J. Adv. Res. in Cloud Computing, Virtualization and Web Applications
Volume 6, Print Issue 2 - 2023, Pg. No. 8-16
Peer Reviewed Journal

Review Article

Ether Vote: Revolutionizing Elections with Blockchain Powered Electronic Voting System

Shaurya Gautam

Department of Electronics & Instrumentation Engineering, Velammal Engineering College, Chennai, India.

I N F O

E-mail Id:
shauryatripathi6@gmail.com
Orcid Id:
https://orcid.org/0009-0001-6926-5192
How to cite this article:
Gautam S. Ether Vote: Revolutionizing Elections with Blockchain-Powered Electronic Voting System. *J Adv Res Cloud Comp Virtu Web Appl* 2023;6(2):8-16.

Date of Submission: 2023-10-18
Date of Acceptance: 2023-11-23

A B S T R A C T

Voting is an essential component of democracy, as it enables citizens to express their will, hold their elected officials accountable, promote diversity in government, foster civic engagement, and protect against tyranny. Considering the technological advancement that people have acquired in recent times voting system is almost outdated. Electronic Voting, or E-Voting, is a modern method of casting and counting votes in an election. It promises greater efficiency, speed, and accuracy compared to traditional paper-based voting systems. However, E-Voting systems face several challenges, such as security concerns and lack of transparency. To address these challenges, the paper proposes Blockchain Technology as a potential solution. Blockchain is a distributed ledger technology that is immutable, transparent, and secure. By using blockchain for E-Voting, it is possible to create a tamper-proof and transparent system that can ensure the accuracy and integrity of the voting process. Ethereum Blockchain, a decentralized open-source platform for building Decentralized Applications (dApps) using smart contracts is being used. Once a vote is recorded on the Ethereum Blockchain, it cannot be changed or deleted, ensuring the integrity of the voting process. Being decentralized there is no central authority controlling the voting process, which increases transparency and reduces the possibility of fraud. Smart contracts can help automation of the process ensuring basic rules and regulations being followed. The proposed system can be foul proof in most cases and there are proposed norms that can be followed to manage some exceptional cases.

Keywords: Blockchain, Voting, Ethereum, Solidity, Decentralized Applications, Automation, Fraud

Introduction

Being part of the largest democracy in the world, Authors know the importance of voting and thus an effective voting system. Considering the technological advancement, Nation have acquired in recent times that existing voting system is outdated. Current process either be a ballot or Electronic Voting Machine which is time consuming both in the voting and counting process, and uncomfortable to most people which affect the polling percentage as well. And there are even allegations of Electronic Voting Machine (EVM) being tampered. So, introducing an effective and advanced Voting system is the need of hour. The Basic Structure of a Voting System is on 3 different groups, The Candidates, The Voters and the agency who are having it Conducted.

Here the agency who is going to conduct the election. In this system the basic thing agency have to ensure is the trust from two other parties, so that few basic concerns have to be considered.

Authenticity of voter: The voter should be eligible for the election, for example in Indian system, who is a citizen of India who is at least 18 years old, and enrolled for a particular constituency

- **Anonymity of Voter:** even though users must confirm the vote is recorded it shouldn't be recorded in such a way that any 2nd party can map a person with a vote.
- **Authenticity of Candidate:** List of Candidates must be managed by the agency who is conducting the election, In Indian Situation, The Election Commission of India.
- **System being tamper-proof:** Once a vote is made it shouldn't be changed at any cost. And there should be only one way to vote which is through verifying the authenticity of voter. Even the Agency which conducts the election can't cross this line.

So, on the process of tackling these objectives the best way out there will be decentralizing voter registration and validation mechanisms, proper voter validation using IDs, making voting data immutable and integral, forming the system in a reliable and robust environment, and then a simple and easy user interface for both voting and counting process. Whereas, blockchain is a cutting-edge Idea that uses consensus algorithms, protocols and cryptographic functions to allow network decentralization without any point of failure. Software developers can build dApps that takes advantage from distribution property of blockchain using the open-source Ethereum Blockchain, which has a Turing-complete scripting language. Consequently, the following blockchain functionalities will be present in dApps:

- Integrity of Data
- Processes of consensus have decentralized verification and oversight.
- Run-time environment with transparency.
- Runtime environment with public business rules.
- Availability

In this work, authors propose a decentralized voting system based on Blockchain Technology to solve the existing problems of traditional E-Voting systems. This scheme uses a cutting-edge technique to verify and authenticate registered voters.

