

E-voting System using Blockchain

Submitted in partial fulfillment of the requirements
of the degree of

BACHELOR OF ENGINEERING (Computer Engineering)

by

- | | |
|--------------------------------------|-------------------|
| 1. Ms. Kalyani S. Singh | (BE/A/107) |
| 2. Ms. Pratiksha R. Jondhale | (BE/A/112) |
| 3. Ms. Srushti A. Suryavanshi | (BE/A/119) |
| 4. Mr. Kishan R. Gupta | (BE/A/140) |

Guide

Prof. Sulbha S. Yadav



**Department of Computer Engineering
Lokmanya Tilak College of Engineering
Sector-4, Koparkhairne, Navi Mumbai
(2022-2023)**

Certificate

This is to certify that the project entitled “**E-voting System using Blockchain**” is a bonafide work of

1.Ms. Kalyani S. Singh (BE-A-107)

2. Ms. Pratiksha R. Jondhale (BE-A-112)

3. Ms.Srushti A. Suryavanshi (BE-A-119)

4. Mr. Kishan R. Gupta (BE-A-140)

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Computer Engineering**”.

(Prof. Sulbha S. Yadav)

Guide

()

Co-Guide/External guide

(Prof. Sonal Bankar)

Head of Department

(Dr. Vivek Sunnapwar)

Principal

Project Report Approval for B.E.

The project report entitled “E-voting System using Blockchain” by **Ms. Kalyani S. Singh (BE/A/107)**, **Ms. Pratiksha R. Jondhale (BE/A/112)**, **Srushti A. Suryavanshi (BE/A/119)**, **Mr. Kishan R. Gupta (BE/A/140)** is approved for the award of “Bachelor of Engineering” degree in “Computer Engineering”.

Examiners

1.

2.

Date:

Place: Koparkhairane, Navi Mumbai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principals of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data / fact / source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- 1. Kalyani S. Singh (107)**
- 2. Pratiksha R. Jondhale (112)**
- 3. Srushti A. Suryavanshi (119)**
- 4. Kishan R. Gupta (140)**

Date:

ABSTRACT

Building a secure electronic voting system that offers the fairness and privacy of current voting schemes, while providing the transparency and flexibility offered by electronic systems has been a challenge for a long time. In this project we evaluate an application of blockchain as a service to implement distributed electronic voting systems.

The project proposes a novel electronic voting system based on blockchain that addresses some of the limitations in existing systems and evaluates some of the popular blockchain frameworks for the purpose of constructing a blockchain-based e-voting system. In particular, we evaluate the potential of distributed ledger technologies through the description of a case study; namely, the process of an election, and the implementation of a blockchain-based application, which improves the security and decreases the cost of hosting a nationwide election. 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 your 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.

ACKNOWLEDGEMENT

We remain immensely obliged to **Prof. Sulbha S. Yadav** for providing us with the idea of the topic, for her invaluable support in gathering resources, her guidance and supervision which made this work successful.

We would like to give my thanks to the Head of the Computer Department, **Prof. Sonal Bankar, Vice Principal Dr. Subhash Shinde and our Principal Dr. Vivek Sunnapwar.**

We are also thankful to the faculty and staff of the Computer Engineering Department and Lokmanya Tilak College of Engineering, Navi Mumbai for their invaluable support.

We would like to say that it has indeed been a fulfilling experience working out this project topic.

LIST OF FIGURES

S. No	Figures	Page No.
1.	Working of Blockchain – An overview	16
2.	Design of Proposed System	19
3.	Methodology	21
4.	Code block to define a smart contract- election smart contract	23
5.	Code block of candidate definition in smart contract	24
6.	Code block to define the voters in the smart contract	25
7.	Code block to define the migration smart contract	26
8.	RSA algorithm- Encryption	27
9.	Transaction initiation by registering a new user	30
10.	Ganache accounts tab	30
11.	Ganache blocks tab	31
12.	Block details of contract creation	32
13.	Block details of contract call	33
14.	Ganache Transaction tab	34

15.	Registration Page	35
16.	Login page	35
17.	Creating election page	36
18.	User verification page	36
19.	Initial result page	37
20.	Final result page	37

TABLE OF CONTENTS

Abstract.....	I
Acknowledgement.....	II
List of figures	III
Table of contents.....	V

Chapter 1. Introduction

1.1 Introduction.....	2
1.2 Motivation.....	3

Chapter 2. Literature Survey

2.1 Survey of Existing System.....	6
2.2 Summarized Findings or Researchgaps.....	8

Chapter 3. Proposed System

3.1 Problem Statement and Objectives.....	12
3.2 Scope of the Work.....	14
3.3 Analysis/Framework/ Algorithm.....	15
3.4 Details of Hardware & Software.....	18
3.5 Design details.....	19
3.4 Methodology	21
3.5 Implementation.....	22

Chapter 4. Results

4.1 Results.....	30
4.2 Conclusion.....	38
4.3 Future scope.....	38

Chapter 5. References

5.1 References (Books, journals and other online references)40
--	-----

**** Annexure 1 (if applicable):** Any paper presentation, research funding, sponsorship information/ certificate may be included.

Chapter 1

Introduction

1.1 Introduction

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 your 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.

E-voting has become increasingly popular in modern society, offering a convenient alternative to traditional methods of voting that require publishing of ballot papers and opening of polling stations. With the rise in popularity, it's crucial for electronic voting systems to be legitimate, accurate, safe, and accessible. However, these systems are susceptible to large-scale manipulations of votes, posing serious concerns for the security and transparency of the election process. Blockchain technology has emerged as a potential solution to address these concerns by providing decentralized platforms for electronic voting. The end-to-end verification advantages offered by blockchain technology make it an ideal choice for the production of electronic voting systems. The adoption of blockchain-based e-voting systems can enhance security, transparency, and efficiency in the voting process, ultimately strengthening democratic elections.

In modern democracies, the right to vote is considered a fundamental aspect of the legislative process that ensures equal participation of all citizens. With the advent of technology, web-based electronic voting systems have emerged as an effective and convenient means of casting votes. These systems offer several benefits, particularly for citizens living in remote or rural areas with limited access to physical polling stations. By exercising their right to vote, citizens can choose their leaders and contribute to the decision-making process in a democratic manner. Blockchain technology, on the other hand, has the potential to serve as a distributed ledger for storing data and information. The decentralized nature of blockchain ensures that no single entity has control over the voting process, making it more resistant to fraud and manipulation. The end-to-end verification provided by blockchain technology makes it an ideal choice for the production of secure and transparent electronic voting systems. Through the use of blockchain-based electronic voting systems, the security, accuracy, and accessibility of the voting process can be significantly enhanced.

In conclusion, the adoption of electronic voting systems based on blockchain technology can revolutionize the way we conduct democratic elections. These systems can offer a more efficient, accessible, and secure means of casting votes, while also ensuring transparency and trustworthiness in the voting process. By empowering citizens to participate in the decision-making process, these systems can contribute to the development of a more equitable and democratic society.

Blockchain technology has emerged as a potential solution for addressing the security and non-repudiation issues that plague traditional electronic voting systems. The study aims to examine the current state of blockchain-based voting research, explore online voting systems, and identify associated challenges to predict future developments. The abstract description of the intended blockchain-based electronic voting operation and the fundamental structure and characteristics of blockchain in relation to electronic voting are presented. The study finds that blockchain systems have the potential to address some of the challenges faced by election systems, but privacy protection and transaction speed remain major concerns in blockchain operations. For a sustainable blockchain-based electronic voting system, the security of remote participation must be feasible, and scalability and transaction speed must be addressed. To utilize blockchain technology in voting systems, the existing framework needs to be improved. The blockchain-based system, combined with the implemented cryptosystem, is designed to provide reliability, security, anonymity of votes and voters.

1.2 Motivation

Our project aims to establish a secure voting environment and demonstrate that a dependable e-voting scheme can be achieved through blockchain technology. By making e-voting accessible to individuals with a computer or mobile phone, we believe that it can facilitate administrative decision-making and promote direct democracy. This is particularly important as traditional elections are prone to corruption and manipulation, particularly in smaller towns or in countries with a corrupt political climate. Moreover, traditional elections can be prohibitively expensive, particularly if there are numerous geographically distributed vote centers and millions of voters. E-voting has the potential to address these challenges, provided that it is implemented with due care. While the idea of

e-voting is not new, our approach utilizes blockchain, which is a decentralized model of computation and storage that offers unique advantages.

Chapter 2

Literature Survey

2.1 Survey of Existing System

The Existing System of Election is running manually. The Voter has to Visit to Booths to Vote a Candidate so there is wastage of Time. Due to this many people don't go out to cast their vote which is one of the most important and Worrying factors. In democracy each and every vote is important. This Traditional system can be replaced by a new online system which will limit the voting frauds and make the voting as well as counting more efficient and transparent.

There are lot of practices are made to introduce the variations in electronic and online voting systems where different techniques and methodologies are used. Some of them guarantees the confidentiality and security to the system at some extent, still the voting information and process need to be control and manage with advanced systems that will ensures and guarantees the security and privacy of voter's and voter's information. The systems that are developed to caste the vote by means of digital approach using online portals and electronic devices use various encryption and decryption techniques to guarantee the secure data transaction. Homomorphic encryption is a well-known powerful technique with many useful applications. Recently, it has been applied to the design of online voting system. The voting system based on this encryption uses the exponential ElGamal cryptosystem. Before submission, the contents of each cast ballot are encrypted using the exponential ElGamal encryption. The additive homomorphism property of this crypto system makes it possible to tally encrypted ballots directly without decrypting them. However, numbers of techniques are present to convert the data in coded format to prevent from manipulation while transferring to the network. One drawback can be discussed here that after the correct data have been stored in the database trust and security is required at substantial level. Centralized storage is inconvenient if the data is esteemed because unauthorized access and attack by hackers will challenge the system in terms of reliability.

Previous models and architectures are used with help of centralized architecture approach. That may cause ethical and security problem. Collecting the data at a centralized location we take the data at the risk. It can be controlled unfairly. So, fair framework overcomes this problem of storing information to the distributed format with the help of blockchain. Blockchain is distributed ledger that stores all processed transaction in chronological order. Traditional databases are maintained by a

single organization, and that organization has complete control of the database, including the ability to manipulate with the stored data, to censor otherwise valid changes to the data, or to add data fraudulently. For most use cases, this is not a problem since the organization which maintains the database does so for its own benefit, and therefore has no motive to falsify the database's contents; however, there are other use cases, such as a financial network, where the data being stored is too sensitive and the motive to manipulate it is too enticing to allow any single organization to have total control over the database. Even if it could be guaranteed that the responsible organization would never enact a fraudulent change to the database (an assumption which, for many people, is already too much to ask), there is still the possibility that a hacker could break in and manipulate the database to their own ends.

In this section, we review past inventions, find loopholes, and later try to build more efficient and error-free models.

