# PROJECT REPORT

**ELECTRONIC VOTING SYSTEM**

|  |  |
| --- | --- |
| **Date** | 27 October 2023 |
| **Team ID** | NM2023TMID05889 |
| **Project Name** | Electronic Voting System |

# TEAM MEMBERS

|  |  |
| --- | --- |
| **S.MOHAMED TAWFEEQ** | **812420104060** |
| **B.MOHAN** | **812420104063** |
| **M.MUFTHI MOHAMED** | **812420104064** |
| **A.MOHAMED SARFUDEEN** | **812420104059** |

1. **ABSTRACT**
2. **INTRODUCTION**
   1. Project Overview
   2. Purpose
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
4. **REQUIREMENTANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution Architecture
6. **BLOCKCHAIN TECHNOLOGY AND ITS DEPENDENCIES**
   1. Architecture
   2. Block & Algorithm
   3. Challenges Faced by Blockchain Technology
7. **CODING & SOLUTIONING**
   1. Feature 1
   2. Feature 2
   3. Database Schema
8. **PERFORMANCE TESTING**
   1. Experimental Setup
   2. Data Collection for Performance Evaluation
   3. Result
9. **ADVANTAGES & DISADVANTAGES**
10. **CONCLUSION**
11. **FUTURE SCOPE**
12. **APPENDIX**

Source Code

GitHub & Project Demo Link

# ABSTRACT:

The modern world is witnessing a paradigm shift in the way elections are conducted, moving away from traditional paper-based voting systems towards electronic methods to make the process more efficient, secure, and transparent. This abstract explores the integration of blockchain technology into electronic voting systems to address the pressing challenges of trust, security, and accessibility in electoral processes.

Blockchain technology, known for its decentralized and immutable ledger system, offers a unique solution to the perennial concerns of electoral fraud, data manipulation, and transparency. This paper discusses the fundamental principles of blockchain technology and how they can be harnessed to create a secure and transparent electronic voting system.

The proposed electronic voting system leverages blockchain's attributes such as decentralization, cryptographic hashing, and consensus algorithms to ensure the integrity and authenticity of each vote cast. The immutability of the blockchain ledger eliminates the possibility of tampering with the election results, enhancing trust in the electoral process.

Furthermore, the use of smart contracts on the blockchain can automate various aspects of the election process, such as voter registration, identity verification, and result tabulation, reducing the need for intermediaries and the associated human errors and vulnerabilities.

This abstract also explores the potential challenges and considerations related to implementing a blockchain-based electronic voting system, including scalability, accessibility, and privacy concerns. It emphasizes the need for further research, pilot projects, and collaboration between governments, technologists, and experts in the field to overcome these challenges and develop a robust and user-friendly electronic voting system.

In conclusion, the integration of blockchain technology into electronic voting systems has the potential to revolutionize the way democracies conduct elections by enhancing security, transparency, and trust in the electoral process. It offers a promising path towards more efficient, accessible, and inclusive elections, setting the stage for a new era of democracy.

# INTRODUCTION

**Project Overview:**

The "Electronic Voting System with Blockchain Technology" project is a groundbreaking initiative aimed at revolutionizing the way elections are conducted by combining the power of electronic voting systems with the security and transparency of blockchain technology. This project seeks to address the critical challenges associated with traditional voting methods, such as electoral fraud, data manipulation, and lack of transparency.

This project has several key objectives. First, it aims to develop a secure and transparent electronic voting system that utilizes blockchain technology. This system is designed to ensure the integrity and authenticity of each vote through the use of decentralized ledger technology. Additionally, the project seeks to streamline the election process by automating voter registration, identity verification, and result tabulation using smart contracts. The overarching goal is to enhance trust in the electoral process, increase accessibility, and improve overall efficiency. As part of this project, we will also explore and address potential challenges associated with the implementation of a blockchain-based voting system.

**Key Components:**

* Blockchain Infrastructure: One of the foundational components of this project is the establishment of a robust, decentralized blockchain network. This network will serve as the foundation of the voting system, and it is crucial to select an appropriate consensus mechanism to secure the network and verify transactions. Furthermore, the project will involve the development and deployment of smart contracts for managing various aspects of the electoral process.
* User Interface: To ensure widespread adoption and usability, the project will create a user-friendly interface for voters. This interface will allow citizens to register, cast their votes, and verify their selections. Accessibility is a key consideration, and efforts will be made to accommodate all citizens, including those with disabilities.
* Security and Identity Verification: Protecting the security and privacy of voter data is paramount. The project will implement strong encryption and cryptographic techniques to safeguard voter data and maintain anonymity. Additionally, a secure identity verification system will be developed to prevent voter impersonation and ensure that only eligible voters can participate.
* Result Tabulation and Transparency: Automation of the vote counting and result tabulation processes is a central feature of the project. Smart contracts based on blockchain technology will facilitate this automation. Furthermore, the project will provide real-time, public access to the voting ledger to ensure transparency and accountability in the electoral process.

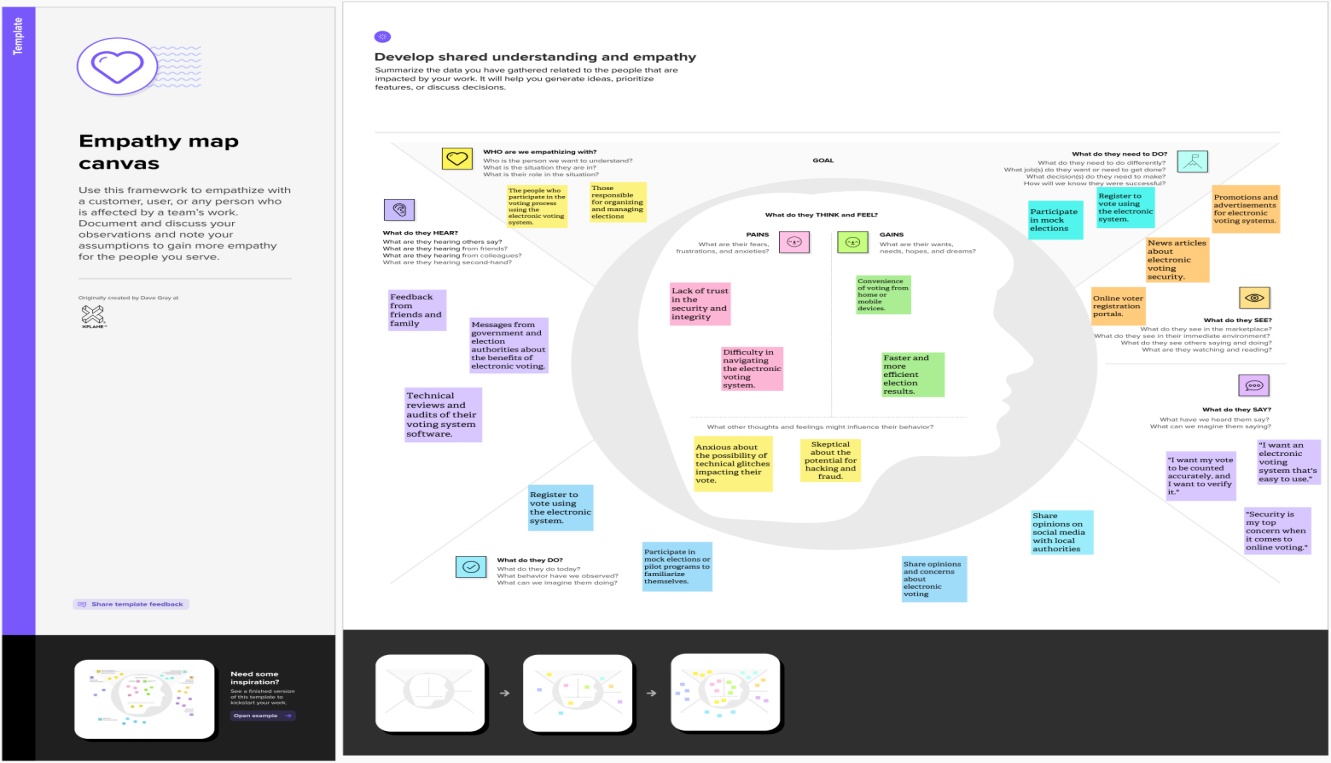
# Purpose:

The purpose of an electronic voting system with blockchain technology is to address several key objectives and challenges associated with traditional voting methods, while harnessing the unique capabilities of blockchain technology. The primary purposes of such a system include:

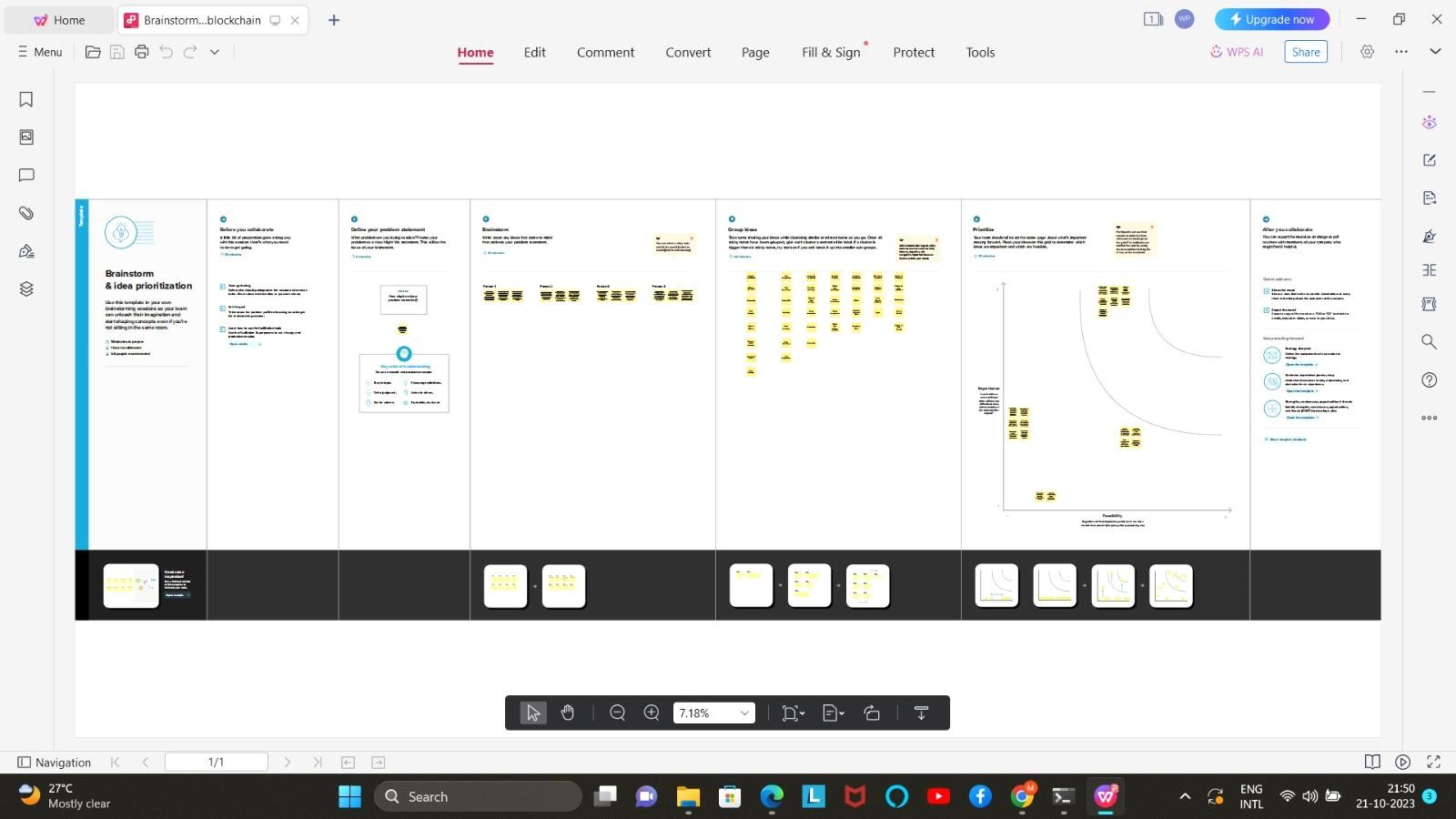
* Enhanced Security: Implementing blockchain technology in electronic voting systems provides a high level of security. The immutability and cryptographic features of blockchain make it extremely difficult for unauthorized individuals to manipulate or tamper with the voting data. This enhances the overall integrity of the electoral process and helps prevent electoral fraud.
* Transparency and Trust: Blockchain's decentralized ledger ensures transparency in the voting process. Every transaction, in this case, every vote, is recorded on the blockchain and is publicly accessible. This transparency builds trust among voters, election authorities, and other stakeholders by allowing them to independently verify the results and the integrity of the voting process.
* Data Integrity: Blockchain technology ensures that once a vote is recorded, it cannot be altered or deleted. This guarantees the integrity of the electoral data, eliminating the possibility of unauthorized changes to the results, protecting against data manipulation, and safeguarding the sanctity of the electoral process.
* Accessibility: Electronic voting systems, especially when integrated with blockchain, can be designed to be more accessible to a wider range of voters. This includes provisions for remote voting, improved usability, and accommodations for individuals with disabilities, making the electoral process more inclusive.
* Efficiency and Accuracy: Automation through smart contracts on the blockchain can streamline various aspects of the voting process, such as voter registration, identity verification, and result tabulation. This not only reduces the potential for human errors but also accelerates the vote-counting process, delivering more accurate and timely results.
* Reduced Costs: Over time, electronic voting systems with blockchain technology can lead to cost savings. The reduction in paper-based processes, the elimination of the need for physical polling stations, and more efficient administrative procedures can contribute to lower election-related expenses.
* Civic Engagement: By leveraging technology and providing secure, user-friendly interfaces for voters, electronic voting systems can encourage greater participation in the democratic process. This is particularly important in engaging younger generations and those who might otherwise face barriers to voting.
* Auditability and Accountability: Blockchain's auditability features allow for detailed tracking of all transactions on the network. This means that any discrepancies or irregularities can be traced back to their source, promoting accountability among election officials and ensuring that the process adheres to established rules and regulations.
* Resilience to DDoS Attacks: Blockchain-based voting systems can be designed to be resilient to Distributed Denial of Service (DDoS) attacks, which are a common concern for online systems. The decentralized nature of blockchain reduces the risk of a single point of failure, making the system more robust against attacks.
* Global and Remote Voting: Blockchain-based voting systems can potentially allow citizens living abroad to participate in their home country's elections, enhancing representation for expatriate communities and enabling remote voting in cases of emergencies or special circumstances.

# IDEATION & PROPOSED SOLUTION

**Empathy Map Canvas**



# Ideation & Brainstorming



**REQUIREMENT ANALYSIS**

# Functional requirement:

1. Voter Registration and Authentication:

* User-friendly voter registration process.
* Secure identity verification mechanisms (e.g., biometrics, government-issued IDs).
* Prevention of duplicate or unauthorized registrations.

