

Indian Institute of Information Technology, Allahabad

Department of Information Technology (B.Tech)

6th Semester



Astra: Secure Voting System Using Blockchain

Guide:

Dr. Ranjana Vyas

Dept. of Information Technology

Soumyadeep Basu (IIT2018001)

Kumar Utkarsh (IIT2018007)

Jeet Mandal (IIT2018039)

Rishabh Gupta (IIT2018085)

Ankur Kumar (IIT2018086)

Contents

1. Candidate Declaration.....	3
2. Certificate.....	3
3. Abstract.....	4
4. Motivation.....	5
5. Introduction.....	6
5.1. Properties of Blockchain.....	6
5.2. Working of Blockchain.....	7
5.3. Ganache,Truffle	9
5.4. Smart Contracts.....	9
5.5. Web3.....	9
5.6. IPFS(Interplanetary File System).....	10
6. Problem Statement.....	12
7. Implementation.....	13
8. Literature Review.....	18
9. Methodology	19
10. Software requirements.....	20
11. Results.....	20
12. Scope of Improvement.....	28
13. Conclusion.....	28
14. References.....	29

1. Candidate declaration

We hereby declare that the work presented in this project entitled “**ASTRA : Secure Voting System Using Blockchain**”, submitted as mini-project report of 6th Semester report of Bachelor of Technology (IT) at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from Jan. 2021 to May 2021 under the guidance of **Dr. Ranjhana Vyas**.

Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: IIIT Allahabad

Date: 11th May 2021

Soumyadeep Basu (IIT 2018 001)

Kumar Utkarsh (IIT 2018 007)

Jeet Mandal (IIT 2018 039)

Rishabh Gupta (IIT 2018 085)

Ankur Kumar (IIT 2018 086)

2. Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Place: IIIT Allahabad

Date: 11th May 2021

Dr. Ranjana Vyas

Assistant Professor

Department of Information Technology

IIIT Allahabad

3. Abstract

Electronic Voting Machines (EVMs) provide a level of transparency and flexibility which is difficult to match in an online voting system. However , with the advent of Blockchain technology properties such as transparency, decentralization, nonrepudiation and irreversibility have become achievable at the same time in an easy manner. Developing a digital voting system with the use of Blockchain technology would allow the deployment of a more secure voting system without the need for any trusted third party along with fulfilling the system requirements.

Using Blockchain not only provides security, but it also provides an easier medium of casting their vote. The computations made on a blockchain are untraceable by the outside world and no one can change the information which is present in the blockchain blocks. In this paper we have discussed the working of a blockchain and how new blocks are mined and added to the blockchain. We then discussed the incorporation of blockchain into a decentralised application. This paper also presents the details of a proposed e-voting system which would tackle the limitations of the current systems.

Keywords - electronic voting, blockchain, verifiable voting, Hashing, Distributed ledger, Proof of work.

4. Motivation

As we all know all the election and voting procedures are done offline. In other words, the voters have to assemble at a particular election booth and cast their vote. In this offline voting system, we make use of Electronic Voting Machines (EVMs) to record the response of the voters. In this offline voting system, the voters have to come to the voting booth and verify their voting identification using their voting ID. Moreover, there are a large number of people waiting in queues to cast their vote in the voting booth.

We all know how COVID-19 has impacted the daily life of the people all around the globe. People have to wear masks and gloves whenever they go outside to get their basic commodities. All the public events have been either postponed or cancelled due to this difficult time of COVID-19. People are afraid to go to public places due to the fear of contraction with the disease. And in this mess, holding the elections offline is not the best idea.

To tackle this problem of elections to be held amid the COVID-19, we propose a digital platform where the people can cast their vote without going to the election booth. This will not only deal with the case of social distancing amid the pandemic, but it will also tackle the case of the waiting in the long queues outside the voting booth. This platform will make the election and voting process more simple and easy to deal with.

5. Introduction

Elections are the figurative pillars on which a democratic system emerges from. Fair elections are the fundamental requirement of any form of government to qualify as a democracy, yet it is undoubtedly one of the biggest challenges in a democracy. It has been the cause of many conflicts due to reasons such as election interference by foreign powers, voter en-franchisement and technical failures which raise a question on the integrity of the entire voting process.

In recent times electronic voting systems have gained popularity over the previously used paper based ballot boxes due to the numerous advantages they offer. It is environment friendly, provides real-time counting and processing, is less prone to errors and provides anonymity. However, Electronic voting systems are not impervious to technical faults. With the advent of Blockchain technology there is a much better promise to build a highly resilient voting system. Blockchain provides transparency, decentralization, irreversibility and nonrepudiation even in the absence of a trusted third party.

Blockchain is a distributed, immutable and incontrovertible public ledger with strong cryptographic functions. This enables applications to use these features for a secure and resilient system. Blockchain uses a data structure which maintains and shares all the transactions being executed in a particular order from its genesis block. It allows every new user to connect to the blockchain network, send/receive transactions, verify other's transactions and create new blocks. To prevent tampering, each block is assigned a hash which remains valid as long as the data in the block remains unchanged. Any change in the data alters the hash immediately.

5.1 Properties of Blockchain

There are basically three main properties that a blockchain should be able to satisfy without any failure. These three properties are discussed below:

- 1. Immutability:** Before adding a new block to the ledger, the previous version of the ledger is referenced for conferring the integrity of the new block. This prevents any unsolicited tampering to the blockchain. Any tampering with the content which is written in the block will change the hash associated with the block and thus all the associated blocks will be lost. Those who validate the transactions and add them in blocks are called miners. These miners validate the transactions by completing a computation intensive job.

2. **Verifiability:** There is no single point of failure and all nodes maintain a consensus version of the ledger ensuring verifiability . Ledger is the record of the transactions completed and since it is visible to everyone, and hence called an open ledger. Everyone who can see this open ledger are the members who are in the blockchain. . As it is a decentralised system, all the nodes verify this transaction and then it adds the transaction in the block.

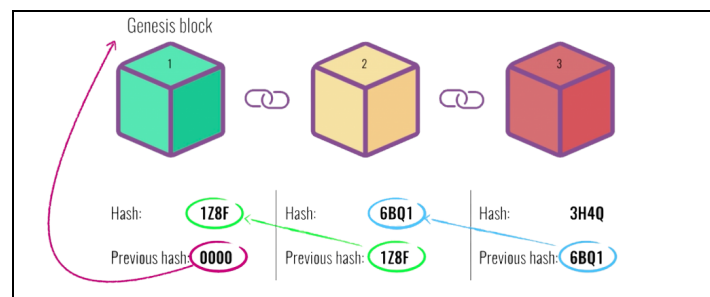
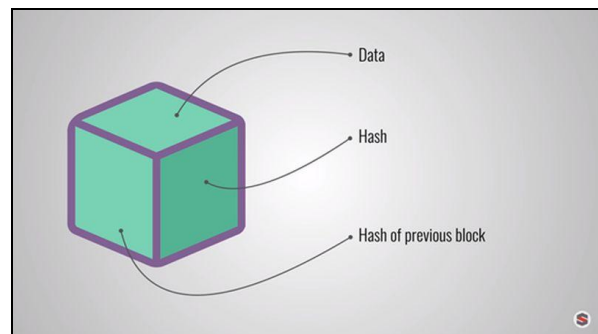
3. **Distributed consensus:** A distributed consensus ensures a consensus of data among nodes in a distributed system. Blockchain consensus protocols consist of some specific objectives such as coming to agreement, collaboration, participation and co-operation of each node in the consensus process. Hence the distributed consensus allows the network to reach an agreement that is agreeable to the entire network.

After discussing all the features of the blockchain, we will now see how a blockchain works generally.

