A BLOCKCHAIN BASED VOTING SYSTEM FOR PUBLIC OPINION

A Project paper
Course code: CSE 4292
Course Title: Thesis/ Project (Part II)

Submitted for the partial fulfillment of the requirements for the degree of B.Sc. (Eng.) in Computer Science and Engineering

Submitted By-

ID: 1837620102
Session: 2017-18

ID: 1837620104
Session: 2017-18

ID:1837620109
Session: 2017-18

Department of Computer Science and Engineering
Rangpur Engineering College, Rangpur
(Affiliated by University of Rajshahi)

ACKNOWLEDGEMENT

It is a pleasure acknowledging with deep sense of gratitude the invaluable help and guidance the invaluable help and guidance rendered by Supervisor and Faculty of our Institute for the successful completion of this project report.

We express our deep gratitude to our project supervisor Md. Khalid Hasan, Project Coordinator for his valuable guidance and suggestion throughout my project work.
Who never hesitated in extending their sincere cooperation from time to time to prepare an interesting and challenging project.
We would also like to thank our families and friends, who helped us in every possible way.

Signature of Supervisor

ABSTRACT

In any democratic country, Voting is a fundamental right of any citizen that enables them to choose the leaders of tomorrow. It gives individuals in a community the facility to voice their opinion. It helps them to realize the importance of citizenship. Online voting systems are software platforms used to securely conduct votes and elections. As a digital platform, they eliminate the need to cast their votes using paper or having to gather in person. They also protect the integrity of your vote by preventing voters from being able to vote multiple times.

Electronic voting or e-voting has fundamental benefits over paper-based systems such as increased efficiency and reduced errors. The electronic voting system tends to maximize user participation, by allowing them to vote from anywhere and from any device that has an internet connection. The blockchain is an emerging, decentralized, and distributed technology with strong cryptographic foundations that promises to improve different aspects of many industries. Expanding e-voting into blockchain technology could be the solution to alleviate the present concerns in e-voting. Here we propose a blockchain-based voting system that will limit the voting fraud and make the voting process simple, secure and efficient.

TABLE OF CONTENT

List of Figure

CHAPTER-1 INTRODUCTION

Democracy is the fundamental right of people to select their leaders through a transparent and impartial electoral system. Voting plays a crucial role in empowering citizens to elect their government representatives. For an electoral system to be considered democratic, it must be independent, fair, and secure, allowing individuals to freely express their viewpoints. Unfortunately, the conventional voting process faces challenges that erode public trust.

These challenges include the presence of intermediaries and lack of control booth capture, dummy voting, and monitoring problems. Long queues, distance to polling booths, false voting, pre-vote casting, redundant voting, and lack of law enforcement and audits further add to the concerns. Moreover, political instability, inadequate awareness, and difficulties faced by older people in voting contribute to reduced voter participation. To enhance the electoral process's credibility, various reforms, such as electronic voting, improved accessibility, and awareness campaigns, need to be considered. Strengthening law enforcement and audit procedures is also vital to prevent electoral malpractices and ensure a more reliable democratic process.

Digital Voting also has certain drawbacks. The secrecy of significant portions of the code is one of the main criticisms of electronic voting systems in Estonia and Norway. The format for the ballot on the Estonian I-Voting system is restricted due to various confidentiality concerns. The centralization of power of the IV thing enables DDOS attacks susceptible, which will allow electoral elections unavailable to voters.People who vote could question the fairness and confidentiality of the voting process. Police and security services have access to network traffic’s variety and processing capacity to examine polling data for possible alterations. System attacks are still likely in all previous schemes, even though security is strengthened .Some enhanced security schemes or processes must also ensure that voting or measuring procedures are reliable and the above listed issues are avoided.

1.1 EXISTING SYSTEM

The current election system operates manually, requiring voters to physically visit polling booths to cast their votes, resulting in significant time wastage. Consequently, many individuals refrain from participating in the voting process, a worrisome issue considering the importance of each vote in a democratic society. To address these concerns and enhance the integrity and efficiency of voting and counting, a modern online system can be implemented. Such an online platform would minimize voting frauds while promoting transparency and efficiency in the electoral process.

Proposed System:

The current voting system requires some improvement in it because of the issues mentioned above. This can be achieved by replacing the existing system by the new system which will limit the voting frauds and make the voting as well as counting more efficient.