Literature review

There are many works going on Electronic Voting Systems. The objective of the paper is being to build an electronic voting system using blockchain with a faultless voter authentication system few of important works that have been reviewed are given below:

In A biometric-secure cloud-based E-Voting system for election processes.[1] If the user's physical characteristics vary even little, the system is unable to identify them. They cannot attempt to "alter" their identification characteristics if the data were taken, unlike when a security breach occurs when they can change their passwords. Because it is an autonomous system that runs on electricity, the biometric technology is also unreliable. Using Homomorphic Cryptographic Solutions on E-voting Systems.[2], authors using homomorphic encryption and non-interactive zero knowledge proof for security, Thus It is a centralized System and remote voting Is Impossible, and the encryption does not prevent DDoS attack thus the hacking possibilities on Database. By implementation of secure Online Voting System.[3], authors propose a twofold System, vote can be done by a mobile phone or webpage, User is verified with OTP and Iris scanning, second fold is where voting with a feature phone by Interactive Voice Assistance (IVR), Since it follows basic Internet Voting Security Risk is there, as well as power failure/System Crash can affect the System. In Secure E-Voting System using Blockchain Technology and authentication via Face recognition and Mobile OTP,[4] Even though It is provided with Really good options for authenticating a user underlying Blockchain Technology is not discussed well and it more like a proposal, Ethereum with its Idea of EVM can be used in there as perfectly working practical System. Using Ring Signatures for an anonymous E-Voting system,[5] users use the mechanism of Ring Signature to come up with a decentralized System, which is a good alternative with the idea of ring signature, which gets the value at the end of each transaction. The problem here is there is no way to prevent dual voting which is a serious problem when it comes to Voting System. To make a Framework Voting System Transparent using Blockchain Technology,[6] authors propose a very good framework for blockchain based voting System, and the approach to prevent attack 51 is a take away from paper. On the design and implementation of a blockchain enabled e-voting application within IoT-oriented smart cities,[7] deals with evaluating the trustworthiness of IoT device in a public network o that a smart-city itself can accommodate a voting system, relying on a public Networking is Debatable still worth consideration.

Hardware Accelerators for Real-Time Face Recognition: A Survey,[8] provides technologies that can accelerate the speed and precision of Face recognition, said that it can be considered, it lies on Expensive side. Face live detection method based on physiological motion analysis, in Tsinghua Science and Technology,[9] provides an alternative software solution for confirming the authenticity of face, it can't be tricked using a photograph etc., which is a cheaper option that can be considered and at the same time foul proof.

Research Methodology

Blockchain

Blockchain is a ground-breaking technology that makes it possible to store and transfer digital data in a secure, decentralized manner.[10] In essence, it is a distributed ledger that keeps track of different transactions across many computers or network nodes. Blockchain enables a visible and impenetrable record of data, in contrast to conventional centralized systems where a single organization maintains control. at its core, a blockchain consists of a series of blocks, each consisting a list of transactions. These blocks are linked together in a chronological order, creating a chain of information. Each block contains a unique identifier which is a cryptographic hash, this is produced depending on the information contained in the block. This hash ensures the block's and its contents' integrity. Decentralization is one of the core characteristics of Blockchain Technology. Blockchain employs a consensus process as opposed to a centralized authority, like a bank or a government, to verify and approve transactions. With the help of a network of participants known as nodes, this technique allows for the validation and agreement of the blockchain's current state. This distributed consensus verifies the validity of the recorded data and assists in preventing fraud. Security is another important feature of blockchain. Cryptography is used to safeguard the transactions that are recorded on the blockchain, making them nearly unchangeable and impervious to fraud. After a transaction is entered into the permanent ledger, it cannot be easily changed without the network's agreement. In a variety of applications, such as supply chain management, financial transactions and digital identity verification, this trait improves confidence and transparency. A lot of people are interested in Blockchain Technology because it has the potential to change many different industries. It provides advantages like improved productivity, cost savings, increased security, and improved traceability. Blockchain can also be used to build smart contracts, which are self-executing contracts with preset rules, automating procedures and lowering dependency on middlemen. Blockchain has far-reaching ramifications outside of financial applications, while being most frequently linked with cryptocurrencies like Bitcoin. By offering a secure and open platform for the exchange of digital assets and information, it has the potential to upend sectors including healthcare, logistics, real estate, and voting systems. In conclusion, blockchain is a secure, decentralized technology that makes it possible to store and transfer digital data in a transparent manner. By leveraging distributed consensus and cryptographic methods, it ensures trust and dependability. Blockchain has the ability to completely change businesses and spur innovation in the digital age. Main Features of blockchain are it being Immutable, Distributed, Decentralized, Secure, Consensus, Unanimous, Faster Settlements etc.

