A REPORT

ON

DECENTRALIZED APPLICATION FOR VOTING USING BLOCKCHAIN ARCHITECTURE ON ETHEREUM

BY

SAHIL SHARMA	2014A7PS0408U
SIMRAN KAUR BHATIA	2014A7PS0334U
MIRAL SHAH	2014A7PS0200U

FOR

F422 INTRODUCTION TO CRYPTOGRAPHY

BITS Pilani, Dubai Campus
Dubai International Academic City, Dubai
UAE

A REPORT

ON

DECENTRALIZED APPLICATION FOR VOTING USING BLOCKCHAIN ARCHITECTURE ON ETHEREUM

BY

SAHIL SHARMA	2014A7PS0408U
SIMRAN KAUR BHATIA	2014A7PS0334U
MIRAL SHAH	2014A7PS0200U

Prepared in Fulfillment of the Course

F422 INTRODUCTION TO CRYPTOGRAPHY

BITS Pilani, Dubai Campus
Dubai International Academic City, Dubai
UAE

BITS Pilani, Dubai Campus
Dubai International Academic City, Dubai
UAE

Subject: Introduction to Cryptography **Subject Code:** F422

Date of Demonstration: 15.05.2018

Title of the Project: Decentralized Application for Voting Using Blockchain Architecture On Ethereum

ID No. / Name of the students: 2014A7PS0334U/ Simran Kaur Bhatia
2014A7PS0408U/ Sahil Sharma
2014A7PS0200U/ Miral Shah

Discipline of Students: Computer Science

Name of the Faculty: Mr. Saurabh Jain

Key Words: Ethereum, Blockchain, Solidity, Cryptography, Proof-of-work, Mining, Encryption/Decryption

Project Area(s): Software engineering, Computer and Information Technology, Web Development, Cryptography, Blockchain implementation

Abstract:

This report pertains to the complete works on the software life cycle development specifically in the fields of the blockchain architecture used in web development. It does so by exploring and documenting the process of creating a decentralized voting application by initializing a set of candidates, allowing anyone who has access to the application to vote for the candidates, and finally, displaying the total votes earned by each candidate. The project aimed to make blockchain architecture- as a concept, Etheureum- as as a platform, and Solidity -as a language more comprehensible. This is achieved by learning the process of compiling, deploying and interacting with the application, as opposed to just writing the code. The report explains the following processes in detail:

1. Setting up of the development environment.
2. The process of writing a smart contract with Solidity IDE, compiling it, and deploying it to a private blockchain network
3. Interacting with the contract on the blockchain through a nodejs console.
4. Interacting with the contract through a web page to display the vote counts and vote for candidates through the page.

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

Blockchain

By allowing digital information to be distributed but not copied, blockchain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin, the tech community is now finding other potential uses for the technology.

A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a cryptographic hash of the previous block, a timestamp and transaction data. By design, a blockchain is inherently resistant to modification of the data. It is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way. For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without the alteration of all subsequent blocks, which requires collusion of the network majority.

Blockchains are secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been achieved with a blockchain. This makes blockchains potentially suitable for the recording of events, medical records, and other records management activities, such as identity management, transaction processing, documenting provenance, food traceability or voting.

