

OrangeHRM Selenium and Apache Jmeter Testing

Group Members

Asif Al Zawad	011201297
Md. Jahangir Alam	011201304
Abir Hossain	011191074
Md. Shamim Bin Nur	011191092



United International University

Dhaka, Bangladesh

Date: September 9, 2023

1. Abstract:

The objective of this testing project is to ensure the reliability, functionality, and performance of the OrangeHRM human resource management system. This project leverages the capabilities of Selenium for automated functional testing and Apache JMeter for performance testing. OrangeHRM is a critical application for managing HR operations, and it is crucial to guarantee its correctness, responsiveness, and scalability.

2. Automated Functional Testing with Selenium:

Selenium WebDriver will be used to automate the testing of OrangeHRM's web-based user interface.

Test cases will be designed to cover various HR functions, including employee management, leave management, recruitment, and user administration.

The goal is to validate the correctness of these functions, ensuring that they perform as expected under various scenarios.

2.1. Initial Feedback:

- **Test in localhost**
- **Show the test in comparative table**

2.2. Test Suit composition:

Group Test cases	Test Description	Comment	Expected Result
Add_vacancies	Create vacancy for the company	Admin needs to create Job Title from admin page	Vacancy added
Add_employee	Create an employee id with proper information	Cannot be added manually has to be added by admin	Employee added
Assign_leave	Give leaves to employee	Assigned by admin	Leave assigned
Apply_leave	Apply for leave	Employee's can apply for themselves	Leave applied
Admin_adimistration	Search for admins and delete or edit	Only can search for Admins	See admins delete & edit
Create_Admin	Promotes users to admin	Only can Admin can promote user to admin	Promote admin
Create_entities	Creates entities like (job titles , pay grade etc)	After creating job, pay grade, skills etc users can use them	Creates_entities
Search_employee	Searches for employee	Search for employees and edits employee	Employee found
Employee_attendance	Give attendance	Employees need to punch in and punch out	successful
Leave_entitlements	Gives entitlements to leave	Admin can give entitlements with leave types	Leave assigned
Leave_approve	After leave apply leave is approved	Leave approve is assigned by admin	Leave approved

2.3. Test case design and purpose

i) Test Apply_leave

(a) Design

- Login User
- Select Side panel Leave
- Select Apply
- Select Leave Type
- Select from date
- Select to date
- Input Comments if necessary;
- Click Apply

(b) Purpose

- All Employees might need leave and this will ensure that they can apply for leave.

(c) Outcome

Types	Expected Result	Result	Comment	Test Failure Reason
_valid	Apply leave (success)	✓	Leave Apply successful	—
_NoComment	Success	✓	Leave Apply successful	—
_Todate	Failed	✓	To date should be after from date	—
_NoLeaveType	Failed	✓	Leave Type Required	—

ii) Test Leave_approve

(a) Design

- Login Admin
- Select Side panel Leave
- Leave list
- Select from date
- Select to date
- Select Status
- Leave Type
- Employee name
- If necessary Sub Units
- Click Search
- Click Approve

(b) Purpose

- The leaves employees ask for leave, and these need to reviewed.

(c) Outcome

Types	Expected Result	Result	Comment	Test Failure Reason
_valid	Leave Approve (success)	✓	Leave Apply successful	—
_NoComment	Success	✓	Leave Apply successful	—
_ToDateLessFromdate	Failed	✗	To date should be after from date	Error but no new data Previously Existing data is still shown.
_NoLeaveStatus	Failed	✗	Leave Status Required	Error but no new data Previously Existing data is still shown.
_NonExistentLeaveStatus	Show no Record	✓	No Records Found	—
_NonExistingEmployeeName	Failed	✓	Invalid	—
_InvalidDate	Show no Record	✓	No Records Found	—

iii) Test Add_vacancy

(a) Design

- Login Admin
- Select Side panel Recruitment
- Select vacancy
- Add Vacancy
- Input vacancy name
- Select Job Title
- Input Description
- Add and select Hiring manager
- Add Number of positions