Ethereum

The possibilities of conventional Blockchain Technology are enhanced by the blockchain-based platform known as Ethereum.[11] Vitalik Buterin made the suggestion, and it was introduced in 2015. Ethereum offers a decentralized framework for executing smart contracts and creating decentralized apps, in contrast to Bitcoin, which focuses solely on Digital Currency Transactions (dApps). Developers can build and implement smart contracts on the Ethereum Blockchain, which is intended to be a worldwide, open-source platform. Because they are written in code, smart contracts fulfil their responsibilities automatically. They automatically finish transactions and uphold agreed-upon criteria without the need for middlemen. Ethereum Turing-complete programming language, Solidity, is one of its fundamental characteristics. Solidity is a programming language used by developers to create smart contracts that are executed by the EVM,[12] a decentralized runtime environment. The EVM makes sure that each node in the network consistently carries out the smart contracts. Decentralized Autonomous Organizations (DAOs), which are businesses regulated by smart contracts, were also introduced by Ethereum. Decentralized decision-making and resource management are made possible by DAOs. Decisions are decided by the agreement of the token holders, and participants' voting rights are determined by their token ownership. Many decentralized applications have been created using Ethereum open and adaptable architecture. Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), gambling, supply chain management, identity verification, and other use cases are just a few examples of the many use cases that these dApps can address. The sizeable developer community and environment of Ethereum have aided in its development and adoption. It's important to remember that Ethereum is always changing, and new releases like Ethereum 2.0 are planned to solve scalability and enhance the network's overall performance. With these improvements, Ethereum should be able to support more users and applications while also becoming more scalable, secure, and efficient. Ethereum is in the process of switching from Proof of Work (PoW) to Proof of Stake as a consensus method (PoS). PoS chooses validators to create new blocks based on the number of coins they hold and stake as collateral, whereas PoW includes miners competing to solve challenging mathematical puzzles to add new blocks to the network. Ethereum will undergo this change in order to become more scalable and energy-efficient.

Ether (ETH): It is the name of the native crypto currency of the Ethereum Blockchain. Transactions are facilitated using

ether, which is also used to pay users for computing work done on the network. It is used to deploy smart contracts, pay transaction fees, and operate as an incentive system for safeguarding the Ethereum network.

Enabling the execution of smart contracts and the creation of decentralized apps, Ethereum is a blockchain platform that enhances the functionality of conventional blockchain. By utilizing its native currency Ether and the EVM, it offers a stable environment for developers to produce creative solutions across a range of industries

Solidity

A high-level programming language called Solidity was developed expressly for using in building smart contracts for the Ethereum network. On the Ethereum platform, it is the most popular language for creating dApps and running smart contracts. Since Solidity is influenced by existing languages like C++, JavaScript, and Python, developers with experience in these platforms will be familiar with it. It supports inheritance, libraries, and sophisticated user-defined types and is statically typed. Additionally, Solidity offers comprehensive support for contract-oriented programming, allowing programmers to specify how smart contracts behave and their underlying data structures.

➢ Key Features of Solidity:

Contract-oriented: Solidity allows developers to define contracts, which are the fundamental building blocks of Ethereum applications. Contracts encapsulate data, state variables, functions, and events, providing a structured approach to programming on the Ethereum Blockchain.

Smart Contract Development: Solidity enables the creation of self-executing smart contracts that automatically enforce predefined rules and conditions. Developers can define functions, events, modifiers, and structuring data within contracts.