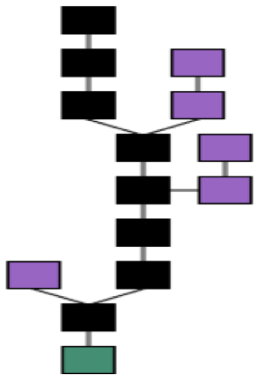


Fig 1. Blockchain Formation

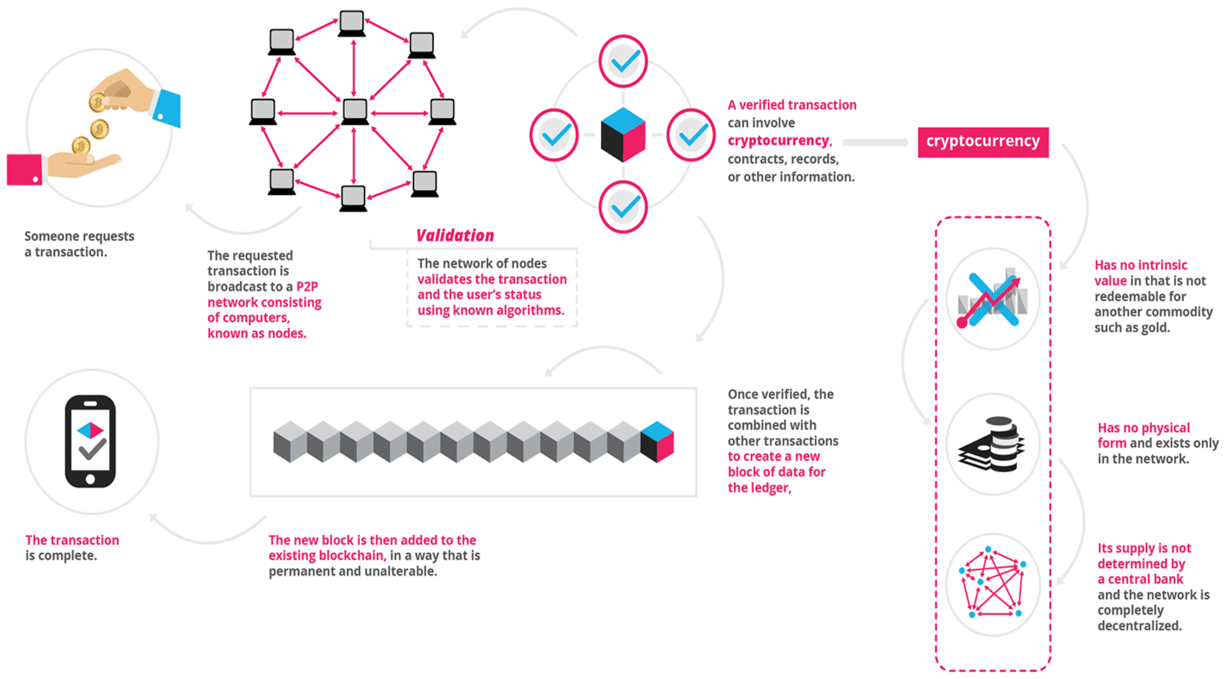

Someone requests a transaction.
The requested transaction is broadcast to a P2P network consisting of computers, known as nodes.
Validation
The network of nodes validates the transaction and the user's status using known algorithms.
A verified transaction can involve cryptocurrency, contracts, records, or other information.
cryptocurrency
Has no intrinsic value in that is not redeemable for another commodity such as gold.
Has no physical form and exists only in the network.
Its supply is not determined by a central bank and the network is completely decentralized.
Once verified, the transaction is combined with other transactions to create a new block of data for the ledger,
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
The transaction is complete.

Fig 2. Blockchain Infographic

Enhanced Security

By storing data across its network, the blockchain eliminates the risks that come with data being held centrally. This is referred to as “decentralization”.

Its network lacks centralized points of vulnerability that computer hackers can exploit. Today’s internet has security problems that are familiar to everyone. We all rely on the “username/password” system to protect our identity and assets online. Blockchain security methods use encryption technology.

The basis for this are the public and private keys. A public key is a users’ address on the blockchain. Information sent across the network gets recorded as belonging to that address. The private key is like a password that gives its owner access to their data or digital assets. Storing data on the blockchain makes it incorruptible.

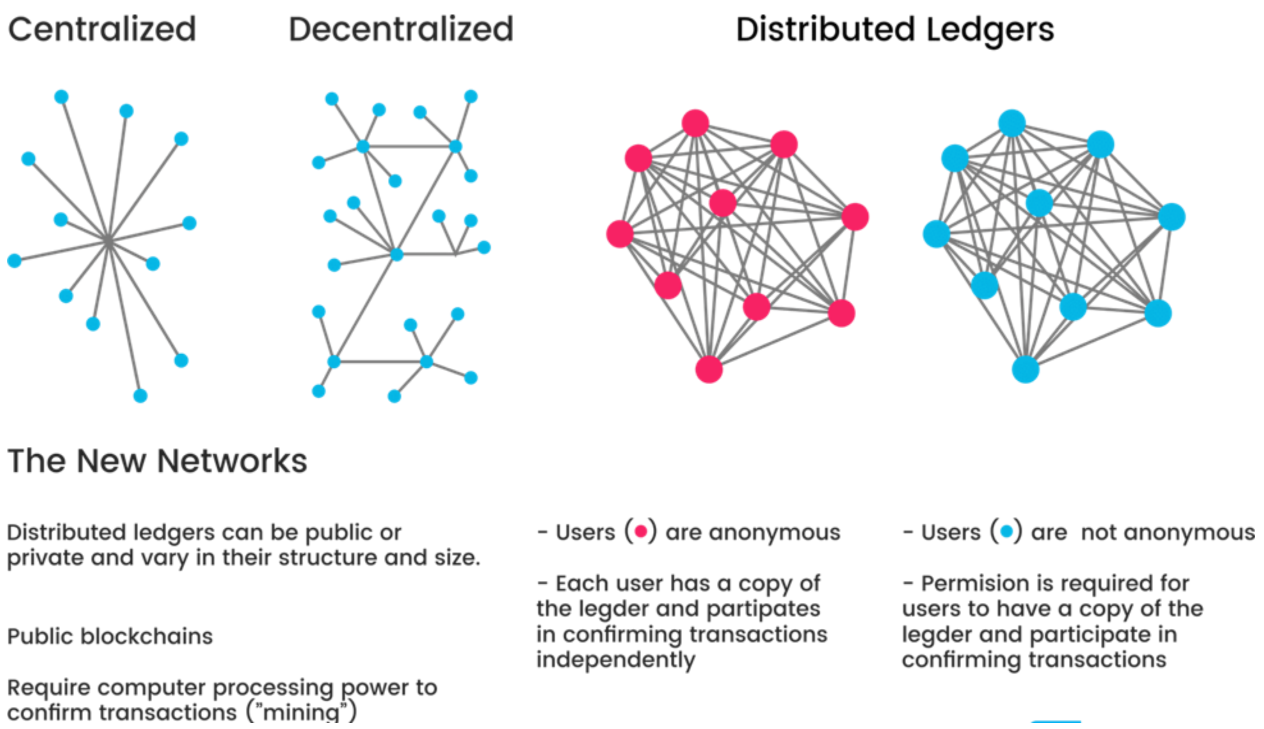

Centralized
Decentralized
Distributed Ledgers
The New Networks
Distributed ledgers can be public or private and vary in their structure and size.
Public blockchains
Require computer processing power to confirm transactions ("mining")
- Users (●) are anonymous
- Each user has a copy of the legder and partipates in confirming transactions independently
- Users (●) are not anonymous
- Permision is required for users to have a copy of the legder and participate in confirming transactions

Fig 3. Enhanced Security Features

<u>Ethereum</u>

Ethereum is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications. Ethereum is a distributed public blockchain network. The Ethereum blockchain focuses on running the programming code of any decentralized application.

