OrangeHRM Testing

Team 1 Presentation

Project Overview

Project Name
OrangeHRM-Project

Report Date
November 15, 2025

Prepared By
Team 1: Kerolos Samuel Awad, Mohamed Radi Goda, Zeyad Osama, Ahmed Ali, Hazem ghobashy

Tested Feature: Save System User

The Save System User page within the OrangeHRM Admin Module is a critical tool for HR administrators. It facilitates the creation, editing, and management of system-level user accounts, ensuring secure and controlled access to the HR system. Administrators can efficiently assign login credentials, define specific user roles, link accounts to existing employees, and meticulously control access permissions.

Key Functionalities Tested

Login Function

Ensuring secure and reliable user authentication.

Add Employee

Verifying the creation of new employee records.

Edit Employee

Confirming accurate modification of employee details.

Delete Employee

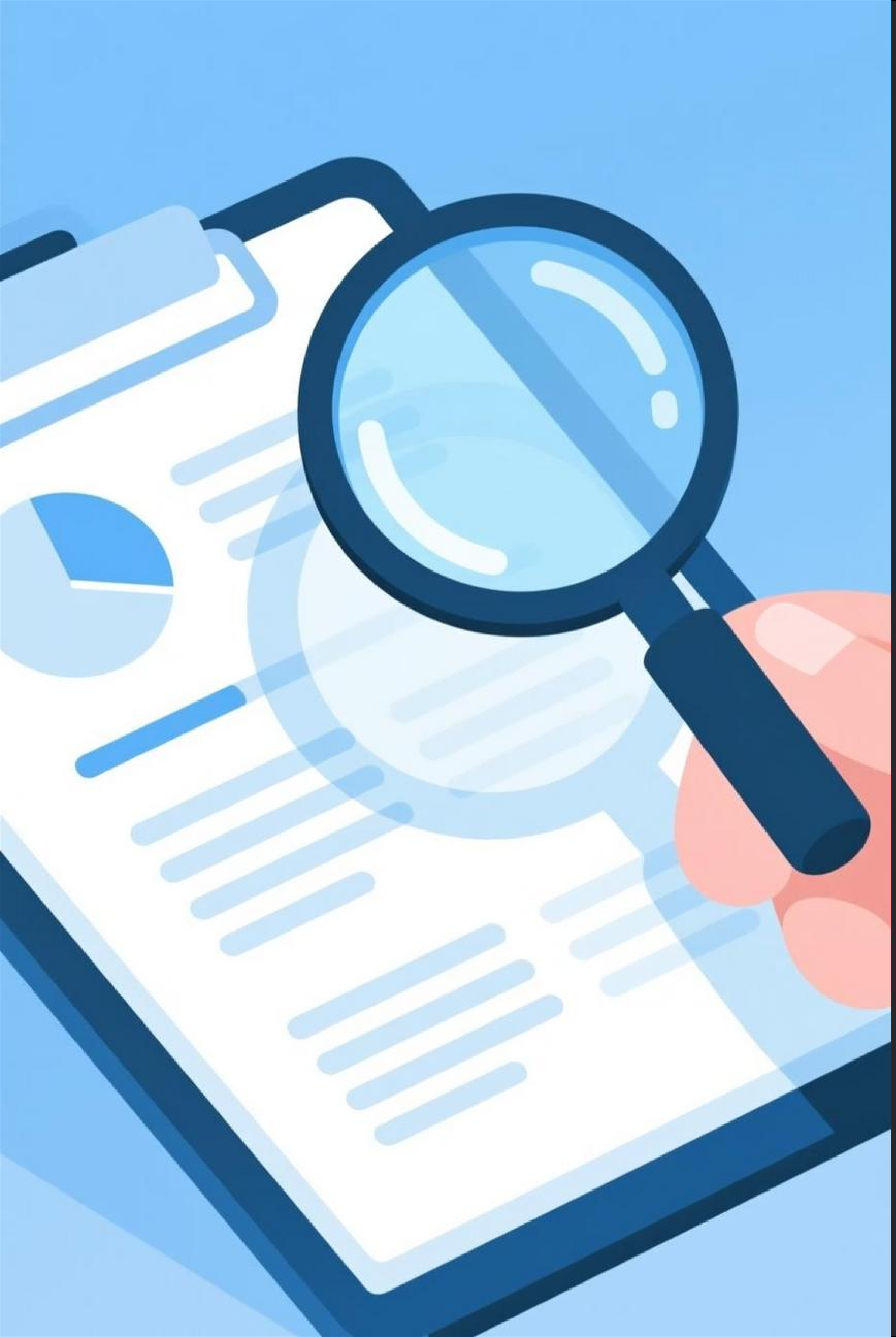
Validating the removal of employee profiles from the system.

