# End-to-End Test Plan for Blockchain Request Portal

## ****Overview****

This test plan outlines the comprehensive testing strategy for the Blockchain Request Portal, ensuring all functionalities work as expected. The test plan includes different types of testing: manual and automated, to validate user actions, business logic, and security.

## ****Testing Types & Objectives****

### ****1. Functional Testing****

Validates that each feature functions according to the requirements.

### ****2. UI/UX Testing****

Ensures the user interface is intuitive, responsive, and provides a seamless experience.

### ****3. API Testing****

Verifies that API endpoints function correctly, return expected responses, and handle errors gracefully.

### ****4. Security Testing****

Identifies vulnerabilities such as SQL injection, authentication issues, and improper authorization.

### ****5. Performance Testing****

Checks application speed, scalability, and stability under varying loads.

### ****6. Compatibility Testing****

Ensures the application works correctly across different browsers, devices, and screen sizes.

### ****7. Regression Testing****

Ensures new changes do not break existing functionality.

### ****8. Automation Testing****

Covers repetitive scenarios, especially for login-related features.

## ****Test Scenarios & Cases****

### ****1. User Authentication (Sign Up, Sign In, Sign Out)****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| AUTH-001 | Register with valid email and password | Successful registration | Automated |
| AUTH-002 | Register with an already used email | Proper error message displayed | Automated |
| AUTH-003 | Login with correct credentials | Successful login | Automated |
| AUTH-004 | Login with incorrect credentials | Error message displayed | Automated |
| AUTH-005 | Logout functionality | User is redirected to login page | Automated |

### ****2. Node Onboarding Requests****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| NODE-001 | Add valid node details | Node added successfully | Automated |
| NODE-002 | Add node with invalid format | Error message displayed | Manual |
| NODE-003 | Add multiple nodes | All nodes appear in list | Automated |
| NODE-004 | Submit request with valid details | Request submitted successfully | Automated |
| NODE-005 | Submit request without nodes | Error message displayed | Manual |

### ****3. Private Blockchain Creation****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| PBC-001 | Enter valid network name and wallet | Proceed to next step | Automated |
| PBC-002 | Enter invalid wallet address | Error message displayed | Manual |
| PBC-003 | Add valid nodes | Nodes added successfully | Automated |
| PBC-004 | Submit request without required fields | Error message displayed | Manual |

### ****4. API Testing****

| **Test Case ID** | **Endpoint** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- | --- |
| API-001 | /auth/signup | Register new user | 201 Created response | Automated |
| API-002 | /auth/login | Authenticate user | 200 OK with token | Automated |
| API-003 | /nodes/add | Add node request | 201 Created response | Automated |
| API-004 | /blockchain/create | Create blockchain | 201 Created response | Automated |
| API-005 | /auth/logout | Logout user | 200 OK | Automated |

### ****5. Security Testing****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| SEC-001 | Attempt SQL injection | Input is sanitized, no SQL execution | Manual |
| SEC-002 | Brute force login attempts | Account locked after multiple failures | Manual |
| SEC-003 | Access unauthorized API | 403 Forbidden response | Manual |
| SEC-004 | Submit invalid token | Authentication failure | Automated |

### ****6. Performance Testing****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| PERF-001 | Load test with 100 users | No crash, response time under 2s | Manual |
| PERF-002 | Stress test with 500 requests/sec | System remains stable | Manual |
| PERF-003 | Simulate slow network | Application handles delay gracefully | Manual |

### ****7. Compatibility Testing****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| COMP-001 | Open app in Chrome, Firefox, Edge | UI renders correctly | Manual |
| COMP-002 | Test mobile responsiveness | UI adjusts to screen size | Manual |
| COMP-003 | Test different OS (Windows, Mac, Linux) | App behaves consistently | Manual |

### ****8. Regression Testing****

| **Test Case ID** | **Description** | **Expected Result** | **Type** |
| --- | --- | --- | --- |
| REG-001 | Re-test login after update | No functionality breaks | Automated |
| REG-002 | Re-test node submission after fix | Works as expected | Automated |
| REG-003 | Check for UI consistency | No design issues | Manual |

## ****Automation Framework & Tools****

* **Selenium WebDriver** for UI automation
* **Postman** for API testing
* **JMeter** for performance testing
* **Playwright** for cross-browser testing
* **Jest/Cypress** for frontend testing