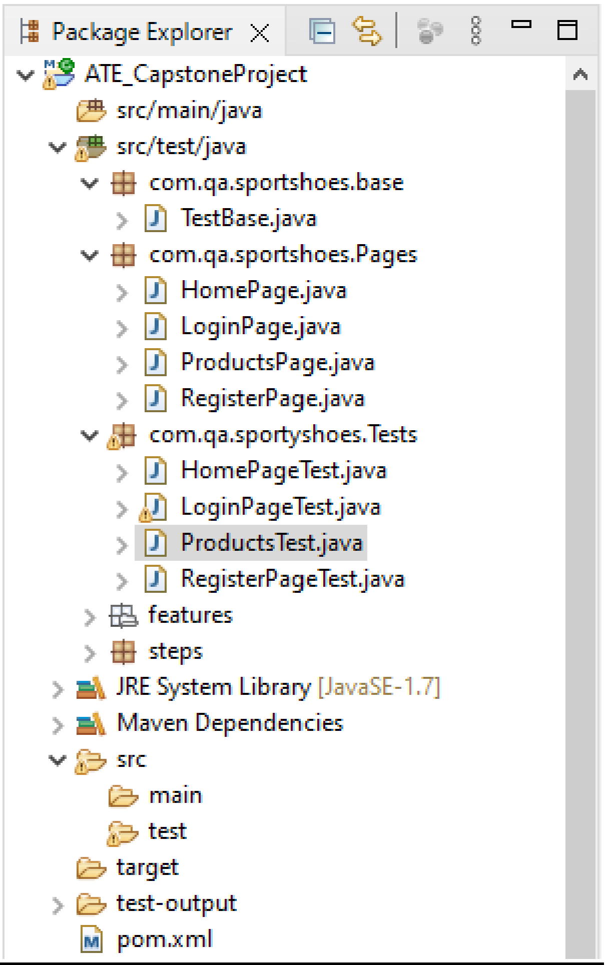
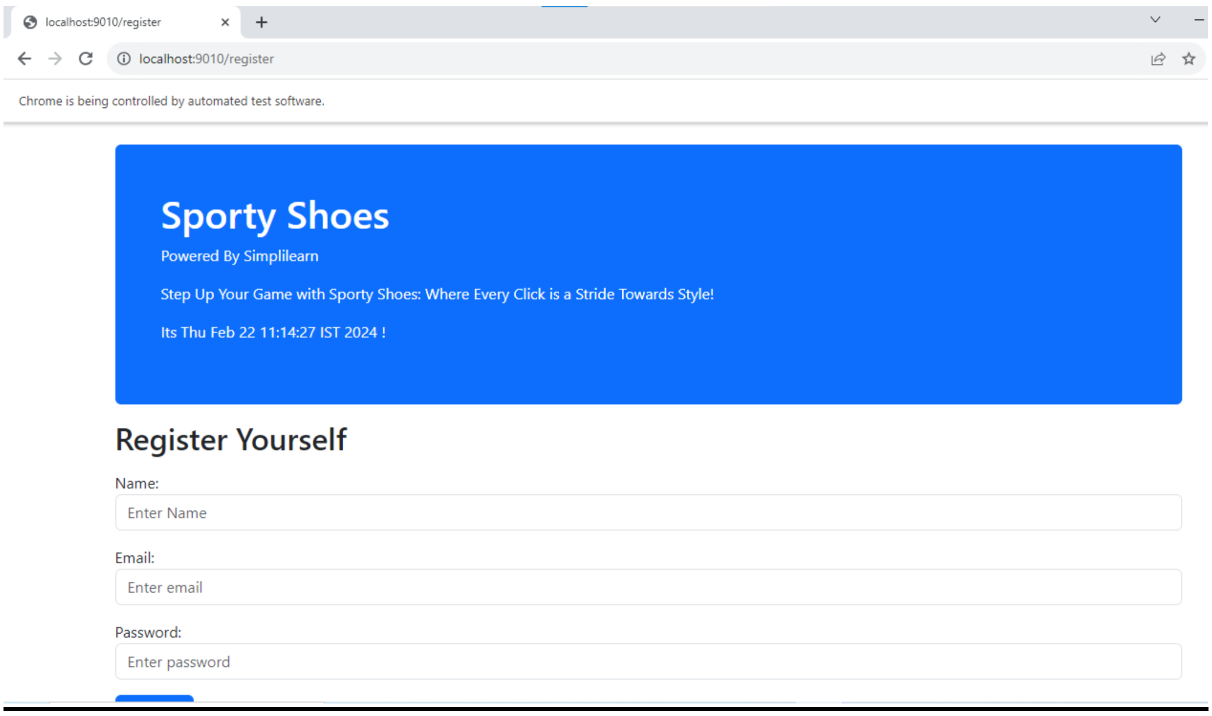
ScreenShots

Capstone Project: Create a Testing Framework for Sporty Shoes Website

Selenium WebDriver with TestNG Framework



```
Package Explorer
ATE_CapstoneProject
    src/main/java
    src/test/java
        com.qa.sportshoes.base
            TestBase.java
        com.qa.sportshoes.Pages
            HomePage.java
            LoginPage.java
            ProductsPage.java
            RegisterPage.java
        com.qa.sportyshoes.Tests
            HomePageTest.java
            LoginPageTest.java
            ProductsTest.java
            RegisterPageTest.java
        features
        steps
    JRE System Library [JavaSE-1.7]
    Maven Dependencies
    src
        main
        test
    target
    test-output
    pom.xml
```



localhost:9010/register

localhost:9010/register

Chrome is being controlled by automated test software.

Sporty Shoes

Powered By Simplilearn

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

Its Thu Feb 22 11:14:27 IST 2024 !

