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

Algorithm

1.Choose Contract for Assessment: Pick the deployed or finished smart contract for security evaluation.
2.Manual Code Examination: Examine the Solidity code manually or using analysis tools (such as Remix Analyzer, MythX, or Slither) to find syntax issues and possible security gaps.
3.Detect Security Weaknesses: Look for typical problems including: ○ Reentrancy vulnerabilities ○ Integer overflow/underflow issues ○ Permission control errors ○ Unverified external function calls
4.Execute Security Analysis Tools: Apply Remix IDE's "Solidity Static Analysis" feature to automatically identify vulnerabilities and efficiency problems.
5.Collaborative Review & Record Keeping: Work with team members to validate code functionality, verify corrections, and record discoveries with recommended enhancements.
6.Final Validation: Execute tests again and redeploy the contract to confirm that all detected problems have been fixed.

* Softwares used

1.Solidity
2.Hardhat
3.VS code
4.MetaMask

*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.

* 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 :

Regn. No. :

Signature of the Faculty:

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