**Test Plan for Xalts Blockchain Request Application**

**# Objective**

To ensure the quality, reliability, and performance of the blockchain request application through rigorous testing of core functionalities: Sign Up, Sign In, Sign Out, Node Onboarding, and Private Blockchain Creation.

**# Scope of Testing**

* Functional Testing
* UI/UX Testing
* Validation Testing
* Negative Testing
* Boundary Testing
* Input Field Testing
* API Testing
* Regression Testing
* Cross-browser Testing
* Security Testing (basic checks)
* Automation (Sign In, Sign Up, Sign Out)

**# Types of Testing and Test Cases**

**1. Functional Testing**

Functional testing validates the core functionalities of the application against the requirements. It ensures features like sign up, sign in, and request submissions behave as expected under both normal and edge-case conditions.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps to Execute | Expected Result | Actual Result | Status |
| TC\_001 | Sign Up | Valid Sign Up | Navigate to Sign Up > Enter valid email & password > Click Sign Up | User is registered and redirected to dashboard | As expected | Pass |
| TC\_002 | Sign Up | Sign Up with invalid email | Enter invalid email format (e.g., test@com) > Click Sign Up | User should not be able to login | Next Button is disabled | Pass |
| TC\_003 | Sign Up | Sign Up with weak password | Enter weak password (e.g., 12345) > Click Sign Up | Validation message: "Password must contain atelast one lowercase letter, uppercase letter, number and special character and be a minimum of 8 characters in length” | As expected | Pass |
| TC\_004 | Sign In | Valid Sign In | Navigate to Sign In > Enter valid credentials > Click Sign In | Redirected to dashboard | As expected | Pass |
| TC\_005 | Sign In | Invalid Credentials | Enter wrong credentials > Click Sign In | User should not be able to login  Error message ”Incorrect E-Mail or Password: | Next Button is disabled  Error message ”Incorrect E-Mail or Password: | Pass |
| TC\_006 | Sign Out | Sign Out functionality | Login > Click on Sign Out | User redirected to login page | As expected | Pass |
| TC\_007 | Onboard Node | Add valid node and wallet details and submit | Navigate to onboarding > Enter Node ID, IP, Wallet address > Submit | Request submitted successfully | User stays at the Review Page | Fail |
| TC\_008 | Onboard Node | Enter invalid node ID format | Enter "Node123" instead of "NodeID-001" | User should not be allowed to navigate to the next screen | Next button is disabled | Pass |

**2️. UI/UX Testing**

UI/UX testing evaluates the application's interface and user experience. It checks for alignment, responsiveness, consistency in colors/fonts, and proper placement of UI elements across different resolutions.

**Test Cases:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status | | TC\_009 | Sign Up | UI elements rendering | Open Sign Up page and observe layout | All fields/buttons properly displayed | As expected | Pass | | TC\_010 | Dashboard | UI rendering after login | Login and observe layout | Elements displayed properly, no overlap | As expected | Pass | |  |

**3️. Boundary Testing**

Boundary testing focuses on the edge limits of input fields,like testing minimum/maximum character limits in input fields. It helps catch off-by-one errors and validate how the system behaves at extreme ends.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_011 | Sign Up | Password minimum length | Enter 5-char password | Rejected with error | As expected | Pass |
| TC\_012 | Sign Up | Password exact min length (8 chars) | Enter 8-char valid password | Accepted if within policy | As expected | Pass |

**4️. Security Testing**

Security testing verifies that the application protects sensitive data and user actions properly. It checks for common vulnerabilities like SQL injection, authentication bypass, and unauthorized access.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_013 | Sign In | Brute force attempt | Try incorrect login 5+ times | Account locked or CAPTCHA triggered | As expected | Pass |

**5️. Smoke Testing**

Smoke testing is a quick, high-level check to ensure the basic functionalities of the application (like sign up, sign in, and navigation) work correctly after a build or deployment. It acts as a gatekeeper before deeper testing begins.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_014 | Basic Flow | End-to-end flow | Sign Up > Sign In > Create Request > Sign Out | All steps work without errors | As expected | Pass |

**6. Regression Testing**

Regression testing ensures that previously working functionalities continue to work after new changes or enhancements. It helps identify bugs that may have been unintentionally introduced.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_015 | Sign In | Use newly created account | Create account > Logout > Try login again | Login successful | As expected | Pass |

**7. Cross-Browser Testing**

Cross-browser testing ensures that the web app works seamlessly across different web browsers like Chrome, Firefox, Safari, and Edge, maintaining consistent functionality and appearance.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_016 | Sign In | Browser compatibility | Try sign in on Chrome, Firefox, Edge | Should work across all supported browsers | As expected | Pass |

**8. Input Validation Testing**

Input validation testing ensures that all user inputs (email, password, node ID, wallet address, etc.) are correctly validated. It verifies that invalid data is rejected with appropriate error messages and valid data is accepted.

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC ID | Module | Test Case Description | Steps | Expected Result | Actual Result | Status |
| TC\_013 | Onboard Node | Invalid IP Address | Enter IP as 300.168.1.1 | Validation error displayed | As expected | Pass |
| TC\_014 | Onboard Node | SQL Injection in Wallet Address | Enter wallet: 0x88fa61d2faA13aad8Fbd5B030372B4A159BbbDFb' OR '1'='1 | Should be sanitized and rejected | As expected | Pass |

**9. Automation Testing Plan**

**Sample Test Scenarios for Automation:**

| **Scenario** | **Steps** | **Expected** |
| --- | --- | --- |
| Sign Up Success | Enter valid email & password > Submit | Redirect to dashboard |
| Sign Up Fail  Sign In | Email already registered  Enter valid credentails | Show error  User should be able to Login |
| Sign In Fail | Wrong password/Email | Show invalid credentials |
| Sign Out | Click Sign Out | Redirect to Login |

**# Tools & Stack**

* **Katalon Stadio** (Web Automation)
* **Groovy**
* **Git** (version control)
* **Postman** (API testing)
* **JIRA** / Bug tracking if integrated