MINI PROJECT :

Decnetralized e-voting system using solidity:

CODE:

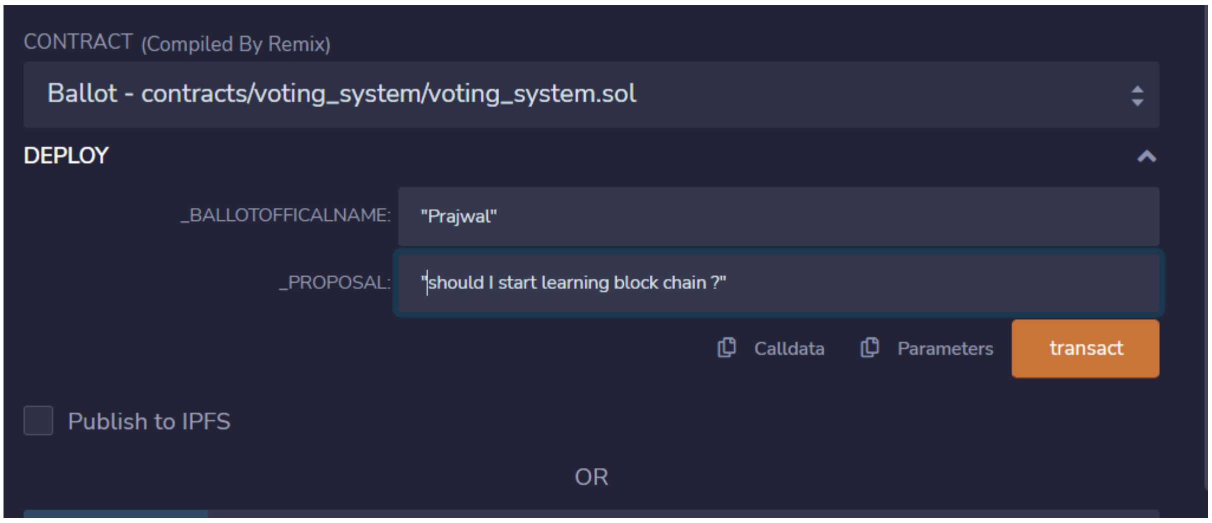
```
// SPDX-License-Identifier: MIT
pragma solidity >= 0.7.0 <0.8.0;

contract Ballot {
   // VARIBLES
   struct vote {
      address voterAddresss;
      bool choice;
   }
   struct voter {
      string voterName;
      bool voted;
   }
   uint private countResult = 0;
   uint public finalResult = 0;
   uint public totalVoter = 0;
   uint public totalVote = 0;

   address public ballotOfficialAddress;
   string public ballotOfficalName;
   string public proposal;

   mapping(uint => vote) private votes;
   mapping(address => voter) public voterRegister;

   enum State { Created, Voting, Ended }
   State public state;
```

```
// MODIFIER
modifier condition(bool _condition) {
   require(_condition);
   _;
}

modifier onlyOfficial() {
   require(msg.sender == ballotOfficialAddress);
   _;
}

modifier inState(State _state) {
   require(state == _state);
   _;
}


// FUNCTION
constructor(
   string memory _ballotofficalName,
   string memory _proposal
) {
   ballotOfficialAddress = msg.sender;
   ballotOfficalName = _ballotofficalName;
   proposal = _proposal;
   state = State.Created;
}
```

```
function addVoter(
    address _voterAdress,
    string memory _voterName
) public
    inState(State.Created)
    onlyOfficial
{
    voter memory v;
    v.voterName = _voterName;
    v.voted = false;
    voterRegister[_voterAdress] = v;
    totalVoter++;
}


function startVote()
    public
    inState(State.Created)
    onlyOfficial
{
    state = State.Voting;
}



function doVote(bool _choice)
    public
    inState(State.Voting)
    returns (bool voted)
{
    bool isFound = false;
```

```
        if(bytes(voterRegister[msg.sender].voterName).length != 0
            && voterRegister[msg.sender].voted == false )
        {
            voterRegister[msg.sender].voted = true;
            vote memory v;
            v.voterAddresss = msg.sender;
            v.choice = _choice;
            if(_choice) {
                countResult++;
            }
            votes[totalVote] = v;
            totalVote++;
            isFound = true;
        }
        return isFound;
    }
    function endVote()
        public
        inState(State.Voting)
        onlyOfficial
    {
        state = State.Ended;
        finalResult = countResult;
    }

}
```