Ethereum Virtual Machine (EVM) Compatibility: Solidity code is compiled into bytecode, which is executed on the Ethereum Virtual Machine (EVM). The EVM is a runtime environment that allows smart contracts to run consistently across all nodes in the Ethereum network.

Security Considerations: Solidity provides features to address security vulnerabilities commonly found in smart contracts, such as reentrancy attacks, integer overflow/underflow, and more. It includes syntax for specifying access control, modifiers, and error handling to help developers write secure and robust contracts.

Libraries and Inheritance: Solidity supports the use of libraries, allowing code reuse and separation of concerns. It also provides inheritance mechanisms for contract inheritance, enabling the creation of hierarchical contract structures.

Events and Logging: Solidity allows the declaration of events within contracts, which emit notifications when specific actions occur. Events provide a way for contracts to communicate with external applications and enable the logging of important information on the blockchain.

Integration with Ethereum Ecosystem: Solidity integrates with various Ethereum tools and frameworks, making it easier for developers to build, test, and deploy smart contracts. Tools like Truffle, Remix, and Ganache provide a development environment and testing framework for Solidity

Solidity is a powerful language, but it necessitates careful consideration of security procedures and exhaustive testing to guarantee the dependability and accuracy of smart contracts. When writing Solidity code, developers should be aware of best practices and conduct audits to find any potential security holes. To sum up, on the Ethereum Blockchain, smart contracts can be developed using the programming language Solidity. It provides contract-oriented programming, support for the EVM, security features, and ecosystem integration. On the Ethereum platform, Solidity enables developers to build intricate, decentralized applications and carry out programmable, self-enforcing agreements

Facial Features Using Haar Cascade

Haar cascade Files: Machine learning models called Haar cascade files are used for object detection, especially for identifying particular features in photos. They have been extensively used for applications like face feature identification since they were initially proposed by Viola and Jones,[13] in 2001.The steps that are commonly taken while using Haar cascade files for face feature recognition are as follows:

Obtaining and Preparing Training Data: The collection of a sizable dataset of positive and negative samples is the first stage in creating a Haar cascade file. Negative samples are photographs lacking the desired feature, whereas positive samples are images with the desired feature (for example, faces). Poses, scales, and lighting variations should be present in the positive samples. The target feature's position is then determined by labelling this dataset.

Training the Cascade Classifier: After the training set of data is ready, the cascade classifier must be trained using the OpenCV library or tools of a similar nature. The positive and negative samples are fed into the classifier throughout the training phase, and the model's parameters are iteratively adjusted to reduce detection errors. This procedure may take a long time and require a lot of computing.

Generating Haar Cascade XML File: The learnt parameters of the cascade classifier are contained in the resultant

model, which is then stored as an XML file and used to generate the Haar Cascade XML file. The Haar cascade is represented in this XML file, which can be used to identify facial features.

Using Haar Cascade for Face Feature Detection: The XML file is imported into a computer vision library, such as OpenCV, to perform facial feature detection using the Haar cascade file. To apply the cascade classifier to an input picture or video stream, the library offers functions and APIs.

Detection Method: The input image is scanned using a sliding window technique using the Haar cascade at various scales. Sub regions of the image are subjected to the application of the Haar features specified in the cascade file at each scale. These Haar features, which are straightforward rectangular filters, are determined by comparing the pixel intensities in two different parts of the image. The cascade assesses if a specific region includes the target feature by comparing these characteristics' responses to predetermined thresholds.

Post-processing: Further post-processing processes might be used after the facial features have been identified. Filtering false positives, improving feature locations, and calculating face landmarks are a few examples of these steps.

Because they can quickly scan images and make conclusions, Haar cascade files are very good at detecting facial features. It's crucial to remember that they could not be as precise as more sophisticated deep learning-based methods, particularly when dealing with complex situations, position variations, or occlusions.[13]

In conclusion, Haar cascade files are applied to an input picture or video stream to detect the desired facial features by training a cascade classifier on positive and negative samples, creating an XML file with the learnt parameters, then applying the cascade.

Design & Implementation

The proposed model that We found feasible from review of literature and available methodology is discussed in this section.