- Online Election System would have user registration, user login and admin login.
- This Online Voting System will manage the Voter's information by which voter can login and use his voting rights.
- At the time of registration voter will be asked for this: Full name, age, National Identity Card no, mobile no. email id and after being verified will be given the access.
- At the time of requesting vote, voter will be asked to enter his National Identity Card . Then voter will be authenticated, and he can give vote from one of the candidate from the list .Voters can vote for a Candidate only once per Election.
- The software system allows the user to login in to their profiles and upload all their details including their previous milestone onto the system. The admin can check each Candidate details.
- The software system also allows Voters to view a list of Candidates in their area. The admin has overall rights over the system and can moderate and delete any details not pertaining to Election Rules.

1.2.1 BLOCKCHAIN [1]

Blockchain can help to implement a system that is immutable, transparent, and efficient and cannot be hacked into. The inability to change or delete information from blocks makes the blockchain the most effective technology for voting systems. Blockchain technology is supported by a distributed network consisting of variety of interconnected nodes. Each of these nodes have their own copy of the distributed ledger (information) that contains the total history of all transactions the network has processed. There is no centralized system that controls the network. If the majority of the nodes agree, then they accept a transaction. This network permits users to stay

anonymous. A basic analysis of the blockchain technology (including sensible contracts) suggests that it is an appropriate basis for e-voting and furthermore, it might have the potential to form e-voting a lot of acceptable and reliable.

Blockchain technology makes e-voting cheaper, easier, and much more secure to implement. It is a considerably new paradigm that can help to form decentralized systems, which assure the data integrity, availability, and fault tolerance. This technology aims to revolutionize the systems. The blockchain systems are formed as decentralized networked systems of computers, which are used for validating and recording the pure online transactions. They also constitute ledgers, where digital data is tied to each other, called the blockchain. The records on the blockchain are essentially immutable.

1.2.2 WHAT IS SOLIDITY? [2]

Solidity is a specialized programming language created for developing smart contracts on blockchain platforms, with a primary focus on Ethereum. It serves as a high-level, contract-oriented, and statically-typed language used to build decentralized applications (DApps) and implement smart contracts on the Ethereum blockchain. Smart contracts are self-executing agreements whose terms are directly written into code. They operate on the Ethereum Virtual Machine (EVM), a decentralized virtual machine that executes code across a distributed network of computers. Solidity equips developers with the necessary tools to define the rules and logic of their smart contracts, enabling the automation of various processes without the need for intermediaries. This capability allows for the creation of a wide range of decentralized applications, including decentralized finance (DeFi) protocols, non-fungible tokens (NFTs), tokenized assets, and more. Key features of Solidity include its contract-oriented nature, support for static typing, seamless integration with the Ethereum blockchain, and consideration for security aspects. The language offers access control modifiers and contract-specific functions to manage security and prevent vulnerabilities such as reentrancy and integer overflow.

1.2.3 WHAT IS REACT? [3]

- ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.
- A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The

components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

- Instead of using regular JavaScript, React codes are written in something called JSX (JavaScript Syntax Extension). JSX is basically a syntax extension of regular JavaScript and is used to create React elements. These elements are then rendered to the React DOM. JSX is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript.

1.2.3.1 WHY USE REACT?

- Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
- Can be used on client and server side as well as with other frameworks.
- Component and data patterns improve readability, which helps to maintain larger apps.

1.2.4 WHAT IS TYPESCRIPT? [4]

TypeScript is a programming language developed by Microsoft in 2012 that extends JavaScript by introducing optional static typing. It is often referred to as a "superset" of JavaScript because any valid JavaScript code is also valid TypeScript code.

The main goal of TypeScript is to provide a more structured and robust development experience for large-scale applications. JavaScript is dynamically-typed, meaning variable types are determined at runtime, which can lead to potential runtime errors that are only detected during execution.

TypeScript introduces static typing, allowing developers to specify data types for variables, function parameters, and return values during development. This enables the TypeScript compiler to catch type-related errors early in the development process, improving code quality and reliability.

1.2.5 WHAT IS NODE JS? [5]

- Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications.

Node.js is open source, completely free, and used by thousands of developers around the world.

- Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.
- Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS
- X, Microsoft Windows, and Linux.
- Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

1.2.5.1 FEATURES OF NODE JS?

1. Extremely fast: Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.

2. I/O is Asynchronous and Event Driven: All APIs of Node.js library are asynchronous i.e. non-blocking. So, a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.

3. Single threaded: Node.js follows a single threaded model with event looping.

4. Highly Scalable: Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.

5. No buffering: Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.

6. Open source: Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js application

1.2.6 WHAT IS MySQL? [6]

- Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like

video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

- Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.
- Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS
- X, Microsoft Windows, and Linux.
- Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Benefits of E-voting system over the current system:

1. Increasing the level of participation

The Internet voting system tends to **maximize user participation**, by allowing them to **vote from anywhere** and **from any device** that has an internet connection.

2. Security

By considering the importance of the e-voting system is implemented using "Blockchain".