1. Ballot Creation and Distribution:

* Ability to create and customize digital ballots for different elections.
* Secure and efficient distribution of digital ballots to eligible voters.

1. Voting Process:

* User-friendly interface for casting votes.
* Anonymity and privacy protection for voters.
* Verification of the voter's eligibility in real-time.
* Option for voters to review and change their choices before final submission.
* Confirmation receipt for voters to verify their votes were recorded correctly.

1. Blockchain Integration:

* Implementation of a decentralized blockchain network.
* Selection of a consensus mechanism (e.g., Proof of Work, Proof of Stake).
* Integration of smart contracts for vote recording, tabulation, and result publishing.

1. Security Measures:

* Strong encryption for data protection.
* Protection against Distributed Denial of Service (DDoS) attacks.
* Multi-factor authentication for voters and election officials.
* Measures to prevent unauthorized access or tampering with the blockchain.

1. Transparency and Auditability:

* Real-time, public access to the blockchain ledger.
* Tools for independent auditing and verification.
* Timestamps and cryptographic hashing for vote records.

1. Result Tabulation:

* Automatic and tamper-proof result tabulation through smart contracts.
* Real-time updating of results as votes are cast and verified.

1. Accessibility:

* Inclusive design for people with disabilities.
* Multiple language support to accommodate a diverse electorate.

1. Resilience and Redundancy:

* Backup and redundancy measures to ensure system availability.
* Disaster recovery plans to handle unexpected system failures.

1. Usability and User Support:

* Clear and intuitive user interfaces for both voters and election officials.
* Comprehensive user training and support services.

# Non-Functional Requirements:

1. Security:

* Data Encryption: All data, including voter information and votes, must be encrypted to protect against unauthorized access.
* Access Control: Implement strict access controls to ensure that only authorized individuals can interact with the system.
* Resilience to Attacks: The system should be resilient to various types of cyberattacks, including Distributed Denial of Service (DDoS) attacks and hacking attempts.

1. Scalability:

* The system should be able to handle a variable number of users and transactions during peak voting periods, ensuring that it remains responsive and available.

1. Performance:

* Response Time: The system should provide fast response times to ensure that voters can complete their transactions efficiently.
* Throughput: The system should support a high throughput of transactions to accommodate a large number of votes within a short timeframe.

1. Availability:

* Ensure that the system is highly available during election periods, with minimal downtime and maintenance windows.

1. Reliability:

* The system should be highly reliable, with minimal errors or disruptions, to maintain the integrity of the voting process.

1. Auditability:

* The system should provide detailed and tamper-evident audit logs to allow for post-election auditing and forensic analysis.

1. Compliance:

* Ensure that the system complies with all relevant legal and regulatory requirements, including data protection and privacy laws.

1. Usability:

* The user interface should be intuitive, accessible, and user-friendly to cater to voters of all backgrounds and abilities.

1. Accessibility:

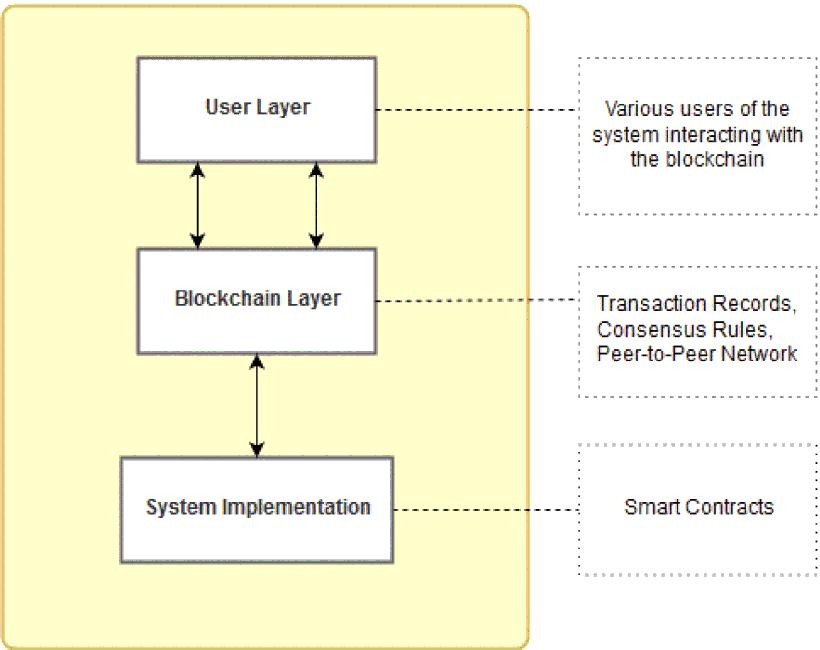
* The system should be accessible to voters with disabilities, adhering to accessibility standards to ensure inclusivity.

1. Interoperability:

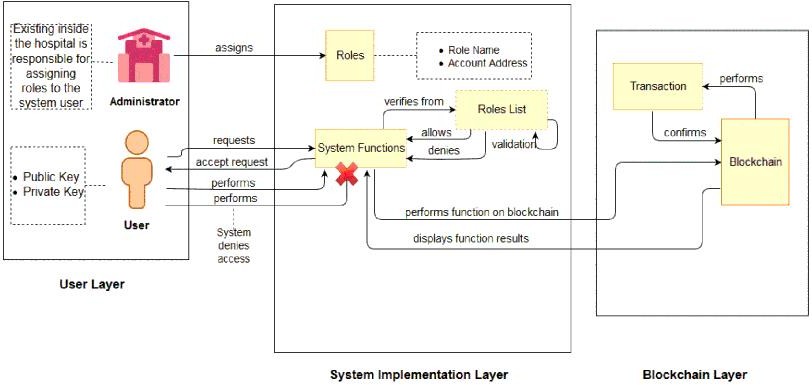
* The system should be designed to work with various hardware and software configurations, ensuring compatibility with different devices and platforms.

# PROJECT DESIGN

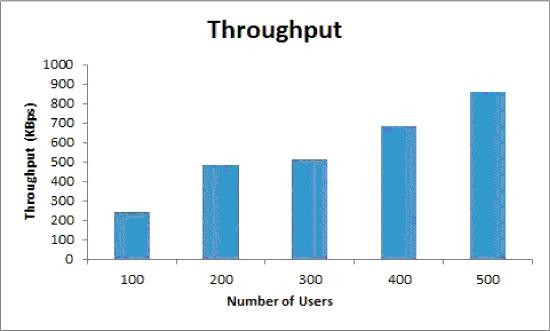
**Data Flow Diagrams**



1. System Design of Proposed Framework

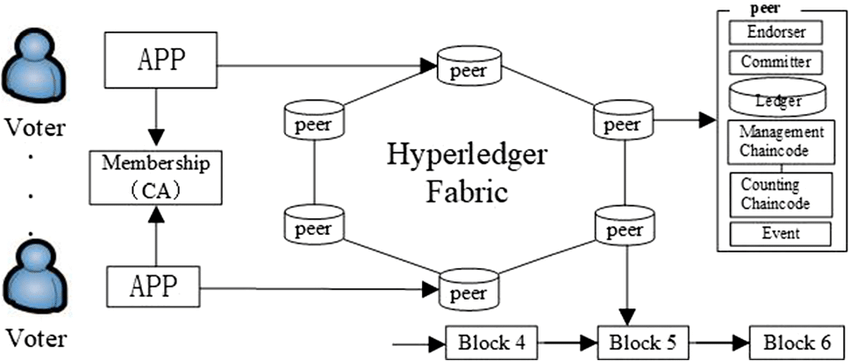


1. User Interaction With DAPP



1. Throughput of Proposed Framework

# Solution Architecture



# BLOCKCHAIN TECHNOLOGY AND ITS DEPENDENCIES

**Architecture:**

To understand the blockchain architecture let us use the following figure 1 that explains the whole process of a transaction being send from a user on the blockchain network.