5.2 Working of Blockchain

A blockchain is a **distributed ledger** that is open to all. Once some data has been recorded inside the blockchain it becomes a difficult task to change it. So we will now discuss in brief how a blockchain works.

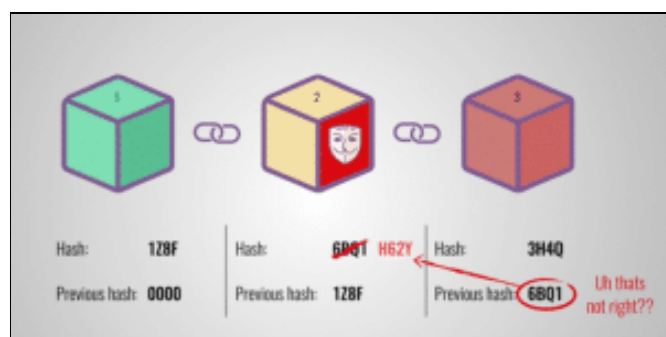
Let's have a look at Blockchain. Each block of blockchain stores data, hash of previous block and hash of current block. The data which is to be stored in blockchain depends on the purpose of blockchain and data is encrypted **cryptographically**. Let us assume the block of bitcoin blockchain stores detailed information about transactions made like sender, receiver and amount of coins. A block also has a hash. It identifies the block and all of its contents and it's always unique, analogy to fingerprint. As soon as a block is created its hash is calculated and any change inside the block will result in a change of hash of the block. So hash plays an important



role to detect any change in blocks and thus helps in achieving block **immutability**. Third element inside block is hash of previous block and this effectively creates change of block and its technique that makes the blockchain so secure.

Let's take an example of a blockchain, each block has a hash of itself and a hash of the previous block. So block number 3 points to block number 2 which in turn points to block number 1. Now the first blockchain is a bit special as it can't point to the previous block because it's the first block and we call this block the genesis block. Now suppose we try to tamper the data stored in the second block. This causes the hash value of block 2 to change as well. In turn it will make block 3 and all subsequent blocks invalid because they are no longer connected to the main blockchain. Thus changing a single block will make all subsequent blocks invalid.

However using hash is not enough to avoid tampering as computers these days are very fast as they can calculate millions of hash per second thus effectively can temper with the blocks and can recalculate the hash of all subsequent blocks consequently making the blockchain valid again.



To mitigate this, blockchain has a mechanism called **proof-of-work(PoW)** that requires a node to solve a computationally expensive problem before it can suggest a new block. The first one to solve this task broadcasts this information to all the other nodes who in turn verify this using the value of the block and the hash values in the blockchain. All this helps in slowing down the creation of new blocks. In the case of blockchain it takes about 10 minutes to create required proof of work to add a new block to the blockchain. This mechanism makes it very hard to tamper with the blocks because if we tamper with 1 block we will need to recalculate the PoW for all the subsequent blocks. So security of blockchain comes from creative use of hashing and PoW mechanism. Further there is one additional feature by which blockchain ensures its security and that's by being distributed. Instead of using a centralized entity to manage the chain, blockchain uses a **peer-to-peer** network where anyone is allowed to join. When someone joins the network he gets a full copy of blockchain and thus can verify everything is still in order. When a new block is to be added to blockchain that block is sent to everyone in blockchain and every node verifies the validity of data of blockchain and if everything sounds fine, only then the block is added to their blockchain. All the blocks agree to **consensus** i.e. they agree about which blocks are valid and which are not. Tampered blocks are rejected by other nodes from

the network. And thus to change a block on blockchain we need to recalculate the hash and PoW for all the subsequent blocks and take control of more than **51%** of peer2peer network, which is computationally impossible.

5.3. Ganache, Truffle and Smart Contracts.

Ganache is a personal blockchain for rapid ethereum distributed application development. It allows the development, deployment and testing of dApps in a safe and deterministic environment. Ganache allows us to compile and migrate to test the changes that take place when smart contracts are deployed. Ganache creates 10 virtual accounts each starting with 100 ether. These virtual accounts are needed to pay the transaction fees when running transactions on blockchain. Additionally 10 private keys are created which are used to sign a transaction which is being written to the blockchain. Ganache can be configured to listen to a particular port but by default it is 8545. Truffle is directed to write to this port when the smart contracts are being tested. Ganache and truffle provide a very good idea of what happens when a transaction is carried out in a smart contract.

Truffle is a tool that makes the deployment and compilation of the smart contract easier while making it a more transparent process. The truffle tool consists of multiple versions of the solidity compiler for writing smart contracts. It also provides us a complete package of tools which assists in the integration of the blockchain with the application. It helps in writing the smart contracts on the local blockchain (with/without metamask). Here we made use of truffle to write the smart contracts on the blockchain and then we connect this local blockchain with the help of different library tools.

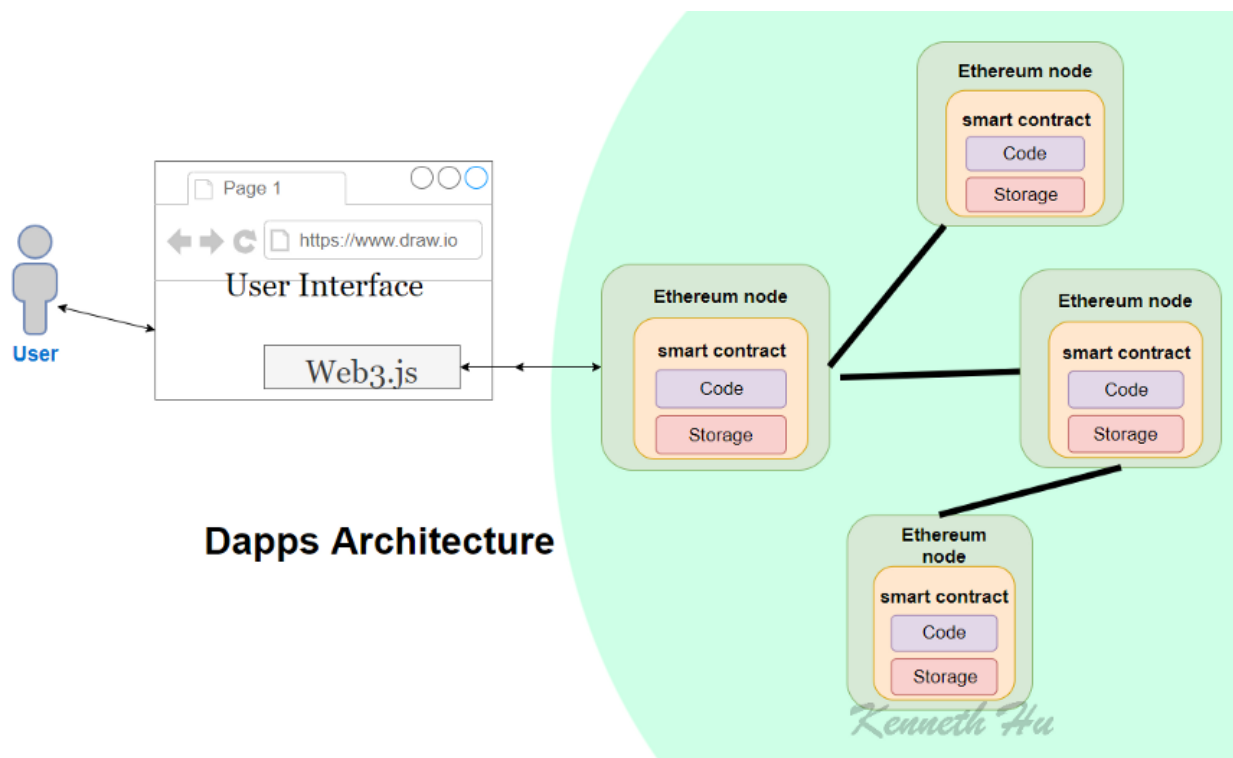
5.4 Smart Contracts

Smart Contracts are computer protocols which are intended to facilitate, verify or enforce the negotiation of contracts. They allow the injection of code into blockchain without any third-party verification. Smart contracts cannot be altered but there are ways to extend or replace parts but this cannot be done without drawing the attention of the network. The logic on which a smart contract is made cannot be distorted i.e there is no room for interpretation. Since there is no third party, there is no risk of manipulation. They reduce administration, save time and offer complete autonomy. However they are irreversible so if the code has bugs, it could lead to some unwanted and irreversible effects.

