



Development of Decentralized application for E-voting using Blockchain

¹ G Rakesh Kumar, ² O Satya Manoj Kumar, ³ E Vamsi Ganesh, ⁴ G Venkatasaikumar, ⁵ S Shakeer Ahamed

¹ Assistant Professor ^{2,3,4,5} B.Tech Scholar,

^{1,2,3,4,5} Department Of Electronics And Communications Engineering

^{1,2,3,4,5} G. Pullaiah College of Engineering and Technology, Nandikotkur Rd, near Venkayapalle, Pasupula Village, Kurnool, Andhra Pradesh 518002, India.

Abstract:

In, the modern world, maintaining the security of data plays a vital role. As we can see, today's world is moving away from Centralized to Decentralized systems, which prevents data from being centralized (or) maintained by central authorities. As a result, Blockchain technology enables Decentralized system by utilizing distributed networks and storing data in the form of interconnected blocks. E-voting is a method of voting that does away with the use of paper ballots. As a result, using Blockchain Technology to create a Decentralized Application for E-Voting will address issues with the current voting system, such as vote tampering, data manipulation during vote counting, and so on. A Blockchain-based Decentralized system ensures great security and transparency by storing data in the form of Casted votes in Blockchain. As a result, if we proceed in this fashion, we will be able to obtain fair election results using e-voting.

Keywords: Centralized system, Decentralized system, Security, Blockchain, interconnected blocks, E-voting, Ethereum, etc.,

1. Introduction:

"Blockchain" is a technology that is rapidly gaining momentum in the modern world due to its high security and transparency features. It is widely utilised in supply chain management systems, healthcare, payments, business, IoT, voting systems, and other applications. Blockchain is a method of storing data in such a way that it is difficult or impossible to change, hack, or cheat it. However, putting together a blockchain system isn't enough. It should be "Decentralized".

So here is a list of benefits that a Blockchain can provide:

- Secure
- You can vote at any time and from any location (During Pandemics like COVID-19 where it is impossible to hold elections physically)
- Immutable
- Quick

In this project, we will develop a Decentralized Application for E-Voting System based on Blockchain technology, with a Topology Network in which each new node implants the hash of the previous node as the root hash of the current node activity, forming a chain of hash-nodes in the end. Although it may not be obvious, such a Merkle tree structure does not necessitate the use of a local server. Members can conduct transactions without the need for a local affirming authority. Long lines of voters, security breaches such as data leaks, vote tampering, a mountain of paperwork, difficulty for differently abled people to attend voting booths, and a high cost of election expenditure are just some of the issues that a nation is currently facing throughout the polling process.

Currently well-known voting frameworks depend on an excellent customer server design. Blockchain, on the other hand, is an emerging technology that enables decentralization with no single point of failure and ensures the unchanging nature of information through encryption and the use of a proof-of-work protocol. The Ethereum Blockchain is an open-source scripting language that

enables programmer to create decentralised applications.

We can also refer to it as the D- application. The decentralized and distributive nature of blockchain are the key features on which the whole system is based upon. Furthermore, the system's immutability of the system ensures that there is no scope of tampering as any and every transaction that has been recorded has been done so permanently.

2. Literature survey

Increasingly digital technology in the present helped many people lives. Unlike the electoral system, there are many conventional uses of paper in its implementation. The aspect of security and transparency is a threat from still widespread election with the conventional system (offline). Block chain technology is one of solutions, because it embraces a decentralized system and the entire database are owned by many users. There is no doubt that the revolutionary concept of the blockchain, which is the underlying technology behind the famous crypto currency Bitcoin and its successors, is triggering the start of a new era in the Internet and the online services. In this work, we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language.

The paper explores the growing of the blockchain development and its usage finish in the law based plan. The blockchain will be openly certain and appropriated with the end goal that no one will have the choice to deteriorate. The way they have explained in implementing the Decentralized Application using web technologies and software tools to create a Block chain network provides high security of Data. But we will explore more details about the code that is related smart contracts.

