### **Test Plan for OrangeHRM**



#### **1. Introduction**

* **Objective**: The objective of this project is to automate and manually test the **OrangeHRM** web application, ensuring the website works across different browsers, platforms, and devices. Both functional and non-functional requirements will be validated.
* **Scope**: The testing will cover key functional areas, including UI component validation, dynamic elements, navigation, cross-browser compatibility, and core functionalities. Manual testing will be performed for exploratory testing, usability testing, and validation of complex user interactions.
* **Tools and Frameworks**:
  + **Selenium WebDriver** for browser automation.
  + **Cucumber** for Behavior-Driven Development (BDD).
  + **Page Object Model (POM)** for maintainability and readability.
  + **TestNG** or **JUnit** for test execution and reporting.
  + **Extent Reports** for reporting.
  + **Manual Testing** for exploratory, usability, and complex user interactions.

#### **2. Test Environment Setup**

* **System Requirements**:
  + **Operating System**: Windows / Linux / MacOS (cross-platform testing)
  + **IDE**: IntelliJ IDEA
  + **JDK**: Java 8 or higher
  + **Browser**: Google Chrome, Mozilla Firefox, Microsoft Edge
  + **Dependencies**:
    - **Selenium WebDriver**
    - **Maven/Gradle**
    - **Cucumber**
    - **JUnit/TestNG**
    - **Extent Reports**

#### **3. Test Approach**

* **Test Strategy**:
  + Implement **Cucumber** feature files for defining test scenarios.
  + Use **Page Object Model (POM)** to organize web elements and interactions.
  + Run **Selenium** scripts for automated functional testing.
  + Execute **TestNG** or **JUnit** tests for management and reporting.
  + Perform **Manual Testing** for exploratory and usability checks, particularly for complex user journeys.

#### **4. Key Functional Areas to Test**

1. **Navigation and Page Consistency**:  
   * **Automated**: Verify that all navigational elements are functional using Selenium scripts.
   * **Manual**: Perform exploratory testing to ensure smooth navigation and check for broken links or unexpected behavior.
2. **UI Components Validation**:  
   * **Automated**: Validate buttons, forms, images, and links using Selenium scripts.
   * **Manual**: Check UI components' visual alignment, usability, and functionality, such as checking the consistency of colors, fonts, and responsiveness.
3. **Dynamic Elements Testing**:  
   * **Automated**: Test dropdowns, pop ups, and modals with Selenium WebDriver.
   * **Manual**: Test unexpected behavior of dynamic elements (e.g., pop ups, error messages) under different conditions.
4. **Cross-browser and Responsive Testing**:  
   * **Automated**: Run automated tests on multiple browsers (Chrome, Firefox, Edge) using Selenium for cross-browser testing.
   * **Manual**: Manually verify the application on different devices (tablet, mobile) for responsiveness.
5. **Core Functionalities**:  
   * **Automated**: Test the login functionality, form submissions, employee management, and reporting via Selenium scripts.
   * **Manual**: Perform exploratory testing on user roles, permissions, and edge cases like invalid data.

#### **5. Testing Goals**

* **Functional Testing**: Automated and manual testing will be used to verify that the application works as per the functional specifications.
* **Non-functional Testing**: Validate performance, security, and usability of the system.
* **Bug Reporting**: Report and track bugs found during both automated and manual testing.
* **Reporting**: Provide detailed test reports (manual and automated) for stakeholders.

#### **6. Test Case Development**

* **Automated Test Cases**:
  + **Login**: Verify valid login, invalid login, and forgotten password functionality.
  + **Navigation**: Ensure that all navigation links (Dashboard, Admin, Leave) lead to the correct pages.
  + **Add Employee**: Validate the process of adding a new employee using a form.
  + **Dynamic Element Handling**: Validate the behavior of dropdowns, modals, and error messages.
* **Manual Test Cases**:
  + **Exploratory Testing**: Test the overall usability and check for inconsistencies or unexpected behavior.
  + **Usability Testing**: Ensure the website is user-friendly and easy to navigate.
  + **Edge Case Testing**: Manually test for edge cases like invalid input, long data entries, or interaction with multiple form elements.

#### **7. Test Execution**

* **Execution Workflow**:
  + **Automated Testing**: Run Selenium tests on multiple browsers (Chrome, Firefox, Edge) for cross-browser validation.
  + **Manual Testing**: Perform exploratory and usability tests on different browsers and devices.
  + **TestNG/JUnit Execution**: Use TestNG for managing parallel test executions and JUnit for validation.

#### **8. Documentation Deliverables**

1. **Test Plan**: Detailed plan for both automated and manual testing activities.
2. **Test Cases and Scenarios**: Includes automated and manual test cases.
3. **Mind Map**: Visual representation of testing areas (for both manual and automated testing).
4. **Bug Reports**: Log issues found during both manual and automated tests using a standard bug reporting template.
5. **Summary Report**: A summary of the test execution results, highlighting passed and failed tests, issues found, and resolutions.

#### **9. Bug Reporting Template**

* **Bug ID**: Unique identifier for each issue.
* **Title**: Brief description of the bug.
* **Description**: Steps to reproduce.
* **Severity**: Low, Medium, High, Critical.
* **Status**: Open, In Progress, Closed.
* **Priority**: Low, Medium, High.
* **Assigned To**: Developer responsible for the fix.
* **Screenshots/Logs**: Attach relevant information.

#### **10. Execution and Reporting with TestNG/JUnit**

* **TestNG**:
  + Run tests in parallel for faster feedback.
  + Use **TestNG** annotations (@BeforeTest, @Test, @AfterTest) to structure test executions.
* **JUnit**:
  + Use **JUnit** assertions to validate automated tests.
* **Extent Reports**:
  + Capture logs, screenshots, and execution details for both manual and automated tests.
  + Generate detailed reports with descriptive summaries for both automated and manual testing.