JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DECENTRALIZED VOTING USING BLOCKCHAIN

Jaival Patel , Faisal Khan , Bhautik Ladva , Rovina D'Britto
Student , Student , Student , Professor
Department of Information Technology , Universal College of Engineering, Mumbai University , Vasai , India

Abstract: **The Election process helps us decide the representative who will handle all the different operations of a nation once they are selected. Hence the election process needs to be fair as well as democratic for all the voters. Thus the voting system needs to be secure, fair, should have integrity as well as be consistent before it can be used for the election. Earlier the voting system used to be based on paper voting but as the technology evolved E-voting concept appeared in which the election systems were converted into the online medium. Now the world is moving towards I-Voting in which the user can vote through the comfort of his home without needing to travel for it.**

As World moves towards a digital era more and more countries are trying to Research and shift towards E-voting as well as I-voting.As the current systems follow a more centralized approach and are not transparent to voters. With help of Blockchain Technology, we can move towards a decentralized architecture that follows a transparent transaction system along with other benefits like tamper-proof as well as a decentralized database. As Blockchain shares all information over its P2P Network its Immutable as well available for all to view at the same time. With the rise in Blockchain technology, we are also able to solve issue-oriented to privacy with help of Web 3.0 as well dApps.Hence we are proposing this Web-based project to use Blockchain and its benefits for an I-Voting Application.

Keywords— ***Blockchain, I-Voting, SHA 256, AES, Node.js***

I. Introduction

Trust, Autonomy, and intermediaries are three big problems which we are facing at the present time like we are obliged to trust banks for securing our money for our transactions. We depend upon these third parties to ensure our privacy and security in terms of our data. So in any case in today's world, we have to trust these organizations which act as intermediaries. These three big problems can be solved through an undeniably ingenious invention called a blockchain. As the applications build using blockchain are decentralized and owned by multiple parties and no one can change or update data in the blockchain. If someone tries to do so it will not be accepted by stakeholders. Hence, making blockchain completely trustable. There is no single owner of blockchain, which means no single authority is controlling it and anyone can participate in the network (depending upon the type of blockchain i.e public, private, and consortium). Blockchain is a read and write-only database i.e once data are written on the blockchain it cannot be altered or changed.

India has been using EVMs since 1999 for election.EVMs are cheaper than the other options for India. It is also cheaper than E-voting System but the process of EVM is not transparent to the voters and it requires physical presence for voters. While in the case of I-Voting a user can vote Online and voters can do that with comfort at home. Along with that benefits of Blockchain like Transparency, Anonymity and consistency are also applied, and it's effectively cheaper too. Therefore since 2006, Estonia has been the first country to use I-Voting for its Election.

The I-Voting system should be able to have functions like as follows:

- Authorized voters should be allowed to vote to whomever they want to and should have equal opportunity.
- The election should be affordable and available to all.
- The voter should be able to verify whom he voted for.
- The vote should be tamper-proof.
- The votes data should be auditable.
- We should be able to verify votes data.

II. Literature Review

This section consists of a thoughtful analysis of some of the other Authors who have developed similar solutions.

In **[1]** Adhaar fingerprint data for registering a user was used and these details are used for voting while this might be secure it requires a physical hardware device to capture fingerprint data.This data is used to generate a key pair and this key pair is used to encrypt data of vote.Here the Blockchain used was made by the authors themselves.Once the user logins into his account he can cast a vote using the same fingerprint id he used to create the account.

In **[2]** creating an eid that is linked to a key, and this key is generated by a Key Authority. The Blockchain used for this project was Prebuilt Ethereum based which were as follows Ganache, Hyperledger, Ropstem all these were tested to store voting and Result of final voting. This System relies on Smart Contracts where they work according to a given condition and only process the request when the criteria are met towards it.The language used to build this System is known as Solidity.It is used to Write, Implement as well test Contracts on Ethereum Blockchain.Here once the user logins into his account he can cast a vote using the balance of his wallet which is provided by ganache or the corresponding blockchain.