Sven Heiberg, Ivo Kubjas, Janno Siim, and Jan Willemson studied had a brief description of conditions that need to be verified in order for the voting event to be considered right. Currently, using smart contracts seems to be the most systematic approach to deal with this issue, but systems using smart contracts so far imply a significant performance penalty, strongly limiting e.g. the number of voters. The consistency verification of block chain based voting systems is rather complex, defying the original target of transparency. It may be the case that simplicity of the verification routines needs to be recognized as a development requirement of its own right.

Ethereum is a decentralized, open-source blockchain with smart contracts functionality. Ether is the native cryptocurrency of the platform. Among cryptocurrencies, Ether is the second only to the Bitcoin in market capitalization. Ethereum allows any one to deploy permanent and immutable decentralized application onto it, with which users can interact

3. Proposed Methodology:

Blockchain:

A blockchain is a database that stores encrypted blocks of data. The chains link them together to form a chronological single-source-of-truth for the data.

Digital assets are distributed instead of copied or transferred, creating an immutable record of an asset. The asset is decentralized, allowing full real-time access and transparency to the public. A transparent ledger of changes preserves integrity of the document, which creates trust in the asset.

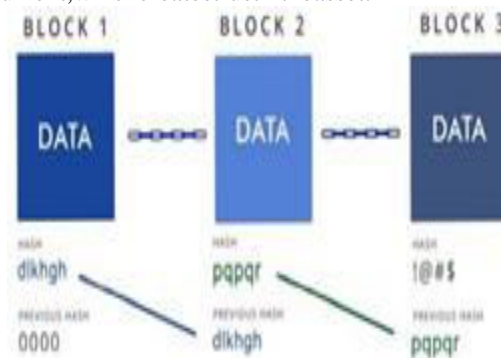


Fig 1: Structure of Blockchain

Working Principle:

Here in this application, we are not going to focus much on interacting the application with the command line terminal.

We mainly focusing on the interacting with application through web browser, for which we need to create graphical user interface which comes under frontend development. In the backend, we will connect blockchain to store the data from client side application along with smart contract. After the completion of both frontend and backend development, we will test our the smart contract with the help of Truffle Framework. After that we will deploy our smart contract into local ethereum running blockchain in order to process the

data the of the application. Then our application will be running on the local server.

There are several tools are used in this project, like Ganache, Truffle framework, Meta mask extension for google chrome, Sublime Text Editor , Node Package Manager and Front Technologies.

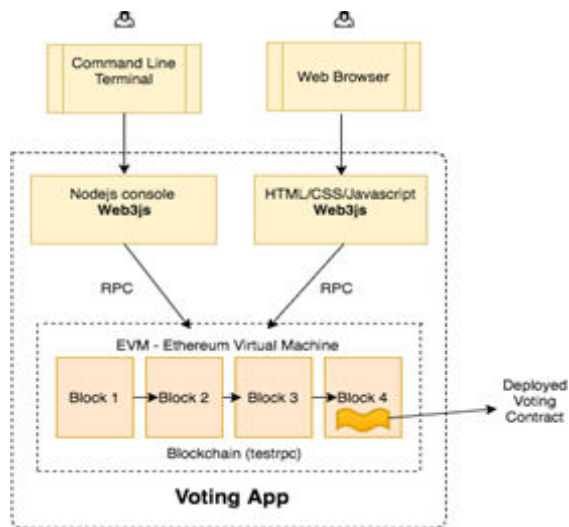


Fig 2 Architecture of Dapps for E-voting.

Before beginning to construct the Decentralized application, we check to see that all of the requirements have been installed. Let's get started now that everything has been completed. Our application will be developed on an Ubuntu OS.

A Terminal window functions as an interface through which we can control our operating system using preset commands. All of our dependencies are installed using terminal. We will not create our decentralised application in such a way that we can interact with it on a terminal from an architectural standpoint. We are primarily concerned with the Graphical User Interface (GUI), which will allow our decentralised programme to interface with the client. Our smart contract will also be deployed, tested, and run on the terminal. Web browser is the interface through which the user client will interact with our application.

We require a graphical user interface to interact with the application in order to participate in voting elections, select the candidate for whom we

will vote, and cast and submit our votes. This graphical user interface may be created using HTML (hypertext markup language), CSS (cascading style sheets) to provide color to our programme, and Bootstrap (CSS Framework) which comes with the web3js package.