Register Yourself

Name:

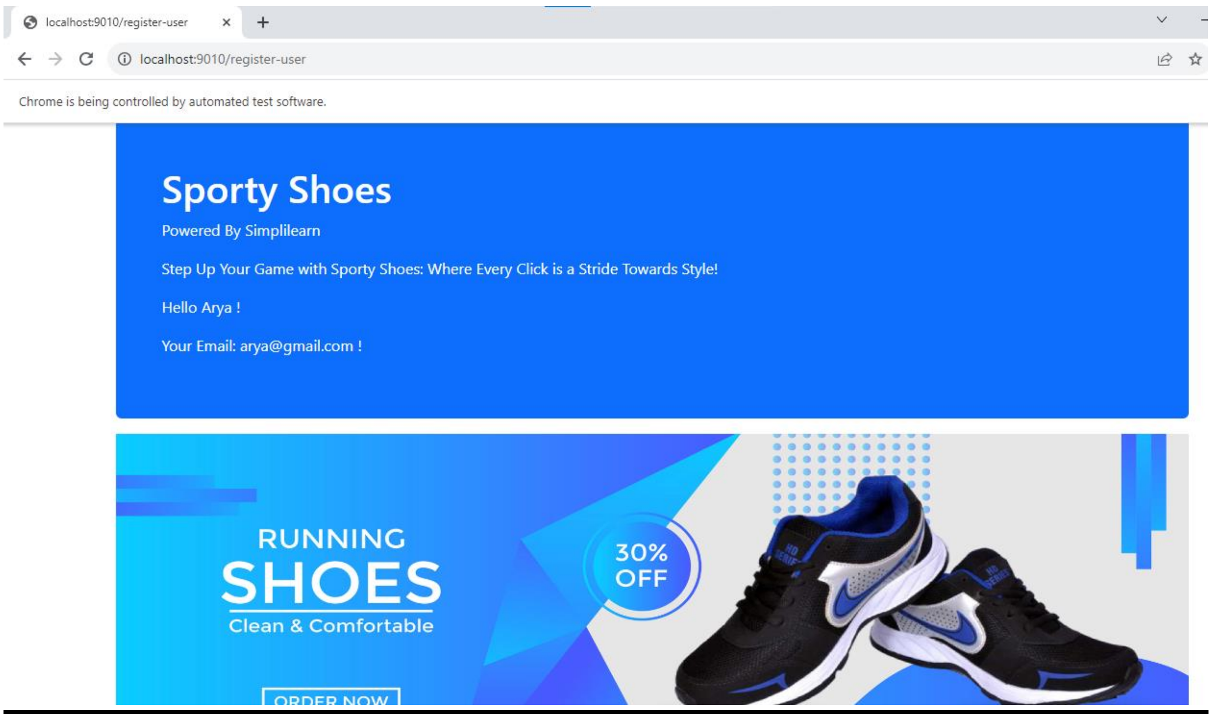
Enter Name

Email:

Enter email

Password:

Enter password



localhost:9010/register-user

localhost:9010/register-user

Chrome is being controlled by automated test software.

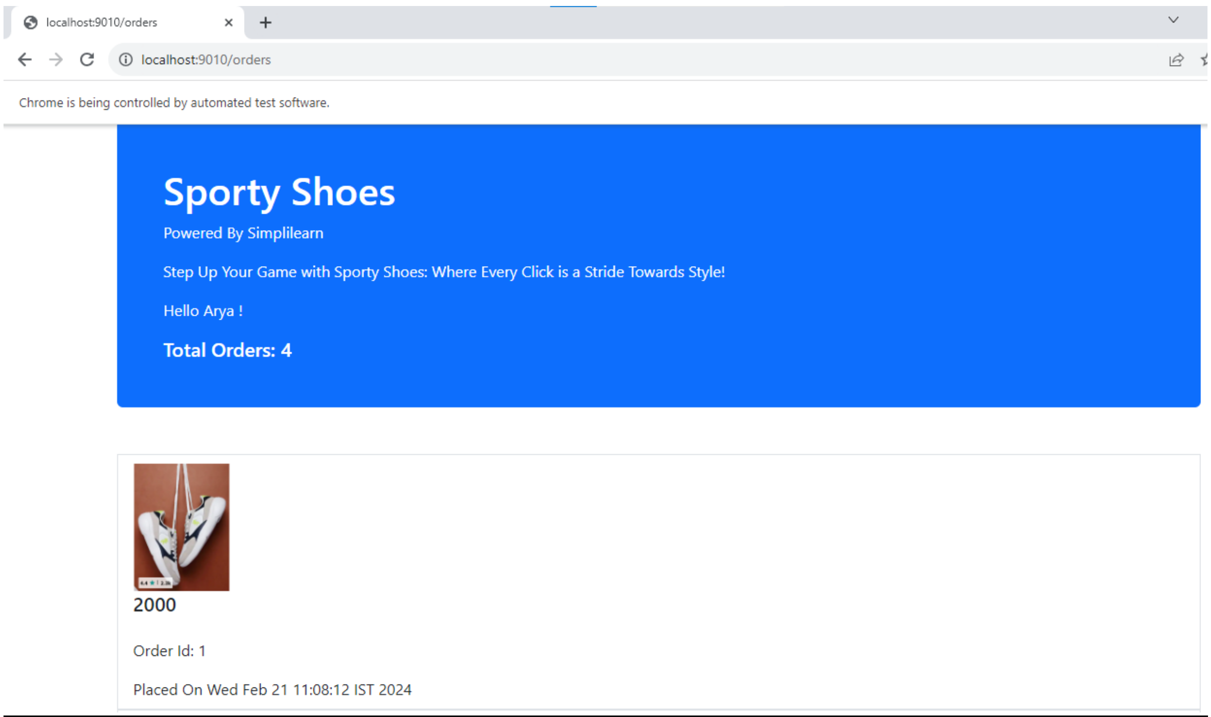
Sporty Shoes

Powered By Simplilearn

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

Hello Arya !

Your Email: arya@gmail.com !


localhost:9010/orders

Chrome is being controlled by automated test software.

Sporty Shoes

Powered By Simplilearn

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

Hello Arya !

