Performance Test Plan

For

Simple Gallery System

**Document Creation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Description/Change** |
| 1.0 | Victor Kipngetich | 12/10/2024 | Initial draft |

**Approvals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Approved by** | **Title** | **Signature** | **Date** |
|  |  |  |  |  |

Table of Contents

[2 Introduction 3](#_Toc179899319)

[3 Objectives 3](#_Toc179899320)

[4 Scope testing 4](#_Toc179899321)

[5 Test Environment 4](#_Toc179899322)

[6 Test Scenarios 4](#_Toc179899323)

[6.1 load Testing 4](#_Toc179899324)

[6.1.1 Create new test plan 5](#_Toc179899325)

[6.1.2 Add thread group (Users simulation) 5](#_Toc179899326)

[6.1.3 Add HTTP Request Sampler for Web Request 5](#_Toc179899327)

[6.1.4 Add Listeners 6](#_Toc179899328)

[6.1.5 Run the test 6](#_Toc179899329)

[6.1.6 Analyze the result 7](#_Toc179899330)

[6.2 Stress Testing 7](#_Toc179899331)

[6.2.1 Response Time Testing 7](#_Toc179899332)

[7 Test Execution Plan 8](#_Toc179899333)

[10.Test Activity Schedule 9](#_Toc179899334)

# Introduction

The goal of this performance test plan is to define a structured approach to testing the website's performance under varying levels of user load. This plan will focus on simulating realistic user behaviour to evaluate the website's ability to handle expected and extreme traffic conditions. The primary objective is to ensure that the website maintains optimal performance during peak usage periods, minimizing downtime and user disruptions.

# Objectives

The primary objective of this load test is to evaluate the performance of the website under a simulated load of multiple users. This test will help identify potential bottlenecks, latency issues, or performance constraints by:

* Measuring response times under varying user loads.
* Assessing server stability and resource utilization during high traffic.
* Identifying the maximum number of concurrent users, the system can handle before performance degrades.

# Scope testing

The load testing will cover the following critical areas of the website:

* **Load Testing**: This will assess the maximum number of users the website can handle under normal operating conditions. It will help ensure that the website can efficiently support regular traffic loads without performance degradation.
* **Stress Testing**: This will evaluate how the website behaves under extreme conditions, such as sudden spikes in traffic beyond normal capacity. The goal is to identify the breaking point and observe how the system recovers from failures.
* **Response Time Testing**: This will measure how fast the website responds to user requests, ensuring that users experience minimal delays during navigation, transactions, and other key operations, even under load.

# Test Environment

* **Test Tool**: Apache, JMeter, Firebase console, Postman (API Rest and SOAP), Selenium (python), ALM, JIRA, Terminal, and GitHub
* **Environment**: The test will be executed in a staging environment that mirrors the production setup.
* **Users**: Virtual users will be simulated using Meter’s Thread Group.
* **Protocols**: HTTP/HTTPS
* **Servers**: Test will be run on the staging server, which has the same capacity as the production server (QA Environment).

# Test Scenarios

## load Testing

* **Objective**: To determine the maximum number of users the website can handle under normal conditions.
* **Test Case**:
  + Simulate 100 concurrent users.
  + Gradually increase the load and monitor performance.
  + Verify key pages and functionalities like homepage, login, search, and checkout.

### **Create new test plan**

Open **JMeter** and click on **File > New** to create a new test plan and rename the test plan **as login Page Load Test**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Element | Configuration | Testing Type(load/stress) | Status(pass/Fail/Not executed) |
| Number of threads users to simulates | 100,200,500,20000,25000 and above users |  |  |
| Ramp-Up period | 10 seconds (users will start over a 10-second period) |  |  |
| loop count | 1 (each user will send one request) |  |  |

### **Add thread group (Users simulation)**

A **Thread group** represent a pool of users sending request

* Right click on Test plan >Add >threads >thread group
* **Configure Thread Group:**

