

Capstone Project: Create a Testing Framework for the Sporty shoes Website Project

objective:

To develop a comprehensive QA and test suite for the Sportyshoes website. The QA effort will require the following:

1. Browser-based end-user testing using Selenium WebDriver with TestNG framework.
2. Load testing using JMeter.
3. API testing with Cucumber
4. API testing with Postman and Rest Assured The end-deliverables will be executable scripts and modules that can be run on demand for testing the Sportyshoes website.

Write-up

Algorithm Browser-based end-user testing using Selenium WebDriver with TestNG framework

Step 1: Create a Maven Project.

Step 2: Set up the project structure with the src/main and src/test directories.

Step 3: Edit the pom.xml file for Selenium WebDriver, TestNG, and any other necessary libraries. After that save the pom.xml file, and Maven will automatically download the dependencies.

Step 4: Create test class for each functionality (e.g., LoginTest, RegistrationTest, productpage, homepage etc.) under the src/test/java directory.

Step 5: Create a base test class

Step 6: Implement test methods in each test class corresponding to the pages (Homepage, loginpage, registerpage, productpage).

Step 7: Create a testng.xml file to manage test suite configurations.

Step 8: Run your tests.

Load testing using JMeter.

Step 1: Open JMeter and create a new test plan.

Step 2: Add a Thread Group to simulate virtual users and set the desired number of threads, ramp-up time, and loop count.

Step 3: Add HTTP Request Samplers for the different pages and functionalities of the Sportyshoes website.

Step 4: Set up JMeter to record your interactions with the Sportyshoes website. Start recording and navigate through the website to capture the test scenarios.

Step 5: Include assertions to verify the correctness of the server responses and include timers to simulate realistic user think times and pacing.

Step 6: Configure JMeter to distribute the load across multiple machines if needed.

Step 7: Execute the load test and monitor the results in real-time.

Step 8: Analyse the results to identify performance bottlenecks and areas for improvement.

Step 9: Use JMeter plugins or other reporting tools to generate comprehensive reports.

API testing with Cucumber

Step 1: Create a Maven Project:

Step 2: Set up the project structure with the `src/main` and `src/test` directories.

Step 3: Edit the `pom.xml` file to include dependencies for Cucumber, Rest Assured, and TestNG. Save the `pom.xml` file, and Maven will automatically download the dependencies.

Step 4: Create Cucumber feature file under the `src/test/resources` directory for each API endpoint.

Step 5: Implement step definitions for each step in the feature files.

Step 6: Create a runner file to run the cucumber feature file.

Step 7: Run your Cucumber tests.

API testing with Postman

Step 1: Open Postman and create a new collection for Sportyshoes API tests.

Step 2: Create individual requests within the collection for each API endpoint. Example for "Retrieve the list of all products": • Request Name: Get All Products • Request Type: GET • Endpoint: `https://api.sportyshoes.com/products` • Create similar requests for other API endpoints.

Step 3: Write tests for each request to validate the response.

Step 4: Execute your Postman tests individually or as part of a collection.

Step 5: Postman automatically generates test reports.

View reports in the Postman app or **export** them for further analysis.

API testing with Rest Assured

Step 1: Create a New Maven Project:

Step 2: Edit the `pom.xml` file to include dependencies for Rest Assured, TestNG, and any other necessary libraries. Save the `pom.xml` file, and Maven will automatically download the dependencies.

Step 3: Write Rest Assured Test Cases

Step 4: Create separate test classes for each API endpoint under the `src/test/java` directory.

Step 5: In each test class, initialize Rest Assured with the base URL.

Step 6: Implement test methods in test class corresponding to the API endpoints. Retrieve the list of all products: Retrieve the list of all registered users: Add a product: Delete a product: Update a product

Step 7: Run your API tests.