Smoke Testing is a software testing procedure that assesses whether a software build has been delivered and is stable. Smoke testing allows the QA team to continue with the rest of the software testing. It is made up of a small number of tests that are run on each build to verify programme functionality. "Build Verification Testing" or "Confidence Testing" are other terms for smoke testing. First, we'll see that the ganache has been downloaded and opened via the command line terminal, and that the local blockchain is now running on the PC.



Fig 3: Terminal window

It will first provide ten false accounts, each with its own unique address and private key. Each account is linked to a total of 100 ethers and runs on port 7545. Each voter's account address will be used as a unique identifier in our election. As can be seen, Ganache has generated ten phoney accounts, each with 100 fake ethers, which will be used to deploy smart contracts and cast votes, among other things.

Let's get started writing our smart contract now. This smart contract will hold all of our decentralised application's business logic. It will be responsible for both reading and writing to the Ethereum blockchain. It will allow us to keep track of all the votes and voters as well as list the candidates that will run in the election. It will also oversee all election rules, such as requiring accounts to vote just once. From the root of our project, Create a new contract file in the contracts directory like this:


```
$touch contracts/Election.sol
```



```
pragma solidity ^0.4.1;
contract Election {
    uint public candidatesCount;
    string public candidate;
    // constructor
    constructor() public {
        candidate = "Candidate 1";
    }
}
```

Fig 4: Smart Contract for smoke Test

Westartbydeclaringthesolidityversionwiththe **pragma solidity** statement. Next, we declare the smart contract with the "**contract**" keyword, followed by the contract name. Next, we declare a state variable that will store the value of the candidate name. We can write data to the blockchain using state variables. This variable has been declared as a string, and its visibility has been set to public. Solidity will provide us with a getter method for free because it is public, allowing us to access this value outside of our contract.



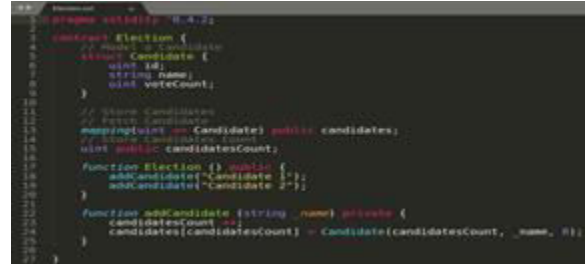
```
2_deploy_contracts.js x
1 var Election = artifacts.require("./Election.sol");
2
3 module.exports = function(deployer) {
4   deployer.deploy(Election);
5 };
6
```

Fig 5: Migrating File for Dapps deployment



```
Truffle v5.1.10 (2021-05-26)
Solidity 0.8.10
Network: main
Using network: main
Deploying Migration for smoke
Deploying Election
Contract Election deployed at 0x0000000000000000000000000000000000000000
Transaction hash: 0x0000000000000000000000000000000000000000
Deployed 1 contract: Election
Contract Election deployed at 0x0000000000000000000000000000000000000000
Transaction hash: 0x0000000000000000000000000000000000000000
Deployed 1 contract: Election
```

Fig 6: Deploying the smart contract for smoke test



```
pragma solidity ^0.4.21;
contract Election {
    // state variable
    Candidate[] public candidates;
    uint public candidatesCount;
}
// constructor
constructor() public {
    addCandidate("Candidate 1");
    addCandidate("Candidate 2");
}
function Election() public {
    candidatesCount = candidates.length;
}
function addCandidate (string _name) public {
    candidates[candidatesCount] = Candidate(candidatesCount, _name, 0);
}
```

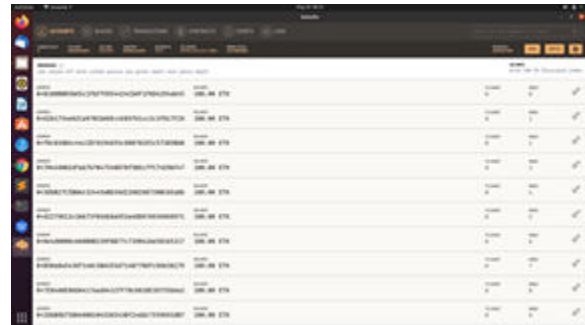
Fig 7: Listing of candidates in smart contract.