3. Efficiency

The **reduction** in organizational and implementation **costs** significantly increases the efficiency of election management compared to traditional paper voting, for example.

4. Precision

The electronic vote eliminates errors in manual count, which brings with it an **accurate and quick publication of results**, with receipt of vote for each vote cast.

Proposed Plan of Works:

For our proposed plan of work we are considering two modules that are to be completed in three phases. Two modules are as follows:

1. Front-end for the application

2. Back-end using Solidity to implement Blockchain.

Each of these module will be considered as one phase and the remaining one phase will cover the connection and testing of these modules.

- Phase 1: In this phase we will cover the front-end module, in which we will build the interactive user-interface for the admin as well as the user. In parallel the research work related to the implementation of Blockchain in decentralized application will be done.

- Phase 2: In this phase we will cover the back-end module, we will implement the Blockchain using Ethereum framework and convert the system into a decentralized application.

- Phase 3: The connection of two different module along with the testing of the platform will be completed in this phase.

CHAPTER -2 LITERATURE SURVEY

Currently increasing digital technology helped many people lives. In contrast to the electoral system, there are many conventional uses of paper in its implementation. The aspect of security and transparency is a threat from still widespread election with the conventional system. Block chain technology is one of solutions, because it embraces a decentralized system and the entire database are owned by many users.

There is no doubt that the revolutionary concept of the blockchain, which is the underlying technology behind the famous crypto currency Bit coin and its successors, is triggering the start of a new era in the Internet and the online services. In this work, we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language. Block chain was first introduced by Satoshi Nakamoto (a pseudonym), who proposed a peer to-peer payment system that allows cash transactions through the Internet without relying on trust or the need for a financial institution. Block chain is secure by design, and an example of a system with a high byzantine failure tolerance.

E-voting is a potential solution to the lack of interest in voting amongst the young tech savvy population. For e-voting to become more open, transparent, and independently auditable, a potential solution would be base it on block chain technology. Block chain technology has a lot of promise; however, in its current state it might not reach its full potential.

Electronic voting has been used in varying forms since 1970s with fundamental benefits over paper-based systems such as increased efficiency and reduced errors. With the extraordinary growth in the use of block chain technologies, a number of initiatives have been made to explore the feasibility of using block chain to aid an effective solution to e-voting. It presented one such effort which leverages benefits of block chain such as cryptographic foundations and transparency to achieve an effective solution to e-voting. The proposed approach has been implemented with Multichain and in-depth evaluation of approach highlights its effectiveness with respect to achieving fundamental requirements for an e-voting scheme.

Recent major technical challenges relating to e-voting systems embrace, however not restricted to secure digital identity management. Any potential citizen ought to be registered to the electoral system before the elections. Their data ought to

be in a very digitally processable format. Besides, their identity data ought to be unbroken personal in any involving information. Ancient E-voting system could face following problems:

• Anonymous vote-casting.

• Individualized ballot processes.

• Ballot casting verifiability by (and only by) the voter.

• High initial setup costs.

• Increasing security problems.

• Lack of transparency and trust.

• Voting delays or inefficiencies related to remote/absentee voting

To mitigate these threats, software mechanisms which promise the following should be deployed:

1. Prevention of evidence deletion.

2. Transparency with privacy.

Using a Blockchain, the most important requirements are satisfied:

• Authentication: Only registered voters will be allowed to vote.

• Anonymity: The system prevents any interaction between the votes casted by the voters and their identities.

• Accuracy: Votes once cast are permanently recorded and cannot be modified or changed under any circumstances.

• Verifiability: The system will be verifiable such that the number of votes is accounted for

As technology advances, many countries have now opted for electronic voting systems. Any voting system must follow principles of transparency and impartiality in order to achieve fairness; the electronic voting process must also be protected against cyberattacks or denial-of-service attacks (DDOS) because such attacks may affect the processing time in voting procedures and even hinder the fairness in voting. This study establishes a network security mechanism for voting systems based on blockchain technology. The blockchain mechanism employs a distributed architecture that can prevent system shutdown resulting from malicious cyber-attacks; additionally, any user in the blockchain can authenticate data integrity, which satisfies requirements of transparency and impartiality in voting systems.

CHAPTER-3 WORK DONE

3.0 Proposed Plan of Works:

For our proposed plan of work we are considering two modules that are to be completed in three phases. Two modules are as follows:

1. Front-end for the application
2. Back-end using Solidity to implement Blockchain.

Each of these module will be considered as one phase and the remaining one phase will cover the connection and testing of these modules.

- Phase 1: In this phase we will cover the front-end module, in which we will build the interactive user-interface for the admin as well as the user. In parallel the research work related to the implementation of Blockchain in decentralized application will be done.

