Automating the Star Health Application Using Selenium

Scenario 1 Cucumber + JUnit

Automating the Star Health Buy Now flow using Cucumber and JUnit.

Step 1: Create a Maven Project

1. Open your IDE (Eclipse/IntelliJ).

2. Create a new Maven project With Maven-Archtype-Quickstart .

3. Provide a group and artifact ID.

4. Click "Finish."

Step 2: Add Dependencies in POM.XML

1. Open the pom.xmlfile.

2. Add dependencies for Cucumber, JUnit, Selenium, and required libraries.

```
<dependencies>
	<dependency>
		<groupId>junit</groupId>
		<artifactId>junit</artifactId>
		<version>4.11</version>
		<scope>compile</scope>
	</dependency>
	<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
	<dependency>
		<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>compile</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>
        <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>
                <artifactId>webdrivermanager</artifactId>
                <version>5.4.1</version>
        </dependency>
        <dependency>
                <groupId>tech.grasshopper</groupId>
                <artifactId>extentreports-cucumber7-adapter</artifactId>
                <version>1.7.0</version>
        </dependency>
</dependencies>
```

Step 3: Create a Cucumber Feature File : starHealth.feature

Step 4: Write a Cucumber Scenario

```
Feature: Automating Star Health Application
@starhealth
  Scenario Outline: Validate the Star Health Buy Now flow
    Given User launches the Star Health application with "<URL>"
    And User waits for the Welcome to Star Health pop-up and closes it
    And User validates the Star Health home page title using a JUnit assertion
    And User clicks on the Buy Now button
    And User types Name as "<FullName>"
    And User types Phone as "<PhNo>"
    And User types the PIN as "<PIN>"
    And User clicks on I need health insurance from the drop-down menu
    And User selects the option "<OptionPlan>"
    And User sees the Plan for My Family page
    And User validates that the mobile number is the same as the previously entered number using a JUnit assertio
    And User clicks on the Star Health logo
    And The Application should redirect to the home page
    And User closes the child tab
    And User navigates back to the parent tab

    Examples:
      | URL                       | FullName | PhNo       | PIN    | OptionPlan |
      | https://www.starhealth.in/ | Karthick | 7397001623 | 636008 | Myself     |
```