(b) Purpose

- This test will add vacancy for applying in the job

(c) Outcome

Types	Expected Result	Result	Comment	Test Failure Reason
_valid	Adds vacancies (success)	✓	Vacancy add successful	—
_NoDescription	Failed	✗	Without any job description it should the jobs should not be added	Job Description Should Be a Mandatory field
_ExistingVacancyName	Failed	✓	Already exists	—
_NoJobtitle	Failed	✓	Job Title Required	—
_NegativeNoOfPosition	Failed	✓	Should be a numeric value	—
_NonExistingHiringManger	Failed	✓	Hiring Manager Invalid	—

iv) Test Add_employee

(a) Design

- Login Admin
- Select Side PIM
- Nav Add Employee
- Enter Employee full name
- Employee Id will be given
- Select Create Login Details
- Username
- Select Status Enabled
- Enter Password
- Enter Confirm Password
- Click Save

(b) Purpose

- Employees need to be added in a management application. And add employee does that.

(c) Outcome

Types	Expected Result	Result	Comment	Test Failure Reason
_valid	Employee Created (success)	✓	Successfully saved	—
_EmployeeNameAsNumber	Failed	✗	Successfully saved	Name Should not be a number/
_WeekPassword	Failed	✓	Your password meets the minimum requirements, but it could be guessable	—
_WrongConfirmPassword	Failed	✓	Passwords do not match	—
_NegativeEmployeeId	Failed	✗	Successfully saved	Employee Id should not be negative integer
_ExistingEmployeeId	Failed	✓	IEmployee Id already exists	—

v) Test Employee_attendance

(a) Design

- Login User
- Select Side PIM
- Nav Attendance
- Select punch In/Out
- Time & Date auto load
- Click In/Out

(b) Purpose

- Employees attendance is a mandatory part for a management site.

(c) Outcome

Types	Expected Result	Result	Comment	Test Failure Reason
_valid	Attendance (success)	✓	Successfully saved	—
_NotinWorkingHours	Failed	✗	Successfully saved	Employee should not be able to punch in at a non working hour then working hour will still be counted
_PunchOutAnytime	Successful	✓	Successfully saved	—

2.4. Outcome Of the selenium test

- We have tested some features of OrangeHRM Unit and Functions.
- The outcome of this entire project is that we have found some bug in the system that may be improved.
- But the result may vary from more testing or from client, stakeholders and from testers perspective.

3. Performance Testing with JMeter:

Apache JMeter will be employed to conduct performance testing on the OrangeHRM application.

Load testing, stress testing, and scalability testing will be performed to assess the system's response under different levels of user activity.

Key performance metrics such as response times, throughput, and resource utilization will be monitored and analyzed.

3.1. Initial Feedback:

- **Test in localhost**
- **Analysis the load on the device/server**
- **Describe few test case in details**
- **Show the test in comparative table**

3.2. Testing Device Configuration

- **Model: Dell Inspiron 15 3580**
- **Processor: Intel Core i5-8265U**
- **Base Frequency: 1.6GHz**
- **Max Turbo Frequency: 3.9GHz**
- **Cache: 6MB SmartCache**
- **RAM: (Varies by configuration, typically 4GB or 8GB)**
- **Storage: (Varies by configuration, typically HDD or SSD options)**
- **Display: 15.6" (1366 x 768) or (1920 x 1080) resolution**

3.3. Test Suit composition:

Group Test cases	Test Description	Comment	Expected Result
Record login test	Record the scenario - Stop Recording - Export .jmx.	Needs import imx file in JMeter & set variable	Capture user interactions with application without having to manually script each request.
Concurrency Testing	Test the application's behavior with multiple users simultaneously.	Examines how the application handles concurrent user interactions.	Should handle concurrent user interactions with minimal performance degradation.
Performance Testing	Verify the application's performance under normal operating conditions.	Evaluates how the application performs under regular user load	Should handle expected load with acceptable response times and minimal errors.
Load Testing	Validate the application's behavior under specific load scenarios.	Assesses the application's response when subjected to higher-than-normal loads.	Should sustain specified load levels while maintaining performance metrics.
Stress Testing	Test the application's stability under extreme load.	This test pushes the application to its limits to observe its failure points.	Application should demonstrate stability and graceful degradation under extreme load.
API Testing	Verify the functionality and performance of APIs using JMeter.	Validates the APIs' responsiveness, data accuracy, and performance.	APIs should return expected responses and handle load without errors.
Server Performance Testing	Evaluate the server's performance and responsiveness.	Assesses the server's ability to handle requests promptly.	The server should handle requests promptly and provide timely responses.
Endurance Testing	Test the application's stability over an extended duration.	Checks for performance degradation over time.	The application should sustain normal load without significant performance degradation.