To study about past inventions in the field, we conducted an extensive research of the following papers and concluded the following:

[1] This thesis discusses the pros and cons of blockchain and bitcoin. Pros include the ability to conduct transactions without third-party intervention and avoid intermediary charges. Cons include high costs and potential price fluctuations, as well as the risk of illegal activities due to partial anonymity.

[2] The focus of this research paper is on smart contracts within blockchain. Smart contracts are essentially a personalized set of rules in code format, which automatically execute when certain transaction conditions are met. Once the smart contract initiates its execution, it is permanently linked to the blockchain and cannot be stopped or disconnected.

[3] This literature review examined Blockvote's implementation, which uses the bitcoin blockchain and ring signature scheme. The ring signature algorithm allows for group verification without revealing individual identities, ensuring voter secrecy. However, the protocol is limited to 3000 users due to constraints with the algorithm. Additionally, bitcoin cannot be used for decentralized application development (DApps), and there is a risk of fraud since critical tasks like tallying are managed by system servers.

[4] This paper examines the necessary requirements for improving the voting environment and suggests Votereum as a solution. Votereum is an e-voting system based on Ethereum blockchain

technology. The literature review highlights the use of technologies such as Truffle in developing this system. The proposed system is managed by two servers, with one handling the overall system and the other managing all blockchain-related requests.

[5] This paper addresses the issue of database manipulation in e-voting systems by proposing a blockchain-based approach. The author and project developers utilize the AES algorithm to encrypt data collected from fingerprint sensors. The research focuses on using blockchain algorithms to record voting results from all locations of the election.

[6] This literature survey concludes that EthVote is a decentralized application that utilizes Ethereum blockchain and smart contracts. It ensures verified voters can vote anonymously using Blind Signature. EthVote and BlockVote are both committed to creating a secure and transparent e-voting system while preserving voter privacy. Smart contracts are used to conduct the tallying phase, preventing fraud and ensuring fairness, and transparency allows the community to verify how ballots are counted.

[7] This paper compares relational databases to blockchain ledgers, highlighting how the former relies on firewalls, access controls, and encryption for security, while the latter relies solely on cryptology for integrity, encryption, confidentiality, transparency, and non-repudiation. An optimal voting system requires both confidentiality and transparency, which is difficult for a relational database to provide. In contrast, a blockchain system can easily provide both and eliminate the single point of failure inherent in a relational database.

2.2 Summarized Findings or Researchgaps

Ancient e-voting systems may encounter several issues that can impede their effectiveness. Some of the problems that such systems might face are as follows:

- 1. Anonymous vote-casting:** One of the primary challenges of an e-voting system is ensuring the privacy and anonymity of each vote. With traditional paper ballots, a person can cast their vote without revealing their identity to anyone else in the process. However, e-voting systems may be vulnerable to various forms of hacking or manipulation that can compromise voter anonymity.

For example, if the system's database is breached, the attacker may be able to associate a particular vote with a specific individual.

- 2. Individualized ballot processes:** In a traditional voting system, each voter receives a unique ballot paper that they use to cast their vote. However, with e-voting, the process is inherently digital and must be designed to account for the fact that each voter may require a slightly different interface or set of instructions to ensure that their vote is cast correctly. If this is not done carefully, it could result in errors or inconsistencies that may undermine the credibility of the entire voting process.
- 3. Ballot casting verifiability by (and only by) the organization:** In an e-voting system, it's important to ensure that voters can verify that their votes have been counted correctly. However, this must be done without revealing their identity or allowing them to change their vote after it has been cast. Ensuring this can be challenging, as the system must be designed to allow the voter to verify that their vote has been recorded correctly, but without allowing them to see the entire vote record.
- 4. High initial setup costs:** Implementing an e-voting system can be expensive, particularly if it requires significant infrastructure upgrades or the purchase of new hardware or software. In addition, the cost of maintaining and updating the system over time can also be significant, especially if security vulnerabilities are discovered or new features need to be added.
- 5. Increasing security problems:** With the increasing use of e-voting systems, there has been a corresponding increase in the number of security problems and vulnerabilities that have been discovered. Hackers and other malicious actors may seek to disrupt or manipulate the voting process in various ways, from altering the vote counts to stealing voter data or disrupting the system's infrastructure. To address these challenges, e-voting systems must be designed to be highly secure and resilient, and must be able to adapt to changing security threats over time.

Apart from these, traditional voting systems, such as ballot box voting or electronic voting, are vulnerable to various security threats that can undermine the accuracy and fairness of election results.

Our e-voting system offers an enhanced level of security compared to other systems in the market. While many e-voting systems rely solely on the RSA encryption algorithm for protection, our system incorporates an additional layer of security through a manual verification process of a voter's citizenship by the system administrator. Before a voter can cast their vote, the system administrator manually verifies their citizenship, ensuring that only eligible voters can participate in the election. This process helps prevent voter fraud, as it prevents hackers from registering multiple users and casting multiple votes, which can skew the election results. Additionally, it provides greater confidence in the accuracy and legitimacy of the election results.

By including a manual verification process, our e-voting system is better equipped to prevent denial of service attacks, which can overwhelm a system with bogus voter registrations or requests, rendering it unusable. The manual verification process ensures that only legitimate users can access the system, thereby reducing the risk of these types of attacks. Furthermore, our system provides the system administrator with the ability to reject voters who do not meet the necessary criteria for participation in the election, adding an extra layer of security to the system. This feature enables the administrator to quickly identify and eliminate suspicious or fraudulent voter registrations, improving the overall integrity of the voting process.

In summary, our e-voting system offers a secure and reliable voting solution that incorporates advanced security measures to protect against various threats and ensure the integrity of the election results.

Chapter 3

Proposed System

3.1 Problem Statement and Objectives

3.1.1 Problem Statement

The current voting system requires some improvement in it because of the issues mentioned above. This can be achieved by replacing the existing system with a new system which will limit the voting frauds and make the voting as well as counting more efficient. Our objective of this E-voting system using blockchain is to solve the issues of digital voting by using blockchain technology. Blockchain enabled e-voting could reduce voter fraud and increase voter access.

In comparison to conventional voting methods, e-voting has enhanced the efficiency and the integrity of the process. Because of its flexibility, simplicity of use, and affordable costs compared to general elections, electronic voting is widely utilized in various ways during elections. Despite this, existing electronic voting methods face the risks of running into over-authority and manipulated details, limiting fundamental fairness, privacy, secrecy, anonymity, and transparency in the voting process. Since e-voting procedures are centralized and licensed by the critical authority that controls, measures, and monitors the process in an electronic voting system, this is a problem that might hamper a transparent voting process. Recent controversies in modern democracies such as USA and India amplify this argument and prove it true. It is essential to ensure that assurance in voting does not diminish. Because of the distributed structure of the blockchain, a smart contract-based electronic voting system reduces the risks involved with electronic voting and allows for the process to be tamper-proof.

3.1.2 Objectives

Our proposed solution to address the security concerns and inefficiencies of traditional voting systems is to develop a blockchain-based e-voting system. The objectives of our system are as follows:

1. Ensure open verifiability and transparency of the election process through a publicly accessible and auditable blockchain ledger:

Our e-voting system will use a blockchain ledger to provide a publicly accessible and auditable record of all votes cast. By using a distributed ledger, every transaction will be recorded and validated by multiple nodes, ensuring that the voting process is transparent and open to scrutiny. This transparency helps to build trust in the system and reduces the risk of manipulation or fraud.

2. Guarantee the integrity of the vote using cryptographic techniques to verify accurate recording and counting:

To ensure the integrity of the vote, our e-voting system will use cryptographic techniques to verify accurate recording and counting of votes. This will provide a secure and tamper-proof method for recording and counting votes. By using cryptographic techniques, the system will provide end-to-end verifiability, allowing voters to confirm that their votes were recorded correctly.

3. Enforce voter eligibility through identity verification mechanisms:

To ensure that only eligible voters can participate in the election, our e-voting system will enforce voter eligibility through identity verification mechanisms. This will include verifying the identity of voters using government-issued identification documents, such as a passport or driver's license. The system will also verify that the voter is registered to vote and is eligible to participate in the election.

4. Ensure tamper-proofing of the system to prevent unauthorized modifications or alterations to the voting data:

To prevent unauthorized modifications or alterations to the voting data, our e-voting system will use cryptographic techniques to ensure that the data is tamper-proof. The system will use a combination of digital signatures and hash functions to create a unique digital fingerprint of each vote, which will be recorded on the blockchain. Any attempts to modify the data will be immediately detected, and the system will reject any tampered data.

5. Provide decentralization and security against manipulation or rigging by central authorities or malicious organizations:

To provide decentralization and security against manipulation or rigging by central authorities or malicious organizations, our e-voting system will use a decentralized blockchain network. This will

ensure that the voting process is not controlled by a single central authority or organization, reducing the risk of manipulation or rigging. Additionally, the use of a blockchain ledger will provide security against attacks on the system, as the distributed nature of the network will make it difficult for attackers to manipulate the data.

3.2 Scope of the Work

THE SCOPE OF OUR PROJECT IS TO DEVELOP A PROJECT THAT:

1. Integrates the blockchain paradigm into the e-voting procedure to create a secure and flexible voting mechanism without relying on a Trusted Third Party (TTP): Our project aims to develop an e-voting system that leverages the blockchain technology, which enables a decentralized and transparent database. By integrating blockchain, we can eliminate the need for a central authority or TTP, making the voting process more secure and flexible.
2. Utilizes cryptography and smart contracts to ensure secure and cost-efficient elections while guaranteeing voter privacy: Our e-voting system will use cryptography techniques, such as digital signatures and hash functions, to ensure secure and private voting. Additionally, smart contracts will be utilized to enforce rules and regulations of the voting process, including voter eligibility and vote counting. This approach helps ensure that the election is cost-efficient and transparent, while also protecting the privacy of the voters.
3. Addresses security concerns, including voter privacy, integrity, verification, non-repudiation of votes, and transparency of counting, through a decentralized application: Our e-voting system will provide solutions to various security concerns, including ensuring voter privacy, vote integrity, verification, non-repudiation of votes, and transparency of the vote counting process. Our system will be decentralized, meaning that no single entity will have control over the election process, thus preventing unauthorized modifications.

4. Provides a feasible and general e-voting protocol that satisfies most of the main requirements for an e-voting system, such as open verifiability, transparency, and tamper-proofing: Our project aims to create a feasible and general e-voting protocol that satisfies most of the main requirements for an e-voting system, such as open verifiability, transparency, and tamper-proofing. This approach will help to improve the credibility of the election process.
5. Utilizes a consensus mechanism and distributed ledger technology to ensure valid transactions and secure, transparent storage of voting data: We plan to utilize a consensus mechanism and distributed ledger technology to ensure that all transactions are valid, and the voting data is securely and transparently stored. This approach will prevent any fraudulent activities, ensuring the accuracy and fairness of the election results.
6. Implements a secure digital identity system to authenticate voters, prevent fraud, and enable remote voting while maintaining the privacy and security of the vote: Our e-voting system will implement a secure digital identity system to authenticate voters, prevent fraud, and enable remote voting while maintaining the privacy and security of the vote. This approach will enable voters to cast their votes remotely while ensuring the security and privacy of their votes.
7. Provides a user-friendly interface for efficient and easy voting, while enabling auditing and monitoring to ensure a fair, transparent, and tamper-proof election process: Our e-voting system will provide a user-friendly interface for efficient and easy voting. We will also enable auditing and monitoring to ensure a fair, transparent, and tamper-proof election process. This approach will help ensure the credibility of the election results and promote public trust in the election process.

