Test Plan

1. Objective.
The objective of this test plan is to ensure the successful implementation and functionality of the token trading and transaction feature. This feature will allow users to buy and sell tokens and perform transactions with other users on the platform.

2. Scope.
The scope of this test plan covers the end-to-end testing of the token trading and transaction feature, including both positive and negative test cases.

3. Test Environment.
The testing will be conducted in a staging environment that closely replicates the production environment. The environment will include:
- Application server
- Database server
- User accounts with varying roles (buyers, sellers, traders)
- Simulated tokens for trading

4. Test Cases
4.1 User Creation
Test Case 1.1.1: Users can create an account
- Steps:
      Open registration form
      Fill in all required details
      Click Register
- Expected Result: User can create a new account/be redirected to User profile
4.2 Token Buying and Selling
4.2.1 Positive Test Cases
Test Case 1.1.1: Users can search for available tokens to buy.
- Steps:
      Log in with a buyer account.
      Navigate to the token trading section.
      Search for available tokens.
- Expected Result: User can see a list of available tokens for purchase.
Test Case 1.1.2: User can initiate a token purchase.
- Steps:
      Log in with a buyer account.
      Select a token to purchase.
      Enter the quantity to buy.
      Confirm the purchase.
- Expected Result: User receives the purchased tokens in their account balance.
Test Case 1.1.3: User can put tokens up for sale.
- Steps:

Log in with a seller account.
Navigate to the token trading section.
Select a token to sell.
Enter the quantity to sell and the selling price.
Confirm the listing.
- ● Expected Result: The token is listed for sale and available to potential buyers.

4.2.2 Negative Test Cases

Test Case 1.2.1: Insufficient funds during token purchase.
- ● Steps:
Log in with a buyer account.
Select a token to purchase.
Enter a quantity that exceeds the account balance.
Confirm the purchase.
- ● Expected Result: User receives an error indicating insufficient funds.

Test Case 1.2.2: Invalid token quantity during sale.
- ● Steps:
Log in with a seller account.
Select a token to sell.
Enter a quantity that exceeds the available tokens.
Confirm the listing.
- ● Expected Result: User receives an error indicating insufficient tokens for sale.

4.3 User Transactions

4.3.1 Positive Test Cases

Test Case 2.1.1: User can initiate a transaction with another user.
- ● Steps:
Log in with a trader account.
Navigate to the user's profile to initiate a transaction.
Select tokens to trade and specify the quantity.
Confirm the transaction.
- ● Expected Result: Both users' token balances are updated accordingly.

Test Case 2.1.2: User can see transaction history
- ● Steps:
Log in with a trader account.
Navigate to the user's profile
Open History details
- ● Expected Result: List of transactions is displayed with all details (amount, from, to, date)

4.3.3 Negative Test Cases

Test Case 2.2.1: Invalid token quantity in a transaction.
- ● Steps:
Log in with a trader account.
Initiate a transaction with another user.

Enter a token quantity that exceeds the user's balance.
Confirm the transaction.
- Expected Result: User receives an error indicating insufficient tokens for the transaction.

5. Performance Testing.
Perform load testing with multiple users engaging in token trading and transactions simultaneously to ensure that the system can handle the expected load without slowdowns or errors.

6. Security Testing. Conduct security testing to ensure that user data and transactions are secure, and no vulnerabilities exist that could compromise the integrity of the system or user accounts.

7. User Acceptance Testing (UAT) Involve a group of representative users to perform real-world scenarios on the feature and provide feedback on usability and functionality.
8. Conclusion By following this test plan, we aim to ensure that the token trading and transaction feature is thoroughly tested, both from a functional and non-functional perspective, before its release to the production environment. Any issues identified during testing will be addressed and resolved to ensure a smooth user experience.