

OrangeHRM Automation Test Plan

1. Introduction

This document provides a detailed test plan for automating the OrangeHRM portal using Selenium with Java. The objective is to automate core functionalities and ensure consistent behavior across different user roles.

2. Project Overview

- **Application:** [OrangeHRM](#)
 - **Automation Tool:** Selenium
 - **Programming Language:** Java
 - **Test Framework:** TestNG
 - **Execution:** Parallel execution using TestNG
-

3. Scope

- **Browsers:** Chrome, Firefox, Edge
 - **Platforms:** Windows, macOS
 - **User Roles:** Admin, Employee
-

4. Objectives

- Automate core functionalities to reduce manual effort.
 - Ensure consistent performance across different environments.
 - Capture and report defects automatically.
-

5. Test Scenarios

5.1 Authentication

- ✓ Verify login with valid Admin and Employee credentials.
- ✓ Test login with invalid credentials and validate error messages.

- ✓ Verify password reset functionality.

5.2 Navigation

- ✓ Verify that all navigation links work correctly.
- ✓ Ensure the user is redirected to the correct pages.
- ✓ Test browser back/forward navigation behavior.

5.3 User Roles and Permissions

- ✓ Confirm that an Admin user can access admin-only features.
- ✓ Verify that an Employee user has limited access.
- ✓ Test role-based access to specific modules.

5.4 Employee Management

- ✓ Automate adding a new employee.
- ✓ Test updating employee information.
- ✓ Confirm deletion of employee records.

5.5 Leave Management

- ✓ Test applying for leave.
- ✓ Verify leave approval/rejection by Admin.
- ✓ Ensure leave balances update correctly.

5.6 Performance Management

- ✓ Test adding performance reviews.
- ✓ Verify review submission and feedback visibility.

5.7 File Upload/Download

- ✓ Verify supported file types for employee documents.
- ✓ Confirm successful upload and storage.
- ✓ Test download functionality for different user roles.

5.8 Notifications

- ✓ Test email notifications for user actions (e.g., leave approval).
- ✓ Confirm real-time notifications appear correctly.

5.9 Post Management

- ✓ Test CRUD operation on post.
 - ✓ Verify the share posts.
-

6. Test Data

- **Admin Credentials** – Username: Admin | Password: admin123
 - **Employee Credentials** – Test account credentials
 - **Sample Files** – PDF, DOCX, PNG for upload testing
 - **Test Inputs** – Employee details, leave requests
-

7. Automation Guidelines

- 👉 Use **Page Object Model (POM)** for test structure.
 - 👉 Implement **data-driven testing** for form submissions and employee data.
 - 👉 Add **logging** and **screenshot capture** for debugging.
 - 👉 Use **TestNG annotations** (`@Test`, `@BeforeMethod`, `@AfterMethod`) for test organization.
 - 👉 Handle dynamic elements using **explicit waits**.
 - 👉 Implement retry logic for flaky tests.
-

8. Pass/Fail Criteria

- ✓ Test passes if the functionality meets expected behavior.
 - ✗ Test fails if the functionality produces incorrect or unexpected behavior.
-

9. Risks and Assumptions

- Internet connectivity should be stable during testing.
- No major structural changes to the portal during testing.

10. Reporting



Generate detailed test reports after each execution using **TestNG Reports**.



Log defects in **JIRA**.



Provide weekly progress updates.

11. Sign-Off

Approved by: _____

Date: _____