Components of System

- Web App and Mobile App
- Application Server
- Smart Contracts
- SMS Gateway
- Foul proof Face ID

Web and Mobile Apps: The web app enables event organizers to design and plan fresh voting activities Within the Blockchain network, each voting event is represented by a different and unique Smart Contract. The administrator enroll the list of all candidates before starting an HTTP request to the server with the entered information. This Web and Mobile app's goal is to serve as an API interface for application programming, allowing anyone to create new voting events.

Server: The idea behind the Application Server is to publish the Smart Contract to blockchain network along with the data that was obtained from the web and mobile app. Hence, it needs a full Ethereum network interface node, an Ethereum wallet,[14] (address) needed to deploy the contract, and a db to store the list of addresses(contract) that will subsequently be queried by the app.

Smart Contracts:

Registration: This Contract is Deployed only once, whatever being the voting cases it cannot be repeated which makes sure an voter will be registered only once, same for the candidates.

Vote: It is written only once and deployed several times making the voting process happen, Data for the voting event is defined only by admin.

Using these two smart contracts Authors designed the voting process which can make a transparent, decentralized system keeping the anonymity of the process. Anonymity is achieved by never storing or mapping voter information with the recorded vote. Once the vote is recorded it will only keep the value of vote but no the information about the voter, by not sharing the same value.

SMS Gateway(MSISDN and OTP): It is only connected to the application server at the time of registration and voting It produces an OTP (One Time Password), Users have to enter the same once It is Received In their phone, It Increases the level of security and In most Cases since it is linked with AADHAAR like systems it can be a source for Biometric Details

Foul-Proof Face ID: A face recognition model is set up to verify that the voter who is registered with the voter ID is the one who really votes. This must be done to implement voting security. Yet, there is a drawback to conventional facial recognition methods. It is simple to spoof using a victim's image or video. We built a real-time liveness detection method to prevent this from happening, which necessitates the voter being alerted with a few random motions.(Smile, Turn left/right, blink the eyes etc.).

To cast a vote, each user must execute three random gestures. The user will not be able to vote until they sign in again if the gesture fails liveness detection.

FLOW CHART AND VOTING PROCESS

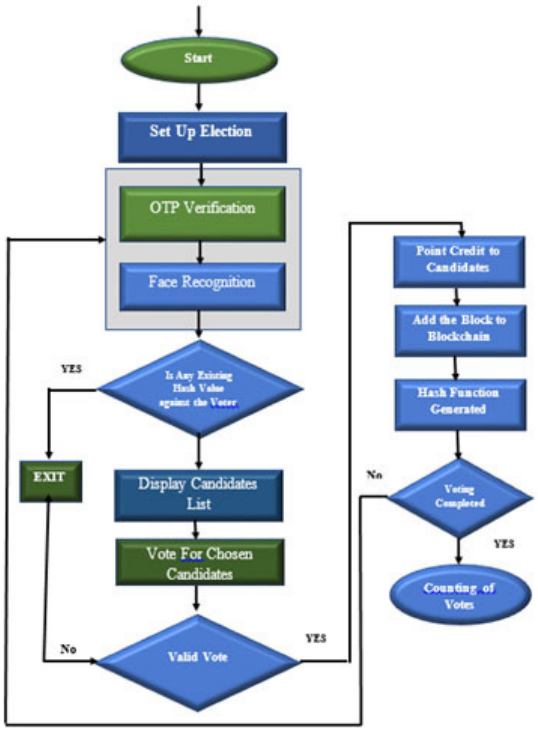

Figure 1. Flow chart of election process

Election Initialization: System is initialized with candidate and voter details, Election Time, Date etc

Authenticating Voters: Any ID card can be used to authenticate the user initially, the at the voting time user will have to use their OTP and the face detection for getting eligible to vote.

Hash Value already Existing Against any Voter's Name: If the voter's authentication is successful, a hash value will be looked up against the voter's name. The hash value will be provided if the voter has cast a vote; otherwise, there won't be one, If the hash value is present, the voter will not be allowed to vote again.[15]

Candidates Detail Displaying: A list of the candidates in the voter's constituency will be shown if they have not yet voted, Voter's ID number will be used to retrieve information about the constituency.[16]

Vote the Candidate of Choice: From the List Voter choose their choice of candidate.