5.5. Web3

Web3 is a collection of multiple javascript libraries which help us in establishing connections with a local ethereum blockchain network using a HTTP or IPC connection. We make use of the Web3 function to establish HTTP connection with the local ethereum blockchain which is created by Ganache in the local environment. After making successful connections with the local blockchain, we can perform multiple operations within the blockchain network. We can access user accounts, send or receive transactions, interact with smart contracts , retrieve the transaction id, and many other operations.

In simple words, Web3 provides a medium of connection between the De-centralised application and the local ethereum network . This helps in the interaction between the application and the local blockchain. To better understand this, we can illustrate this with a figure.



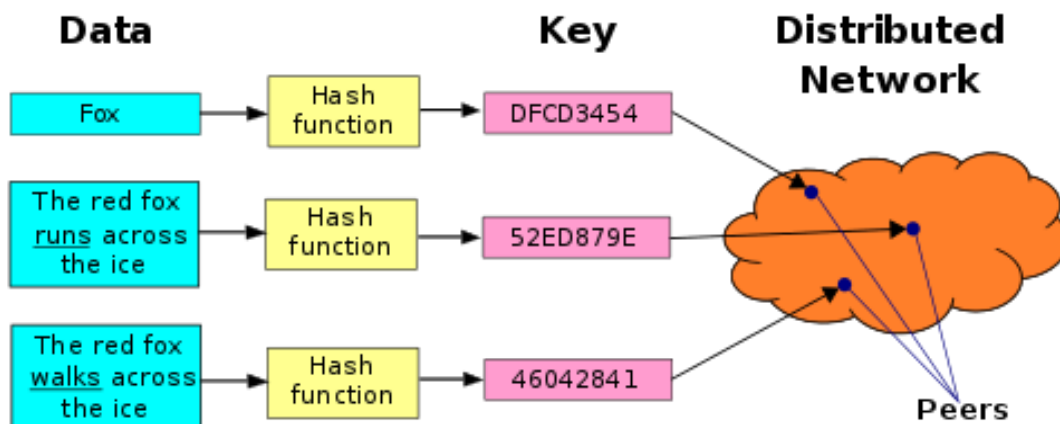
In the above figure, we can see how Web3 helps in establishing a connection with the local/remote blockchain and then performs different operations on that blockchain.

There are different types of Web3. We make use of the *eth* that helps in working with Ethereum related routines. We can retrieve the information of a particular node. We can also check the balance of a user using these subroutines. These subroutines give us a direct method for the interaction between the application and the local blockchain network. To connect to a remote blockchain, we make use of a blockchain extension for browsers known as **metamask**.

5.6. IPFS (Interplanetary File System)

IPFS is basically a file system which can store files and can keep track of their versions over time, It defines how files can move across networks, making it more like a distributed file system. This is like a web-based file system. Now we will discuss how IPFS works.

When a file is uploaded to IPFS, it is split into chunks each containing at most 256 Kb of data and links to other chunks. Each chunk is identified by a cryptographic hash, which is called it's identifier. This identifier is computed from the content of the chunk which guarantees it's uniqueness. These links form a Merkle Directed Acyclic Graph that can be used to reconstruct any file from it's chunks. This Merkle DAG can identify an entire file by just using the root hash. After the Merkle DAG is formed, the node registers itself as a provider in the Distributed Hash Table (DHT). The DHT is a key-value store. When looking for a value using a certain key, a node attempts to find nodes that are closer to the key that itself and requests the information from them and this continues until the key is found. When writing a value to a node, a node determines the number of nodes closer to the key than itself and informs them of the new value for that key. However, the keys in a DHT are only valid for a certain amount of time and need to be updated.



6. Problem Statement

Electronic voting is not impervious to malpractices and technical faults. Many times there are conflicts between the parties contesting the elections because of the voting process. Also, in the current pandemic of COVID-19, electronic voting at voting booths is not feasible due to the fear of corona. To tackle these problems a digital alternative is required which would allow the same reliability and better ease of access than electronic voting.

Moreover, apart from the fear of COVID-19, people have to wait in long queues to cast their votes in the voting booth. There is a long waiting time associated with the offline voting system. We also solve this problem using the proposed application. This will drag the large waiting time of the voters to almost zero and will also ensure that there is consistency in the recording of the votes.

A voting system platform using Blockchain technology should allow for the following features:

- a. allow a method of secure authentication via identity verification.
- b. prevent traceability of votes to the voters.
- c. provide transparency and assurance to each voter about their casted votes.
- d. prevent any third party from tampering with the voting process.
- e. prevent centralization of power and control over the voting process.
- f. only allow eligible individuals to vote.

7. Implementation

The following components have been used primarily in the project:

1. Ganache: This dependency is a personal blockchain, which is a local development blockchain that can be used to act as a public blockchain. Ganache is used to deploy smart contracts and for running tests. Here we have a personal blockchain network running. Ganache provides 10 accounts with 100 Ether to test our smart contracts on local blockchain bases.
2. Node.js: Since we have a private or local blockchain running, we need to configure our environment for developing smart contracts. For that, we will require Node Package Manager or NPM, which includes Node.js
3. Truffle Framework: Truffle Framework is an essential tool for developing Ethereum smart contracts. It uses the Solidity programming language to develop smart contracts. Truffle framework provides following functionalities:
 - Client Side Development
 - Script Runner
 - Network Management
 - Development Console
 - Deployment & Migrations
 - Smart Contract Management
 - Automated Testing

Project Setup

Our proposed system involves a user who wants to upload the data or any file to the IPFS. We are configuring a private local blockchain provided by Ganache and integrate it to a web browser.

Blockchain uses a peer-to-peer network of computers to perform and validate transactions. We will use DApp which is built using a private blockchain via Ganache. The building blocks of the project are as follows:

1. The frontend is made using HTML,CSS and Javascript for an uncomplicated interface.
2. The backend is made up of smart contracts, virtual blockchain network and IPFS for file sharing. Solidity is used to write smart contracts and the virtual blockchain network is run on Ganache.

The system was built to cater to the needs of 3 users:

1. Organizer: responsible for creating elections and accepting candidate requests.
2. Candidate: can create their profiles, send requests to the organizer to be a candidate in an election and observe the election results.
3. Voter: can view elections, vote for the candidates of their choice and observe the election results.

The following parameters were also considered in the system:

1. Eligibility: Only eligible candidates can contest in an election and only eligible voters can cast their votes.
2. Uniqueness: Once a voter has cast their vote, they cannot do so again.
3. Privacy: Nobody can see the votes which were casted by an individual.
4. Accuracy: System ensures that the votes are deposited to the intended candidate only.
5. Efficiency: The vote counting is simultaneous and provides a seamless and easy experience.

