E-voting using Blockchain Technology

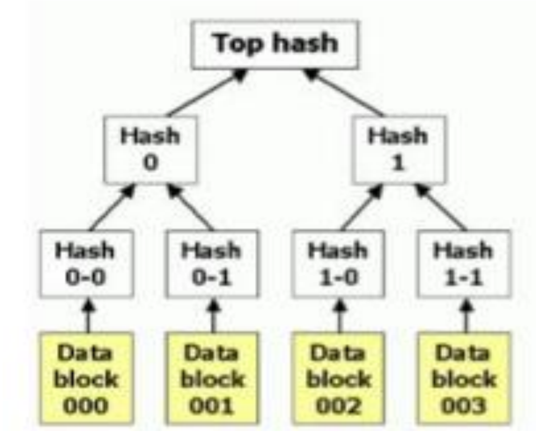
P Rujul Jadav[1], Sharath Kirubakaran[2], S Poorna Sai Teja[3], Sai Kiran S[4]
[1,2,3,4]*Dept. Computer Science, REVA University,* Bengaluru, India

Abstract— Traditional voting techniques, such as paper ballots or electronic voting machines, have fundamental weaknesses that compromise their credibility. These shortcomings include lack of transparency, poor voter turnout, potential vote tampering, distrust in electoral authorities, forging unique IDs, delays in result declaration, and security issues. Therefore, it is crucial to guarantee the security of electronic voting systems. Blockchain technology is one promising answer to the security challenges associated with digital voting. Blockchain is a distributed ledger technology that provides crucial qualities such as immutability, security, transparency, decentralization, and anonymity, making it a potentially effective alternative for developing a more secure e-voting system. This paper describes the development and testing of an example Ethereum network smart contract application for electronic voting using blockchain technology. In order to avoid the chance of duplicate votes, the system restricts the quantity of token (gas) provided in the wallet, which is used up when the user casts their vote. The pros and cons of employing blockchain technology for electronic voting are examined, and a web application for voting is used to demonstrate the practical system and highlight its drawbacks. In conclusion, creating a more secure, transparent, and reliable e-voting system seems to be a viable application of blockchain technology with smart contracts. However, before using the technology on a larger scale, it is essential to understand its limitations and deal with any potential problems.

I. INTRODUCTION

Since the introduction of Bitcoin, the first widely used cryptocurrency, blockchain technology has attracted a lot of attention in the software industry. Blockchain has grown to be a popular topic of study for its possible uses in many sectors due to its high level of transparency. Without the need for a centralized authority, the decentralized structure of Bitcoin makes it possible to track global coin volumes and transaction volumes in real-time. Additionally, this feature enables the safe storage of various types of information, including marriage licenses, bank account books, medical records, and more. After Bitcoin, another cryptocurrency called Ethereum showed that the blockchain technology could be used to build software that could hold structured data. The blockchain-written, unchangeable smart contracts cannot be changed or revoked. Thus, they can carry on operating openly and independently for however long is required.

Transactions are kept in blocks, which are added to the blockchain in a logical and chronological order. This is how the blockchain works. The genesis block, also referred to as Block 0, is the first block in a blockchain and is typically hardcoded into the program. Each block has a transaction data section that combines and hashes each transaction's hashes until only one is left, known as the Merkle root. The Merkle root is then stored in the block header, and each block additionally keeps track of the header of the preceding block to prevent the change of a transaction. While blockchain technology offers a potential solution for e-voting projects because of its transparency and privacy features, there are currently few trustworthy and operational e-voting implementations. This paper suggests a web application that makes use of blockchain technology by deploying smart contracts on the Ethereum server in order to get around the drawbacks of current electronic voting systems. The second part of the essay examines the existing status of electronic voting systems, their flaws, and the requirement for a more dependable and open system.