Valid Vote: When all conditions are satisfied it is counted as a valid vote

Points Credited and Added Consequently to the Candidate: The candidate account will be updated once a legitimate vote has been cast. There is one attribute for each candidate that must be maintained in order to keep the votes coming in. This will make it easy to display the results.

Blockchain of Constituency added with new block: A block is formed for each legitimate vote that includes the voter's name, the recipient of their vote, the time the vote was cast, and the previous block's hash.

Generating Hash value Against that Voter's Name: A hash is generated against the voter's name once block is successfully added

Vote Recorded: This will verify whether or not the voting has been finished. The system continues to receive votes if it is not finished; if it is, it will continue by updating counter value of the candidates' account.

The most recent produced hash will be safely sent to the admin upon the end of voting: Getting the Value of Points from Blockchain (Created to Each Candidate)

The entire number of points given to each candidate will be used to calculate their vote total, and this information will be pulled from the blockchain used for electronic voting.

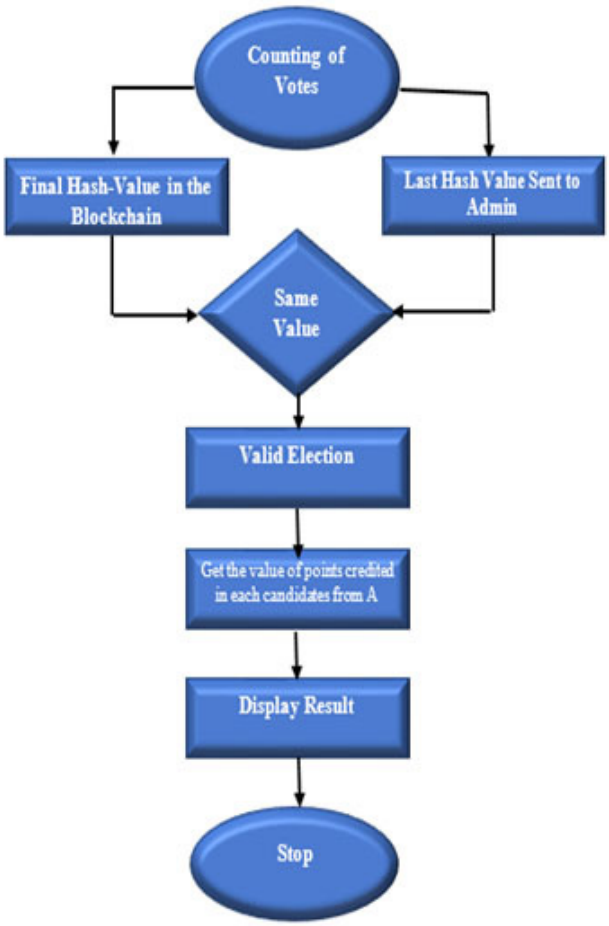

Figure 2. Flowchart Process of Counting

Result Display: After the counting of votes Result is displayed Different Visualization Techniques are also included for better communication

Prevention of Attack 51%: To Prevent attack 51% Method mentioned in A Framework to Make Voting System Transparent Using Blockchain Technology[16] is recommended.

Data Flow

Fig 3 depicts the Data Flow Diagram for the project. Voter registers into the system and receives a secret key as the receipt of successful registration. The admin has the privilege to update the candidates and other election parameters. Voters then uses this secret key as the identity to cast their vote. Vote is updated in the Blockchain through Smart Contracts which in turn depends on an Ethereum wallet for transaction. Miners at Ethereum, checks the blocks containing voting data and accepts the block if consensus is reached.