1. **Deploy contract for election**



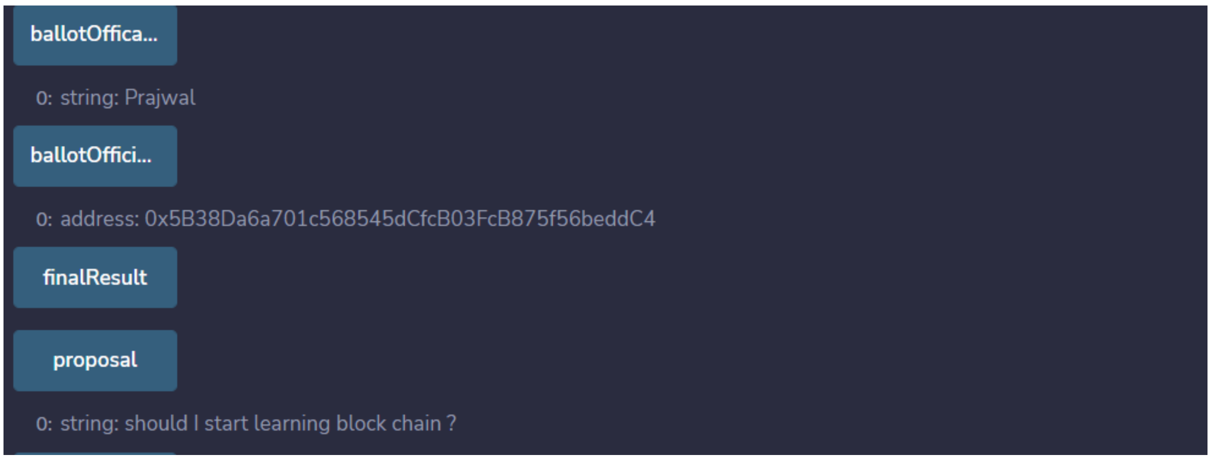
CONTRACT (Compiled By Remix)

Ballot - contracts/voting_system/voting_system.sol

DEPLOY

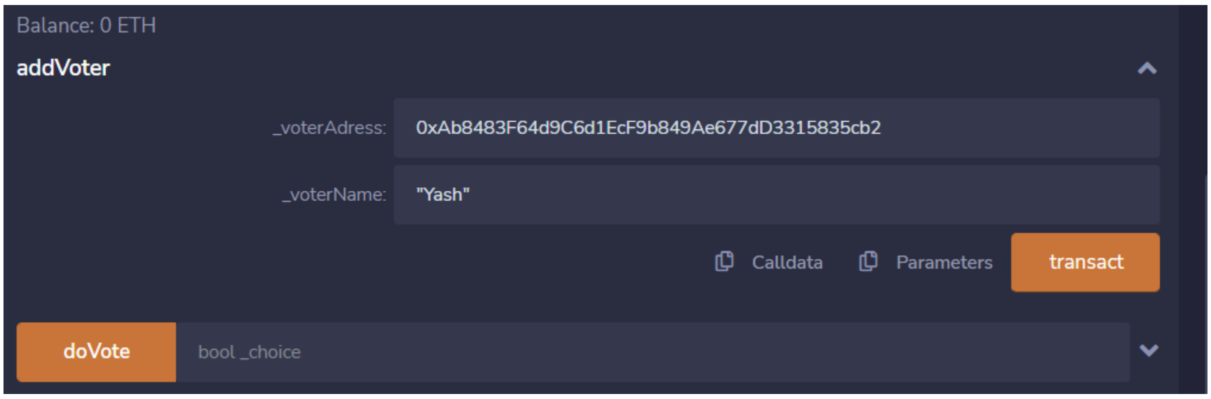
_BALLOTOFFICALNAME: "Prajwal"

_PROPOSAL: "should I start learning block chain ?"



```
ballotOffica...
0: string: Prajwal
ballotOffici...
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
finalResult
proposal
0: string: should I start learning block chain ?
```