- Phase 2: In this phase we will cover the back-end module, we will implement the Blockchain using Ethereum framework and convert the system into a decentralized application.

- Phase 3: The connection of two different module along with the testing of the platform will be completed in this phase.

3.1 Division of Phase One:

We have considered 2 main modules which are as follows:

A. Admin- The admin module is divided into 5 components-
    1. Dashboard-It will contain various charts to display information such as number of parties, number of voters etc.
    2. Add Candidate - In this feature of admin, he can add candidates who are standing in the election. After candidate is added it will be displayed on the user side.
    3. Create Election- This feature of admin will allow him to create election. A user can cast his vote only after the election is created by admin. A user can cast vote between the start date and end date.

4. Election Details- In this section admin can update election details such as start date, end date etc.
5. Candidate Details- In candidate details all the candidates added by admin will be displayed. Admin can update the candidate details if incase a wrong entry is done.

B. User- The user module is divided into 4 components

1. Dashboard- The user dashboard contains information about parties and their candidates. A user can see all the information about candidate.
2. Voter Register- In this section first user will have to register himself only then he will be able to cast his vote.
3. Voting Area- After user is registered, then only he will be directed to this page and then he can cast his vote.
4. Results- In this component the user will be able to see the results of the election.

3.1.1 Phase One Flow Diagram

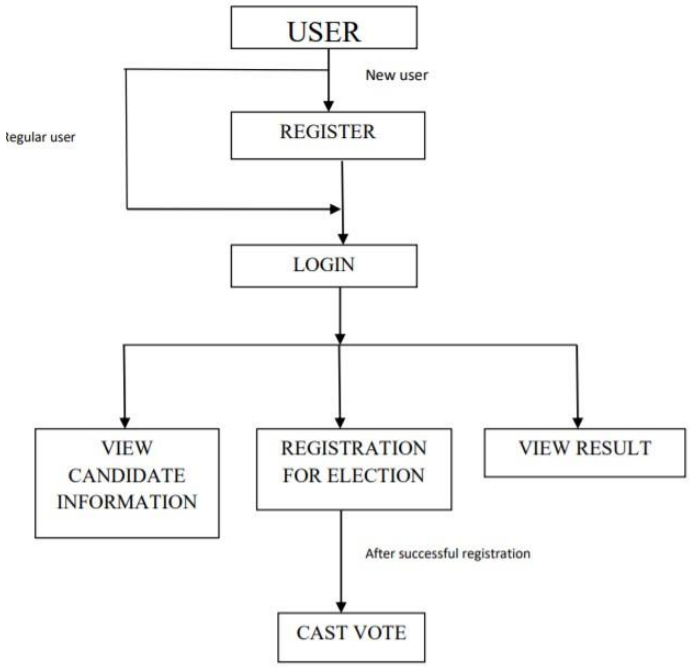

Figure-3.1 User flow diagram

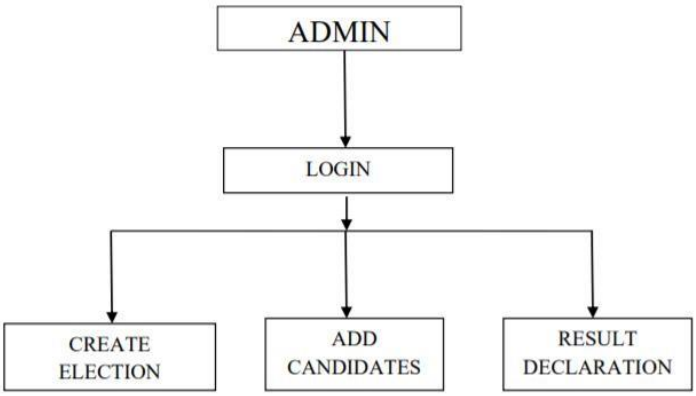

Figure-3.2 Admin flow diagram

3.2 Research Methodology of Phase Two

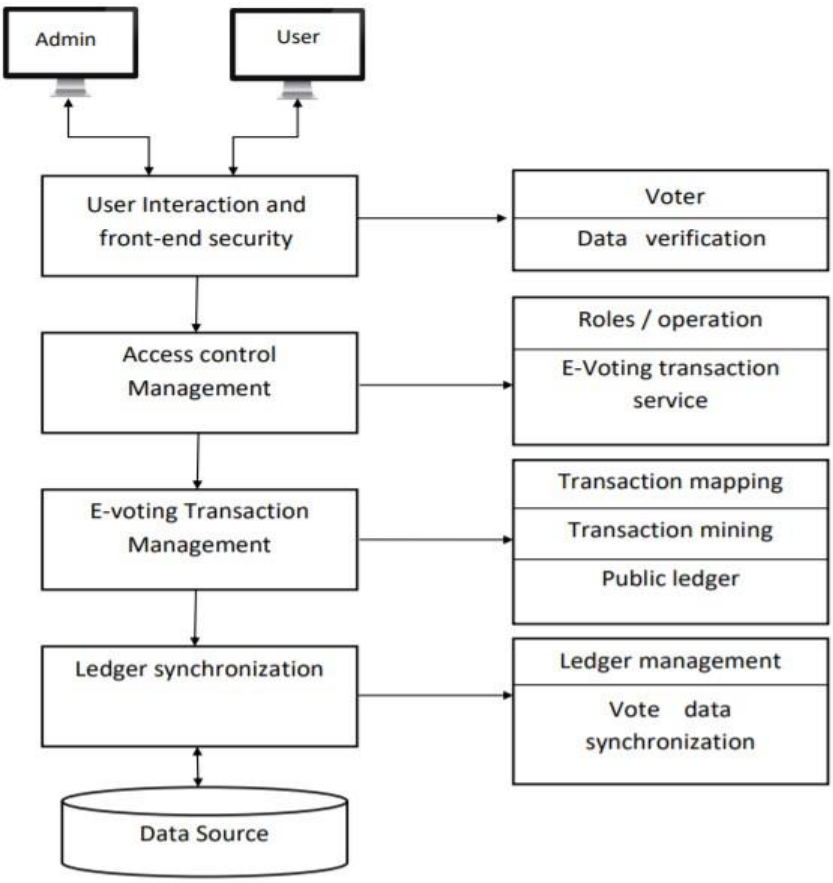

Figure-3.3 Research methodology

CHAPTER-4 IMPLEMENTED ALGORITHM

4.1 Voter Registration

Algorithm1: Voter Registration

Input: id, nm_v, Mb_v, sk
Output: hv, pub_{kv}, pv_{kv}
1 Input id
2 **if** *IDValidation(id)* **then**
3 Input nm_v, Mb_v, sk
4 hv = generateHash(id,nm_v, Mb_v, sk)
5 goto step 8
6 **else**
7 Return "Invalid NID".

People who have the right to vote and are registered to vote in their local election district are called voters. The Election Commission provides and maintains an up-to-date list of registered voters. As a result, every eligible voter must visit their local voter registration center and provide the necessary information to be recognized as a genuine voter. It is the first stage in the system and is needed as part of the identity verification phase in keeping a track of which individuals have casted vote. It also serves as a control mechanism to prevent unregistered individuals from participating in the election by preventing them from casting a ballot.
