# Improvement of the Voting System using Blockchain

We went for a hackathon (Arithemania 3.0) on 13th April 2024. It was a 30-hour hackathon consisting of 4 tracks.

The tracks were as follows:

1. Bio-Informatics
2. Finance
3. Networking
4. Security

We had decided on the networking domain. After thinking and researching we chose our problem statement.

The Lok Sabha elections in India have always been a cumbersome process with a lot of money, time, and manpower put into it. Apart from these, rigging, multiple voting by a single voter, identity threats, etc. have been a constant problem during the election process.

Using blockchain, these problems can be easily incurred.

Blockchain is a decentralized ledger technology that records transactions across a network of computers. It's transparent, secure, and immutable. Transactions are stored in blocks that form a chain, ensuring data integrity. Blockchain operates without a central authority, using consensus mechanisms to validate transactions. Its applications range from cryptocurrencies to various industries requiring secure and transparent record-keeping.

Bitcoin was the first application of blockchain which uses proof of work (Pow). It is used for making transactions using

bitcoins, where miners acting as nodes validate the transaction by guessing the true hash value (generating an integer which is converted to a 256-bit binary number using SHA256(used in cryptography) as a hashing method of the first transaction block, hence adding block after block, creating a chain of blocks. The longer the chain, the more authentic the transaction. Also, the anonymity of the person initiating the transaction remains as only their public key is what is used for the transaction and not the private key. Two transactions from one source at a time are not possible. It takes an average of 1 hour for a transaction to be successfully carried out.

Ethereum is another blockchain platform that uses ethers instead of bitcoins (a token of transaction/real money is used to buy ethers which is used for transactions). It is better than the technology of bitcoin as it uses proof of stake (Pos) which uses a beacon-chain as well for validation which uses 99.95% less energy compared to that used in Pow.

We wanted to create an online voting platform that uses blockchain to make the process more accessible, transparent, and efficient.

- Accessibility: Everyone with a phone or computer can cast their vote irrespective of their location and ability to go to a voting booth (increasing participation drastically).
- Transparent: An open-source blockchain implements the system to increase trust. The blockchain also helps prevent voting fraud like rigging, identity theft, multiple voting, fraudulent counting of votes, etc.

- Efficient: Elections can be held faster, for cheaper whilst increasing integrity in the process.

Our Solution:

An Open-Source blockchain platform for citizens to cast votes seamlessly, anonymously, and securely. We aim to modernize voting by using blockchain, enabling people to vote from anywhere with their devices. This approach ensures transparency, as the blockchain makes tampering nearly impossible. It also streamlines the process, reducing costs and making elections more efficient. Votes remain anonymous, protecting voter privacy.

Tech Stack:

- Blockchain Platform/Framework: Ethereum Fabric Smart Contract
- Development: Solidity
- Frontend Development: React.js
- Backend Development: Node.js + Express.js and implement APIs for blockchain interaction
- Database: MongoDB
- Development Tools: Remix IDE, VS Code
- Technical Workflow: Setup Development Environment
- Integration: Integrate frontend, backend, and blockchain components

Implementation:

Blockchain was something completely new for all of us until the 13$^{th}$ of April. We first read about the basics of blockchain and Ethereum.

We read about the principle behind blockchain, how it works, its applications in the real world, and its pros and cons. We

also read about real-life usage of the concept of blockchain in the voting system. It was used in Virginia, where they used blockchain to stop rigging by fitting their technology into EVM machines. We also read about case studies that talked about whether using blockchain in a voting system would be feasible or not.

After that, we started reading about the necessary software, knowledge, etc. that would be needed to create an online voting platform using blockchain.

A smart contract is used to define the functions which in turn defines how the blockchain needs to act. Smart contracts are typically used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when predetermined conditions are met.

Solidity (a programming language) was used to write this smart contract which needed to later be deployed on Ethereum.

The functions defined were as follows:

- Registering a Voter: The <registerVoter> function allows a voter to register by marking them as registered in the voters mapping.
- Casting a Vote: The <castVote> function allows a registered voter to cast a vote for a candidate by incrementing the vote count for that candidate in the <votesReceived> mapping. It also marks the voter as having voted.>
- Adding a Candidate: The <addCandidate> function allows the contract owner to add a candidate to the list of candidates.

