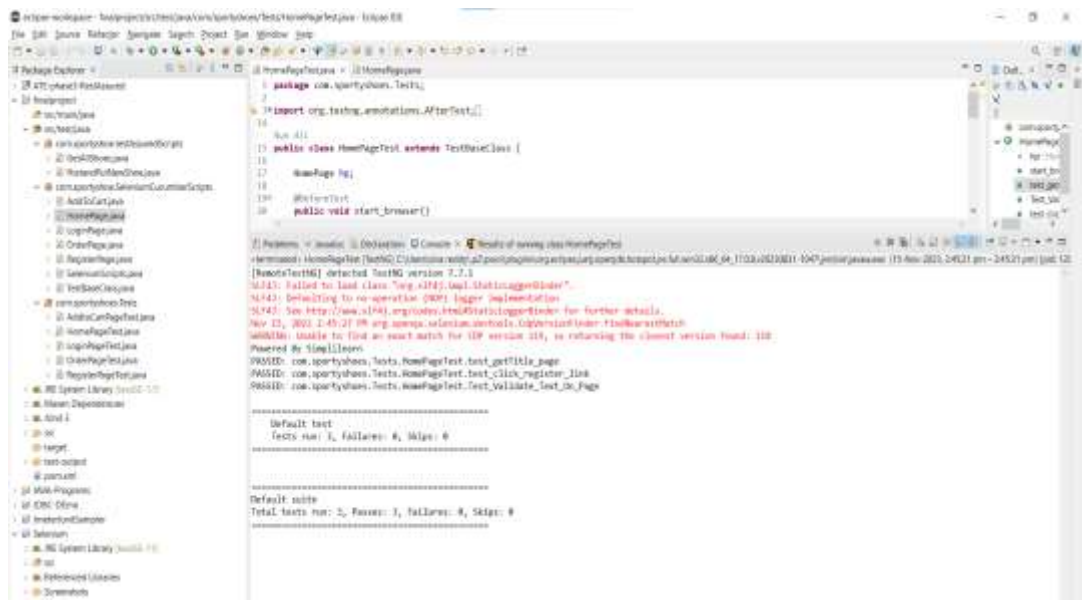


A screenshot of a web browser displaying the 'Sporty Shoes' registration page. The browser's address bar shows 'localhost:3000/register'. The page has a blue header with the text 'Sporty Shoes', 'Powered By Simplilearn', and 'Step Up Your Game with Sporty Shoes. Where Every Click is a Stride Towards Style!'. Below the header, the section is titled 'Register Yourself'. It contains three input fields: 'Name' with placeholder text 'Enter Name', 'Email' with placeholder text 'Enter email', and 'Password' with placeholder text 'Enter password'. A blue 'Register' button is positioned below the password field. At the bottom of the register section, there is a link that reads 'Existing User? Login Here'.

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



2. LOGIN PAGE:

Code for Login 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 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) {
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
        PageFactory.initElements(driver, this);
    }

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

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

Code for login page 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.TestBaseClass;

public class LoginPageTest extends TestBaseClass {

    HomePage hp;
    RegisterPage rp;
    LoginPage lp;

    @BeforeTest
    public void start_browser()
    {
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

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

```
@Test(priority='4')

public void test_click_cart()
{
    lp.click_cart();
}
```

```
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

OUTPUTS:

```
package com.sportyshoe.seleniumwebdriverScripts;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

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("shiv@gmail.com");
        loginPassword.sendKeys("shiv@227");
        loginBtn.click();
    }

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

```
package com.sportyshoe.seleniumwebdriverScripts.Tests;

import org.junit.Assert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class LoginPageTest extends TestBase {

    @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 start_browser() {
        WebDriver driver = new ChromeDriver();
        driver.get("http://localhost:8080/login");
        driver.manage().window().maximize();
    }

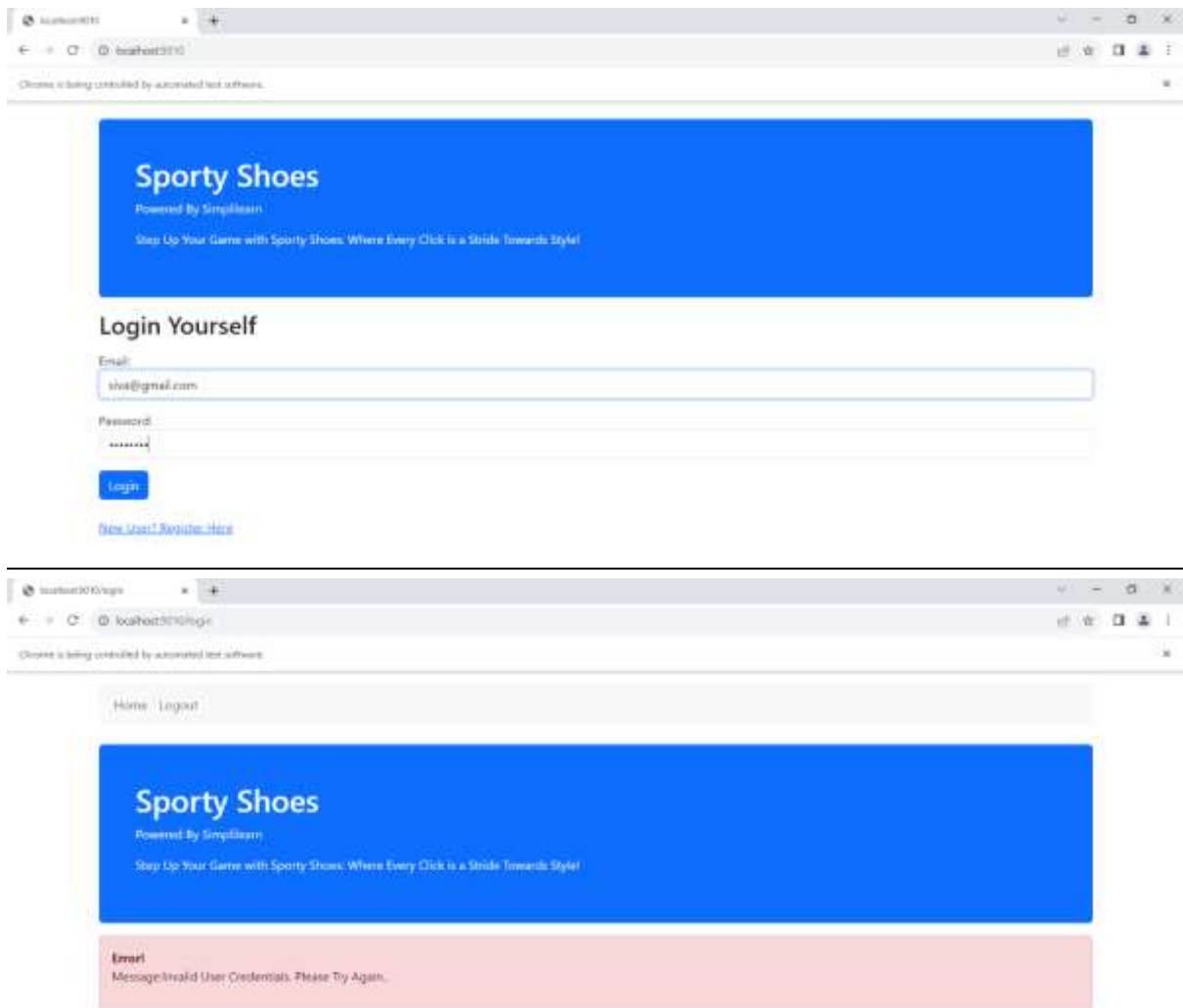
    public void test_login() {
        LoginPage lp = new LoginPage(driver);
        lp.user_login();
    }

    public void test_getTitle_page() {
        String expected = "http://localhost:8080/login";
        String actual = lp.getTitle_page();
        Assert.assertEquals(actual, expected);
    }
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



3. Register Page:

Code for 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 TestBaseClass{
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@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("siva");  
    registeremail.sendKeys("siva@gmail.com");  
    registerpassword.sendKeys("siva@123");  
    registerBtn.click();  
}
```