A new transaction being sent by a user on the blockchain network suggests that a new block is created. A block in the blockchain is used for keeping transactions in them and these blocks are distributed to all of the connected nodes in the network. That transaction placed inside a block is broadcasted to all of the nodes in the network. All the nodes in the network have a copy of the complete blockchain that helps them in verification process. When a block containing the user transaction is broadcasted to all of the connected nodes, they

verify that the block is not tampered by any means. If this verification results in success then the nodes add that block in their own copy of blockchain.

This whole process of the block being added on the blockchain is done by the nodes reaching upon a consensus where they decide which blocks are valid to be added on the blockchain and which are not. This validation is performed by the connected nodes using some known algorithms to verify the transaction and to ensure that sender is an authenticated part of the network. When a node succeeds in performing the validation that node is rewarded with crypto- currency. This process of validating the transaction is known as mining and the node performing this validation is known as miner.

After validation is done that block is added to the blockchain.

After the whole process of validation is performed the transaction is completed.

Some basic concepts of blockchain technology can be understood in the following descriptions.

# Block & Algorithm

Block:

As explained earlier blockchain are formed together by a number of blocks connected together in a peer-to-peer network thus making a decentralized application. The header of these blocks contains hashes of previous blocks in them. A block contains three things in it which are data, hash of current block and hash of previous block. The data could be anything as it depends on the type of blockchain. As in case of bitcoin, the data consists of coins that are actually electronic cash [13]. The hash that is stored in these blocks contains a SHA-256 cryptographic algorithm which is used for unique identification of a block on the chain.

Consensus Algorithm:

Each block that is added on the chain would need to follow some consensus rules for it to be added on the blockchain. For this purpose blockchain technology uses co

nsensus algorithms. The most common consensus algorith

used is Proof of Work (PoW) algorithm and it was used by Nakamoto [13], in bitcoin network. The basic working of this algorithm is that there are number of nodes or participants on a blockchain network so when a transaction is requested to be added on the network by any participating node it needs to be calculated. This process is called mining and the nodes that are performing these calculations are miners .

Algorithm 1 Smart Contract for Patient Records Assign Roles:

function Define Roles (New Role, New Account) add new role and account in

roles mapping end function Add Data:

function Add Patient Record (contains variables to add data) if (msg.sender == doctor) then

add data to particular patient’s record else Abort session

end if

end function Retrieve Data:

functionView Patient Record (patient id) if (msg.sender == doctor || patient) then if (patient id) == true then

retrieve data from specified patient (id) return (patient record) to the account that requested the retrieve operation

else Abort session end if

end if

end function Update Data:

function Update Patient Record (contains variables to update data) if (msg.sender == doctor) then

if(id == patient id && name == patient name) then update data to particular patient’s record

return success else return fail end if

else Abort session end if

end function Delete Data:

function Delete Patient Record (patient id)

if (msg.sender == doctor) then if (id == patient id) then

delete particular patient’s record return success

else return fail end if

else Abort session end if

end function

# Challenges Faced by Blockchain Technology

1. Scalability and Storage Capacity

Storage of data on the blockchain causes two main problems, i.e., confidentiality and scalability. The data on the blockchain is visible to everyone that is present on the chain this makes the data vulnerable which is not a desired outcome for a decentralized platform. The data stored on the blockchain would contain patient medical history, records, lab results, X-rays reports, MRI results and many other reports, all of this voluminous data is to be stored on the blockchain that would highly affect the storage capacity of blockchain.

1. Lack of Social Skills

The way the blockchain technology works is understandable by very few people. This technology is still in its initial phases and is constantly evolving. Moreover, the shift from trusted EHR systems to the blockchain technology would take

time as hospitals, or any other healthcare institutes need to completely shift their systems to blockchain.

1. Lack of Universally Defined Standards

As this technology is still in the initial phases and is constantly evolving so there is no defined standard for it. Due to this the implementation of this technology in healthcare sector would also take more time and effort. As it would require certified standards from international authorities that overlook the standardization process of any technology. These universal standards would benefit in deciding upon the data size, data format and type of data that could be stored on the blockchain. Moreover, the adaptation of this technology would become easier due to the defined standards, as they could be easily enforced in the organizations.

# CODING & SOLUTIONING

**Features 1**

1. Decentralization

With blockchain the information is distributed across the network rather than at one central point. This also makes the control of information to be distributed and handled by consensus reached upon by shared input from the nodes connected on the network. The data that was before concentrated at one central point is now handled by many trusted entities.

1. Data Transparency

Achieving data transparency in any technology is to have a trust based relationship between entities. The data or record at stake should be secured and temper proof. Any data being stored on the blockchain is not concentrated at one place and is not controlled by one node but is instead distributed across the network. The ownership of data is now shared and this makes it to be transparent and secure from any third party intervention.

1. Security and Privacy

Blockchain technology uses cryptographic functions to provide security to the nodes connected on its network. It uses SHA-256 cryptographic algorithm on the hashes that are stored on the blocks. SHA stands for Secure Hashing Algorithm, these hashes provide security to the blockchain as data integrity is ensured by them. Cryptographic hashes are strong one way functions that generate checksum for digital data that cannot be used for data extraction. This makes blockchain as such a decentralized platform made secure by the cryptographic approaches which makes it to be a good option for privacy protection of certain applications.

# Features 2

Blockchain technology was designed by Nakamoto [13], the basic idea was to have a cryptographically secured and a decentralized currency that would be helpful for financial transactions. Eventually, this idea of blockchain was being used in various other fields of life; healthcare sector also being one of them intends to use it. A number of researchers have carried out the research on this area, these research works focus on the fact that whether the idea of using blockchain for healthcare sector is feasible or not. They also identify the advantages, threats, problems or challenges associated by the usage of this technology. Some researchers also discussed the challenges that would be faced while actually implementing this on a larger scale.

1. Theoretical/Analytical Blockchain-Based Research

Gordon and Catalini [14], conducted a study that focused on the methods by which blockchain technology would facilitate the healthcare sector. They identified, that healthcare sector is controlled by hospitals, pharmaceutical companies and other involved third parties. They specified data sharing as the key reason why blockchains should be used in healthcare. This study also identified four factors or approaches due to which healthcare sector needs to transform for usage of blockchain technology. These include way for dealing of digital access rights, data availability, and faster access to clinical records and patient identity. It also discusses the on-chain and off-chain storage of data. The study also included the challenges or barriers faced by usage of blockchain technology these were huge volume of clinical records, security and privacy, patient engagement.

Eberhardt and Tai [20], conducted a study to understand possible approaches to solve the scalability problem of blockchain and also to identify such projects that intend to solve this problem. They define blockchain as composition of various computational and economical concepts based on peer-to-peer system. The aim of this study was to find which data should be stored on-chain and what could be stored off-chain. This study presented five patterns for off-chain storage of data and also includes the basic ideas and implementation framework of these patterns. The authors explain on-chain data is any data that is stored on the blockchain by performing transactions on it. While off-chain data storage is to place data elsewhere on any other storage medium but not on-chain and it also would not include any transactions.

