CAPSTONE PROJECT

SELENIUM USING TESTNG:

TestBase->

```
package com.sportshoe.seleniumcuccumbersripts;
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();
      }
}
=========================================================================
LoginPage->

package com.sportshoe.seleniumcuccumbersripts;
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("Aparna@gmail.com");
            loginpassword.sendKeys("appu@123");
            loginbtn.click();
      }


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

LoginPageTest->

```
package com.sportshoe.Tests;
import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportshoe.seleniumcuccumbersripts.HomePage;
import com.sportshoe.seleniumcuccumbersripts.LoginPage;
import com.sportshoe.seleniumcuccumbersripts.RegisterPage;
import com.sportshoe.seleniumcuccumbersripts.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 Aparna !";
            String actualText = rp.validate_registration_Text();
            Assert.assertEquals(actualText, expected);
      }
@Test(priority='4')
public void test_click_cart()
{
        lp.click_cart();
}
}
```

HomePage->

```
package com.sportshoe.seleniumcuccumbersripts;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.WebDriver;
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();
}
}
```

HomePageTest->

```
package com.sportshoe.Tests;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import com.sportshoe.seleniumcuccumbersripts.TestBase;
import com.sportshoe.seleniumcuccumbersripts.HomePage;
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();
      }

}
```

RegisterPage->

```
package com.sportshoe.seleniumcuccumbersripts;

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("Aparna");
		registeremail.sendKeys("Aparna@gmail.com");
		registerpassword.sendKeys("appu@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;
	}
}
RegisterPageTest->
package com.sportshoe.Tests;
import static org.testng.Assert.assertEquals;

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

import com.sportshoe.seleniumcuccumbersripts.HomePage;
import com.sportshoe.seleniumcuccumbersripts.RegisterPage;
import com.sportshoe.seleniumcuccumbersripts.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')
```

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

AddToCartPage->

```java
package com.sportshoe.seleniumcuccumbersripts;
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;

    }
}
```

AddToCartPageTest->

```java
package com.sportshoe.Tests;
import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportshoe.seleniumcuccumbersripts.AddtoCartPage;
import com.sportshoe.seleniumcuccumbersripts.HomePage;
import com.sportshoe.seleniumcuccumbersripts.LoginPage;
import com.sportshoe.seleniumcuccumbersripts.RegisterPage;
import com.sportshoe.seleniumcuccumbersripts.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);
}
}
```

OrderPage->

```java
package com.sportshoe.seleniumcuccumbersripts;

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();

    }

}
```

OrderPageTest->

```
package com.sportshoe.Tests;
import org.testng.Assert;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportshoe.seleniumcuccumbersripts.HomePage;
import com.sportshoe.seleniumcuccumbersripts.LoginPage;
import com.sportshoe.seleniumcuccumbersripts.OrderPage;
import com.sportshoe.seleniumcuccumbersripts.RegisterPage;
import com.sportshoe.seleniumcuccumbersripts.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);
        }   }
```

CUCUMBER:

GetAllShoe and Users:

```
Feature: Test sportyshoe page to get all shoes and users
Scenario: Get All Shoes and Users in localhost:9010
Given User Enters localhost URl
Given User send a request to get all shoes
When User executes HTTP get method
Then validate the response status code

Scenario:et All Shoes and Users in localhost:9010
Given User Enters localhost URl
Given User send a request to get all users
When User executes HTTP get method
Then validate the response status code
```

Add,Update and Delete->

```
Feature: Test sportyshoe page to add,update and delete the product
Scenario: Add new product in localhost:9010
Given User Enters localhost URl
Given User send a request to add a new product
When User executes HTTP get method
Then validate the response status code

Scenario: update product in localhost:9010
Given User Enters localhost URl
Given User send a request to update product
When User executes HTTP get method
Then validate the response status code