Add Personal Details

Testing validation of adding personal deatils successfully

Leave Function

Testing the request and management of employee leave.



Objective of This Report

The primary objective of this Test Summary Report is to provide a comprehensive overview of the testing activities conducted for OrangeHRM. This includes presenting:

- Overall testing activities and their scope.
- Detailed results and coverage achieved during testing.
- Summary of identified defects and their impact.
- Assessment of the quality status of the application under test.
- A clear indication of the product's readiness for release.

Scope of Testing

In Scope

- Functional Testing: Verification of all specified features and functionalities.
- UI/UX Testing: Evaluation of user interface and user experience design.
- Compatibility Testing: Ensuring functionality across different browsers and OS.
- Integration Testing: Validating interactions between different modules.
- Smoke Testing: Quick checks to ensure basic functionalities are working.
- Security Testing: Identifying vulnerabilities and protecting data integrity.

Out of Scope

- Performance Testing: Assessment of system responsiveness and stability under load
- Non-functional Testing: Any non-functional aspects

Test Approach

Our testing process adhered to a structured and standard Quality Assurance (QA) cycle to ensure thorough coverage and defect identification.

01

Exploratory Testing

Initial flexible testing to discover application behavior.

02

Test Scenario & Case Creation

Development of detailed test scenarios and test cases.

03

Test Data Preparation

Generation and setup of necessary data for test execution.

04

Manual Test Case Execution

Systematic execution of all defined test cases.

05

Defect Logging

Documentation of defects with steps, severity, and screenshots.

06

Re-testing & Regression

Verification of fixes and ensuring no new issues are introduced.