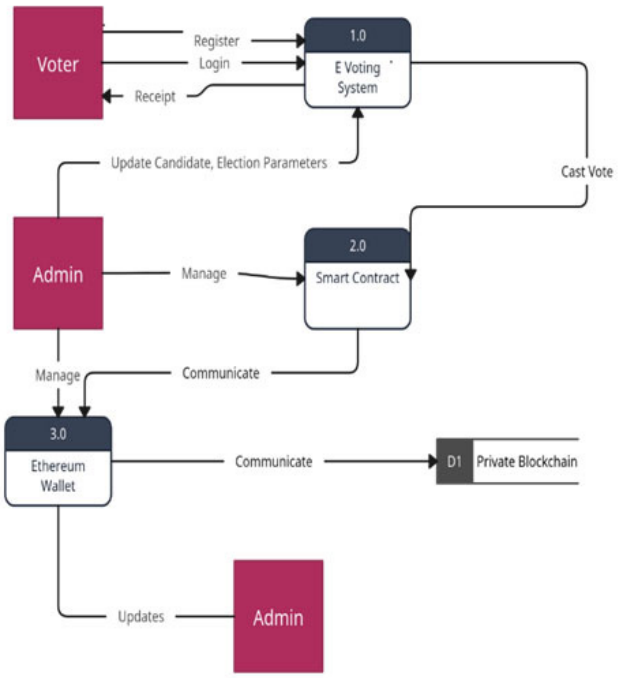

Figure 3. Data Flow Diagram

Implementation

Smart Contracts: A self-executing digital contract known as a "smart contract" automatically upholds and facilitates the conditions of an agreement between parties.[3] Agency created a smart contract using Solidity, which is the programming language of Ethereum. The smart contract contains functions for changes in blockchain. In Solidity, smart contracts can be made by basic codes. There are two types of transactions present, payable and non-payable for example; a transaction that modifies the blockchain's state by increasing the number of votes cast when a voter casts a ballot is a payable function, whereas a transaction that verifies the amount of votes cast is a non-payable transaction.

Voter Registration: In order to cast a valid vote, a user must first register with the system. Users must submit their name, voter ID, and constituency when registering. The programmer creates an Ethereum wallet for each user. Together with other functions, the register function is available. The web application makes use of these functions. A request is made to the server upon calling. The server has the ABI that connects the smart contract. The server performs the function using the ABI and the data from the request, and the blockchain is updated as a result.

As explained in proposed model no details of Voter are being recorded in this process it can only make a voter ineligible to vote again after one vote, since vote is not being mapped to any voter, this system keeps the anonymity of the voter at the same time keeping transparency

Compiling the Contracts: Authors used a new framework 'eth-brownie' for compiling and deploying smart contracts, Brownie is a popular development framework for Ethereum smart contracts. This makes creating, evaluating, and implementing smart contracts on the Ethereum Blockchain simpler. It is a Python-based development environment and it includes a number of tools and features to speed up the process of creating smart contracts.

Because of the same using Brownie Authors integrated solidity code with python, solidity running inside python made it easy to develop a web app as UI, etc.

Web App and Application Backend: Authors used Django 4 to structure application backend and connect it with a front app, which is HTML+CSS.

Django: A high-level Python web framework for creating web apps called Django makes the same easy. The Model-View-Controller (MVC) architectural pattern is used, and they Don't Repeat (DRY) principle is heavily emphasized. With the large range of tools and functionalities that Django offers, developers can concentrate on creating application logic rather than worrying about low-level implementation concerns. Django is a well-liked option for creating web applications due to its adaptability, thorough documentation, and active community support. It is frequently used to create many different kinds of applications, including social networks, e-commerce platforms, Content Management Systems (CMS), and more.

Face Authentication: As Discussed in Chapter 3, Haar Cascade Files are used to classify liveliness of face, then compare the face data with uploaded image, Once the user tries to vote camera will open and asks the voter to perform 3 random gestures, if it succeeds liveliness is confirmed and then the

facial data will be matched with registered photograph of user. Therefore, both liveliness and authenticity of the user confirmed and they can proceed to vote.

Library Used: Deep Face Library, A hybrid facial recognition framework, Deep Face library encases cutting-edge models including Google Face Net, Facebook Deep Face, Open Face, VGG-facial, Dlib, Deep ID, and Arc Face. On the LFW Dataset (Labelled Faces in the Wild), accuracy was 99.87%.

Results and discussions

To test the application first thing, authors need is a local blockchain; here authors have used one of the popular Ethereum Blockchain stimulator called ganache

As main objective of the authors to make system secure and reliable. Researchers created a successful Blockchain E-Voting system, which eliminates dual voting, and vote rigging. Apart from that, researchers also created a strong authentication system that detects identity fraud.

