# INTRODUCTION

This test plan outlines the strategy, scope, tools, and approach for testing a web application that includes user login and product management functionalities (Create, Read, Update, Delete – CRUD). The objective is to ensure that both the UI and backend APIs function as expected, providing a secure, efficient, and seamless experience for end users. The testing will involve automated UI validation using Playwright and backend API verification using RestAssured.

# SCOPE

|  |  |
| --- | --- |
| In Scope | Out of Scope |
| 1. **UI Automation Testing** using Playwright:  * Form validations * Navigation and page flows * Success/error messages and alerts  1. **API Testing** using RestAssured:  * Authentication endpoints (login/logout) * Product endpoints (GET, POST, PUT, DELETE) * Status codes, schema validation, and response times | Performance/load testing of APIs or UI  UI testing on mobile browsers or devices  Database migration and upgrade testing |

# Testing Strategy

|  |  |
| --- | --- |
| UI Automation Testing (Playwright/js) | API Testing (RestAssured) |
| Login & Register form input validation  Product creation, editing, and deletion through the UI  CI/CD Integration: Tests will run on pull requests and nightly builds using GitHub Actions | Login API tested with valid/invalid credentials  Product CRUD API Tests:  Validate request/response structure and status codes  Negative testing with invalid inputs |

## Test execution

**UI testing**

# cd to playwright directory and do `npm install`

# To Run all UI tests use `npx playwright test`

# Run UI tests in headed mode for debugging `npx playwright test –headed`

# Run tests in specific browser `npx playwright test --project=chromium`

**API testing**

# cd to RestAssured directory and Build project (Maven) `mvn clean install`

# To Run tests `mvn test`

# Generate Allure report `mvn allure:serve`

# Risks/Assumptions

**Risks**

* UI elements may change frequently, causing automation flakiness
* API rate limits or authentication policies might block test execution
* Test environment may differ from production in configuration or data
* Delays in receiving updated test data or API documentation

**Assumptions**

* Stable test environment with predictable test data
* APIs and UI endpoints are version-controlled and backward-compatible
* Developer team provides prompt support for defect resolution
* SSO or complex auth mechanisms are out of current implementation scope

# Tools

| **Tool** | **Purpose** | **Why It’s Chosen** |
| --- | --- | --- |
| **Playwright** | UI automation (E2E flows, cross-browser testing) | Fast, modern, parallel tests, headless capability, Photo and Video capture. |
| **RestAssured** | API testing | Robust Java-based framework for API testing |
| **Allure** | Test reporting | Clean, visual reporting for both UI and API tests |
| **GitHub Actions** | CI/CD pipeline | Automates test execution on commits/pull requests |