

ELECTRONIC VOTING SYSTEM

SOLIDITY CODE :

INPUT:

```
// SPDX-License-Identifier: MIT
```

```
pragma solidity ^0.8.0;
```

```
contract VoteSystem{
```

```
    address public owner;
```

```
    constructor(){
```

```
        owner= msg.sender;
```

```
    }
```

```
struct candidate {  
    uint voterId;  
    string name;  
    uint age;  
    uint voteCount;  
}
```

```
mapping (uint => candidate)  
candidateMap;
```

```
struct voters {  
    uint voterId;  
    string name;  
    uint age;  
    bool votingState;  
}
```

```
mapping (uint => voters) votersMap;  
mapping (uint=>bool) registeredVoter;
```

```
modifier checkVoterVoted(uint  
_votersVoterId){  
    require  
(votersMap[_votersVoterId].votingState  
== false);  
    _;  
}
```

```
modifier checkRegisteredVoter(uint  
_votersVoterId){  
  
require(registeredVoter[_votersVoterId]==  
true, "Voter is not Registered");  
    _;  
}
```

```
uint[] voterIdlist;
```

```
uint[] candidateIdList;
```

```
function enrollCandidate(uint  
_voterId,string memory _name,uint _age )  
public {
```

```
    require (_age >= 25);
```

```
    require (candidateMap[_voterId].voterId  
!= _voterId);
```

```
        candidateMap[_voterId].voterId =  
_voterId;
```

```
        candidateMap[_voterId].name = _name;
```

```
        candidateMap[_voterId].age = _age;
```

```
        candidateIdList.push(_voterId);
```

```
}
```

```
function enrollVoter(uint _voterId,string  
memory _name,uint _age) public  
returns(bool){
```

```
    require (_age >= 18);
```

```
    require (votersMap[_voterId].voterId !=  
_voterId);
```

```
        votersMap[_voterId].voterId =  
_voterId;
```

```
        votersMap[_voterId].name = _name;
```

```
        votersMap[_voterId].age = _age;
```

```
        voterIdlist.push(_voterId);
```

```
        return registeredVoter[_voterId]=true;
```

```
}
```

```
function getCandidateDetails(uint  
_voterId) view public returns(uint,string  
memory,uint,uint) {
```

```
    return  
(candidateMap[_voterId].voterId,candidate  
Map[_voterId].name,candidateMap[_voter  
Id].age,candidateMap[_voterId].voteCount  
);  
}
```

```
function getVoterDetails(uint _voterId)  
view public returns (uint,string  
memory,uint,bool){
```

```
    return  
(votersMap[_voterId].voterId,votersMap[_
```

```
voterId].name,votersMap[_voterId].age,votersMap[_voterId].votingState);
```

```
}
```

```
function vote(uint _candidateVoterId,uint  
_votersVoterId) public  
checkVoterVoted(_votersVoterId)  
checkRegisteredVoter(_votersVoterId) {
```

```
candidateMap[_candidateVoterId].voteCount  
+= 1;
```

```
    votersMap[_votersVoterId].votingState  
= true;  
}
```

```
function getVoteCountOf(uint _voterId)  
view public returns(uint){
```

```
    require(msg.sender== owner, "Only  
owner is allowed to Check Results");
```

```
    return  
    candidateMap[_voterId].voteCount;  
}
```

```
function getVoterList() view public  
returns (uint[] memory){
```

```
    return  voterIdlist;  
}
```

```
function getCandidateList() view public  
returns(uint[] memory){
```

```
    return candidateIdList;  
}
```



```
}
```