3.3 Analysis/Framework/ Algorithm

A blockchain is defined as a network of blocks that are interconnected together and carry specific information(database) in a secure, transparent and real manner (peer- to- peer). In other words, blockchain is a collection of connected computers rather than a single, centralized servers, making the entire network decentralized. Decentralization, responsibility (accountability), and

security are the three main tenets of blockchain technology. This system can cut costs dramatically while adding functional effectiveness. Operations created on blockchain armature will continue to be in demand and used. Blockchain Architecture essentially includes the following:

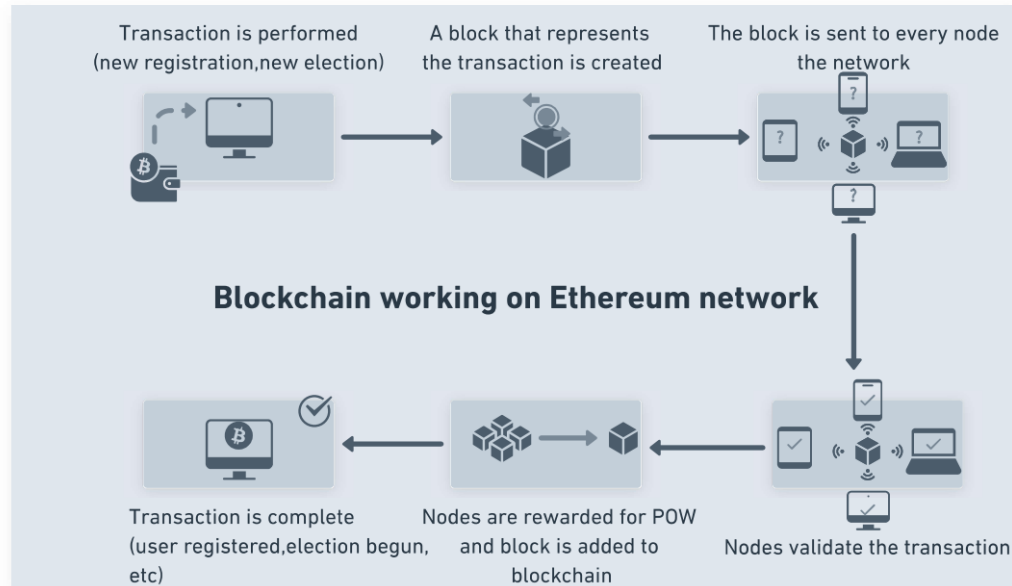


Fig. 1 Working of Blockchain

1) Transactions: A transaction in blockchain refers to the transfer of digital assets or information from one participant to another. These transactions are recorded on a decentralized and distributed ledger, such as a blockchain, which ensures their integrity and makes them immutable. The information typically included in a transaction varies depending on the type of blockchain and the assets being transferred. The proposed system can support the following transactions:

- **Votecasting:** Voters can cast their votes using the e-voting system. The vote will be recorded on the blockchain, which provides transparency and immutability to the voting process.
- **Votecounting:** The votes casted can be counted automatically by the blockchain network, ensuring accuracy and transparency.
- **Verification of votes:** The votes can be verified using the blockchain network, ensuring that no fraudulent activity has taken place.
- **Auditing:** Auditors can access the blockchain network to audit the voting process and verify its integrity.

- **Secure data storage:** The blockchain network can securely store all the data related to the voting process, including voter registration details, voting data, and results.
- **Time-stamping:** The blockchain network can provide a time-stamp for each transaction, providing a clear record of when each vote was casted and counted.

2) Block: Blocks are made up of transactions and block header information. Blocks are data structures that serve as containers for groups of transactions and are replicated across all network nodes. Miners build blocks in the blockchain. The Block header is the metadata that used in confirming a block's validity. Transactions are stored in a block's remaining available space. Depends on the miner's preference, a block may include many numbers of transactions. For Ethereum specifically, 70 transactions can be fit. However, since is our system is built on test network, each transaction is given a specific block.

3) Broadcasting of Block: A Peer-to-peer (P2P) network are based on the IP protocol. There is no centralized nodes. While working together using a consensus mechanism, all nodes equally provide and can consume all services. Similarly, when a block is generated in a blockchain-based system, it needs to be broadcasted to all the nodes in the network for verification. This process is important to ensure that the block is valid, and all transactions included in it are legitimate.

4) Validation of transaction: The validation of a block in a blockchain-based system is an essential process that ensures the accuracy and integrity of the network. Here's a plagiarism-free explanation of how the validation of a block is performed:

- **Hashing:** The first step in the validation process is to hash the block. This involves creating a unique digital fingerprint of the block using a cryptographic hashing algorithm.
- **Verification of Previous Block:** The hash of the previous block is then checked to ensure that it matches the reference included in the current block. This process ensures that the blocks are linked in a secure and tamper-proof way.
- **Validating Transactions:** The nodes then validate each transaction included in the block to ensure that they meet the required criteria. This process typically involves checking that the transactions are legitimate, have not been previously spent, and that the sender has sufficient funds to complete the transaction.

- **Consensus:** Once the transactions have been validated, the nodes on the network reach consensus on whether the block is valid. This process involves comparing the results of the validation process and agreeing on whether the block should be added to the blockchain or rejected.

5) Block is added to the blockchain: Blockchain technology has prioritized fraud prevention. Any transaction will be added to the Blockchain network only after it's validated. This is to help fake/fraud deals. So, once a block is created, verified, and validated, it is ready to be added to the blockchain. This process involves appending the block to the existing chain of blocks, creating a permanent and tamper-proof record of all transactions that have taken place on the network.

3.4 Details of Hardware & Software

- 1. Truffle v5.0.9:** Truffle is a popular development framework used for building decentralized applications (dapps) on the Ethereum blockchain. It provides a suite of tools and utilities for smart contract development, testing, deployment, and management, making it easier for developers to create and deploy blockchain-based applications. Key features include automated contract testing, in-built support for smart contract compilation and deployment, and a development console for interacting with smart contracts.
- 2. Ganache v7.4.0:** Ganache is a personal blockchain that allows developers to simulate an Ethereum blockchain environment for testing and debugging smart contracts. It is part of the Truffle Suite, which is a collection of tools and utilities for Ethereum development. With Ganache, developers can create a local blockchain environment with a set of accounts, each with a balance of ether that can be used for testing transactions and interactions between smart contracts. It also provides a user-friendly interface for managing the blockchain and inspecting transactions, blocks, and other blockchain events in real-time.
- 3. NodeJs- v11.12.0:** Node.js is an open source, cross-platform runtime environment and library that is used for running web applications outside the client's browser. It is used for server-side programming, and primarily deployed for non-blocking, event-driven servers, such as traditional

web sites and back-end API services, but was originally designed with real-time, push-based architectures in mind.

3.5 Design details

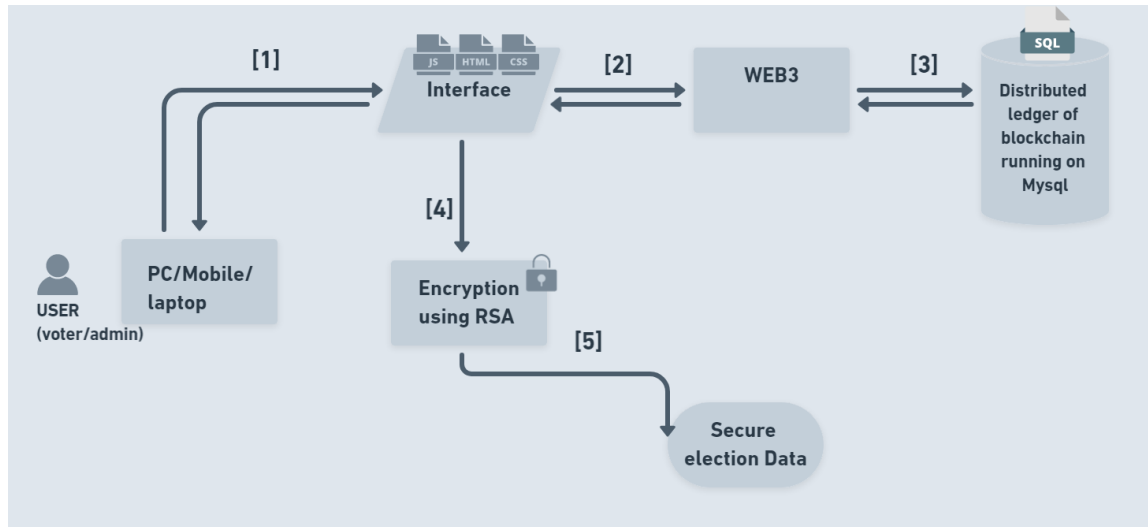


Fig 2. Design of Proposed System

1. Interaction with the Frontend: The user (voter/admin) will access the web application built with React, Node.js, HTML, CSS, and JavaScript by entering the URL in their browser. The web application will load in the user's browser, and they will be able to interact with the application through the UI. Once the user interacts with the application, the frontend code written in React will send requests to the backend code written in Node.js, which will handle the requests and respond with the appropriate data.

2. Encrypting the transaction request: Encrypting the transaction request is done to ensure the confidentiality and security of the transaction. In a blockchain network, transactions are broadcast to all nodes in the network and stored on a public ledger. If a transaction is sent in plain text, anyone can intercept and read the content of the transaction, including sensitive information such as user details, voting preferences, or any other data that should be kept private. Additionally, encryption can

also help prevent unauthorized access to the blockchain network and protect against malicious attacks.

3. Sending the encrypted content back to the frontend interface: When the transaction request is encrypted in the frontend, it is then sent as a parameter to the web3 function call that interacts with the blockchain network. The encrypted data can be sent through a method call such as `web3.eth.sendTransaction` or `web3.eth.call`, with the encrypted data passed as a parameter.

4. Sending encrypted transaction request web3 to interact with the blockchain network: On receiving the encrypted data web3 then interacts with the blockchain network (in our case Ganache). To establish a connection with any blockchain network it requires 'Provider' which is a gateway between the client and the network. Ganache being a local network provides the provider for users to test their applications locally without the need of a real network. Web3 is a library that provides a way for frontend applications to interact with the blockchain network. It allows developers to send transactions, read data from smart contracts, and interact with the blockchain in a secure and reliable manner.

5. Connecting with the distributed ledger: Once web3 interacts with the Ganache blockchain network, it establishes a connection with the distributed ledger, which is essentially a decentralized database that records all transactions made on the blockchain. This connection allows web3 to communicate with the ledger and execute transactions. When a transaction request is received by the network, the distributed ledger validates it and creates a new block on the blockchain, which includes the details of the transaction. This block is then added to the existing chain of blocks, creating an immutable record of the transaction.