Scenario: delete product in localhost:9010
Given User Enters localhost URl
Given User send a request to delete product
When User executes HTTP get method
Then validate the response status code
```

RESTASSURE:

GetAllshoeandusers->

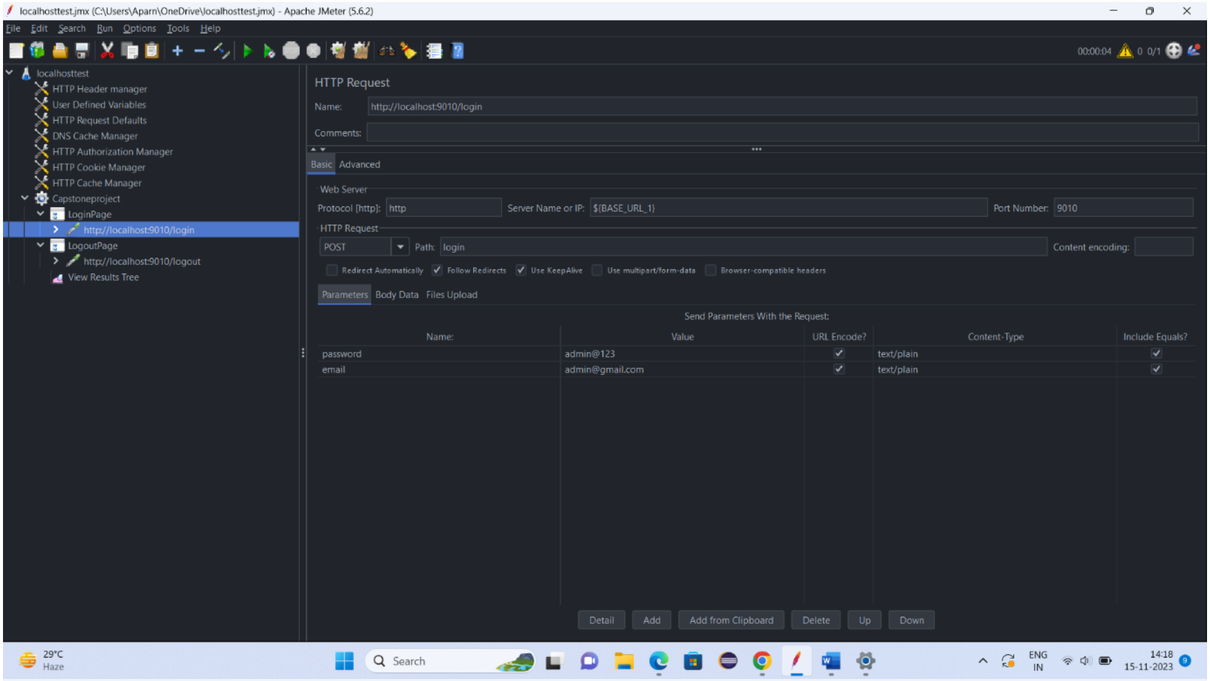
```
package com.sportsshoe.restassure;

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(401)
            .log()
            .all();

      }
}
```

Add,update and delete->

```
package com.sportsshoe.restassure;

import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class Postandput {
@Test(priority='1')
public void add_new_product()
      {
            RestAssured.given()
            .baseUri("http://localhost:9010")
            .basePath("/add-shoe")
            .queryParam("id","1020")
            .queryParam("image", "www.image.com")
            .queryParam("name","Nike")
            .queryParam("category", "Running")
            .queryParam("sizes","8,9,10")
            .queryParam("price", "1100")
            .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.image123.com")
            .queryParam("name","Reebok")
            .queryParam("category", "Running")
            .queryParam("sizes","8,9,10")
            .queryParam("price", "1100")
            .when()
            .put()
            .then()
            .log().all();

}

@Test(priority='3')
public void delete_product()
      {
            RestAssured.given()
            .baseUri("http://localhost:9010")
            .basePath("/delete-shoe")
            .queryParam("id", "1020")
            .when().delete()
            .then().log().all();