ABI CODE:

```
[
```

```
{
```

```
    "inputs": [],
```

```
    "stateMutability": "nonpayable",
```

```
    "type": "constructor"
```

```
},
```

```
{
```

```
    "inputs": [
```

```
        {
```

```
            "internalType": "uint256",
```

```
            "name": "_voterId",
```

```
            "type": "uint256"
```

```
    },  
    {  
        "internalType": "string",  
        "name": "_name",  
        "type": "string"  
    },  
    {  
        "internalType": "uint256",  
        "name": "_age",  
        "type": "uint256"  
    }  
],  
"name": "enrollCandidate",  
"outputs": [],  
"stateMutability": "nonpayable",  
"type": "function"  
},
```

```
{  
  "inputs": [  
    {  
      "internalType": "uint256",  
      "name": "_voterId",  
      "type": "uint256"  
    },  
    {  
      "internalType": "string",  
      "name": "_name",  
      "type": "string"  
    },  
    {  
      "internalType": "uint256",  
      "name": "_age",  
      "type": "uint256"  
    }  
  ]  
}
```

```
],  
  "name": "enrollVoter",  
  "outputs": [  
    {  
      "internalType": "bool",  
      "name": "",  
      "type": "bool"  
    }  
  ],  
  "stateMutability": "nonpayable",  
  "type": "function"  
},  
{  
  "inputs": [  
    {  
      "internalType": "uint256",  
      "name": "_voterId",
```

```
        "type": "uint256"
    }
],
"name": "getCandidateDetails",
"outputs": [
    {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
    },
    {
        "internalType": "string",
        "name": "",
        "type": "string"
    },
    {
        "internalType": "uint256",
```

```
        "name": "",
        "type": "uint256"
    },
    {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
    }
],
"stateMutability": "view",
"type": "function"
},
{
    "inputs": [],
    "name": "getCandidateList",
    "outputs": [
        {
```

```
        "internalType": "uint256[]",
        "name": "",
        "type": "uint256[]"
    }
],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "_voterId",
            "type": "uint256"
        }
    ],
    "name": "getVoteCountOf",
```

```
"outputs": [  
  {  
    "internalType": "uint256",  
    "name": "",  
    "type": "uint256"  
  }  
,  
  "stateMutability": "view",  
  "type": "function"  
},  
{  
  "inputs": [  
    {  
      "internalType": "uint256",  
      "name": "_voterId",  
      "type": "uint256"  
    }  
  ]  
}
```



```
],  
  "name": "getVoterDetails",  
  "outputs": [  
    {  
      "internalType": "uint256",  
      "name": "",  
      "type": "uint256"  
    },  
    {  
      "internalType": "string",  
      "name": "",  
      "type": "string"  
    },  
    {  
      "internalType": "uint256",  
      "name": "",  
      "type": "uint256"
```

```
    },  
    {  
        "internalType": "bool",  
        "name": "",  
        "type": "bool"  
    }  
],  
    "stateMutability": "view",  
    "type": "function"  
},  
{  
    "inputs": [],  
    "name": "getVoterList",  
    "outputs": [  
        {  
            "internalType": "uint256[]",  
            "name": "",
```

```
        "type": "uint256[]"
    }
],
"stateMutability": "view",
"type": "function"
},
{
    "inputs": [],
    "name": "owner",
    "outputs": [
        {
            "internalType": "address",
            "name": "",
            "type": "address"
        }
    ],
    "stateMutability": "view",
```

```
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name":
"_candidateVoterId",
        "type": "uint256"
      },
      {
        "internalType": "uint256",
        "name": "_votersVoterId",
        "type": "uint256"
      }
    ],
    "name": "vote",
```

```
    "outputs": [],  
    "stateMutability": "nonpayable",  
    "type": "function"  
  }  
]
```

BYTE CODE :

```
608060405234801561001057600080fd5b5  
0336000806101000a81548173ffffffffffff  
ffffffffffffffffffffffff021916908373ffff  
ffffffffffffffffffffffffffffffff1602179055506  
111ae806100606000396000f3fe60806040  
5234801561001057600080fd5b506004361  
06100935760003560e01c80638da5cb5b11  
6100665780638da5cb5b1461015e578063b  
28d92ae1461017c578063b384abef146101  
9a578063b9041443146101b6578063fdb4  
006146101d257610093565b806327f9fd50  
146100985780634ecb515d146100cb57806  
37b8fe166146100fb57806387d858fc1461  
012e575b600080fd5b6100b260048036038
```

101906100ad9190610864565b6101f0565b
6040516100c29493929190610930565b60
405180910390f35b6100e56004803603810
1906100e09190610864565b6102ed565b6
040516100f2919061097c565b6040518091
0390f35b610115600480360381019061011
09190610864565b61039c565b604051610
12594939291906109b2565b60405180910
390f35b6101486004803603810190610143
9190610b33565b6104a6565b6040516101
559190610ba2565b60405180910390f35b6
1016661058f565b6040516101739190610b
fe565b60405180910390f35b6101846105b
3565b6040516101919190610cd7565b604
05180910390f35b6101b460048036038101
906101af9190610cf9565b61060b565b005
b6101d060048036038101906101cb91906
10b33565b61070a565b005b6101da6107c2
565b6040516101e79190610cd7565b6040
5180910390f35b600060606000806001600
0868152602001908152602001600020600

0015460016000878152602001908152602
0016000206001016001600088815260200
1908152602001600020600201546001600
0898152602001908152602001600020600
3015482805461025e90610d68565b80601f
0160208091040260200160405190810160
4052809291908181526020018280546102
8a90610d68565b80156102d75780601f106
102ac57610100808354040283529160200
1916102d7565b820191906000526020600
020905b815481529060010190602001808
3116102ba57829003601f168201915b5050
5050509250935093509350935091935091
93565b60008060009054906101000a9004
73ffffffffffffffffffffffffffffffffffff1673fff
ffffffffffffffffffffffffffffffffffff163373fffff
ffffffffffffffffffffffffffffffffffff161461037e57
6040517f08c379a0000000000000000000000000
00
000815260040161037590610e0b565b604
05180910390fd5b60016000838152602001