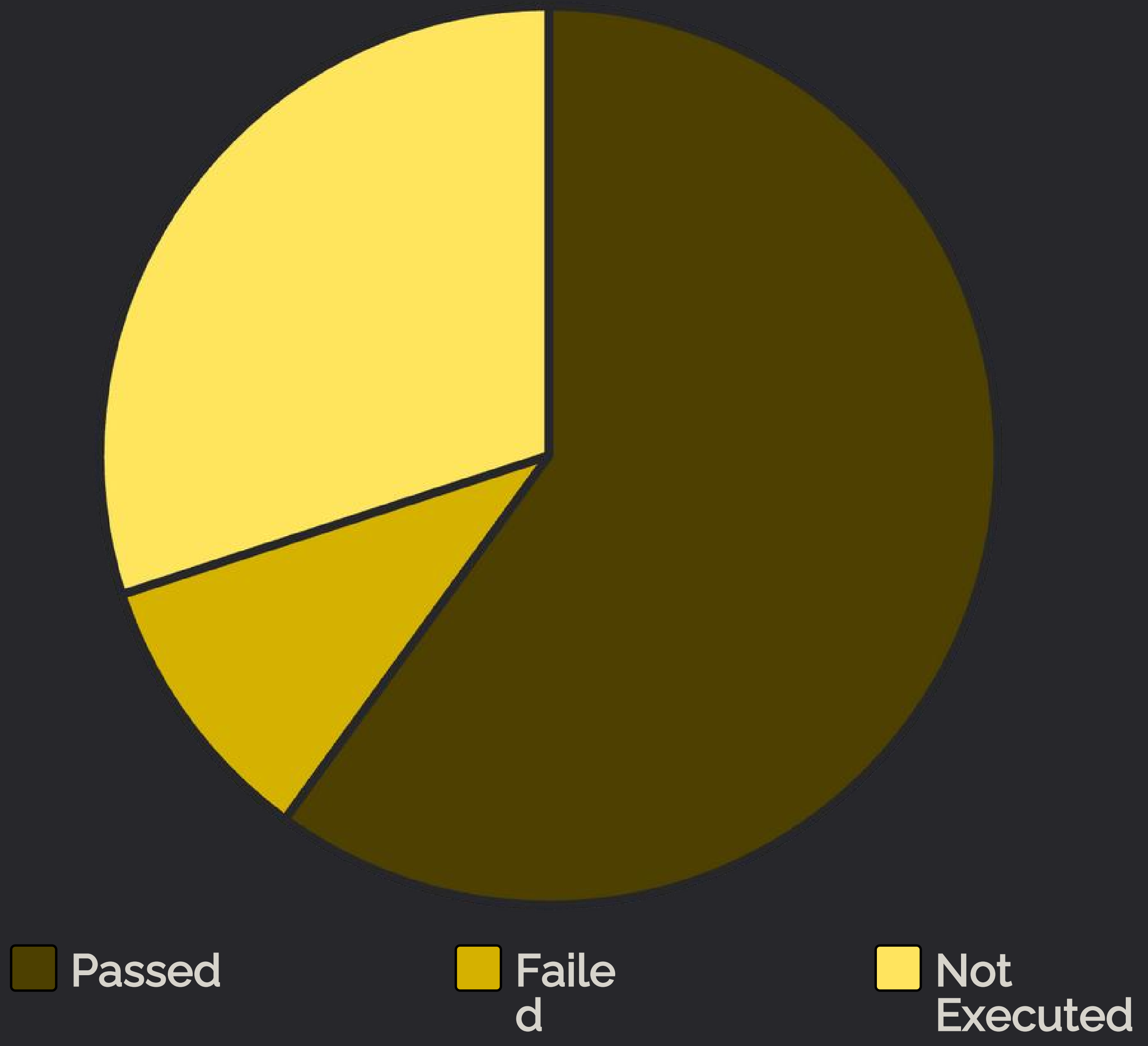
Test Environment

Key Environment Details

- Environment: QA Platform
- Browser(s): Chrome, Firefox, Edge
- Operating System: Windows 10
- Build Version: 64-bit

The testing was conducted on a dedicated QA platform, ensuring a controlled and consistent environment for accurate results. We tested across multiple leading web browsers and a widely used operating system to guarantee broad compatibility and optimal user experience.