We'll need to write some tests first, so make sure Ganache is up and running. Then, using the command, create a new test file in the command line from the root of your project.

```
touch test/election.js
```

4. Results and Discussions:

We start by checking that the local blockchain (ganache) is up and running. And these are Ganache's false accounts, each of which is linked to 100 bogus ethers.



Account	Balance (ETH)	Private Key
0x00	100.00	0x00
0x0000000000000000000000000000000000000001	100.00	0x0000000000000000000000000000000000000001
0x0000000000000000000000000000000000000002	100.00	0x0000000000000000000000000000000000000002
0x0000000000000000000000000000000000000003	100.00	0x0000000000000000000000000000000000000003
0x0000000000000000000000000000000000000004	100.00	0x0000000000000000000000000000000000000004
0x0000000000000000000000000000000000000005	100.00	0x0000000000000000000000000000000000000005
0x0000000000000000000000000000000000000006	100.00	0x0000000000000000000000000000000000000006
0x0000000000000000000000000000000000000007	100.00	0x0000000000000000000000000000000000000007
0x0000000000000000000000000000000000000008	100.00	0x0000000000000000000000000000000000000008
0x0000000000000000000000000000000000000009	100.00	0x0000000000000000000000000000000000000009
0x000000000000000000000000000000000000000a	100.00	0x000000000000000000000000000000000000000a
0x000000000000000000000000000000000000000b	100.00	0x000000000000000000000000000000000000000b
0x000000000000000000000000000000000000000c	100.00	0x000000000000000000000000000000000000000c
0x000000000000000000000000000000000000000d	100.00	0x000000000000000000000000000000000000000d
0x000000000000000000000000000000000000000e	100.00	0x000000000000000000000000000000000000000e
0x000000000000000000000000000000000000000f	100.00	0x000000000000000000000000000000000000000f
0x0000000000000000000000000000000000000010	100.00	0x0000000000000000000000000000000000000010
0x0000000000000000000000000000000000000011	100.00	0x0000000000000000000000000000000000000011
0x0000000000000000000000000000000000000012	100.00	0x0000000000000000000000000000000000000012
0x0000000000000000000000000000000000000013	100.00	0x0000000000000000000000000000000000000013
0x0000000000000000000000000000000000000014	100.00	0x0000000000000000000000000000000000000014
0x0000000000000000000000000000000000000015	100.00	0x0000000000000000000000000000000000000015
0x0000000000000000000000000000000000000016	100.00	0x0000000000000000000000000000000000000016
0x0000000000000000000000000000000000000017	100.00	0x0000000000000000000000000000000000000017
0x0000000000000000000000000000000000000018	100.00	0x0000000000000000000000000000000000000018
0x0000000000000000000000000000000000000019	100.00	0x0000000000000000000000000000000000000019
0x000000000000000000000000000000000000001a	100.00	0x000000000000000000000000000000000000001a
0x000000000000000000000000000000000000001b	100.00	0x000000000000000000000000000000000000001b
0x000000000000000000000000000000000000001c	100.00	0x000000000000000000000000000000000000001c
0x000000000000000000000000000000000000001d	100.00	0x000000000000000000000000000000000000001d
0x000000000000000000000000000000000000001e	100.00	0x000000000000000000000000000000000000001e
0x000000000000000000000000000000000000001f	100.00	0x000000000000000000000000000000000000001f
0x0000000000000000000000000000000000000020	100.00	0x0000000000000000000000000000000000000020
0x0000000000000000000000000000000000000021	100.00	0x0000000000000000000000000000000000000021
0x0000000000000000000000000000000000000022	100.00	0x0000000000000000000000000000000000000022
0x0000000000000000000000000000000000000023	100.00	0x0000000000000000000000000000000000000023
0x0000000000000000000000000000000000000024	100.00	0x0000000000000000000000000000000000000024
0x0000000000000000000000000000000000000025	100.00	0x0000000000000000000000000000000000000025
0x0000000000000000000000000000000000000026	100.00	0x0000000000000000000000000000000000000026
0x0000000000000000000000000000000000000027	100.00	0x0000000000000000000000000000000000000027
0x0000000000000000000000000000000000000028	100.00	0x0000000000000000000000000000000000000028
0x0000000000000000000000000000000000000029	100.00	0x0000000000000000000000000000000000000029
0x000000000000000000000000000000000000002a	100.00	0x000000000000000000000000000000000000002a
0x000000000000000000000000000000000000002b	100.00	0x000000000000000000000000000000000000002b
0x000000000000000000000000000000000000002c	100.00	0x000000000000000000000000000000000000002c
0x000000000000000000000000000000000000002d	100.00	0x000000000000000000000000000000000000002d
0x000000000000000000000000000000000000002e	100.00	0x000000000000000000000000000000000000002e
0x000000000000000000000000000000000000002f	100.00	0x000000000000000000000000000000000000002f