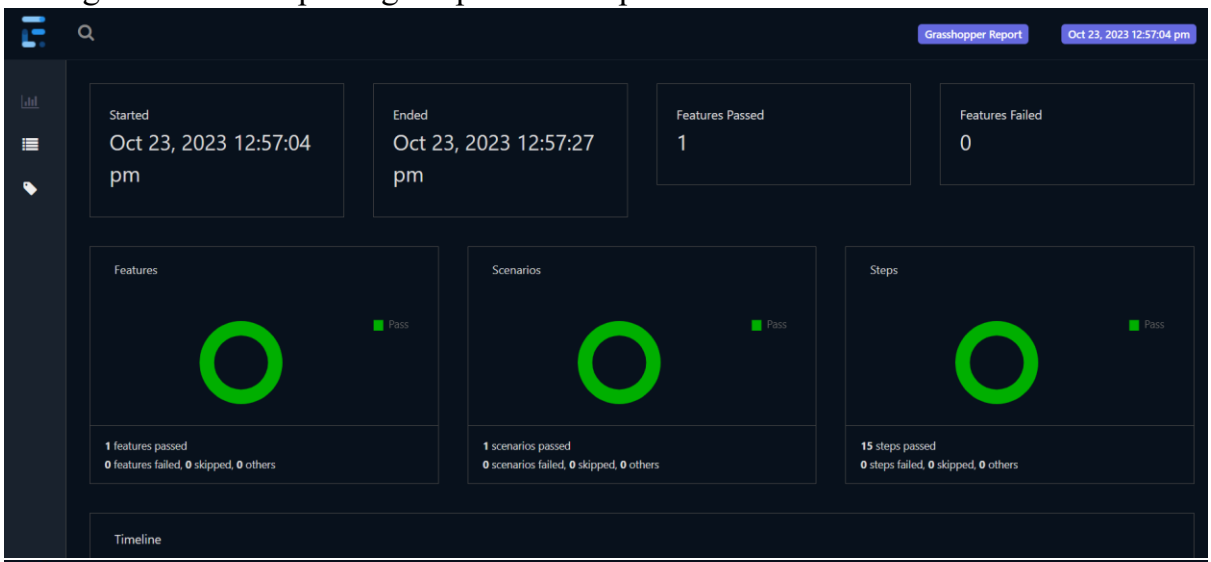
Step 5: Parameterize Test Data

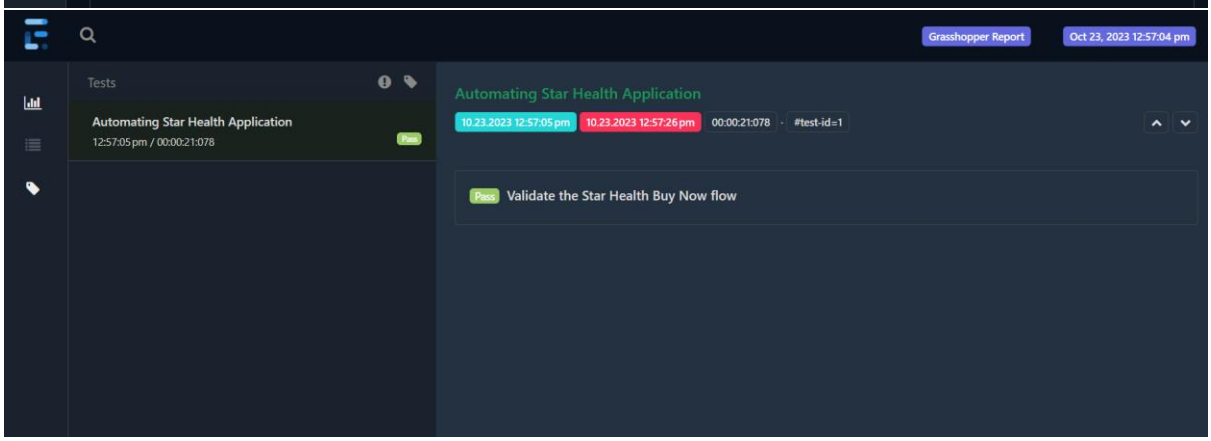
1. Parameterize test data using the Cucumber Table.

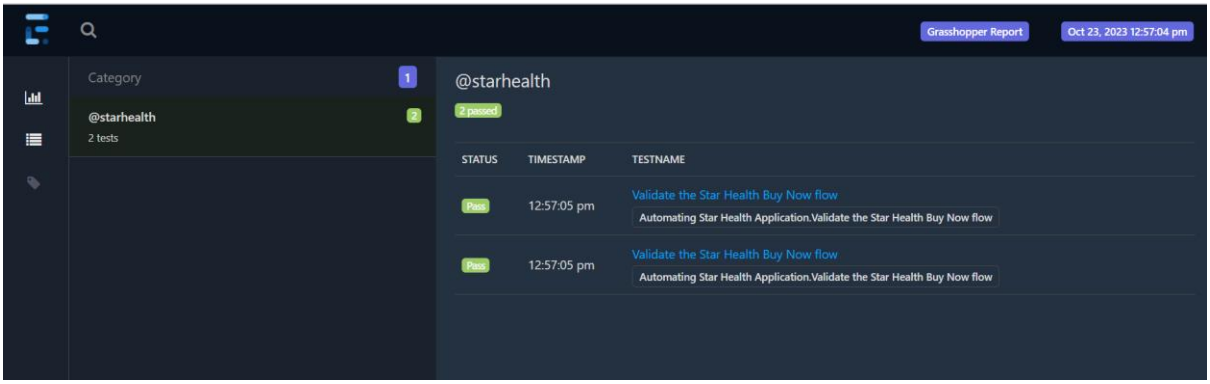
2. Define examples for the scenario.