A smart contract is a list of functions which is going to be triggered whenever certain preconditions are satisfied. Smart contracts were written in solidity to provide support for the above parameters. The following smart contracts were written:

1. Election.Sol: The entire voting process code is written in this smart contract. This contract also verifies whether the voter is valid or not and keeps count of the votes. It acts as a virtual ballot.In this smart contract mapping was done to store hash of candidate, voter, organiser, elections personnel data. It contains functions which contain the entire logic of the voting.

2. Migration.sol: This is a secondary smart contract which keeps track of all the migrations that occur each time the blockchain is restarted and reset after using commands like truffle migrate --reset. After this command it initializes the count to 0.

The smart contract that we implemented in our local blockchain are as shown below:

Function for Organizers

1. setOrganizerCredentials
2. setOrganizerPersonal
3. getOrganizerCredentials
4. getOrganizerPersonal
5. getOrganizersCount
6. findOrganizer

Function for voting for the candidate of election

1. vote
2. voteSelectedCandidates

Function for Elections

1. addElection
2. getElection
3. organizerElectionCount
4. getElections
5. electionsCount

Function for selected Candidate of Election

1. addCandidate
2. countElectionCandidates
3. getSelectedCandidates

Functions for Voter

1. addVoterAccount
2. getVoterAccount
3. addVoterToElection
4. getVoterToElection

Functions for checking the Aadhaar Card voted or not

1. aadhaarCount
2. checkAadhaar
3. addAadhaar

Functions for Candidates

1. setCandidatePersonal
2. setCandidateCredentials
3. getCandidateCredentials
4. getCandidatePersonal
5. getCandidatesCount()
6. findCandidate

Function for Election Requests

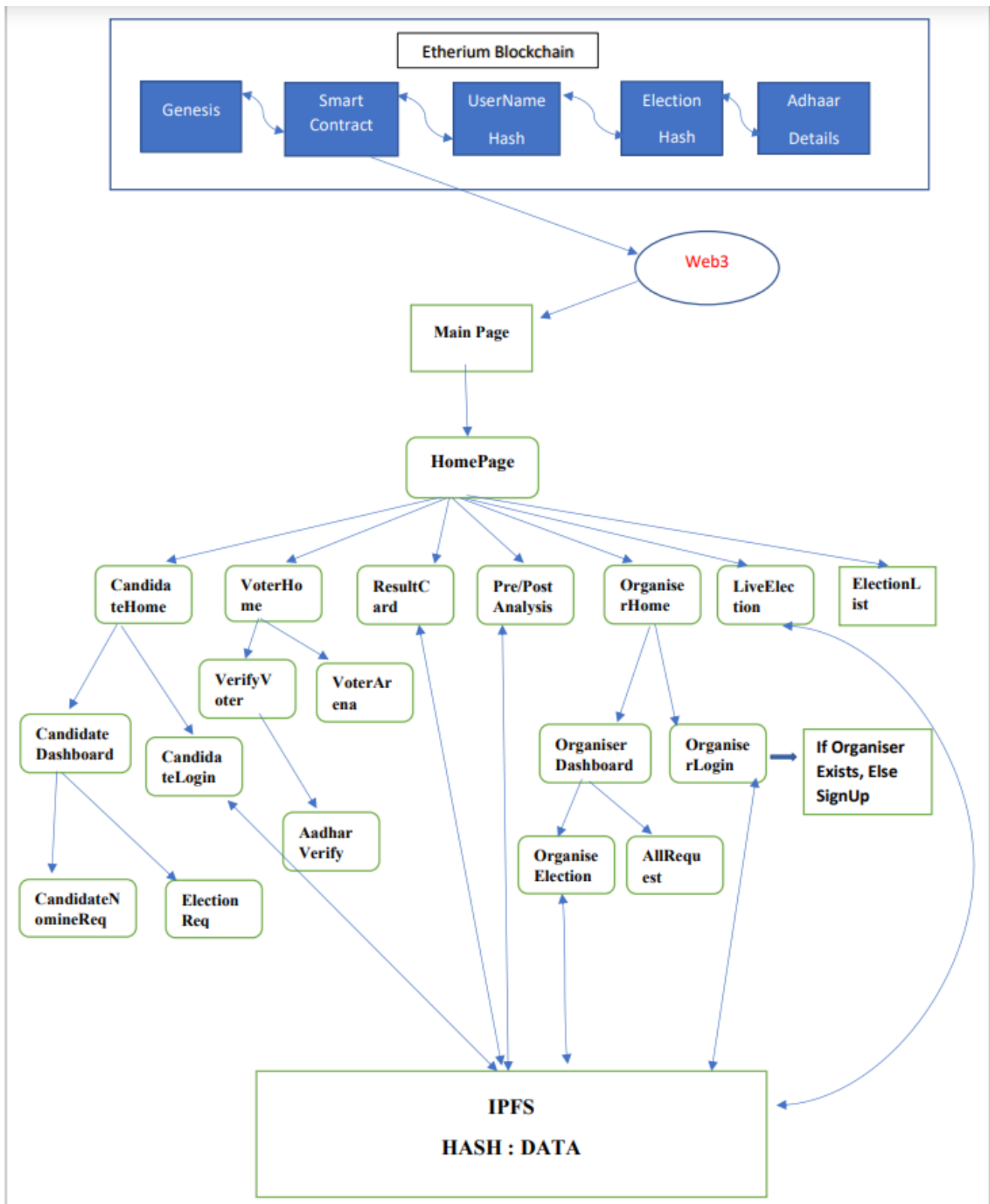
1. setRequest
2. getRequest
3. candidateRequestsCount
4. getAllRequest
5. requestsCount
6. updateRequestStatus

Set up web3: web3.js is a javascript library that allows our client-side application to talk to the blockchain. We configure web3 inside the "initWeb3" function. The deployed instance of the smart contract is fetched inside this function and assigned some values that will allow us to interact with it.

React.js was used for building the client side application. For login/Signup of organizer and candidate, IPFS is used in the backend. Data is sent on IPFS using **ipfsSender** API which returns a hash corresponding to the data. That hash is stored on blockchain to ensure immutability.

Render function: The render function lays out all the content on the page with data from the smart contract. For now, we list the candidates we created inside the smart contract. We do this by looping through each candidate in the mapping, and rendering it to the table. We also fetch the current account that is connected to the blockchain inside this function and display it on the page.

Next, a flowchart depicting the various functionalities and their behavior with each other is shown:



8. Literature Review

There is a wide range of existing papers on blockchain technology being used for the online voting process. Different papers propose different architecture for implementing blockchain in online voting systems. Some cases even provide several extensions for some specific voting scenarios.

Bitcoin is the world's first crypto currency and **"Bitcoin: A Peer-to-Peer Electronic Cash System"** by Satoshi Nakamoto is the paper which explains the network model, the structure, the underlying concepts and working of Bitcoin. It provides a fundamental understanding of the Proof-of-work concept, which helps to make the majority decision by using the longest chain's work as the legitimate one. The implementation of a currency without a trusted third party was the first one shown in this paper.