While all blockchains have the ability to process code, most are severely limited. Ethereum is different. Rather than giving a set of limited operations, Ethereum allows developers to create whatever operations they want. This means developers can build thousands of different applications.

Smart Contract

Smart contract is a computer code that can facilitate the exchange of money, content, property, shares, or anything of value. When running on the blockchain a smart contract becomes like a self-operating computer program that automatically executes when specific conditions are met. Because smart contracts run on the blockchain, they run exactly as programmed without any possibility of censorship, downtime, fraud or third party interference.

Ethereum’s core innovation, the Ethereum Virtual Machine (EVM) is a Turing complete software that runs on the Ethereum network. It enables anyone to run any program, regardless of the programming language given enough time and memory. The Ethereum Virtual Machine makes the process of creating blockchain applications much easier and efficient than ever before. Instead of having to build an entirely original blockchain for each new application, Ethereum enables the development of potentially thousands of different applications all on one platform.

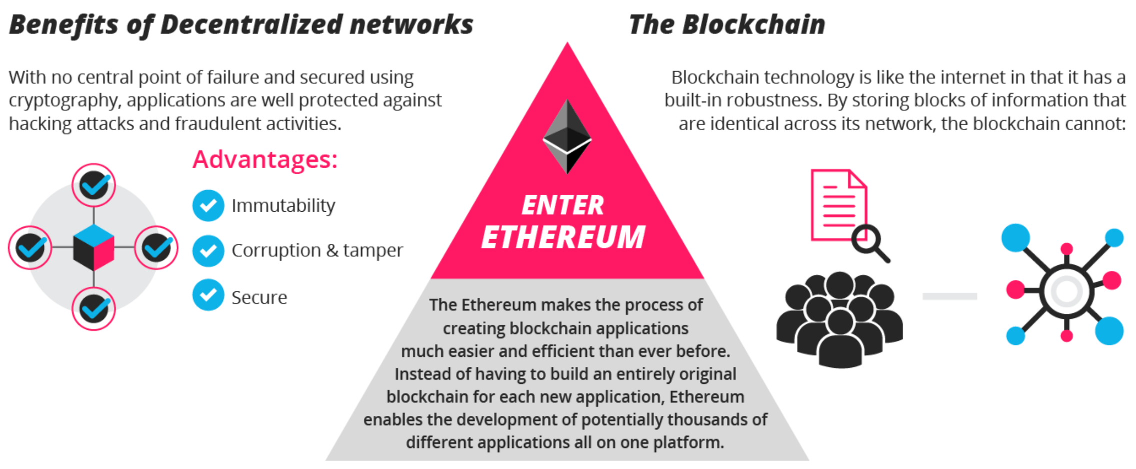

Benefits of Decentralized networks
With no central point of failure and secured using cryptography, applications are well protected against hacking attacks and fraudulent activities.
Advantages:
Immutability
Corruption & tamper
Secure
ENTER ETHEREUM
The Ethereum makes the process of creating blockchain applications much easier and efficient than ever before. Instead of having to build an entirely original blockchain for each new application, Ethereum enables the development of potentially thousands of different applications all on one platform.
The Blockchain
Blockchain technology is like the internet in that it has a built-in robustness. By storing blocks of information that are identical across its network, the blockchain cannot:

Fig 4. Ethereum Infographic

Solidity

Solidity is a contract-oriented programming language for writing smart contracts. It is used for implementing smart contracts on various blockchain platforms. Solidity is a statically-typed programming language designed for developing smart contracts that run on the EVM. Solidity is compiled to bytecode that is executable on the EVM. With Solidity, developers are able to write applications that implement self-enforcing business logic embodied in smart contracts, leaving a non-repudiable and authoritative record of transactions. The language itself was influenced by JavaScript, C++, Python, and PowerShell.