Algorithm1 illustrates the voter registration process. ID, voter name, security key, and voter's mobile number are the inputs of this algorithm. The outputs are the hash value of these inputs and two keys named public key and private key. validation() function is used to check the validation of NID at step 2. If the NID is not valid, then Invalid NID will be returned at step 7. If the validation of NID is true, then voters input their credentials in step 3. If all the inputs are taken correctly, then a hash value will generate using the generateHash() function in step 4, where the inputs are the parameters.

4.2 Candidate Registration

Since a candidate is also a voter, the candidate registration procedure is similar to voter registration. They must complete several additional steps following key generating in order to be considered a candidate. The full process of voter and candidate registration process is shown in Fig. 3.

Algorithm2: Candidate Registration

Input: id, nm, Mb_v,sk,rg,sn,ps
Output: Cj_{info}
1 **if** *(IsVoter(*id, nm, Mb_v,sk*))* **then**
2 Input rg,ps,sn
3 SC_{cdt} add Cj_{info} in BC
4 Return Registration Complete Successfully
5 **else**
6 Call *voterRegistration* (id, nm, Mb_v,sk)

4.3 Voting Setup Phase

This phase is divided into three parts:

(i) Create Election
(ii) Active Election

4.3.1. Create Election

The election is created by EC. Algorithm3 illustrates the election creation process by EC. EC joins Blockchain using a key pair of public keys and private keys. Then it sends a transaction to the registration contract with n vote coin, starting and ending time of election at step 2. Then transfer the keypair of public key and private key to the crypto server for vote encryption and decryption.

Algorithm3: Create Election

Input: vc, st, et
1 Join BC using $pub_{k_{Ec}}$ and $pv_{k_{Ec}}$
2 Send a tx =(vc_n, s_t, e_t) to SC_v
3 TransferKey($pub_{k_{Ec}}, pv_{k_{Ec}}$) to CS

4.3.2 Active Election

Algorithm4 illustrates the election activation process where the inputs are: starting time, ending time ,vote coin, and public key of voters. Voter contract sends a transaction of 1 vote coin, starting time, and ending time of election to each voter's public key. All the transactions are added to Block Chain. So the transactions of sending vote coin to the voters and voter's voting status are not hidden in this system.