      }
      }
```

JMETER:



localhosttest.jmx (C:\Users\Aparn\OneDrive\localhosttest.jmx) - Apache JMeter (5.6.2)

- localhosttest
  - HTTP Header manager
  - User Defined Variables
  - HTTP Request Defaults
  - DNS Cache Manager
  - HTTP Authorization Manager
  - HTTP Cookie Manager
  - HTTP Cache Manager
  - Capstoneproject
    - LoginPage
      - http://localhost:9010/login
    - LogoutPage
      - http://localhost:9010/logout
    - View Results Tree

HTTP Request

Name: http://localhost:9010/login

Comments:

Web Server

Protocol [http]: http Server Name or IP: ${BASE_URL_1} Port Number: 9010

HTTP Request

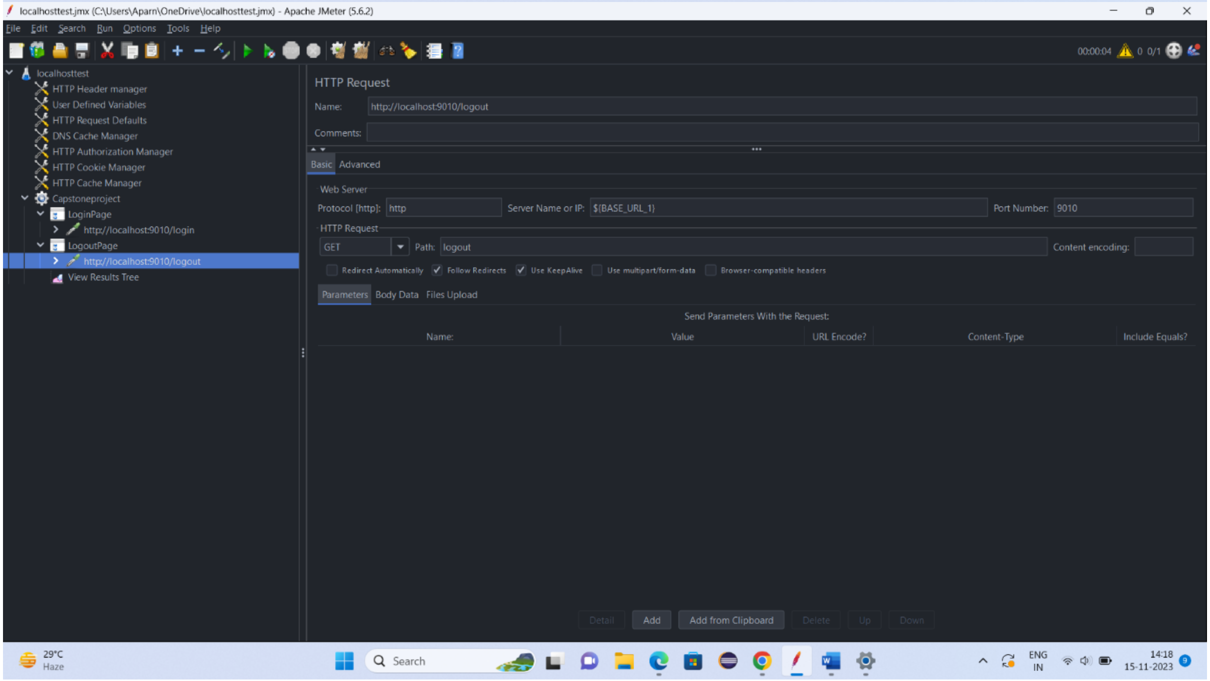