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

```
public String validate_registration_Text()  
{
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

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

}
```

Code for Register Page 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.TestBaseClass;

public class RegisterPageTest extends TestBaseClass {

    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);
```


CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
        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')

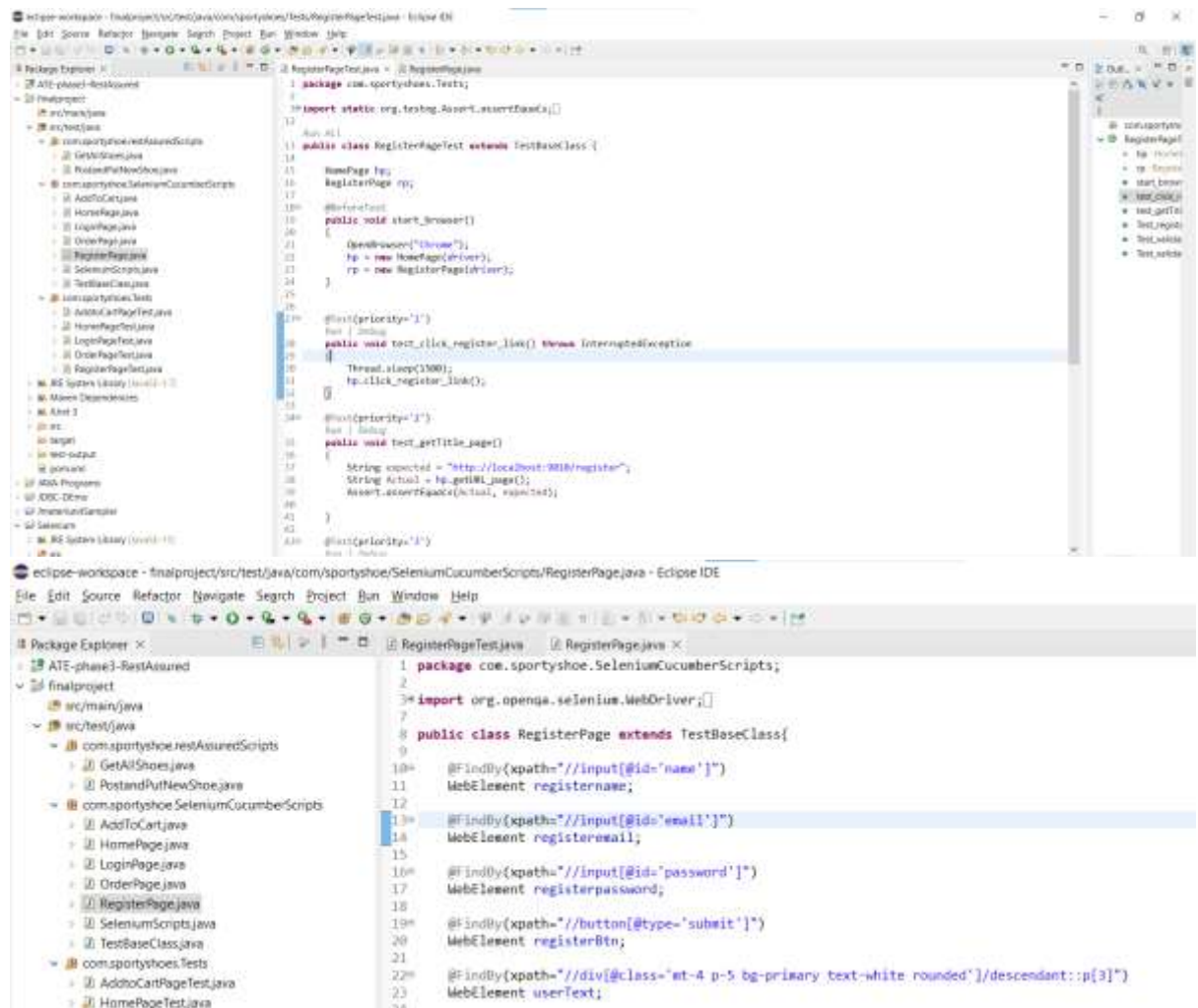
    public void Test_validate_registration_Text()
    {
        String expected = "Hello siva !";
        String actualText = rp.validate_registration_Text();
        Assert.assertEquals(actualText, expected);
    }
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

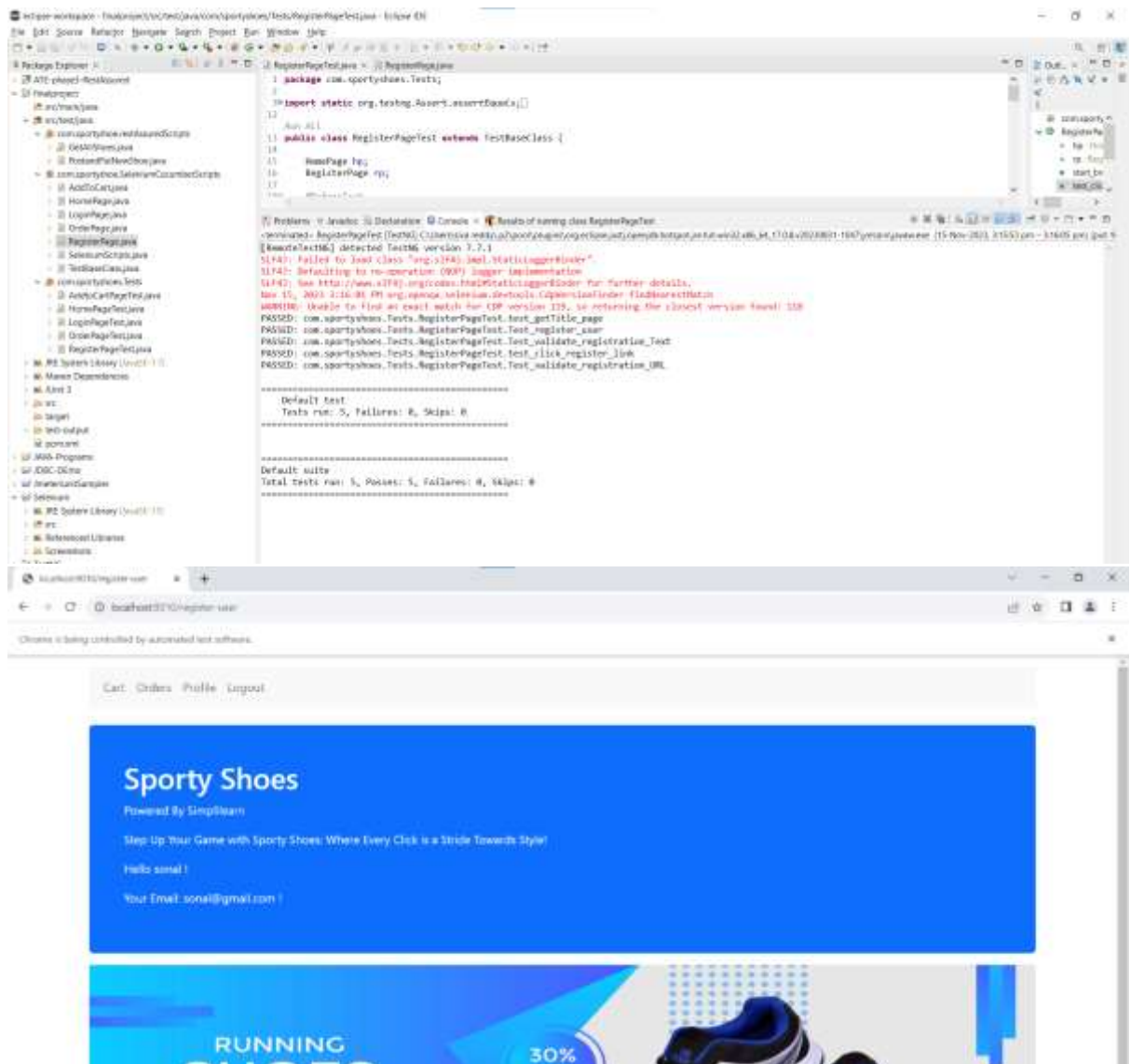
OUTPUTS:



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



4. Add To Cart

Code for add to cart.