Figure 4. Liveliness Verification

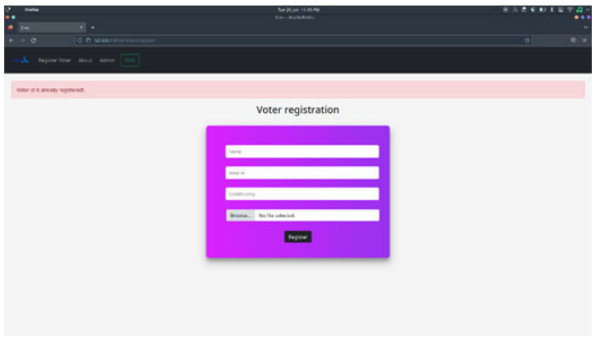

Figure 5. GUI of Vote E

Authors have tested the voting system for all scenarios like Repeated voter registration, Invalid Votes, Registration after starting election etc. and Voting System Found to be reliable in all Conditions.

Conclusion

In this paper, authors proposed and implemented a decentralized E-Voting System using Ethereum. Conventional voting system being outdated, expensive and slow it leads to different problems including wastage of resources and fall in polling percentage. At the same time existing E- Voting systems, which are centralized, and having many security issues cannot be considered as an alternative. That is where the need of a Voting System based on blockchain Comes up. Being on blockchain the voting system is Decentralized, Immutable, Transparent and Secure, the only question comes up with transparency is of anonymity, with algorithms Authors could ensure the anonymity of a voter. Included with a proper authentication system that ensures the liveliness and authenticity of voter, and considering data is immutable, this can be an efficient and better Voting System.

References

1. J A Samsul, & M B Limkar, "A biometric-secure cloud based e-voting system for election processes", International Journal of Electrical and Electronics Engineering Research (IJEEER), 4(2), 2014,145-152., Chicago.
2. A A A Aziz, H N Qunoo, & AAA Samra, " Using homomorphic cryptographic solutions on e-voting systems", International Journal of Computer Network and Information Security,2018, 12(1), 44
3. A Nadaph, R Bondre, A Katiyar, D Goswami, & T Naidu, "An implementation of secure online voting system", International journal of engineering research and general science, 3(2),2015, 1110-1118.
4. A Parmar, S Gada, T Loke, Y Jain, S Pathak, & S Patil, " Secure E-Voting System using Blockchain technology and authentication via Face recognition and Mobile OTP", In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) ,2021, pp. 1-5, IEEE.
5. O Kurbatov, P Kravchenko, N Poluyanenko, O Shapoval, & T Kuznetsova, "Using ring signatures for an anonymous e-voting system" In 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT),2019,pp. 187-190. IEEE
6. M S Farooq, U Iftikhar, &A Khelifi, "A framework to make voting system transparent using blockchain technology", IEEE Access, 2022, 10, 59959-59969.
7. G Rathee, R Iqbal, O Waqar, & A K Bashir, " On the design and implementation of a blockchain enabled e-voting application within iot-oriented smart cities", IEEE Access, 2019,9, 34165-34176.
8. A Baobaid, M Meribout, VK Tiwari, & J P Pena, " Hardware accelerators for real-time face recognition: A survey", 2022, pp.1-10,IEEE Access.
9. L Wang, X Ding, & C Fang, " Face live detection method based on physiological motion analysis", Tsinghua

Science & Technology, 2009, 14(6), 685-690.
10. S Nakamoto, “Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 2008, 1-6.
11. C Dannen, “ Introducing Ethereum and solidity”, Vol. 1, 2017, pp. 159-160, Berkeley: Apress.
12. G Wood, “Ethereum: A secure decentralised generalised transaction ledger”, Ethereum project yellow paper, 151(2014), 1-32.
13. A Atri, A Bansal, M Khari, & S Vimal, “De-CAPTCHA: A novel DFS based approach to solve CAPTCHA”, schemes. Computers & Electrical Engineering, 97, 2022,1-10.
14. M Khari, M Kumar, S Vij, & P Pandey, “ Internet of Things: Proposed security aspects for digitizing the world”, In 2016 3rd international conference on computing for sustainable global development (INDIACom) ,2016,pp. 2165-2170). IEEE.
15. P Viola, & M Jones, “Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR, 2001,Vol. 1, 2001,pp. I-I0. Ieee.
16. N Gailly, P Jovanovic, B Ford, J Lukasiewicz, & L Gammar, “Agora: bringing our voting systems into the 21st century”,2018,pp.1-6.