

# Automation Testing Capstone Project

## Create a Testing Framework for Sporty Shoes Website

### Name : Rishav Rajak

#### Problem Statement and Motivation Real-Time Scenario:

Sporty Shoes has an e-commerce website that has the following existing features in place: I used <https://neemans.com/> for the task.

- Users can view products , I wrote some functionality for that also .
- If users want to purchase something, they can first sign up and then log in
- Users can add multiple items to their cart and do a checkout.
- Users have a dashboard that lets them edit their profile, view past purchases, and view their cart
- Once users do a checkout, the items are cleared from their cart and an order is generated which is stored in their order history The above application is already functional. What is needed now is to add a testing layer that will ensure that everything is passed through QA.

#### Task:

The following tasks outline the testing framework creation:

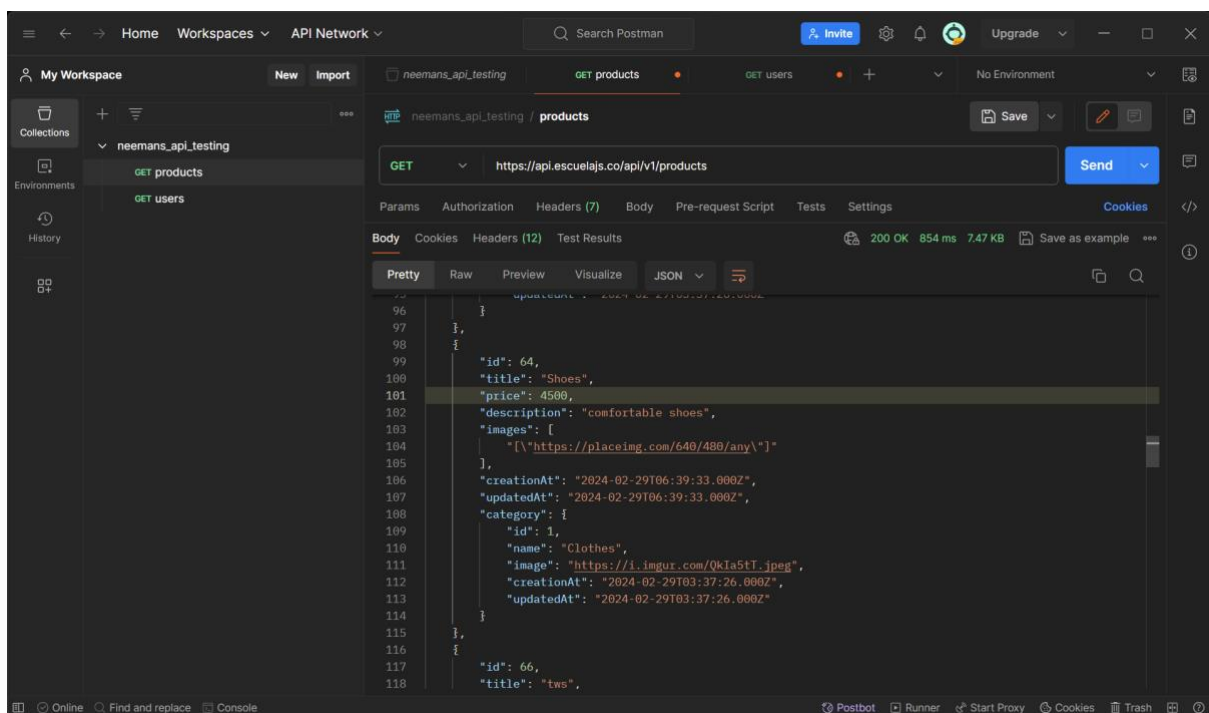
1. Add additional code to the original project to add a REST API module. This module will have two API endpoints:
  - Retrieve the list of all products in the store
  - Retrieve the list of all registered users
2. Create Selenium scripts to test all the pages in the web app
3. Do unit testing for all the backend classes and methods
4. Create JMeter scripts to do load testing of the homepage and the product detail page

How to execute :

