Performance Test Plan

## 1. Introduction

This document outlines the performance testing strategy for the eCommerce Web Application. The goal is to assess the system's responsiveness, stability, and scalability under various load conditions.

## 2. Performance Test Objectives

- Ensure the system meets performance requirements for 1000 concurrent users.  
- Identify bottlenecks and scalability issues.  
- Assess response times, throughput, and resource utilization under load.

## 3. Performance Metrics

| Response Time | Must be less than 2 seconds under normal load. |
| --- | --- |
| Throughput | System should handle 200 transactions per second. |
| CPU Usage | Should not exceed 80% under peak load. |
| Memory Usage | Should not exceed 70% under peak load. |
| Error Rate | Should not exceed 1% of total transactions. |
| Concurrent Users | Must support up to 1000 users without degradation. |

## 4. Test Scope

In-Scope: Login, product search, checkout process, API response times.  
Out-of-Scope: Third-party payment gateways, mobile app performance.

## 5. Test Approach

- Load Testing: Evaluating system behavior under normal and peak loads.  
- Stress Testing: Identifying system failure points by overloading.  
- Scalability Testing: Determining if the system can scale with increasing load.  
- Endurance Testing: Evaluating long-duration performance stability.  
- Spike Testing: Checking system response to sudden surges in load.

## 6. Test Environment

- Hardware Configuration: 16-core CPU, 64GB RAM, SSD storage.  
- Software Configuration: Linux OS, MySQL Database, Node.js backend.  
- Network Configuration: Simulated WAN with 1Gbps bandwidth.  
- Test Data: 100,000 user accounts, 500,000 product listings.

## 7. Performance Test Tools

- Load Testing Tool: JMeter  
- Monitoring Tools: Grafana, New Relic, AWS CloudWatch  
- Scripting Language: JavaScript (for custom test scripts in k6)

## 8. Test Execution Plan

- Entry Criteria: Environment setup, test data uploaded, test scripts ready.  
- Execution Schedule: Load tests from Feb 5 to Feb 10, stress tests on Feb 11-12.  
- Exit Criteria: All critical scenarios tested, acceptable response times met.

## 9. Risk Assessment & Mitigation

- Potential Risk: Hardware limitations → Mitigation: Use cloud-based testing infrastructure.  
- Potential Risk: Network latency issues → Mitigation: Simulate different network conditions.  
- Potential Risk: High defect count → Mitigation: Prioritize fixes before load testing.

## 10. Reporting & Analysis

- Test Logs & Reports: Response time trends, server utilization graphs.  
- Defect Tracking: Issues logged in JIRA, categorized by severity.  
- Recommendations: Optimize database queries, add caching for static content.

## 11. Approval & Sign-Off

Reviewed by: QA Lead, DevOps Engineer  
Approved by: Project Manager, Product Owner