



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 Experiment :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Steps (Implementation):

1. Wrote and deployed smart contract using Remix
2. users staked different amounts of ETH
3. Used getChance to view % chances for each validator
4. Simulated validator selection using JavaScript script

* Softwares used


1. Remix IDE
2. Solidity ^0.8.x
3. MetaMask
4. Remix VM

* Implementation Phase: Final Output (no error)

```


1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.0;
3  contract StakingPoS {
4      struct Validator {
5          address account;
6          uint256 stake;
7      }
8      Validator[] public validators;
9      uint256 public totalStake;
10     function stake() public payable {  infinite gas
11         require(msg.value > 0, "Stake must be greater than 0");
12         for (uint256 i = 0; i < validators.length; i++) {
13             if (validators[i].account == msg.sender) {
14                 validators[i].stake += msg.value;
15                 totalStake += msg.value;
16                 return;
17             }
18         }
19         validators.push(Validator(msg.sender, msg.value));
20         totalStake += msg.value;
21     }
22     function getChance(address _validator) public view returns(uint256) {  infinite gas
23         for (uint256 i = 0; i < validators.length; i++) {
24             if (validators[i].account == _validator) {
25                 return (validators[i].stake * 100) / totalStake;
26             }
27         }
28         return 0;
29     }
30 }


```

 **Account 1** ⓘ ⚙️


Deploy a contract

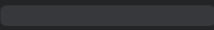
This site wants you to deploy a contract

Estimated changes ⓘ 


Network  **Sepolia**

Request from ⓘ

 **HTTP** localhost:49589




Network fee ⓘ 

\$2.62

Speed  **Market** ~12 sec

Max fee ⓘ 0.0007

Cancel **Confirm**



✓ STAKINGPOS AT 0X944...C2F2   

Balance: 0 ETH

stake

GETCHANCE ^


_validator: 0xe4ad95ddeb8a2c5cc1646171f

 **Calldata**  **Parameters** **call**

0: uint256: 0

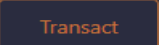
totalStake

0: uint256: 0

validators uint256 

Low level interactions ⓘ

CALLDATA



* Implementation Phase: Final Output (no error)

Applied and Action Learning

```
[block:9537734 txIndex:3] from: 0xe4a...ca52e to: StakingPoS.(constructor)
value: 0 wei data: 0x608...e0033 logs: 0 hash: 0x825...c62a7
view on Etherscan view on Blockscout
call to StakingPoS.getChance

CALL [call] from: 0xe4aD95DdeB8a2C5cC1646171850841668E8Ca52E
to: StakingPoS.getChance(address) data: 0x312...ca52e
call to StakingPoS.totalStake

CALL [call] from: 0xe4aD95DdeB8a2C5cC1646171850841668E8Ca52E
to: StakingPoS.totalStake() data: 0x8b0...e9f3f
```

0x9440Ec21ECf666659f8ba73dA0f8f0c9a6Ec2F2c

* Observations

- 1.Validators with a higher stake had a greater chance of being selected for block validation.
- 2.The total stake percentage directly influenced the probability of validator selection.
- 3.Even validators with low stake had a chance of being selected, but it was significantly reduced.
- 4.The simulation successfully showed how PoS promotes fairness and energy efficiency compared to Proof of Work.

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:

Page No.....

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