In **[3]** we see that the Ethereum Solidity Smart contract was used for Blockchain and the user details were Stored on Mongoose Db. This System Relies on the Same smart Contracts and it is also based on Ganache Ethereum Blockchain. They used a basic Web app for the front. The voting data is stored on Ganache as a transaction and results are calculated from it. Here also the project relies on Solidity for writing contracts on Ethereum Blockchain. Here also once the user logins into his account he can cast a vote using the balance of his wallet which is provided by ganache or the corresponding blockchain.

While doing this survey we learned that these systems relied on traditional blockchain like Ethereum which meant it was limited for analysis purposes and also required Metamask to work correctly to eliminate all this we created a custom blockchain accordingly which is scalable and flexible as per needs.

III. Proposed System

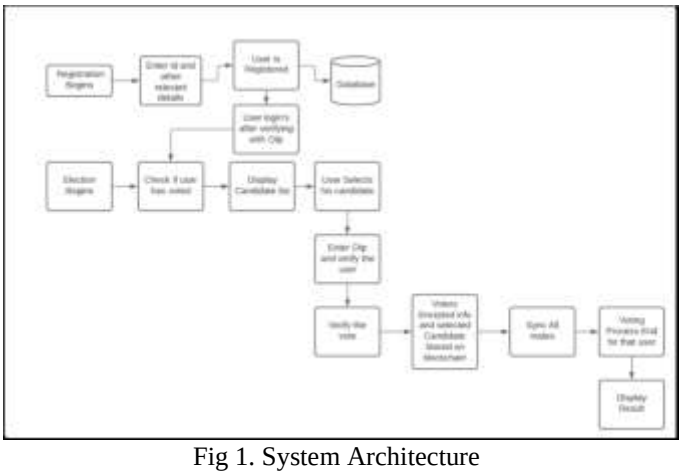

Fig 1. System Architecture

The Adhaar id is used to generate a unique id as shown in user registration this makes the user details anonymous and as well as secure and prevents the issue of generating a private key pair as we learned in the literature survey overall making the experience seamless.

We used MongoDB to store all registration details and used additional server Encryption on it. MongoDB is an cross-platform document-oriented database program. It uses NoSQL, MongoDB follows a JSON-like documents format for schemas.

1. User Registration

The User is registered using their Adhaar-Id and Other relevant details which are Encrypted before storing on the database

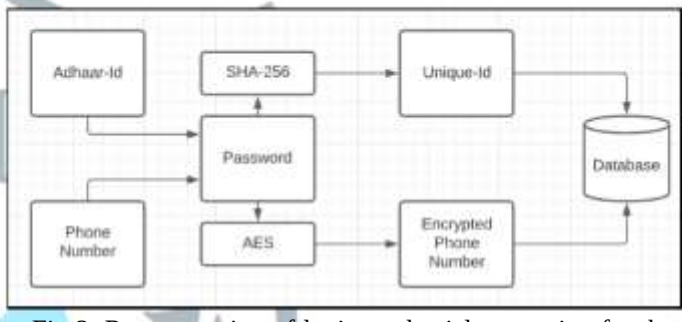

Fig 2. Representation of login credential generation for the user.

The User details are further Encrypted with additional encryption on the server. The Unique id(Uid) generated is unique to each user and it is non Identifiable by anyone other than the user himself. The phone is encrypted using the password as a key which then helps us send OTP to that number once the password is correct.

2. User login

The user Logins using his Adhaar id and password and once the details match up an OTP is sent to the User.

3. User Voting

Once the user login he can cast a vote before voting he will need to enter the OTP sent to the number to confirm his vote.

4. Blockchain

The Blockchain contains hash and previous to link itself the hash value is its own hash value and the previous hash contains the hash of the previous block.

