

Capstone Project: Create a Testing Framework for Sporty Shoes Website

WriteUp:

Project objective:

The objective is to develop a comprehensive QA and test suite for the Sporty Shoes 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 Sporty Shoes web app.

Background of the problem statement:

Sporty Shoes has an e-commerce website that has the following existing features in place:

- Users can view products.
- If users want to purchase something, they can first sign up and then log in.
- Users can add multiple items to their cart and do a checkout.
- Users have a dashboard that lets them edit their profile, view past purchases, and view their cart.
- Once users do a checkout, the items are cleared from their cart and an order is generated which is stored in their order history.

The above application is already functional. What is needed now is to add a testing layer that will ensure that everything is passed through QA.

Implementation Requirements

The following deliverables are expected:

1. Automate the below API endpoints using Rest-Assured
 - Retrieve the list of all products in the store.
 - Retrieve the list of all registered users.
 - Add the product.
 - Delete the product.
 - Update the product.
2. Create Selenium scripts using TestNG to test all the pages in the web app that will automate:
 - Login page
 - Registration Page
 - Add Product to cart page.
 - Place Order Page
3. Create JMeter scripts to do load testing of the homepage and the product detail page.
4. Setup Cucumber in Java Project and write Feature Files using Gherkin to test the API endpoints mentioned in point 1 above.

5. Create Postman scripts to test the following API endpoints:

- Retrieve the list of all products in the store.
- Retrieve the list of all registered users.
- Add the product.
- Delete the product.
- Update the product.

API Endpoints:

Action	Method	Endpoint
Retrieve the list of all products in the store	GET	http://localhost:9010/get-shoes
Retrieve the list of all registered users	GET	http://localhost:9010/get-users
Add the product	POST	http://localhost:9010/add-shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=1000
Delete the product	DELETE	http://localhost:9010/delete-shoe?id=101
Update the product	PUT	http://localhost:9010/update-shoe

Add Product URL Sample with POST Method:

http://localhost:9010/add-shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=1000

Update Product URL Sample with PUT Method:

Request body (<http://localhost:9010/update-shoe>)

```
{  
  "id": 101,  
  "name": "Updated Shoe Name",  
  "category": "Updated Category",  
  "sizes": "8,9,10",  
  "price": 1500  
  "image": "updated_image_url",  
}
```

Delete Product URL Sample with DELETE Method:

<http://localhost:9010/delete-shoe?id=101>

You must use the following:

1. Source code editing and modification: Eclipse IDE
2. End-User Black Box Testing: Selenium WebDriver (A Browser testing framework where only the Java version is used.) with TestNG Framework.
3. Load Testing: JMeter (A load testing application for Java applications.)
4. API Testing: Cucumber (The Gherkin syntax used in Cucumber allows you to define test scenarios in a natural language format)
5. API Testing: Postman (A standalone application that allows testing of API-based services.) and Rest-Assured.
6. Git: To connect and push files from the local system to GitHub
7. GitHub: To store the application code and track its versions.
8. Specification document: Any open-source document or Google Docs

The following requirements should be met:

1. All testing scripts and code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
2. The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link to the repository. You can add a section in your document.
3. Document the step-by-step process starting from creating test cases and then executing them and recording the results.
4. You need to submit the final specification document which should include:
 - Project and tester details
 - Concepts used in the project.
 - Links to the GitHub repository to verify the project completion.
 - Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

Project to be tested:

Download the project from <https://github.com/Simplilearn-Edu/SportyShoes>

Then execute:

java -jar project_name.jar

GIT URL : <https://github.com/harshithanap/Capstone-Project-Sporty-Shoes->