Total Orders: 4

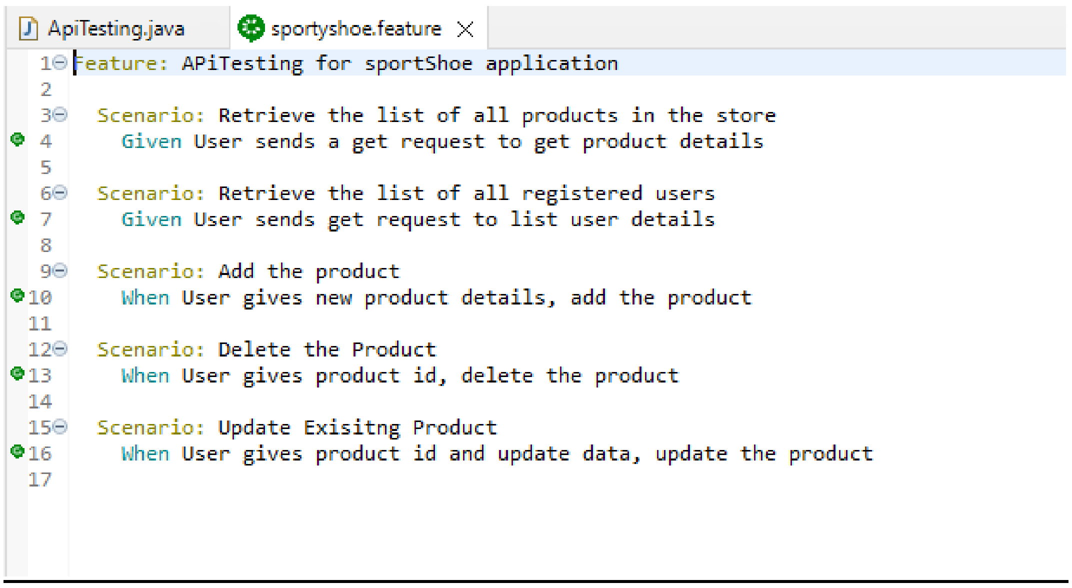
2000

Order Id: 1

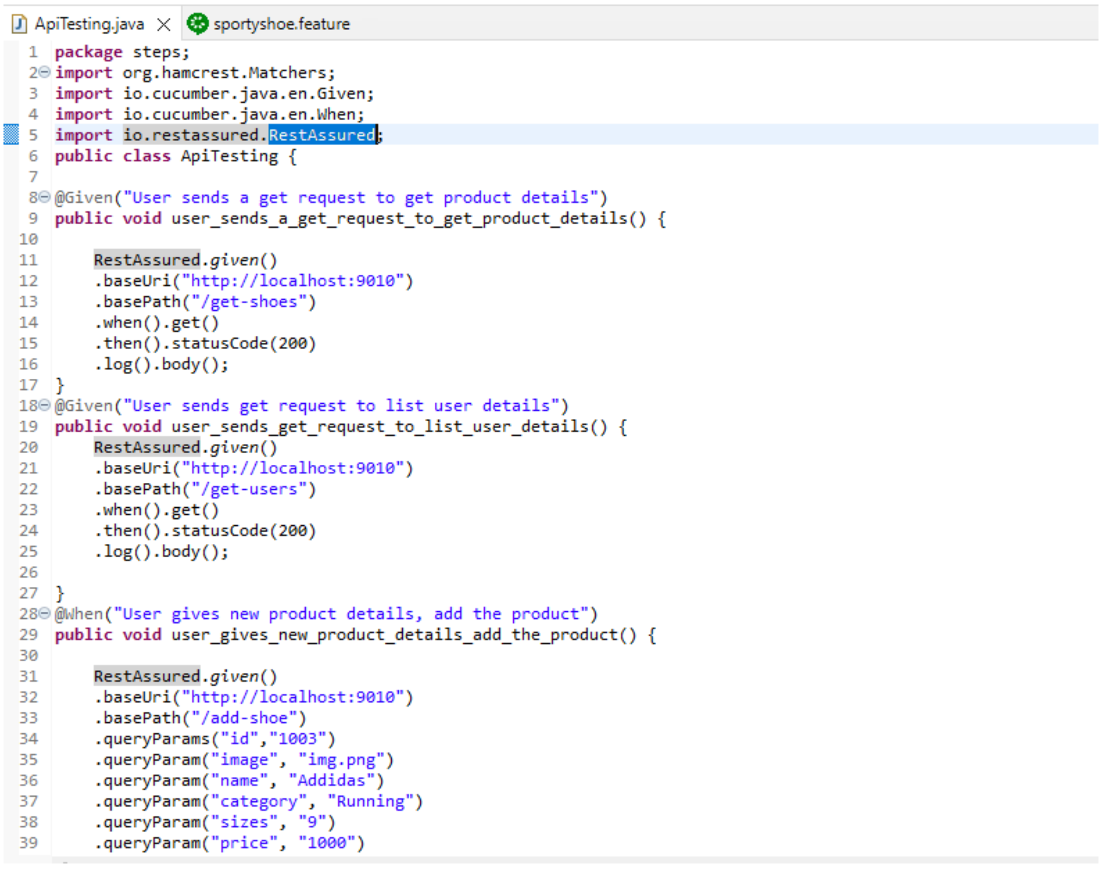
Placed On Wed Feb 21 11:08:12 IST 2024

```
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_CartBTN
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_orders
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_product_addtoCart
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_logout
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_HomeBTN
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_user_login
PASSED: com.qa.sportyshoes.Tests.ProductsTest.test_placeorderBTN

===============================================
    Default test
    Tests run: 7, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 7, Passes: 7, Failures: 0, Skips: 0
===============================================
```

