Test Plan- OrangeHRM (Open-Source HR Management)

1. Introduction

The purpose of this test plan is to outline the testing strategy and approach for the OrangeHRM demo web application. OrangeHRM is an open-source Human Resource Management software that provides modules like Admin, PIM, Leave, Time, Recruitment, etc. The test plan will help ensure that all functionalities are verified and the application performs as expected. The primary objective of this testing initiative is to validate that the demo application functions as expected from a user perspective. This includes ensuring that all accessible modules work correctly, user interface elements are displayed and behave properly, and users can navigate through the system without errors. Both functional and UI aspects will be evaluated. Testing will be conducted using a combination of manual and automated approaches, with a focus on critical workflows such as login authentication, navigation across modules, data input in forms, and record management. The results of the tests will provide insight into the reliability, usability, and robustness of the OrangeHRM demo system.

2. Scope of Testing

In Scope:

- Authentication (Login/Logout)
- Navigation to Dashboard and Modules (Admin, PIM, Leave, etc.)
- Form submissions (e.g., adding users or employees)
- UI validation (elements display correctly)
- Functional workflows (e.g., searching employee records)

Out of Scope:

- Backend database validation
- Performance/load testing
- Email/SMS notifications

3. Testing Objectives

- To validate user login and logout functionality.
- To ensure proper navigation across different modules.
- To verify that key forms accept input, validate it, and submit data correctly.
- To identify and log any UI or functional bugs.
- To ensure that the site meets usability standards.

4. Test Environment Setup

- **Browser**: Chrome, Firefox, Edge
- **OS**: Windows 10/11 or macOS
- **Network**: Stable internet connection
- **Test Data**: Dummy users and employees will be created for testing.

5. Tools & Technologies Used

- **Testing Tools**:
  - o Manual Testing (via browser)
  - o Selenium WebDriver (for automation)
- **Bug Tracking**: GitHub Issues or JIRA
- **Documentation**: MS Word or Google Docs for Test Cases and Reports
- **Technologies**: HTML, CSS, JavaScript (frontend); server-side details are abstracted

6. Test Types

- **Functional Testing**: Login, navigation, form submission, workflows
- **UI Testing**: Element placement, colours, icons, responsiveness
- **Regression Testing**: If any updates are deployed to the demo
- **Compatibility Testing**: Across different browsers
- **Negative Testing**: Invalid logins, empty form submissions

7. Entry & Exit Criteria

Entry Criteria:

- Test environment is set up
- URLs are accessible
- Functional requirements are understood
- Test cases are reviewed and approved

Exit Criteria:

- All planned test cases executed
- All critical and major bugs are resolved or documented
- Test summary report is shared

8. Assumptions and Risks

Assumptions:

- The demo site is stable during testing
- No login restrictions or IP blocks
- Functionalities mimic the production version of OrangeHRM

Risks:

- The demo site might reset or go offline unexpectedly
- Limited access to backend may restrict deeper testing
- Incomplete feature coverage due to demo limitations

9. Team Roles and Responsibilities

Role	Responsibility
Test Plan	Mirza Uzair Baig and Xabad Nadeem collaboratively created test plan with all the resources and risks associated.
Test Engineer	Mirza Uzair Baig (1-15) test cases, Xabad Nadeem (15-30) test cases, Saim Javed (30-45) test cases and Zain Ali (45-50) test cases. All these persons involved are responsible for creating test cases and manually testing them.
Automation Tester & Developer	Zain Ali will be doing automation testing by preparing the automation script using Selenium. Also, he will be pushing the code to GitHub
Execution Report	Zain Ali and Saim Javed will be creating test execution report from the results

10. Test Schedule

Phase	Timeline
Test Planning	3 Days
Test Case Design	4 Days
Environment Setup	2 Days
Test Execution	5 Days
Bug Reporting & Re-testing	4 Days
Test Summary Report	3 Days
Total	**21 Days (3 Weeks)**