For this specific project, we built our own blockchain-based on Javascript node.js and used it to store Data. Because of this, we were able to store and analyze as many details of users as we needed. We were also able to make the mining algorithm and amount of votes per block adjustable as per needs.

We also Implemented the consensus algorithm for our own blockchain which uses the longest verified chain rule.

This rule states that the chain which is valid and longest in the network that chain acts as the main chain of the network. More about it in the Consensus Algorithm section below.

For this Project we implemented our own Blockchain using Node.js.For frontend purposes, we used React and its libraries and implemented an OTP mechanism for verification.

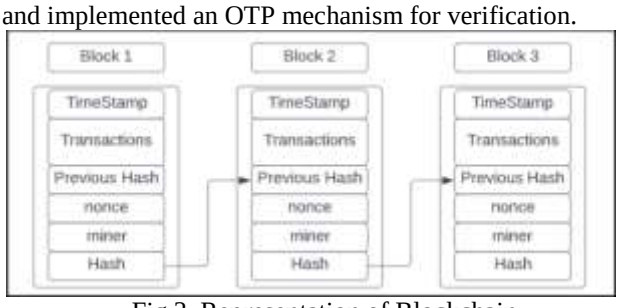

Fig 3. Representation of Blockchain

The vote of the user is added to Blockchain if the user is voting the first time and rejects the vote if the user has voted already. This vote is then broadcasted to all Network Nodes present in the network.

Fig 4. Live Blockchain

5. Admin Panel

This is where all the configurations of Blockchain are done. In case a node fails it can be revived using the consensus algorithm. We can adjust all the blockchain parameters in this panel from the Mining limit(Automines Once limit is matched) to amount nodes in Network. We can also set the number of votes per Transaction Block in Blockchain.

Fig 5. Admin Panel of project

IV. Methodology and Working

Step 1:Run following command in directory client to start frontend

Fig 6. npm start

Step 2: Start the Mongo Db server by running command nodemon in server directory

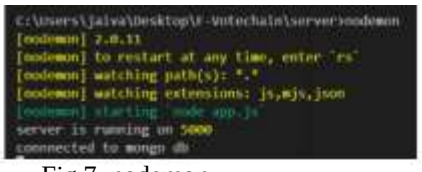

Fig 7. nodemon

Step 3: Start Custom Blockchain by using following command

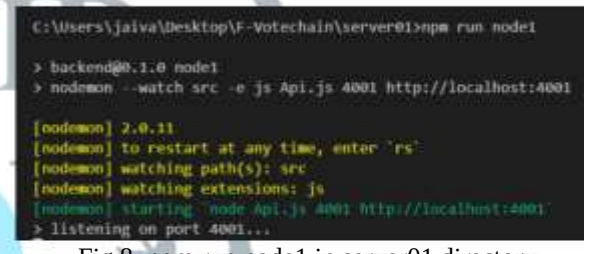

Fig 8. npm run node1 in server01 directory

The 3rd steps needs to be repeated for the number of nodes we want to add in the network.
We can automate the creation of all nodes by using .bat too which we used to create multiple nodes at once.
All nodes store Blockchain and can be recovered from failure too if needed.

Step 4: Once all the above steps are executed you can log in as admin in frontend and configure the blockchain accordingly.

Step 5:Once parameters are set, Admin can click Start voting and voting begins and the result is displayed once Admin clicks End voting result is mailed as well as shown on Dashboard.

Consensus Algorithm

To Make the Blockchain Secure and Recoverable we Implemented Consensus Algorithm. Before running consensus we need to make the Node aware of its network so again we register nodes inside the network once a node is registered in the admin panel admin can write the name of the node that needs to be revived.

In case a node in the network fails like node1 fails it can be revived using Run Consensus Option from the Admin panel Which returns the node to Original State. This is done in several steps as follows

1. Get the latest valid Blockchain from all nodes
2. Verify if all of them are valid and select the longest chain from all of those
3. Finally, replace all parameters as well as the longest verified blockchain.