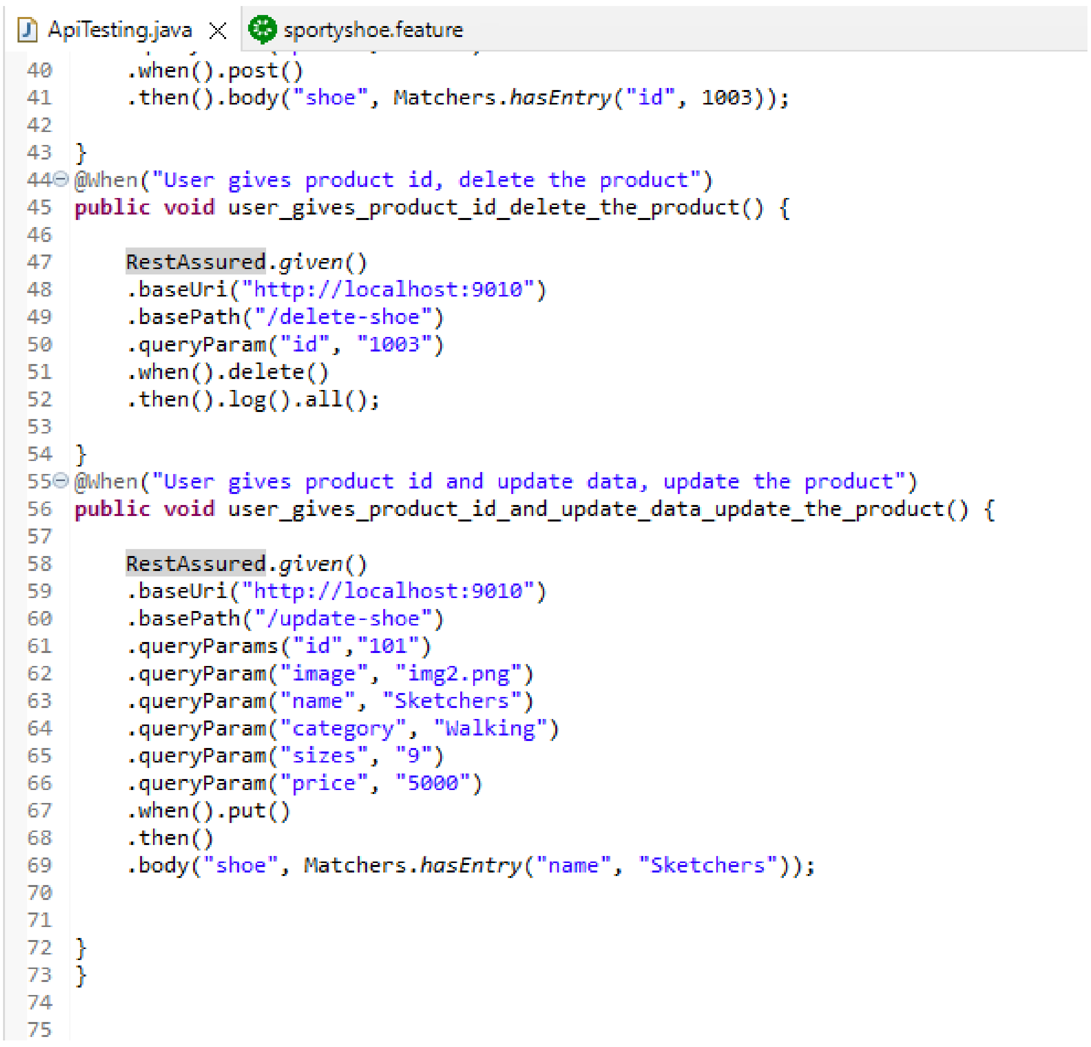
Rest-Assured with Cucumber



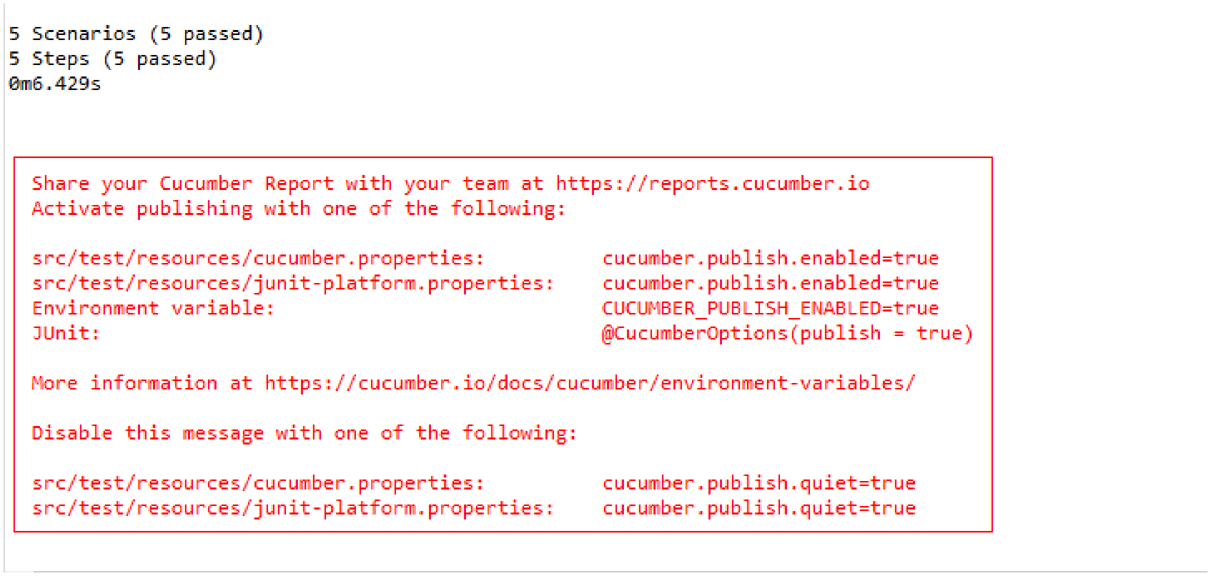
```
Feature: APiTesting for sportShoe application

  Scenario: Retrieve the list of all products in the store
    Given User sends a get request to get product details

  Scenario: Retrieve the list of all registered users
    Given User sends get request to list user details

  Scenario: Add the product
    When User gives new product details, add the product

  Scenario: Delete the Product
    When User gives product id, delete the product

  Scenario: Update Exisitng Product
    When User gives product id and update data, update the product

```



```
package steps;
import org.hamcrest.Matchers;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;
public class ApiTesting {

@Given("User sends a get request to get product details")
public void user_sends_a_get_request_to_get_product_details() {

    RestAssured.given()
    .baseUri("http://localhost:9010")
    .basePath("/get-shoes")
    .when().get()
    .then().statusCode(200)
    .log().body();
}
@Given("User sends get request to list user details")
public void user_sends_get_request_to_list_user_details() {
    RestAssured.given()
    .baseUri("http://localhost:9010")
    .basePath("/get-users")
    .when().get()
    .then().statusCode(200)
    .log().body();

}
@When("User gives new product details, add the product")
public void user_gives_new_product_details_add_the_product() {

    RestAssured.given()
    .baseUri("http://localhost:9010")
    .basePath("/add-shoe")
    .queryParams("id","1003")
    .queryParam("image", "img.png")
    .queryParam("name", "Addidas")
    .queryParam("category", "Running")
    .queryParam("sizes", "9")
    .queryParam("price", "1000")
```