Algorithm4: Active Election

Input: st, et, vc, pub_{kv}
Output: vc
1 SC_v create a tx =$(vc = 1)$
2 Send tx to V_i's pub_{kv}
3 tx's are added into BC

4.4 Voter Authentication

The Voter contract is in charge of the voter authentication procedure. Voters must first sign into their wallets using the private key in order to complete the authentication procedure. After that, the voter must enter their credentials for authentication. In this case, the voter contract receives the credentials and generates a hash value from them in order to compare the hash value with other hash values already present in the blockchain. If both hash values are found equal, the voter is valid for voting.

Algorithm5: Voter Authentication

Input: id, nm, Mb_v, sk
Output: hv
1 $hv = \text{generateHash}(id, nm, Mb_v, sk)$
2 **if** *(Hashmatch(hv)* **then**
3 Return"True"
4 **else**
5 Return "False"

4.5 Vote Counting

Publish Result After voting, every vote will form a block and add it to the chain. The vote will be counted instantaneously after the vote is submitted, as there will be no risk of vote tampering and vote manipulation

Algorithm6: Vote Counting

Input: EB_i, pvk_{Ec}
Output: C_{acj}
1 $B_i = decryptBallot(EB_i, pvk_{Ec})$
2 $tvc = sendVoteCoin(B_i, C_i)$
3 $C_{acj} = countVote(tvc)$

CHAPTER-5 USER INTERFACE

The existing system for voting in any democratic country like the EVM (Electronic Voting Machine) based system. Prior to this system there were paper ballots and manual counting. The paper ballots method was widely criticized because of fraudulent voting and booth capturing, where party loyalists captured booths and stuffed them with pre-filled fake ballots. Replacing the existing system with a new election system is critical to limit fraud and having the voting process traceable and verifiable. As we can see the internet has brought a revolution in each and every domain possible, by trying to shift the existing system towards the online platform to make the proceedings fast and user convenient.

We know that the existing system is reliable but that does not mean that we should not take a step ahead towards the betterment of the existing system. Online voting system using blockchain can be that step.

A blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology works through four main features:

- The ledger exists in many different locations: No single point of failure in the maintenance of the distributed ledger.
- There is distributed control over who can append new transactions to the ledger.
- Any proposed “new block” to the ledger must reference the previous version of the ledger, creating an immutable chain from where the blockchain gets its name, and thus preventing tampering with the integrity of previous entries.

We have tried to build a user interface of the decentralized application that will make the voting process more convenient. The UI portion can be assumed as 30% of the total work that is to be completed. The major portion in this system is related to the blockchain technology which will be responsible for the reliable voting process.

UI of Website

Figure-4.1 Homepage

- The homepage consists of 3 options –
  1. First option Login.
  2. Second option is for user signup/login.
  3. Third is for voting laws.

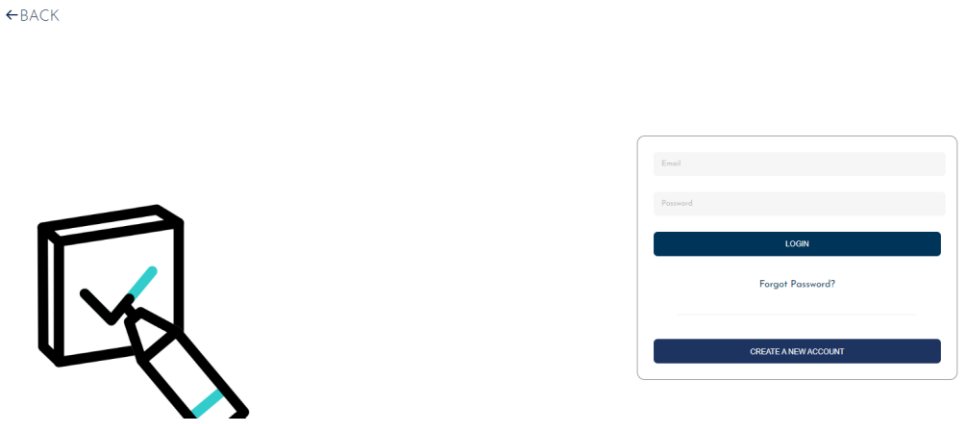

Figure-4.2 Login

- This is the login page for admin.
  After admin is logged in he is directed to the dashboard