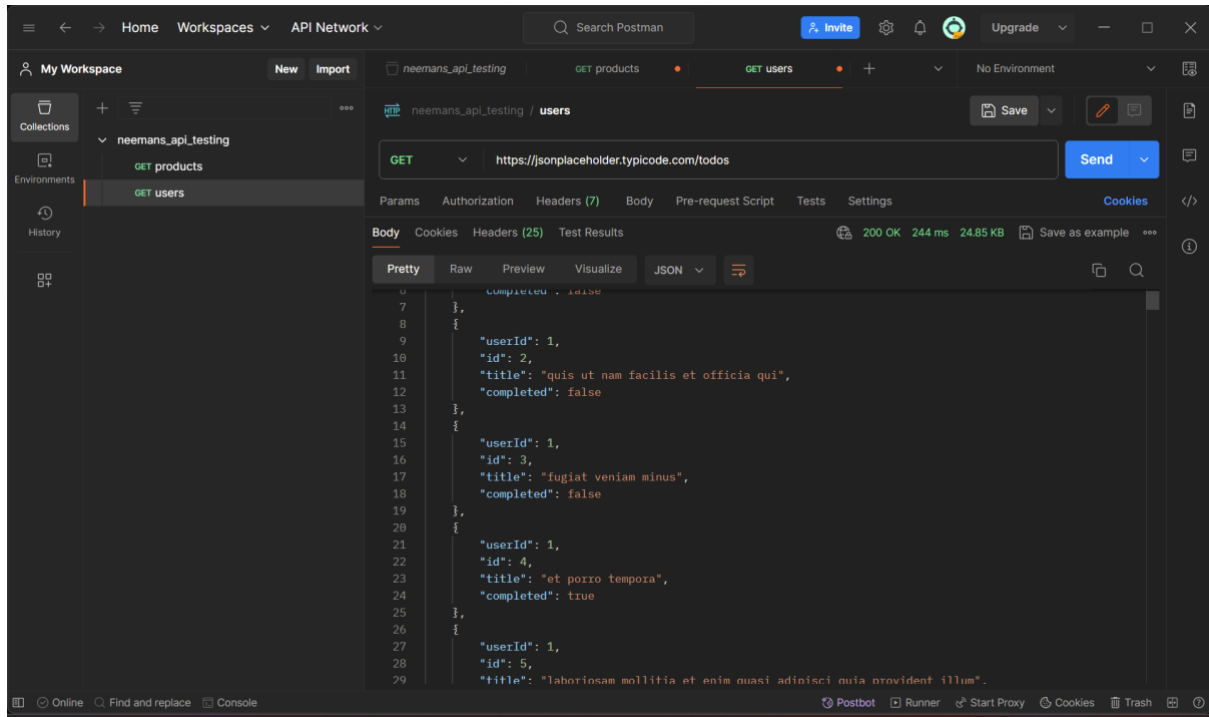
1. Import the project in eclipse IDE
2. Identify testNG.xml
3. Run testNG.xml
4. Open the jmeter file provide there and run it .

## Task 1 Screenshots :

For products :

