# Assignment Overview

This document covers the detailed test plan and automation scripts for the CRUD Blockchain Request Portal – `xaltsocnportal.web.app`

# Part 1: Exhaustive Test Plan for End-to-End Testing

## 1. Objective

To test all user journeys and ensure the application behaves correctly in both success and failure scenarios.

## 2. Types of Testing

| Type of Testing | Purpose |
|---|---|
| **Functional Testing** | To verify that each function of the application works as expected. (e.g. Sign Up, Node submission, etc.) |
| **UI Testing** | To ensure all UI elements (buttons, forms, validations) are working and user-friendly. |
| **Boundary Testing** | To validate behavior when inputs are at their edge. (e.g. minimum and maximum characters in a Node ID) |
| **Negative Testing** | To ensure the application handles incorrect input and invalid actions. |
| **End-to-End Testing** | To validate complete user workflows from Sign Up to submission and Sign Out. |
| **Form Validation Testing** | To test field-level validation (e.g. invalid wallet format, empty fields). |
| **Security Testing** | To check session handling, password security, and data privacy. |
| **Cross-browser & Responsive Testing** | To ensure app works on all major browsers and screen sizes. |

| | |
|---|---|
| **Automation Testing** | To accelerate repetitive tests like login/logout via Selenium. |

## 3. Test Scenarios and Test Cases

**A. Sign Up Flow**

| Scenario | Input | Expected Result |
|---|---|---|
| Valid sign up | Valid email + password | Account created, redirected to Dashboard |
| Invalid email format | `abc.com` | Error: Please enter valid email |
| Weak password | `123` | Error: Password strength |
| Duplicate email | Already registered email | Error: user already exists |

---

**B. Sign In Flow**

| Scenario | Input | Expected Result |
|---|---|---|
| Valid login | Correct email & password | User lands on Dashboard |
| Wrong password | Valid email, wrong pass | Error: incorrect credentials |
| Non-existent user | Email not registered | Error: user not found |

| | | |
|---|---|---|
| Blank fields | Empty inputs | Form should not submit, show validation errors |

---

**C. Forgot Password Flow**

| Scenario | Input | Expected Result |
|---|---|---|
| Valid email | Registered email | Password reset link sent to email |
| Invalid email format | abc.com | Error: Enter a valid email |
| Non-registered email | Email not registered | Error: Email not found |
| Empty email field | No input provided | Error: Email field cannot be empty |

**D. Sign Out Flow**

| Scenario | Action | Expected Result |
|---|---|---|
| Click Sign Out | While logged in | User session ends, redirected to Login |
| Refresh after Sign Out | Reload browser | Should stay logged out |

---

**E. Node Onboarding to Existing Blockchain**

| Scenario | Action | Expected Result |
|---|---|---|
| Valid Node ID + IP | Add multiple nodes | Nodes are listed and added |
| Invalid IP address | e.g. `999.1.1.1` | Error message shown |
| Duplicate node | Same Node ID | Error or ignore |
| No nodes added | Try to click "Next" | Should block and show error |

---

**F. Wallet Addition Flow**

| Scenario | Action | Expected Result |
|---|---|---|
| Valid wallet & permission | Add multiple | All are added to list |
| Invalid wallet | e.g. `0xZZZ123` | Error: invalid wallet format |
| No wallets added | Try to proceed | Show error message |

---

**G. Create New Private Blockchain**

| Scenario | Action | Expected Result |
|---|---|---|
| Valid network + wallet | Add nodes & submit | Blockchain request sent |
| Empty fields | Try to proceed | Show required field errors |
| Add 0 nodes | Should block submission | Show error |

---

**H. End-to-End Flow**

| Scenario | Action | Expected Result |
|---|---|---|
| Complete onboarding flow | Add nodes + wallets + submit | Request successfully created |
| Create private chain | Add network + nodes + submit | New chain request submitted |
| Logout after request | Sign out and refresh | Should not stay logged in |

---

# Part 2: Automation Test Cases (Sign Up, Sign In, Sign Out)

### Test Case 1: Sign Up - Valid Input

| Field | Details |
|---|---|
| **Objective** | Verify user can sign up with valid inputs |
| **Test Steps** | 1. Navigate to Sign Up page. |
| | 2. Enter valid email and password. |
| | 3. Click Submit. |
| **Expected Result** | User is registered and redirected to the dashboard. |

---

### Code Snippet

```
@Test(dataProvider = "signUpData", dataProviderClass = TestDataUtil.class)

   public void testSuccessfulSignUp(String email, String password, String
confirmPassword) {
       System.out.println("Testing Successful Sign-Up: " + email);
       signUpPage.signUp(email, password, confirmPassword);
       if (isElementPresent(By.xpath("//h1[contains(text(),'Open Capital
Network')]"))) {
           Assert.assertTrue(true, "Successful signup scenario passed");
       } else {
           Assert.fail("Success message not displayed.");
       }
   }
```

---

*To view the complete suite of automation scripts, visit:*
https://github.com/Harshitha01/Blockchain-Request-Portal-Automation-Suite

---

#### Test Case 2: Sign Up - Duplicate Email & Password

- **Scenario**: Attempt to sign up with a duplicate email address and any password.

- **Expected Outcome**: The system should display an error message saying, “Provided E-Mail is already in use.”

**Test Case 3: Sign Up - Weak Password**

- **Scenario**: Attempt to sign up with a weak password.
- **Expected Outcome**: The system should display an error message saying, “`Password must contain at least one lowercase letter`”

**Test Case 4: Sign Up - Mismatched Password**

- **Scenario**: Attempt to sign up with password not matching with Confirm Password.
- **Expected Outcome**: The system should display an error message saying, “Passwords do not match.”

**Test Case 5: Sign In - Invalid Email**

- **Scenario**: Attempt to sign in with an incorrect email and the correct password.
- **Expected Outcome**: The system should display an error message saying, “`User not found”` alert box should popup.

**Test Case 6: Sign In - Invalid Password**

- **Scenario**: Attempt to sign in with an correct email and the incorrect password.
- **Expected Outcome**: The system should display an error message saying, “`Incorrect E-Mail or Password”` alert box should popup.

**Test Case 7: Sign In - Valid Credentials**

- **Scenario**: Sign in with valid email and password.
- **Expected Outcome**: The user should be successfully logged in and redirected to the dashboard.

**Test Case 8: Sign Out - Successful Log Out**

- **Scenario**: After logging in, click on the Sign Out button.

- **Expected Outcome**: The user should be signed out and redirected back to the login page.

# Test Automation Framework and Environment Setup :

## Automation Framework:

- **Selenium WebDriver**: Used for automating browser interactions.
- **TestNG**: Used for structuring and running the test cases.
- **Maven**: For dependency management and build automation.

## Testing Environment Setup:

- **Browsers**:
    - Google Chrome (Version 135)
- **Operating System**: Windows 11.
- **WebDriver Version**:
    - ChromeDriver (version 135)
- **Java Version**: JDK 11 or higher.
- **IDE**: Eclipse.