Test Execution Summary



150

Test Cases Prepared

105

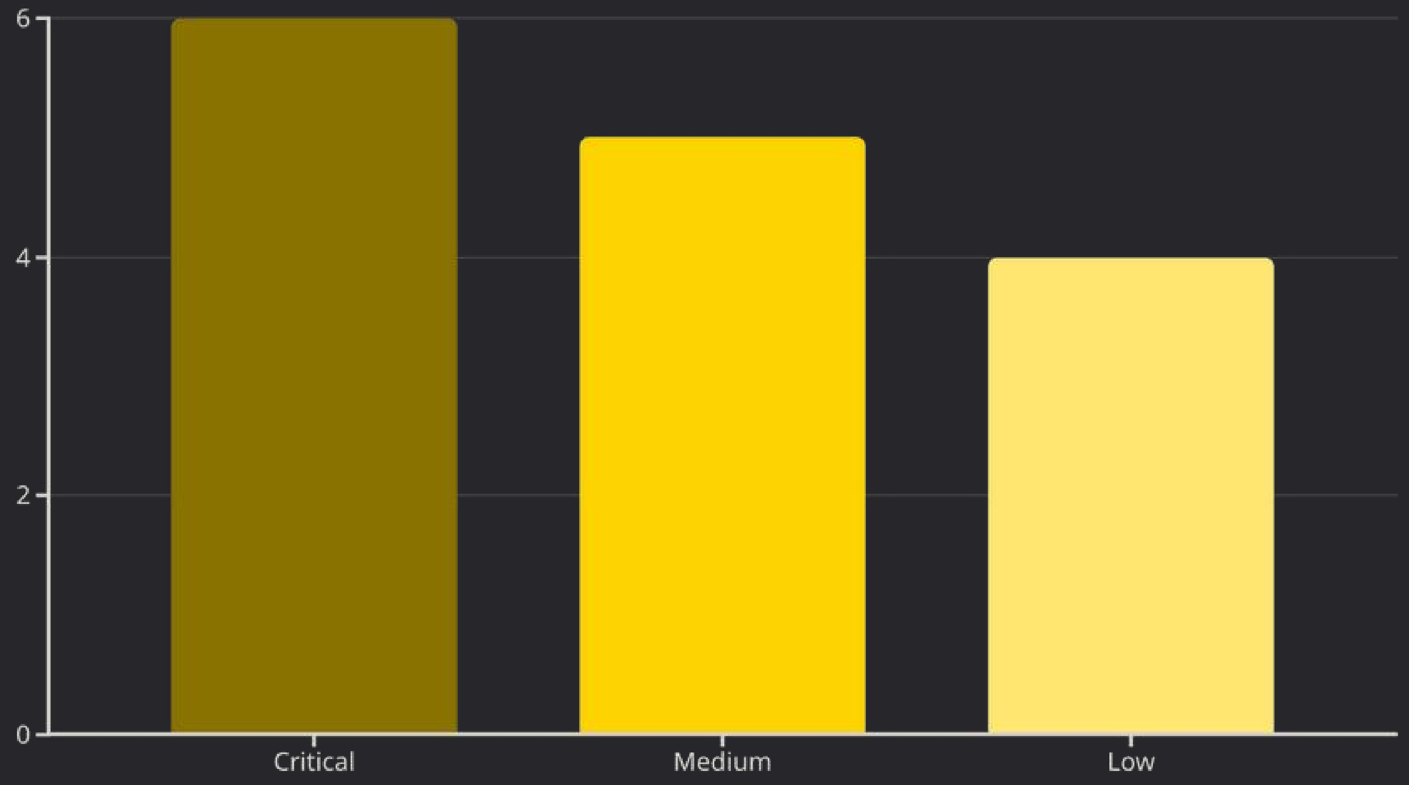
Test Cases Executed

85%

Pass Rate

Out of 150 prepared test cases, 105 were executed, achieving an 85% pass rate. This indicates a high level of stability for the tested features.

