```java
package com.sportyshoe.restAssuredScripts;


import org.testng.annotations.Test;


import io.restassured.RestAssured;


public class PostandPutnewShoe {


	@Test(priority='1')

	public void add_new_product()

	{


		RestAssured.given()

		.baseUri("http://localhost:9010")

		.basePath("/add-shoe")

		.queryParam("id","1020")

		.queryParam("image", "www.imge.com")

		.queryParam("name","Nike")

		.queryParam("category", "Running")

		.queryParam("sizes","5,6,7")

		.queryParam("price", "2000")

		.when()

		.post()

		.then()

		.log().all();


}


	@Test(priority='3')

	public void delete_product()

	{
```

```java
        RestAssured.given()

        .baseUri("http://localhost:9010")

        .basePath("/delete-shoe")

        .queryParam("id", "1020")

        .when().delete()

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


}
```

```java
@Test(priority='2')

public void update_a_product()

{

        RestAssured.given()

        .baseUri("http://localhost:9010")

        .basePath("/update-shoe")

        .queryParam("id","1020")

        .queryParam("image", "www.imge123.com")

        .queryParam("name","Reebok")

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

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

                .queryParam("price", "2500")

                .when()

                .put()

                .then()

                .log().all();


    }
}
package com.sportyshoe.restAssuredScripts;


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();
```

```java
        }



        @Test(priority='2')

        public void get_all_users()

        {


                RestAssured.given()

                .baseUri("http://localhost:9010")

                .basePath("/get-users")

                .when()

                .get()

                .then()

                .statusCode(401)

                .log()

                .all();


        }


}
```

package com.sportyshoe.Tests;


import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;


import com.sportyshoe.SeleniumCucumberScripts.AddtoCartPage;

import com.sportyshoe.SeleniumCucumberScripts.HomePage;

import com.sportyshoe.SeleniumCucumberScripts.LoginPage;

import com.sportyshoe.SeleniumCucumberScripts.RegisterPage;

import com.sportyshoe.SeleniumCucumberScripts.TestBase;

```java
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();

        }
```

```java
        @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);

        }




}
package com.sportyshoe.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.TestBase;


import static org.testng.Assert.assertEquals;


import org.testng.Assert;

import org.testng.Assert.*;


public class HomePageTest extends TestBase {


        HomePage hp;


        @BeforeTest
```

```java
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();

}
```

```java
        }
package com.sportyshoe.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.TestBase;
```

```java
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);
```

```java
        }


@Test(priority='3')


        public void Test_validate_registration_Text()

        {

                String expected = "Hello sonal !";

                String actualText = rp.validate_registration_Text();

                Assert.assertEquals(actualText, expected);

        }



@Test(priority='4')


public void test_click_cart()

{

        lp.click_cart();

}




}
package com.sportyshoe.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;
```

```java
import com.sportyshoe.SeleniumCucumberScripts.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);


        }


}
```

package com.sportyshoe.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.TestBase;


public class RegisterPageTest extends TestBase {


        HomePage hp;

        RegisterPage rp;


        @BeforeTest

```java
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')

public void Test_register_user()

{

        rp.register_user();

}


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

```java
        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 sonal !";

                String actualText = rp.validate_registration_Text();

                Assert.assertEquals(actualText, expected);

        }



}
package com.sportyshoe.SeleniumCucumberScripts;


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)
```

```java
        {
                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();

        }



}
package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;


public class SetupCheck {
```

```java
        @Test

        public void test_setup_selenium()

        {

                WebDriver driver = new ChromeDriver();


                driver.manage().window().maximize();


                driver.get("http://localhost:9010/");


                String text = driver.findElement(By.xpath("//div[@class='container mt-
3']/descendant::p[1]")).getText();


                System.out.println(text);


                System.out.println(driver.getTitle());

        }


}
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 TestBase{


        @FindBy(xpath="//input[@id='name']")

        WebElement registername;
```

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

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

```java
        }


        public String validate_registration_Text()

        {

                String user_name = userText.getText();

                return user_name;

        }
```
```java
}
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 {


        @FindBy(linkText="Orders")

        WebElement orderlink;
```

```java
        public OrderPage(WebDriver driver) {

                PageFactory.initElements(driver, this);

        }


        public void click_orderPage()

        {

                orderlink.click();


        }


}
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 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']")
```

```java
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("sonal");
        registeremail.sendKeys("sonal@gmail.com");
        registerpassword.sendKeys("sonal@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;
}
```

```java
}package com.sportyshoe.SeleniumCucumberScripts;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;


public class SetupCheck {



    @Test
    public void test_setup_selenium()
    {
            WebDriver driver = new ChromeDriver();


            driver.manage().window().maximize();


            driver.get("http://localhost:9010/");


            String text = driver.findElement(By.xpath("//div[@class='container mt-
3']/descendant::p[1]")).getText();
```

```java
				System.out.println(text);


				System.out.println(driver.getTitle());
		}


}
package com.sportyshoe.SeleniumCucumberScripts;


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();
```

```java
                driver.manage().deleteAllCookies();

                driver.get("http://localhost:9010/");


        }


        public static void closebrowser()
        {

                driver.close();