Group Test cases	Test Description	Comment	Expected Result
Distributed Testing	Validate the application's performance in distributed environments.	This test assesses how the application performs in distributed setups.	Distributed load should be managed effectively, maintaining performance.
Regression Testing	Validate the application's performance after updates.	Ensures performance remains consistent or improved after updates.	Performance should remain consistent or improved after updates.
Benchmarking	Compare performance metrics against industry standards.	This test benchmarks the application's performance against industry standards.	The application's performance should meet or exceed industry benchmarks.

3.4. Test case design and purpose

i) Test Record login test Load

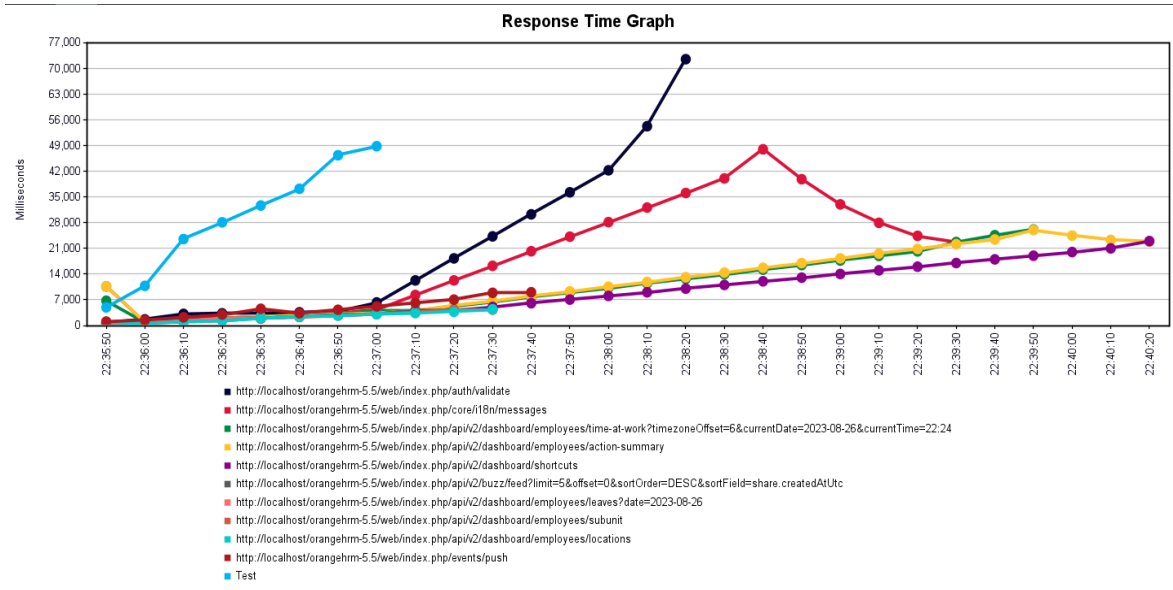
(a) Design

- Add blazemeter plugin to chrome browser
- Start blazemeter plugin and login to blazemeter
- Record the scenario - Stop Recording - Export .jmx. If find jmx option disabled export as json and use this link to convert to jmx file <http://converter.blazemeter.com/>
- Import imx file in JMeter & set variable
- Add listeners
- Run and validate

(b) Purpose

- This test will capture user interactions with application simulating user behavior and creating test scenarios without having to manually script each request.