3.4 Methodology

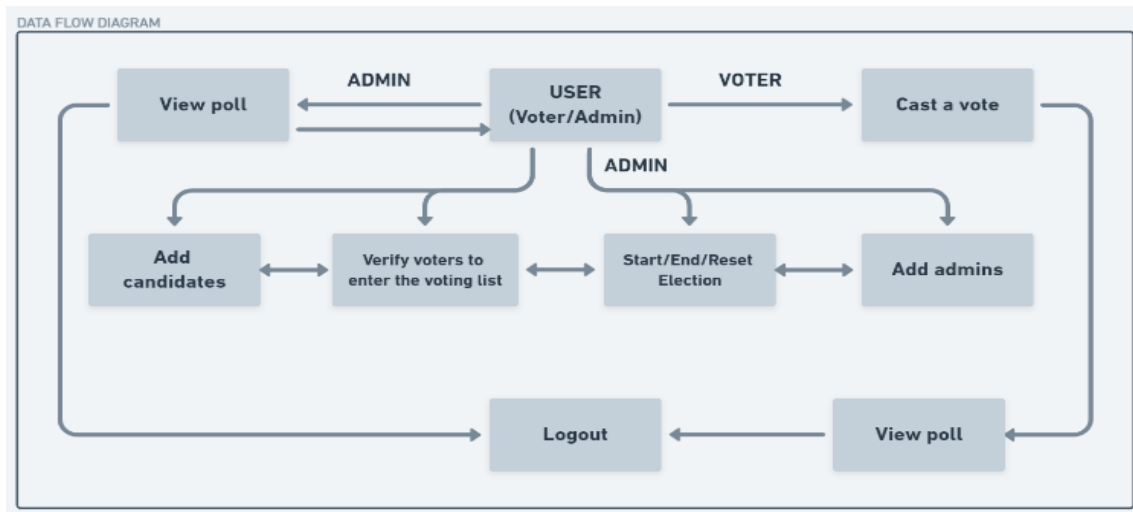


Fig 3. Methodology

The proposed E-voting system using Blockchain is designed in such a way that there are two main users who are responsible for the functioning of the entire system. A user can be either a voter or an Admin.

- A. Administrator:** To manage the lifecycle of an election. Multiple trusted institutions and companies may be enrolled in this role. The admin of the system has various privileges like adding candidates that wish to stand in the election. The admin can start and end the election and also reset the election for a new election to begin. Additionally, the admin can verify the voters based on their credentials and either accept them into the voting list or reject them. The admin will not be able to view a voter vote choice and also his credentials as they will be stored in an encrypted format in the ledger. Likewise, the admin can also add other admin in the system.
- B. Voter:** An individual who is eligible to vote. The voter can login/register into the system and cast a vote. Apart from that the voter can also view the poll and election result while the election is going and in the end of the election as well. All of these tasks can be performed by the voter if and only if he/she is verified to do so by the system Admin.

Following process will explained the entire process performed to complete the election using blockchain technology:-

1. **VOTER REGISTRATION:** The proposed system used registration website as for the users to register themselves. The website collects details like name, e-mail, citizenship number at the time of registration. These details are collected from the user and stored in the database.
2. **AUTHENTICATION:** Once the details are collected from the user the next step is the authentication process. Aadhar verification is done.. After successful verification the user is provided with the blockchain account address and private key. User has to keep these details intact and protected at all costs. Proof Verification of the voter is shown in fig
3. **VOTING PORTAL:** The voter logs into the website using their unique Blockchain account. This page will be redirected to the vote casting page where parties are displayed, the voter will cast their vote to a specific party.
4. **VOTE TRANSACTION:** The next step is the casting of vote by the user. The voter logs into the e-voting website by entering the login credentials, the blockchain account which was given to the user after the authentication process was complete. The home page displays the candidates and parties of the election. The voters select the party they want to cast their vote for. Once the user casts his vote the page connects to the Metamask, a local blockchain environment where the voter can confirm their vote transaction.
5. **RESULT VALIDATION** The next step is the result validation process. The vote is incrementally added to the data as it is cast. Once all the voters complete casting their vote the system computes the overall result and the result is displayed in the page.
6. **VOTE VERIFICATION:** Once the voter is voted, he cannot cast his vote next time because of the blockchain transaction, there is a gas limit set for each transaction once the user finishes the gas limit he can't cast another vote. The gas limit is set as such only one vote can be casted for one cast.

3.5 Implementation

The Ethereum blockchain enables us to write code for our application by utilizing the Ethereum Virtual Machine (EVM), which runs smart contracts on the blockchain. Our application utilizes smart contracts to handle data input/output and execute logic, which are coded using the Solidity programming language. To run our smart contract along with the entire web application we have made use of the following simulation tools and frameworks:

1. Truffle
2. Ganache

In addition to the aforementioned tools used in blockchain development, we utilized HTML, CSS, and ReactJS to enhance the user interface and interactivity of our web application's frontend. For storing the voter's details, we employed a MySQL database.

The initial step in building our application involves the installation of all the necessary dependencies. Once this is completed, we proceed to write the contract and successfully deploy it to the blockchain. The contract can be created by declaring a smart contract with the "contract" keyword, followed by the name of the contract. Next, we declare a mapping variable using 'mapping(address => bool) admins;' which create a mapping variable called admin. Mapping is a key-value data structure which allows us to store key-value pairs, helping us associate a key of one data type to a value of another data type.

3.5.1 Smart Contract:

The project consists of mainly two smart contracts written in solidity language and a descriptive analysis of the two is given below:

```
//SPDX-License-Identifier: UNLICENSED
pragma solidity >=0.4.22 <0.9.0;

contract Election {
    mapping(address => bool) admins;
    string name; // name of the election. example: for president
    string description; // description of the election
    bool started;
    bool ended;

    constructor() {
        admins[msg.sender] = true;
        started = false;
        ended = false;
    }

    modifier onlyAdmin() {
        //require(admins[msg.sender] == true, "Only Admin");
        _;
    }

    function addAdmin(address _address) public onlyAdmin {
        admins[_address] = true;
    }
}
```

Fig. 4Code block to define a smart contract

In Fig. 4 the "admins" variable is a mapping variable that stores the addresses of the election admins as keys, and their access privileges as boolean values. This allows the contract to easily check whether a given address is an administrator or not, by looking up its value in the "admins" mapping.

Subsequently, the variables name, description are of string type and used to define the election name and its brief description. The variables started and ended are of bool type that help to indicate whether the election has started or ended. Further, the constructor() is a constructor function that is executed when the contract is deployed to the blockchain. In this case, the constructor sets the address of the contract creator as an administrator, and initializes the "started" and "ended" boolean variables to "false". In addition to that, as shown in Fig. 4.

Next, to define the candidate and its role in the entire election process the smart contract further includes a struct of the name Candidate which has the three fields as shown in Fig. 5 viz. name, info, exists which defines the name, information and existence status of the candidate respectively. Further, the contract also defines a mapping variable 'Candidates' which maps the string candidate name to the Candidate object. The candidateNames variable is an array of strings that holds the names of all the candidates.

```

struct Candidate {
    string name;
    string info;
    bool exists;
}
mapping(string => Candidate) public candidates;
string[] candidateNames;

function addCandidate(string memory _candidateName, string memory _info)
    public
    onlyAdmin
{
    Candidate memory newCandidate = Candidate({
        name: _candidateName,
        info: _info,
        exists: true
    });

    candidates[_candidateName] = newCandidate;
    candidateNames.push(_candidateName);
}

function getCandidates() public view returns (string[] memory) {
    return candidateNames;
}

```

Fig. 5 Code block of candidate definition in smart contract

The addCandidate function takes two string parameters: _candidateName and _info. It creates a new Candidate object with the given name and information, and sets the exists flag to true. Then it adds the candidate to the candidates mapping by mapping the candidate's name to the new Candidate object. Finally, it adds the candidate's name to the candidateNames array.

Moreover, Fig. 6 displays a code block of the voter section wherein it defines a struct called 'Vote' that stores the details of a vote, including the voter's address, ID, name, and the candidate they voted for. The votes are stored in an array of Vote structs called 'votes'. The contract also includes a mapping called 'voterIds' that maps the voter's ID to a boolean value indicating whether the voter has already voted or not. A string array called 'votersArray' is used to keep track of all the voters who have cast their votes.

The 'vote' function is used to cast a vote. It takes in the voter's ID, name, and the candidate they want to vote for. Before accepting the vote, the function performs three checks:

1. The election must have started but not ended.
2. The candidate the voter wants to vote for must exist.
3. The voter must not have already voted.

If all the checks pass, the function creates a new Vote struct with the details of the vote and adds it to the 'votes' array. It also sets the 'voterIds' mapping to true for the voter's ID and adds the ID to the 'votersArray'. The 'getVoters' function is used to retrieve the array of voter IDs that have cast their votes. It is a public view function, meaning it can be called by anyone on the blockchain network. It returns the 'votersArray' containing the IDs of all voters who have cast their votes in the election.

```
struct Vote {
    address voterAddress;
    string voterId;
    string voterName;
    string candidate;
}
Vote[] votes;
mapping(string => bool) public voterIds;
string[] votersArray;

function vote(
    string memory _voterId,
    string memory _voterName,
    string memory _candidateName
) public {
    require(started == true && ended == false);
    require(candidates[_candidateName].exists, "No such candidate");
    require(!voterIds[_voterId], "Already Voted");

    Vote memory newVote = Vote({
        voterAddress: msg.sender,
        voterId: _voterId,
        voterName: _voterName,
        candidate: _candidateName
    });
```

Fig. 6 Code block to define the voters in the smart contract

As shown in the Fig. 7 the code block is for the contract called "Migrations". It has two state variables: owner and last_completed_migration. owner is an address variable that is initialized to the address of the contract deployer. last_completed_migration is a uint variable that will store the last completed migration number.

The restricted modifier is defined to restrict access to certain functions in the contract. It uses the require statement to check if the caller of the function is the owner of the contract. If the caller is not the owner, the function will not execute and an error message will be returned. The setCompleted function is a public function that sets the last_completed_migration variable. It takes in a uint parameter called completed. This function is restricted and can only be called by the owner of the contract. When called, it will update the value of last_completed_migration to the value of the completed parameter passed in.

When the Election contract is ready for deployment to the blockchain network, it is compiled and migrated using a tool such as Truffle. The Migrations contract is deployed first, and then the Election contract is deployed after it. The last_completed_migration variable in the Migrations contract is updated to indicate that the deployment of the Election contract has been completed.

In a nutshell, The Migrations contract is used in the Truffle framework for managing the deployment of contracts to a blockchain network. It provides a simple mechanism for keeping track of which contracts have been deployed and their respective versions.