### **Add HTTP Request Sampler for Web Request**

Right click on the **Thread group > Add > Sampler >HTTP Request**

* **In the HTTP Request sampler**

1. **Server name or IP**: Enter the URL of the applications
2. **Methods:** choose the HTTP Method GET, POST, PUT, DELETE, OPTIONS
3. **Path:** Specify the endpoint of the test
4. **Add the Parameter**: For username and password

|  |  |
| --- | --- |
| Test Element | Configuration |
| **Server name or IP** | **localhost** |
| **Methods** | **POST** |
| **Path** | **/gallery/login.php** |
| **Parameter Username and passwords** |  |

### **Add Listeners**

Listeners allow you visualize test results

* Right click on the **thread group >add >listener**

1. **View Results:** Display result in table
2. **View results in graph: Graphical** view of the performance metrics
3. **Aggregate report:** summarizes average response time <throughout and performance metric

|  |  |
| --- | --- |
| Test Element | Configuration |
| Listener | View Results, View results in graph Aggregate report, summary Graph, Response time Graph, Response Latency graph, Response time percentage, Aggregate graph |

### **Run the test**

* Click the **start** button (green) in the toolbar to start load test
* JMeter will simulate the defined number of users accessing the login page, submitting request, results and status

### Analyze the result

Open the selected listener to view the test results

* **View Results**
* **View results in graph**
* **Aggregate report/Error rate**
* **Response time graph**
* **Latency graph**

**Success Criteria**:

* Average response time should be below 2 seconds for all pages.
* Error rate should be less than 1%.

## **Stress Testing**

* **Objective**: To analyze how the website performs under extreme load conditions above expected traffic.
* **Test Case**:
  + Simulate a higher number of users 2000,2500,3000 to stress the system.
  + Evaluate how the website behaves when traffic exceeds normal load.
  + Check if the website crashes, times out, or generates errors.
  + Identify the breaking point where the website starts failing.

**Success Criteria**:

* Identify the maximum number of users before the system fails.
* Evaluate how quickly the system recovers after a failure.
* Response time and throughput should degrade gracefully under stress.

### **Response Time Testing**

* **Objective**: To measure how fast the website responds to requests during different traffic conditions.
* **Test Case**:
  + Record the time it takes to respond to key actions login, search, checkout.
  + Simulate 100 users to assess response time under load.
  + Measure average and peak response times for each action.

**Success Criteria**:

* Average response time for each action should be below 3 seconds.
* 95% of requests should complete within the threshold.
* Minimal performance degradation with increasing users.

# **Test Execution Plan**

* **Load Testing**:
  + Test with 100,200,500 users.
  + Gradually increase user load and observe performance metrics.
* **Stress Testing**:
  + Stress the system with 20000,25000,30000 and beyond to identify the breaking point.
  + Capture the systems response under heavy load and failures.
* **Response Time Testing**:
  + Test with normal (100 users) and peak (500+ users) loads.
  + Capture the response time for login, search, and checkout pages.

#### **8. Risks and Assumptions**

* **Risks**:
  + Server may crash during stress testing, causing downtime in the test environment.
  + Network or infrastructure limitations could skew performance results.
* **Assumptions**:
  + The staging environment is a mirror of the production environment.
  + No other tests or activities will be running during the performance test.
  + Adequate monitoring tools are available to collect performance metrics.

#### ****9.** Reporting and Recommendation**

After completing the tests, a report will be generated that includes:

* **Summary of Test Execution**: Detailing the user load, performance metrics, and any identified bottlenecks.
* **Response Time Graphs**: Showing the time taken for key transactions.
* **Throughput and Error Rates**: Indicating server behaviour under load.
* **Recommendations**: Suggested optimizations based on the test results.

# 10.Test Activity Schedule

|  |  |  |
| --- | --- | --- |
| **#** | **Activity** | **Number of Days** |
| 1 | Test data creation | 2 (11th October 2024) |
| 2 | Test script creation/debugging | 4 (12th October 2024) |
| 3 | Test Execution and reporting | 2 (12th October 2024) |

#### ****10.** Conclusion**

This performance test plan will help ensure the website can handle user traffic efficiently, provide a smooth user experience under load, and maintain stability even under stress conditions. The insights gained will help optimize the system’s performance before deployment into production.

**LAST PAGE**