

# OrangeHRM Testing Report

## Overview

This report explains the automation testing scripts created for the OrangeHRM system. The tests are grouped into four main areas: logging in, managing employees (adding and editing), and handling data. The tests use a method called the Page Object Model (POM) to make the code easier to manage and reuse.

### ### 1. Test Data Package

**Purpose:** This package includes the test data needed for the scripts.

#### **File:**

- TestData.py:
  - Contains constants like `BASE\_URL`, `USERNAME`, `PASSWORD`, and details for employees.
  - Acts as a single source for test data, which helps avoid using hard-coded values.

### ### 2. Locator Package

**\*\*Purpose\*\*:** This package holds the locators (ways to identify elements) used to interact with the web application.

#### **File:**

- WebLocators.py:
  - Defines locators for elements like username and password fields, buttons, and input areas.
  - Uses methods like XPath for reliable element identification.

### ### 3. Pages Package

**Purpose:** This package follows the POM design pattern, organizing actions related to web pages into specific classes.

#### **Files:**

- LoginPage.py:
  - Contains methods for logging in, including entering credentials and clicking the login button.
- PIMPage.py:
  - Has methods for going to the PIM module, adding and searching for employees, clicking the edit button, and updating employee details.
  - Includes methods to verify employee information after actions.

### ### 4. Test Package

**Purpose:** This package includes the actual test scripts that run the test cases.

#### **Files:**

- test\_login.py:
  - test\_login\_success(): Tests successful login with valid credentials and checks for the dashboard.
  - test\_login\_invalid(): Tests login with invalid credentials and checks for the correct error message.
- test\_pim.py:
  - test\_add\_employee(): Tests adding a new employee and checks that the "Personal Details" page appears after submission.
  - test\_edit\_employee(): Tests editing an employee's details and ensures the employee ID updates correctly.

#### ### Execution Environment

- **Python Version:** Works with Python 3.12.
- **Selenium Version:** Uses the latest version of Selenium WebDriver for browser automation.
- **Browser Drivers:** ChromeDriver is managed automatically using the `webdriver\_manager` package.

#### ### Conclusion

This automation testing framework is designed to reliably test the OrangeHRM application. By using POM and structured test data management, the framework aims to make tests easier to maintain and expand. More test cases can be added to cover additional scenarios as needed.