

Aayush Shah

19BCE245

18 September 2022

# Blockchain Technology

## Practical 6

### Implementing voting system

#### • Code :

```
pragma solidity ^0.4.0; // Solidity with versions 4 and above

contract Prac6 {
    struct Candidate {
        /*Candidate structure with attributes name and number
of votes for the candidate. */
        string name;
        uint voteCount;
    }

    struct Voter {
        /*
Voter structure with attributes:
authorized – to check the authority of the voter,
voted – to check if the voter has already voted or
not,
vote – to get the index of the candidate the voter
voted.
*/
        bool authorized;
        bool voted;
        uint vote;
    }

    address public owner; // Owner
    string public electionName; // Name of the election
    mapping(address => Voter) public voters; // Maintain a
list of voters
    Candidate[] public candidates; // Maintain a list of
candidates
    uint public totalVotes; // Total number of votes
```

```

modifier ownerOnly () {
    /*A new modifier to check the authenticity */
    require(msg.sender == owner);
    _;
}

constructor() public {
    /*To give the editing rights to the author */
    owner=msg.sender;
}

function Election(string _name) public {
    /*To set the name of the election */
    owner = msg.sender ;
    electionName = _name;
}

function addCandidate (string _name) ownerOnly public {
    /*To add a candidate to the list */
    candidates.push (Candidate(_name,0));
}

function getNumCandidate () public view returns(uint) {
    /*To print the number of candidates */
    return candidates.length ;
}

function authorize(address _person) ownerOnly public {
    /*Authenticity of the user */
    voters[_person].authorized = true;
}

function vote(uint voteIndex) public {
    /*To vote a candidate */
    require(!voters[msg.sender].voted); // Voter should
have not voted
    require(voters[msg.sender].authorized); // Voter
should be authenticated
    voters[msg.sender].vote = voteIndex ; // Set the index
of the voter
    voters[msg.sender].voted = true; // Set the voted
boolean to true
    candidates[voteIndex].voteCount += 1; // Increment the
total vote count of the candidate
    totalVotes += 1; // Increment the total vote count
}

function end() ownerOnly public {

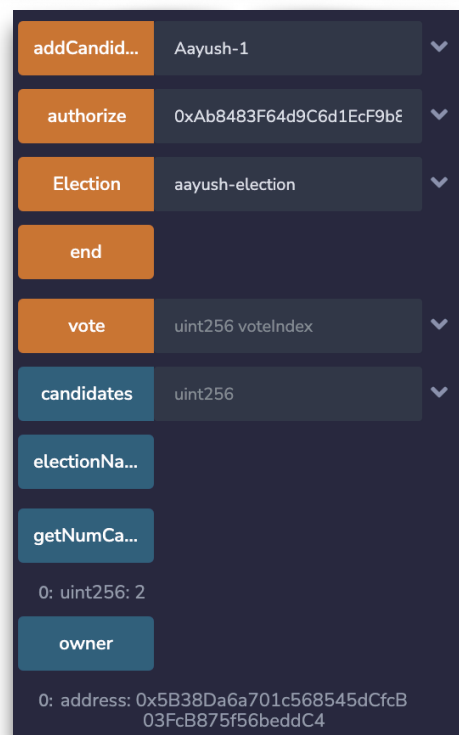
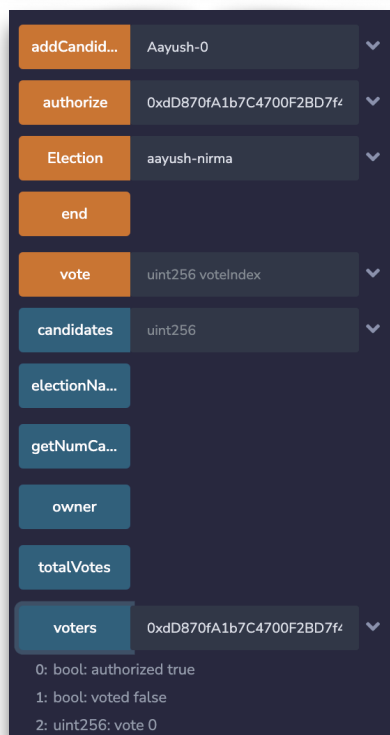
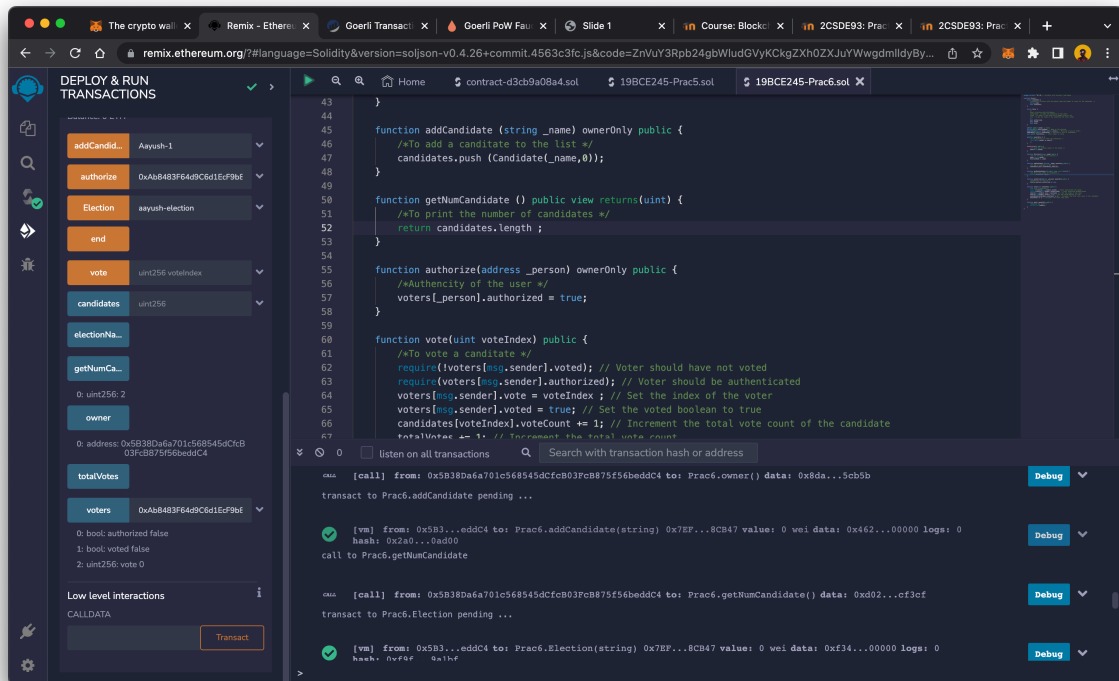
```