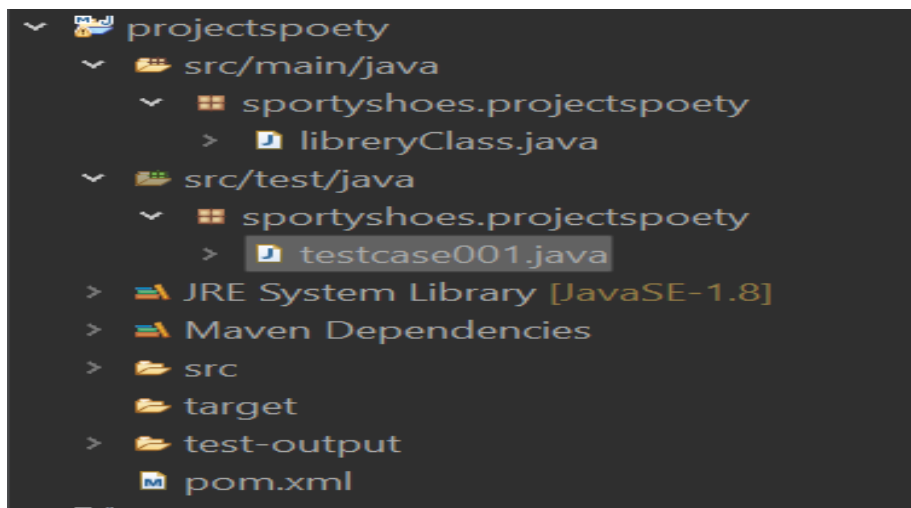


For users :



## Task 2 & 3 Screenshots :

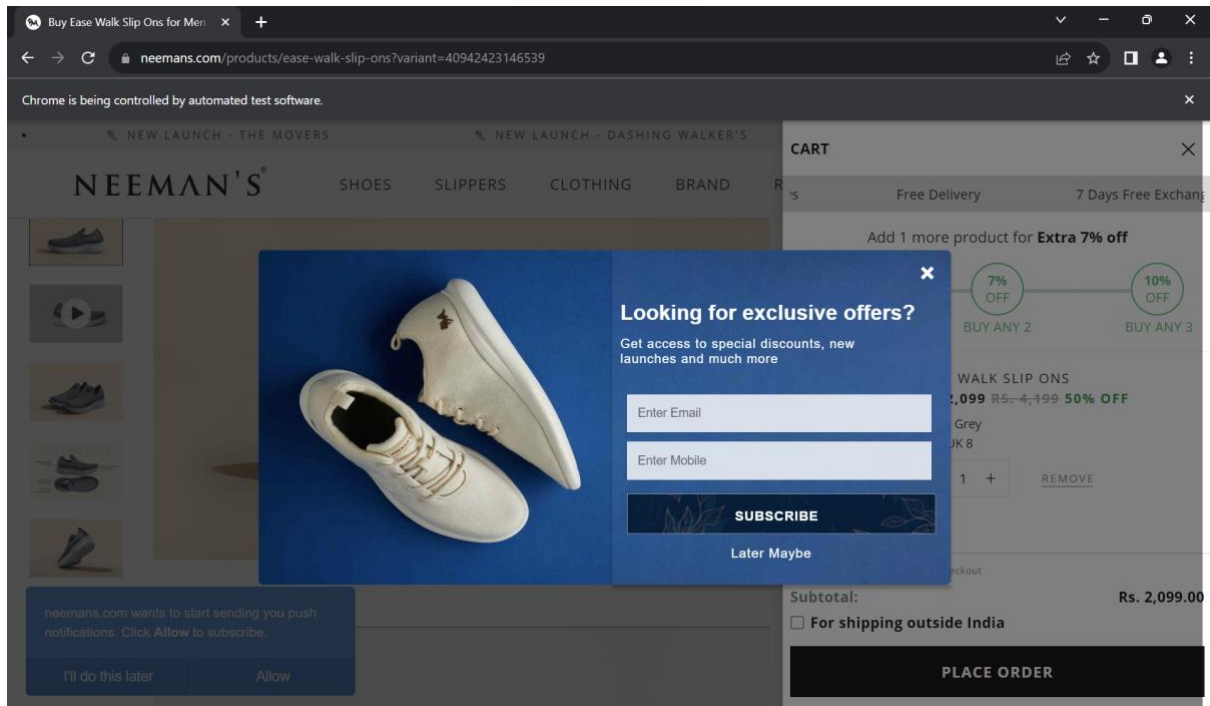
### Folder structure :



