Test Plan for CRUD Web Application

1.Overview

This test plan outlines comprehensive strategy for end-to-end testing of a CRUD web application that enables users to manage blockchain nodes and create private blockchains. The goal is to ensure all functionalities work as expected, providing a seamless user experience.

2.Objectives

- Validate the complete workflow of user the actions from account creation to request submission.

- Ensure data integrity and the functionality of CRUD operations.

- Identify any issues in user interactions, data handling, and system responses.

3.Scope

The testing will cover the following functionalities:

1. User Sign Up

2. User Sign In

3. Submit Request to Onboard Nodes

4. Submit Request to Create New Private Blockchain

5. User Sign Out

4.Resources

The List of the Browsers:

Name of Browser	Version
Chrome	Latest
Firefox	Latest
Edge	Latest
Safari	Latest

List Of Devices:

Name of Devices	OS
I phone	All supported OS
Android	All supported OS

5.Types of Testing:

1. Functional Testing

  - Scenario: Verify that each feature operates according to specified requirements.

  - Test Cases:

    - Successful account creation with valid email and password.

    - Failure scenarios such as existing email during sign-up.

    - Successful login with valid credentials.

    - Negative scenarios such as incorrect password during login.

    - Successful submission of requests to onboard nodes and create new private blockchains.

2. Usability Testing

  - Scenario: Ensure the application is user-friendly and intuitive.

  - Test Cases:

    - Evaluate ease of navigation through various sections.

    - Assess clarity and helpfulness of error messages.

5. Compatibility Testing

  - Scenario: Ensure application functionality across different browsers and devices.

  - Test Cases:

    - Verify functionality on major browsers (Chrome, Firefox, Safari, Edge).

    - Test responsiveness on mobile devices.

6. Automated Testing

  - Scenario: Automate repetitive test cases for efficiency.

  - Test Cases:

    - Automated scripts for sign-up, sign-in, and sign-out functionalities.

6.Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

The **steps for reporting a defect**, such as using a designated template, providing detailed reproduction steps, and attaching screenshots or logs.

The **process for triaging and prioritizing defects, s**uch as assigning severity and priority levels, and assigning them to the appropriate team members for investigation and resolution.

The **tools and systems** that will be used for tracking and managing defects, such as a defect tracking software or a project management tool.

The **roles and responsibilities of the team members** involved in the defect reporting process, such as testers, developers, and the test lead.

7.Test Case Design

Develop comprehensive test cases covering all possible user scenarios:

- Valid inputs for all fields (e.g., email format, password strength).

- Invalid inputs (e.g., incorrect email format, empty fields).

- Edge cases (e.g., maximum character limits).

8.Detailed Test Scenarios

1. User Sign Up

- Success Scenario:

  1. Navigate to the sign-up page.

  2. Enter a valid email address and password.

  3. Click "Sign Up."

  4. Verify account creation confirmation message.

- Negative Scenarios:

  1. Attempt to sign up with an already registered email address.

  2. Attempt to sign up with an invalid email format.

  3. Attempt to sign up without entering a password.

2. User Sign In

- Success Scenario:

1. Navigate to the sign-in page.
2. Enter valid credentials (email and password).
3. Click "Sign In."
4. Verify successful login by checking for dashboard access.

- Negative Scenarios:

1. Attempt to log in with incorrect password.
2. Attempt to log in with unregistered email address.

3. Submit Request To Onboard Nodes

- Success Scenario:

1. Log in to the application.
2. Navigate to the onboarding request page.
3. Enter valid node details (Node ID and public IP).
4. Click "ADD NODE" and verify addition to the list.
5. Repeat for multiple nodes, then click "NEXT."

- Negative Scenarios:

1. Attempt to add a node with an invalid Node ID format.
2. Attempt to add a node with an invalid IP address format.

4. Submit Request To Create New Private Blockchain

- Success Scenario:

1. Log in to the application.
2. Navigate to create new blockchain section.
3. Enter network name and wallet address correctly.
4. Add nodes as described in previous scenarios, then click "SUBMIT."

- Negative Scenarios:

1. Attempt to create a blockchain without entering network name or wallet address.

5.User Sign Out

- Success Scenario:

1. Click on "Sign Out."
2. Verify successful logout message or redirection to login page.

9.Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

Requirement Analysis

Entry Criteria:

- Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

- List of Requirements are explored and understood by the Testing team
- Doubts are cleared.

10.Test Execution

Entry Criteria:

• Test Scenarios and Test Cases Documents are signed-off by the Client

• Application is ready for Testing

Exit Criteria:

• Test Case Reports, Defect Reports are ready

11.Test Closure

Entry Criteria:

• Test Case Reports, Defect Reports are ready

Exit Criteria:

• Test Summary Reports