POST Path: login Content encoding:

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

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
password	admin@123	✓	text/plain	✓
email	admin@gmail.com	✓	text/plain	✓

File Edit Search Run Options Tools Help
00:00:04 0 0/1
Basic Advanced
Parameters Body Data Files Upload
Detail Add Add from Clipboard Delete Up Down
29°C Haze
Search
ENG IN
14:18 15-11-2023



localhosttest.jmx (C:\Users\Aparn\OneDrive\localhosttest.jmx) - Apache JMeter (5.6.2)

- localhosttest
  - HTTP Header manager
  - User Defined Variables
  - HTTP Request Defaults
  - DNS Cache Manager
  - HTTP Authorization Manager
  - HTTP Cookie Manager
  - HTTP Cache Manager
  - Capstoneproject
    - LoginPage
      - http://localhost:9010/login
    - LogoutPage
      - http://localhost:9010/logout
    - View Results Tree

HTTP Request

Name: http://localhost:9010/logout

Comments:

Web Server

Protocol [http]: http Server Name or IP: ${BASE_URL_1} Port Number: 9010

HTTP Request

GET Path: logout Content encoding:

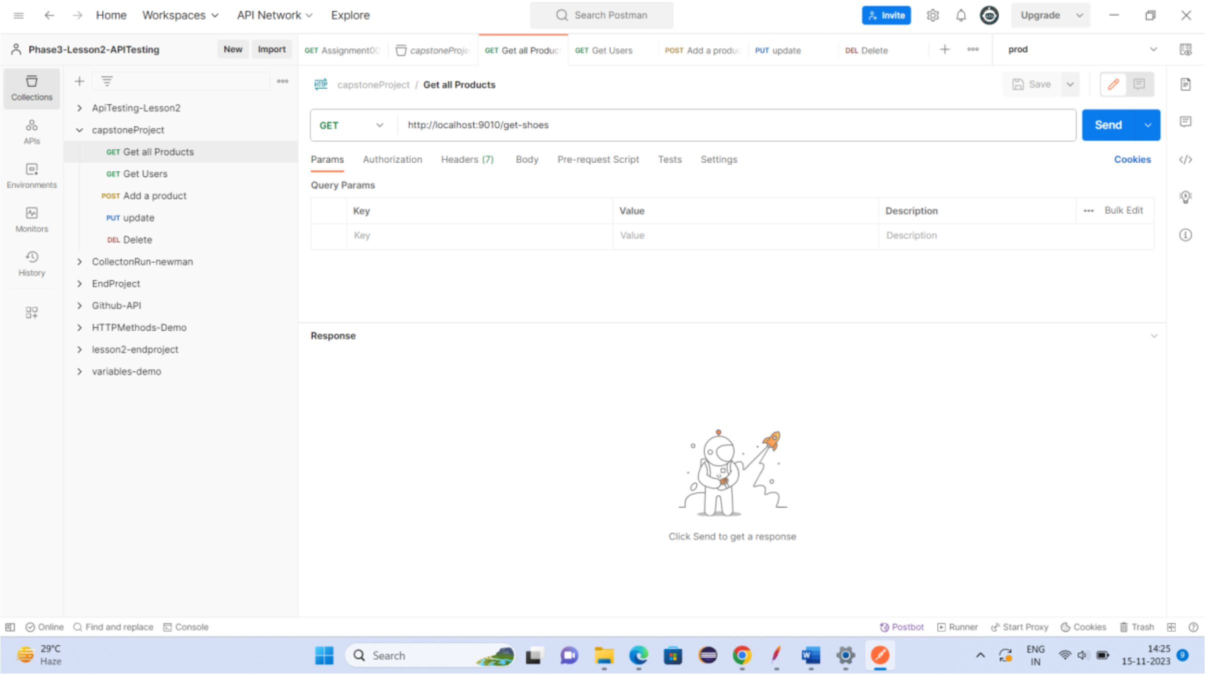
□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

Send Parameters With the Request:

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

File Edit Search Run Options Tools Help
00:00:04 0 0/1