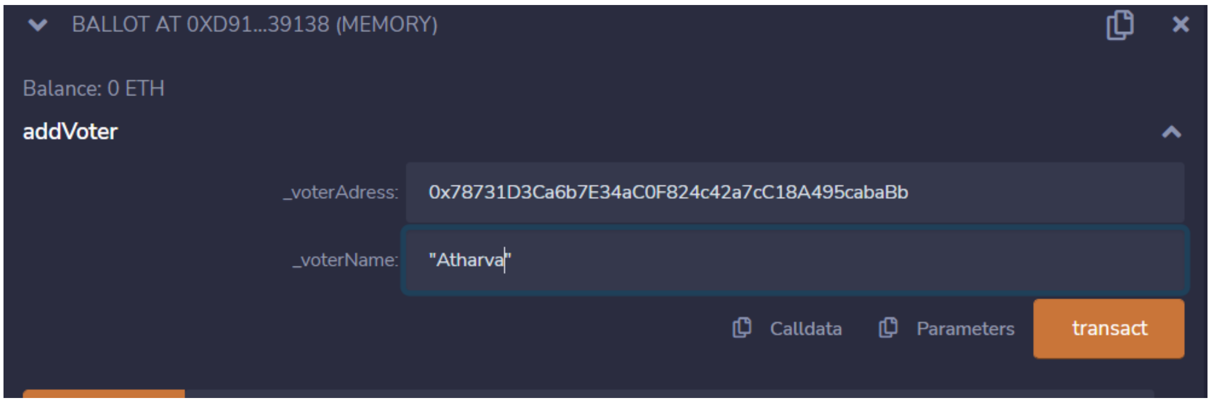
2. Add voters


Balance: 0 ETH

addVoter

_voterAdress: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

_voterName: "Yash"

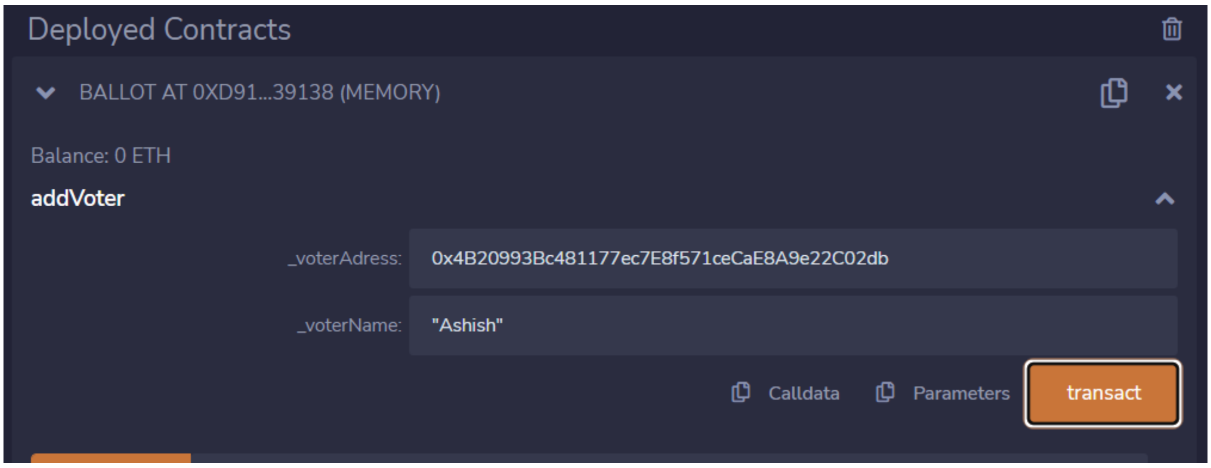

BALLOT AT 0XD91...39138 (MEMORY)

Balance: 0 ETH

addVoter

_voterAdress: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaBb

_voterName: "Atharva"


Deployed Contracts

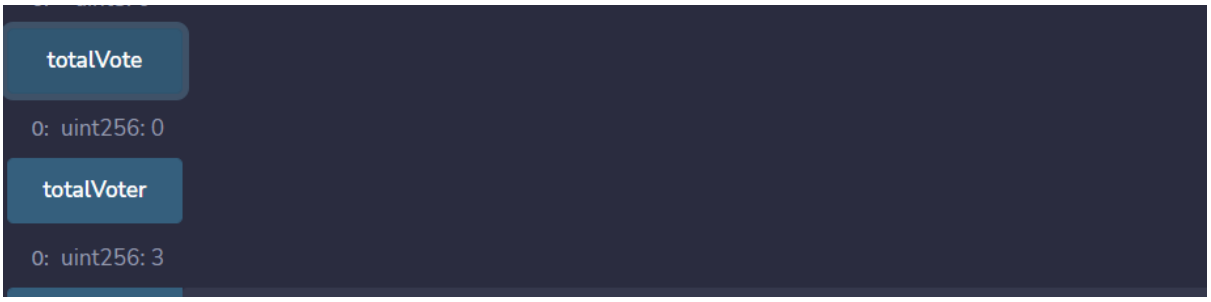
BALLOT AT 0XD91...39138 (MEMORY)

