

Sourcecode :

Step 1: Setting up the Environment :

1. Install JMeter 5.1.1 and Java Development Kit (Version 8) on your system.

Step 2: Recording Test Scenario :

1. Launch JMeter and create a new Test Plan.
2. Add a Thread Group to the Test Plan.
3. Add an HTTP Request Defaults and configure the base URL of the e-learning application.
4. Add an HTTP Cookie Manager to handle session cookies.
5. Start recording the test scenario using JMeter's HTTP(S) Test Script Recorder.
6. Configure your browser to use JMeter as a proxy and navigate through the e-learning application, performing actions such as logging in, accessing courses, etc.
7. Stop the recording once you've captured the desired user actions.

Step 3: Refining the Test Plan :

1. Review the recorded requests and adjust any parameters or settings as needed.
2. Add Assertions to verify the responses and ensure the application behaves as expected.
3. Parameterize any dynamic values like user credentials to simulate multiple users.
4. Add timers to simulate realistic user behavior, such as think time between actions.

Step 4: Running the Test :

1. Validate the Test Plan by running it with a small number of users and checking for errors.
2. Gradually increase the number of users and iterations to simulate load on the application.
3. Monitor system resources and performance metrics during the test to identify bottlenecks.

Step 5: Analyzing Results :

1. Once the test is complete, analyze the results to identify any performance issues.
2. Generate reports and graphs using JMeter's built-in reporting tools.
3. Identify areas for optimization and fine-tuning in the e-learning application.

Step 6: Documenting and Tracking :

1. Document the entire process, including configuration settings, test scenarios, and results analysis.
2. Track source code changes in your GitHub repository, ensuring to include necessary files while ignoring any sensitive data or configuration files.
3. Write detailed README.md file explaining how to run the test and interpret the results.
4. Share the GitHub repository link with the stakeholders for review and tracking purposes.

Step 7: Finalizing and Submission

1. Review the completed work and ensure all requirements are met.
2. Make any necessary refinements or corrections based on feedback.
3. Submit the GitHub repository link as per the submission guidelines.