(c) Outcome

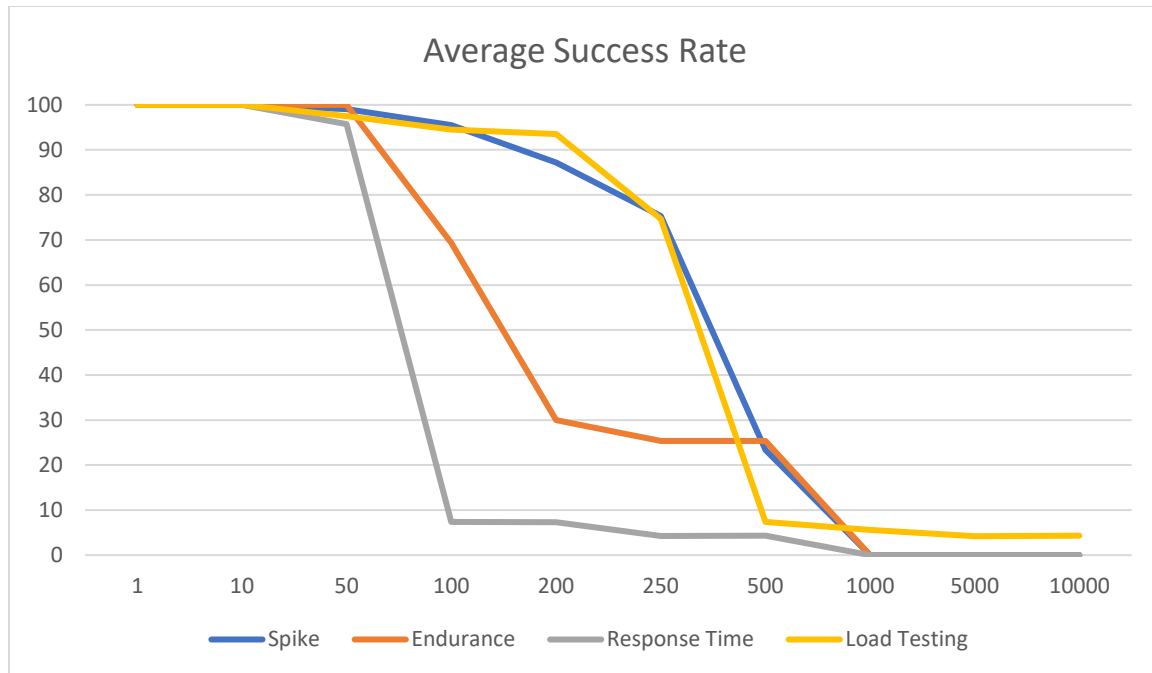


Graph 3.4.1: shows the response time graph.

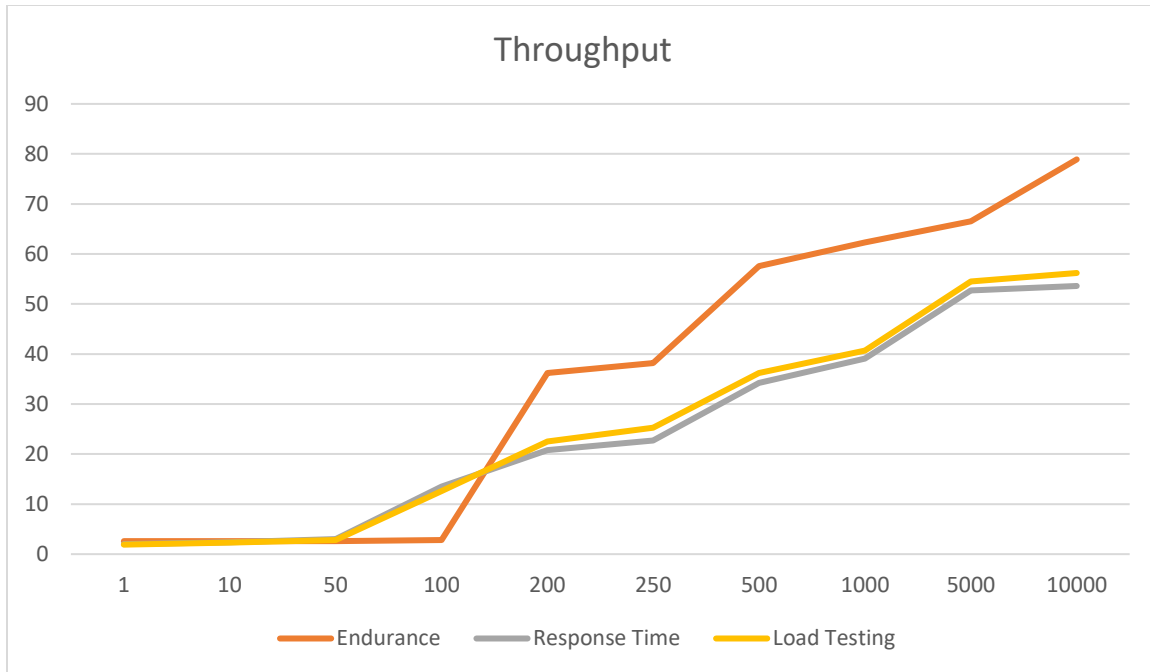
- 5000 users
- 81.0% CPU, 607.0MB memory, 2.5 MB/s Disk speed.

