

CAPSTONE PROJECT

By using <http://localhost:9010/> this website I can automate to buy a product by using POM(Page Object Model).Here I use Selenium and Testng.

Source code for Landing Page:

```
package in.sportyshoes.pages;
import org.openqa.selenium.WebDriver;
public class LandingPage {
private WebDriver driver;
public LandingPage(WebDriver driver) {
this.driver = driver;
}
public void goToHomePage() {
driver.get("http://localhost:9010/");
}
}
```

Source code for Register Page:

```
package in.sportyshoes.pages;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
public class RegisterPage {
private WebDriver driver;
public RegisterPage(WebDriver driver) {
this.driver = driver;
}
public void fillRegistrationForm(String name, String email, String password){
driver.findElement(By.xpath("/html/body/div[2]/form/a")).click();
driver.findElement(By.id("name")).sendKeys(name);
driver.findElement(By.id("email")).sendKeys(email);
driver.findElement(By.id("password")).sendKeys(password);
}
public void submitRegistrationForm() {
driver.findElement(By.xpath("//button[contains(@type,'submit')]")).click();
}
}
```

Source code for Add Product To Cart page:

```
package in.sportyshoes.pages;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
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 AddProductToCartPage {
private WebDriver driver;
public AddProductToCartPage(WebDriver driver) {
this.driver = driver;
}
public void clickProductButton() throws InterruptedException {
WebDriverWait wait = new WebDriverWait(driver,Duration.ofSeconds(60));
WebElement element =
wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//a[contains(@class,'nav-link')])[1]")));
element.click();
}
```

```

        driver.findElement(By.xpath("//a[contains(@class,'nav-link')])[1]")).click();
        Thread.sleep(2000);
        WebElement element1 =
        driver.findElement(By.xpath("//a[contains(@class,'btn btn-primary')])[1]"));
        ((JavascriptExecutor) driver).executeScript("arguments[0].click();",
        element1);
    }
}

```

Source code for Place Order Page:

```

package in.sportyshoes.pages;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class PlaceOrderPage {
private WebDriver driver;
public PlaceOrderPage(WebDriver driver) {
this.driver = driver;
}
public void navigateToCartAndPlaceOrder() throws InterruptedException {
driver.findElement(By.xpath("//a[contains(@class,'nav-link')])[1]")).click();
Thread.sleep(2000);
driver.findElement(By.xpath("//a[contains(@class,'nav-link')])[1]")).click();
Thread.sleep(2000);
WebElement element2 =
driver.findElement(By.xpath("//a[contains(@class,'btn btn-primary')])[1]"));
((JavascriptExecutor) driver).executeScript("arguments[0].click();",
element2);
}
}