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Migrations {
    address public owner = msg.sender;
    uint public last_completed_migration;

    modifier restricted() {
        require(
            msg.sender == owner,
            "This function is restricted to the contract's owner"
        );
        _;
    }

    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }
}
```

Fig. 7 Code block to define the migration of the election smart contract

3.5.2 ENCRYPTION:

The data of the voters, especially the votes they have casted, is stored in an encrypted format to guarantee the authenticity and integrity of the information. The encryption algorithm used for the same is the RSA algorithm, fig. 2 explain the general working of the RSA algorithm. The RSA algorithm is a public-key cryptographic algorithm that is widely used for secure data transmission. It involves the use of two keys, a public key for encryption and a private key for decryption. The algorithm is based on the mathematical properties of large prime numbers and is secure because of the difficulty of factoring large composite numbers.

```
1 Select two large prime numbers, p and q.
2 Calculate  $n = p * q$ .
3 Calculate the totient function of  $n$ ,  $\phi(n) = (p-1) * (q-1)$ .
4 Select an integer e such that  $1 < e < \phi(n)$  and  $\gcd(e, \phi(n)) = 1$ .
5 Calculate d such that  $d * e \equiv 1 \pmod{\phi(n)}$ .
6 The public key is  $(n, e)$  and the private key is  $(n, d)$ .
7 To encrypt a message m, compute  $c = m^e \pmod{n}$ .
8 To decrypt the encrypted message c, compute  $m = c^d \pmod{n}$ .
```

Fig 8. RSA algorithm

The RSA encryption algorithm was used to encrypt the data in the e-voting system to ensure the privacy and security of the voting process in the following ways:

1. **Generate Keys:** First, the e-voting system generates a public-private key pair using the RSA algorithm. The public key is used for encrypting the data, while the private key is used for decrypting the data.
2. **Voter Authentication:** Before the voting process starts, each voter is authenticated by the e-voting system. The voter is issued a unique ID, which is used to encrypt their vote using the RSA algorithm. The voter's public key is stored on the blockchain, and the voter's private key is kept secret.
3. **Vote Encryption:** When a voter casts their vote, the e-voting system encrypts the vote using the voter's public key. This ensures that only the voter's private key can be used to decrypt the vote.
4. **Vote Verification:** The encrypted vote is then added to the blockchain, where it can be verified by the network. The vote remains encrypted until the end of the voting process.
5. **Vote Counting:** Once the voting process is complete, the encrypted votes are decrypted using the private keys of the voters. The decrypted votes are then counted, and the results are announced.

Using the RSA encryption algorithm in the e-voting system provides a secure and tamper-proof way to protect the privacy of the voters and ensure the integrity of the voting process. By encrypting the votes using the public keys of the voters, only the authorized individuals with the corresponding private keys can access the votes given out. This prevents other voters or the admin to view which voter has issued which vote. This provides a secure and transparent voting process that ensures the trustworthiness of the election results.

Chapter 4

Results

4.1 Results

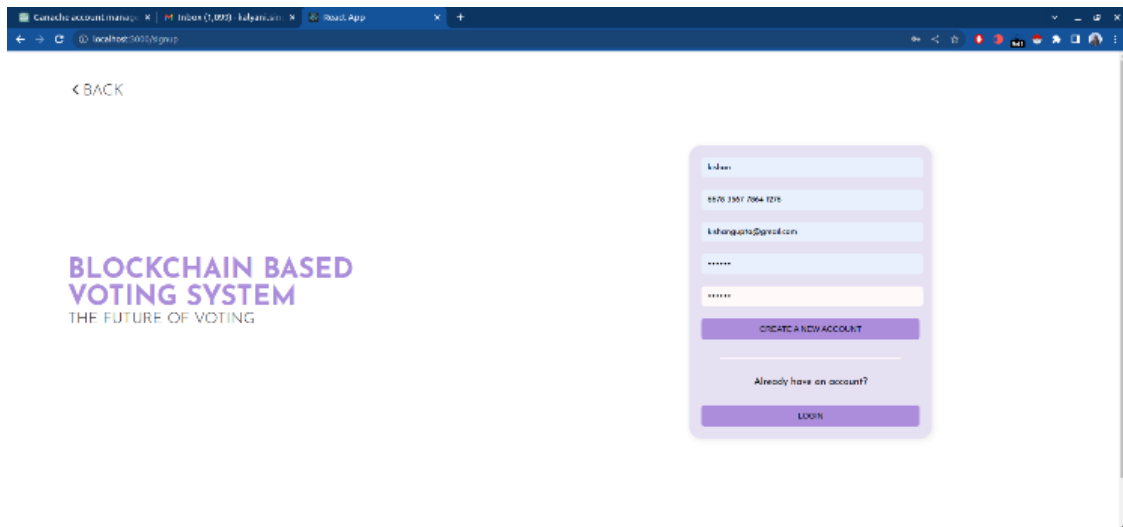


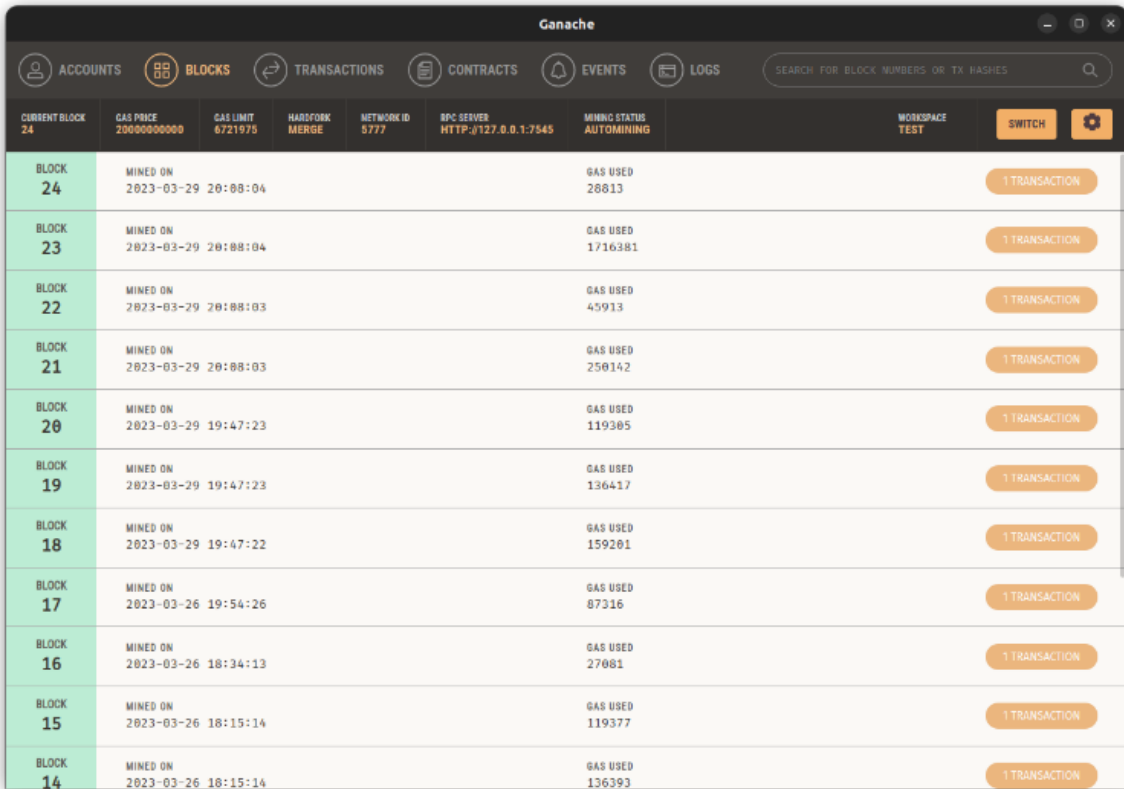
Fig. 9 transaction initiation by registering a new user

As shown in the fig. 9 a new user is being registered into the system. When a new user registers in a blockchain-based system, their information can be recorded on the blockchain by initiating a transaction. This transaction is created by the smart contract and includes the user's information. The transaction is then signed with the private key of the user who initiated the transaction, ensuring its authenticity.

Ganache				
ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS
LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES			
CURRENT BLOCK 24	GAS PRICE 20000000000	GAS LIMIT 6721976	HARDWARE MERGE	NETWORK ID 5777
			RPC URL HTTP://127.0.0.1:7545	MINI STATE AUTOMINING
MNEMONIC		HD PATH		
kid census approve three exit you idle wall coach clock pact journey		m/44'/60'/0'/0'/account_index		
ADDRESS	BALANCE	TX COUNT	INDEX	
0x8CF1211225E6Fbd8E7A32DD6cb47BEae79c07EF0	99.98 ETH	24	0	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xAd7beBAE9713735130922E875604ce8AAba55682	100.00 ETH	0	1	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xA7d06d09c8EC7aB3538d346bCe85CF2639D33b89	100.00 ETH	0	2	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x6e74871E875265D167C86272D2Ef1b9d90F0a394	100.00 ETH	0	3	