Vujičić et al. [21], presented an overview of blockchain technology, bitcoin and Ethereum. The authors define that information technology landscape is constantly changing and blockchain technology is benefiting the information systems. They explained bitcoin as a peer-to-peer distributed network used for performing bitcoin transactions. They also defined that proof-of-work consensus algorithm along with the mining of blockchain concept. The authors emphasize on the fact that scalability is a severe problem faced by blockchain and that certain solutions are proposed for solution of scalability problem these include SegWit and Lightning, Bitcoin Cash and Bitcoin Gold. The paper also explained Ethereum and its dependencies and it also differentiates Ethereum blockchain from bitcoins’ blockchain.

Wang et al. [22], conducted a study that focused on smart contracts and its application in blockchain technology. They first introduce the smart contracts, their working framework, operating systems and other important concepts attached with them. The authors also discuss that how could smart contracts be used for the new concept of parallel blockchains. They identify that reason of using smart contracts in blockchain is due to the decentralization that is offered through the programming language code written in them. After introducing the basics of smart contract the author explained the various layers of blockchain that combine together to keep system functioning. These layers are data, network, consensus, incentive, contract, and application layer. The paper not only discusses the architecture and framework followed by smart contracts but it also gives an insight on its applications and challenges. The paper also discusses an important future trend of parallel blockchain that intends to create such blockchain that can optimize two different but important modules.

Kuo et al. [23], conducted a review that discussed several applications of blockchain in biomedical and healthcare sector. The authors identified that using blockchains for this domain offers many advantages and some of these are

decentralization, persistence of clinical or medical records, data pedigree, and continuous accessibility to data and lastly secure information being accessible to biomedical or healthcare stakeholders. The limitations of blockchain technology were identified to be, confidentiality, speed, scalability and threat of malicious attack, i.e., 51% attack. The authors identified these limitations to be critical for healthcare or biomedical sector as they are being used to store sensitive medical or clinical records. The solution to these problems were presented by authors to store sensitive medical data off-chain, encryption of data to ensure confidentiality, and lastly to use VPNs (Virtual Private Networks) to ensure safety from malicious attacks.

1. Prototype/Implementation Blockchain-Based Research

Sahoo and Baruah [24], proposed a scalable framework of blockchain using Hadoop database. In order to solve the scalability problem of blockchain, they proposed to use the scalability provided by the underlying Hadoop database along with the decentralization provided by the blockchain technology. They used the method to store blocks on the Hadoop database, the blockchain on top of this framework includes all of the needed dependencies of blockchain but the blocks are stored on Hadoop database to improve scalability of the blockchain technology. To tackle the scalability problem of blockchain platform this study offers to use Hadoop database system, along with SHA3-256 for hashing used for transactions and blocks. The programming language used for this architecture was Java. This study, was helpful in understanding that blockchain can be used with other platforms that are scalable to improve or solve the scalability of this platform.

Zhang et al. [25], proposed a scalable solution to the blockchain for clinical records. The basic aim of this study was to design such an architecture that complies with the Office of National Coordinator for Health Information Technology (ONC) requirements. This study identified the barriers that this technology faces mainly include concerns related to privacy, security of blockchain, and scalability problems related to huge volume of datasets being transmitted on this platform, and lastly there is no universal standard enforced for data being exchanged on blockchain. This study also include a demonstration of a decentralized application (DAPP) based on the design formulated on the ONC requirements as mentioned before. They also included the lessons learnt and how can FHIR chain be improved.

Kim et al. [26] proposed a system for management of medical questionnaires and the aim of this system is data sharing through blockchain technology. The authors explain that selection of data storage and sharing of medical

questionnaire is to use this data for further medical and clinical research purposes. They emphasized that it would be helpful for developing diagnosis system, resolving terminologies being used in EHR systems and security issues associated with these systems was also a reason due to which authors selected blockchain technology for their proposed framework. This study contains two main functions, i.e., to create, store the data gathered by questionnaires and to share that data. Another benefit proposed by the system is the validation of the questionnaire being submitted in the system. The questionnaires that are added on this system are first validated to be correct specified format and then are parsed to differentiate the personal data and specific data related to questionnaire results. This would ensure that data could be shared for future research purposes. The authors also address the scenario when a third party requests to access this questionnaire data, this would need the patients’ permission that is asked by the doctor to let third party view that data.

# DataBase Schema

Entities and Their Attributes:

1. Patients:
   * Patient ID (Primary Key)
   * First Name
   * Last Name
   * Date of Birth
   * Contact Information
2. Healthcare Providers:
   * Provider ID (Primary Key)
   * Name
   * Specialization
   * Contact Information
3. User Accounts:
   * User ID (Primary Key)
   * Username
   * Password (hashed)
   * Role (Patient, Healthcare Provider, Administrator)
4. Electronic Health Records (EHRs):
   * Record ID (Primary Key)
   * Patient ID (Foreign Key)
   * Provider ID (Foreign Key)
   * Date of Entry
   * Diagnosis
   * Treatment Plan
   * Lab Results
   * Medication History
   * Notes
5. Access Control:
   * Access Control ID (Primary Key)
   * Patient ID (Foreign Key)
   * Provider ID (Foreign Key)
   * Consent Granted (Boolean)
   * Consent Expiry Date
   * Timestamp of Consent
6. Blockchain Transactions:
   * Transaction ID (Primary Key)
   * Record ID (Foreign Key)
   * Transaction Type (Create, Read, Update, Delete)
   * Timestamp
   * Digital Signature Table Relationships:

* Each patient is associated with multiple EHR records.
* Each healthcare provider can be associated with multiple EHR records.
* Users are linked to either patients or healthcare providers through their role.
* Access Control defines the permissions for accessing specific EHR records.
* Blockchain Transactions record all actions related to EHRs, including creation, updates, and access requests.

# PERFORMANCE & TESTING

In this section we evaluate the performance of the proposed framework. By assessing the performance we can mitigate the risks associated with this novel technology that is understandable by very few individuals.

# Experimental Setup

For testing performance of the proposed framework we have conducted experiments by using the following configurations:

Intel Core i7-6498DU CPU @ 2.50GHz 2.60 GHz processor And 8.00 GB of memory with Windows 64-bit OS (version 10)

We developed our proposed framework by using the Solidity which is programming language of Ethereum. JavaScript and Python are encapsulated in the Solidity language which is provided by the Ethereum to write code in smart contracts.

# Data Collection for Performance Evaluation

This section explains what kind of data is used for evaluation of performance of the proposed framework. This section also discusses the metrics that are used to explain the results of this performance evaluation being conducted.

1. Transaction Data

To evaluate the performance of the proposed framework following transaction data with its details are used.

Transaction Deployment Time (tx1)

It is defined as the time when transaction gets deployed. In Ethereum, a smart contract is deployed using the transaction so this deployment time refers to that time.

Transaction Completion Time (tx2)

It is defined as the time when the transaction is completed and confirmed by the blockchain which in this case is Ethereum.

1. Evaluation Metrics

The metrics used for evaluation include the execution time, latency and throughput of the proposed framework. These are explained briefly as follows:

Execution Timeis defined as time duration (in seconds) between the transaction confirmation and its execution in the blockchain network. Mathematically, it is (max (tx2) - min (tx1)).

Throughputrefers to the amount of data that could be transferred from one location to another in a unit amount of time.

Latencyis known as the delay that occurs when a system component is waiting for another component of the system to respond to an action. In terms of time it could be referred as the difference of deployment and completion time of transaction.