```

Source code for SportyShoeOrder:

```

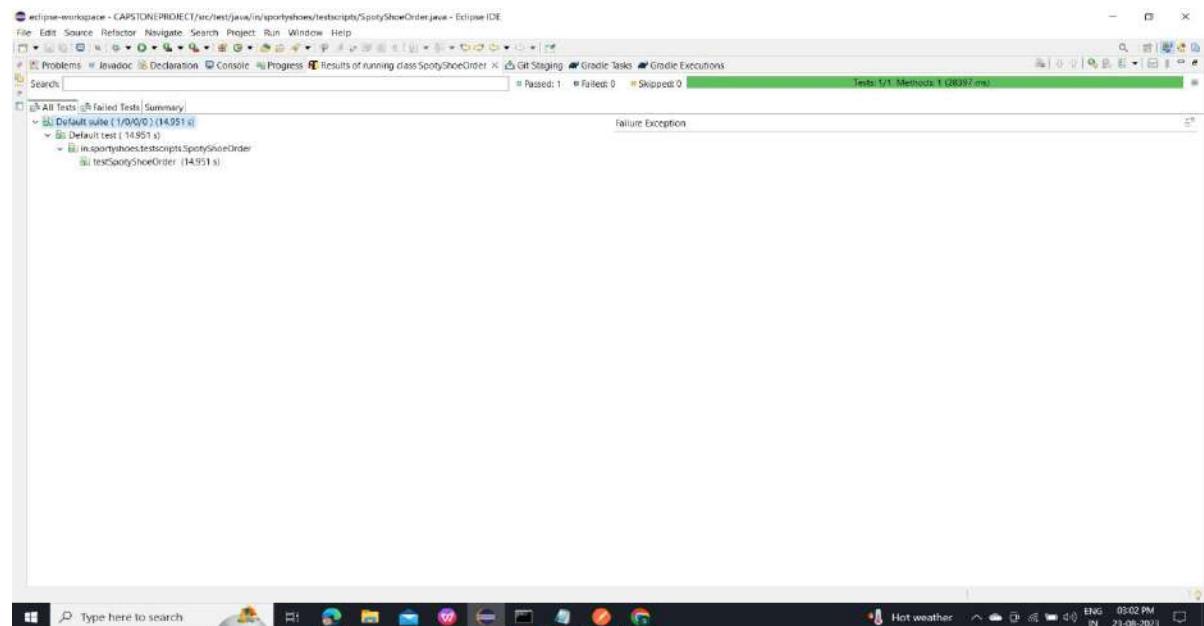
package in.sportyshoes.testscripts;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import in.sportyshoes.pages.LandingPage;
import in.sportyshoes.pages.PlaceOrderPage;
import in.sportyshoes.pages.AddProductToCartPage;
import in.sportyshoes.pages.RegisterPage;
public class SpottyShoeOrder {
private WebDriver driver;
private LandingPage landingPage;
private RegisterPage registerPage;
private AddProductToCartPage addproducttocartPage;
private PlaceOrderPage placeorderPage;
@BeforeMethod
public void setUp() {
driver = new ChromeDriver();
driver.manage().window().maximize();
landingPage = new LandingPage(driver);
}
}

```

```

registerPage = new RegisterPage(driver);
addproducttocartPage = new AddProductToCartPage(driver);
placeorderPage = new PlaceOrderPage(driver);
landingPage.goToHomePage();
}
@Test
public void testSpotyShoeOrder() throws InterruptedException {
registerPage.fillRegistrationForm("john", "abcdef@gmail.com", "123456");
Thread.sleep(2000);
registerPage.submitRegistrationForm();
Thread.sleep(2000);
addproducttocartPage.clickProductButton();
Thread.sleep(2000);
placeorderPage.navigateToCartAndPlaceOrder();
}
@AfterMethod
public void tearDown() {
driver.quit();
}
}

```



```
eclipse-workspace - CAPSTONEPROJECT/src/test/java/in/sportsheets/testscripts/SpottyShoeOrder.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Progress Results of running class SpottyShoeOrder Git Staging Gradle Tasks Gradle Executions
terminated: SpottyShoeOrder [testNG] C:\Users\hp\AppData\Local\Temp\org.eclipse.jdt.core\jdt.ls\full\win32\x86_64\17.0.7\20230425-1502\rel\bin\javaw.exe [23-Aug-2023, 2:33:02 pm - 2:33:35 pm] [pid: 14228]
SLF4J: Defaulting to no-operation (NOP) logger implementation.
SLF4J: See http://www.slf4j.org/codes.html#staticLoggerBinder for further details.
Aug 23, 2023 2:33:17 PM org.openqa.selenium.devtools.CdpVersionFinder findMoreRecentMatch
WARNING: Unable to find an exact match for CDP version 110, so returning the closest version found: 115
PASSED: in.in.sportsheets.testscripts.SpottyShoeOrder

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

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

By using jmeter load scripts:

```
SORTYSHOES.jmx (C:\Users\hp\Downloads\SPORTYSHOES.jmx) - Apache JMeter (5.6.2)
File Edit Search Run Options Tools Help
00:01:28 0/0/1 ENG
View Results Tree
Name: View Results Tree
Comments:
Write results to file / Read from file
Filename:
Search Case sensitive Regular exp Search Reset
+ Test
+ http://localhost:9010/
+ http://localhost:9010/register
+ http://localhost:9010/register-user
+ http://localhost:9010/cart
+ http://localhost:9010/home
+ http://localhost:9010/add-to-cart?id=101
+ http://localhost:9010/logout
+ http://localhost:9010/cart
+ http://localhost:9010/place-order
View Results Tree
Raw Parsed
35°C Mostly cloudy 02:57 PM IN 23-08-2023
```

By using postman:

Using GET Request for get-shoes:

The screenshot shows the Postman application interface. The left sidebar displays a workspace named 'ws_simpilearn_aug_10' containing several collections: GoRest, OpenWeatherMap, RegExes, and REGRES-1. Under REGRES-1, there are four requests: GET All users, POST Create user, PUT Update user, and DELETE user. Below these is a collection named SpottyShoes with three requests: GET http://localhost:9010/get-shoes, GET http://localhost:9010/get-users, and POST http://localhost:9010/add-shoe... . The current request being viewed is 'GET http://localhost:9010/get-shoes'. The 'Body' tab of the request details panel shows a JSON response with code 101, message "3 Shoes Fetched Successfully.", and a list of three shoe objects. The first shoe object has id 101, image "1.png", name "Bluewave Running Shoes", category "Sports", sizes "7, 8, 9", and price 3000. The second shoe object has id 102, image "2.png", name "Redline Basketball Shoes", category "Sports", sizes "10, 11, 12", and price 4000. The third shoe object has id 103, image "3.png", name "Sprint Runner Shoes", category "Sports", sizes "13, 14, 15", and price 3500. The response status is 200 OK with a time of 497 ms and size of 1.64 KB.

```
1
2   "code": 101,
3   "message": "3 Shoes Fetched Successfully.",
4   "shoes": [
5     {
6       "id": 101,
7       "image": "1.png",
8       "name": "Bluewave Running Shoes",
9       "category": "Sports",
10      "sizes": "7, 8, 9",
11      "price": 3000
12    },
13    {
14      "id": 102,
15      "image": "2.png",
16      "name": "Redline Basketball Shoes",
17      "category": "Sports",
18      "sizes": "10, 11, 12",
19      "price": 4000
20    },
21    {
22      "id": 103,
23      "image": "3.png",
24      "name": "Sprint Runner Shoes",
25      "category": "Sports",
26      "sizes": "13, 14, 15",
27      "price": 3500
28    }
29  ],
30  "total": 3
```

Using GET Request for get-users:

The screenshot shows the Postman application interface. The left sidebar displays a workspace named 'ws_simpilearn_aug_10' containing several collections: GoRest, OpenWeatherMap, RegExes, and REGRES-1. Under REGRES-1, there are four requests: GET All users, POST Create user, PUT Update user, and DELETE user. Below these is a collection named SpottyShoes with three requests: GET http://localhost:9010/get-shoes, GET http://localhost:9010/get-users, and POST http://localhost:9010/add-shoe... . The current request being viewed is 'GET http://localhost:9010/get-users'. The 'Body' tab of the request details panel shows a JSON response with code 101, message "3 Users Fetched Successfully.", and a list of three user objects. The first user object has name "John Watson", email "john@example.com", and password "john123". The second user object has name "Fionna Flynn", email "fionna@example.com", and password "fionna123". The third user object has name "Liam O'Brien", email "liam@example.com", and password "liam123". The response status is 200 OK with a time of 28 ms and size of 440 B.