Balance: 0 ETH

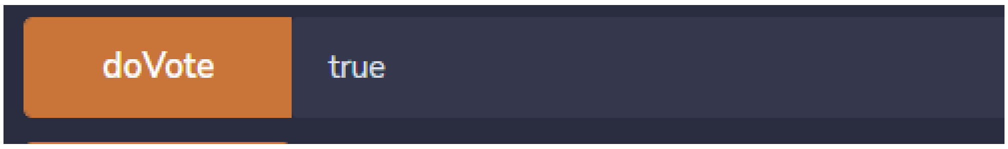
addVoter

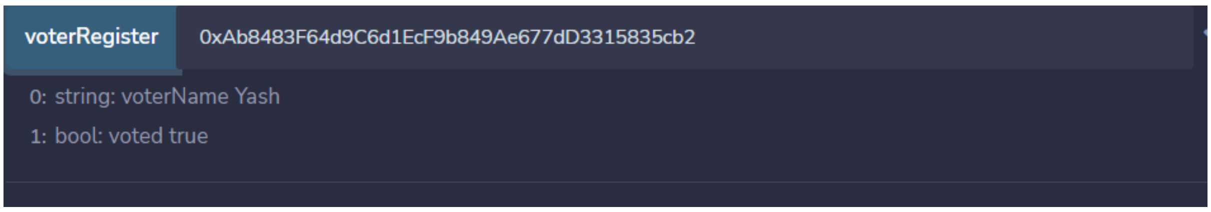
_voterAdress: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

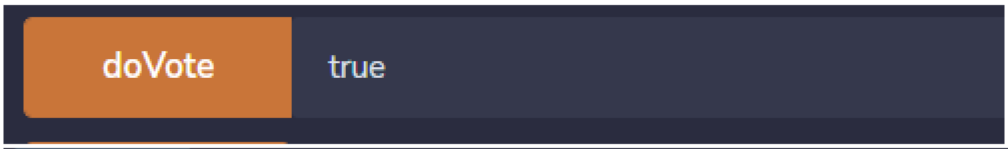
_voterName: "Ashish"

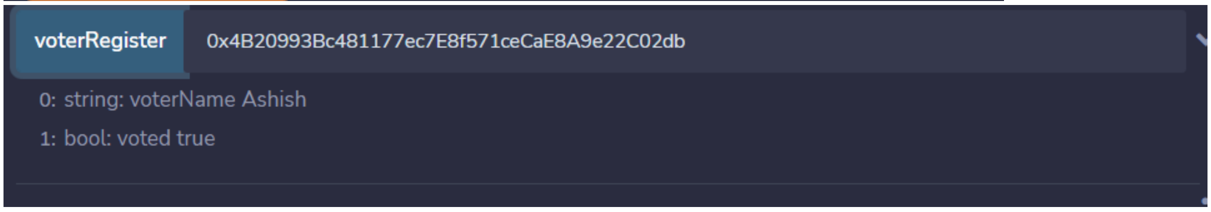

```
totalVote
0: uint256: 0
totalVoter
0: uint256: 3
```

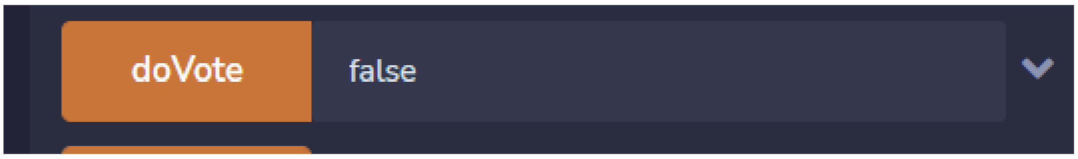
3. **Start election**

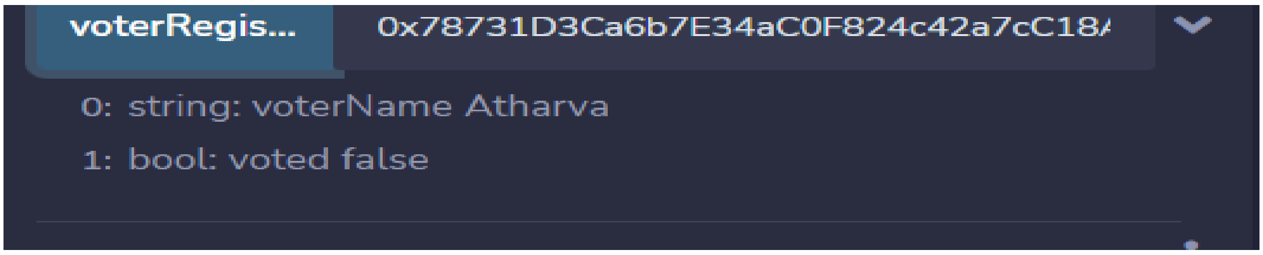

```
doVote    true
```


```
voterRegister    0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
0: string: voterName Yash
1: bool: voted true
```


```
doVote    true
```


```
voterRegister    0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
0: string: voterName Ashish
1: bool: voted true
```


```
doVote    false
```