# Results

1. Performance Assesment

In order to understand how our proposed framework would perform in real-case scenario of various users performing different functions on the framework we conducted performance evaluation using Apache JMeter version 5.1.1 and Apache Version 2.00. Apache JMeter is a desktop performance testing tool which is used for analysis and testing of applications.

a: Average Execution Time

The execution time increases with the number of transactions being increased. These transactions are performed for the various functions that are included in the smart contract whose algorithm is defined in Section V. When there is only one user using the system the functions Assign Roles, Add Patient Recordsand View Patient Records would take 18.29 sec, 1 min 48 sec and 50 sec respectively for these functions to be executed. This time would increase when 100 users are using the system simultaneously.

b: Throughput

Algorithm 1 explains various functions that are included in the smart contract of the proposed framework. By using JMeter we simulated number of users from 100 users to 500 users (with period of 10 to 35), who are using the system and performing its various functions. In JMeter the throughput is represented in Data/time i.e. KB/sec units. While conducting the experiments we simulated the number of users as specified above and evaluated the performance of the system. These simulations are run on the proposed framework and at the end throughput is analyzed.

c: Average Latency

Latency as defined earlier is the delay or difference in time when one system component sends a request and a response is generated by any other system component. The difference between these two actions is defined as latency. Here we have evaluated the average latency of the proposed framework by using JMeter. While evaluating the latency of the proposed framework we simulated the number of users by JMeter. In JMeter latency is measured in terms of milliseconds.

1. Performance Evaluation (Transaction)

Every transaction on Ethereum contains a data payload field. Data payload is included in that transaction which is meant to invoke smart contract functions. This data payload is in the hex-serialized format and has bytes associated with it. Here we would discuss two functions from Algorithm 1 in order to understand the data payload included in the transactions being generated.

Data payload is the optional field of a transaction which is only used when there is some form of interaction with contract functions. It has two important parts,

* + Function Selector
  + Function Arguments

The function selector are first 4 bytes of Keccak-256 hash, it is used for identification of the smart contract function which is being invoked. The function arguments include various static and dynamic element types which have different rules for encoding them in payload.

1. Comparison of Proposed Framework With Related Work

We also discuss some parameters that are present in our framework and are used for comparison with the related work in this domain. While ensuring the presence of these parameters in the framework it is also considered that it would not compromise the security and privacy of the system. For this both security and privacy are discussed in each of the parameters discusses below.

1. Scalability

Scalability in simpler terms refers to the ability of an information system to perform it functions well in such situations when the storage volume of the system increases or decreases. In case of blockchain technology scalability is an issue that needs some permanent solution. As data size or volume is increasing on the blockchain. Our proposed system used the off-chain storage mechanism as the patient’s data stored on the blockchain contains the basic information of patient along with the IPFS hash, i.e., the off-chain scaling solution used in our proposed system framework. This solves the scalability issue mentioned as now huge volume of patient medical record is not stored on the blockchain. As, the data size being stored on the blockchain has now decreased the transactions could also be performed faster. As mentioned earlier, IPFS uses cryptographic hash which is stored in the decentralized manner using peer-to-peer network.

This also ensures that while solving the scalability problem the security of the framework is not compromised.

1. Content-Addressable Storage

Content-addressable storage refers to the off-chain storage mechanism of IPFS used in the proposed framework [20]. The sensitive record of patient is stored on the IPFS, which ensures that a hash of the stored record is generated. That hash is now stored in the blockchain and is accessed when needed by the doctors and patients. The IPFS generates the cryptographically secure hash which ensures the security of the data being stored on it. And this also ensures security in our proposed framework.

1. Integrity

Integrity of a system is measured by the trustfulness of that system and also that system storing that information is temper-proof and reliable. This blockchain- based system ensures that it does not compromise this feature. The information stored in this system is intact and is not changed by any unauthorized channel. Moreover, information is available to only the associated parties that are doctors and patients. The users of the system and any third party do not have the right to make any changes in the smart contract as they are not having any access to it.

This is done by using the access rules which ensure that the private data or medical records of patients are not accessible and remain temper-proof.

Moreover, using IPFS for storage of records also ensures the security of the medical records of the patients.

1. Access Control

Using the Role-based access mechanism, this framework makes sure that every entity of the system is assigned a role. Any third party who is not authorized to have access to the system would not be able to access the system. This system provides a two core security as firstly blockchain technology in itself is secure and uses certain protocols and mechanism to keep itself secure from third-part intrusions. And secondly our system uses the Role-based access that also only allows the users having defined roles to have access to the system and its functions. So, our system would not only ensure security of patient records but would also make sure the access control of entities associated with it. This parameter also ensures that the security of the patient’s personal medical data is not compromised and the access is provided to only the authorized users of the system.

1. Information Confidentiality

The patient medical records stored on the blockchain should be secured from any third party access to ensure the confidentiality of the patients’ record. The

patient’s data include the important information of patient such as the patient medical history, blood group, records, lab results, X-rays reports, MRI results and many other related results and reports. All of this information is critical not only to patients but also to the hospital. Smart contracts are a really helpful element in this system as they ensure transparency, precision and trust on the transactions being performed. The record being stored and accessed in the system are only accessible by the trusted parties. Any untrusted third party trying to access the system is denied access by the system. With the information being kept as confidential from third party access the framework would ensure that it would the aspect of privacy as well.

# ADVANTAGES & DISADVANTAGES

**Advantages**

* Enhanced Security:
  + Blockchain technology provides a high level of security, making it difficult for unauthorized individuals to manipulate voting data.
  + The immutability of the blockchain ensures that once a vote is recorded, it cannot be altered, enhancing the integrity of the electoral process.

Transparency and Trust:

Advantage: Blockchain's decentralized ledger ensures transparency in the voting process, fostering trust among voters, election authorities, and other stakeholders.

Data Integrity:

Advantage: Blockchain ensures the integrity of the voting data, protecting against data manipulation and safeguarding the sanctity of the electoral process.

Accessibility:

Advantage: Electronic voting systems with blockchain can be designed to be more accessible, accommodating individuals with disabilities and facilitating remote voting.

Efficiency and Accuracy:

Advantage: Smart contracts on the blockchain streamline various aspects of the voting process, reducing human errors and accelerating vote counting.

Reduced Costs:

Advantage: Over time, electronic voting systems can lead to cost savings by reducing the need for physical polling stations and paper-based processes.

Civic Engagement:

Advantage: Electronic voting systems encourage greater participation in the democratic process, particularly among younger generations and those facing voting barriers.

Auditability and Accountability:

Advantage: Blockchain's auditability features allow discrepancies to be traced back to their source, promoting accountability among election officials and adherence to regulations.

Resilience to DDoS Attacks:

Advantage: Blockchain-based systems can be designed to be more resilient to Distributed Denial of Service (DDoS) attacks, improving system security.

Global and Remote Voting:

Advantage: Blockchain-based voting systems can enable citizens living abroad to participate in their home country's elections, increasing representation and facilitating remote voting.

# Disadvantages

1. Implementation Challenges: Integrating blockchain into existing EHR systems can be complex and costly. Healthcare organizations may face resistance to change and may struggle with the technical aspects of implementation.
2. Scalability: Managing large volumes of healthcare data on a blockchain network can be challenging. Ensuring scalability while maintaining data security is a significant concern.
3. Regulatory Compliance: Healthcare is heavily regulated, and ensuring that blockchain-based EHRs comply with all relevant laws and standards can be a complex and ongoing process.
4. Initial Investment: Implementing blockchain technology requires a significant initial investment in infrastructure, training, and development. Smaller healthcare organizations may find this cost-prohibitive.
5. User Adoption: Healthcare professionals may need time to adapt to new EHR systems based on blockchain technology. Training and change management efforts are required for successful user adoption.
6. Technological Maturity: Blockchain technology is evolving, and the healthcare industry may need to wait for it to mature further before achieving its full potential.
7. Data Recovery: In cases of lost private keys or other technical issues, recovering patient data from a blockchain can be challenging, if not impossible.
8. Energy Consumption: Some blockchain implementations, especially proof- of-work systems, can have high energy consumption, which may not align with environmental sustainability goals.