```
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 AddToCart {
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@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 AddToCart(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;

}
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Code for add to cart test.

```
package com.sportyshoes.Tests;

import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportyshoe.SeleniumCucumberScripts.AddToCart;
import com.sportyshoe.SeleniumCucumberScripts.HomePage;
import com.sportyshoe.SeleniumCucumberScripts.LoginPage;
import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;
import com.sportyshoe.SeleniumCucumberScripts.TestBaseClass;

public class AddtoCartPageTest extends TestBaseClass {

    HomePage hp;
    RegisterPage rp;
    LoginPage lp;
    AddToCart ac;

    @BeforeTest
    public void start_browser()
    {
        OpenBrowser("Chrome");
        hp = new HomePage(driver);
        rp = new RegisterPage(driver);
        lp = new LoginPage(driver);
        ac = new AddToCart(driver);
    }

    @Test(priority='1')
    public void test_login()
    {
        lp.user_login();
    }
}
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@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);
}

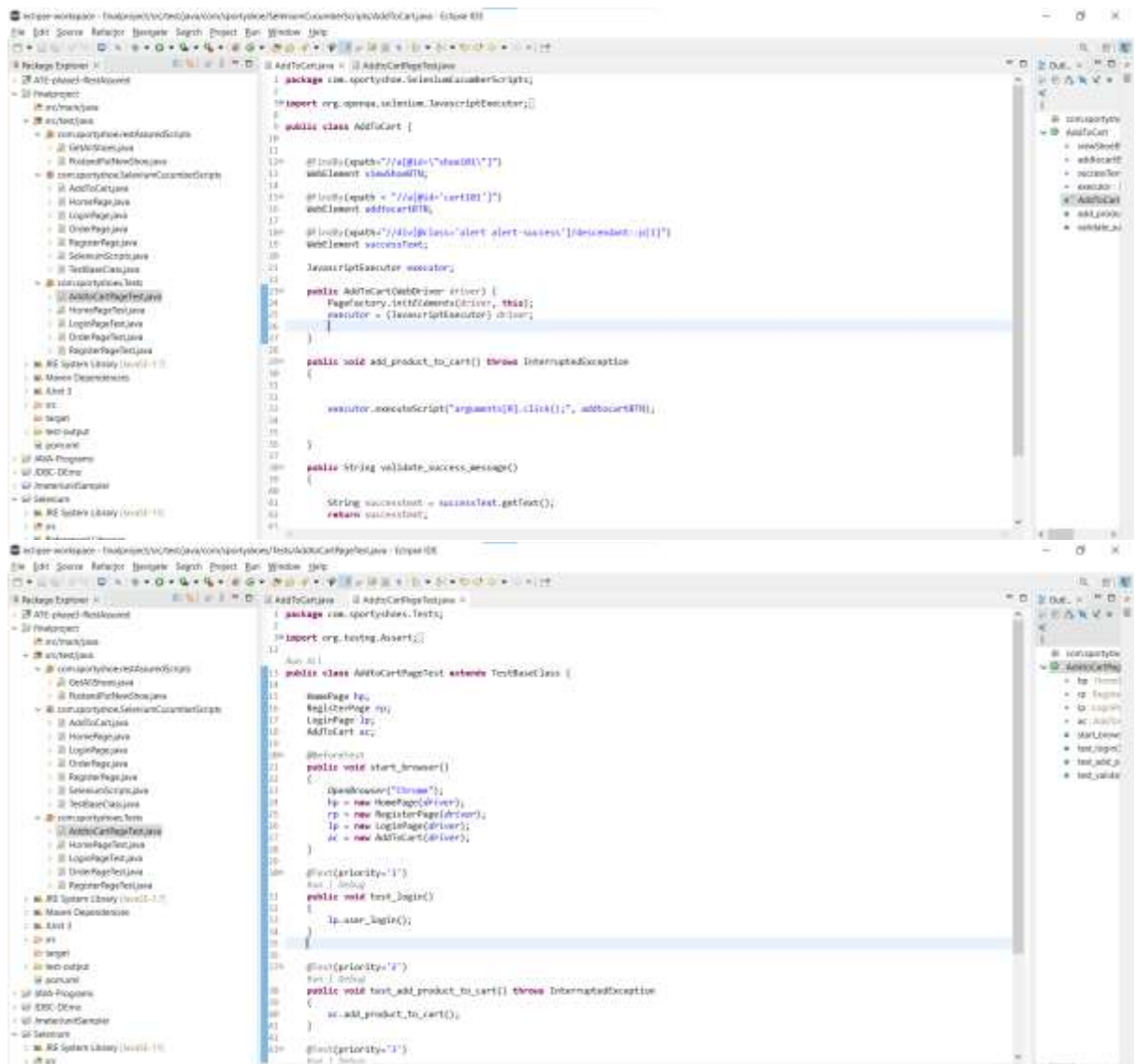
}
```