Defect Summary & Test Cases



Total Defects Logged
15 Bugs

Critical Defects

6

Medium Defects

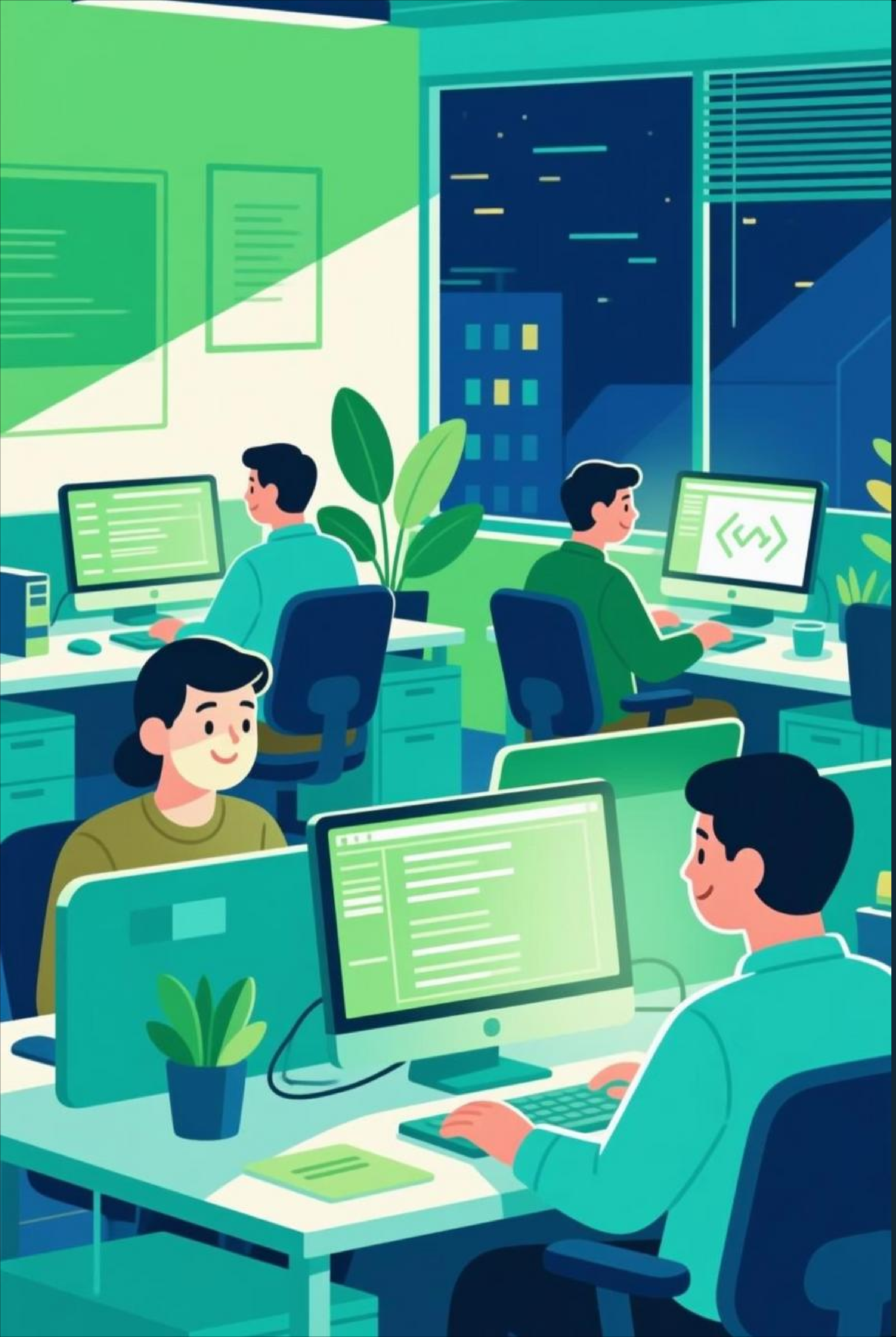
5

Low Defects

4

A total of 15 defects were logged, with a distribution across critical, medium, and low severities. The detailed test cases and their results can be reviewed via the provided link:

Test Cases Link



OrangeHRM Test Automation Framework

QA Team Automation Excellence

A comprehensive automation solution for core OrangeHRM modules built by our dedicated QA team

Team Introduction

Ahmed Ali

Automation Tester

Login Module

Kerolos Samuel

Automation Tester

PIM Module

Mohamed Goda

Automation Tester

Admin Module

Zeyad Osama

Automation Tester

My Info Module

Tools & Technologies

Java, Selenium WebDriver, TestNG, Maven, Page Object Model (POM), IntelliJ IDEA, GitHub, Apache POI

Project Overview

Purpose

Automate core OrangeHRM modules
(Login, PIM, Admin, MyInfo)

Goals

- Improve reliability
- Reduce manual testing
- Build scalable automation
- Follow clean POM architecture

