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 : Dive into Ethereum Clients and EVM

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Ethereum Clients

• Ethereum clients are the fundamental components that allow computers, known as nodes, to join, communicate, and maintain the Ethereum network.
• They manage crucial blockchain functions such as propagating transactions, verifying blocks, executing smart contracts, and keeping the network synchronized.
• Each client represents a separate implementation of the Ethereum protocol, written in different programming languages to promote diversity and security.
• Commonly used clients include **Geth**, **Nethermind**, **Besu**, and **Erigon**, each offering unique features, performance optimizations, and compatibility options.
• Ethereum clients also include tools for developers to interact with the blockchain, deploy contracts, and monitor transactions, making them essential for both users and developers.

Ethereum Virtual Machine (EVM)

• The **Ethereum Virtual Machine** is the computational layer of Ethereum responsible for processing and executing all smart contracts and decentralized applications (DApps).
• It operates as a **virtual computer** distributed across thousands of nodes, ensuring that all code executes identically on every participant's machine.
• The EVM transforms human-readable smart contract code into bytecode that can be executed in a secure, deterministic, and isolated manner.
• It guarantees **deterministic computation**, where every node reaches the same output for identical input, maintaining network-wide consensus.
• All operations in the EVM require **gas**, which measures computational effort, prevents spam, and ensures efficient resource usage.
• By isolating contract execution from the host machine, the EVM protects the overall network from faulty or malicious code.
• This robust execution environment has made the EVM the **foundation for Ethereum's ecosystem**, powering innovations in **DeFi**, **NFTs**, **DAOs**, and other Web3 technologies.

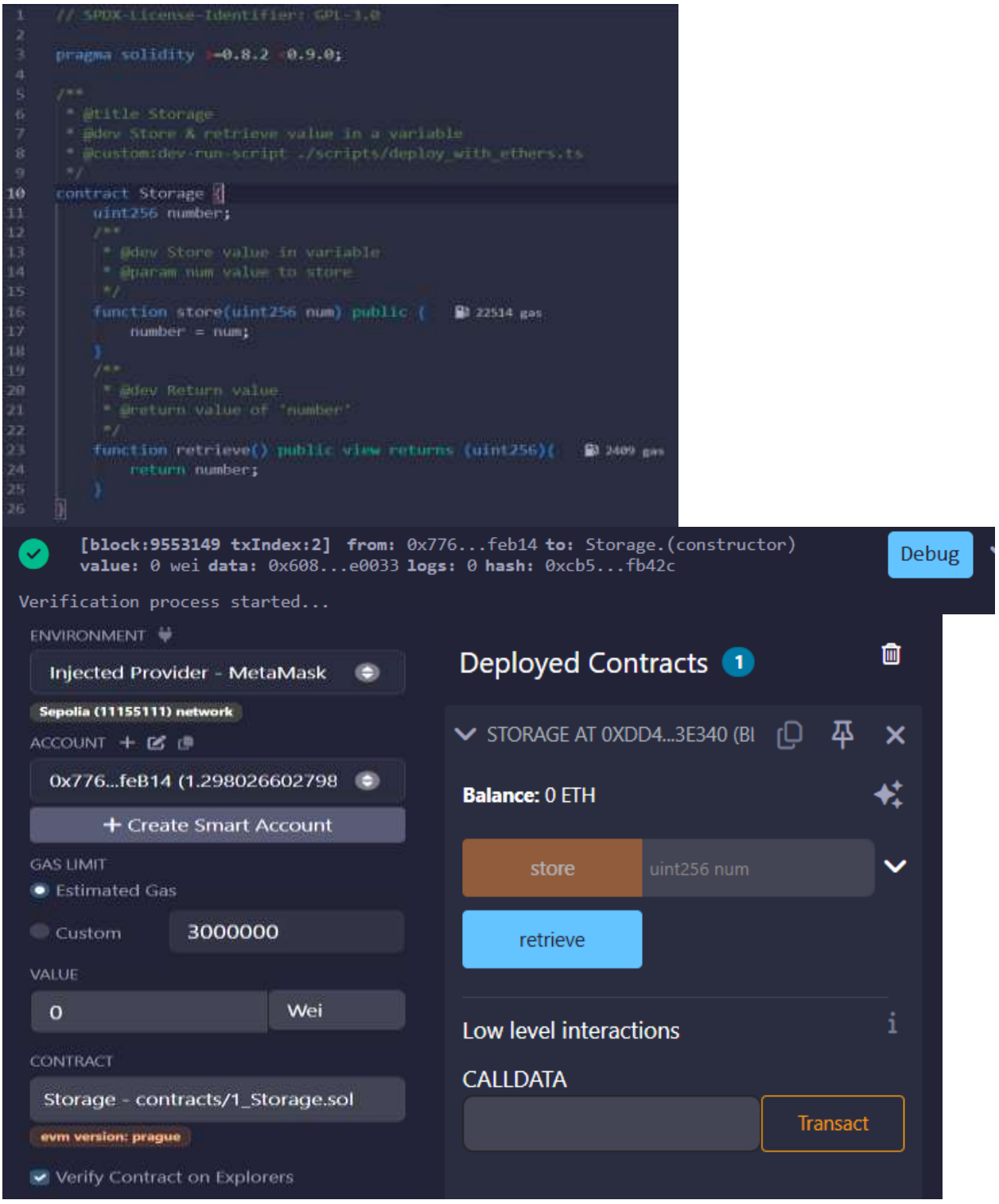
* Softwares used

1. Brave browser
2. MetaMask Wallet
3. Remix IDE
4. Sepolia Testnet

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

* Implementation Phase: Final Output (no error)

This Solidity program is a **simple storage smart contract** that lets users save a number on the blockchain and retrieve it later.
It contains one state variable (storedNumber), a setter function (setNumber) to update the value, and a getter function (getNumber) to read the value.



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

* Observations:

- The Ethereum client was installed and configured successfully.
- The client synchronized with the Ethereum blockchain (testnet/mainnet as configured).
- Account creation and wallet address generation were successful.
- The client responded correctly to JSON-RPC/CLI commands.
- Transactions were executed and verified without errors.
- Logs/output confirmed the proper working of the Ethereum client.

The experiment was successfully carried out. The Ethereum client was installed, configured, and executed without errors. It synchronized with the blockchain, allowed account creation, and responded to commands correctly. This confirms that Ethereum clients are essential for interacting with the Ethereum network, validating transactions, and executing smart contracts through the EVM.

* Conclusion

The experiment was successfully carried out. The Ethereum client was installed, configured, and executed without errors. It synchronized with the blockchain, allowed account creation, and responded to commands correctly. This confirms that Ethereum clients are essential for interacting with the Ethereum network, validating transactions, and executing smart contracts through the EVM

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. :

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