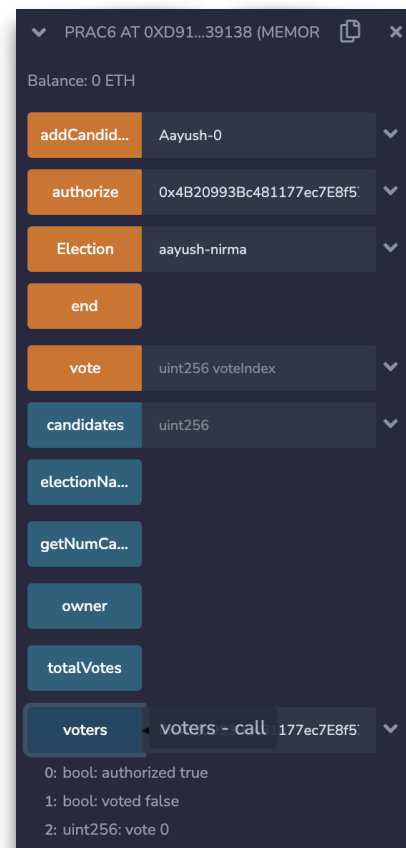
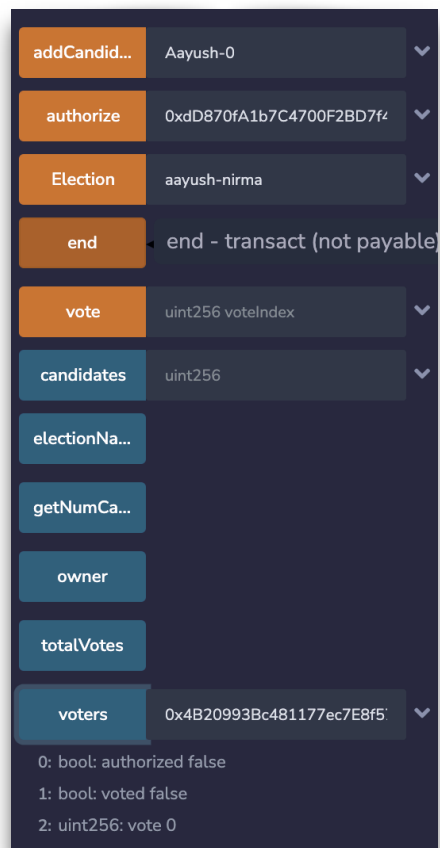
```

    }
    /*Destroy the owner */
    selfdestruct(owner);
}
}