Framework Stack

Framework built using **Java + Selenium + TestNG** using Page Object Model

Automation Framework Architecture

1. Page Objects

   src/test/java/pages → Login, PIM, Admin,MyInfo page classes

2. Test Classes

   src/test/java/tests → Test scenarios for each module

3. Utilities

   src/main/java/utils → Helpers, waits, actions

4. Test Data

   src/test/resources/testdata → Excel or JSON data

5. Base Configuration

   BaseTest → WebDriver setup, configuration

6. Execution

   TestNG XML → Test execution flow

Module 1: Login Module

Ahmed Ali – Automation Tester

1 LoginPage Structure

Organized page object with locators and methods

2 Test Scenarios

Valid login, Invalid login, Error message validation

3 Locator Strategy

ID, CSS, XPath for element identification

4 Assertions & Verification

Comprehensive validation steps for each scenario

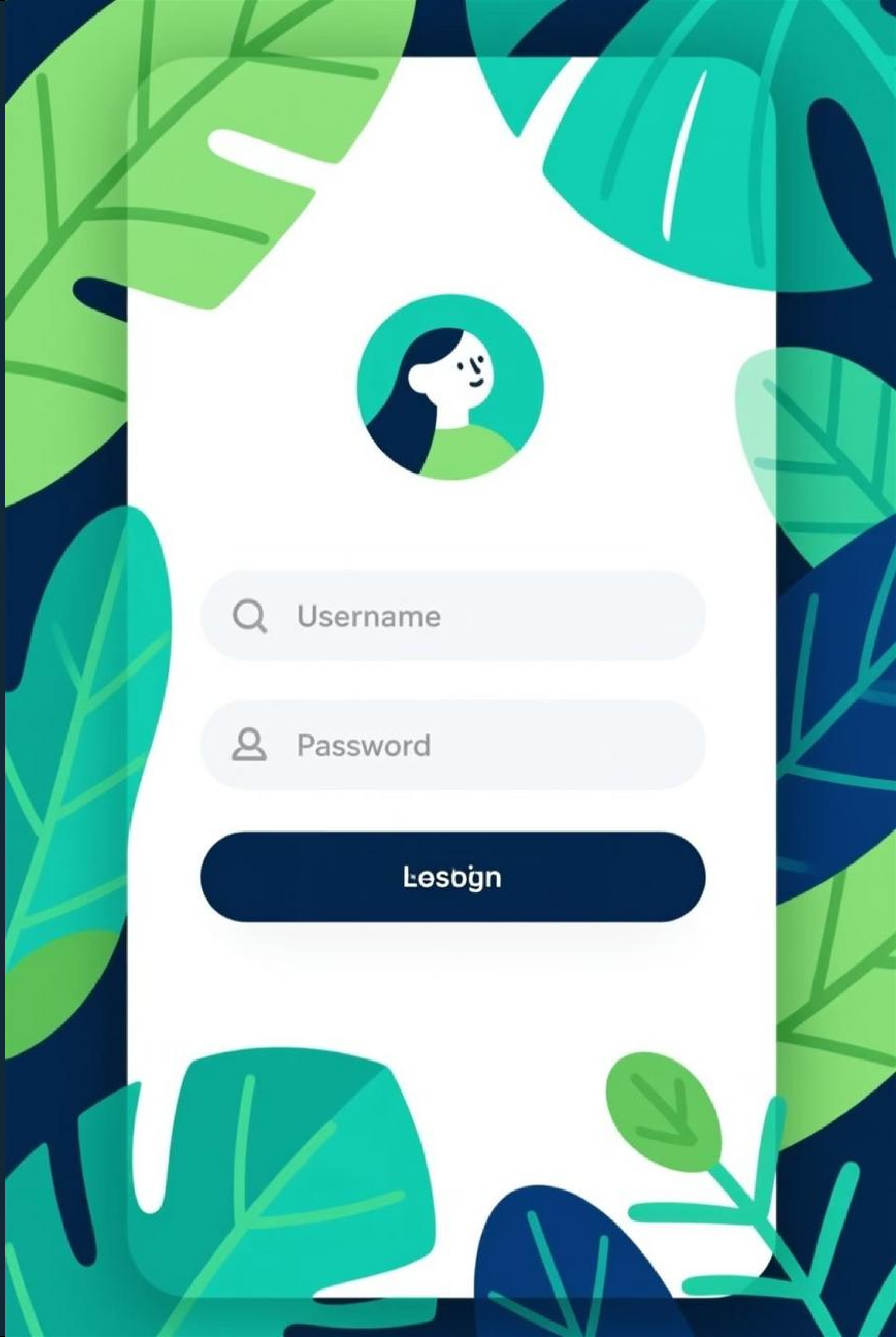
Challenges & Solutions

Dynamic Validation Messages

Handled with explicit waits and dynamic locators

Timing Issues

Resolved using explicit waits instead of hard sleeps



Module 2: PIM Module

Kerolos Samuel – Automation Tester

Page Objects Implemented

PimPage

Main PIM module page object

AddEmployeePage

Employee creation functionality

EmployeeListPage

Employee listing and search

Test Cases Automated

- Add Employee
- Search Employee
- Edit Employee
- Delete Employee
- Filter and advanced search

Implementation Highlights

Locator strategies for dynamic tables, reusable methods, explicit waits, actions, and data-driven testing

Module 3: Admin Module

Mohamed Goda – Automation Tester

Admin Page Responsibilities

Add User

Create new user accounts with roles and permissions

Search User

Locate users by various criteria

Edit User

Modify user details and access levels

Delete User

Remove user accounts from system

Technical Implementation

Locating dynamic table rows, handling dropdowns and user roles, managing complex interactions

Challenges & Solutions

Dynamic table elements handled with robust locator strategies, dropdown selections managed with explicit waits and action chains

Module 4: MyInfo Module

Zeyad Osama – Automation Tester

Page Objects Implemented

MyInfo Page
Main module object containing navigation and section access

Personal Details Page
Handles fields like name, employee ID, gender, nationality, DOB, and marital status

Contact Details Page
Phone numbers, addresses, email, and postal information.