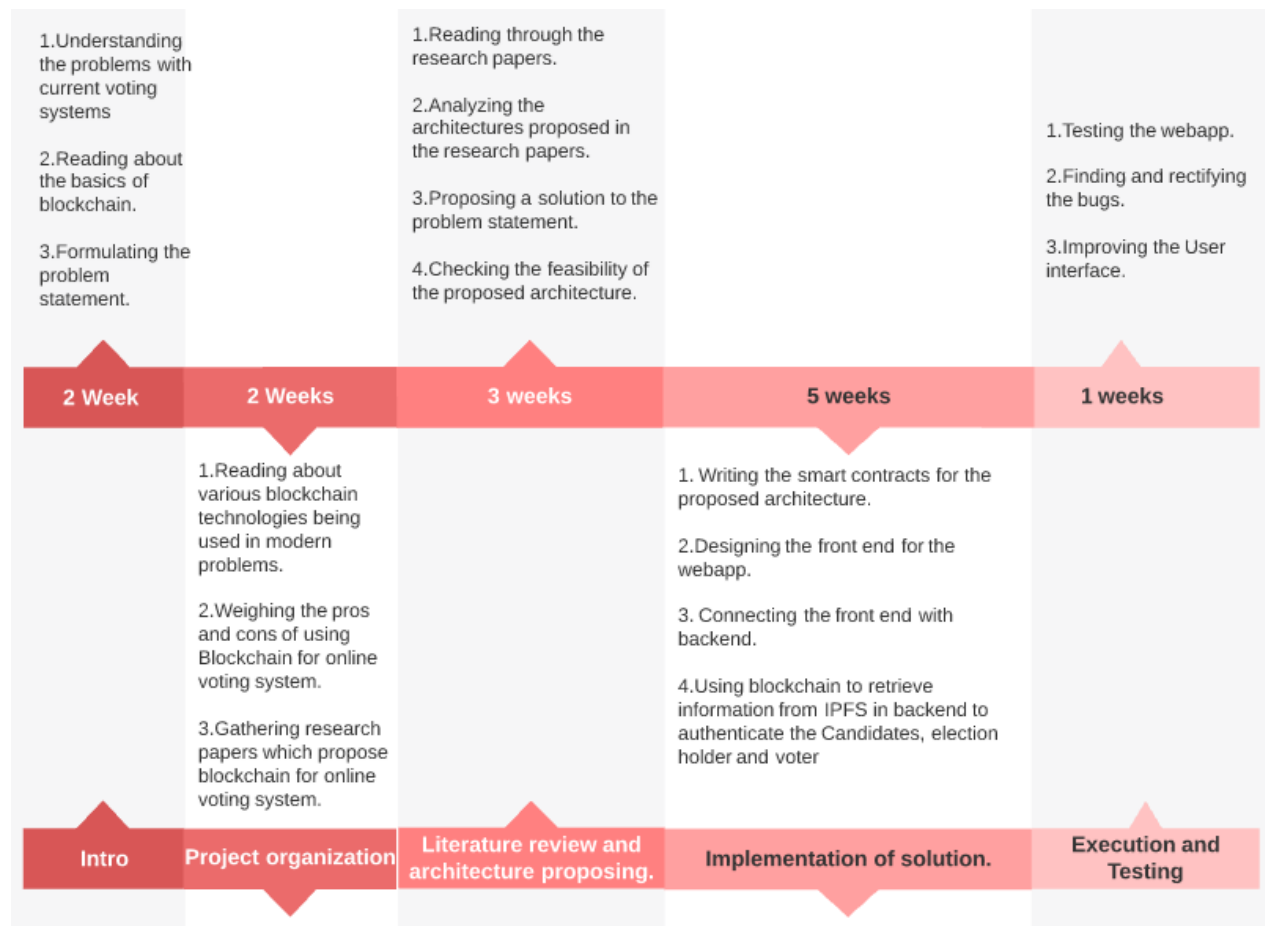
Shuai Wang et.al in their paper, **"An overview of Smart Contracts: Architecture, Applications and Future Trends"** have provided a detailed description on working of smart contracts. They have defined smart contracts as predefined computer protocols intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts have been a topic of scrutiny due to certain technical challenges faced by them like reentrancy vulnerability, Timestamp dependence, lacking of trustworthy data feeds and privacy issues leading to deanonymization attacks and this paper discusses recent advances which have been made to tackle them.

In the paper **"An E-voting Protocol Based on Blockchain"**, Yi Liu and Qi Wang emphasise on the use of blind signatures along with Blockchain to maintain user verifiability and anonymity in the voting process. Blind signatures are used to sign encrypted messages without the need to decrypt them. Different phases involved in a voting process from a digital point of view have been discussed and the main challenge faced in this paper is the functionality tradeoff between transparency and coercion-resistance.

"SecEVS : Secure Electronic Voting System Using Blockchain Technology" puts forward a network model for voting on a university level in which voting at each level is done in a blockchain and the result from those blockchains is joined together to form the blockchain for the upper level. Voter verification is done separately in the pre-voting and voting phase and is used to limit one user to one vote only. Voter confidentiality is safeguarded using SHA-256 hash algorithm and the fundamental logic of merkle hash prevents any tampering with the blocks in the blockchain making the system secure.

Paper titled “**Blockchain Based Voting System Can Better the Way of Elections in India**” proposed a system for the Indian Election System based on the Hyperledger Network. The booth agents at different polling booths act as different nodes. For each phase, the consent of 5 nodes is to be considered. They have described the usage of 3 voting phases, namely the pre-voting, voting and post-voting. Each transaction is endorsed by at least 5 nodes. The final validation of votes is done by 5 nodes which makes the manipulation of votes computationally impossible.

