Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : **Peer Audit – Contract Security Review**

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Introduction:

The Peer Audit – Contract Security Review phase focuses on evaluating smart contracts for potential vulnerabilities, logic flaws, and compliance with secure coding standards.
Through peer review, contracts are analyzed for reentrancy, overflow/underflow, access control, and gas optimization issues to ensure robust and reliable blockchain applications.
This collaborative verification enhances code quality, transparency, and trust in decentralized ecosystems.

Algorithm / Procedure:

1. **Select Contract for Review:**
   Choose the deployed or completed smart contract for security testing.
2. **Static Code Analysis:**
   Review the Solidity code manually or with tools (e.g., Remix Analyzer, MythX, or Slither) to detect syntax errors and potential vulnerabilities.
3. **Identify Vulnerabilities:**
   Check for common issues like:
   - Reentrancy attacks
   - Integer overflow/underflow
   - Access control misconfigurations
   - Unchecked external calls
4. **Run Security Tools:**
   Use **Remix IDE's "Solidity Static Analysis" plugin** to automatically scan for vulnerabilities and performance issues.
5. **Peer Review & Documentation:**
   Collaborate with peers to cross-check code logic, confirm fixes, and document findings with suggested improvements.
6. **Verification:**

* Softwares used

- **Visual Studio Code (VS Code)** – for writing and testing smart contracts.
- **MetaMask** – to connect and deploy contracts on test networks.
- **Hardhat** – for compiling, testing, and debugging smart contracts locally.
- **Solidity Compiler (solc)** – integrated for smart contract compilation.
- **Slither / MythX** – for performing smart contract security analysis.

******As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

Blockchain Security Audits ensure that smart contracts and blockchain systems are secure, efficient, and error-free before deployment. The process identifies vulnerabilities, improves trust, and maintains compliance through systematic checks.

1 Penetration Testing

- Simulates real-world attacks to find weak points.
- Tests network and contract defense strength.
- Ensures system resistance to hacking.

2 Code Review

- Line-by-line inspection of smart contract code.
- Detects logic errors, bugs, and vulnerabilities.
- Ensures security and functional correctness.

3 Threat Modeling

- Predicts possible attack paths and weak spots.
- Prioritizes high-risk areas for protection.
- Helps design proactive defense strategies.

4 Architecture Analysis

- Reviews overall network and contract design.
- Checks cryptography, consensus, and data flow.
- Confirms secure, scalable, and stable setup.

◇ Final Output (No Error):

- All security tests passed successfully.
- No major bugs or vulnerabilities found.
- Smart contracts verified and deployment-ready.

***As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.*

* Implementation Phase: Final Output (no error)

- **Vulnerability Detection & Risk Assessment:**
Conducted systematic audits to identify potential code flaws, access control issues, and logic vulnerabilities in smart contracts to ensure security and integrity.

- **Performance & Compliance Verification:**
Validated contract behavior under various conditions, ensuring efficient gas usage, proper execution flow, and adherence to blockchain security standards.

- **Code Validation & Peer Review:**
Cross-checked contract logic and functionality through peer audits, confirming alignment with best practices and eliminating inconsistencies before deployment.

- **Scalability & Security Enhancement:**
Established a repeatable auditing framework promoting continuous improvement, faster debugging, and long-term smart contract reliability.

* Observations

- Peer auditing helped in identifying hidden vulnerabilities and improving the overall security of smart contracts.

- Cross-verification by multiple reviewers ensured accuracy, transparency, and code reliability.

- The audit process enhanced understanding of secure coding practices and strengthened deployment readiness.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :

****As applicable according to the experiment.***
Two sheets per experiment (10-20) to be used.