- Retrieving Votes for a Candidate: The <getVotesForCandidate> function allows anyone to retrieve the total votes received by a candidate.
- Retrieving Candidate List: The <getCandidateList> function allows anyone to retrieve the list of candidates.

After this, we tried to use Truffle Ganache to deploy it in Ethereum. But we were unable to do so. Instead, we used Remix IDE to compile and deploy it on Ethereum. Since Ethereum Mainnet needed us to use money to buy ethers we instead used Remix VM to run the same. It took us quite some time to make it work. However, after a lot of trial and error, the creation of blockchain was finally successful.

We then started working on creating the backend using Node.js and Express.js.

Backend:

- Blockchain Node: Stores data, validates transactions, and executes smart contracts for the voting process.
- Smart Contracts: Developed using Solidity, handle voter registration, ballot creation, vote casting, and result tallying.
- API Layer: Provides a RESTful interface for client interaction with the blockchain and smart contracts.
- Database: Stores non-sensitive data like voter registration details, ballot configurations, and election results.
- Authentication and Authorization: Ensures only eligible voters can cast votes, integrating with identity verification services.

- Security and Auditing: Implements encryption, secure API endpoints, and transaction logging for security and auditing.
- Scalability and Performance: Designed to handle large numbers of users and transactions, especially during peak voting periods.

After troubleshooting we compiled the backend successfully.

After that, we started working on the Frontend using React.

Frontend:

The frontend consisted of a main page which connects to the admin page or the voter page depending on the end user.

Admin Page:

- Register Voters: Form for the election commission to register voters by providing voter details.
- Add Candidates: Form to add candidates to the election by entering candidate details.
- Retrieve Election Results: Display of election results including candidate names and vote counts.
- Show Candidate List: Display the list of candidates running in the election.

Voter Page:

- Authenticate Voter: Form for registered voters to authenticate themselves (e.g., using a voter ID or other credentials).
- Vote for Candidate: Display the list of candidates with an option to select and cast a vote for a favored candidate.

While connecting the front end with the back end and the blockchain, we faced a lot of problems. After troubleshooting and successfully compiling, the output was a blank page. On inspection, it was giving a header error. We suspected that it was a problem with the database that we were using (Mongodb) but after reinstalling, the error remained.

We tried our best to solve this error till the last second before our final presentation.

We were the first to present. After the presentation, the judges and fellow teams asked us a few questions, which were as follows:

1. How did you think about your problem statement and why did you end up thinking about using blockchain?
2. Are there previous examples of using blockchain in the voting system?
3. Are there any security concerns or security problems related to your solution?
4. How are you going to ensure that there won't be network overload during the process of voting?
5. How do you ensure the anonymity of the voter?

Answers:

1. Since elections are coming up, we thought about all the problems that one faces while voting. Then we thought about how blockchain is very secure and can be used for creating an online voting platform.
2. Virginia: Blockchain in EVM, case studies.
3. Blockchain is very secure which prevents malicious tampering of cotes by hacking due to the 51% rule. A hacker needs to ensure that he has 51% control over the network/nodes which is acting as miners which is nearly impossible.

4. Our blockchain voting platform would be active only on the three days of elections and instead of having just a single central node i.e. election commission, we would have multiple nodes distributed over the country (election commission) which would take over the burden of having overload on a single node. We could also create slots so that not the full population would vote at a single time but it would be distributed over these slots thereby reducing the chances of overload.
5. The voter would need to go offline to get verified by the election commission and would get access to his/her private key. During the voting process, only their public key would be visible to the election commission and thus their identity would not be exposed also who they voted would also be unknown. The election commission would only be able to know that a "human" voted for a valid candidate. The election commission would be able to see the number of votes each candidate received.

After all the presentations the judges told us to keep working hard and continue hacking. He also mentioned how it's important to be sure about one's problem statement and its implementation. It is not always possible to end up with a full working prototype which is completely fine and we should be sure about our idea.

-Sampriti Saha