Fig 9. Inside Postman running Consensus Algo with Auth token.

Once the node is revived it becomes active again in-network and stores new data on its own.
With help of this, the blockchain becomes immutable and secure which makes it perfect for applications like real-time elections which may face a node failure over longer hours.
Using Consensus Algorithm the blockchain can be made as it was during the original working condition and helps resume election without needing for a user to vote again which would be needed if an EVM got damaged or got glitched.

V. RESULT

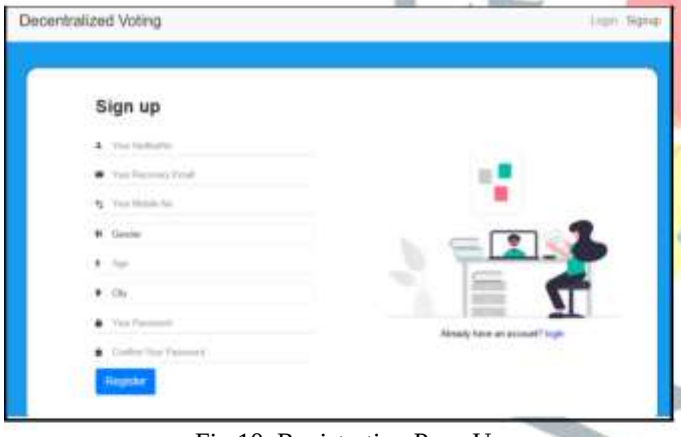

Fig 10. Registration Page User Enters their details over this page

Once the user is registered a mail is sent with corresponding generated Uid.

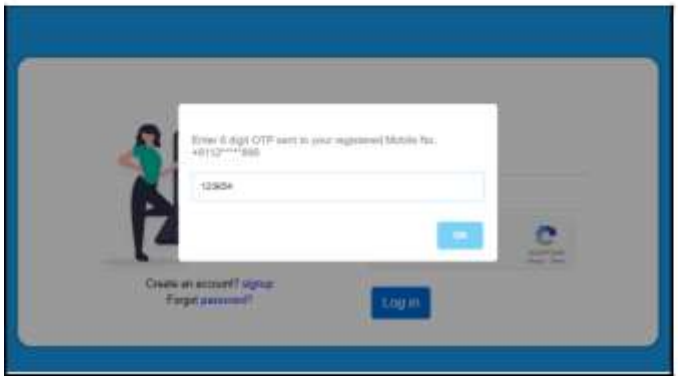

Fig 11. OTP Request

Once the user logins successfully an OTP is sent which he needs to enter to successfully log in.

Users can further reset their account credentials if in case they are forgotten. The user will need to enter his Uid to initialize the process of resetting credentials. As he enters his Uid a mail will be sent to the user with a reset account link where the user can enter new details but used the same old Adhaar-id. User can update their details like phone numbers as well as passwords the same way. The User cannot reset account details during as Election period.

Fig 12. Voting Process

Once the user clicks on the vote button he will need to again enter the password and OTP to verify himself and then this vote will be added to the blockchain.