9081526020016000206003015490509190
50565b6000606060008060026000868152
6020019081526020016000206000015460
0260008781526020019081526020016000
2060010160026000888152602001908152
6020016000206002015460026000898152
6020019081526020016000206003016000
9054906101000a900460ff1682805461041
790610d68565b80601f0160208091040260
2001604051908101604052809291908181
5260200182805461044390610d68565b80
156104905780601f1061046557610100808
354040283529160200191610490565b820
191906000526020600020905b815481529
0600101906020018083116104735782900
3601f168201915b50505050509250935093
50935093509193509193565b6000601282
10156104b657600080fd5b8360026000868
1526020019081526020016000206000015
4036104d857600080fd5b83600260008681
5260200190815260200160002060000181

9055508260026000868152602001908152
6020016000206001019081610516919061
0fd7565b508160026000868152602001908
1526020016000206002018190555060048
4908060018154018082558091505060019
0039060005260206000200160009091909
1909150556001600360008681526020019
0815260200160002060006101000a81548
160ff021916908315150217905590509392
505050565b60008054906101000a900473f
ff1681565b
6060600480548060200260200160405190
8101604052809291908181526020018280
5480156106015760200282019190600052
6020600020905b81548152602001906001
01908083116105ed575b50505050509050
90565b8060001515600260008381526020
0190815260200160002060030160009054
906101000a900460ff16151514610640576
00080fd5b81600115156003600083815260
2001908152602001600020600090549061

[illegible]

5480602002602001604051908101604052
8092919081815260200182805480156108
1057602002820191906000526020600020
905b815481526020019060010190808311
6107fc575b5050505050905090565b60006
04051905090565b600080fd5b600080fd5b
6000819050919050565b6108418161082e
565b811461084c57600080fd5b50565b600
08135905061085e81610838565b9291505
0565b60006020828403121561087a57610
879610824565b5b6000610888848285016
1084f565b91505092915050565b61089a81
61082e565b82525050565b600081519050
919050565b600082825260208201905092
915050565b60005b838110156108da5780
820151818401526020810190506108bf565
b60008484015250505050565b6000601f19
601f8301169050919050565b60006109028
26108a0565b61090c81856108ab565b9350
61091c8185602086016108bc565b610925
816108e6565b840191505092915050565b

6000608082019050610945600083018761
0891565b81810360208301526109578186
6108f7565b9050610966604083018561089
1565b6109736060830184610891565b959
45050505050565b6000602082019050610
9916000830184610891565b92915050565
b60008115159050919050565b6109ac816
10997565b82525050565b6000608082019
0506109c76000830187610891565b81810
360208301526109d981866108f7565b9050
6109e86040830185610891565b6109f5606
08301846109a3565b95945050505050565
b600080fd5b600080fd5b7f4e487b710000
0000000000000000000000000000000000
0000000000000000000006000526041600452
60246000fd5b610a40826108e6565b81018
1811067fffffffffffffffff82111715610a5f576
10a5e610a08565b5b80604052505050565b
6000610a7261081a565b9050610a7e82826
10a37565b919050565b600067fffffffffffff
ff821115610a9e57610a9d610a08565b5b6

10aa7826108e6565b905060208101905091
9050565b82818337600083830152505050
565b6000610ad6610ad184610a83565b610
a68565b905082815260208101848484011
115610af257610af1610a03565b5b610afd8
48285610ab4565b509392505050565b600
082601f830112610b1a57610b196109fe56
5b5b8135610b2a848260208601610ac3565
b91505092915050565b600080600060608
486031215610b4c57610b4b610824565b5
b6000610b5a8682870161084f565b935050
602084013567fffffffffffffffff811115610b7b
57610b7a610829565b5b610b8786828701
610b05565b9250506040610b9886828701
61084f565b9150509250925092565b60006
02082019050610bb760008301846109a35
65b92915050565b600073fffffffffffffffff
fffffffffffffffffffffffff82169050919050565b600
0610be882610bbd565b9050919050565b6
10bf881610bdd565b82525050565b600060
2082019050610c136000830184610bef565

b92915050565b600081519050919050565
b600082825260208201905092915050565
b6000819050602082019050919050565b6
10c4e8161082e565b82525050565b600061
0c608383610c45565b6020830190509291
5050565b6000602082019050919050565b
6000610c8482610c19565b610c8e8185610
c24565b9350610c9983610c35565b806000
5b83811015610cca578151610cb18882610
c54565b9750610cbc83610c6c565b925050
600181019050610c9d565b508593505050
5092915050565b60006020820190508181
036000830152610cf18184610c79565b905
092915050565b600080604083850312156
10d1057610d0f610824565b5b6000610d1e
8582860161084f565b9250506020610d2f8
582860161084f565b915050925092905056
5b7f4e487b710000000000000000000000
0000000000000000000000000000000006
00052602260045260246000fd5b60006002
820490506001821680610d8057607f82169