```
1
2   "code": 101,
3   "message": "3 Users Fetched Successfully.",
4   "users": [
5     {
6       "name": "John Watson",
7       "email": "john@example.com",
8       "password": "john123"
9     },
10    {
11      "name": "Fionna Flynn",
12      "email": "fionna@example.com",
13      "password": "fionna123"
14    },
15    {
16      "name": "Liam O'Brien",
17      "email": "liam@example.com",
18      "password": "liam123"
19    }
20  ],
21  "total": 3
```

Using POST Request add-shoe:

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'Environments', 'Monitors', and 'History'. The main area shows a 'POST' request to 'http://localhost:9010/add-shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=1000'. The 'Tests' tab contains a JavaScript test script:

```
1 pm.test("Response status code is 200", function () {
2     pm.response.to.have.status(200);
3 });
4
5
6 pm.test("Response code is 0", function () {
7     pm.expect(pm.response.code).to.equal(0);
8 });
9
10
11 pm.test("Response message is empty", function () {
12 });
13 }
```

Below the script, the 'Body' tab shows a JSON response:

```
1 {
2     "code": 101,
3     "message": "SampleShoe Added Successfully.",
4     "shoe": {
5         "id": 101,
6         "image": "image_url",
7         "name": "SampleShoe",
8         "category": "Running",
9         "sizes": "9",
10         "price": 1000
11     }
12 }
```

The status bar at the bottom indicates 'Status: 200 OK Time: 1728 ms Size: 388 B Save as Example'.

Rest Assured Using Requests:

GET Request for shoes:

```
package RestAssured;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class GETDemo {
    @Test
    public void getServerResponse() {
        RestAssured
            .when()
            .get("http://localhost:9010/get-shoes")
            .then()
            .assertThat()
            .statusCode(200)
            .log().all();
    }
}
```

```

{
  "code": 200,
  "message": "34 Shoes Fetched Successfully.",
  "shoes": [
    {
      "id": 101,
      "image": "1.png",
      "name": "Bluekve 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": 3500
    }
  ]
}

```

GET Request for users:

```

package RestAssured;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class GETDemo1 {
@Test
public void getServerResponse() {
RestAssured
.given()
.when()
.get("http://localhost:9010/get-users")
.then()
.assertThat()
.log().all()
.statusCode(200);
}
}

```

```

{
  "code": 200,
  "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"
    }
  ]
}

```

PASSSED: RestAssured.GITDemo1.getServerResponse

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

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

POST Request for add shoe:

```
package RestAssured;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;
public class POSTDemo {
@BeforeMethod
public void setUp() {
RestAssured.baseURI = "http://localhost:9010"; // Update the base URI
}
@Test
public void createShoeResourceWithParameters() {
Response response = given()
.param("id", 101)
.param("image", "image_url")
.param("name", "SampleShoe")
.param("category", "Running")
.param("sizes", "9")
.param("price", 1000)
.when()
.post("/add-shoe");
JsonPath jsonPath = response.jsonPath();
jsonPath.prettyPrint();
response.then()
.assertThat().statusCode(200) // Response status code is 200
.assertThat().body("code", equalTo(101)) // Response code is 101
.assertThat().body("message", equalTo("SampleShoe Added Successfully.")) // Response message is correct
.assertThat().body("shoe.id", equalTo(101)) // Shoe ID is 101
.assertThat().body("shoe.image", equalTo("image_url")) // Shoe image is correct
.assertThat().body("shoe.name", equalTo("SampleShoe")) // Shoe name is correct
.assertThat().body("shoe.category", equalTo("Running")) // Shoe category is correct
.assertThat().body("shoe.sizes", equalTo("9")) // Shoe sizes are correct
.assertThat().body("shoe.price", equalTo(1000)); // Shoe price is correct
}
}
```