# CONCLUSION

In conclusion, the integration of blockchain technology into electronic voting systems holds great promise for the future of democratic processes. By enhancing security, transparency, and efficiency, these systems offer a robust solution to the challenges associated with traditional voting methods. The immutability of blockchain ensures data integrity and minimizes the risk of electoral fraud, while its transparency builds trust among voters and stakeholders.

Moreover, the automation of election processes through smart contracts streamlines the voting process and increases accessibility. However, it is crucial to acknowledge and address potential challenges such as scalability, privacy, and security. Nevertheless, through rigorous testing, pilot projects, and collaboration with experts and government authorities, electronic voting systems with blockchain technology have the potential to revolutionize the electoral landscape, making elections more secure, transparent, and inclusive, in alignment with the fundamental principles of democracy.

# FUTURE SCOPE

The future scope for electronic voting systems with blockchain technology is promising, as it offers innovative solutions to long-standing challenges in the electoral process. As technology continues to evolve and as societies place an increased emphasis on secure and efficient voting methods, the future of electronic voting systems with blockchain technology holds several key opportunities and potential advancements:

* Wider Adoption: The adoption of electronic voting systems with blockchain technology is likely to expand globally. More countries and regions may consider implementing such systems to improve the integrity and efficiency of their elections, leading to greater acceptance and trust in electronic voting.
* Improved Usability: Future systems will focus on enhancing user interfaces to ensure ease of use and accessibility for all voters, including those with disabilities. This may involve the development of user-friendly mobile applications and other innovative interfaces.
* Enhanced Security Features: Ongoing research and development will result in even more robust security measures, making it exceedingly difficult for malicious actors to compromise the voting process. Innovations in encryption and identity verification will be key areas of focus.
* Privacy Enhancements: Future systems may offer advanced privacy features, allowing voters to verify their choices while maintaining the secrecy of their votes. Zero-knowledge proofs and other cryptographic techniques may be integrated to address privacy concerns.
* Scalability: Scalability remains a challenge, especially during high-traffic elections. Future systems will focus on addressing this issue to ensure that they can accommodate a large number of voters without compromising performance.
* Interoperability: Standards and protocols for interoperability between different electronic voting systems may emerge, allowing for greater consistency and cross-compatibility between systems, especially in international elections or for citizens living abroad.
* Global and Remote Voting: Future systems may enable citizens living abroad to vote in their home country's elections more seamlessly. Remote voting options may become standard for all voters, making it easier for individuals facing mobility issues or other barriers to participate.
* Hybrid Voting Models: Hybrid models that combine electronic and traditional paper voting methods may gain popularity. These systems would allow voters to choose the method they are most comfortable with, promoting inclusivity and addressing concerns about technology literacy.
* Blockchain Innovations: Ongoing developments in blockchain technology, such as improvements in consensus algorithms and network performance, will further enhance the security and efficiency of electronic voting systems.
* Research and Regulation: Future scope includes more extensive research into electronic voting systems with blockchain, along with the development of regulatory frameworks and best practices to ensure the integrity of these systems and protect against potential risks.
* International Collaboration: Countries may collaborate on developing international standards for electronic voting with blockchain technology to facilitate cross-border elections and build trust in global democratic processes.
* Voter Engagement and Education: Systems may include features that enhance voter engagement, such as providing information about candidates and issues, facilitating voter education, and encouraging greater voter turnout.

# APPENDIX

**Source Code:**

import React, { useState } from "react";

import { votingContract } from "../utils/constants";

function Voting() {

    const [CandidateName, setCandidateName] = useState("");

    const [CandidateAge, setCandidateAge] = useState("");

    const [CandidateID, setCandidateID] = useState("");

    const [VoterID, setVoterID] = useState("");

    const [VoterName, setVoterName] = useState("");

    const [VoterAge, setVoterAge] = useState("");

    const [VoterVoteID, setVoterVoteID] = useState("");

    const [PartyID, setPartyID] = useState("");

    const [VoteCount1, setVoteCount1] = useState("");

    const [VoteCount2, setVoteCount2] = useState("");

    const [VoteCount3, setVoteCount3] = useState("");

    const [HighestCount, setHighestCount] = useState("");

    const handleCandidatename = (e) => {

        setCandidateName(e.target.value);

    };

    const handleCandidateAge = (e) => {

        const value = e.target.value.replace(/\D/g, "");

        setCandidateAge(Number(value));

    };

    const handleCandidateID = async (e) => {

        const value = e.target.value.replace(/\D/g, "");

        setCandidateID(Number(value));

    };

    const handleCandidateRegistration = async (e) => {

        e.preventDefault();

        const enrollCanddidateTx = await votingContract.enrollCandidate(CandidateID, CandidateName, CandidateAge);

        await enrollCanddidateTx.wait();

        console.log(enrollCanddidateTx);

        alert(enrollCanddidateTx.hash);

    };

    const handleVoterID = async (e) => {

        const value = e.target.value.replace(/\D/g, "");

        setVoterID(Number(value));

    };

    const handleVoterName = (e) => {

        setVoterName(e.target.value);

    };

    const handleVoterAge = async (e) => {

        const value = e.target.value.replace(/\D/g, "");

        setVoterAge(Number(value));

    };

    const handleVoterRegistration = async (e) => {

        e.preventDefault();

        const enrollVoterTx = await votingContract.enrollVoter(VoterID, VoterName, VoterAge);

        await enrollVoterTx.wait();

        console.log(enrollVoterTx);

        alert(enrollVoterTx.hash);

    };

    const handlePartyID = async (e) => {

        setPartyID(Number(e.target.value));

    };

    const handleVoterVoteID = async (e) => {

        const value = e.target.value.replace(/\D/g, "");

        setVoterVoteID(Number(value));

    };

    const handleVote = async (e) => {

        e.preventDefault();

        const voteTx = await votingContract.vote(PartyID, VoterVoteID);

        await voteTx.wait();

        console.log(voteTx);

        alert(voteTx.hash);

    };

    const handleQuery1 = async (e) => {

        let vote = Number(e.target.id);

        const voteCountTx = await votingContract.getVotecountOf(vote);

        setVoteCount1(voteCountTx.toString());

    };

    const handleQuery2 = async (e) => {

        let vote = Number(e.target.id);

        const voteCountTx = await votingContract.getVotecountOf(vote);

        setVoteCount2(voteCountTx.toString());

    };

    const handleQuery3 = async (e) => {

        let vote = Number(e.target.id);

        const voteCountTx = await votingContract.getVotecountOf(vote);

        setVoteCount3(voteCountTx.toString());

    };

    const handleResult = async () => {

        let number1 = await votingContract.getVotecountOf(1);

        let number2 = await votingContract.getVotecountOf(2);

        let number3 = await votingContract.getVotecountOf(3);

        let num1 = number1.toString();

        let num2 = number2.toString();

        let num3 = number3.toString();

        if (num1 > num2 && num1 > num3) {

            setHighestCount("BJP");

        } else if (num2 > num1 && num2 > num3) {

            setHighestCount("TRS");

        } else if (num3 > num1 && num3 > num2) {

            setHighestCount("Congress");

        } else {

            setHighestCount("");