1505b602082108103610d9357610d92610
d39565b5b50919050565b7f4f6e6c79206f
776e657220697320616c6c6f77656420746
f20436865636b205260008201527f657375
6c74730000000000000000000000000000
0000000000000000000000000000006020820152
50565b6000610df56026836108ab565b915
0610e0082610d99565b604082019050919
050565b600060208201905081810360008
30152610e2481610de8565b90509190505
65b6000819050816000526020600020905
0919050565b60006020601f830104905091
9050565b600082821b905092915050565b
600060088302610e8d7fffffffffffffffffff
ffffffffffffffffffffffffffffffffffff82610e
50565b610e978683610e50565b95508019
8416935080861684179250505093925050
50565b6000819050919050565b6000610e
d4610ecf610eca8461082e565b610eaf565b
61082e565b9050919050565b6000819050
919050565b610eee83610eb9565b610f026

10efa82610edb565b848454610e5d565b82
5550505050565b600090565b610f17610f0
a565b610f22818484610ee5565b50505056
5b5b81811015610f4657610f3b600082610
f0f565b600181019050610f28565b505056
5b601f821115610f8b57610f5c81610e2b5
65b610f6584610e40565b81016020851015
610f74578190505b610f88610f8085610e4
0565b830182610f27565b50505b50505056
5b600082821c905092915050565b600061
0fae60001984600802610f90565b1980831
691505092915050565b6000610fc7838361
0f9d565b915082600202821790509291505
0565b610fe0826108a0565b67fffffffffffff
f811115610ff957610ff8610a08565b5b611
0038254610d68565b61100e828285610f4a
565b600060209050601f831160018114611
041576000841561102f578287015190505b
6110398582610fbb565b8655506110a1565
b601f19841661104f86610e2b565b60005b
8281101561107757848901518255600182

0191506020850194506020810190506110
52565b8683101561109457848901516110
90601f891682610f9d565b8355505b60016
00288020188555050505b5050505050505
65b7f566f746572206973206e6f74205265
6769737465726564000000000000000000
600082015250565b60006110df601783610
8ab565b91506110ea826110a9565b602082
019050919050565b600060208201905081
8103600083015261110e816110d2565b90
50919050565b7f4e487b7100000000000000
0000000000000000000000000000000000
0000000000600052601160045260246000fd
5b600061114f8261082e565b915061115a8
361082e565b92508282019050808211156
1117257611171611115565b5b929150505
6fea26469706673582212205d4ee2f6fd85f
60fd950af88a59b3cd65b66f8e58830a1ab8
a9cc825cf5b54b564736f6c634300081200
33

OUTPUT:

The screenshot displays the Remix IDE interface during the deployment of a Solidity contract named 'VoteSystem'. The left sidebar contains the 'DEPLOY & RUN TRANSACTIONS' panel, which shows the environment set to 'Remix VM (Shanghai)', the account '0x5B3...eddC4 (99.999999%)', a gas limit of '3000000', and a value of '0 Wei'. The contract name 'VoteSystem - Electronic voting system' is entered, and the 'Deploy' button is highlighted. Below the deployment options, it indicates 'Transactions recorded 1' and 'Deployed Contracts'.

The main editor displays the Solidity code for the 'VoteSystem' contract, which includes functions for voting, checking results, and retrieving voter and candidate lists. The code is as follows:

```
function vote(uint _candidateVoterId, uint _votersVoterId) public checkVoterVoted(_votersVoterId) checkCandidateVoted(_candidateVoterId) {
    candidateMap[_candidateVoterId].voteCount += 1;
    votersMap[_votersVoterId].votingState = true;
}

function getVoteCountOf(uint _voterId) view public returns(uint) {
    require(msg.sender == owner, "Only owner is allowed to Check Results");
    return candidateMap[_voterId].voteCount;
}

function getVoterList() view public returns (uint[] memory) {
    return voterIdList;
}

function getCandidateList() view public returns(uint[] memory) {
    return candidateIdList;
}
```

The bottom status bar shows a successful deployment message: '[vm] from: 0x5B3...eddC4 to: VoteSystem.(constructor) value: 0 wei data: 0x00...20033 logs: 0 hash: 0x0b8...c5240'. A 'Debug' button is visible on the right side of the status bar.