9. Methodology



10. Software requirements

We now discuss which technology we used to make our E-Voting Application

Blockchain: Ganache from Truffle Suite for Virtual Memory Blockchain

Language: JavaScript

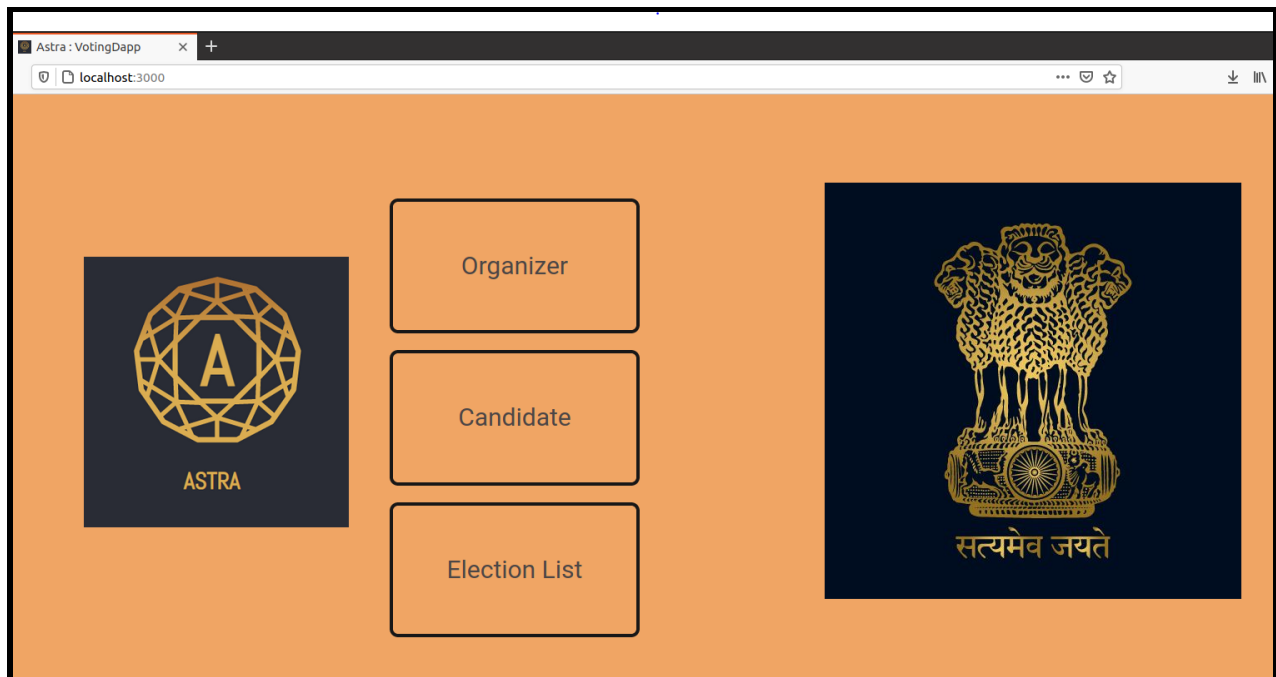
Framework: Node.js and React.js

Essential Packages: Web3 ,IPFS , Truffle Suite, Solidity

IDE: VS Code

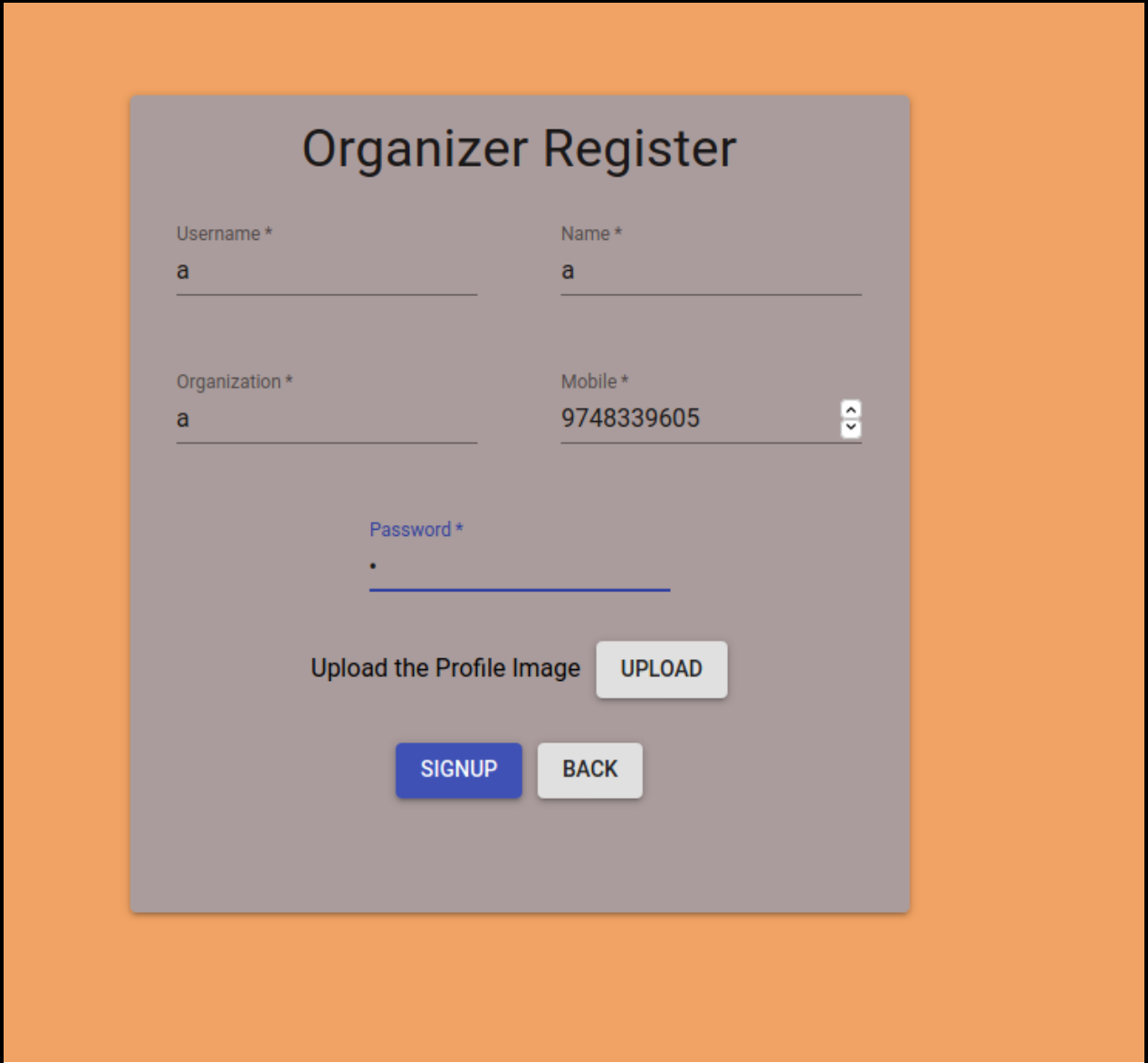
11. Results

- We start at the election homepage



Homepage

- Registering an election organizer 'a'



The screenshot shows a web form titled "Organizer Register" on an orange background. The form is a light gray rectangle with the following fields and controls:

- Username ***: Input field containing the letter 'a'.
- Name ***: Input field containing the letter 'a'.
- Organization ***: Input field containing the letter 'a'.
- Mobile ***: Input field containing the number '9748339605' with up and down arrow icons on the right.
- Password ***: Input field with a single dot visible, indicating a password field.
- Upload the Profile Image**: Text label next to an **UPLOAD** button.
- SIGNUP**: A blue button at the bottom.
- BACK**: A white button with a gray border at the bottom.

Organizer Registration

- Registering a candidate 'b'

The screenshot shows a 'Candidate Register' form with a light gray background and an orange border. The form contains the following fields and values:

Username *	Name *	Ellection Party *
b	b	b
Father Name *	Mother Name *	Citizenship *
b	b	b
Age *	Date of Birth *	Mobile *
20	May 9th	8871697651
Password *	Education Qualification *	Constituency *
•	b	Bhopal

Below the form fields, there is a section for uploading the election party symbol:

Upload the ElectionParty Symbol

At the bottom of the form, there are two buttons: and .

Candidate Registration

- Registering an election for the ‘Bhopal’ constituency.

The screenshot shows the 'Organizer Dashboard' with a dark header. On the left, there are two buttons: 'CREATE ELECTION' and 'ALL REQUEST'. The main area features a large 'Election Register' form. The form includes fields for 'Election Type *', 'Organizer *', and 'Constituency *'. Below these are 'Election Starts Date' and 'Election Starts Time' (with a dropdown arrow), and 'Election Ends Date *' and 'Election Ends Time *'. Further down are 'Cand Reg. StartDate' and 'Cand Reg. StartTime *', and 'Cand Reg. EndsDate' and 'Cand Reg. EndsTime *'. At the bottom of the form are 'Result Date *', 'Result Time *', and a 'Total Votes *' field with a value of '0' and a spinner control. A blue 'CREATE' button is positioned at the bottom right of the form.

Election Registration

- Once candidate registration starts, candidates can apply to contest an election

The screenshot shows the 'Candidate Dashboard' with a dark header and a 'LOGOUT' button. On the left, there are two buttons: 'ELECTIONS' and 'YOUR REQUEST'. The main area features a large grey rectangle. Overlaid on this is a white card with a black icon of a building with columns, the text 'a Bhopal', and 'a'. Below the card, it says 'Sun May 09 2021 14:47:11' and has a blue 'REQUEST IT' button.