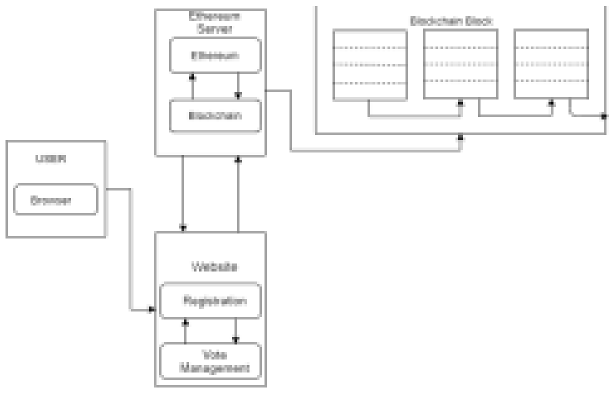


II. MOTIVATION AND RELATED WORKS

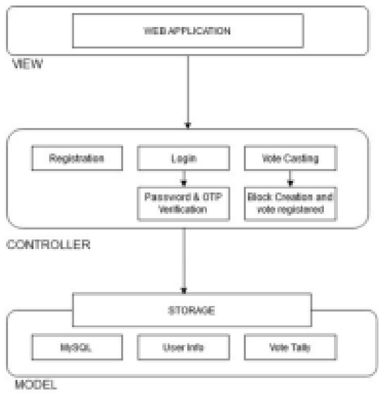
The aim of this project is to develop a secure and accessible e-voting system that can promote transparency in administrative decision-making, ultimately leading to a more direct democracy. Traditional large-scale elections can be costly and may have low voter turnout, making e-voting an attractive solution if implemented correctly. While e-voting pre-dates blockchain technology, previous implementations have relied on centralized computation and storage models, which have potential vulnerabilities. Estonia has successfully implemented a comprehensive e-voting system since 2003, using smart digital ID cards and personal card readers for authentication. Swiss startup Agora has also successfully utilized blockchain technology for tallying votes in the 2018 Sierra Leone general election. However, some online polling systems, like straw poll.me, do not provide sufficient security for real cases that require voter authentication and non-repudiation of votes. This paper proposes an e-voting protocol that integrates blockchain technology, creating a flexible and secure mechanism that meets the main requirements for a reliable e-voting system.

III. IMPLEMENTATION AND DISCUSSION

This section describes the layout and features of our web-based electronic voting system, which provides a safe and open platform for voting. Phase of Registration: Voters must first register on the platform by providing their individual ID and personal information such as their name, and mobile number. For future use, this information is safely kept in a database. Voters can log in to the portal using their password after registering. An OTP verification procedure is used to ensure real-time authentication for increased security. Blockchain Technology: To deliver cutting-edge security features, our platform makes use of blockchain technology. The voter's message, or cast vote, is encrypted and recorded on the blockchain ledger using an asymmetric encryption technique. While the private key is safely stored with the host, the public key provided by the blockchain is used for verification. Database: A database is used to securely store all user data, including their name, gender, and unique ID. For this, MySQL is what we recommend. Ethereum Network: Our platform makes use of the Ethereum network, which offers a structure for building and storing blockchains. To guarantee great fault tolerance, each block that is formed is divided among nodes and kept on an encrypted ledger. Phase of Results: Votes are processed and tallied during this stage. Results are created and shown on the website, and each user is able to use their public key to confirm their vote. The voting process becomes much more transparent as a result. Overall, our e-voting system provides a versatile and secure method that meets the majority of the key criteria for an e-voting system. A safe and open environment is ensured by the use of blockchain technology, and the platform's security is further increased by the use of OTP verification and database storage.



Model-View-Controller (MVC) architecture, which is made up of three logical parts (the view, the controller, and the model), is used to create the Program art view layer, which is the topmost layer of the application, is in charge of interacting with the user through various features like buttons, text input fields, radio buttons, camera access, and file upload choices. Depending on the needs of the application, this layer shows the user all the data or a subset of the data. The view layer also acts as a channel for user and application communication. The controller layer, which is the application's middle layer, houses the application's core functionality and business logic. The response is processed in this layer right away once a user interacts with the application. It consists of every background process, including user login and vote casting. Most of the controller layer's components are used to transmit output to the view layer. The Interplanetary File System (IPFS), a decentralized and distributed database management system, is used by the model layer, which is in charge of managing user data. This layer controls the system for storing and retrieving data, acting as the foundation of the program.



On the Ethereum network, where our voting application runs, users must have an account, a wallet address, and some Ether (the money used by Ethereum), in order to vote. They spend a modest transaction cost called "gas," which is given to the network's miner-node upon completion of the transaction, to publish their vote to the blockchain. While viewing the list of candidates on the blockchain costs Ether, reading data from the blockchain is free. We use Solidityprogrammed smart contracts that run code on the Ethereum Virtual Machine (EVM) on the blockchain to create our application. These smart contracts are in charge of carrying out the logic and reading and publishing data to the blockchain. They signify a commitment that votes cast by users will be taken into account, that votes cast by others will only be counted once, and that the candidate receiving the most votes would be proclaimed the winner. Installing all required dependencies and creating our contract come first in the process of developing our application, which is then successfully deployed to the blockchain. The "contract" keyword and the contract name are used to declare the smart contract. Next, we declare a state variable to hold the candidate's name's value. Every time the contract is deployed, the constructor is invoked. In order to create a safe and transparent voting platform that ensures every user's vote is correctly counted, our application's design and functionality significantly rely on the Ethereum blockchain and smart contracts. Overall, our application uses Ethereum's blockchain and smart contracts to give users access to a reliable, transparent, and effective voting system.