ADDRESS	BALANCE	TX COUNT	INDEX	
0xB5C82D203E5c8e64fD2ea804bADCE53b14732D6C	100.00 ETH	0	4	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x46565F6a4F5b77CC6a4e41B8793e32106aD3e2e8	100.00 ETH	0	5	
ADDRESS	BALANCE	TX COUNT	INDEX	
0x0482BFFe51D151a23744D0a6b5579dfbD115eb3d	100.00 ETH	0	6	
ADDRESS	BALANCE	TX COUNT	INDEX	

Fig. 10 Ganache accounts tab

As the transaction is added to the blockchain, the balance and transaction count of the affected accounts will be updated accordingly in the Ganache interface, as shown in the Fig.10. The account associated with the new user's registration information will have a transaction count of one, reflecting the fact that a new transaction has been sent from that account. This helps create a permanent, tamper-proof record of the user's registration on the blockchain, which can be used for various purposes, such as authentication and verification during the voting process. The Accounts tab in the Ganache interface provides a real-time view of the status of each account on the network, including their current balance, transaction count, and other details related to their activity on the network. Since we are using a simulated version of the blockchain network on ganache, all of the transaction take place from the same account, here with the first account having a txcount = 24.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
24	20000000000	8721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	TEST

BLOCK	MINED ON	GAS USED	TRANSACTION
24	2023-03-29 20:08:04	28813	1 TRANSACTION
23	2023-03-29 20:08:04	1716381	1 TRANSACTION
22	2023-03-29 20:08:03	45913	1 TRANSACTION
21	2023-03-29 20:08:03	258142	1 TRANSACTION
20	2023-03-29 19:47:23	119305	1 TRANSACTION
19	2023-03-29 19:47:23	136417	1 TRANSACTION
18	2023-03-29 19:47:22	159201	1 TRANSACTION
17	2023-03-26 19:54:26	87316	1 TRANSACTION
16	2023-03-26 18:34:13	27081	1 TRANSACTION
15	2023-03-26 18:15:14	119377	1 TRANSACTION
14	2023-03-26 18:15:14	136393	1 TRANSACTION

Fig. 11 Ganache block tab

In ganache, every transaction is given its own separate block as seen in the block tab in Fig. 11. In the Block tab of the Ganache interface, a new block will appear when a new transaction is generated and added to the blockchain. Each block contains a set of transactions and a reference to the previous block, forming a chain of blocks (i.e., the blockchain). When a new transaction is validated by nodes on the network and added to a block, the block's details, such as its block number, hash value, timestamp, and transaction count, will be updated in real-time in the Ganache interface. The block

number and transaction count will increase by one, reflecting the addition of the new transaction to the block.

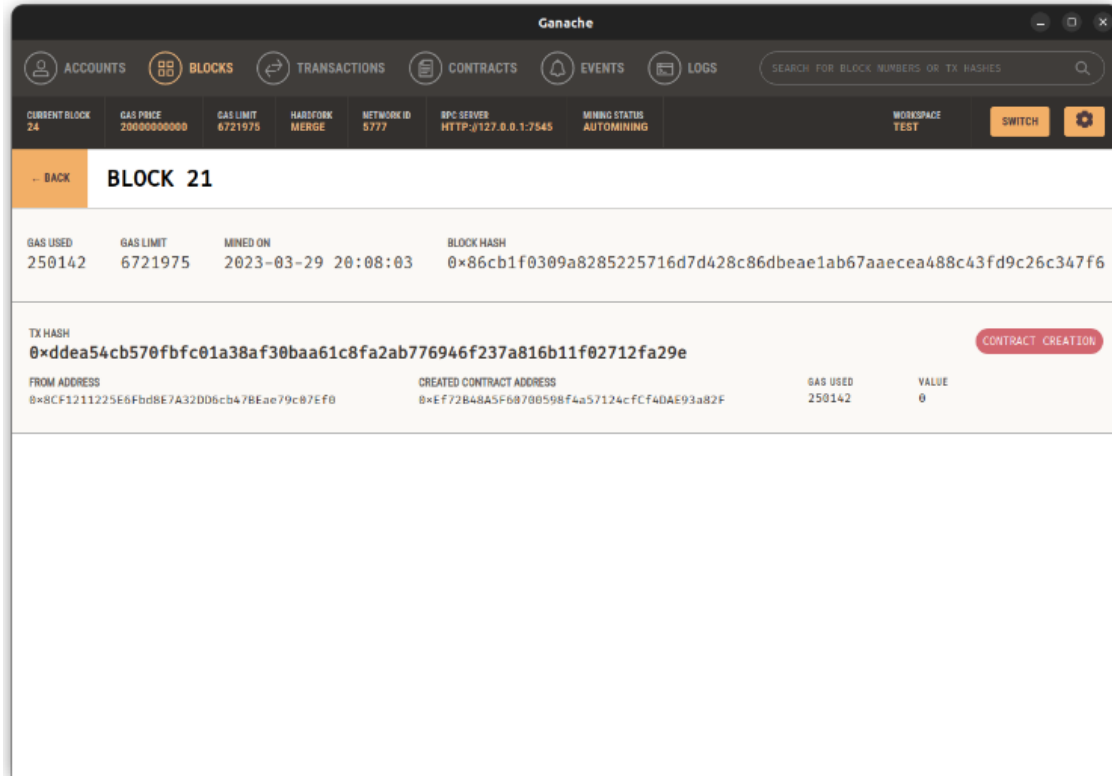


Fig. 12 Block details of contract creation

When clicked on a block in the Ganache interface, we can view detailed information about that particular block, including the amount of gas used in the block and the maximum amount of gas that could have been used in the block. The gas used and gas limit are important metrics for evaluating the efficiency and performance of the blockchain network. A lower gas usage indicates that the transactions included in the block are more efficient and less expensive to process, while a higher gas usage indicates the opposite. In addition to the gas usage and gas limit, the block details also provide information about the time at which the block was mined, represented by the MINED-ON timestamp. The block hash is a unique identifier for the block and can be used to trace the block's history and transaction flow. The transaction hash value represents the unique identifier for the transaction that created the block. If the block was created due to a smart contract creation transaction as shown in Fig. 12, the Ganache interface will display additional details, such as the from address, the contract creation address, the amount of gas used, and the value of the transaction (if any). These details are useful for debugging and troubleshooting smart contracts and for monitoring the activity of the blockchain network. Overall, the Ganache interface provides a comprehensive view of the

blockchain network and allows developers and users to explore and analyze the various components of the blockchain, including blocks, transactions, and smart contracts.

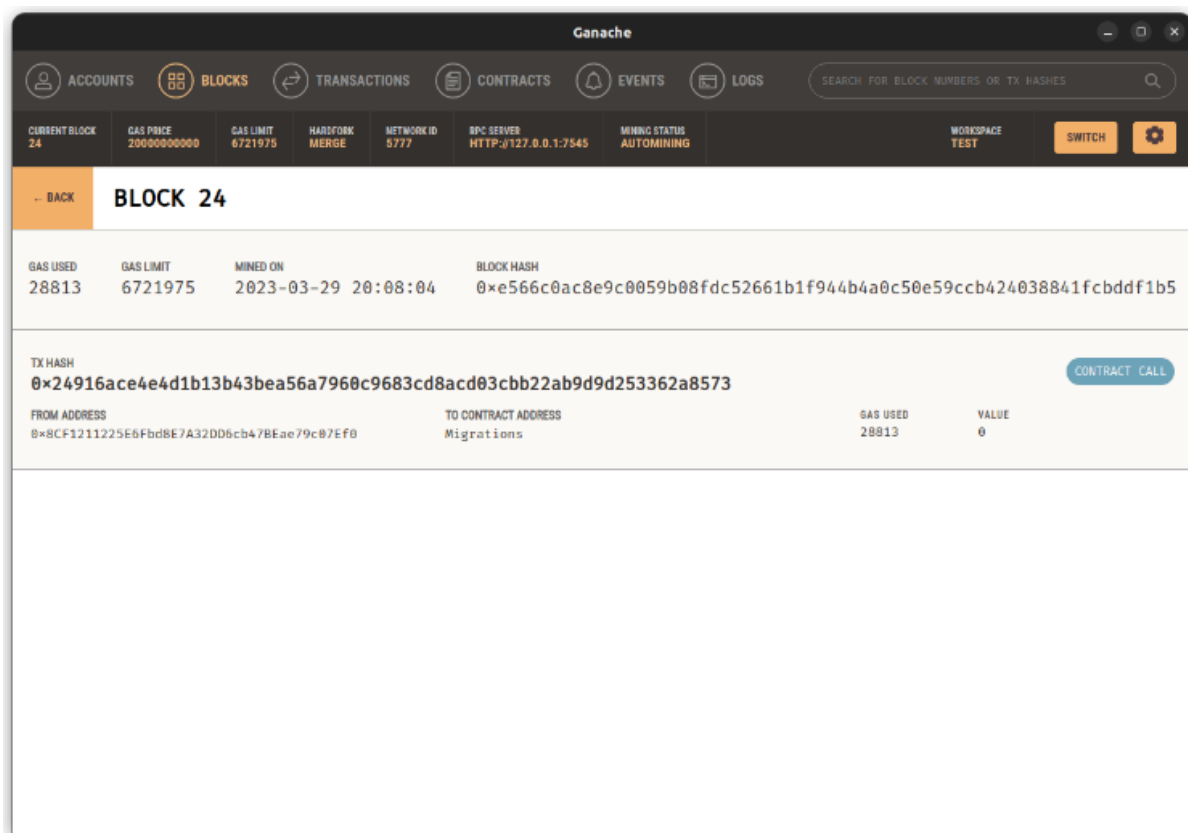


Fig. 13 Block details of contract call