3.5. Test case design and purpose

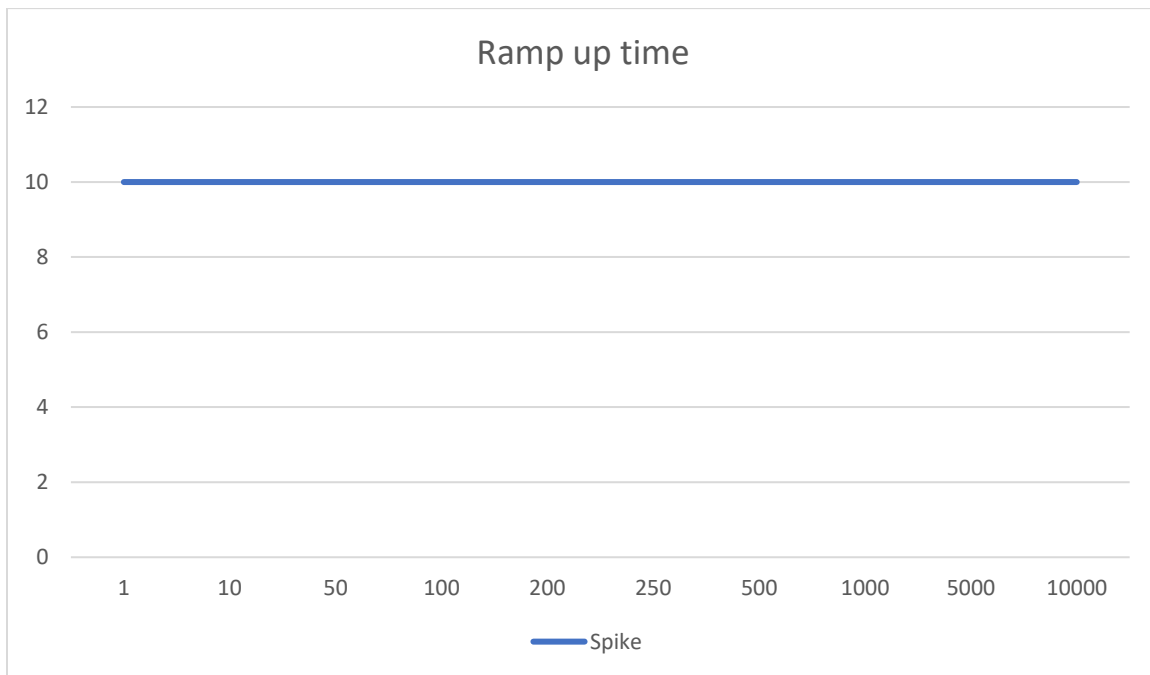
- I. Test Concurrent Users Load
 - II. Load testing on Search Functionality
 - III. Home page Response Time
 - IV. Home page Response Time
 - V. Admin Dashboard Endurance Testing
 - VI. Admin Dashboard Spike Test
- (d) Outcome



Graph 3.5.1: shows the average success rate.



Graph 3.5.2: shows the throughput.



Graph 3.5.3: shows the ramp up time.

4. Conclusion:

The successful completion of this testing project will help ensure the robustness and scalability of the OrangeHRM system. It will enable stakeholders to make informed decisions regarding the application's release and maintenance, ultimately enhancing the user experience and organizational HR operations.