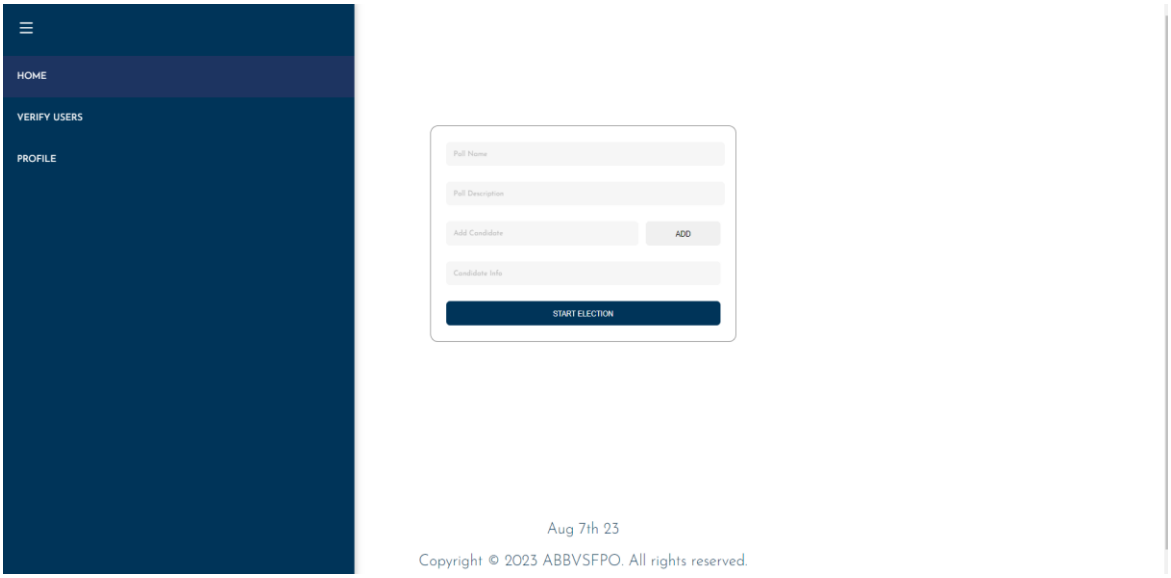

Figure-4.3 Admin Dashboard

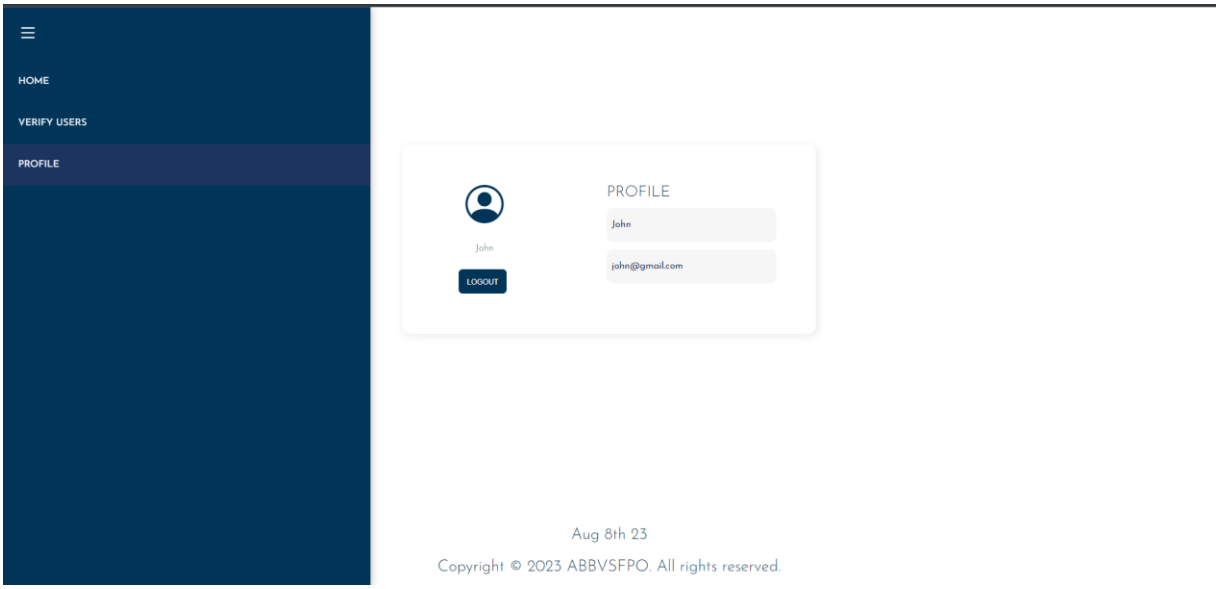

Figure-4.4 Profile Page

Figure-4.5 Create Election Page

- This feature of admin will allow him to create election.
- A user can cast his vote only after the election is created by admin.
- A user can cast vote between the start date and end date.