```

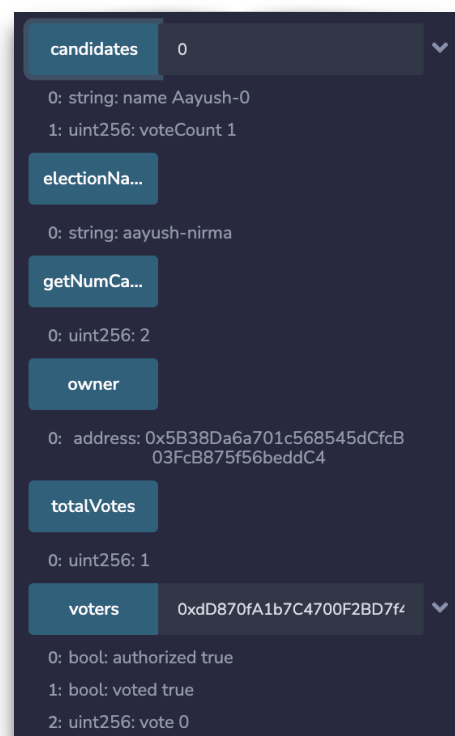
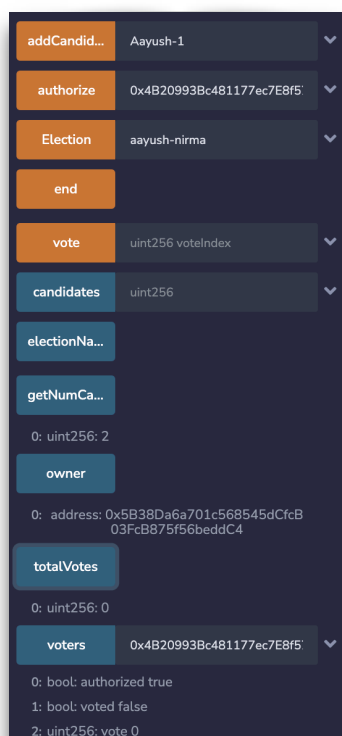
### • Output Screenshots:





Two candidates Aayush-0 and Aayush-1 has been created. Along with two voters.

Authorizing voters and giving votes from specific voter to candidate 'Aayush-0' who has index 0. After that we can see that authorized field of that specific voter is set to true and he also has field voter as true.



**Conclusion**

In this practical, I implemented voting system in which user can define candidates and voters as per own choice. According to him, any specific voter can give votes after having authorized access from owner. Finally, one can see the count of total votes to any candidates.