Fig 5. Solidity Logo

DESIGN & IMPLEMENTATION

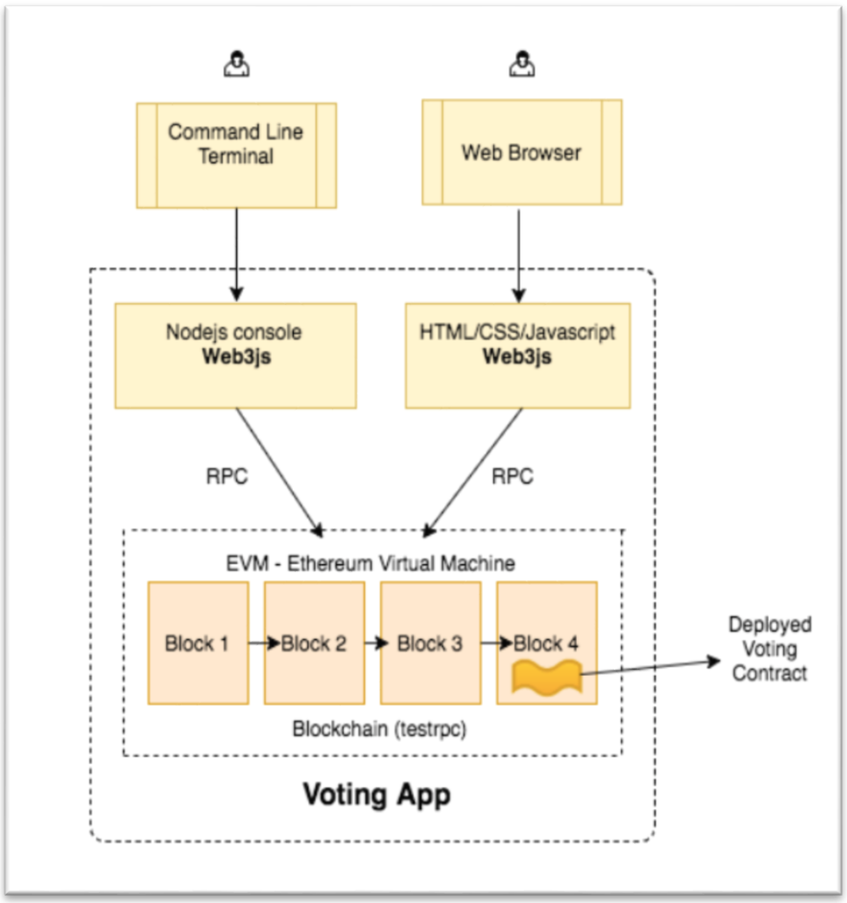
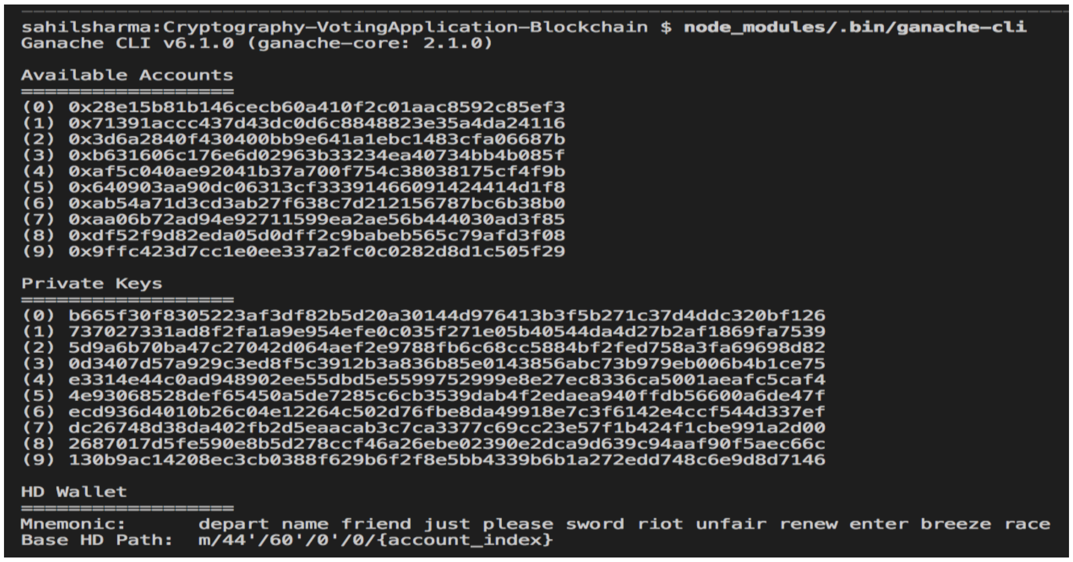


Fig 6. Working of Voting Application

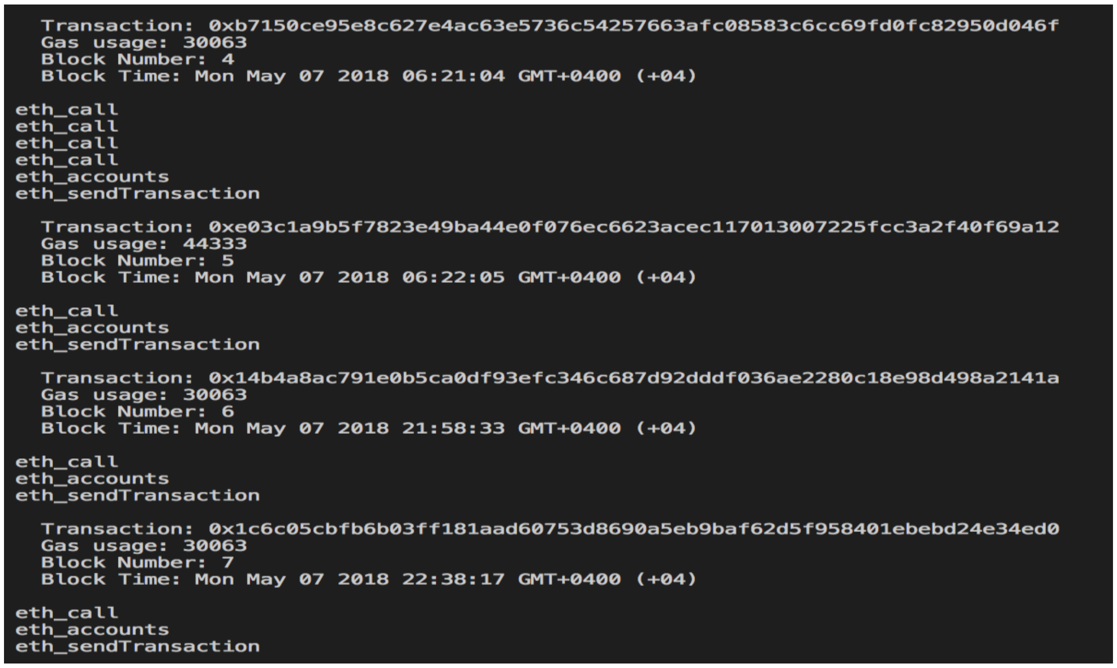
INITIAL SETUP -

The in-memory blockchain called ganache-cli creates 10 test accounts to play with automatically. Here are the 10 public and private keys-