Figure-4.6 Candidate Vote Section

In candidate vote section, all the candidates are able to vote.

Figure-4.7 Result

CHAPTER-6 TESTING

6.1 Testing [10]

Testing is the set of activities that can be planned in advance and conducted systematically. Numbers of testing strategies are proposed. All provide software developer with a template for testing and all have following characteristics.

Testing begins at component level & works “outward” towards the integration of the entire computer based system.

- Different testing techniques are appropriate at different points in time.
- Testing is conducted by the developer of the software & independent test group.
- Testing & debugging are different activities, but debugging must be accommodated in any testing strategy.

6.2 SYSTEM TESTING

6.2.1 Recovery Testing

Recovery testing is a system test that enforces the software to fail in a variety of ways and verifies that recovery is properly performed. If the recovery is automatic, re-initialization, check pointing mechanism and data recovery and restart are each evaluated for correctness. If recovery requires human intervention, the mean time to repair is evaluated to determine whether it is within acceptable limits.

6.2.2 Security Testing

Security testing attempts to verify that protection mechanisms built into a system will infract protect it from improper penetration.

6.2.3 Stress Testing

Stress tests are designed to handle programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume.

CHAPTER-7 FUTURE WORK & CONCLUSION

7.1 NEXT TO ABBVSFPO

In future we will add some new features that can enhance its functionality and user experience. Here are some new feature ideas:

i. Smart Contract Upgradeability:

Implement a mechanism to upgrade the smart contract logic while preserving the existing data and state. This will allow you to improve and fix any bugs in the smart contract without disrupting the ongoing or past elections.

ii. Token-Based Voting:

Introduce a native utility token that users must hold to be eligible to vote. This token-based voting system can incentivize more participation and engagement in the voting process.

iii. Delegated Voting:

Implement a delegated voting mechanism, where voters can delegate their voting power to trusted representatives or individuals who vote on their behalf. This feature can increase voter engagement and make it more convenient for some users.

iv. Anonymous Voting:

Add a feature that ensures voter anonymity by using cryptographic techniques to protect voters' identities while maintaining the integrity of the voting process.

v. Multi-language Support:

Allow users to interact with the voting system in multiple languages to make it more accessible to a diverse user base.

vi. Mobile App Integration:

Create a mobile app for your voting system to make it easier for users to vote on the go and access election results from their smartphones.

vii. Real-time Election Monitoring:

Implement a dashboard that displays real-time election updates, results, and statistics to keep users informed and engaged during the voting process.

viii. Secure Voter Verification:

Develop a secure and decentralized method for verifying the identity of voters to prevent fraudulent voting while preserving privacy.

ix. Proxy Re-encryption for Result Decryption:

Enhance result decryption using proxy re-encryption, a technique that allows authorized parties to decrypt encrypted data without exposing the original private keys.

x. Automatic Voter Registration:

Allow automatic voter registration based on specific criteria to ensure eligible voters are included without manual sign-up processes.

7.2 CONCLUSION

In this project, we introduced a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient election while guaranteeing voters privacy. Blockchain technology offers a new possibility to overcome the limitations and adoption barriers of electronic voting systems which ensures the election security and integrity and lays the ground for transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain. To achieve this we have divided this overall work into three modules.

1. The front-end module
2. The back-end module
3. Connection of two different module along with the testing.

Until now we have completed our first module, in which we created the interactive user interface for the admin as well as the user. We have also done some research work related to the implementation of Blockchain in decentralized application.

And In the future to make the voting process more secure and to correctly identify the person who is voting we can use ML and AI concepts. Using these concepts we can verify whether the person voting is the same as the person who has registered during the registration process.

REFERENCES

[1] Block chain.

Available : https://www.blockchain.com/

[2] Solidity and its working princple .

Available : https://docs.soliditylang.org/en/v0.8.21/

[3] What Is React JS?

Available: https://legacy.reactjs.org/docs/getting-started.html

[4] What is TypeScript?

Available:https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html

[5] Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Available: https://nodejs.org/en/

[6] What is MySQL?

Available: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

[7] Why You Should Use React.js For Web Development, Author: Suraj Surve

Available:https://www.freecodecamp.org/news/why-use-react-for-web-development/

[8]Alan Dennis, Barbara Haley Wixom, and Roberta M. Roth, Systems Analysis and Design, 5th Edition, November, 2014. System design lifecycle.

[9]Getting Started - About Version Control, Local Version Control Systems,

Available:https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

[10] A Review on Web Application Testing and its Current Research Directions. Available:https://www.researchgate.net/publication/320248662_A_Review_on_Web_Application_Testing_and_its_Current_Research_Directions