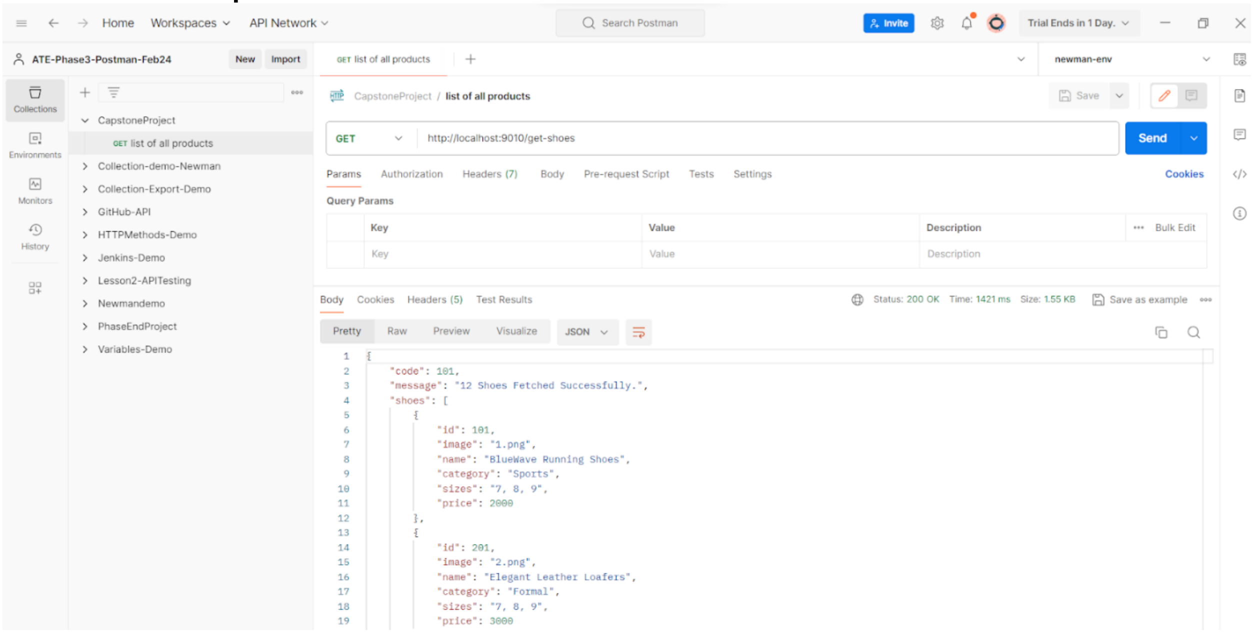
```java
    .when().post()
    .then().body("shoe", Matchers.hasEntry("id", 1003));

}
@When("User gives product id, delete the product")
public void user_gives_product_id_delete_the_product() {

    RestAssured.given()
    .baseUri("http://localhost:9010")
    .basePath("/delete-shoe")
    .queryParam("id", "1003")
    .when().delete()
    .then().log().all();

}
@When("User gives product id and update data, update the product")
public void user_gives_product_id_and_update_data_update_the_product() {

    RestAssured.given()
    .baseUri("http://localhost:9010")
    .basePath("/update-shoe")
    .queryParams("id","101")
    .queryParam("image", "img2.png")
    .queryParam("name", "Sketchers")
    .queryParam("category", "Walking")
    .queryParam("sizes", "9")
    .queryParam("price", "5000")
    .when().put()
    .then()
    .body("shoe", Matchers.hasEntry("name", "Sketchers"));


}
}
```



```
5 Scenarios (5 passed)
5 Steps (5 passed)
0m6.429s


Share your Cucumber Report with your team at https://reports.cucumber.io
Activate publishing with one of the following:

src/test/resources/cucumber.properties:          cucumber.publish.enabled=true
src/test/resources/junit-platform.properties:    cucumber.publish.enabled=true
Environment variable:                            CUCUMBER_PUBLISH_ENABLED=true
JUnit:                                           @CucumberOptions(publish = true)

More information at https://cucumber.io/docs/cucumber/environment-variables/

Disable this message with one of the following:

src/test/resources/cucumber.properties:          cucumber.publish.quiet=true
src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true
```