Candidate registration for election

- Confirm candidate registration 'c' for a specific election organized by 'a'

Election Details

Election Type : a
Organizer : a
Constituency : Bhopal
Total Voters : 5
Election Date : Sun May 09 2021
14:48:00
Result Date : Sun May 09 2021
15:10:00
Candidate Registration Last Date :
Sun May 09 2021 14:47:11

Candidate Details

Username : c
Name : c
Election Party : c
Mobile : 1234567891
Constituency : Bhopal

☒ Agree with Details

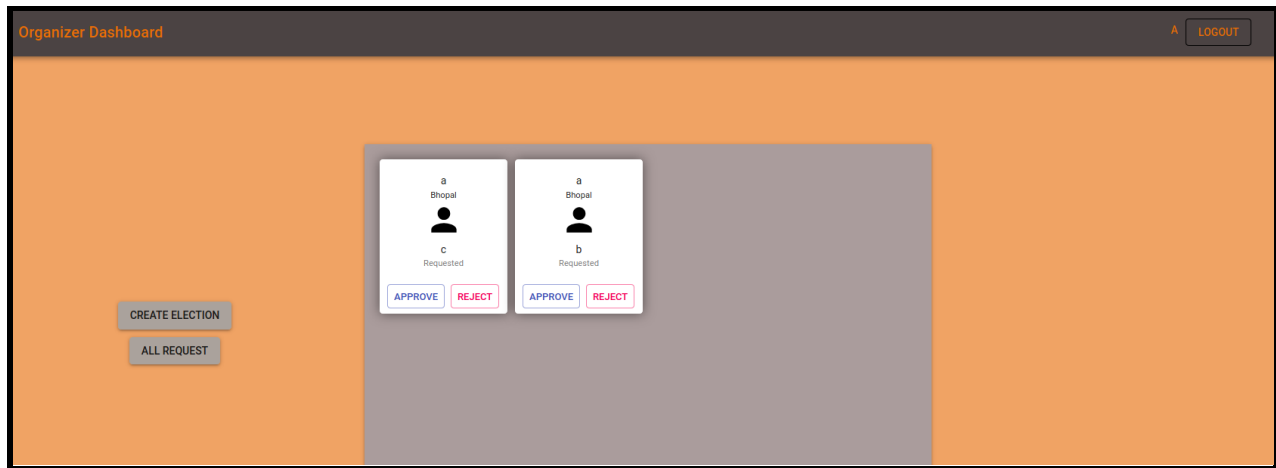
Place *

Bhopal

SUBMIT

Candidate Registration confirmation

- Organizer will approve the candidate's request for participation in a specific election event.



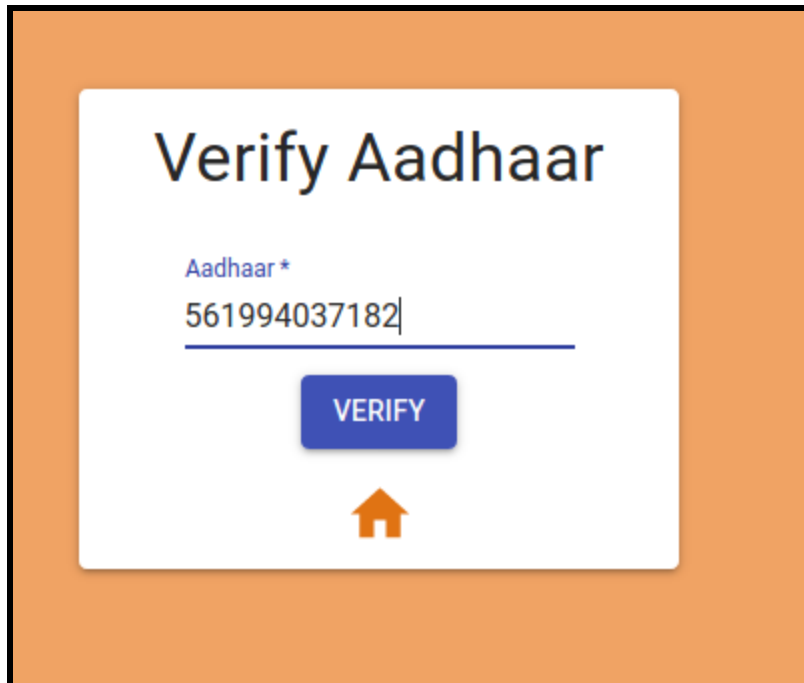
Admin accept candidate registration

- When voting period starts voters are able to vote for registered candidates



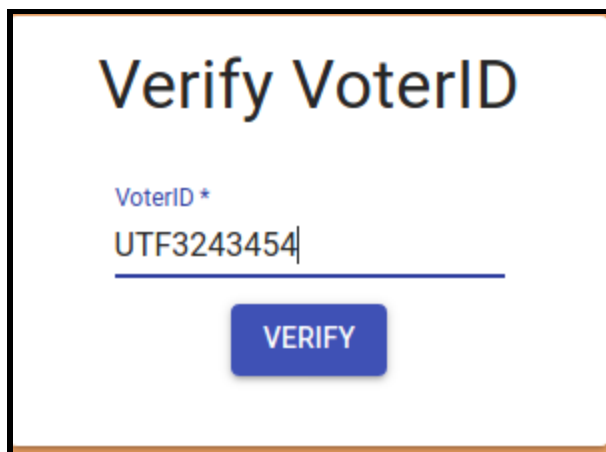
Voting start

- Voters need to verify their Aadhar

A screenshot of a mobile application interface for Aadhaar verification. The screen has an orange background. In the center is a white rounded rectangle. At the top of this rectangle, the text "Verify Aadhaar" is displayed in a large, dark blue font. Below this, the label "Aadhaar *" is in a smaller blue font. Underneath the label is a text input field containing the number "561994037182". A blue horizontal line is positioned below the input field. Centered below the line is a blue rectangular button with the word "VERIFY" in white capital letters. At the bottom of the white rectangle is an orange house icon.

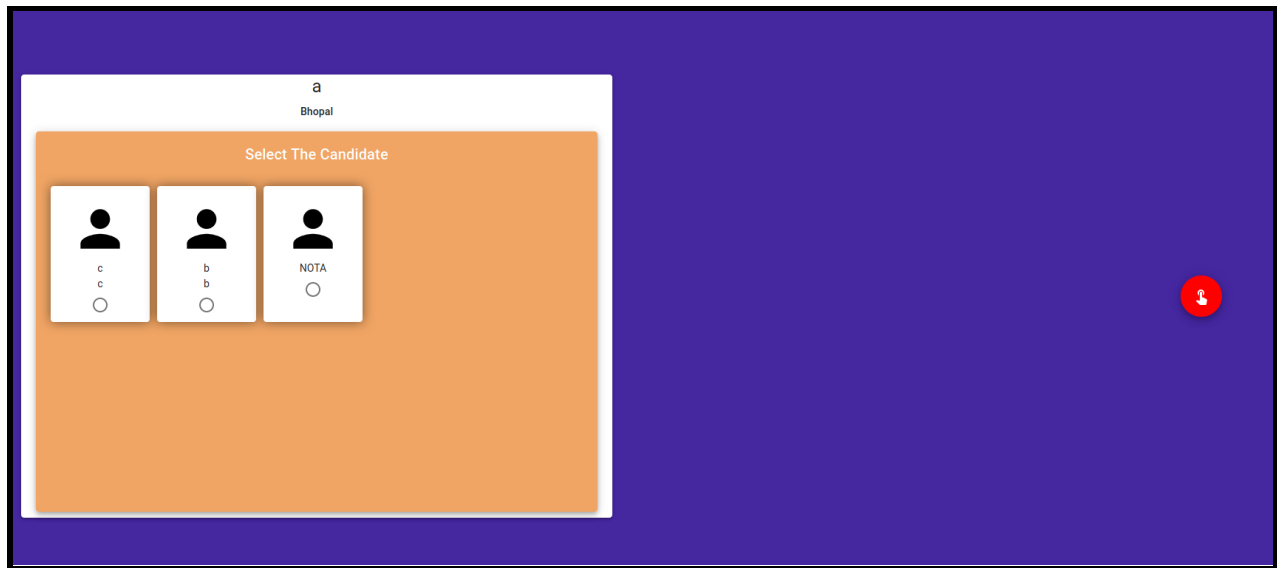
Aadhaar verification

- Voters need to verify their VoterID

A screenshot of a mobile application interface for VoterID verification. The screen has a white background with a thin orange border. In the center, the text "Verify VoterID" is displayed in a large, dark blue font. Below this, the label "VoterID *" is in a smaller blue font. Underneath the label is a text input field containing the number "UTF3243454". A blue horizontal line is positioned below the input field. Centered below the line is a blue rectangular button with the word "VERIFY" in white capital letters.

