2CSDE93 - Blockchain Technology

Practical 6

Aim: Voting System Smart Contract

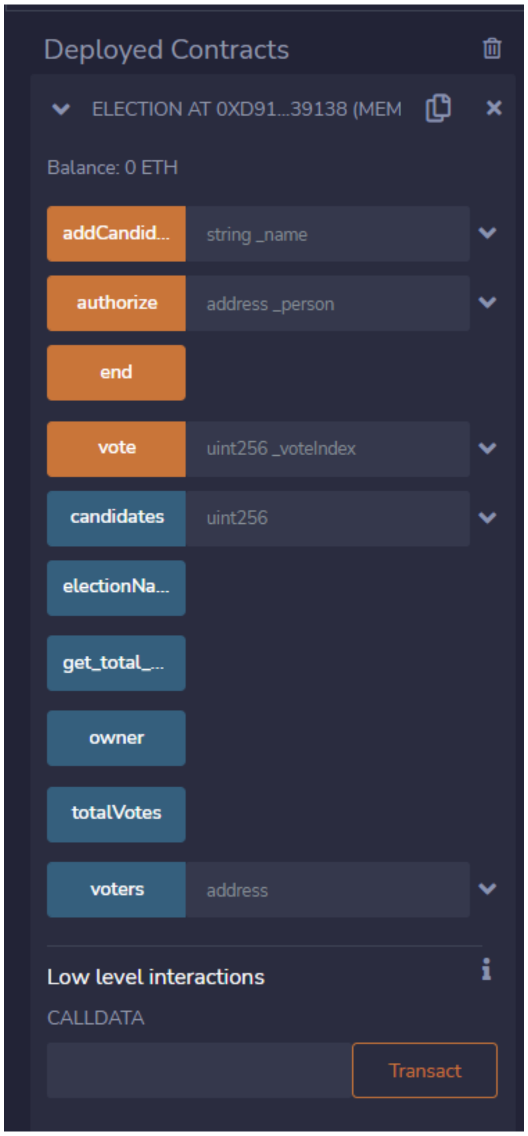
Author: Darshil Maru 20BCE514

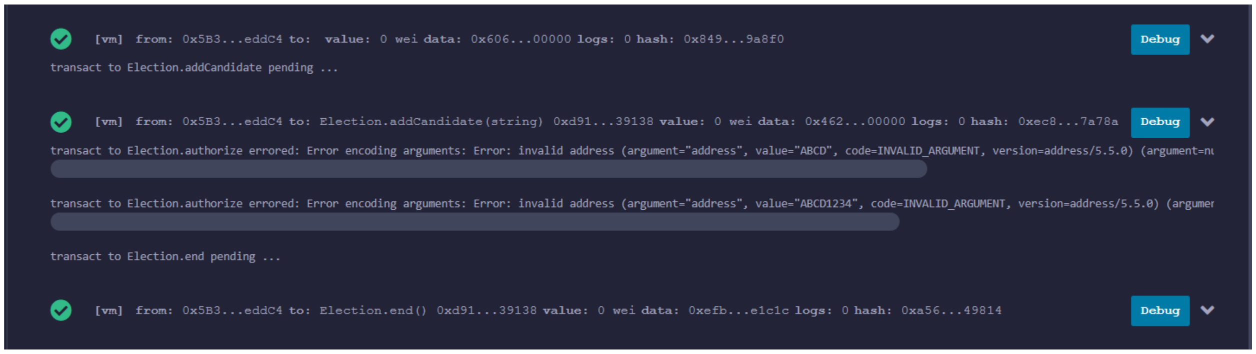
Code:

```
pragma solidity ^0.4.21;


contract Election{
    struct Candidate{
        string name;
        uint voteCount;
    }
    struct Voter{
        bool authorized; // is the voter authorized to vote or not
        bool voted; // voted or not
        uint vote;
    }
    address public owner; // owner of this contract , whoever deploys this
contract
    string public electionName; // election name , like presidential
election etc
// owner , electionName , boters are public thus state variables
    mapping(address => Voter) public voters;

    Candidate[] public candidates; // array of type Candidate
    uint public totalVotes;

    modifier ownerOnly(){
        require(msg.sender == owner); // its kind of like assert , use for
checking the parameters values
        _;
    }
// its convenction in solidity to mark the function paramenters with
underscore

    function Election(string _name) public{
        owner = msg.sender; // msg is a global variable , which comes
default in solidity ,
// it contains the address of the user account whoever deployed the
contract
```

```
        electionName = _name;
    }
    function addCandidate(string _name) ownerOnly public{
        candidates.push(Candidate(_name,0));
    }

    function get_total_candidates() public view returns(uint){
        return candidates.length;
    }
    function authorize(address _person) ownerOnly public{
        voters[_person].authorized = true;
    }
    function vote(uint _voteIndex) public{
        require(!voters[msg.sender].voted);
        require(voters[msg.sender].authorized);
        voters[msg.sender].vote = _voteIndex;
        voters[msg.sender].voted = true;
        candidates[_voteIndex].voteCount += 1;
        totalVotes += 1;
    }
    function end() ownerOnly public{
        selfdestruct(owner);
// end of the contract , so it destroys the contract so that no one
further can call any function or change
// the state, if the contract has any etheriums inside , it will be sended
to the owner
    }
}
```

Deployed Contract:





```
[vm] from: 0x5B3...eddC4 to:  value: 0 wei data: 0x606...00000 logs: 0 hash: 0x849...9a8f0
transact to Election.addCandidate pending ...

[vm] from: 0x5B3...eddC4 to: Election.addCandidate(string) 0xd91...39138 value: 0 wei data: 0x462...00000 logs: 0 hash: 0xec8...7a78a
transact to Election.authorize errored: Error encoding arguments: Error: invalid address (argument="address", value="ABCD", code=INVALID_ARGUMENT, version=address/5.5.0) (argument=nu

transact to Election.authorize errored: Error encoding arguments: Error: invalid address (argument="address", value="ABCD1234", code=INVALID_ARGUMENT, version=address/5.5.0) (argumer

transact to Election.end pending ...

[vm] from: 0x5B3...eddC4 to: Election.end() 0xd91...39138 value: 0 wei data: 0xefb...e1c1c logs: 0 hash: 0xa56...49814
```