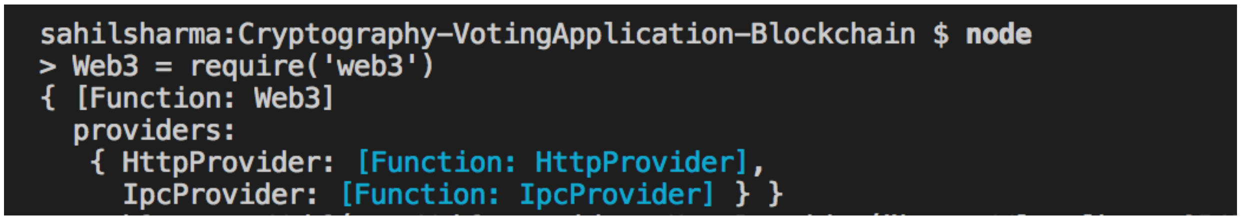
```
sahilsharma:Cryptography-VotingApplication-Blockchain $ node_modules/.bin/ganache-cli
Ganache CLI v6.1.0 (ganache-core: 2.1.0)

Available Accounts
==================
(0) 0x28e15b81b146cecb60a410f2c01aac8592c85ef3
(1) 0x71391accc437d43dc0d6c8848823e35a4da24116
(2) 0x3d6a2840f430400bb9e641a1ebc1483cfa06687b
(3) 0xb631606c176e6d02963b33234ea40734bb4b085f
(4) 0xaf5c040ae92041b37a700f754c38038175cf4f9b
(5) 0x640903aa90dc06313cf33391466091424414d1f8
(6) 0xab54a71d3cd3ab27f638c7d212156787bc6b38b0
(7) 0xaa06b72ad94e92711599ea2ae56b444030ad3f85
(8) 0xdf52f9d82eda05d0dff2c9babeb565c79afd3f08
(9) 0x9ffc423d7cc1e0ee337a2fc0c0282d8d1c505f29

Private Keys
==================
(0) b665f30f8305223af3df82b5d20a30144d976413b3f5b271c37d4ddc320bf126
(1) 737027331ad8f2fa1a9e954efe0c035f271e05b40544da4d27b2af1869fa7539
(2) 5d9a6b70ba47c27042d064aef2e9788fb6c68cc5884bf2fed758a3fa69698d82
(3) 0d3407d57a929c3ed8f5c3912b3a836b85e0143856abc73b979eb006b4b1ce75
(4) e3314e44c0ad948902ee55dbd5e5599752999e8e27ec8336ca5001aeafc5caf4
(5) 4e93068528def65450a5de7285c6cb3539dab4f2edaea940ffdb56600a6de47f
(6) ecd936d4010b26c04e12264c502d76fbe8da49918e7c3f6142e4ccf544d337ef
(7) dc26748d38da402fb2d5eaacab3c7ca3377c69cc23e57f1b424f1cbe991a2d00
(8) 2687017d5fe590e8b5d278ccf46a26ebe02390e2dca9d639c94aaf90f5aec66c
(9) 130b9ac14208ec3cb0388f629b6f2f8e5bb4339b6b1a272edd748c6e9d8d7146

HD Wallet
==================
Mnemonic:      depart name friend just please sword riot unfair renew enter breeze race
Base HD Path:  m/44'/60'/0'/0/{account_index}
```

Every vote submitted by the voter creates a block with a unique transaction ID, Gas usage, block number and the block time.