```
eclipse-workspace - CAPSTONEPROJECT\src\test\java\RestAssured\POSTDemo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Java Declaration Console X Results of running class POSTDemo
terminated - PCSDemo [TestNG] C:\Users\hp\AppData\Local\Temp\testing-eclipse-518812495\testng-customsuite.xml
1055 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1055 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1058 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1058 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1058 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1058 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1061 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 1 ms.
1063 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1065 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1065 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1065 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1066 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1077 [main] INFO org.testng.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1330 [main] INFO org.testng.internal.Utils - [Utils] DynamicGraphHelper.createDynamicGraph() took 9 ms.
{
    "code": 101,
    "message": "SampleShoe Added Successfully.",
    "shoe": {
        "image": "image_url",
        "size": "9",
        "price": 1000,
        "name": "SampleShoe",
        "id": 101,
        "category": "Running"
    }
}
PASSED: RestAssured_POSTDemo.createShoeResourceWithParameters

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

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

4879 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.ExitCodeListener@300e3588: 0 ms
32°C Mostly cloudy
EPP 11:52 AM IN 24.08.2023
```

Here we can change manually to get output of the different shoes:

File Edit Source Refactor Navigate Search Project Run Window Help

Problems: 0 Active 0 Declaration 0 Console X Progress Results of running class POSTDemo Git Staging Godele Tasks Grade Executions

<terminated> [main] INFO [TestNG] C:\Users\hp.p2\workspace\CapstoneProject\src\test\java\RestAssured\POSTDemo.java - [pid: 14208]

1324 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodByPriority() took 0 ns.

1324 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1327 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodsByGroup() took 0 ns.

1328 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1328 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodsByGroup() took 0 ns.

1329 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1330 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodsByGroup() took 0 ns.

1331 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1337 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodsByGroup() took 0 ns.

1337 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1337 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.collectMethodsByGroup() took 0 ns.

1337 [main] INFO org.testng.internal.Utils - [Utils] MethodOnGroupHelper.sortMethods() took 0 ns.

1338 [main] INFO org.testng.internal.Utils - [TestNG] Running:

C:\Users\hp\AppData\Local\Temp\testng-eclipse-783884276\testng-customsuite.xml

1405 [main] INFO org.testng.internal.Utils - [Utils] DynamicGraphHelper.createDynamicGraph() took 10 ms.

{

 "code": 201,
 "message": "PowerStride Training Shoes Added Successfully.",
 "show": {
 "image": "4_.png",
 "size": "4",
 "price": 0000,
 "name": "PowerStride Training Shoes",
 "id": 481,
 "category": "Sports"
 }

}

PASSED: BestAssured.POSTDemo.createShoeResourceWithParameters

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

0000 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.internal.ExitCodeListener@6753fe3a: 0 ms

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

All in one code to form GET and POST Requests:

```
package RestAssured;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;
public class GET_GET_POST {
@BeforeMethod
public void setUp() {
RestAssured.baseURI = "http://localhost:9010"; // Update the base URI
}
```

```

@Test
public void getShoesServerResponse() {
    RestAssured
        .when()
        .get("/get-shoes")
        .then()
        .assertThat()
        .statusCode(200)
        .log().all();
}
@Test
public void getUsersServerResponse() {
    RestAssured
        .given()
        .when()
        .get("/get-users")
        .then()
        .assertThat()
        .log().all()
        .statusCode(200);
}
@Test
public void createShoeResourceWithParameters() {
    Response response = given()
        .param("id", 401)
        .param("image", "4.png")
        .param("name", "PowerStride Training Shoes")
        .param("category", "Sports")
        .param("sizes", "6, 7, 8, 9")
        .param("price", 6000)
        .when()
        .post("/add-shoe");
    JsonPath jsonPath = response.jsonPath();
    jsonPath.prettyPrint();
    response.then()
        .assertThat().statusCode(200)
        .assertThat().body("code", equalTo(101))
        .assertThat().body("message", equalTo("PowerStride Training Shoes Added
Successfully."))
        .assertThat().body("shoe.id", equalTo(401))
        .assertThat().body("shoe.image", equalTo("4.png"))
        .assertThat().body("shoe.name", equalTo("PowerStride Training Shoes"))
        .assertThat().body("shoe.category", equalTo("Sports"))
        .assertThat().body("shoe.sizes", equalTo("6, 7, 8, 9"))
        .assertThat().body("shoe.price", equalTo(6000));
}
}

```

