Capstone Project: Create a Testing Framework for Sporty Shoes Website

GITHUB LINKS:

git@github.com:kalalabhilash1/Testing-1.git

https://github.com/kalalabhilash1/Testing-1/tree/main/Capstone%20Practice%20Project

AIM: To Create a Testing Framework for Sporty Shoes website

Procedure:

Step-by-Step Process for Creating a Testing Framework for Sporty Shoes Website

1. Setup and Configuration

a. Download and install the necessary tools and software:

i. Eclipse IDE for source code editing and modification

ii. Selenium WebDriver (Java version) for browser-based end-user testing

iii. TestNG framework for test case management and execution

iv. JMeter for load testing

v. Cucumber for API testing using Gherkin syntax

vi. Postman for API testing

vii. Rest-Assured for API testing in Java

viii. Git for version control and collaboration

ix. GitHub for code hosting and repository management

b. Download the Sporty Shoes project from GitHub:

i. Navigate to the Simplilearn GitHub repository: https://github.com/Simplilearn-Edu/SportyShoes

ii. Clone or download the project to your local system

iii. Extract the project files and navigate to the project directory

c. Run the Sporty Shoes application:

i. Open a terminal or command prompt and navigate to the project directory

ii. Execute the command: java -jar project_name.jar

iii. The Sporty Shoes application should start running

2. API Testing with Rest-Assured

a. Create a new Java project in Eclipse IDE

b. Add the necessary dependencies to the project:

i. Add the Rest-Assured library to the project's classpath

ii. Add any other required libraries for working with JSON and HTTP requests

c. Create Rest-Assured test classes:

i. Create separate test classes for each API endpoint

ii. Use Rest-Assured methods to send HTTP requests to the API endpoints

iii. Assert the responses to verify the API functionality

d. Execute the Rest-Assured test classes:

i. Run the test classes in Eclipse to execute the test cases

ii. Verify that the API endpoints are functioning correctly

3. Web Application Testing with Selenium WebDriver and TestNG

a. Create a new Selenium project in Eclipse IDE

b. Configure Selenium WebDriver:

i. Add the Selenium WebDriver library to the project's classpath

ii. Set up the browser driver for the desired browser (e.g., Chrome, Firefox)

c. Create Selenium test classes:

i. Create separate test classes for each web page or user flow

ii. Use Selenium WebDriver methods to interact with web elements

iii. Assert the behavior of the web pages to verify the application functionality

d. Create TestNG test suites:

i. Create TestNG test suites to group related test cases

ii. Configure TestNG annotations to control test execution and reporting

e. Execute the Selenium test classes:

i. Run the test suites in Eclipse to execute the test cases

ii. Verify that the web pages are functioning correctly

4. Load Testing with JMeter

a. Create a new test plan in JMeter:

i. Open JMeter and create a new test plan

ii. Add HTTP Request samplers for the homepage and product detail page

iii. Configure the samplers with the appropriate URLs and parameters

b. Configure load testing parameters:

i. Set the number of threads to simulate concurrent users

ii. Set the ramp-up time to gradually increase the load

iii. Set the duration to determine the total test duration

c. Execute the load test:

i. Run the test plan to simulate user load on the specified pages

ii. Monitor the results to identify performance bottlenecks

d. Analyze the load test results:

i. Analyze the response times, throughput, and error rates

ii. Identify areas for performance improvement and optimize the application

5. API Testing with Cucumber

a. Set up Cucumber in the Java project:

i. Add the Cucumber Maven plugin to the project's pom.xml file

ii. Configure Cucumber with the appropriate features and step definitions

b. Create Feature Files:

i. Write Feature Files in Gherkin syntax to describe the API testing scenarios

ii. Use feature keywords (Background, Feature, Scenario, Given, When, Then) to structure the scenarios

c. Implement Step Definitions:

i. Create Java classes containing step definitions for each step in the Feature Files

ii. Use Cucumber annotations to map step definitions to Gherkin steps

d. Execute the Cucumber test scenarios:

i. Run the Cucumber test runner to execute the scenarios

Steps to complete the Capstone project:

To Launch the Application under test

- Go to github URL https://github.com/Simplilearn-Edu/SportyShoes.git

- Click on Green button which has Code written on it. Click on Download Zip
- The download zip file will be in your download folder
- Extract all the files in downloads folder

Go inside the folder SportyShoes-main and then again go in the folder SportyShoes-main

Go inside the folder where you see the sporty-shoes-v1 jar file

Start cmd prompt in the same folder

Execute the java command to run the jar file and deploy the application

java -jar sporty-shoes-v1.jar

OR

java -jar -Dserver.port=8100 sporty-shoes-v1.jar