Basic Advanced
Parameters Body Data Files Upload
Detail Add Add from Clipboard Delete Up Down
29°C Haze
Search
ENG IN
14:18 15-11-2023

POSTMAN:



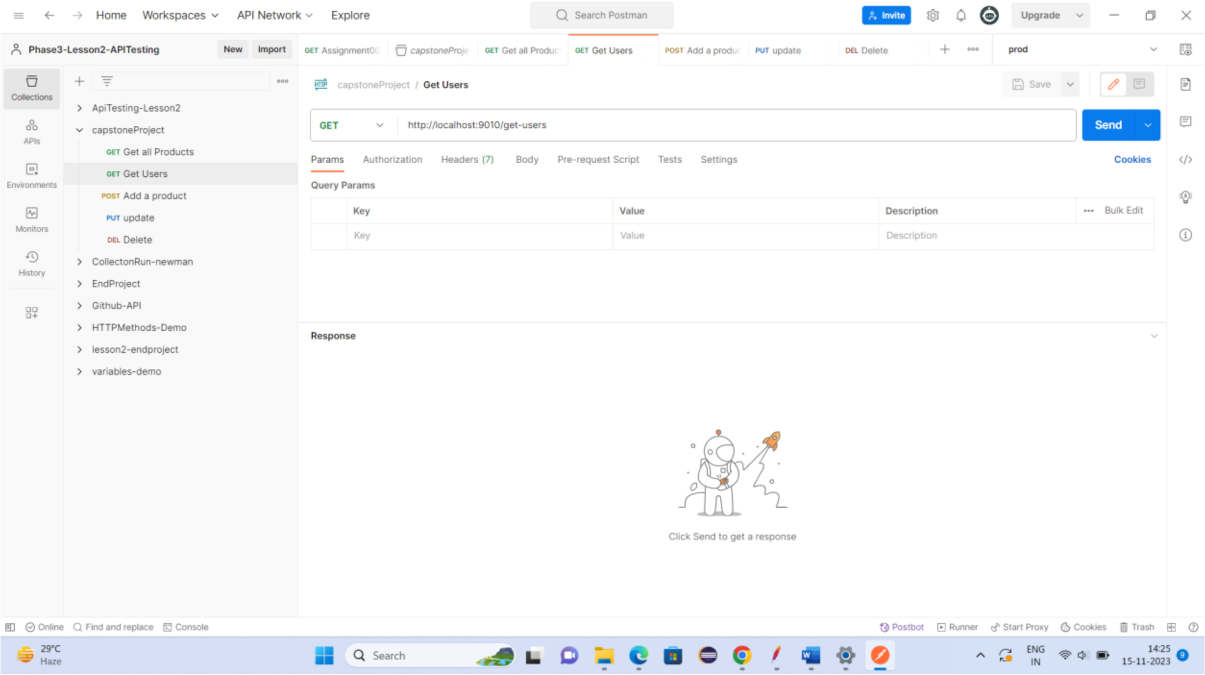
capstoneProject / Get all Products

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

Query Params

Key	Value	Description
Key	Value	Description

Click Send to get a response



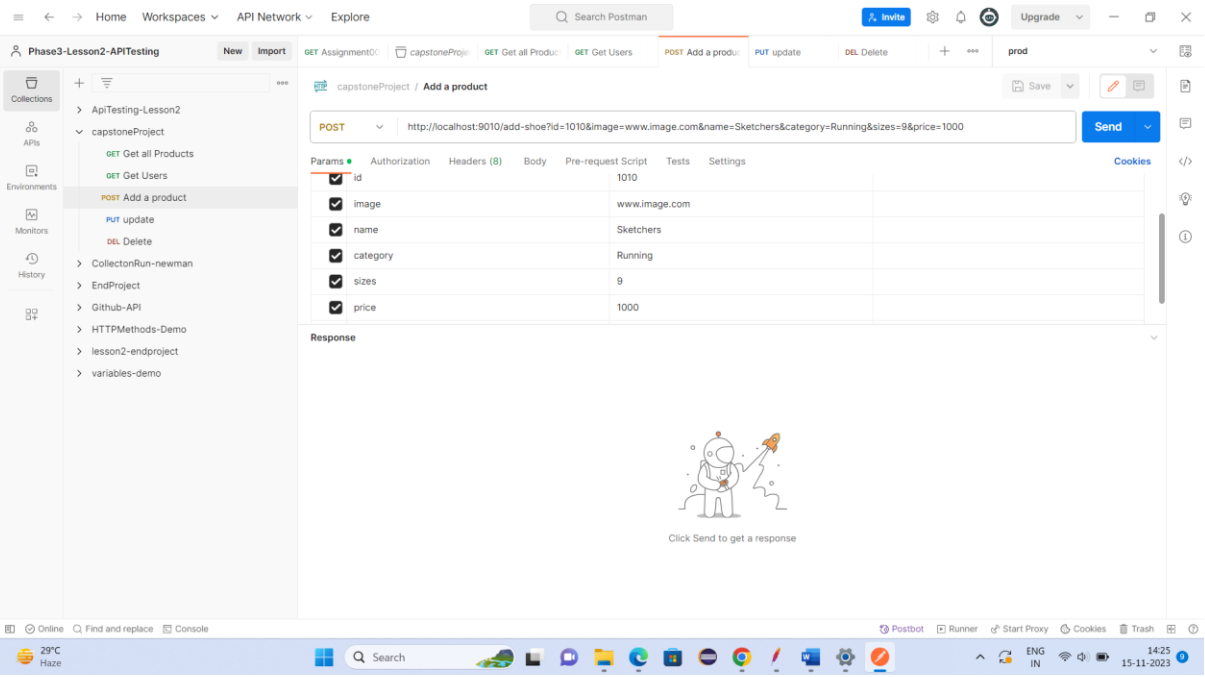
capstoneProject / Get Users

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

Query Params

Key	Value	Description
Key	Value	Description

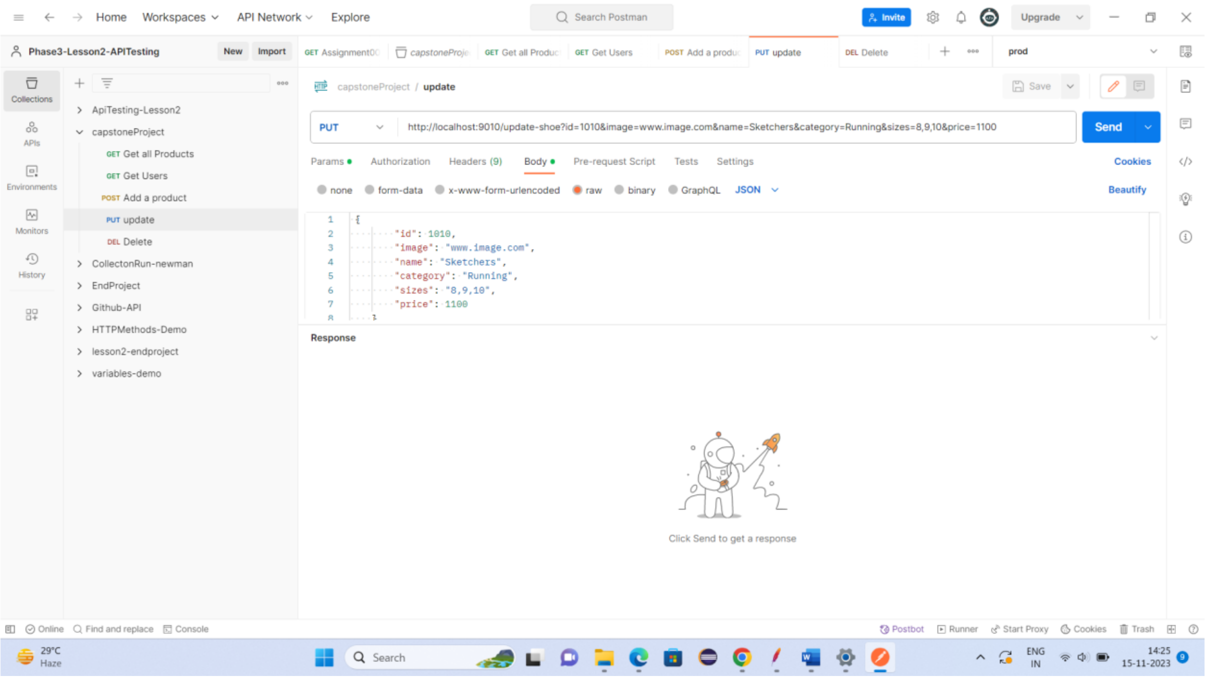
Click Send to get a response