0x0000000000000000000000000000000000000030	100.00	0x0000000000000000000000000000000000000030
0x0000000000000000000000000000000000000031	100.00	0x0000000000000000000000000000000000000031
0x0000000000000000000000000000000000000032	100.00	0x0000000000000000000000000000000000000032
0x0000000000000000000000000000000000000033	100.00	0x0000000000000000000000000000000000000033
0x0000000000000000000000000000000000000034	100.00	0x0000000000000000000000000000000000000034
0x0000000000000000000000000000000000000035	100.00	0x0000000000000000000000000000000000000035
0x0000000000000000000000000000000000000036	100.00	0x0000000000000000000000000000000000000036
0x0000000000000000000000000000000000000037	100.00	0x0000000000000000000000000000000000000037
0x0000000000000000000000000000000000000038	100.00	0x0000000000000000000000000000000000000038
0x0000000000000000000000000000000000000039	100.00	0x0000000000000000000000000000000000000039
0x000000000000000000000000000000000000003a	100.00	0x000000000000000000000000000000000000003a
0x000000000000000000000000000000000000003b	100.00	0x000000000000000000000000000000000000003b
0x000000000000000000000000000000000000003c	100.00	0x000000000000000000000000000000000000003c
0x000000000000000000000000000000000000003d	100.00	0x000000000000000000000000000000000000003d
0x000000000000000000000000000000000000003e	100.00	0x000000000000000000000000000000000000003e
0x000000000000000000000000000000000000003f	100.00	0x000000000000000000000000000000000000003f
0x0000000000000000000000000000000000000040	100.00	0x0000000000000000000000000000000000000040
0x0000000000000000000000000000000000000041	100.00	0x0000000000000000000000000000000000000041
0x0000000000000000000000000000000000000042	100.00	0x0000000000000000000000000000000000000042
0x0000000000000000000000000000000000000043	100.00	0x0000000000000000000000000000000000000043
0x0000000000000000000000000000000000000044	100.00	0x0000000000000000000000000000000000000044
0x0000000000000000000000000000000000000045	100.00	0x0000000000000000000000000000000000000045
0x0000000000000000000000000000000000000046	100.00	0x0000000000000000000000000000000000000046
0x0000000000000000000000000000000000000047	100.00	0x0000000000000000000000000000000000000047
0x0000000000000000000000000000000000000048	100.00	0x0000000000000000000000000000000000000048
0x0000000000000000000000000000000000000049	100.00	0x0000000000000000000000000000000000000049
0x000000000000000000000000000000000000004a	100.00	0x000000000000000000000000000000000000004a
0x000000000000000000000000000000000000004b	100.00	0x000000000000000000000000000000000000004b
0x000000000000000000000000000000000000004c	100.00	0x000000000000000000000000000000000000004c
0x000000000000000000000000000000000000004d	100.00	0x000000000000000000000000000000000000004d
0x000000000000000000000000000000000000004e	100.00	0x000000000000000000000000000000000000004e
0x000000000000000000000000000000000000004f	100.00	0x000000000000000000000000000000000000004f
0x0000000000000000000000000000000000000050	100.00	0x0000000000000000000000000000000000000050
0x0000000000000000000000000000000000000051	100.00	0x0000000000000000000000000000000000000051
0x0000000000000000000000000000000000000052	100.00	0x0000000000000000000000000000000000000052
0x0000000000000000000000000000000000000053	100.00	0x0000000000000000000000000000000000000053
0x0000000000000000000000000000000000000054	100.00	0x0000000000000000000000000000000000000054
0x0000000000000000000000000000000000000055	100.00	0x0000000000000000000000000000000000000055
0x0000000000000000000000000000000000000056	100.00	0x0000000000000000000000000000000000000056
0x0000000000000000000000000000000000000057	100.00	0x0000000000000000000000000000000000000057
0x0000000000000000000000000000000000000058	100.00	0x0000000000000000000000000000000000000058
0x0000000000000000000000000000000000000059	100.00	0x0000000000000000000000000000000000000059
0x000000000000000000000000000000000000005a	100.00	0x000000000000000000000000000000000000005a
0x000000000000000000000000000000000000005b	100.00	0x000000000000000000000000000000000000005b
0x000000000000000000000000000000000000005c	100.00	0x000000000000000000000000000000000000005c
0x000000000000000000000000000000000000005d	100.00	0x000000000000000000000000000000000000005d
0x000000000000000000000000000000000000005e	100.00	0x000000000000000000000000000000000000005e
0x000000000000000000000000000000000000005f	100.00	0x000000000000000000000000000000000000005f
0x0000000000000000000000000000000000000060	100.00	0x0000000000000000000000000000000000000060
0x0000000000000000000000000000000000000061	100.00	0x0000000000000000000000000000000000000061
0x0000000000000000000000000000000000000062	100.00	0x0000000000000000000000000000000000000062

