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 : Dive into Ethereum – Clients and EVM

Pseudo Code / Algorithm:

1. Start
2. Learn about Ethereum architecture
3. Identify different types of Ethereum Clients
4. Explore Ethereum Virtual Machine (EVM)
5. Understand how smart contracts are executed by the EVM
6. End

Apparatus/Software Used:

- Remix IDE
- MetaMask
- Ganache / Hardhat
- Web Browser

Theory concept:

1. Ethereum Clients

- Ethereum clients are software programs that connect to the Ethereum network.
- They store blockchain data, validate transactions, and execute smart contracts.

2. Ethereum Virtual Machine (EVM)

- The EVM is the execution environment for smart contracts.
- It works as a global computer that processes code in a secure, deterministic way

.3. Working Principle

- A smart contract written in Solidity is compiled to bytecode.
- The bytecode is deployed on the Ethereum network.

4. Importance

- Ensures interoperability between all Ethereum nodes.
- Makes smart contracts platform-independent.

Procedure:

- Open Remix IDE in a web browser at https://remix.ethereum.org.
- Create a new Solidity file and name it EVM_Test.sol.
- Select the Solidity compiler version (e.g., 0.8.0 or above).
- Write the following simple smart contract to test EVM functionality:

```
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.0;
3
4  contract EVM_Test {
5      uint public value;
6
7      function setValue(uint _v) public {    22514 gas
8          value = _v;
9      }
10 }
11
```

- Click on the “Compile” button to compile the code and check for errors.
- Open the Deploy & Run Transactions tab.
- Under Environment, choose JavaScript VM (London) — it runs a simulated Ethereum Virtual Machine directly in your browser.
- Click Deploy to deploy the contract.
- After successful deployment, the contract instance will appear in the “Deployed Contracts” section.
- Enter any value (e.g., 10) in the input box next to the setValue function and click the button.
- Observe the output in the Remix console — the transaction is executed, and the EVM updates the stored value.
- Click on the value button to verify that the value has been set successfully.

Observation:

- Ethereum has multiple client implementations (e.g., Geth, Besu, Nethermind).
- Each client follows the same consensus and EVM rules.
- The EVM (Ethereum Virtual Machine)executes smart contract bytecode securely and deterministically.
- Smart contracts run identically on all clients due to EVM standardization.

.

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.