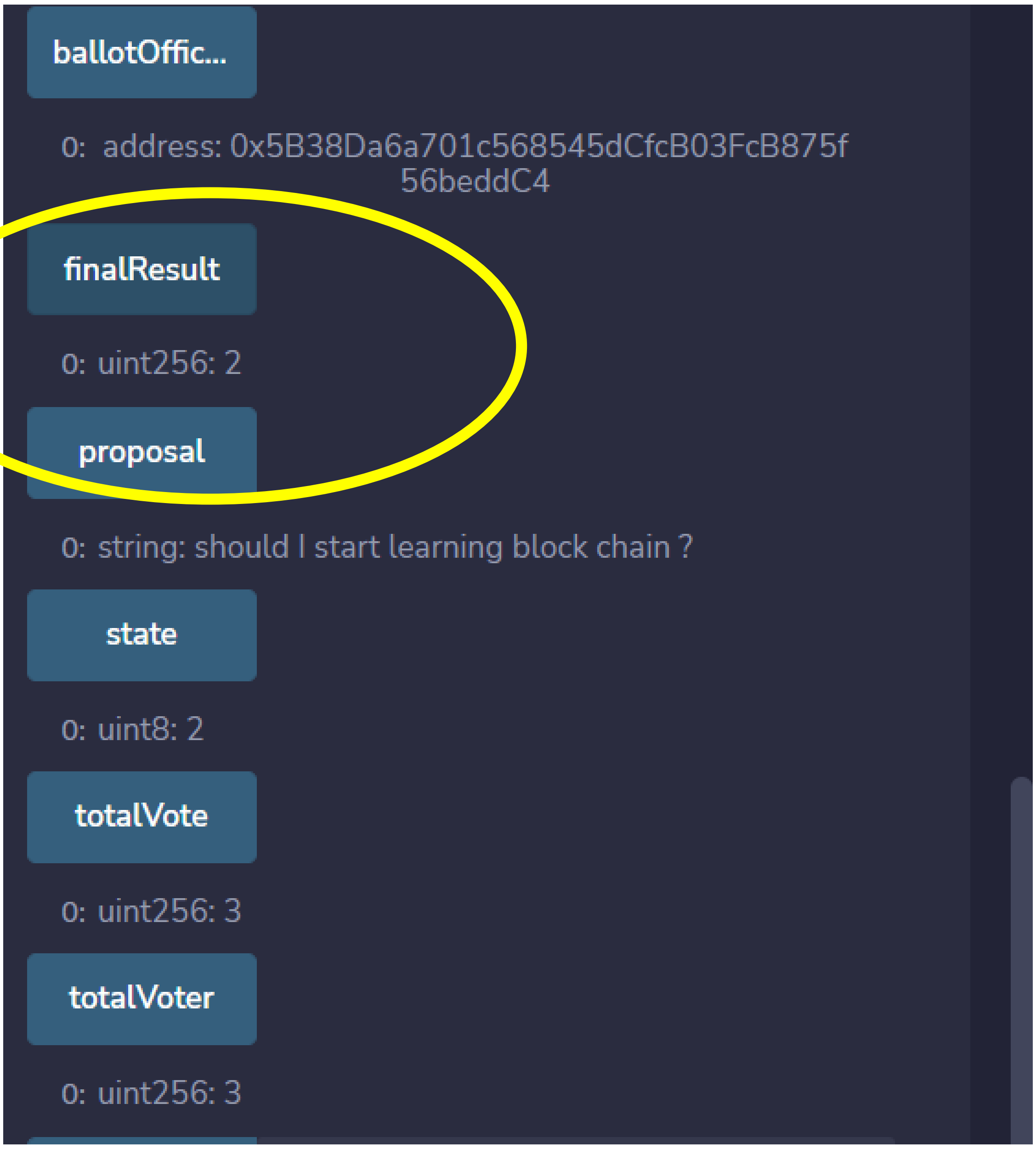

```
voterRegis...    0x78731D3Ca6b7E34aC0F824c42a7cC18
0: string: voterName Atharva
1: bool: voted false
```

4 . results

2 votes :yes

1 vote :no

2/3



```
0:  address: 0x5B38Da6a701c568545dCfcB03FcB875f
                        56beddC4
0: uint256: 2
0: string: should I start learning block chain ?
0: uint8: 2
0: uint256: 3
0: uint256: 3
```