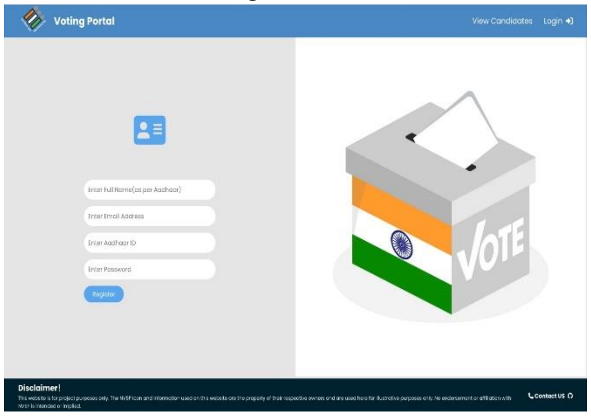
```
// SPDX-License-Identifier: AFL-3.0
pragma solidity ^0.8.18;

contract VotingSystem {
struct Candidate {
string candidateName;
uint32 voteCount;
}

Candidate[] public candidates;

function addCandidate(string memory _name) public {
candidates.push(Candidate(_name, 0));
}
}
```

We use Solidity mapping in our voting application to securely and logically store the candidate data. A candidate's structure is specified by their unsigned integer type ID, string type name, and unsigned integer type vote total. A getter function can be used to quickly obtain the mapping, which is built with the key-value pair of an unsigned integer and the Candidate structure type and is set to public visibility.

Our contract code has a function to add candidates as well as a mapping. "Contract electi" is the name of the complete smart contract, which was created using the Solidity programming language. The Ethereum Virtual Machine (EVM) on the blockchain is used to carry out the contract, ensuring a safe and open voting process. Our application is a safe and stable platform for voting thanks to the use of mapping and Solidity smart contracts, which ensure the reliable storage and execution of the voting process on the blockchain.

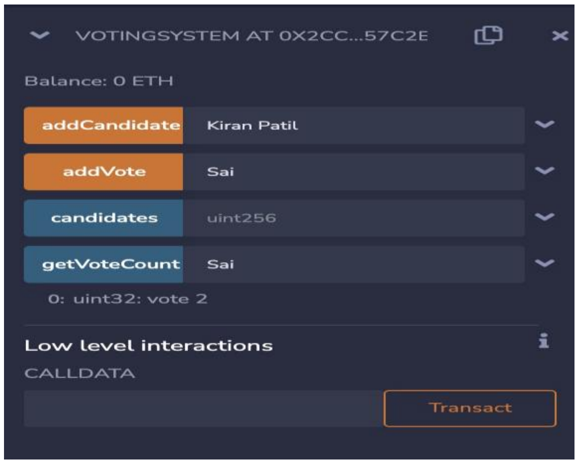
```
contract Election {
  struct Candidate {
    uint id;
    string name;
    uint32 voteCount;
  }

  Candidate[] public candidates;
  mapping(string => bool) private candidateExists;

  function addCandidate(string memory _name) public {
    require(!candidateExists[_name], "Candidate already
exists");
    candidates.push(Candidate(candidates.length, _name,
0));
    candidateExists[_name] = true;
  }

  function addVote(string memory _name) public {
    require(candidateExists[_name], "Candidate does not
exist");
    for (uint i=0; i<candidates.length; i++) {
      if(keccak256(bytes(candidates[i].name)) ==
keccak256(bytes(_name))) {
        candidates[i].voteCount++;
        break;
      }
    }
  }

  function getVoteCount(string memory _name) public
view returns(uint32) {
    require(candidateExists[_name], "Candidate does not
exist");
    for (uint i=0; i<candidates.length; i++) {
      if(keccak256(bytes(candidates[i].name)) ==
keccak256(bytes(_name))) {
        return candidates[i].voteCount;
      }
    }
    revert ("Candidate does not exist");
  }
}
```

After establishing the server-side application, we focused on developing a client-side application that could communicate with our smart contract. OTP (one-time password) verification was introduced as an additional layer of protection to our front end, which was constructed using HTML and JavaScript. To receive an OTP, users had to input their mobile numbers. The OTP was then used to confirm the users' identities. We used MetaMask, a browser extension that lets users interact with the Ethereum network, to import one of the accounts from Ganache, a dependency that offers 10 fictitious identities with phony Ethereum addresses and ethers. This enabled us to access our smart contract and account information by logging into the blockchain and interacting with it.



