## **1. Introduction**

### **1.1 Purpose**

The purpose of this test plan is to define the testing approach for the Blockchain Portal, a web application that allows users to onboard nodes to an existing blockchain or create new private blockchains.

### **1.2 Scope**

This test plan covers the functional, UI, security, and performance testing of the following features:

* User Registration (Sign Up)
* User Authentication (Sign In & Sign Out)
* Node Onboarding to an Existing Blockchain
* Private Blockchain Creation

### **1.3 Testing Objectives**

* Verify that the application functions as expected across different user actions.
* Ensure proper validation for Node ID, Wallet Address, and IP Address formats.
* Identify security vulnerabilities in authentication and data handling.
* Validate UI consistency and responsiveness.
* Ensure smooth performance under normal and high-load conditions.

### **1.4 Assumptions & Dependencies**

* The application should be accessible at xaltsocnportal.web.app.
* Test accounts should be created for execution.
* API documentation should be available for backend testing.
* Browser compatibility testing will focus on Chrome, Firefox, and Edge.

## **2. Testing Approach**

### **2.1 Testing Types**

#### **2.1.1 Functional Testing**

* Verify all user workflows (Sign Up, Sign In, Sign Out, Node Onboarding, Private Blockchain Creation).
* Validate input field restrictions and required field checks.

#### **2.1.2 UI/UX Testing**

* Ensure UI consistency across different devices and screen sizes.
* Verify navigation and error message clarity.

#### **2.1.3 Security Testing**

* Check for unauthorized access to blockchain records.
* Ensure password hashing and secure authentication.
* Test for SQL injection, XSS, and CSRF vulnerabilities.

#### **2.1.4 Performance Testing**

* Measure response times under normal and peak load.
* Test concurrent user handling.

#### **2.1.5 API Testing**

* Validate API request-response structure using Postman or similar tools.
* Ensure proper error handling for invalid API calls.

## **3. Test Environment**

### **3.1 Hardware & Software Requirements**

* **Operating System:** Windows / macOS / Linux
* **Browsers:** Chrome, Firefox, Edge
* **Automation Tools:** Selenium (Java/Python) / Cypress
* **Performance Tools:** JMeter (if required)

## **4. Test Case Strategy**

### **4.1 Test Cases Overview**

Each feature will have manual and automated test cases covering success and failure scenarios.

| **Feature** | **Test Type** | **Manual / Automated** |
| --- | --- | --- |
| Sign Up | Functional, Security | Both |
| Sign In | Functional, Security | Both |
| Sign Out | Functional | Both |
| Node Onboarding | Functional, UI | Manual |
| Private Blockchain Creation | Functional, UI | Manual |
| API Testing | API, Security | Manual |
| UI Testing | UI/UX | Manual |

## **5. Entry & Exit Criteria**

### **5.1 Entry Criteria**

* Test environment setup is completed.
* Access to the test application is available.
* Test cases are reviewed and approved.

### **5.2 Exit Criteria**

* All high-priority test cases have passed.
* Critical defects are fixed and verified.
* Test report is generated and reviewed.

## **6. Defect Management**

* All defects will be logged in a defect tracking tool (Jira, Bugzilla, etc.).
* Defects will be categorized based on severity and priority.

## **7. Deliverables**

* Test Plan Document
* Manual Test Cases Document
* Automated Test Scripts
* Test Execution Report
* Defect Report (if applicable)

## **8. Risks & Mitigation**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Application downtime | Schedule tests during off-peak hours |
| Unavailable API documentation | Work with developers for clarifications |
| Changing requirements | Follow Agile methodology and adapt test cases accordingly |

## **9. Conclusion**

This test plan ensures comprehensive testing of the Blockchain Portal, covering both manual and automated testing approaches. It aims to deliver a high-quality, secure, and reliable application for users.