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 : DAO in Action – Governance Simulation

Objective/Aim:

To understand and simulate the working of a Decentralized Autonomous Organization (DAO) using smart contracts for governance and voting mechanisms on blockchain.

Apparatus/Software Used:

1.Chrome web browser

2.rif technology (https://rif.technology/content-hub/dao-governance-models//)

Theory:

A DAO (Decentralized Autonomous Organization) is a blockchain-based system that allows community members to govern projects through proposals and voting.
It removes centralized control and ensures transparency, as all rules and votes are recorded on-chain via smart contracts.

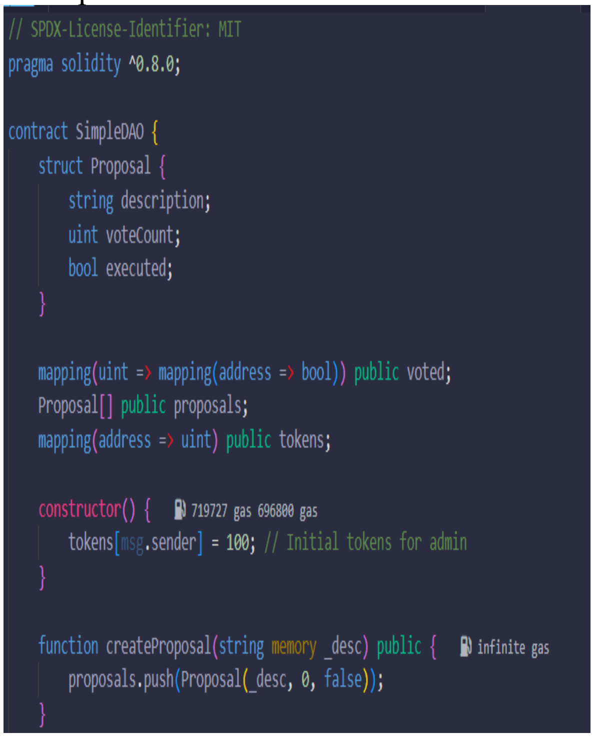
Key Components:

- Members: Token holders with voting rights.
- Proposals: Submitted ideas or actions requiring approval.
- Voting Mechanism: Members vote for or against proposals.
- Execution: Approved proposals are automatically executed by the DAO contract.

Procedure:

1. **Setup Environment:**
    - Open Remix IDE and connect MetaMask to a test network (e.g., Sepolia).
2. **Write Smart Contract:**
    - Create a Solidity contract for DAO governance.
    - Include functions for proposal creation, voting, and execution.
3. **Deploy Contract:**
    - Compile and deploy the contract on the testnet using Remix + MetaMask.
4. **Create Proposals:**
    - Members propose actions (e.g., funding, rule changes).
5. **Vote on Proposals:**
    - Token holders vote **for** or **against** proposals within a deadline.
6. **Execute Approved Proposals:**
    - If votes in favor exceed threshold, the proposal is executed automatically.
7. **Observe Results:**
    - Check the event logs for voting outcomes and execution status.

Example:



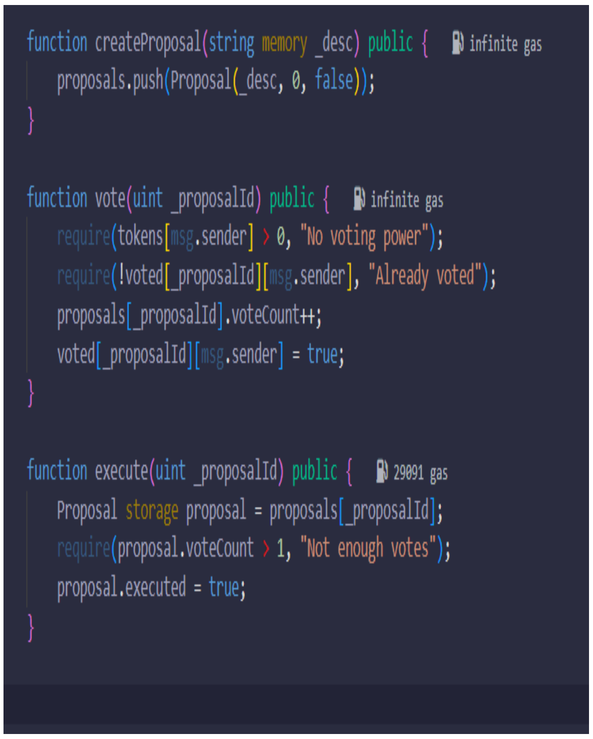
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleDAO {
    struct Proposal {
        string description;
        uint voteCount;
        bool executed;
    }

    mapping(uint => mapping(address => bool)) public voted;
    Proposal[] public proposals;
    mapping(address => uint) public tokens;

    constructor() {    719727 gas 696800 gas
        tokens[msg.sender] = 100; // Initial tokens for admin
    }

    function createProposal(string memory _desc) public {    infinite gas
        proposals.push(Proposal(_desc, 0, false));
    }
```



```
function createProposal(string memory _desc) public {    infinite gas
    proposals.push(Proposal(_desc, 0, false));
}

function vote(uint _proposalId) public {    infinite gas
    require(tokens[msg.sender] > 0, "No voting power");
    require(!voted[_proposalId][msg.sender], "Already voted");
    proposals[_proposalId].voteCount++;
    voted[_proposalId][msg.sender] = true;
}

function execute(uint _proposalId) public {    29091 gas
    Proposal storage proposal = proposals[_proposalId];
    require(proposal.voteCount > 1, "Not enough votes");
    proposal.executed = true;
}
```

Observation:

1. Members were able to submit and view governance proposals.
2. Voting process was recorded transparently on the blockchain.
3. Execution occurred only for proposals with majority approval.
4. The contract ensured decentralized, rule-based decision making.

A DAO-based governance system was successfully simulated.
Participants could create proposals, vote, and execute decisions securely without a central authority, demonstrating decentralized governance principles.

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. :

Page No...........

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