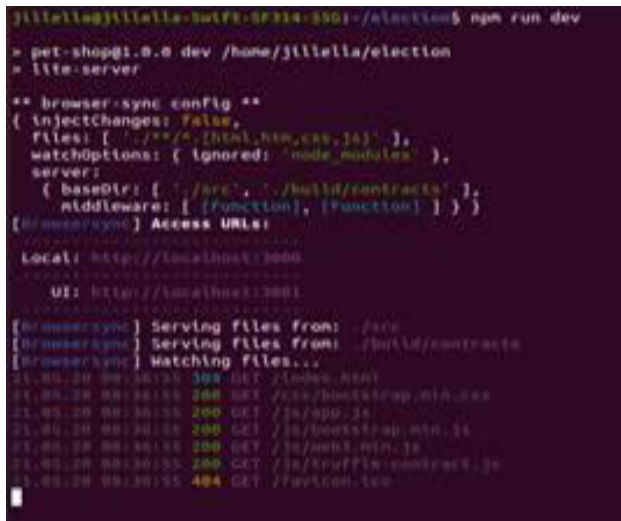


Fig 8 :Using of terminal to run application

Now we can vote using account 1 and it will cost us a gas tax to store the results of our vote and if the account 1 is voted once he/she can't be vote another time and to the person's id and to whom he voted can't be seen . The account name will be shown in the form of id which can't be easily determined. So that it provides security and anonymity to the user.

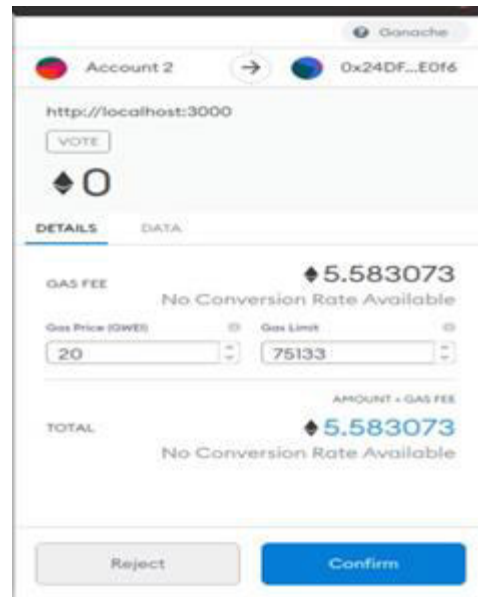


Fig 9:Meta mask pop up while casting the vote

After the confirmation of vote by the user and clicking confirm button and then gas will be cut, and even if the user try to cast their vote again it doesn't show any option to caste their vote. This will reduces vote tampering and one person can vote only one time. After the casting of vote as it is decentralized we can see instantly how the votes are being altered with number of votes being casted.