API Testing with Postman

1. Get request



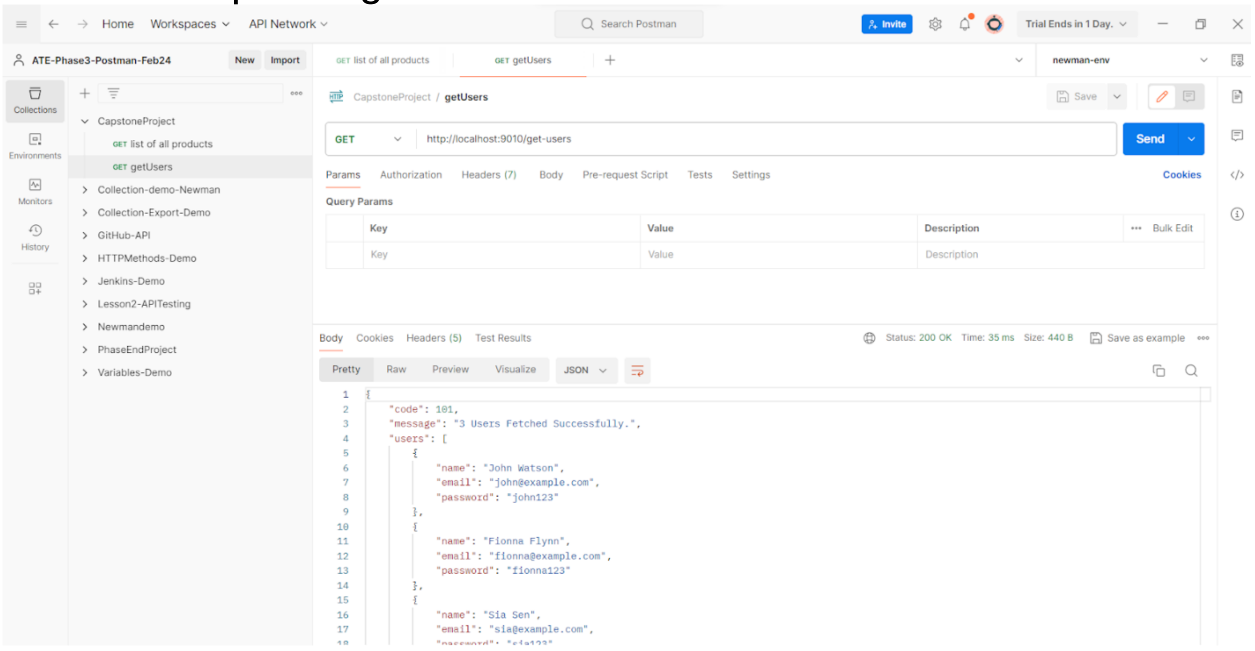
CapstoneProject / list of all products

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

Status: 200 OK Time: 1421 ms Size: 1.55 KB

```
{
    "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
```

2. Get request - get users



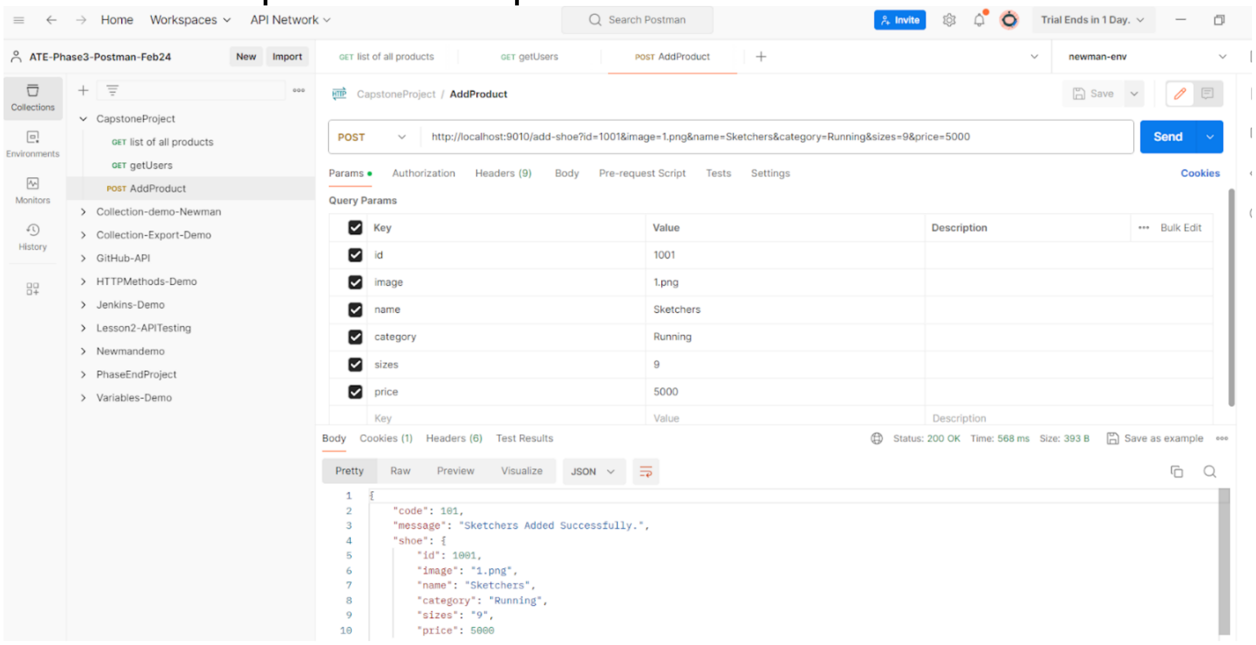
CapstoneProject / getUsers

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

Status: 200 OK Time: 35 ms Size: 440 B

```
{
    "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"
```

3. Post request - Add a product



POST http://localhost:9010/add-shoe?id=1001&image=1.png&name=Sketchers&category=Running&sizes=9&price=5000

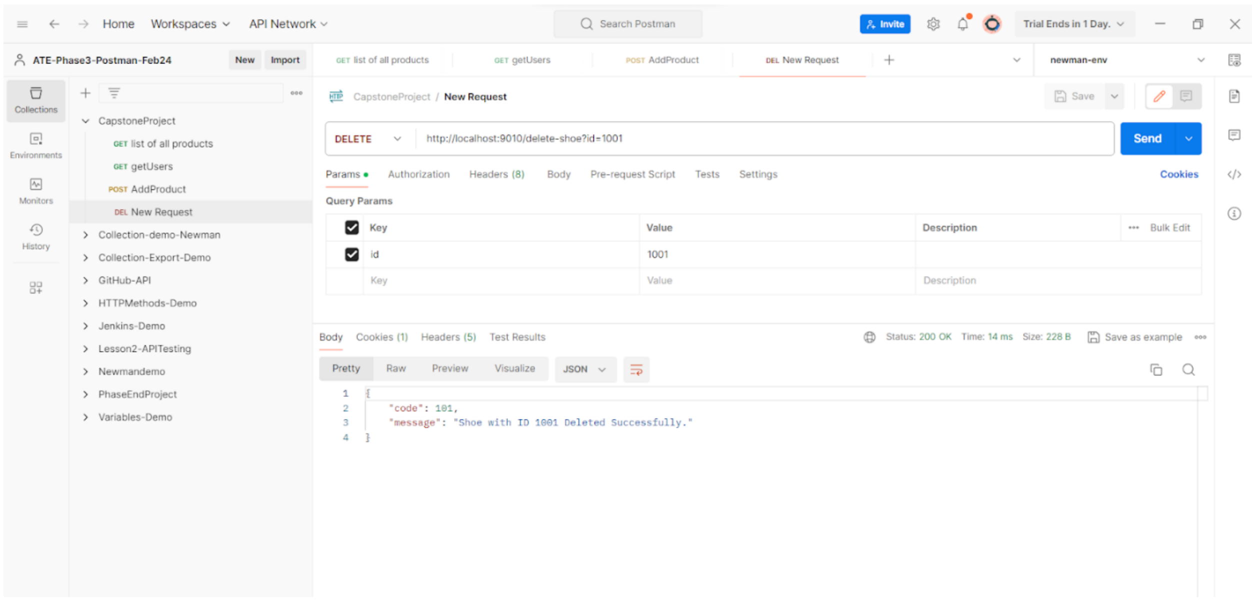
Query Params

Key	Value	Description
id	1001	
image	1.png	
name	Sketchers	
category	Running	
sizes	9	
price	5000	