On the other hand, for smart contract calls like a voter casts their vote by sending a transaction to the smart contract with their vote preference. The smart contract then checks the voter's identity and verifies that they are eligible to vote before recording their vote on the blockchain is a smart contract call as well. So, when a block that was created due to a smart contract call as shown in Fig. 13 is viewed it shows information about the gas used, gas limit, block timestamp, block hash, and transaction hash. We can also see the from address, the to address, the amount of gas used, and the value of the transaction (if any). These details are useful for monitoring the activity of the blockchain network and for debugging and troubleshooting smart contracts.

TX HASH	FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE	TYPE
0x24916ace4e4d1b13b43bea56a7960c9683cd8acd03cbb22ab9d9d253362a8573	0x8CF1211225E6Fbd8E7A32D06cb47BEae79c07EF0	Migrations	28813	0	CONTRACT CALL
0xe15a7f4bdef7bdac55ee7b2bb8d10e0ac4bac041490b049af5f8788fe0e4bcd1	0x8CF1211225E6Fbd8E7A32D06cb47BEae79c07EF0	CREATED CONTRACT ADDRESS 0x914A8e2Ad8b8DC31a242CEb188084aA87eA013e8C	1716381	0	CONTRACT CREATION
0x29bfa3b321b363336ddf751539764f50b133b4e00766ca0e1346c7a414d420b	0x8CF1211225E6Fbd8E7A32D06cb47BEae79c07EF0	Migrations	45913	0	CONTRACT CALL
0xddea54cb570fbfc01a38af30baa61c8fa2ab776946f237a816b11f02712fa29e	0x8CF1211225E6Fbd8E7A32D06cb47BEae79c07EF0	CREATED CONTRACT ADDRESS 0xEF72B48A5F60700598f4a57124cfCf4DAE93a82F	250142	0	CONTRACT CREATION
0xc414d2697e0318b88ed92580a98c60f2c022fdb10b17eec27b9160ee5b98798f	0x8CF1211225E6Fbd8E7A32D06cb47BEae79c07EF0	0x5cc372D0B017611b0A3a177992DDAec1DD72222de	119305	0	CONTRACT CALL

Fig. 14 Ganache transactions tab

The Ganache transaction tab shows a list of all the transactions that have been sent to the blockchain. Each transaction is represented by a row in the table and includes information such as the transaction hash, the status of the transaction (pending or confirmed), the gas price, and the gas limit. The transaction tab also displays the sender and recipient addresses, the amount of cryptocurrency transferred, and the time that the transaction was sent. Additionally, the transaction tab provides a detailed view of each transaction when clicked on. This view includes additional information such as the input data that was sent with the transaction, the gas used, and the status of the transaction. The transaction tab is a useful tool for monitoring the activity on the blockchain and for debugging and troubleshooting smart contracts.

Following steps explains each and every step we have to follow during the process of election:-

1. User registration page:-

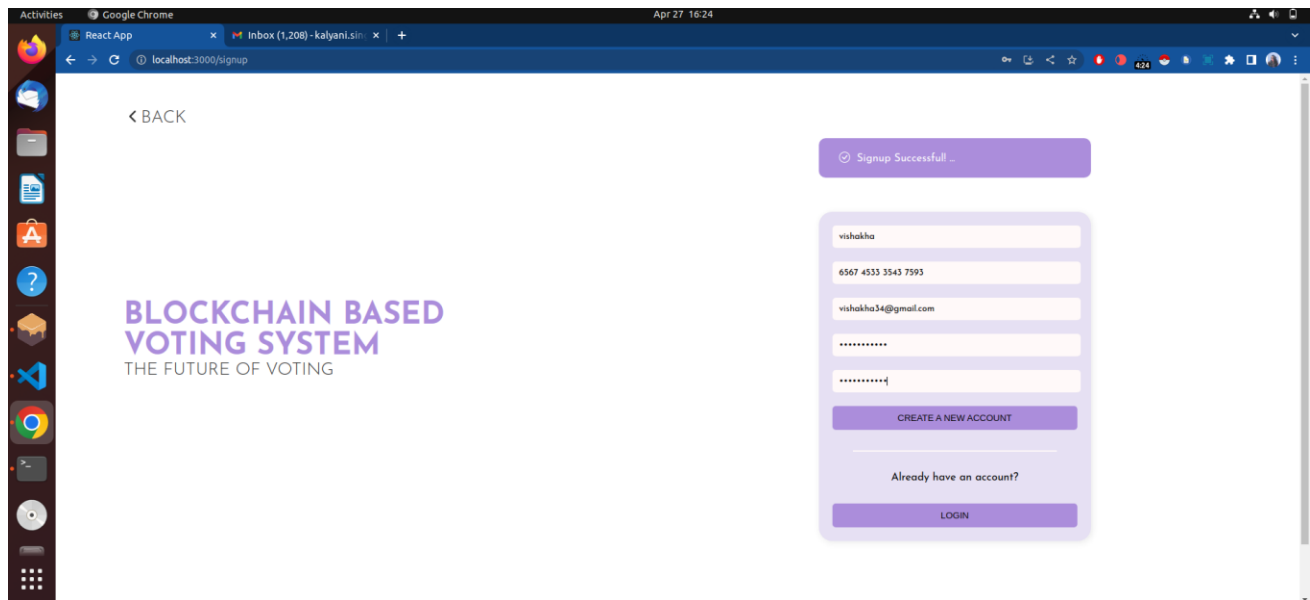


Fig. 15 Registration Page

2. User and Admin sign in page:-

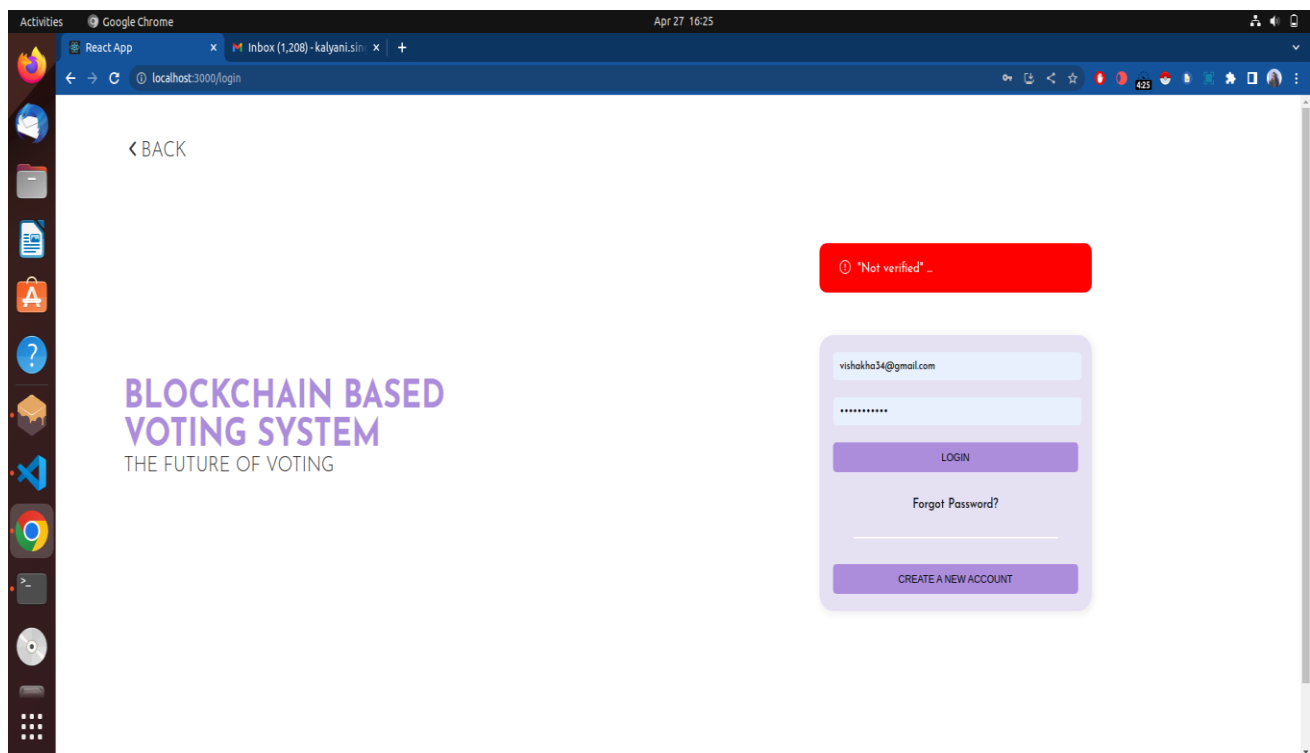


Fig. 16 Login page

3. Admin creating election by entering details of election:-

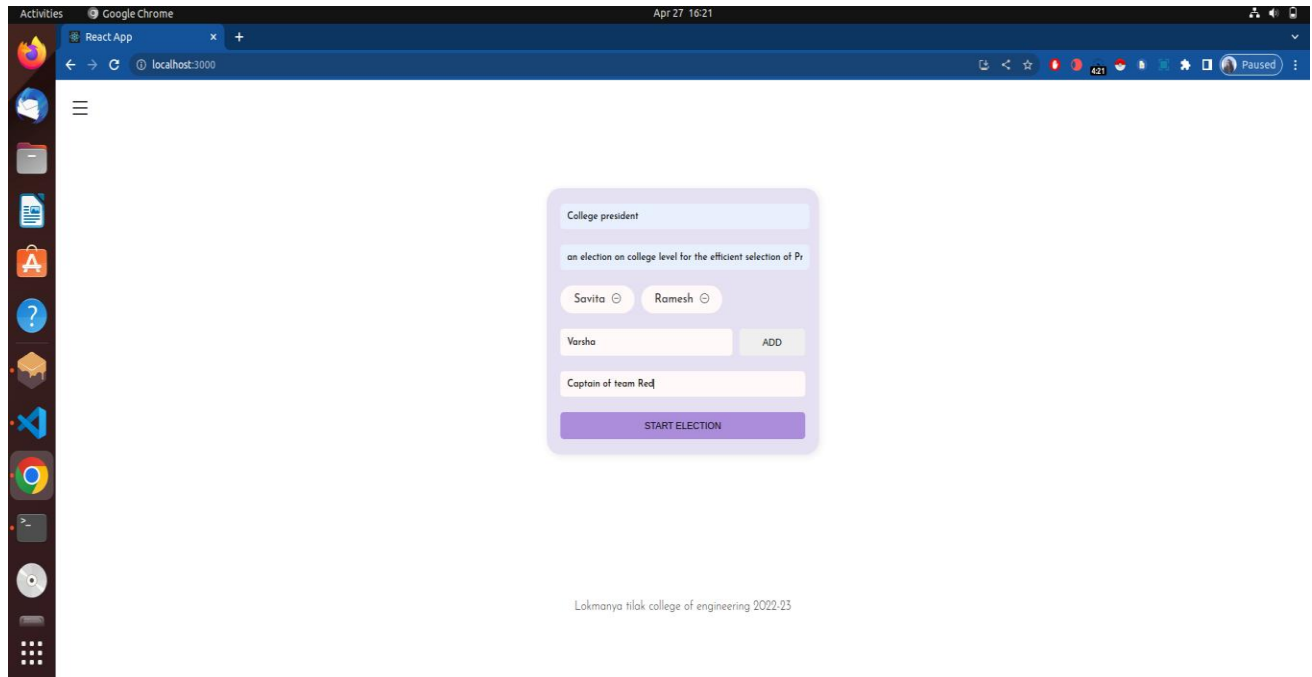


Fig. 17 Creating election page

4. Admin authenticating and verifying voters :-

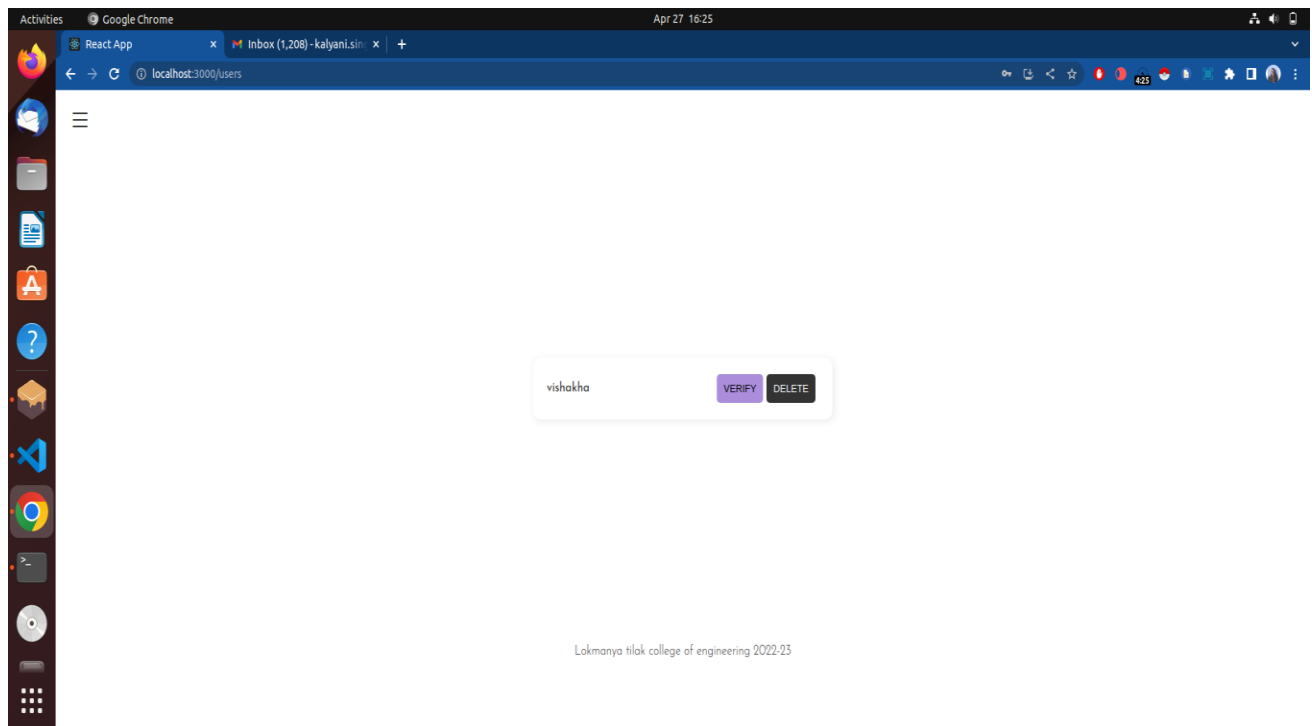


Fig. 18 User verification page

5. After creating election the initial view of poll results look like this:-

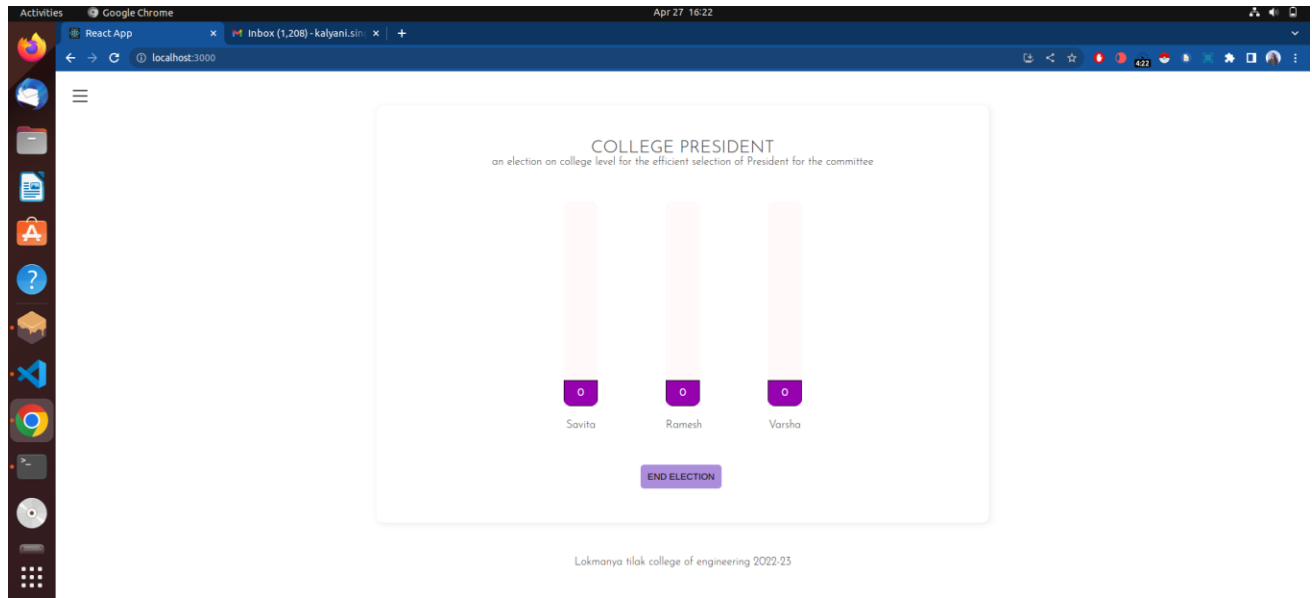


Fig. 19 Initial election result page

6. At the completion of election process result will be shown in such a manner and election will be ended.

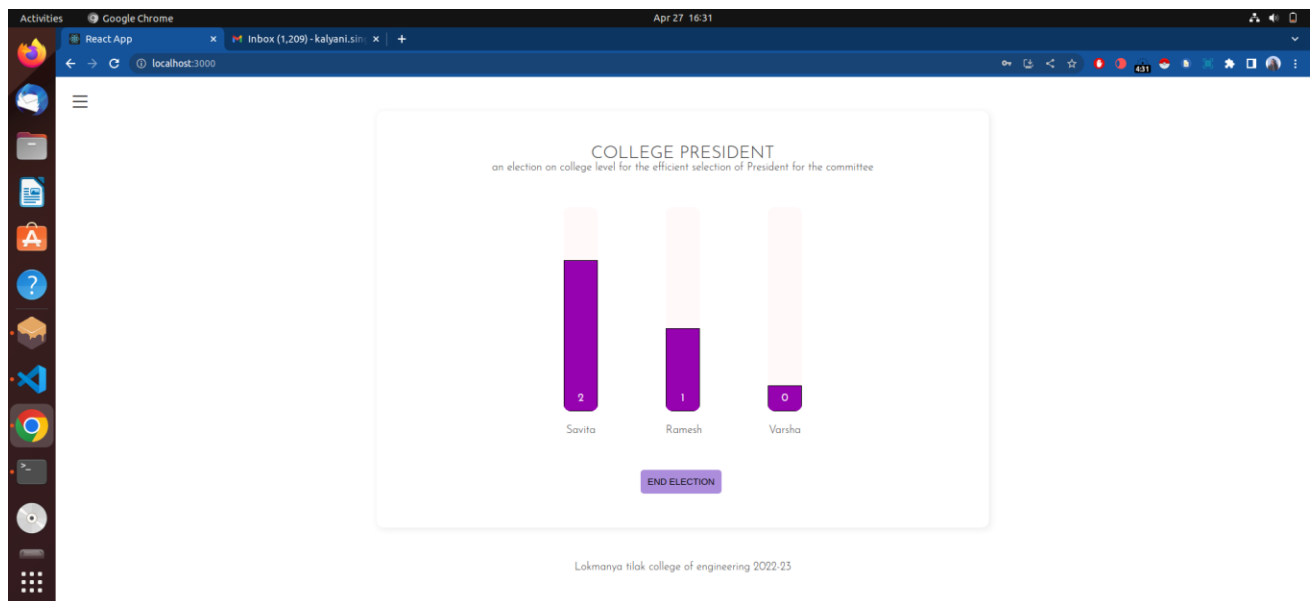


Fig. 20 Final result page

4.2 Conclusion

Blockchain-based e-voting systems are unlikely to fully replace traditional offline voting procedures but can be successfully implemented alongside them. Current e-voting solutions relying on blockchain technology are widely applied not only for conducting national elections but also as a polling tool within enterprises and small organizations. In contrast to other solutions for online voting, solutions that leverage blockchain technology offer improved data security, contain convenient identity verification mechanisms, and make it easier to maintain the right balance between ballot secrecy and voting results verification.

In this project, we introduced a blockchain- grounded electronic voting system that utilizes smart contracts to enable secure and cost-effective election while guaranteeing voters privacy. In conclusion, e-voting systems using blockchain technology have the potential to revolutionize the way we conduct elections. By leveraging the decentralized and immutable nature of blockchain, these systems can provide a more transparent, secure, and trustworthy voting process. The use of smart contracts can help to automate and streamline the voting process, while also ensuring that only authorized individuals can participate. Despite the many benefits of e-voting systems using blockchain, there are also several challenges and limitations that need to be addressed. These include issues related to voter privacy, scalability, and accessibility. Overall, further research and development are needed to fully realize the potential of e-voting systems using blockchain, but the technology holds great promise for the future of democratic elections.

