Jest Framework Proof of Concept (POC)

1. Technical Requirements Document

1.1 Project Overview

This POC addresses the critical need for establishing a robust, scalable testing strategy that ensures software quality, reduces production defects, and accelerates development cycles. The initiative focuses on implementing comprehensive test automation practices that cover unit testing, integration testing, and continuous quality assurance across the entire software development lifecycle

1.2 Technical Stack Requirements

- Node.js
- Jest
- JavaScript
- Playwright

1.3 System Requirements

- Operating System: Windows, macOS, or Linux
- RAM: Minimum 4GB (8GB recommended)
- Node.js Package Manager: npm or yarn
- IDE: VS Code, WebStorm, or any text editor with JavaScript support

2. Test Scenarios

ID	Module	Scenario Description	Purpose
TS01	Authentication	Login with valid credentials	Confirm that user can log in successfully
TS02	Authentication	Login with incorrect credentials	Verify the error message is displayed
TS03	Personal Details	Check First, Middle and Last Name	Ensure user details are displayed accurately and saved
TS04	Contact Details	Update Address	Verify that the (Street1, Street2, City) can be updated.
TS05	Contact Details	Update State, Postal Code, and Country	Verify that State/Province and Zip/Postal Code can be updated and saved, and that Country is correctly selected from the dropdown

TS06	Contact Information	Update Phone Number	Confirm that phone number is updated correctly
TS07	Contact Details	Update Email	Confirm that work email and other email are updated correctly and saved
TS08	Emergency Contacts	Add a new Emergency Contact by clicking the Add button	Ensure new emergency contact (Name, Relationship, Mobile) is successfully added and saved
TS09	Emergency Contacts	Required fields criteria	Required fields are not null values and saved successfully.
TS10	Emergency Contacts	Delete Emergency Contact (Optional)	Verify removal of an emergency contact works

3. Planned Approach

1. **Browser Setup**
   - Launch the browser using Playwright.
   - Navigate to the OrangeHRM demo site login page.
2. **Test Execution**
   - Execute each test case sequentially in the same browser session where possible.
   - Apply the Page Object Model (POM) design pattern for modularity and reusability.
3. **Validation**
   - Perform functional checks using Jest expect assertions to validate outcomes.
4. **Failure Handling**
   - Capture screenshots automatically for failed test steps.
   - Attach failure evidence in the execution report for Invalid inputs, network failures, boundary conditions.
5. **Reporting**
   - Generate an HTML-based test execution report using jest-html-reporters.
   - Report contains executed tests, passed/failed/skipped count, execution time, and failure screenshots.

6. **Execution Environment**
   - Tests will run with Jest as the test runner and Playwright as the browser automation library.
   - Execution will be done in a browser context managed by Playwright.
7. **Deliverables**
   - Test Scenarios & Test Cases
   - Automated test scripts.

- HTML execution report.
- Screenshots for any failed test cases.