```

eclipse-workspace - CAPSTONEPROJECT\src\test\java\RestAssured\GET_POST.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigator Search Project Run Window Help
Problems JavaDoc Declaration Console Progress Results of running class GET_POST Git Staging Gradle Tasks Gradle Executions
terminated - GET_POST [TestNG] C:\Users\hp\p2\pool\plugins\org.eclipse.jdt.core\openjdk\hotspot\src\full\win32x86_64\17.0.7.v20230425-1502\jre\bin\javaw.exe (24-Aug-2023, 12:05:25 pm - 12:05:35 pm) [pid: 13806]
1011 [main] INFO org.testng.internal.Utils - [Utils] DynamicGraphHelper.createDynamicGraph() took 12 ms.
{
    "code": 101,
    "message": "PowerStride Training Shoes Added Successfully.",
    "shoes": [
        {
            "image": "4.png",
            "sizes": "6, 7, 8, 9",
            "price": 6000,
            "name": "PowerStride Training Shoes",
            "id": 401,
            "category": "Sports"
        }
    ]
}
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 24 Aug 2023 06:35:35 GMT
Keep-Alive: timeout=60
Connection: keep-alive
{
    "code": 101,
    "message": "28 Shoes Fetched Successfully.",
    "shoes": [
        {
            "id": 101,
            "image": "1.png",
            "name": "Blueknee 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": "Black Stylish Formal Shoes",
            "category": "Formal",
            "sizes": "7, 8, 9",
            "price": 3500
        }
    ]
}

```

```

eclipse-workspace - CAPSTONEPROJECT\src\test\java\RestAssured\GET_GET_POST.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigator Search Project Run Window Help
Problems JavaDoc Declaration Console Progress Results of running class GET_GET_POST Git Staging Gradle Tasks Gradle Executions
terminated - GET POST [TestNG] C:\Users\hp\p2\pool\plugins\org.eclipse.jdt.core\openjdk\hotspot\src\full\win32x86_64\17.0.7.v20230425-1502\jre\bin\javaw.exe (24-Aug-2023, 12:05:25 pm - 12:05:35 pm) [pid: 13806]
Transfer-Encoding: chunked
Date: Thu, 24 Aug 2023 06:35:35 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": "Fiona Flynn",
            "email": "fiona@example.com",
            "password": "fiona123"
        },
        {
            "name": "Sia Sen",
            "email": "sia@example.com",
            "password": "sia123"
        }
    ]
}
PASSED: RestAssured.GET_POST_getShoesServerResponse
PASSED: RestAssured.GET_POST_createShoeResourceWithParameters
PASSED: RestAssured.GET_POST_getUsersServerResponse
-----
Default test
Tests run: 3, Failures: 0, Skips: 0
-----
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
-----
5723 [main] INFO org.testng.internal.Utils - [TestNG] Time taken by org.testng.internal.ExitCodeListener@7ef92c32: 0 ms

```

TestBase:

```

package RestAssured;
import org.apache.log4j.Logger;
import org.apache.log4j.Level;
import org.apache.log4j.PropertyConfigurator;
import org.testng.annotations.BeforeClass;
import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;
public class TestBase {
    public static RequestSpecification httpRequest;
    public static Response response;
    public Logger logger;
    @BeforeClass
    public void setUp() {
        logger = Logger.getLogger("LogDemo");
    }
}

```

```
PropertyConfigurator.configure("log4j.properties");
logger.setLevel(Level.DEBUG);
}
}
```

This is my GitHub link where I can push my files:

<https://github.com/Guntikala-Samyuktha-Rani/java-fsd.git>