# Test Plan for Blockchain CRUD Web Application

**1. Test Plan Overview**

Purpose: This test plan outlines the strategy for end-to-end testing of the blockchain CRUD web application. The application enables users to sign up, sign in, onboard nodes to an existing blockchain, create a new private blockchain, and sign out. The testing strategy includes manual and automated tests to ensure robust functionality, reliability, and security of the application.

Scope: The testing will cover the following functionalities:

* User Sign Up
* User Sign In
* Onboarding Nodes
* Creating a New Private Blockchain
* User Sign Out

**2. Objectives**

* Ensure the application functions as expected.
* Validate the user experience and usability of the application.
* Identify and resolve any defects before deployment.
* Ensure the application is secure and performs well under load.

**3. Types of Testing**

**A. Functional Testing**

* **Objective:** Verify that all features work as expected according to the requirements.
* **Scope:** Sign Up, Sign In, Submit Request to Onboard Nodes, Submit Request to Create New Blockchain, and Sign Out.

**B. Integration Testing**

* **Objective:** Verify that different modules and components of the application work together seamlessly.
* **Scope:** Validate that integrated components function as expected when combined.

**C. End-to-End Testing**

* **Description:** Validate the complete workflow of the application from start to finish.
* **Scope:** A test case that simulates a user signing up, signing in, adding nodes, submitting a request to create a new private blockchain and signing out

**D. User Acceptance Testing (UAT)**

* **Description:** Ensure that the application meets business requirements and is ready for deployment.
  + **Scope:** Gather feedback from users on the usability of the sign-up and onboarding processes.

**E. Performance Testing**

* **Description:** Assess the application's responsiveness and stability under load.
* **Scope:** Ensure the application can handle expected user loads without performance degradation.

**F. Security Testing**

* **Description**: Identify vulnerabilities in the application.
* **Scope:** Ensure that the application is secure from common threats and vulnerabilities.

**G. Usability Testing**

* **Description:** Evaluate the application's user interface and user experience.
* **Scope:** Conduct user testing sessions to gather feedback on the sign-up and onboarding processes.

**H. Regression Testing**

* **Description:** Ensure that new changes do not adversely affect existing functionalities.
* **Scope:** After a new feature is added, re-test the sign-up, sign-in, and onboarding functionalities to ensure they still work as expected.

**I. Compatibility Testing**

* **Description:** Check how the application performs across different browsers, devices, and operating systems.
* **Scope:** Verify that the application works correctly on Chrome, Firefox, Safari, and Edge.

**4. Test Cases Overview**

**A. Sign Up**

* **Success Scenarios:**

TC1: User successfully signs up with a valid email and password.

* **Failure Scenarios:**

TC2: User attempts to sign up with an already registered email.

TC3: User attempts to sign up with an invalid email format.

TC4: User attempts to sign up with an invalid password pattern (e.g., less than 8 characters).

TC5: User attempts to sign up without providing an email or password.

**B. Sign In**

* **Success Scenarios:**

TC6: User successfully signs in with valid credentials.

* **Failure Scenarios:**

TC7: User attempts to sign in with an unregistered email.

TC8: User attempts to sign in with an incorrect password.

TC9: User attempts to sign in with an invalid email format.

**C. Onboard Nodes**

* **Success Scenarios:**

TC10: User successfully adds a node with valid Node ID and IP address.

TC11: User successfully adds multiple nodes and submits the request.

TC12: User attempts to add a node with a duplicate Node ID.

* **Failure Scenarios:**

TC13: User attempts to add a node with an invalid Node ID format.

TC14: User attempts to add a node with an invalid IP address format.

TC15: User attempts to submit without adding any nodes.

**D. Create New Private Blockchain**

* **Success Scenarios:**

TC16: User successfully creates a new private blockchain with valid inputs.

* **Failure Scenarios:**

TC17: User attempts to create a blockchain without providing a network name.

TC18: User attempts to create a blockchain with an invalid wallet address format.

TC19: User attempts to add nodes with invalid formats.

**E. Sign Out**

* **Success Scenarios:**

TC20: User successfully signs out of the application.

**5. Test Environment**

* **Browsers:** Chrome
* **Operating Systems:** Windows
* **Test Data:** Valid and invalid email addresses, passwords, Node IDs, IP addresses, and wallet addresses.

**6. Automation Framework**

* **Framework Choice:** Selenium WebDriver with TestNG
* **Programming Language:** Java

**7. Roles and Responsibilities**

* **Test Engineers:** Execute test cases and report defects.

**8. Schedule**

* Testing Start Date: 31-01-2025
* Testing End Date: 01-02-2025

**9. Approval**

* Stakeholders: Satish-xalts , Reuvab and Arun-Vallapu