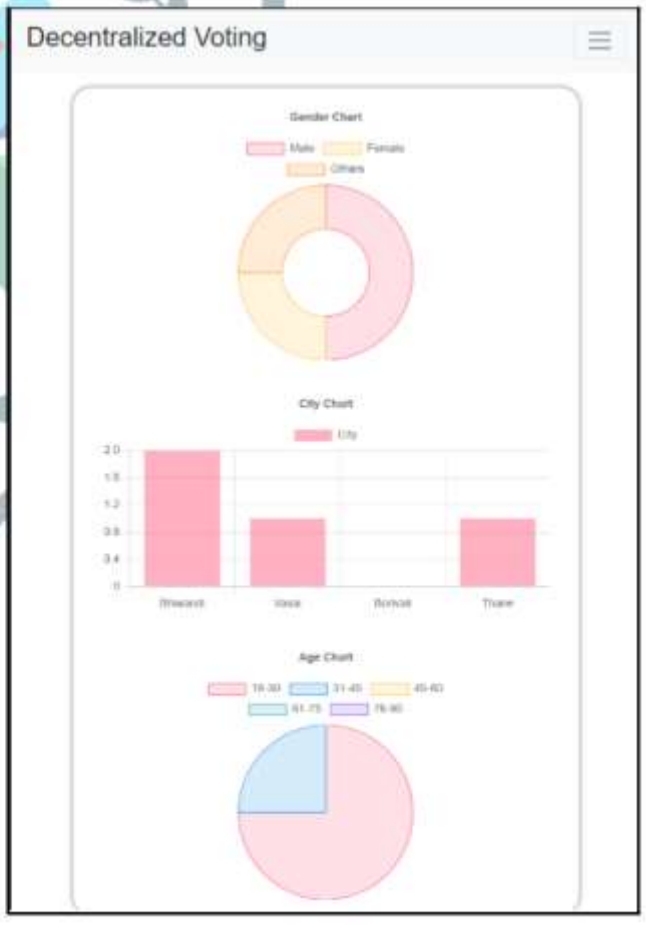

Fig 13. Live Dashboard Statistics

Here you can see the live statistics of voting

When Admin Clicks End voting button in Admin Panel the Winning Party name is Shown and results are mailed to all users.