Step 6: Use Extend Reporting

1. Configure Extend Reporting for pass/fail steps.



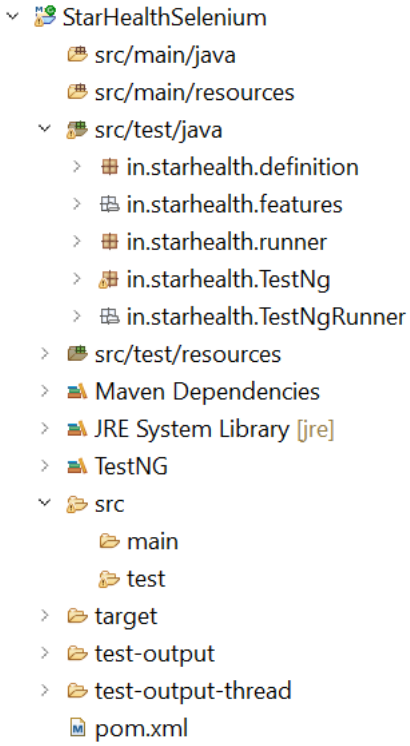




Step 7: Implement Page Object Model

1. Create page objects for the web elements.

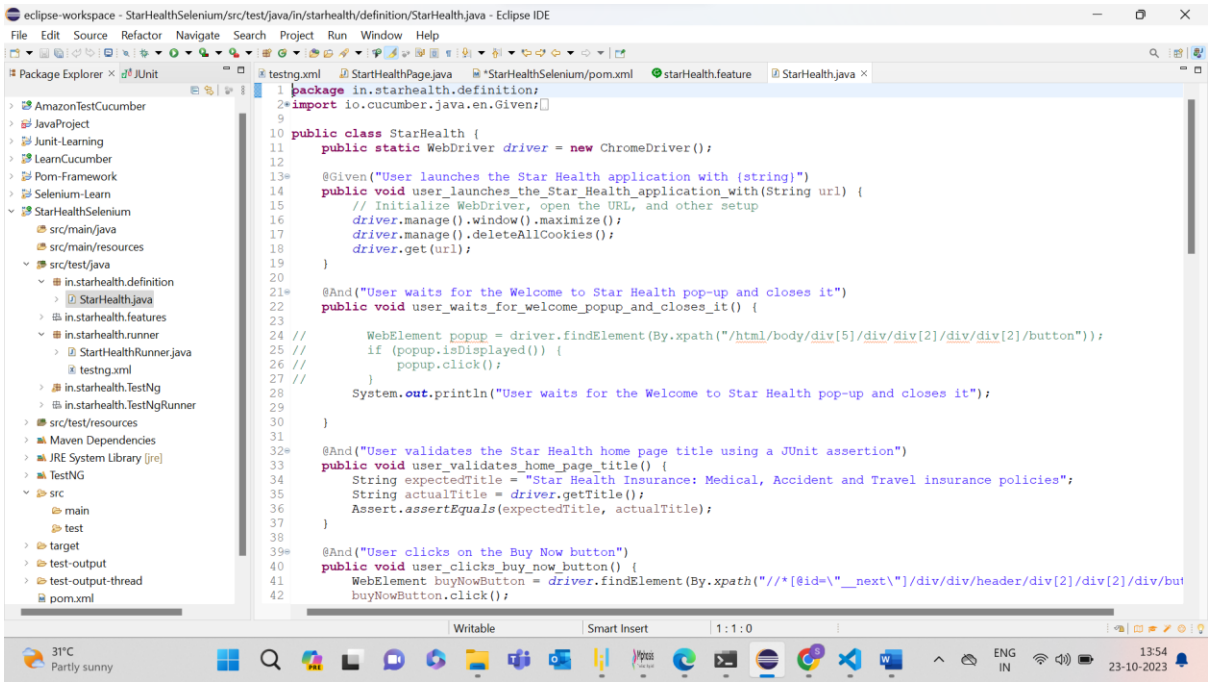
2. Implement Page Object Model design pattern for a clean structure.