Status: 200 OK Time: 568 ms Size: 393 B

```
{
    "code": 101,
    "message": "Sketchers Added Successfully.",
    "shoe": {
        "id": 1001,
        "image": "1.png",
        "name": "Sketchers",
        "category": "Running",
        "sizes": "9",
        "price": 5000
```

4. Delete request



DELETE http://localhost:9010/delete-shoe?id=1001

Query Params

Key	Value	Description
id	1001	

Status: 200 OK Time: 14 ms Size: 228 B

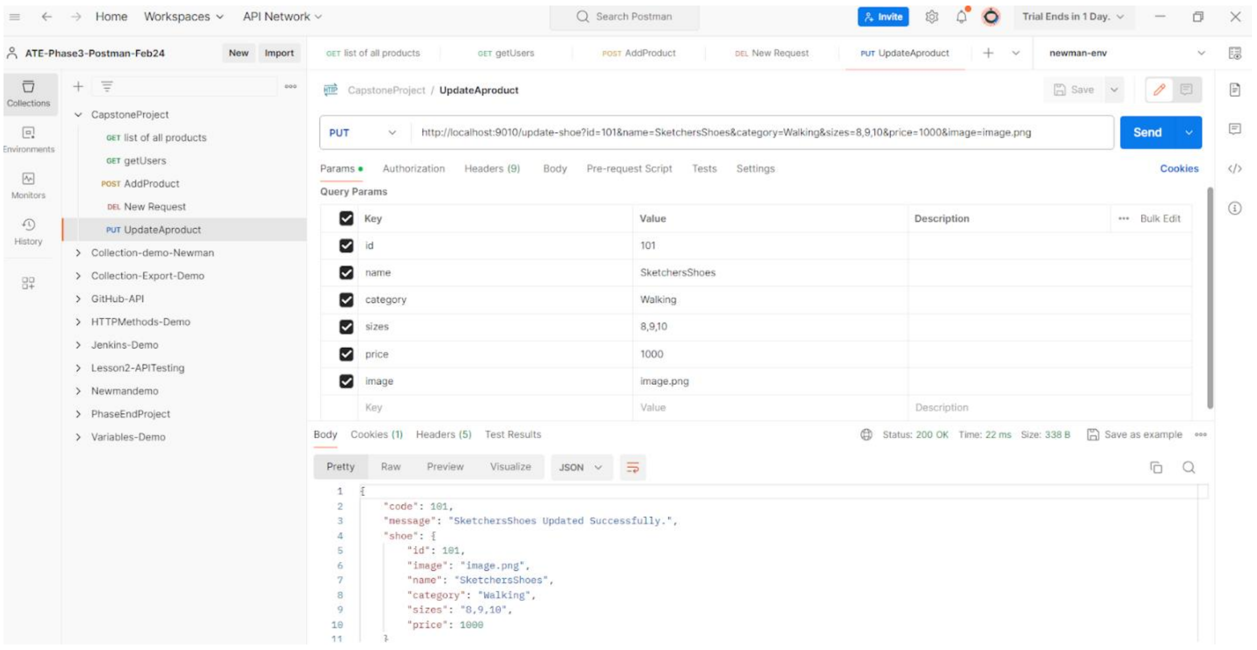
```
{
    "code": 101,
    "message": "Shoe with ID 1001 Deleted Successfully."
}
```

5. Put request

URL: http://localhost:9010/update-shoe

```
{
  "id": 101,
  "name": "SketchersShoes",
  "category": "Walking",
  "sizes": "8,9,10",
  "price": 1500,
  "image": "myimage.png"
}
```

Add above body as query parameters and send the request


PUT http://localhost:9010/update-shoe?id=101&name=SketchersShoes&category=Walking&sizes=8,9,10&price=1000&image=image.png

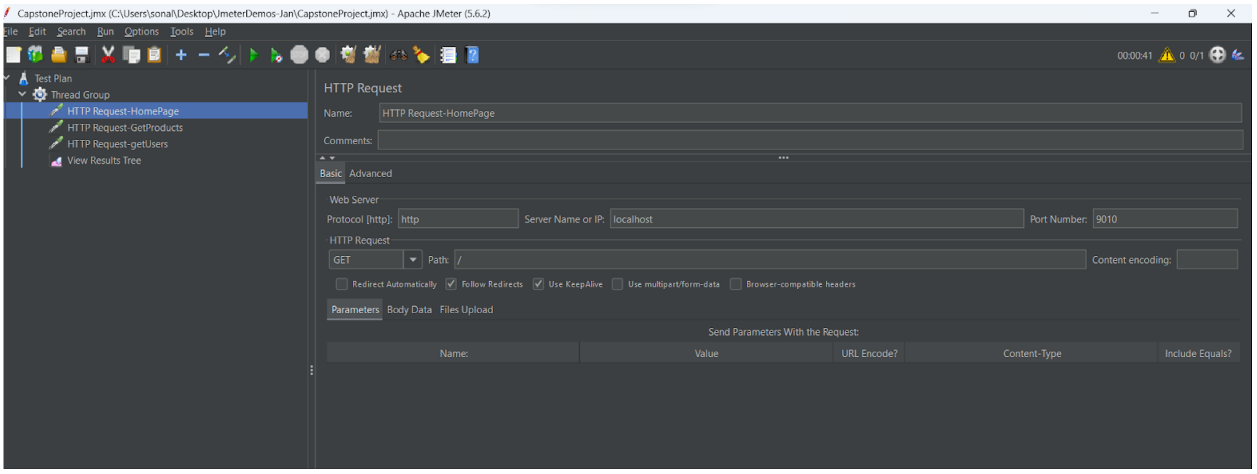
Query Params

Key	Value	Description
✓ id	101	
✓ name	SketchersShoes	
✓ category	Walking	
✓ sizes	8,9,10	
✓ price	1000	
✓ image	image.png	