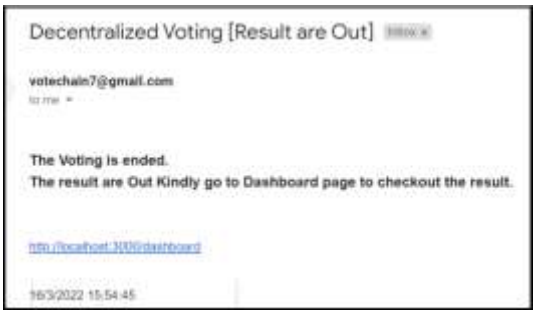

Fig 14. Mail sent to users

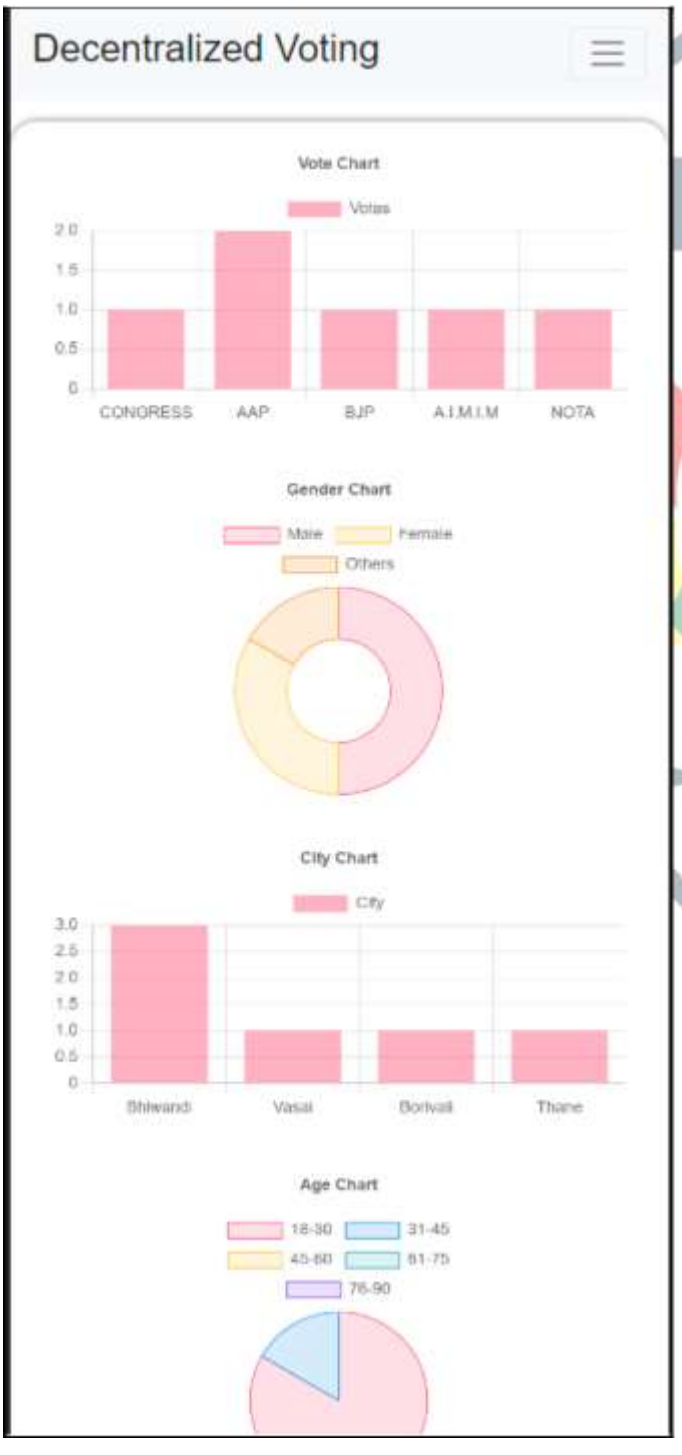

Fig 15. Final Analysis After Voting Ends

Once the voting ends all Blockchain data is analyzed and shown in the graph to all users as well the admin. These stats are generated and stored as long as the next voting is not started or the blockchain is not reset.

Time Analysis of Blockchain

S.No	No of Votes	Time Taken (in Seconds)
1	250	0.601
2	500	0.999
3	750	1.832
4	1000	1.98
5	1200	2.12

Fig 16. Analyzing of Time taken for votes

The table shows how much time was taken as the number of votes increased. It helps us understand the time required for votes to successfully enter the blockchain. It also shows how it's not too dependent on the machine as more votes do not correspond to a much higher value of time. Instead, it shows how a huge volume of votes would take less time whereas a traditional system of EVM would require much more as there needs to be verification of each voter during the voting process.

VI. Conclusion

The following proposed system helps make the voting process much more seamless, efficient as well as transparent. Once the project is deployed a user can register and vote remotely as both secure as well as anonymously. We successfully implemented and deployed this web app based on Nodejs and used Blockchain which outperforms the traditional way of voting along with other benefits such as transparency, Security as well anonymity of user data is achieved all at once. Due to this web app, the Election result is also analyzed and processed all at once and the result can also be declared instantaneously all at once without waiting a day.

It is highly Scalable and can be changed as peruse of the election since it is also using custom blockchain the data to be stored and analyzed can also be changed as per needs and it is not bounded like traditional blockchains technology.

VII. Future Scope

- Improving Mining Algorithm and implementing new sync methods to improve the efficiency of the blockchain.
- Adding Fingerprint Support for handheld Devices.
- Adding Support for Multiple Simultaneous Election over a large Region.

VIII. References

[1] T. M. Roopak and R. Sumathi, "Electronic Voting based on Virtual ID of Aadhar using Blockchain Technology," *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 71-75, DOI: 10.1109/ICIMIA48430.2020.9074942.

[2] K. Košťál, R. Bencel, M. Ries and I. Kotuliak, "Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain," *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 592-595, DOI: 10.1109/ICSESS47205.2019.9040770.

[3] M. Navarrete, R. Huancas, P. DÍaz and M. Rivadeneira, "Blockchain electronic vote system," 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2019, pp. 1-7, doi: 10.1109/CHILECON47746.2019.8988084