capstoneProject / Add a product

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

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

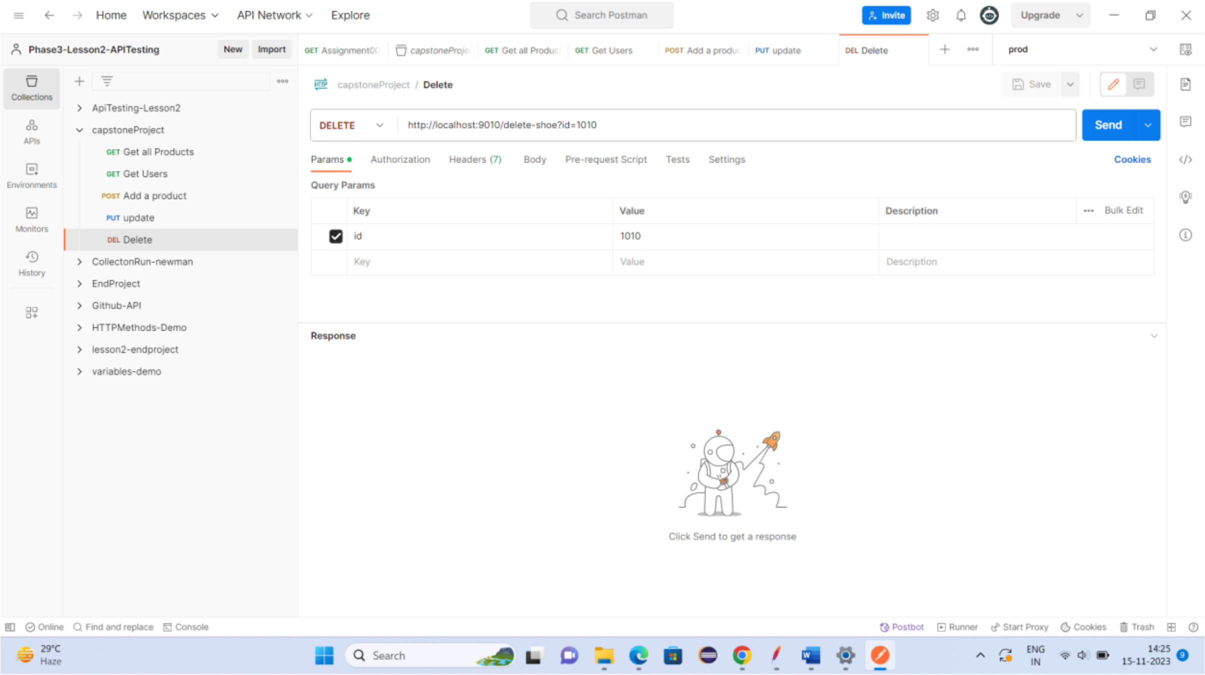


capstoneProject / update

PUT http://localhost:9010/update-shoe?id=1010&image=www.image.com&name=Sketchers&category=Running&sizes=8,9,10&price=1100

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id": 1010,
    "image": "www.image.com",
    "name": "Sketchers",
    "category": "Running",
    "sizes": "8,9,10",
    "price": 1100
}
```



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

Query Params

	Key	Value	Description
☑	id	1010	
	Key	Value	Description