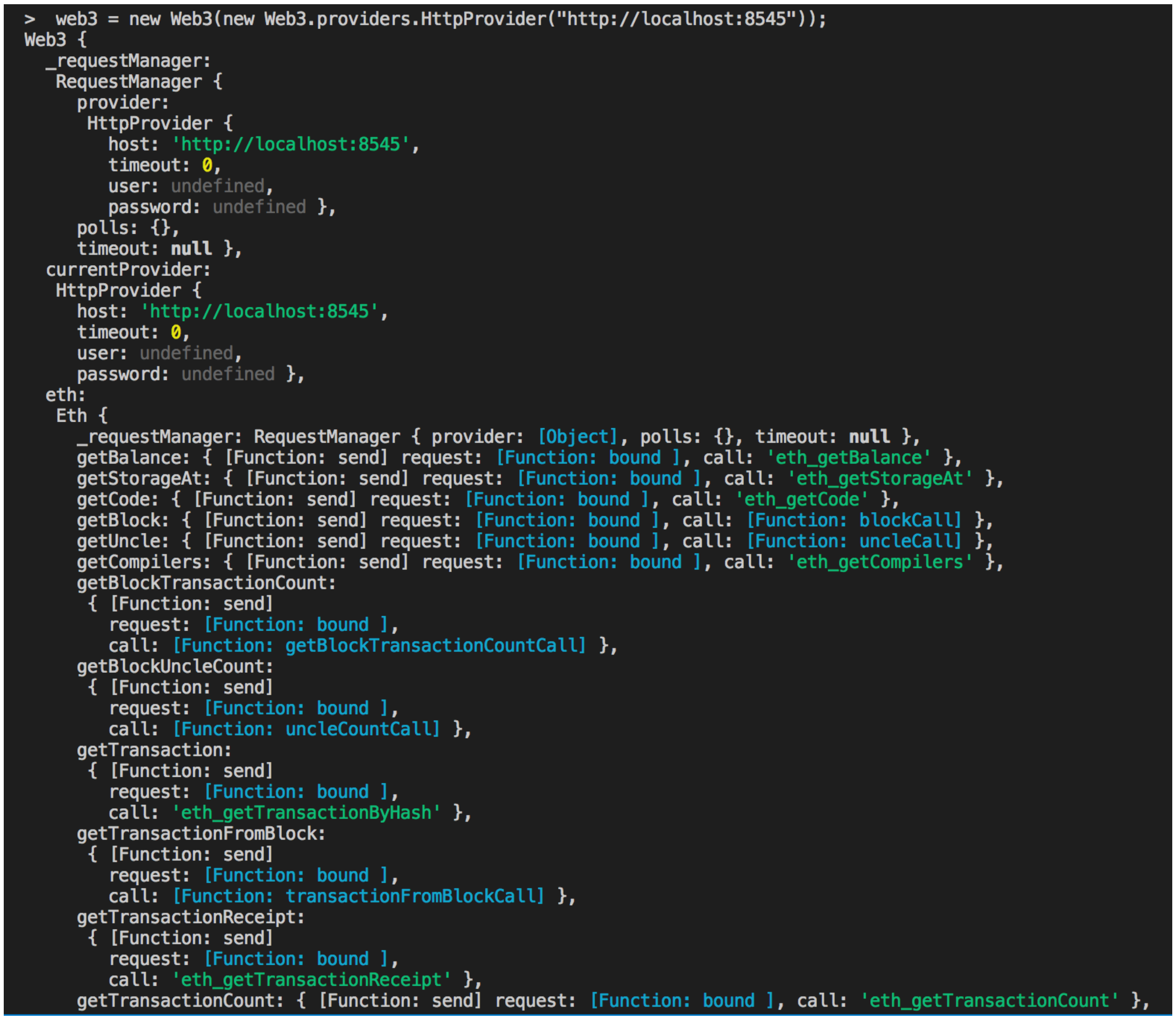
```
  Transaction: 0xb7150ce95e8c627e4ac63e5736c54257663afc08583c6cc69fd0fc82950d046f
  Gas usage: 30063
  Block Number: 4
  Block Time: Mon May 07 2018 06:21:04 GMT+0400 (+04)

eth_call
eth_call
eth_call
eth_call
eth_accounts
eth_sendTransaction

  Transaction: 0xe03c1a9b5f7823e49ba44e0f076ec6623acec117013007225fcc3a2f40f69a12
  Gas usage: 44333
  Block Number: 5
  Block Time: Mon May 07 2018 06:22:05 GMT+0400 (+04)

eth_call
eth_accounts
eth_sendTransaction

  Transaction: 0x14b4a8ac791e0b5ca0df93efc346c687d92dddf036ae2280c18e98d498a2141a
  Gas usage: 30063
  Block Number: 6
  Block Time: Mon May 07 2018 21:58:33 GMT+0400 (+04)

eth_call
eth_accounts
eth_sendTransaction

  Transaction: 0x1c6c05cbfb6b03ff181aad60753d8690a5eb9baf62d5f958401ebebd24e34ed0
  Gas usage: 30063
  Block Number: 7
  Block Time: Mon May 07 2018 22:38:17 GMT+0400 (+04)

eth_call
eth_accounts
eth_sendTransaction
```

Web3js is a library which lets you interact with the blockchain through RPC which we use to deploy our application and interact with it.



```
sahilsharma:Cryptography-VotingApplication-Blockchain $ node
> Web3 = require('web3')
{ [Function: Web3]
  providers:
   { HttpProvider: [Function: HttpProvider],
     IpcProvider: [Function: IpcProvider] } }
```

Using the localhost: 8545 to host our web application-



```
>  web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
Web3 {
  _requestManager:
   RequestManager {
     provider:
      HttpProvider {
        host: 'http://localhost:8545',
        timeout: 0,
        user: undefined,
        password: undefined },
     polls: {},
     timeout: null },
  currentProvider:
   HttpProvider {
     host: 'http://localhost:8545',
     timeout: 0,
     user: undefined,
     password: undefined },
  eth:
   Eth {
     _requestManager: RequestManager { provider: [Object], polls: {}, timeout: null },
     getBalance: { [Function: send] request: [Function: bound ], call: 'eth_getBalance' },
     getStorageAt: { [Function: send] request: [Function: bound ], call: 'eth_getStorageAt' },
     getCode: { [Function: send] request: [Function: bound ], call: 'eth_getCode' },
     getBlock: { [Function: send] request: [Function: bound ], call: [Function: blockCall] },
     getUncle: { [Function: send] request: [Function: bound ], call: [Function: uncleCall] },
     getCompilers: { [Function: send] request: [Function: bound ], call: 'eth_getCompilers' },
     getBlockTransactionCount:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: getBlockTransactionCountCall] },
     getBlockUncleCount:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: uncleCountCall] },
     getTransaction:
      { [Function: send]
        request: [Function: bound ],
        call: 'eth_getTransactionByHash' },
     getTransactionFromBlock:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: transactionFromBlockCall] },
     getTransactionReceipt:
      { [Function: send]
        request: [Function: bound ],
        call: 'eth_getTransactionReceipt' },
     getTransactionCount: { [Function: send] request: [Function: bound ], call: 'eth_getTransactionCount' },
```

