






# IMPLEMENTATION

Date	20 October 2023
Team ID	NM2023TMID02239
Project Name	Electronic Voting System
Maximum Mark	4 Marks

## REMIX IDE

The screenshot displays the Remix IDE interface, which is used for developing and testing smart contracts. The interface is divided into several sections:

- SOLIDITY COMPILER:** This section on the left contains settings for the compiler. It includes a dropdown for 'Solidity' (set to 'Solidity'), a dropdown for 'EVM VERSION' (set to 'compiler default'), and a 'COMPILER CONFIGURATION' section with checkboxes for 'Auto compile', 'Enable optimization' (set to '200'), and 'Hide warnings'. A large blue button labeled 'Compile 1\_Storage.sol' is present.
- CONTRACT:** Below the compiler settings, the 'CONTRACT' section shows the selected contract as 'Storage (1\_Storage.sol)'. It includes buttons for 'Publish on Swarm', 'Publish on Ipfs', and 'Compilation Details'.
- Code Editor:** The central area displays the Solidity code for '1\_Storage.sol'. The code includes a license header, a pragma statement for Solidity version 0.7.0, and two functions: 'store' and 'retrieve'. The 'store' function takes a 'uint256' value and stores it in a variable 'number'. The 'retrieve' function returns the value of 'number'.
- Transaction Log:** At the bottom, the transaction log shows a successful transaction. It indicates that the transaction was executed at block 9066036, transaction index 5, from address 0x40d...fFFAB to the contract address. The data field shows the result of the 'store' function: 'data: 0x605...001a4'. The log also shows the gas used (0) and the hash (0x5df...f354c).



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

JavaScript VM (London)

ACCOUNT

0x5B3...eddC4 (99.999999%)

GAS LIMIT

3000000

VALUE

0 wei

CONTRACT

Election - Election.sol

Deploy

Annual Elections

☐ Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded

Election.sol

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract Election {
5
6     struct Candidate {
7         string name;
8         bool registered;
9         uint voteCount;
10    }
11
12     struct Voter {
13         bool voted;
14         bool registered;
15         address vote;
16    }
17
18     address[] candidateAddresses;
19     address public owner;
20     string public electionName;
21
22     mapping(address => Voter) public voters;
23     mapping(address => Candidate) public candidates;
24     uint public totalVotes;
25
26     enum State { Created, Voting, Ended }
27 }
```

0

☐ listen on network

Search with transaction hash or address

✓ [vm] from: 0x5B3...eddC4 to: Election.constructor value: 0 wei data: 0x608...0000 logs: 0 hash: 0xe9f...5a536

Debug