PRACTICAL 3

AIM:

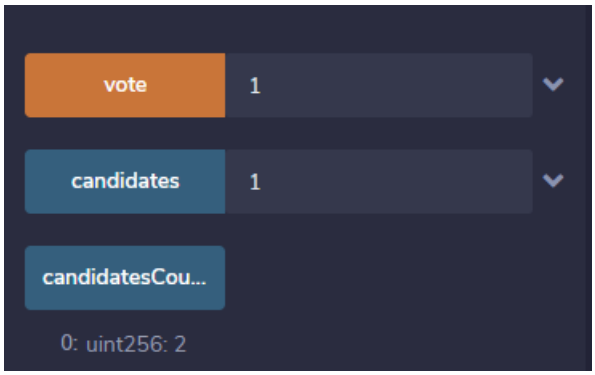
Write a smart contract of Election with following tasks:
a. Cast vote against two candidates
b. Vote should be cast only once.
c. No voter can vote any third candidate.

CODE:

```
pragma solidity 0.4.23;

contract Election {

    struct Candidate{
        uint id;
        string name;
        uint votecount;
    }

mapping(address => bool) public voters;
mapping(uint => Candidate) public candidates;

uint public candidatesCount;

    event votedEvent (
        uint indexed_candidateId
        );
        constructor () public
        {
            addCandidates("BJP");
            addCandidates("Congress");
        }
        function addCandidates(string _name) private {
            candidatesCount++;
            candidates[candidatesCount] = Candidate(candidatesCount, _name , 0);
        }
         function vote(uint _candidateId) public {
            require(!voters[msg.sender]);
            require(_candidateId> 0 && _candidateId<= candidatesCount);

            voters[msg.sender] = true;

            candidates[_candidateId].votecount++;

            emit votedEvent(_candidateId);
```

```
    }
}
```

OUTPUT:



CONCLUSION:

In this practical, We learned about Smart Contract of Election.