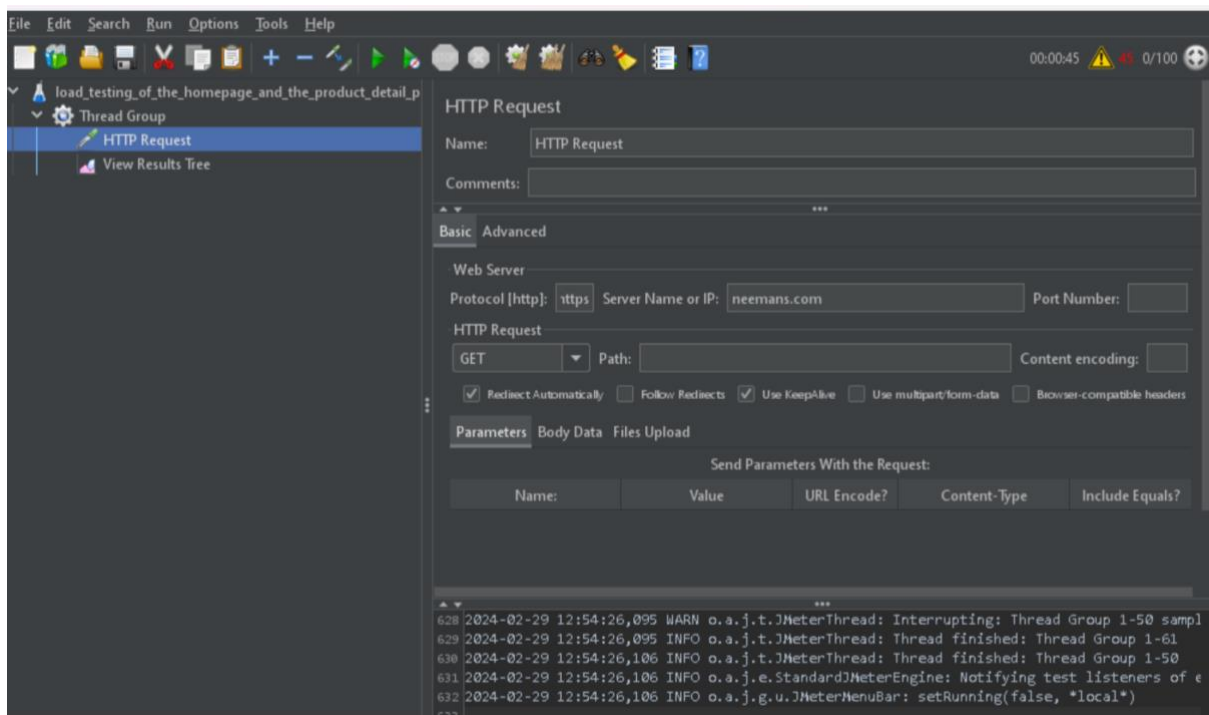
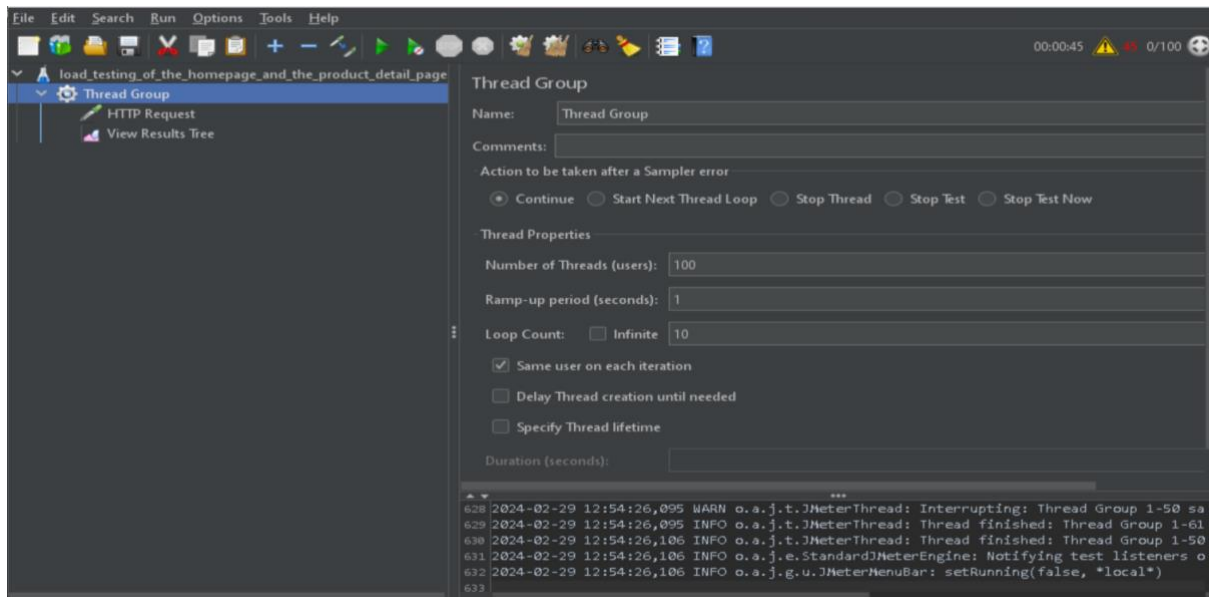
### Moduler code :

```
libraryClass.java × testcase001.java
1 package sportyshoes.projectspoety;
2
3 import java.util.NoSuchElementException;
4
5 import org.junit.Assert;
6 import org.openqa.selenium.By;
7 import org.openqa.selenium.WebDriver;
8
9
10
11
12 public class libraryClass {
13     WebDriver driver;
14     public void InitDriver(WebDriver driver) {
15         this.driver = driver;
16     }
17
18     public void launchApp(String url) {
19         driver.get(url);
20         System.out.println("The app is launched successfully");
21     }
22     public void TypeSomething(String selector ,String value) {
23         driver.findElement(By.xpath(selector)).sendKeys(value);
24     }
25     public void clickOn(String selector) {
26         driver.findElement(By.xpath(selector)).click();
27     }
28     public void checkTitle(String expectedTitle) {
29         Assert.assertEquals(driver.getTitle(), expectedTitle, "Page title doesn't match expected title");
30     }
31     public void isdisplaiedClick(String loc) {
32         try {
33             if(driver.findElement(By.xpath(loc)).isDisplayed()) {
34                 driver.findElement(By.xpath(loc)).click();
35                 System.out.println("Element is displayed.");
36             }
37         } catch (NoSuchElementException e) {
38             System.out.println("Element is not displayed.");
39         }
40     }
41 }
42
43 }
44
45
```