Test Cases

Update Personal Details

Validate Minimum Age Requirement

Upload & Remove Profile Picture

Field Validation (required fields, formats, dropdown consistency)

Challenges & Solutions

Dynamic Form Behavior (Different sections load dynamically and require accurate waits)

✓ Solved using explicit waits, stable locators, and field readiness checks

Validation Rules (DOB, email format, phone formats, and required fields vary across sections)

✓ Implemented robust negative test scenarios and validation message assertions

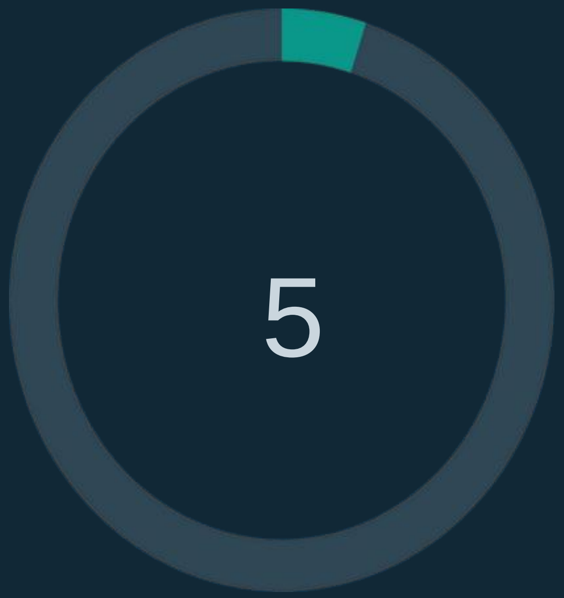
Test Cases Summary

Team Member	Module	Test Cases
Ahmed Ali	Login Module	3 scenarios
Kerolos Samuel	PIM Module	5 scenarios
Mohamed Goda	Admin Module	4 scenarios
Zeyad osama	MyInfo Module	1 scenario

Coverage Overview

Login Tests

Ahmed Ali

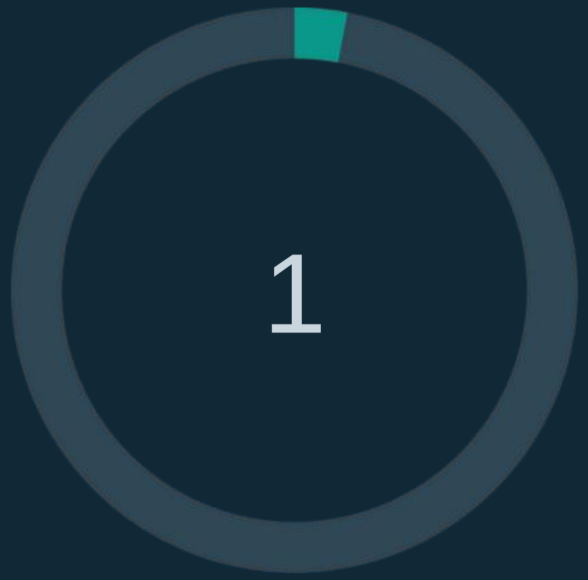


PIM Tests

Kerolos Samuel

Admin Tests

Mohamed Goda



MyInfo Tests

Zeyad Osama

Execution Flow

Launch WebDriver

Initialize browser and navigate to OrangeHRM

Login

Authenticate with valid credentials

Navigate Module

Access requested module (Login, PIM, or Admin)

Execute Scenario

Run automated test case steps

Validate Results

Assert expected outcomes

Generate Reports

TestNG reports with detailed results

Challenges, Solutions & Achievements

Challenges & Solutions

Dynamic Web Elements

Implemented robust locator strategies and dynamic XPath expressions

Slow Loading Pages

Solved with explicit waits and proper synchronization techniques

Repetitive Actions

Solved with reusable utilities and helper methods

POM Structure Consistency

Maintained clean architecture across all 3 modules

Final Results & Achievements

- Full automation coverage for Login, PIM, Admin modules
- Clean and scalable Page Object Model architecture
- Strong teamwork and clear task division
- High test pass rate across all scenarios
- Ready to expand automation to more modules