Status: 200 OK Time: 22 ms Size: 338 B

```
{
    "code": 101,
    "message": "SketchersShoes Updated Successfully.",
    "shoe": {
        "id": 101,
        "image": "image.png",
        "name": "SketchersShoes",
        "category": "Walking",
        "sizes": "8,9,10",
        "price": 1000
    }
```

Load Testing using JMeter


CapstoneProject.jmx (C:\Users\sonal\Desktop\JmeterDemos-Jan\CapstoneProject.jmx) - Apache JMeter (5.6.2)

HTTP Request

Name: HTTP Request-HomePage

Comments:

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 9010

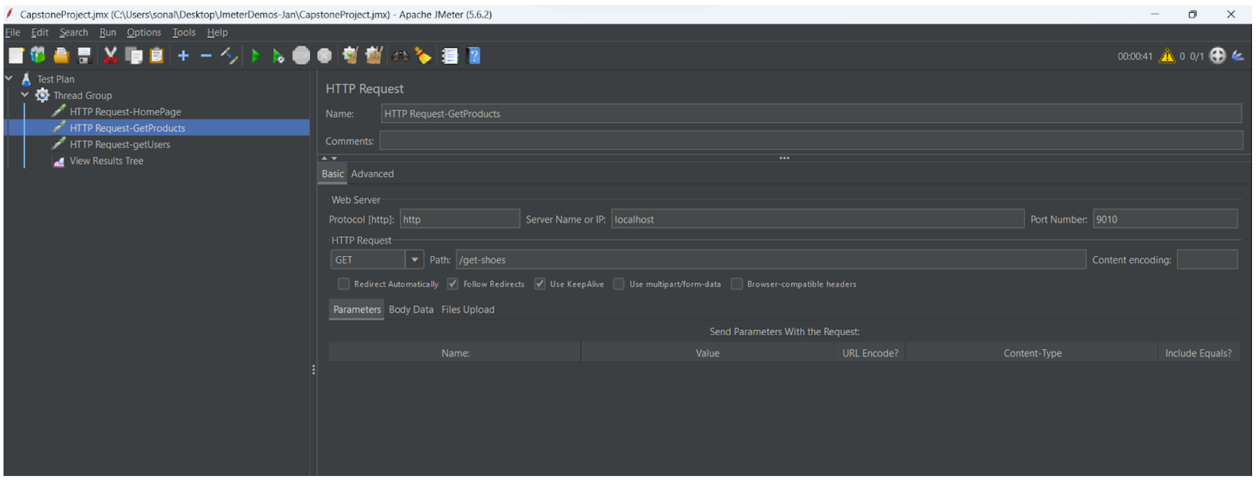
HTTP Request

GET Path: / Content encoding:

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?


CapstoneProject.jmx (C:\Users\sonal\Desktop\JmeterDemos-Jan\CapstoneProject.jmx) - Apache JMeter (5.6.2)

- Test Plan
  - Thread Group
    - HTTP Request-HomePage
    - HTTP Request-GetProducts
    - HTTP Request-getUsers
    - View Results Tree

HTTP Request

Name: HTTP Request-GetProducts

Comments:

Basic | Advanced

Web Server

Protocol [http]: http | Server Name or IP: localhost | Port Number: 9010

HTTP Request

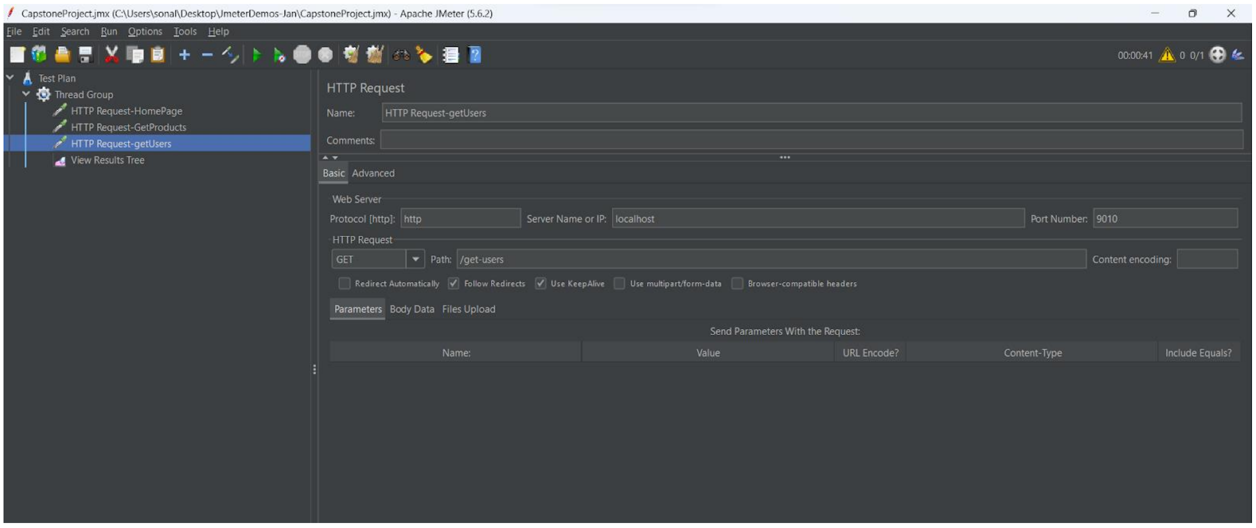
GET | Path: /get-shoes | Content encoding:

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?


CapstoneProject.jmx (C:\Users\sonal\Desktop\JmeterDemos-Jan\CapstoneProject.jmx) - Apache JMeter (5.6.2)

- Test Plan
  - Thread Group
    - HTTP Request-HomePage
    - HTTP Request-GetProducts
    - HTTP Request-getUsers
    - View Results Tree

HTTP Request

Name: HTTP Request-getUsers

Comments:

Basic | Advanced

Web Server

Protocol [http]: http | Server Name or IP: localhost | Port Number: 9010

HTTP Request

GET | Path: /get-users | 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?