4.3 Future scope

For our project to be more secure and add an additional layer of protection it can be integrated with machine learning and artificial intelligence technologies like Face recognition and finger print recognition to verify user and thereby automate the process of verification. The cost of implementation is too high due to the introduction of blockchain which can be improvised. Thus the project has various future scopes and a lot of them can be discovered with the introduction of the system in the community and asking for feedbacks.

Chapter 5

References

5.1 References (Books, journals and other online references)

1. S.K. Vivek et al., 2020. E-voting systems using blockchain: An exploratory literature survey. *IEEE Xplore*. Available at: <https://ieeexplore.ieee.org/document/9183185> [Accessed August 1, 2022].
2. Parmar, A. et al., 2021. Secure e-voting system using blockchain technology and authentication via face recognition and mobile OTP. *IEEE Xplore*. Available at: <https://ieeexplore.ieee.org/document/9580147> [Accessed August 1, 2022].
3. Khan KM, Arshad J, Khan MM. Investigating performance constraints for blockchain based secure e-voting systems. *Future Generation Computer Systems*. 2020 Apr 1;105:13-26.
4. Kirillov, Denis, Vladimir Korkhov, Vadim Petrunin, Mikhail Makarov, Ildar M. Khamitov, et al., "Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain", *International Conference on Computational Science and Its Applications*, pp. 509-521, 2019.
5. Yi Haibo, "Securing e-voting based on blockchain in P2P network", *EURASIP Journal on Wireless Communications and Networking*, no. 1, pp. 137, 2019.
6. FririkHjálmarsson, Gunnlaugur K. Hreiðsson, Mohammad Hamdaqa and GísliHjalmtýsson, "Blockchain-based e-voting system", 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 983-986, 2018.
7. Harsha V. Patil, Kanchan G. Rathi and Malati V. Tribhuwan, "A Study on Decentralized E-Voting System Using Blockchain Technology", *International Research Journal of Engineering and Technology (IRJET)*, vol. 05, no. 11, Nov 2018.
8. Fusco Francesco, Maria Ilaria Lunesu, Filippo Eros Pani and Andrea Pinna, "Crypto-voting a Blockchain based e-Voting System", *KMIS*, pp. 221-225, 2018. 31

9. Ganji Raghavendra and B. N. Yatish, ELECTRONIC VOTING SyS-TEM USING BLOCKCHAIN., 2018.
10. Yu Bin, Joseph K. Liu, Amin Sakzad, Surya Nepal, Ron Steinfeld, Paul Rimba, et al., "Platform-independent secure blockchain-based voting system", International Conference on Information Security, pp. 369-386, 2018.
11. Rajput, S., Singh, A., Khurana, S., Bansal, T., & Shreshtha, S. (2019). Blockchain Technology and Cryptocurrencies. 2019 Amity International Conference on Artificial Intelligence (AICAI). J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
12. Andrian, H. R., Kurniawan, N. B., & Suhardi. (2018). Blockchain Technology and Implementation : A Systematic Literature Review. 2018 International Conference on Information Technology Systems and Innovation (ICITSI). R. Nicole.
13. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
14. Votereum: An Ethereum-based E-voting system :Linh Vo-Cao-Thuy, Khoi Cao-Minh, Chuong Dang-Le-Bao and Tuan A. Nguyen, 2019, "Votereum: An Ethereum-based E-voting system", University of Information Technology Vietnam National University HCMC, Vietnam.
15. Survey on Blockchain Based E-Voting Recording System Design: G Bhavan, i"Survey on Blockchain Based E-Voting Recording System Design", 2018.
16. M. Flint, "The Future of Democracy: Blockchain Voting," no. April, pp. 1–21, 2016.

17. Umut Can Çabuk¹, Eylül Adıgüzel², Enis Karaarslan² (2018); A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems; International Journal of Advanced Research in Computer and Communication Engineering.
18. Aayushi Gupta¹, Jyotirmay Patel², Mansi Gupta¹, Harshit Gupta¹ (2017); Issues and Effectiveness of Blockchain Technology on Digital Voting; International Journal of Engineering and Manufacturing Science. ISSN 2249-3115 Vol. 7, No.1 (2017).
19. PavelTarasov and Hitesh Tewari (2017); the Future of E-Voting; IADIS International Journal on Computer Science and Information Systems Vol.12, No. 2, pp. 148- 165 I.
20. Dalia, K., Ben, R. , Peter Y. A, and Feng, H. (2012). “A fair and robust voting system by broadcast.”, 5th International Conference on E-voting, 2012.

Annexure 1

(Any paper presentation, research funding, sponsorship information/ certificate may be included.)