        }




}
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
        <classpathentry kind="src" output="target/classes" path="src/main/java">
                <attributes>
                        <attribute name="optional" value="true"/>
                        <attribute name="maven.pomderived" value="true"/>
                </attributes>
        </classpathentry>
        <classpathentry excluding="**" kind="src" output="target/classes"
path="src/main/resources">
                <attributes>
                        <attribute name="maven.pomderived" value="true"/>
                        <attribute name="optional" value="true"/>
                </attributes>
        </classpathentry>
        <classpathentry kind="src" output="target/test-classes" path="src/test/java">
                <attributes>
                        <attribute name="optional" value="true"/>
                        <attribute name="maven.pomderived" value="true"/>
```

```xml
            <attribute name="test" value="true"/>

        </attributes>

    </classpathentry>

    <classpathentry excluding="**" kind="src" output="target/test-classes"
path="src/test/resources">

        <attributes>

            <attribute name="maven.pomderived" value="true"/>

            <attribute name="test" value="true"/>

            <attribute name="optional" value="true"/>

        </attributes>

    </classpathentry>

    <classpathentry kind="con"
path="org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardV
MType/JavaSE-1.7">

        <attributes>

            <attribute name="maven.pomderived" value="true"/>

        </attributes>

    </classpathentry>

    <classpathentry kind="con" path="org.eclipse.m2e.MAVEN2_CLASSPATH_CONTAINER">

        <attributes>

            <attribute name="maven.pomderived" value="true"/>

        </attributes>

    </classpathentry>

    <classpathentry kind="output" path="target/classes"/>

</classpath>

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

    <name>ATE-CapstoneProject</name>

    <comment></comment>

    <projects>

    </projects>

    <buildSpec>
```

```xml
                    <buildCommand>

                            <name>org.eclipse.jdt.core.javabuilder</name>

                            <arguments>

                            </arguments>

                    </buildCommand>

                    <buildCommand>

                            <name>org.eclipse.m2e.core.maven2Builder</name>

                            <arguments>

                            </arguments>

                    </buildCommand>

            </buildSpec>

            <natures>

                    <nature>org.eclipse.jdt.core.javanature</nature>

                    <nature>org.eclipse.m2e.core.maven2Nature</nature>

            </natures>

</projectDescription>

<?xml version="1.0" encoding="UTF-8"?>


<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">

 <modelVersion>4.0.0</modelVersion>


 <groupId>ATE-CapstoneProject</groupId>

 <artifactId>ATE-CapstoneProject</artifactId>

 <version>0.0.1-SNAPSHOT</version>


 <name>ATE-CapstoneProject</name>

 <!-- FIXME change it to the project's website -->

 <url>http://www.example.com</url>
```

```xml
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.7</maven.compiler.source>
  <maven.compiler.target>1.7</maven.compiler.target>
</properties>

<dependencies>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.14.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.14.1</version>
</dependency>

        <dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.11</version>
  <scope>test</scope>
</dependency>

  <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
```

```xml
        <groupId>io.cucumber</groupId>

        <artifactId>cucumber-java</artifactId>

        <version>7.10.1</version>

</dependency>


    <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->

<dependency>

        <groupId>io.cucumber</groupId>

        <artifactId>cucumber-junit</artifactId>

        <version>7.10.1</version>

        <scope>test</scope>

</dependency>


    <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->

<dependency>

        <groupId>io.cucumber</groupId>

        <artifactId>cucumber-core</artifactId>

        <version>7.10.1</version>

</dependency>


<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->

<dependency>

        <groupId>io.rest-assured</groupId>

        <artifactId>rest-assured</artifactId>

        <version>5.3.0</version>

        <scope>test</scope>

</dependency>


<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured-common -->
```

```xml
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured-common</artifactId>
    <version>5.3.0</version>
</dependency>


<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured-all -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured-all</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>



<!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-path</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>



<!-- https://mvnrepository.com/artifact/org.json/json -->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20180813</version>
</dependency>
```

```xml
<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest -->
<dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest</artifactId>
    <version>2.2</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.7.1</version>
    <scope>test</scope>
</dependency>


<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.15.2</version>
</dependency>

    <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.11.0</version>
</dependency>


<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
    <groupId>io.github.bonigarcia</groupId>
```

```xml
      <artifactId>webdrivermanager</artifactId>

      <version>5.4.1</version>

  </dependency>


   </dependencies>


  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be
moved to parent pom) -->

      <plugins>

        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-
core/default-bindings.html#Plugin_bindings_for_jar_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-compiler-plugin</artifactId>

          <version>3.8.0</version>

        </plugin>

        <plugin>

          <artifactId>maven-surefire-plugin</artifactId>

          <version>2.22.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-jar-plugin</artifactId>
```

```xml
      <version>3.0.2</version>

    </plugin>

    <plugin>

      <artifactId>maven-install-plugin</artifactId>

      <version>2.5.2</version>

    </plugin>

    <plugin>

      <artifactId>maven-deploy-plugin</artifactId>

      <version>2.8.2</version>

    </plugin>

    <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_Lifecycle -->

    <plugin>

      <artifactId>maven-site-plugin</artifactId>

      <version>3.7.1</version>

    </plugin>

    <plugin>

      <artifactId>maven-project-info-reports-plugin</artifactId>

      <version>3.0.0</version>

    </plugin>

    </plugins>

  </pluginManagement>

 </build>

</project>
```