Election Results		
#	Name	Votes
1	Candidate 1	1
2	Candidate 2	0

Select Candidate
Candidate 1

Your Account: 0x90db1be94ae5ab3d6887fcfd75519ca8b0d6f9

Fig 9Casting of vote by account1

A meta mask pop up will be displayed so that the particular gas fees is cut during casting the vote which is already given to cast single vote.

Election Results		
#	Name	Votes
1	Candidate 1	2
2	Candidate 2	0

Your Account: 0x90db1be94ae5ab3d6887fcfd75519ca8b0d6f9

Fig 10: Election result page

5. Conclusion:

The usage of ballot papers was first utilized in polling systems, but as the electronic era progressed, the use of electronic voting machines (EVMs) became more common. Now that the world is moving toward digitization and automation, new polling approaches are needed to maintain transparency and eliminate flaws in the current polling system. With breakthroughs in technology such as Machine Learning, Blockchain, and the Internet of Things, an e-Voting system in the future appears to be a



possibility. To conclude, our implementation of this project will overcome the major issues facing by the today's voting system and eliminate the third party involvement to manage the data. Since it is P2P distributed network, Even if single node in the network gets damaged will not affect the other nodes, Whereas in centralization it will affect the whole network.

References:

1. Anubhav Mishra, Anuroop Mishra, Abhyudya Bajpai and Abhinav Mishra. (2020). Implementation of Blockchain for Fair Polling System. 10.1109/ICOSEC49089.2020.9215354.
2. P. Sri Subhash, K. Kiran Kumar, R. Goresh Chowdary, B. Sai Teja, P. S. G. Aruna Sri "Decentralized Application on Voting System" International Journal of Engineering and Technology 2016, Volume 8, Issue 4.
3. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73. [4] King-Hang Wang, Subrota K. Mondal, Ki Chan, Xiaoheng Xie, "A review of contemporary e-voting: Requirements, technology, systems and usability", Data Science and Pattern Recognition, vol. 1, no. 1, pp. 31-47, [2017].
4. King-Hang Wang, Subrota K. Mondal, Ki Chan, Xiaoheng Xie, "A review of contemporary e-voting: Requirements, technology, systems and usability", Data Science and Pattern Recognition, vol. 1, no. 1, pp. 31-47, [2017].
5. Snehal Kadam¹, Khushaboo Chavan², Ishita Kulkarni³, Prof. Amrut Patil⁴, "Survey on Digital E-Voting System by using Blockchain Technology". || Volume 4 || Issue 2 || February 2019 || ISSN (Online) 2456-0774.
6. Mahesh Murthy, "Full Stack Hello World Voting Ethereum

murthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-140d2d0d807c2.

7. Friðrik P. Hjálmarsson, Gunnlaugur K. Hreiðarsson, "Blockchain Based E-Voting System", Online: <https://skemman.is/bitstream/1946/31161/1/Research-PaperBBEVS.pdf>.
8. <https://en.wikipedia.org/wiki/Ethereum>
9. <https://en.wikipedia.org/wiki/Blockchain>
10. Fusco, Francesco, Maria Iliaria Lunesu, FILIPPO EROS Pani, and Andrea Pinna. "Crypto-voting, a Blockchain based e-Voting System." In KMIS, pp. 221-225. 2018.
11. Ayed, Ahmed Ben. "A conceptual secure blockchain-based electronic voting system." International Journal of Network Security & Its Applications 9, no. 3 (2017): 01-09.

Election Results

#	Name	Votes
1	Candidate 1	2
2	Candidate 2	0

Your Account: 0x90d11be94ae5ab3d9588875cfd075519cadb0d6f9

DappTutorial": <https://medium.com/@mv>