WORKFLOW -

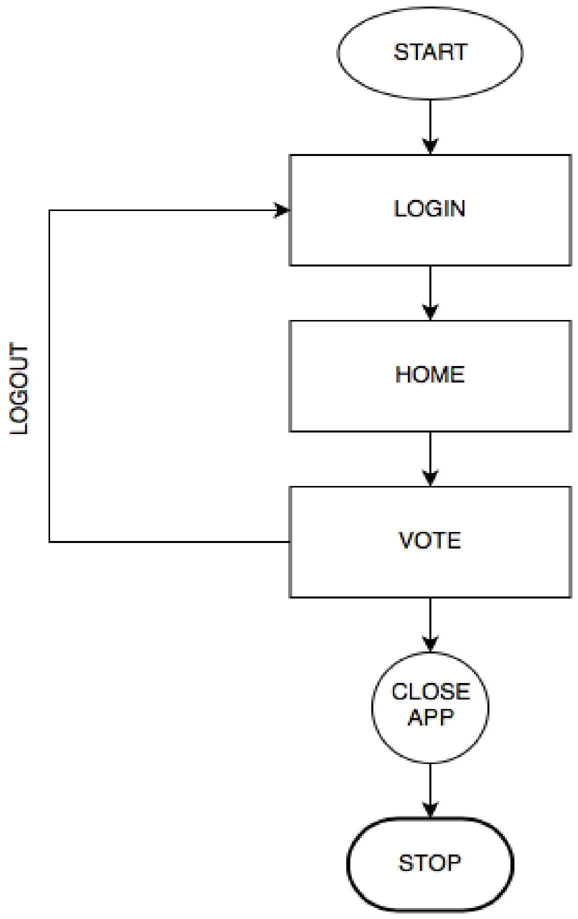


Fig 7. Workflow

RESULTS

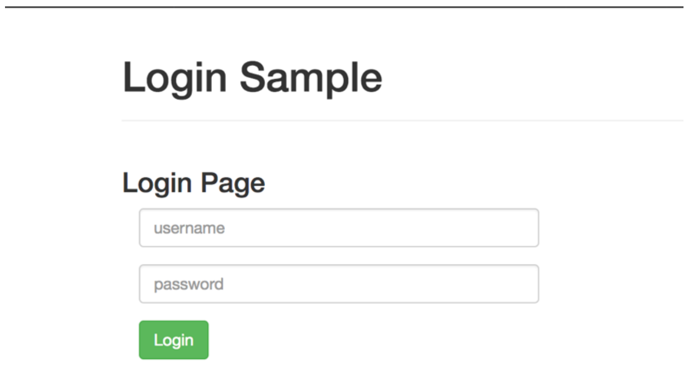
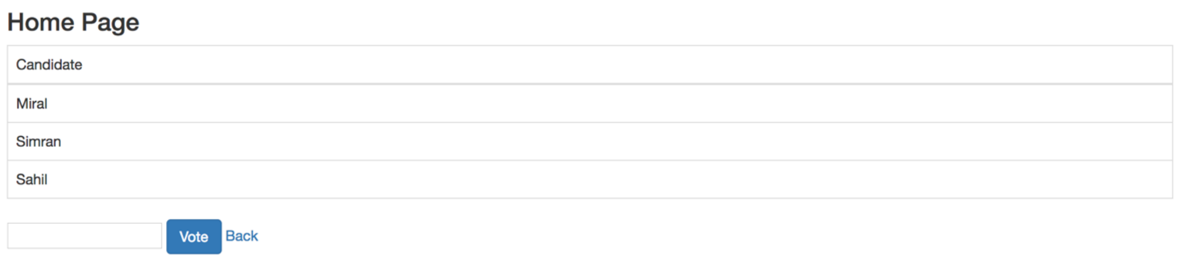


Fig 8. Login page

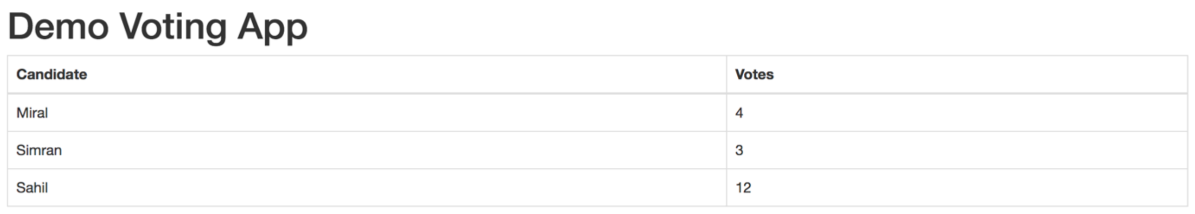
The login page enables the users to login with their username and password in order to cast their vote and facilitates authentication



Candidate
Miral
Simran
Sahil

Fig 9. Homepage

The homepage of the Voting Application displays the names of the candidates to enable the voters to vote accordingly.
The voters type in the name of the candidate that they want to cast their vote for. Once a voter votes for a candidate, no changes can be made.



