BLOCKCHAIN TESTS 1 – BLUME TOKEN

CHECKLIST FOR SUBMISSION REVIEW

Candidate Name:Mohammed Jabbir

Date: 15-07-2025

IMPORTANT NOTE
Complete the whole of this document. It is an integral part of your test and validly confirms your skill/abilities. Candidates who do not submit the checklist and proof of original work are deemed to have failed.

Here’ s a detailed checklist for candidates to submit as part of their task test review process. This will ensure submissions meet the requirements and evaluation criteria outlined in the task description:

Submission Checklist

To successfully complete the **BLOCKCHAIN TESTS 1 – BLUME TOKEN**, you must submit the following deliverables. Ensure that each submission is well-organized and properly documented.

---

1. Smart Contract Source Code (GitHub Repository)

You must provide the complete **Solidity-based smart contract source code** for BLUME TOKEN (BLX), along with the supporting contracts for:

- **BLX Token (ERC-20).**
- **Liquidity Pool contracts** for BLX token swaps and rewards.
- **Vault contracts** for secure token storage.
- **Staking and Liquid Staking contracts** for earning rewards.

What we expect:

Well-structured, optimized, and properly commented Solidity code.

Implementation of security best practices (e.g., reentrancy protection, access control, and gas optimization).

Contracts should be tested and verified on a public Ethereum Testnet (Ropsten, Goerli, Sepolia).

README file in the repository explaining how to compile, deploy, and interact with the contracts.

Accepted Formats: GitHub Repository (preferred) containing the smart contract files.

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

---

2. Frontend & Backend Code (GitHub Repository)

Your **React.js frontend** must be capable of interacting with the deployed BLX smart contracts.

Frontend (React.js) Expectations:

User Interface for interacting with smart contracts, including:

- Token balance display.
- Staking and Liquid Staking Interaction.
- Liquidity Pool deposit and withdrawal functionality.
- Vault deposit and withdrawal management.

MetaMask Integration for user transactions.

Frontend must display relevant contract data using Web3.js or Ethers.js.

Backend Expectations:

API for smart contract interaction, handling:

- Retrieving on-chain data.
- Processing user transactions.
- Managing staking rewards and liquidity pool data.

Security Measures, including input validation and rate limiting.

Accepted Formats: GitHub Repository (preferred) file containing the frontend and backend source code.

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

---

3. Security Audit Report (Documenting Tests Conducted and Results)

A comprehensive **security audit report** of your smart contracts, covering:

Security Analysis & Tests Conducted

Audit readiness verification: Use tools like Slither, MythX, or CertiK for static analysis.

Manual security review for vulnerabilities like **reentrancy, flash loan attacks, integer overflows/underflows, front-running, and gas inefficiencies**.

Test case results demonstrating that the smart contract is free of major security issues.

Breaches & fixes: If you identify security weaknesses, explain how you mitigated them.

Accepted Formats: PDF file (detailed with screenshots of testing results).

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

BLXToken:
https://sepolia.etherscan.io/address/0x0BDA83f742ba63B38a0E51b7F7253a393441fCD9#code

Vault: https://sepolia.etherscan.io/address/0xcDaEd6dd4251B6f602f037BB6E8dD39B4985054B#code

---

4. Deployment Details (Testnet Address, Transaction Hashes, and Contract Verification)

Your contracts must be deployed on a **public testnet** and verified on an **explorer**). **What to include:**

Deployed Smart Contract Addresses (BLX Token, Staking, Vault, and Liquidity Pool Contracts).

Transaction Hashes for smart contract deployment.

Link to Verified Contracts on Etherscan to enable contract interaction.

Instructions on how to interact with the deployed contracts via a blockchain explorer.

Accepted Formats: A structured document (PDF) with all relevant links.

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

BLXToken:
https://sepolia.etherscan.io/address/0x0BDA83f742ba63B38a0E51b7F7253a393441fCD9#code

Vault:
https://sepolia.etherscan.io/address/0xcDaEd6dd4251B6f602f037BB6E8dD39B4985054B#code

---

5. Video Showcase Link (YouTube/Vimeo Private Video)

You must submit a detailed **video showcase** that clearly demonstrates the features and functionalities of your

implementation. The video should provide a step-by-step explanation of the developed components, highlighting how they work and the approach taken.

Instead of creating one long video, you can submit multiple shorter videos, each focusing on a specific feature or functionality. This will make it easier to review and understand the different aspects of the project.

The video must include a clear voiceover explanation, where you walk through your implementation, justifying your decisions and the logic behind your work. All videos should be uploaded as **private or unlisted on YouTube or Vimeo**, with the **links shared** in the submission.

- **Demonstrates the contract deployment process.**
- **Showcases the frontend interface** and **how users interact with BLX staking, vaults, and liquidity pools**.
- **Explains security testing** and audit procedures.

Video Submission Guidelines:

Video must be uploaded privately to YouTube or Vimeo.
Enable access via a shareable link.

Accepted Format: Private YouTube or Vimeo link (shared with us).

https://youtu.be/QkiiHkMCgUQ

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

---

6. Documentation File (Justifying Decisions & Implementation Approach)

You must submit a **technical documentation file** explaining the reasoning behind your implementation, security choices, and DeFi mechanisms.

What to include:

Technical Overview: A high-level explanation of the BLX smart contract architecture.
Liquidity Pool Design: Explanation of how rewards and swaps are handled.
Staking Mechanism: How users stake, earn rewards, and withdraw BLX tokens.
Security Strategy: Summary of **implemented security measures** and audit findings.
Performance Optimization: Steps taken to minimize **gas fees and enhance contract efficiency**.

Accepted Formats: PDF document.

https://github.com/Mdjabirshaik/MohammedJabbir_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmission-

Task Submission Checklist

#	REQUIREMENT	COMPLETED	DEMONSTRATED	ATTACHED
1	Smart contract source code & proof of verification	☐	☐	☐
2	Liquidity Pool, Vault, Staking Contracts	☐	☐	☐
3	React.js Integration	☐	☐	☐
4	Security audit report	☐	☐	☐
5	Deployment details	☐	☐	☐
6	Video showcase link	☐	☐	☐
7	Documentation file	☐	☐	☐

Number of Required Submissions

ITEM	NUMBER REQUIRED
Smart Contract Code and verification proof	1 (Deployed & Secured)
Frontend & Backend Code	1 (Functional with UI)
Security Audit Report	1 (Detailed findings)
Deployment Details	1 (Contract Address & TX Hashes)
Video Showcase	1 (Private YouTube/Vimeo Link)
Documentation File	1 (Explanation & Justification)

Important Notes

- **Naming Convention:** Use the format [YourName]_BLOCKCHAIN_TEST_BLUME_TOKEN_TaskSubmissionfor your folder.
- **Deadline:** All submissions must be completed within **24 hours**.
- Failure to comply will result in **disqualification**.

If you have any questions, contact us before the submission deadline.

CANDIDATE DECLARATION

I confirm that I have submitted all required deliverables for **BLOCKCHAIN TESTS 1 – BLUME TOKEN**

as outlined above.

Candidate Name: Mohammed Jabbir

Date: 15-07-2025

Final Notes

Ensure that all files, codebases, and reports are **properly structured, well-documented, and submitted before the deadline. Failure to meet the submission requirements will affect your evaluation score.**

Good Luck!