The popup :



## Task 4 Screenshots :



Result of the load testing .

File Edit Search Run Options Tools Help

load\_testing\_of\_the\_homepage\_and\_the\_product\_detail\_page

Thread Group

HTTP Request

View Results Tree

### View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes Configure

Search: Case sensitive Regular exp. Search Reset

Text	Sampler result	Request	Response data
HTTP Request	Thread Name: Thread Group 1-10		
HTTP Request	Sample Start: 2024-02-29 12:49:15 IST		
HTTP Request	Load time: 4137		
HTTP Request	Connect Time: 26		
HTTP Request	Latency: 203		
HTTP Request	Size in bytes: 4827684		
HTTP Request	Sent bytes: 116		
HTTP Request	Headers size in bytes: 2917		
HTTP Request	Body size in bytes: 4824767		
HTTP Request	Sample Count: 1		
HTTP Request	Error Count: 0		
HTTP Request	Data type ("text/bin"): text		
HTTP Request	Response code: 200		
HTTP Request	Response message: OK		
HTTP Request	HTTPSampleResult fields:		
HTTP Request	ContentType: text/html; charset=utf-8		
HTTP Request	DataEncoding: utf-8		

627 2024-02-29 12:54:26,095 INFO o.a.j.t.JMeterThread: Stopping: Thread Group 1-50

628 2024-02-29 12:54:26,095 WARN o.a.j.t.JMeterThread: Interrupting: Thread Group 1-50 sampler: HTTP Request

629 2024-02-29 12:54:26,095 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-61

630 2024-02-29 12:54:26,106 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-50

631 2024-02-29 12:54:26,106 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test

12:57 29-02-2024