        }

    };

    return (

        <div>

            <div className="flex flex-row space-x-52 mt-10 ml-96">

            <div>

                    <div className="mt-14 ">

                        <h3 className="text-2xl">Candidate Registration</h3>

                        <form onSubmit={handleCandidateRegistration}>

                            <div className="form-group mb-6">

                                <div className="mt-3"></div>

                                <div className="space-y-2">

                                    <div>

                                        <label>

                                            Candidate ID

                                            <select className="w-64 ml-2 rounded-full text-slate-900" value={CandidateID} onChange={handleCandidateID}>

                                                <option name="BJP">1</option>

                                                <option name="TRS">2</option>

                                                <option name="CONGRESS">3</option>

                                            </select>

                                        </label>

                                    </div>

                                    <div>

                                        <label>

                                            Candidate Name

                                            <span>

                                                <input className="ml-2 rounded-full text-slate-900" value={CandidateName} onChange={handleCandidatename} />

                                            </span>

                                        </label>

                                    </div>

                                    <div>

                                        <label>

                                            Candidate age

                                            <span>

                                                <input className="ml-2 rounded-full text-slate-900" value={CandidateAge} onChange={handleCandidateAge} />

                                            </span>

                                        </label>

                                    </div>

                                </div>

                                <input className="bg-blue-500 hover:bg-blue-900 text-white font-bold py-1 px-2 rounded-full mt-4" type="submit" value="Register" />

                            </div>

                        </form>

                    </div>

                    <div className="mt-14">

                        <h3 className="text-2xl">Voter Registration</h3>

                        <form onSubmit={handleVoterRegistration}>

                            <div>

                                <label>

                                    VotedID

                                    <span className="ml-2 mr-2 ">

                                        <input className="rounded-full text-slate-900" value={VoterID} onChange={handleVoterID} />

                                    </span>

                                </label>

                            </div>

                            <div className="mt-2">

                                <label>

                                    Voter Name

                                    <span className="ml-2 mr-2 ">

                                        <input className="rounded-full text-slate-900" value={VoterName} onChange={handleVoterName} />

                                    </span>

                                </label>

                            </div>

                            <div className="mt-2">

                                <label>

                                    Voter Age

                                    <span className="ml-2 mr-2 ">

                                        <input className="rounded-full text-slate-900" value={VoterAge} onChange={handleVoterAge} />

                                    </span>

                                </label>

                            </div>

                            <button className="bg-blue-500 hover:bg-blue-900 text-white font-bold py-1 px-2 rounded-full mt-2">Register</button>

                        </form>

                    </div>

                </div>

                <div>

                    <form onSubmit={handleVote}>

                        <p className="text-2xl">Vote</p>

                        <div>

                            <div>

                                <img

                                    className="max-w-sm max-h-40 full-auto"

                                    src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT1PeilVpA3QSlwxU5Z34Oc1Y-x\_Idy3bU8nLrhTLtUhQ&s"

                                    alt="BJP"

                                />

                            </div>

                            <div>

                                <input

                                    className="form-check-input appearance-none rounded-full h-4 w-4 border border-black border-x-2 border-y-2 bg-white checked:bg-blue-600 checked:border-black focus:outline-none transition duration-200 mt-1 align-top bg-no-repeat bg-center bg-contain  mr-2 cursor-pointer"

                                    type="radio"

                                    name="flexRadioDefault"

                                    value="1"

                                    onChange={handlePartyID}

                                />

                                <label className="form-check-label inline-block text-gray-800" htmlFor="flexRadioDefault1">

                                    BJP ID - 1

                                </label>

                            </div>

                        </div>

                        <div>

                            <div>

                                <img

                                    className="w-56 max-h-40 full-auto"

                                    src="https://4.bp.blogspot.com/-usfE8G6o\_wY/WsjahdsdZGI/AAAAAAAAWBI/dexdE3O-Jt0XN0v2GvdWswfDww8HuVbAQCLcBGAs/s1600/trs03%2Bcopy.jpg"

                                    alt="TRS"

                                />

                            </div>

                            <div>

                                <input

                                    className="form-check-input appearance-none rounded-full h-4 w-4 border border-black border-x-2 border-y-2 bg-white checked:bg-blue-600 checked:border-black focus:outline-none transition duration-200 mt-1 align-top bg-no-repeat bg-center bg-contain  mr-2 cursor-pointer"

                                    type="radio"

                                    name="flexRadioDefault"

                                    value="2"

                                    onChange={handlePartyID}

                                />

                                <label className="form-check-label inline-block text-gray-800" htmlFor="flexRadioDefault1">

                                    TRS ID - 2

                                </label>

                            </div>

                        </div>

                        <div>

                            <div>

                                <img

                                    className="max-w-sm max-h-40 full-auto"

                                    src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRTLtqW8T4TEv-Ni-3NbGT28sFKUQOkoUEnOCryW3ZGuA&s"

                                    alt="CONGRESS"

                                />

                            </div>

                            <div>

                                <input

                                    className="form-check-input appearance-none rounded-full h-4 w-4 border border-black border-x-2 border-y-2 bg-white checked:bg-blue-600 checked:border-black focus:outline-none transition duration-200 mt-1 align-top bg-no-repeat bg-center bg-contain  mr-2 cursor-pointer"

                                    type="radio"

                                    name="flexRadioDefault"

                                    value="3"

                                    onChange={handlePartyID}

                                />

                                <label className="form-check-label inline-block text-gray-800" htmlFor="flexRadioDefault1">

                                    CONGRESS ID - 3

                                </label>

                            </div>

                        </div>

                        <div className="mt-5">

                            <label>

                                VotedID

                                <span className="ml-2 mr-2 ">

                                    <input className="rounded-full w-20 text-slate-900" value={VoterVoteID} onChange={handleVoterVoteID} />

                                </span>

                            </label>

                        </div>

                        <input className="bg-red-500 hover:bg-blue-900 text-white font-bold py-3 px-16 rounded-full mt-4" type="submit" value="Vote" />

                    </form>

                </div>

                {/\* =>>>>>>>............................................................................. \*/}

                <div>

                    <div className="">

                        <p className="text-2xl">Result</p>

                        <div className="mt-12"></div>

                        <button onClick={handleQuery1} id="1" className="bg-blue-500 hover:bg-blue-900 text-white font-bold py-1 px-2 rounded-full mt-2">

                            Query

                        </button>

                        <p>{VoteCount1}</p>

                    </div>

                    <div className="mt-20">

                        <div></div>

                        <button onClick={handleQuery2} id="2" className="bg-blue-500 hover:bg-blue-900 text-white font-bold py-1 px-2 rounded-full mt-2">

                            Query

                        </button>

                        <p>{VoteCount2}</p>

                    </div>

                    <div className="mt-20">

                        <button onClick={handleQuery3} id="3" className="bg-blue-500 hover:bg-blue-900 text-white font-bold py-1 px-2 rounded-full mt-2">

                            Query

                        </button>

                        <p>{VoteCount3}</p>

                    </div>

                    <div className="mt-28">

                        <button onClick={handleResult} className="bg-red-500 hover:bg-blue-900 text-white font-bold py-3 px-5 rounded-full ">

                            Winner

                        </button>

                        <p className="text-3xl">{HighestCount}</p>

                    </div>

                </div>

                {/\* =>>>>>>>........................................................................ \*/}

            </div>

        </div>

    );

}

export default Voting;

# GitHub & Project Demo Link

<https://github.com/MohamedTawfeeq25/naanmudhalvan-project>

[https://drive.google.com/drive/folders/1lKXMonG4mtiK7Wmbo1WfKM0uvrn2tbac?usp=sharing](https://drive.google.com/drive/folders/1lKXMonG4mtiK7Wmbo1WfKM0uvrn2tbac?usp=sharing%0c)