Step 8: Implement Step Definitions

1. Create a step definition file.

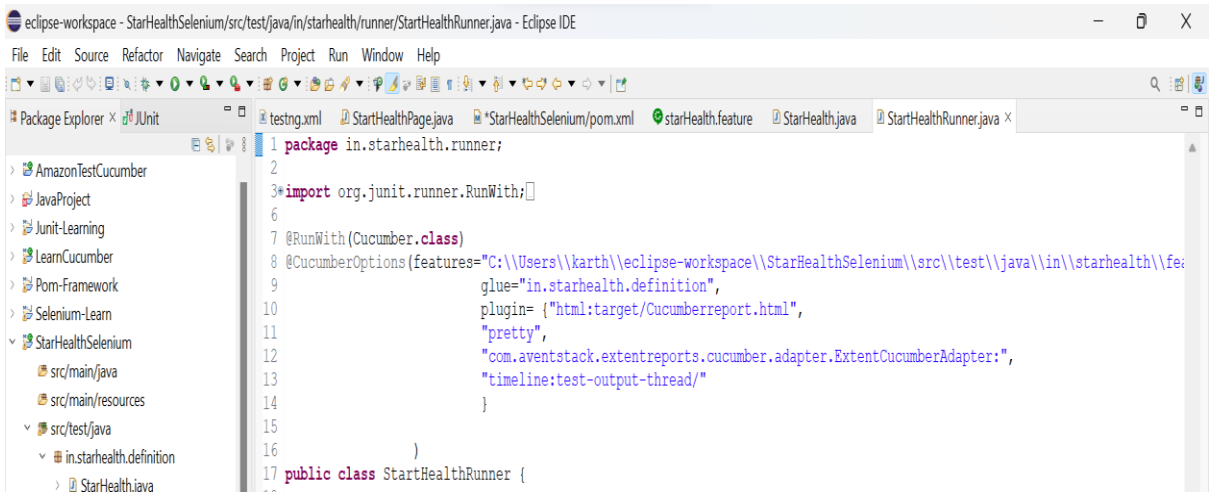
2. Implement step definitions using Selenium and JUnit assertions.



Step 9: Create a Test Runner

1. Create a test runner class to run the Cucumber tests.

2. Configure the runner to execute scenarios from starHealth.feature.



Step 10: Run Tests with Maven

1. Use Maven to run the Cucumber tests.

2. Verify test results

```
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.1</version>
     <configuration>
     <includes>
         <include>**/*StartHealthRunner.java</include>
     </includes>
  </configuration>
</plugin>
```

.

Step 11: Set Up Jenkins Integration

1. Create a FreeStyle Jenkins project.

2. Configure Jenkins to run this Cucumber project.

3. Trigger the Jenkins job and observe the results.

Scenario 2 TestNG + Selenium:

Automating various tasks using TestNG and Selenium.

Step 1: Create a testng.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
<parameter name="name"  value="Karthick"/>
<parameter name="mobile"  value="7397001345"/>
<parameter name="email"  value="karthick@gmail.com"/>
    <classes>
      <class name="in.starhealth.TestNg.StartHealthPage"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

1. In the Maven project, create a testng.xml file.

2. Define a TestNG suite and test cases.

Step 2: Implement TestNG Test

1. Create a Java class for the TestNG test.

2. Implement test methods and assertions.

3. Use Extend Reporting for pass/fail steps.

Step 3: Configure TestNG Annotations

1. Use @BeforeClass and @BeforeSuite for setup.

2. Use @AfterClass and @AfterSuite for cleanup.

Step 4: Parameterize Data

1. Use parameterized data for test inputs, possibly from an external source like testNG.xml or Excel.

Step 5: Page Object Model

1. Implement the Page Object Model for a clean structure.

2. Create page objects for web elements.

Step 6: Run TestNG Scenarios

1. Run TestNG scenarios using the testNG.xml file.

2. Ensure tests execute as expected.

Step 7: Parallel Execution

1. Create another TestNG test in testng.xml.

2. Configure it to run in parallel with the first test.

3. Execute both tests concurrently.

Step 8: Git Version Control

1. Upload the entire code to a Git repository.

Step 9: Jenkins Integration

1. Configure Jenkins to fetch the code from Git.

2. Set up Jenkins to execute the TestNG tests through Jenkins jobs.

3. Run the Jenkins job and observe results.

GitHub repository : https://github.com/Karthick-Office/Project01.git

Automating the Star Health Application Using Selenium Project is present under the “Selenium Phase 2 End Project” folder in the My GitHub repository