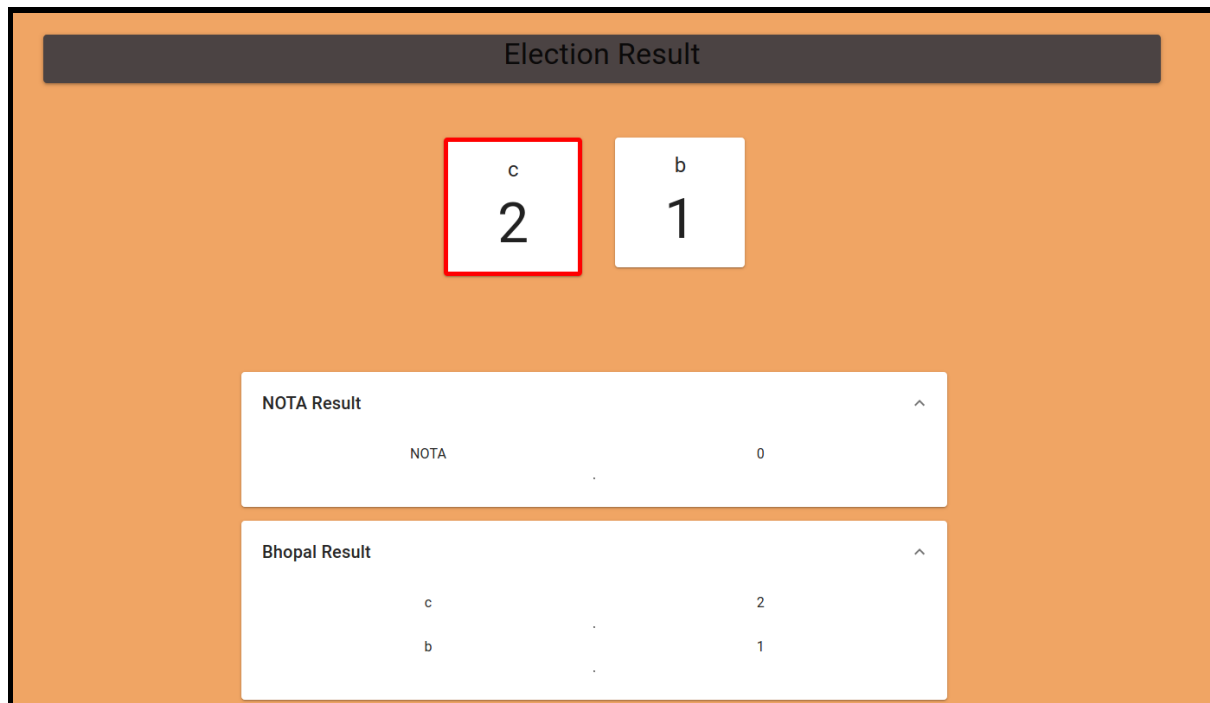
Voter verification

- Finally after all verification voting portal is available to the voters



Election portal

- Shortly after voting period ends results are declared



Election results

12. Scope of Improvement

Listed below are the possible scenarios which can be improved upon:

1. If someone is capturing them and forcibly making them vote, it will become a very big issue similar to the present voting systems.
2. In some areas literacy rate is very low and residents of those areas cannot use mobiles, computers or other technical gadgets. In these areas, E-voting cannot be implemented.

13. Conclusion

In this paper we discussed the different types of problems associated with the offline voting systems in the present scenario of the pandemic. We also discussed the different problems such as long waiting times of the people in the queue in the offline elections system. Our application deals with these problems keeping all the aspects of security and transparency in check. We also discussed the technology and methods we used in the application which would help us in better understanding about the application.

The different literature reviews mentioned in the paper gives us a deeper insight about how the different technologies and methods used in the applications have evolved. These literature reviews give us a better understanding of the topics related to blockchain. We came across different types of enhancement that were brought in the context of blockchain.

We then proposed our own application which makes use of a local blockchain with our own smart contracts written on it. We then run our Decentralised Voting Application on our local blockchain and run the voting system. We simulated a demo run of the voting system where we discussed the basic steps to run the application. In this demonstration we created multiple election and registered candidates who contest in the election and at last which voter votes to the candidate.

14. References

- [1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system. online." Available in <https://bitcoin.org/bitcoin.pdf>(2008).
- [2] Khan, Kashif Mehboob, Junaid Arshad, and Muhammad Mubashir Khan. "Secure digital voting system based on blockchain technology." *International Journal of Electronic Government Research (IJEGR)* 14.1 (2018): 53-62.
- [3] Liu, Yi, and Qi Wang. "An E-voting Protocol Based on Blockchain." *IACR Cryptol. ePrint Arch.* 2017 (2017): 1043.
- [4] Ante, Lennart. "Smart Contracts on the Blockchain–A Bibliometric Analysis and Review." *Telematics and Informatics* (2020): 101519.
- [5] Singh, Ashish, and Kakali Chatterjee. "Secevs: Secure electronic voting system using blockchain technology." *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2018.
- [6] Kodinariya, T. M., and Ravi Seta. "Visual data mining in indian election system." *International Journal on Computer Science and Engineering* 4.7 (2012): 1323.
- [7] Wang, Shuai, et al. "An overview of smart contract: architecture, applications, and future trends." *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018.
- [8] Park, Sunoo, et al. "Going from bad to worse: from internet voting to blockchain voting." *Journal of Cybersecurity* 7.1 (2021): tyaa025.
- [9] Steichen, Mathis, et al. "Blockchain-based, decentralized access control for IPFS." *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018.
- [10] Fink RA , Sherman AT , Carback R. Tpm meets dre: reducing the trust base for electronic voting using trusted platform modules. *Trans Info For Sec* 2009;4:628–637.

- [11] Iovino V , Rial A , Rønne PB et al. Using selene to verify your vote in JCJ. In: Brenner M , Rohloff K , Bonneau J et al. (eds.), *Financial Cryptography and Data Security*. Springer International Publishing, 2017, 385–403. ISBN: 978-3-319-70278-0.
- [12] Dwork C , Naor M. Pricing via processing or combatting junk mail. In: *Annual International Cryptology Conference*, Springer, 1992, 139–47.
- [13] Ben Sasson E , Chiesa A , Garman C et al. Zerocash: decentralized anonymous payments from bitcoin. In: *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, 459–74.
- [14] Park S , Albert K. A Researcher's Guide to Some Legal Risks of Security Research. A joint publication of the Cyberlaw Clinic at Harvard Law School and the Electronic Frontier Foundation, October 2020.
- [15] Hopwood D , Bowe S , Hornby T et al. Zcash protocol specification. In: *Technical report 2016–1.10*. Zerocoin Electric Coin Company, 2016.
- [16] Juels A , Catalano D , Jakobsson M. Coercion-resistant electronic elections. In: *Proceeding 2005 ACM Workshop on Privacy in the Electronic Society*, ACM, 2005, 61–70.
- [17] Springall D , Finkenauer T , Durumeric Z et al. Security analysis of the Estonian internet voting system. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 703–715.
- [18] Nizamuddin, Nishara, Haya R. Hasan, and Khaled Salah. "IPFS-blockchain-based authenticity of online publications." *International Conference on Blockchain*. Springer, Cham, 2018.
- [19] Bhosale, Kumar, et al. "Blockchain based Secure Data Storage." *International Research Journal of Engineering and Technology (IRJET)* 6.03 (2019): 4.