OUTPUTS:

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



5. ORDER PAGE:

Code for order

```
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 {
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@FindBy(linkText="Orders")
WebElement orderlink;
```

```
public OrderPage(WebDriver driver) {
    PageFactory.initElements(driver, this);
}
```

```
public void click_orderPage()
{
    orderlink.click();
}
```

```
}
```

Code for order 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.OrderPage;
import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;
import com.sportyshoe.SeleniumCucumberScripts.TestBaseClass;
```

```
public class OrderPageTest extends TestBaseClass {
```

```
    HomePage hp;
    RegisterPage rp;
    LoginPage lp;
    OrderPage op;
```

```
@BeforeTest
public void start_browser()
```


CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
{
    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();
}

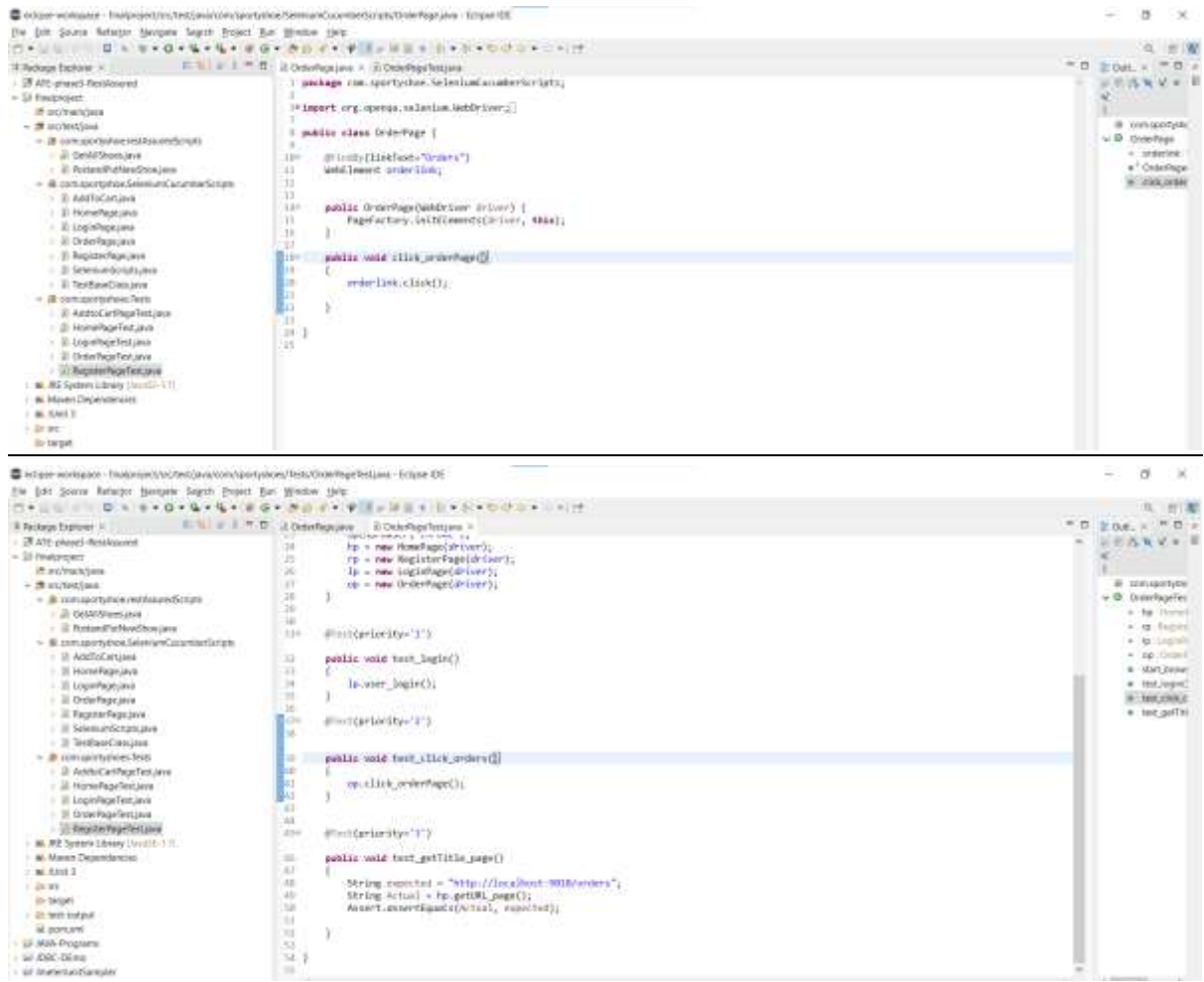
@Test(priority='3')
public void test_getTitle_page()
{
    String expected = "http://localhost:9010/orders";
    String Actual = hp.getURL_page();
    Assert.assertEquals(Actual, expected);
}
}
```

OUTPUTS:

CAPSTONE PROJECT

NAME: K SIVA REDDY

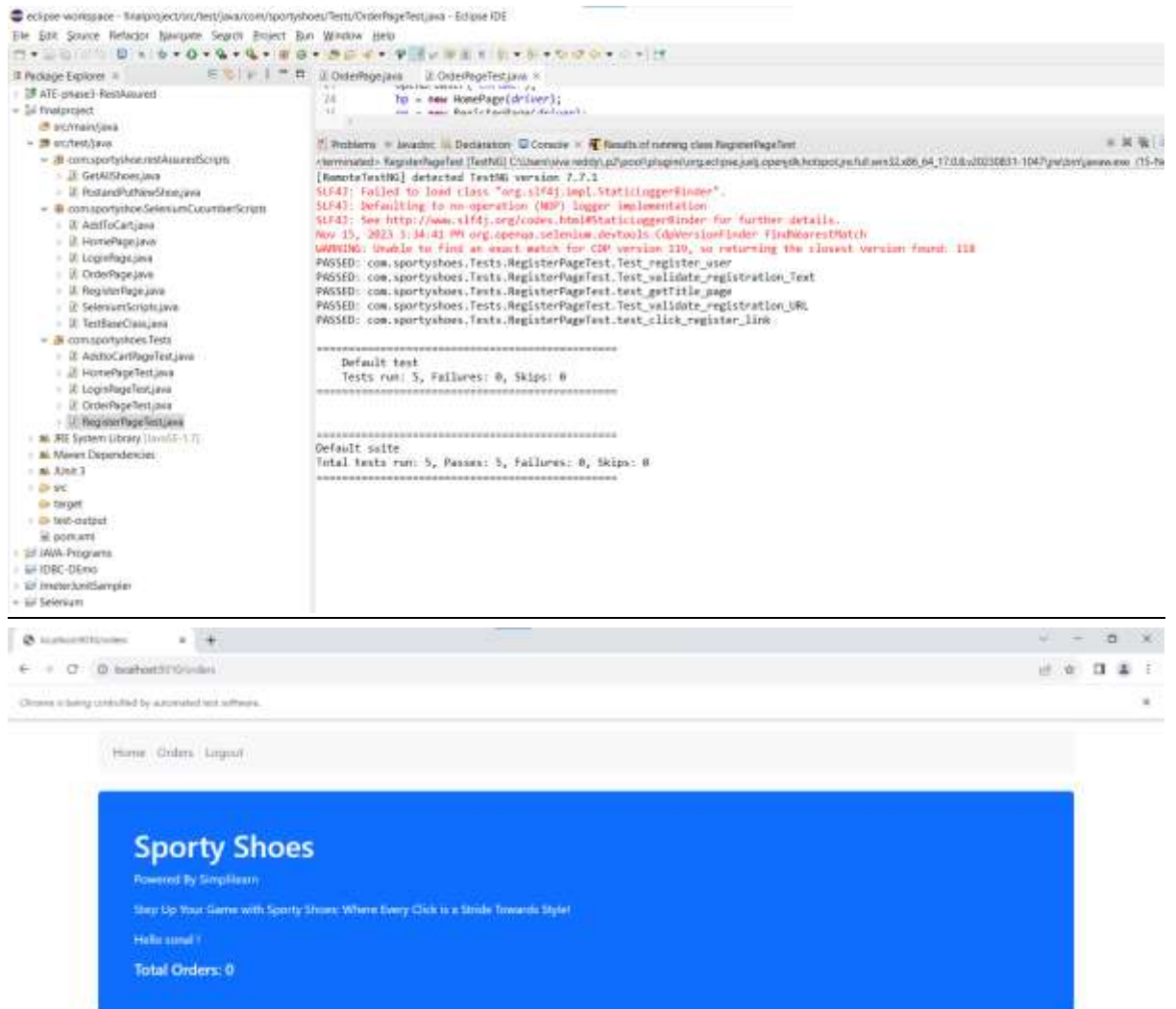
ID: 2576232



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Automate the below API endpoints using Rest-Assured

Now we retrieve all products and all registered users in rest assured.

Code

```
package com.sportyshoe.restAssuredScripts;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.hasItems;

import org.json.JSONObject;
import org.testng.annotations.Test;

import io.restassured.RestAssured;
import io.restassured.http.ContentType;

public class GetAllShoes {

    @Test (priority='1')

    public void get_all_shoes()
    {

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

    }

    @Test(priority='3')

    public void get_all_users()
    {

        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/get-users")
            .when()
            .get()
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

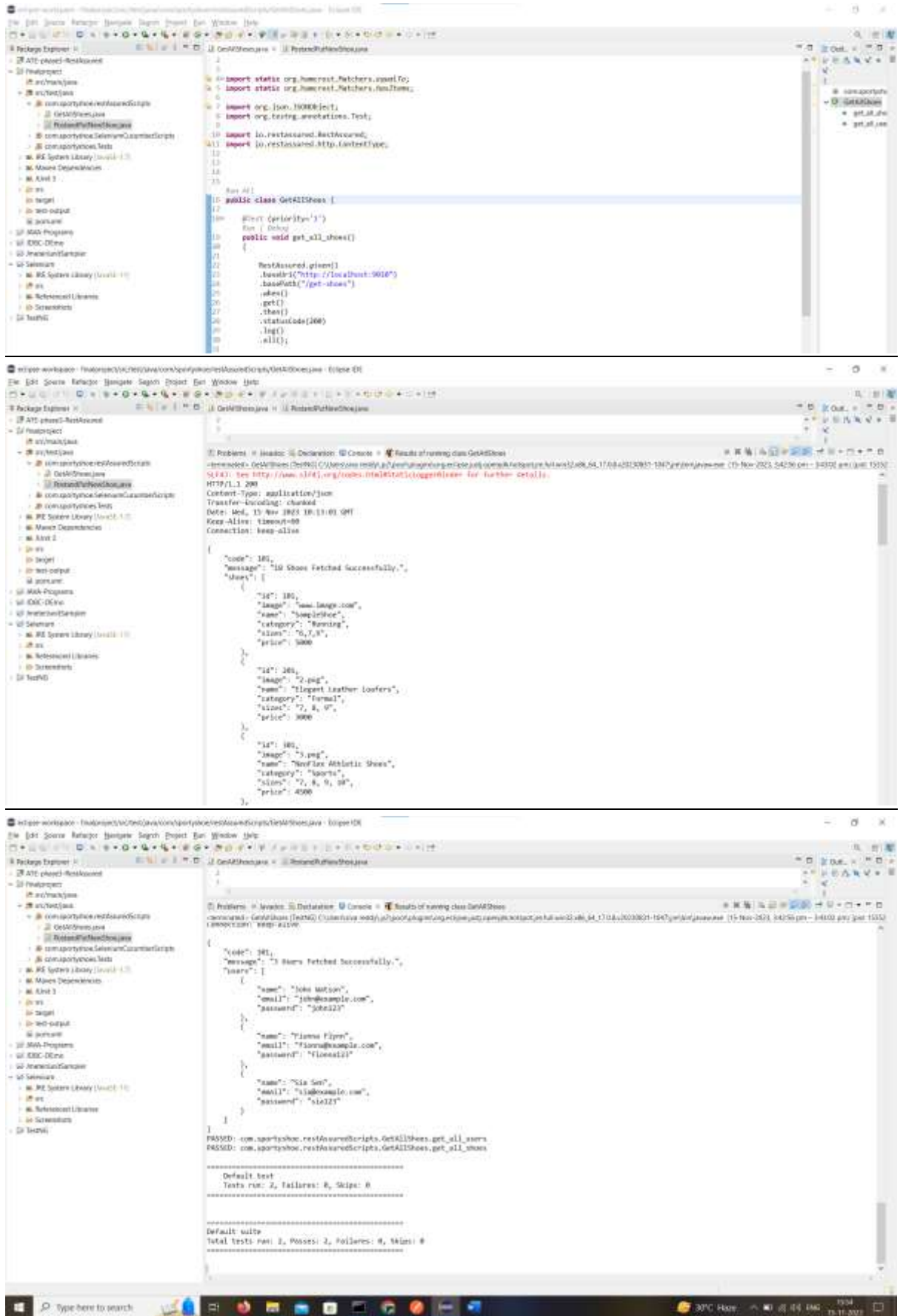
```
.then()  
.statusCode(200)  
.log()  
.all();  
}  
}
```

OUTPUTS:

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

Code for add delete product.

