

## **Project Title:**

**“Performance Testing on OrangeHRM Web Application”**

## **Website/Endpoint:**

- **Website URL:** <https://opensource-demo.orangehrmlive.com/>
- **Pages for Testing:**
  - Login Page: /index.php/auth/login
  - Dashboard: /index.php/dashboard
  - Employee Information: /index.php/pim/viewEmployeeList
  - Leave Management: /index.php/leave/viewLeaveList
  - Recruitment Module: /index.php/recruitment/viewCandidates
  - Payroll: /index.php/pim/viewPayroll
  - Admin Module: /index.php/admin/viewSystemUsers

## **Project Overview:**

The goal of this project is to conduct a comprehensive performance test of the **OrangeHRM** web application. This will include testing the key functionalities such as login, employee management, leave applications, recruitment, and payroll operations under different loads. The results will help in identifying bottlenecks, ensuring the application can handle peak user traffic, and assessing the system's overall stability.

---

## **Project Scope:**

The scope of the performance testing includes:

1. **Load Testing:** Evaluating the application's performance under regular user loads (e.g., employee login, updating profiles, applying for leave).
2. **Stress Testing:** Pushing the system to its limits by increasing the user load to identify when performance starts to degrade.
3. **Spike Testing:** Measuring the response and recovery of the system during unexpected spikes in traffic.
4. **Endurance Testing:** Running tests over an extended period to check for memory leaks or degradation over time.
5. **Volume Testing:** Assessing how the system performs with large amounts of data, such as bulk employee records.

## **Project Requirements:**

### **1. Setup and Access:**

- Access the OrangeHRM demo version at <https://opensource-demo.orangehrmlive.com/>.
- Focus on critical modules for testing:
  - **Login/Logout Workflow**
  - **Employee Management** (adding/updating employees)

- **Leave Management** (applying for leave)
- **Recruitment** (adding candidates, shortlisting, and interviews)
- **Payroll Processing**
- **Admin Module** (user management, role assignments)

## 2. Tools Required:

- **Apache JMeter** for creating test plans, simulating virtual users, and running performance tests.
- **GitHub** for storing performance test plans and scripts.
- **Jenkins** for continuous integration and automated testing.

## Detailed Benchmarks to Measure:

### A. Response Time Metrics:

#### 1. Average Response Time:

- **Description:** Measure the time it takes for OrangeHRM to load pages such as employee list, leave requests, and payroll.
- **Benchmark:** **Less than 2.5 seconds** for 90% of requests.

#### 2. 95th Percentile Response Time:

- **Description:** Response time for the slowest 5% of requests.
- **Benchmark:** Should be **under 4 seconds** for 95% of requests.

### B. Throughput and Load Handling:

#### 3. Requests per Second (Throughput):

- **Description:** Measure how many requests can be handled by the system per second.
- **Benchmark:** Application should handle **1,000-1,500 requests per second** under load.

#### 4. Concurrent Users:

- **Description:** Number of simultaneous users who can log in and perform actions without performance degradation.
- **Benchmark:** **500-1,000 concurrent users** for key actions like login, employee updates, and leave applications.

### C. Error Rate:

#### 5. Error Rate under Load:

- **Description:** Measure the percentage of failed requests as the user load increases.
- **Benchmark:** **Less than 2%** error rate at normal load, and **below 5%** during stress testing.

### D. Network Performance:

#### 6. Latency:

- **Description:** Measure latency from the user's browser to the OrangeHRM servers.
- **Benchmark:** **Less than 250ms** for most requests.

### **Test Scenarios to Execute:**

#### **1. Scenario 1: Load Testing (Normal Workflow Testing)**

- Simulate **500 concurrent users** logging in, viewing the dashboard, searching for employees, and applying for leave over a **30-minute period**.
- Measure response time, throughput, and error rates for regular load conditions.

#### **2. Scenario 2: Stress Testing (Pushing System to its Limits)**

- Gradually increase the load to **1,500 users** and beyond to identify at which point the system starts failing or showing performance degradation.
- Capture the point at which errors increase and response times exceed benchmarks.

#### **3. Scenario 3: Spike Testing (Sudden Traffic Bursts)**

- Simulate **1,000 users** logging in within **30 seconds**, performing employee searches, and applying for leave.
- Measure the system's ability to handle a sudden burst of users and its recovery time after the spike.

#### **4. Scenario 4: Endurance Testing (Long-term Stability)**

- Simulate **300-400 users** performing key operations (login, employee updates, leave management) over a **4-hour period**.
- Evaluate system stability, and performance degradation over time.

#### **5. Scenario 5: Volume Testing (Data Handling with Large Records)**

- Upload a large dataset of **5,000-10,000 employee records**, followed by **bulk updates** (e.g., changing departments).
- Measure how well OrangeHRM handles operations with a high volume of data and the time taken for bulk processing.

#### **6. Scenario 6: Recruitment Module Load Testing**

- Simulate **400 users** accessing the recruitment module, adding candidates, shortlisting, and processing applications.
- Test response times and errors during heavy use of the recruitment feature.

#### **7. Scenario 7: Payroll Processing Load Testing**

- Simulate **200 HR admins** simultaneously accessing the payroll module to process employee salaries.
- Measure the response time and system load during payroll processing and calculate the time for complete payroll generation.

#### **8. Scenario 8: Admin Module Stress Testing**

- Simulate **300 admin users** logging in to assign roles, manage user access, and update system configurations.

- Test performance degradation and response times under a high administrative load.

## **Deliverables:**

### **1. Performance Test Plan:**

- Detailed strategy, tools, and test scenarios for load, stress, spike, and endurance testing.

### **2. Performance Testing Report:**

- Summary of test results, including:
  - Average and 95th percentile response times
  - Throughput (requests per second)
  - Error rates (failed requests)
  - Latency and network performance
  - Performance bottlenecks
  - Recommendations for improvements

### **3. JMeter Test Scripts:**

- Submit all JMeter scripts used for the testing (.jmx files).

### **4. Graphical Analysis:**

- Include graphs and charts showing response time trends, throughput, error rate progression, and system performance under load.

## **Report Submission Template:**

### **1. Introduction:**

- Overview of OrangeHRM and the importance of performance testing.
- Objectives: To ensure OrangeHRM can handle large user traffic and provide a smooth user experience during critical operations.

### **2. Test Setup:**

- Tools: **JMeter, OctoPerf/BlazeMeter, Jenkins.**
- Environment: Server details, user simulation setup, number of virtual users, etc.
- Test Scenarios: Detailed workflows (login, employee management, recruitment, payroll) that were tested.

### **3. Test Results:**

- **Response Times:** Average, 95th percentile, and maximum response times for key scenarios.
- **Throughput:** Requests per second for each scenario.
- **Error Rates:** Percentage of failed requests.
- **System Limits:** Identify at what point the application failed or became unstable during stress testing.
- **Latency:** Network latency across user loads.

### **4. Conclusion:**

- Summary of performance across the test scenarios.
- Whether the system met the performance benchmarks.
- Recommendations for system optimization.

#### 5. **Appendix:**

- Attachments: Test scripts, graphs, detailed test logs, and any other relevant information.

#### **Additional Notes:**

- **Project Duration:** 7-10 days.
  - **Days 1-2:** Setup tools and environment.
  - **Days 3-4:** Create and validate JMeter scripts.
  - **Days 5-7:** Execute test scenarios, gather data.
  - **Day 8:** Analyze results and create performance report.