Practical 6

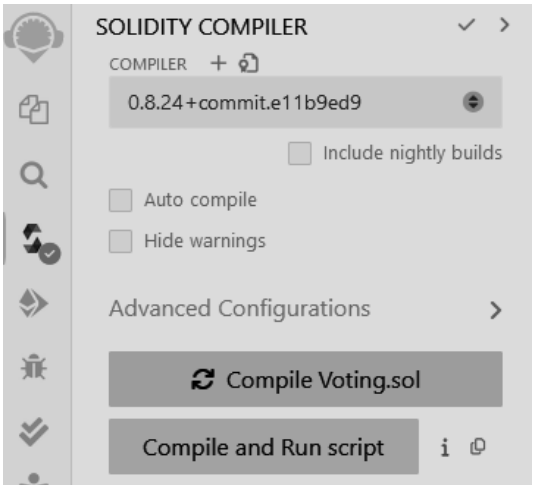
Aim: Develop a voting application using Hyperledger and Ethereum. Build a decentralised app that combines Ethereum's Web3 and Solidity smart contracts with Hyperledger hosting Fabric and Chaincode EVM.

Code:

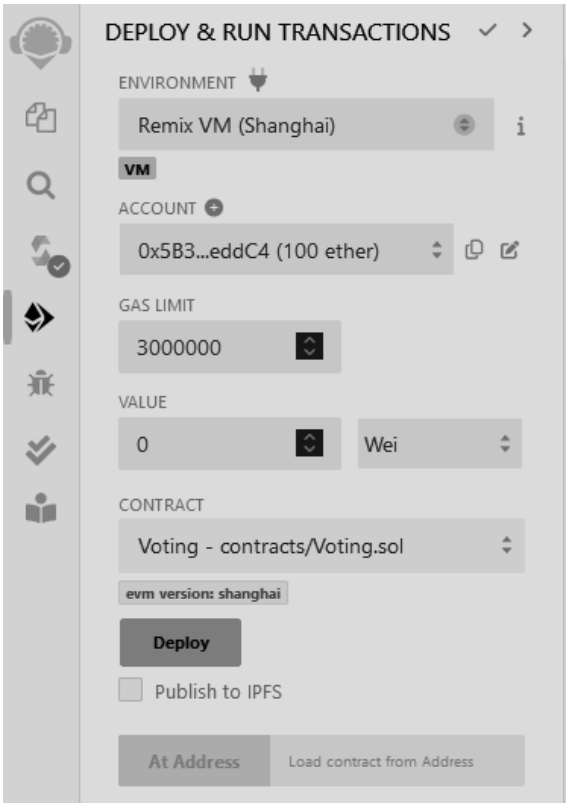
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Voting {
    // Declare variables to store the total votes for each team
    uint256 public votesForTeamA;
    uint256 public votesForTeamB;
    uint256 public votesForTeamC;
    address public owner;
    mapping(address => bool) authorizedVoters;

    constructor() {
        owner = msg.sender;
    }

    modifier onlyAuthorizedVoter() {
        require(
            authorizedVoters[msg.sender] == true,
            "You are not authorized to vote"
        );
        _;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can call this
function");
        _;
    }

    // Function to allow a voter to cast their vote for a team
    function vote(uint256 _team) public onlyAuthorizedVoter {
        // Check the value of _team and increment the corresponding team's
vote count
        if (_team == 1) {
            votesForTeamA += 1;
        } else if (_team == 2) {
            votesForTeamB += 1;
        } else if (_team == 3) {
            votesForTeamC += 1;
        }
    }
```

```
    // Function to add a voter to the list of authorized voters
    function addVoter(address _voter) public onlyOwner {
        require(msg.sender == owner, "Only the owner can add a voter");
        authorizedVoters[_voter] = true;
    }

    // Function to declare results of the vote
    function getWinner() public view returns (string memory result) {
        // Check the vote counts and return the winner
        if (votesForTeamA > votesForTeamB && votesForTeamA > votesForTeamC)
{
            result = "Team A is the winner!";
        } else if (
            votesForTeamB > votesForTeamA && votesForTeamB > votesForTeamC
        ) {
            result = "Team B is the winner!";
        } else if (
            votesForTeamC > votesForTeamA && votesForTeamC > votesForTeamB
        ) {
            result = "Team C is the winner!";
        } else {
            result = "There is a tie!";
        }
    }

    function getTotalVotes() public view returns (uint256) {
        return votesForTeamA + votesForTeamB + votesForTeamC;
    }
}
```

Compile The Contract



Deploy The Contract



Adding voters

(Note: you can get wallet address from above dropdown; change it to another and copy it)

Voting

Checking the winner