```
package com.sportyshoe.restAssuredScripts;
```

```
import static org.hamcrest.Matchers.equalTo;
```

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

```
import io.restassured.RestAssured;
```

```
public class PostandPutNewShoe {
```

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

```
    public void add_new_product()
```

```
    {
```

```
        RestAssured.given()
```

```
        .baseUrl("http://localhost:9010")
```

```
        .basePath("/add-shoe")
```

```
        .queryParams("id","1020")
```

```
        .queryParams("image", "www.imge.com")
```

```
        .queryParams("name","Nike")
```

```
        .queryParams("category", "Running")
```

```
        .queryParams("sizes","5,6,7")
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
.queryParams("price", "2000")  
.when()  
.post()  
.then()  
.log().all();  
  
}
```

```
@Test(priority='2')  
public void update_a_product()  
{
```

```
    RestAssured.given()  
        .baseUrl("http://localhost:9010")  
        .basePath("/update-shoe")  
        .queryParams("id", "1020")  
        .queryParams("image", "www.imge123.com")  
        .queryParams("name", "Reebok")  
        .queryParams("category", "Running")  
        .queryParams("sizes", "5,6,7")  
        .queryParams("price", "2500")  
        .when()  
        .put()  
        .then()  
        .log().all();
```


CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
}
```

```
@Test(priority='3')
```

```
public void delete_product()
```

```
{
```

```
    RestAssured.given()
```

```
        .baseUrl("http://localhost:9010")
```

```
        .basePath("/delete-shoe")
```

```
        .queryParams("id", "1020")
```

```
        .when().delete()
```

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

```
}
```

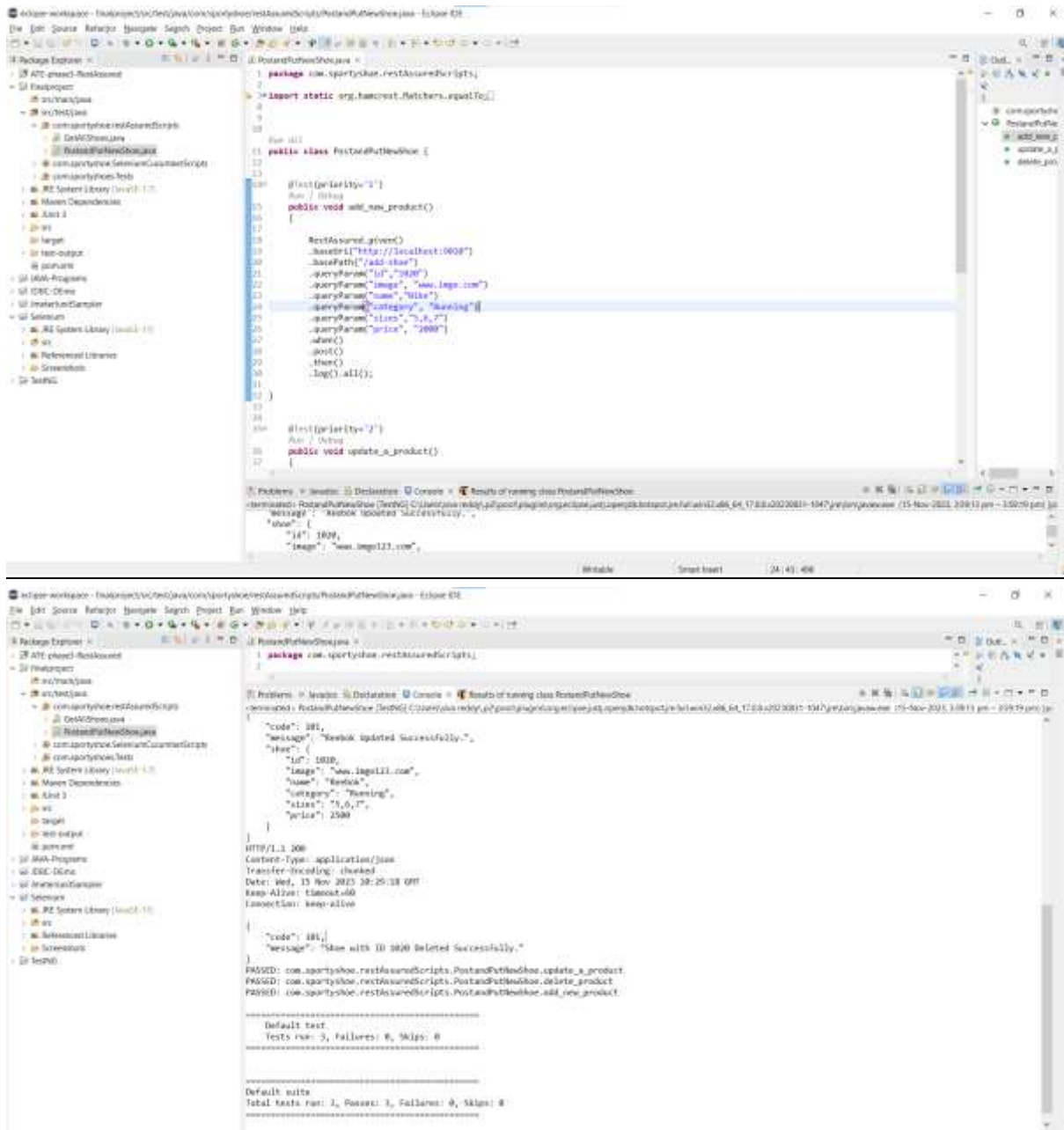
```
}
```

OUTPUT:

CAPSTONE PROJECT

NAME: K SIVA REDDY

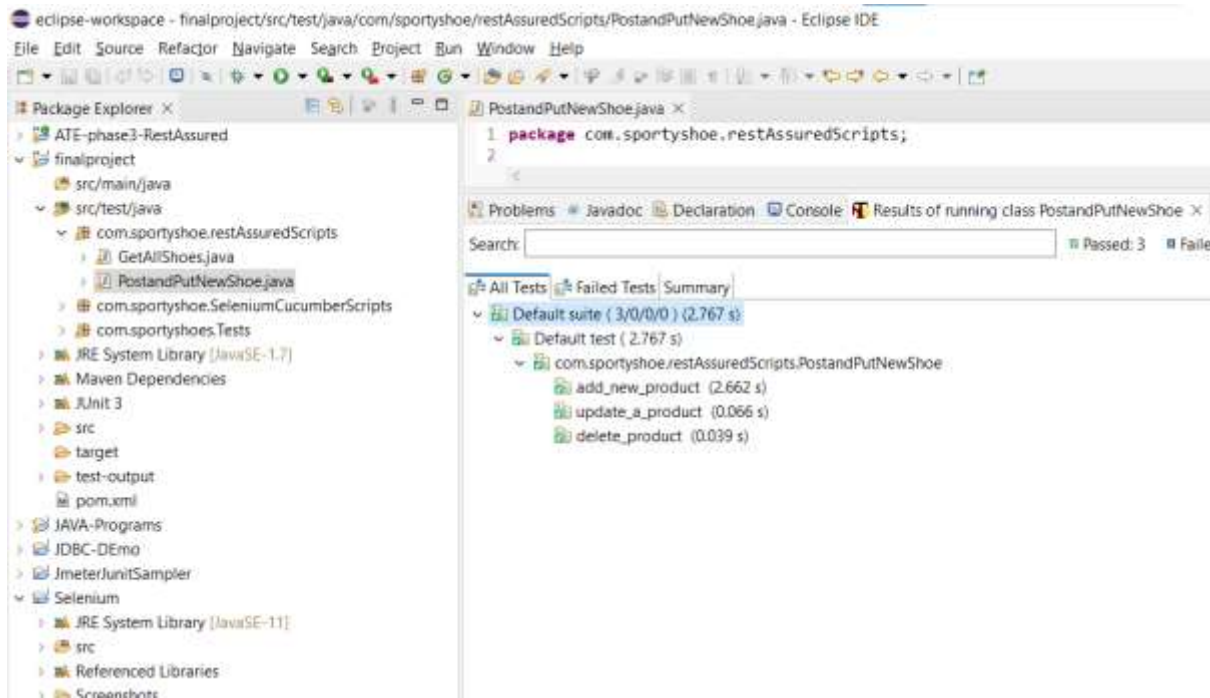
ID: 2576232



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



Now we can run the feature file and step file in cucumber.

Code for feature file

Feature: Implement The Lesson End Project1.

Scenario: Rest API testing on reqres.in Automate the below API endpoints using Rest-Assured

Given User get Retrieve the list of all products in the store.

Given User Retrieve the list of all registered users.

Then User Add the product.

When User Delete the product.

And User Update the product.

Code for Step file

```
package steps;
```

```
import org.json.JSONObject;
```

```
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;
```

```
import io.restassured.http.ContentType;
```

```
public class Final {
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

```
@Given("User get Retrieve the list of all products in the store.")
public void user_get_retrieve_the_list_of_all_products_in_the_store() {

    JSONObject body = new JSONObject();
    body.put("name", "Siva");
    body.put("job", "Tester");

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

@Given("User Retrieve the list of all registered users.")
public void user_retrieve_the_list_of_all_registered_users() {

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

@Then("User Add the product.")
public void user_add_the_product() {
    RestAssured.given()
        .baseUrl("http://localhost:9010/add-shoe?id=10001&image=www.image.com&name=SampleShoe&category=Running&sizes=12&price=1500")
        .when().get()
        .then().statusCode(405);
}

@When("User Delete the product.")
public void user_delete_the_product() {

    RestAssured.given()
        .baseUrl("http://localhost:9010")
        .basePath("/delete-shoe?id=10001")
        .when().get()
        .then().statusCode(405);
}

