**Voting application – Test cases & Screenshots**

**Professional Certificate Program in Blockchain by IIT Kanpur**

**Rajendra kumar,** [**rkumar084@gmail.com**](mailto:rkumar084@gmail.com)

**Introduction** This document provides sample test data consisting of all addresses taken from ganache blockchain, test cases and screenshots of test results

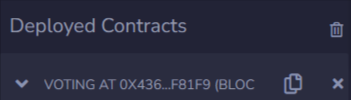
**Test Data**

|  |  |
| --- | --- |
| Voters | 0x8789b0f86349370BC36E4a7ab63e33c69B193E43 **(Admin)**  Name: Rajendra (Voted for Modi)    0x0e1E1c7eA6cf9a4990044cF8C2628997e5aa7a8f  Name: Rahul (Voted for Self)  0x3Efd90f99834Ed0D4F6962Aa45CA337A9342f8CB  Name: Modi (Voted for Self)  0x18DE60f6e93F77d55deE379a0826486d96c00C19  Name: Narayana (Voted for Rahul)  0xa56a786E789F809b11847911dFCe89731B65FFA6  Name: Hanumanth (Voted for Modi)  0x8d723F0B74295f27DfE2aa45438CF10c67c523E0  Name: Srinivas (Voted for Modi)  0x63b8845c8780183bFeac21aDf26a1Bee152A5E2b  Name: Chandhra (Voted for Modi)  0xf82b9BDaF8C250B73EF92d214Bec6580178Cd382  Name: Rajeshwer (Voted for Modi)    0x000654DD0f54c1F3E6f5EfAa719E26220B6b0bF1  Name: Manohar (Voted for Rahul via delegate Rajendra)  0x74C6E382508F4D3D1A8Eaa78E79A3Dc1f1AedC04  Name: Vinay (Voted for Modi via delegate Rajendra) |
| Candidates | 0x0e1E1c7eA6cf9a4990044cF8C2628997e5aa7a8f  Id: 1, Name: Rahul, Proposal: No clear agenda  0x3Efd90f99834Ed0D4F6962Aa45CA337A9342f8CB  Id: 2, Name: Modi, Proposal: Growth |

**Test Cases and Screenshots**

All the test cases have below things as common

* Admin address: 0x8789b0f86349370BC36E4a7ab63e33c69B193E43
* Smart contract methods are called from Remix IDE
* Contract Address: 0x436bea5D9CB3Aa1cA0Cdc8CDC210b8f08e4f81F9



|  |  |
| --- | --- |
| Testcases for checkState() | Expected result |
| msg.sender = any | Status of the election should be returned. IT should be NOTSTARTED just after deployment |

|  |  |
| --- | --- |
| Testcases for startElection() | Expected result |
| msg.sender = Non Admin | Error! Only admin can perform operation |
| msg.sender = Admin | Election status should be changed to ONGOING |

|  |  |
| --- | --- |
| Testcases for addCandidate( ) | Expected result |
| msg.sender = Non Admin | Error! Only admin can perform operation |
| msg.sender = Admin | Candidate details are added and candidateMapping is updated. Verified with displayCandidateDetails call |

|  |  |
| --- | --- |
| Testcases for displayCandidateDetails() | Expected result |
| with non-existing candidate  msg.sender = any | Error! Candidate does not exist |
| with an existing candidate  msg.sender = any | Candidate details are returned |

|  |  |
| --- | --- |
| Testcases for getCandidateCount( ) | Expected result |
| msg.sender = any | Get number of candidates contesting in election |

|  |  |
| --- | --- |
| Testcases for addVoter() | Expected result |
| msg.sender = Non Admin | Error! Only admin can perform operation |
| msg.sender = Admin | Voter details are added and the voter should be added to voterMapping. |

|  |  |
| --- | --- |
| Testcases for voterProfile() | Expected result |
| With a non-existing Voter  msg.sender = any | Error! Voter does not exist |
| with an existing voter  msg.sender = any | Voter details are returned |

|  |  |
| --- | --- |
| Testcases for getVoterCount() | Expected result |
| msg.sender = any | Number of voters should be returned |

|  |  |
| --- | --- |
| Testcases for delegateVote() | Expected result |
| msg.sender not same as voteraddress  msg.sender = voter | Unauthorized! Voter cannot delegate for other voters |
| with a non-existing voteraddress  msg.sender = voter | Error! Voter does not exist. |
| election status as not ONGOING  msg.sender = voter | Error! Election should be in progress for delegation. |
| with none of the above conditions  msg.sender = voter | Voter should be assigned with delegate address, and voter mapping should be updated. |

|  |  |
| --- | --- |
| Testcases vote() | Expected result |
| with non-existing voter address  msg.sender = any | Error if voter does not exist |
| when election status is not ONGOING  msg.sender = any | Error! Election should be in progress for voting |
| Duplicate call to Vote  msg.sender = any | Error! Voter has already voted |
| -Voter address is not msg.sender  -voter delegate address not msg.sender | Error! sender is not delegated to cast vote    There is no delegate for Address 0x8789b0f86349370BC36E4a7ab63e33c69B193E43 |
| -Voter address is not msg.sender  -voter delegate address is msg.sender | Vote is casted against voter address by sender.  Voter details should be updated as voted and candidate id updated.  Candidate voteCount increased |
| -Voter address is same as msg.sender  msg.sender = any | Vote is casted against voter address by self.  Voter details should be updated as voted and candidate id updated.    Vote count increased for candidate |

|  |  |
| --- | --- |
| Testcases for showResults() | Expected result |
| with a non-existing candidate ID  msg.sender = any | Error! Candidate does not exist. |
| Call to showResults in remix with an existing candidate ID  msg.sender = any | Candidate details with number of votes to be returned |

|  |  |
| --- | --- |
| Testcases endElection() | Expected result |
| msg.sender = Non Admin | Error! Only admin can perform operation |
| msg.sender = Admin | Election status should be changed to COMPLETE |

|  |  |
| --- | --- |
| Testcases for showWinner() | Expected result |
| election status is not COMPLETE  msg.sender = any | Error! Election should be complete for a winner to be shown |
| election status is COMPLETE  msg.sender = any | Candidate details with maximum number of votes should be returned |