Demo Voting App

Candidate	Votes
Miral	4
Simran	3
Sahil	12

Fig 10. Results page

The final results are displayed as above and enables to see the vote counts of all candidates.

BENEFITS

- Immutability – A third party cannot make any changes to data.
- Corruption & tamper proof – App is based on a network formed around the principle of consensus, making censorship impossible.
- Secure – With no central point of failure and secured using cryptography, application is well protected against hacking attacks and fraudulent activities.
- Zero downtime – App never goes down and can never be switched off.
- Anonymity – Voter anonymity is guaranteed by transparent crypto algorithms
- Transparent technology – One of the main characteristics of blockchain technology is its transparency. The crypto algorithms that we use on top of it are merely mathematics.
- Convenience – The online voting application enables any voter to vote online which makes it convenient for the voters (students, senior citizens, out of town voters) to cast their vote without wasting their time to go to a polling station. This also encourages more voters to vote
- Less routine – The application ensures that a lot of time and resources- human and material are saved. No wasting time printing and distributing ballots, counting votes.
- Accuracy – Every vote is counted and cannot be changed, duplicated or removed.

LIMITATIONS

- We assume that voters will use a secure device to cast their vote. Even while our system is secure, hackers have the ability to cast or alter a vote using malicious software already installed on the voter's device.
- One of the drawbacks of our system is the inability to change a vote in case of a user mistake. The user will be able to cast its vote only once.
- The user interface engineering of our voting application is basic and only enables voters to cast their votes using a link of the localhost
- Limited candidates in the system take very less processing time. When we deal with a real blockchain, every write to the blockchain (voteForCandidate) will take a few seconds (The miners have to include your transaction in a block and the block to the blockchain). Once we add in more candidates and users, we would require servers with high processing time
- Our voting app doesn't ensure authentication of the voters- anyone with the link can vote whether they are eligible or not and can also vote any number of times.
- Another limitation of our design is its case sensitivity- A voters vote will only be counted when they type in the candidates name that they want to vote for keeping in mind the case sensitiveness.
- With decentralized systems, and especially with our online Voting Blockchain-based system, a problem of concusses may occur. This happens when different voters cast their votes at approximately the same time.

CONCLUSION

We have proposed an online voting system based on the Blockchain technology. The system is decentralized and does not rely on trust. Any voter will have the ability to vote using any device connected to the Internet. The Blockchain will be publicly verifiable and distributed in a way that no one will be able to corrupt it. It is secure, immutable and extremely convenient.

FUTURE SCOPE

- Write a new smart contract function that counts the votes for BOTH candidates at once. Currently, we have to make two separate calls for two candidates, requiring the contract to loop through all the Users twice.
- Allow new Candidates to be added. This means adding a new form to add Candidates, but also changing up a little on how we display and vote for candidates in the frontend.
- Better UIE to enhance the voting experience for the voters in order to make it more convenient and user friendly. Easier navigation and ensuring a voter can only vote once.
- Servers with high processing time. While dealing with a real blockchain, for addition of more voters and candidates, every write to the blockchain (voteForCandidate) will take a few seconds (The miners have to include your transaction in a block and the block to the blockchain).
- Authentication of voter using QR code scanner. In order to verify the identity of a voter and their eligibility, enabling a voter to login by scanning the QR code on their Aadhar card and facilitate voting.
- Real time voter turnout. Monitor voter participation and enabling a voter to know which candidate is in lead after they have casted their vote.
- Developing a ‘Live Talk’ feature for the voters to discuss their views and debate upon the candidates agendas and goals.

LITERATURE SURVEY

Existing electronic voting systems all suffer from a serious design flaw: They are centralized by design, meaning there is a single supplier that controls the code base, the database and the system outputs while also supplying the monitoring tools to verify the result. The lack of an independently verifiable system means that, once voters mark their ballot choice, they must place their trust in the organization, that their vote is recorded and counted as intended. The lack of an independently verifiable output, makes it difficult for these centralized systems to acquire the trustworthiness required by voters, which potentially limits voter participation, or cast doubt upon the published output of an election.

Despite the digitalization of many aspects of modern life, elections are still being largely conducted offline, on paper although the use of Electronic Voting Machines has been steadily growing over recent years. Paper ballots are the traditional tool for voting and are typically marked by a human (voter) and then tallied by a machine. While costing less than most electronic systems to run they rely on physical security and trust in polling stations to not manipulate and to properly handle them. Postal votes also utilize paper ballots and are used to allow voters to not have to physically attend a location in order to vote. These also suffer from the same flaws as traditional paper ballots while increasing the opportunities for attack during their traversal through the postage system.

REFERENCES

1. S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2009, [online] Available: http://bitcoin.org/bitcoin.pdf.
2. J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, E. W. Felten, "Research Perspectives and Challenges for Bitcoin and Cryptocurrencies", S&P, 2015.
3. Saravanan Raju, Sai Boddepalli, Suraj Gampa, Qiben Yan, Jitender S. Deogun, "Identity management using blockchain for cognitive cellular networks", Communications (ICC) 2017 IEEE International Conference on, pp. 1-6, 2017, ISSN 1938-1883.
4. G. Wood, Ethereum: A secure decentralized transaction ledger, [online] Available: http://gavwood.com/paper.pdf.