@And("User Update the product.")
public void user_update_the_product() {

    RestAssured.given()
```

CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232

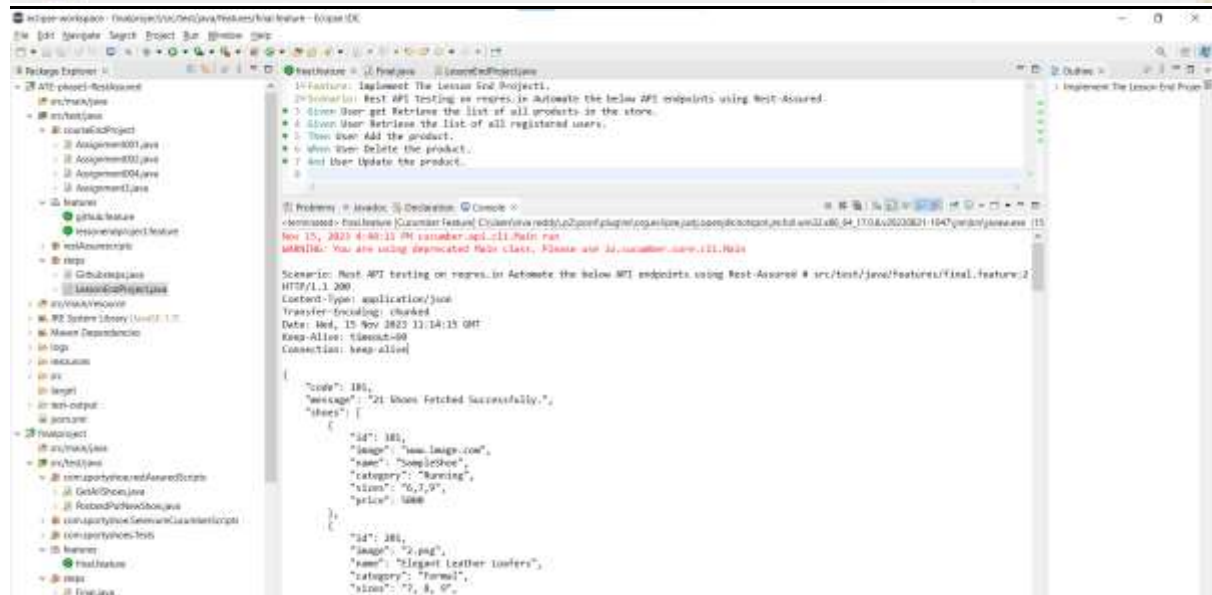
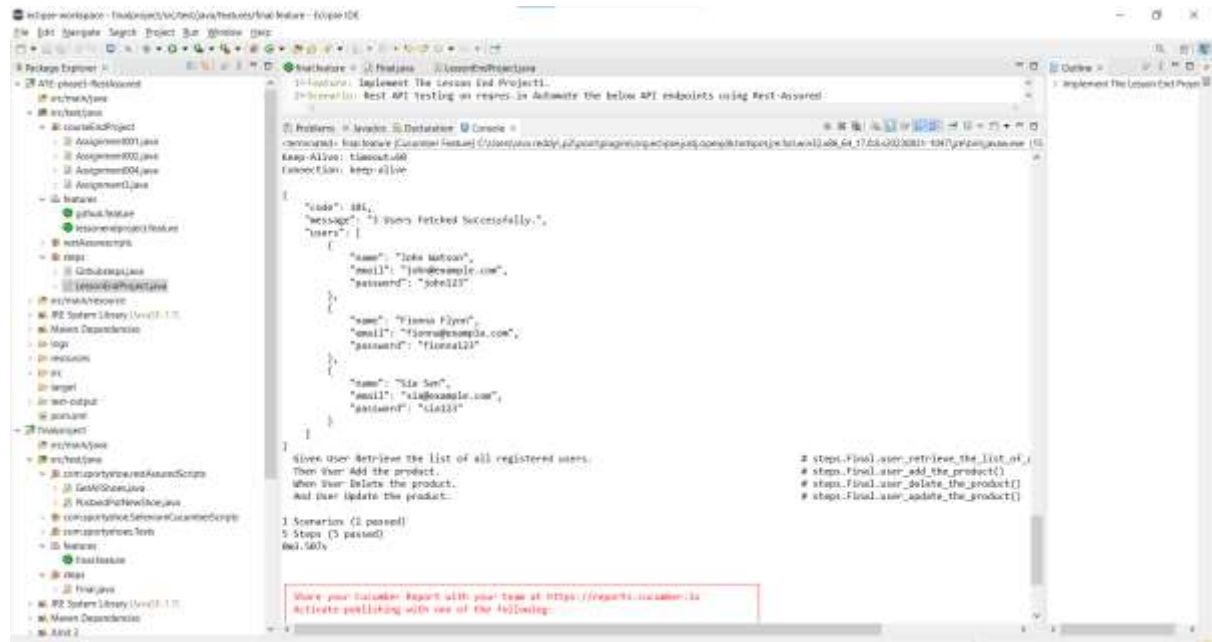
```
.baseUri("http://localhost:9010/add-shoe?id=10001&image=www.image.com&name=SampleShoe&category=Running&sizes=12&price=1500")
```

```
.when().get()  
.then().statusCode(405);
```

```
}
```

```
}
```

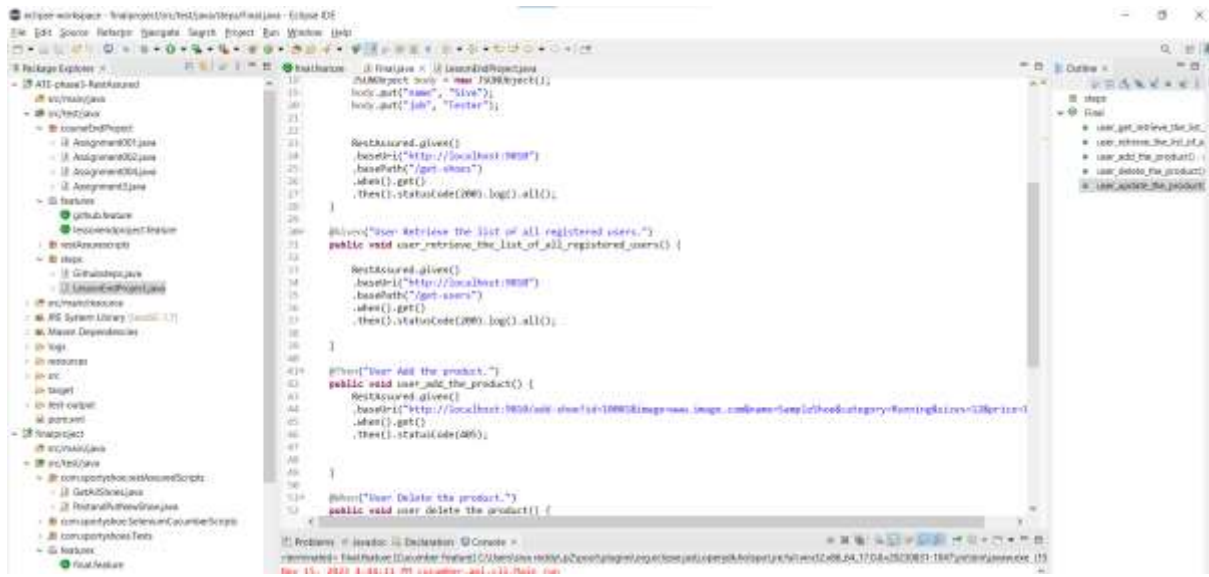
OUTPUTS:



CAPSTONE PROJECT

NAME: K SIVA REDDY

ID: 2576232



This is my GitHub link to upload my project

<https://github.com/sivaji143843/Capstone-Project>