Disclaimer!
This website is for project purposes only. The NVSP icon and information used on this website are the property of their respective owners and are used here for illustrative purposes only. No endorsement or affiliation with NVSP is intended or implied.

We have included a mapping in our smart contract to keep track of the accounts that have voted in order to make voting in the election easier. We also created a 'vote' function to the smart contract that only accepts one argument, the candidate ID. This procedure runs several checks to make sure the user hasn't voted before, verifies the candidate ID, logs the user's vote, and finally updates the candidate's vote total. It then causes a "voted" event. The user must not have previously cast a ballot in order to use the 'vote' function, and the candidate ID must be legitimate. If both conditions are satisfied, the function updates the candidate's vote total and logs the user's vote. In order to signal that a vote has been cast, it additionally emits a 'voted' event. The accounts that have voted are stored using the mapping function. It associates an address with a Boolean value that represents whether or not the account has cast a vote.



```
VOTINGSYSTEM AT 0X2CC...57C2E

Balance: 0 ETH

addCandidate    Kiran Patil
addVote         Sai
candidates      uint256
getVoteCount    Sai
  0: uint32: vote 2

Low level interactions
CALLDATA
                                Transact
```

Achieving anonymity in blockchain-based electronic voting systems is a major difficulty but is necessary for secure and fair voting. In such systems, every transaction, including votes and money transfers, is publicly recorded on the blockchain, making it possible for anyone with access to the chain to examine them. Due to this, it is challenging to use such systems for official or crucial elections without endangering voter privacy. Researchers have suggested a number of solutions to this problem, including a two-round Diffie-Hellman process-based referendum system that makes use of random numbers and public/private key pairs. However, due to scalability issues, these solutions are not yet widely used and might not be appropriate for large-scale elections with millions of voters. Our product is intended for local elections and polls, such college elections. The election results are shown following a successful vote cast, and a record of the vote is also recorded on the blockchain. The chain contains information about each transaction, including its hash, total cost, number of blocks created thus far, contract address, date, account, and block number. The Ethereum blockchain, which is available through any browser on any device or platform, can be used to carry out our contracts. The scalability of the Ethereum network is still a topic of active research, so we cannot, at least for the time being, propose using these contracts for national elections. To make secure and anonymous blockchain-based e-voting a reality for everyone, we remain committed to constantly developing our system and looking into new options.

IV. CONCLUSION

The innovative electronic voting system described in this paper is built on blockchain technology and use smart contracts to provide safe, affordable, and private elections. The study demonstrates how blockchain technology may provide democratic nations with a fresh and improved voting system, swapping out the old pen-and-paper voting procedures for a system that is more effective, secure, and transparent. Despite some successful e-voting system examples, many have fallen short in terms of offering the required security and privacy features or have encountered usability and scalability problems. Nevertheless, blockchain-based e-voting solutions, such as the one used in this study that uses smart contracts on the Ethereum network, address nearly all of the security issues related to voter privacy, integrity, verification, and transparency of counting. However, some characteristics, like individual-level voter authentication, might necessitate the incorporation of extra mechanisms, such as biometric factors. It's crucial to remember that blockchain technology has a lot of potential, but more study is necessary before it can reach its full potential. Therefore, efforts should be made in the future to enhance its support for sophisticated applications.

REFERENCE

[1] "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551,

April 1955; G. Eason, B. Noble, and I. N. Sneddon.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, Third Edition, Volume 2, Oxford: Clarendon, 1892, pp. 68–73.

[3] "Fine particles, thin films, and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds., New York: Academic, 1963, pp. 271-350. I. S. Jacobs and C. P. Bean.

[4] K. Elisa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalised," Journal of Name Stand.

[6] "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Trans. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982]. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa.

[7] The Technical Writer's Handbook by M. Young. University Science, Mill Valley, California, 1989.