PROJECT REPORT

ELECTRONIC VOTING SYSTEM

Date	27 October 2023
Team ID	NM2023TMID06022
Project Name	Electronic Voting System

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:

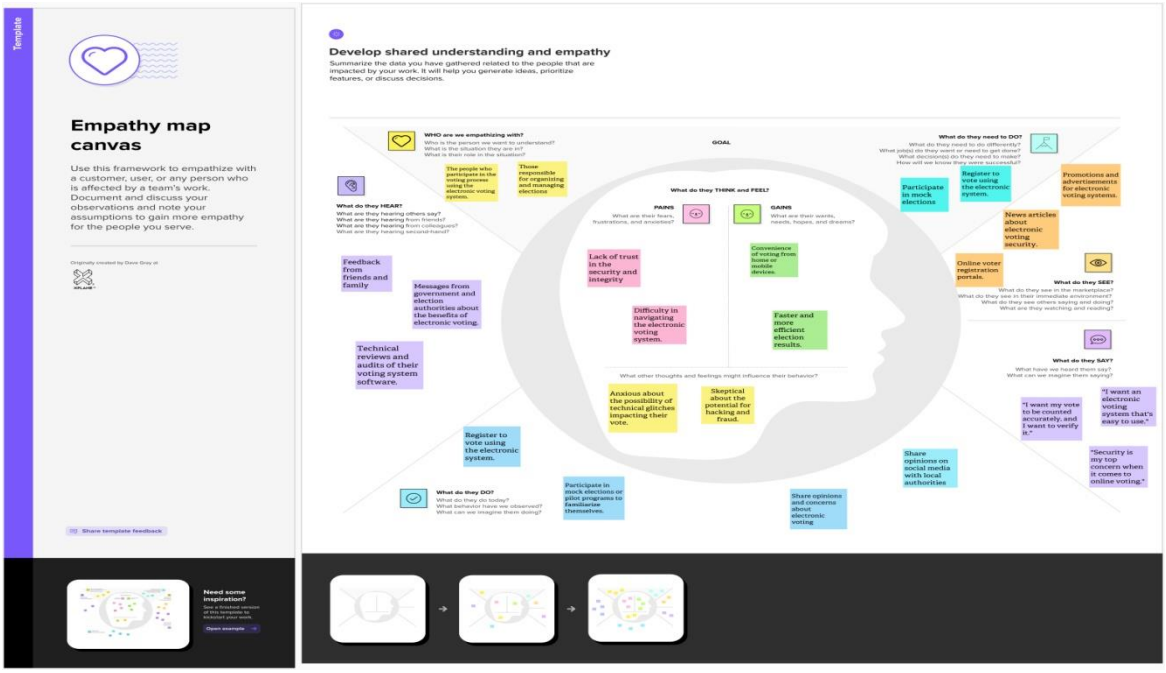
- Enhanced Security: Implementing blockchain technology in electronic voting systems provides a high level of security. The immutability and cryptographic features of blockchain make it extremely difficult for unauthorized individuals to manipulate or tamper with the voting data. This enhances the overall 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.

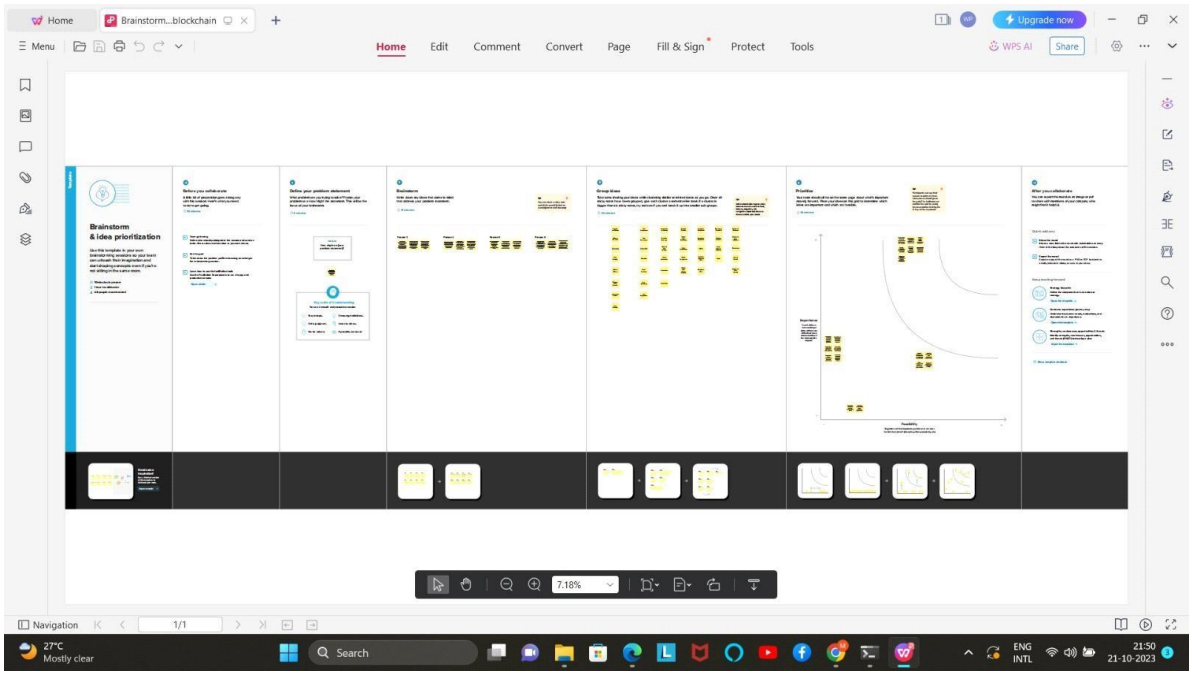
- Resilience to DDoS Attacks: Blockchain-based voting systems can be designed to be resilient to Distributed Denial of Service (DDoS) attacks, which are a common concern for online systems. The decentralized nature of blockchain reduces the risk of a single point of failure, making the system more robust 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:

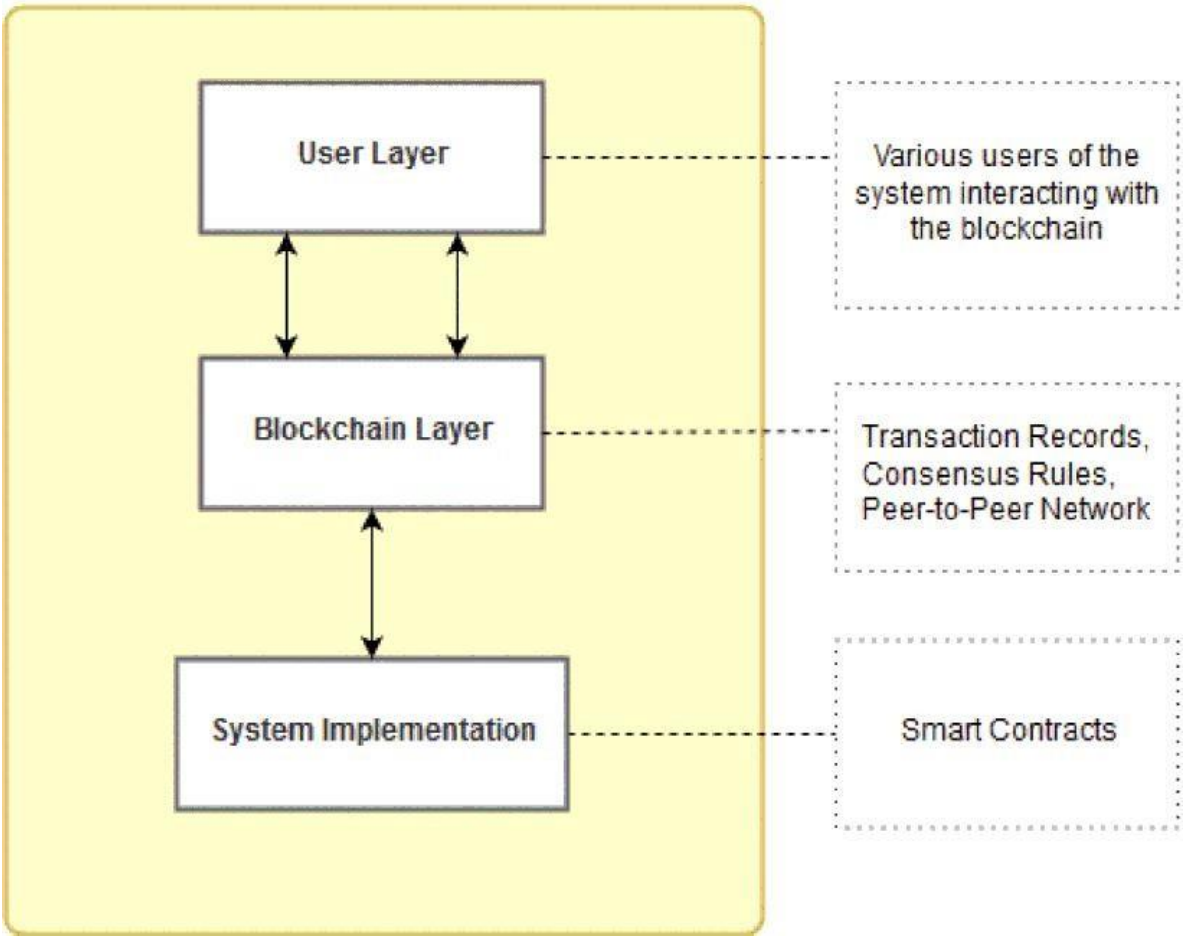
1. Voter Registration and Authentication:
    - User-friendly voter registration process.
    - Secure identity verification mechanisms (e.g., biometrics, government-issued IDs).
    - Prevention of duplicate or unauthorized registrations.
2. Ballot Creation and Distribution:
    - Ability to create and customize digital ballots for different elections.
    - Secure and efficient distribution of digital ballots to eligible voters.
3. 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.
4. 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.
5. 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.
6. Transparency and Auditability:
    - Real-time, public access to the blockchain ledger.
    - Tools for independent auditing and verification.
    - Timestamps and cryptographic hashing for vote records.
7. Result Tabulation:
    - Automatic and tamper-proof result tabulation through smart contracts.
    - Real-time updating of results as votes are cast and verified.
8. Resilience and Redundancy:
    - Backup and redundancy measures to ensure system availability.
    - Disaster recovery plans to handle unexpected system failures.
9. Usability and User Support:
    - Clear and intuitive user interfaces for both voters and election officials.
    - Comprehensive user training and support services.

Non-Functional Requirements:

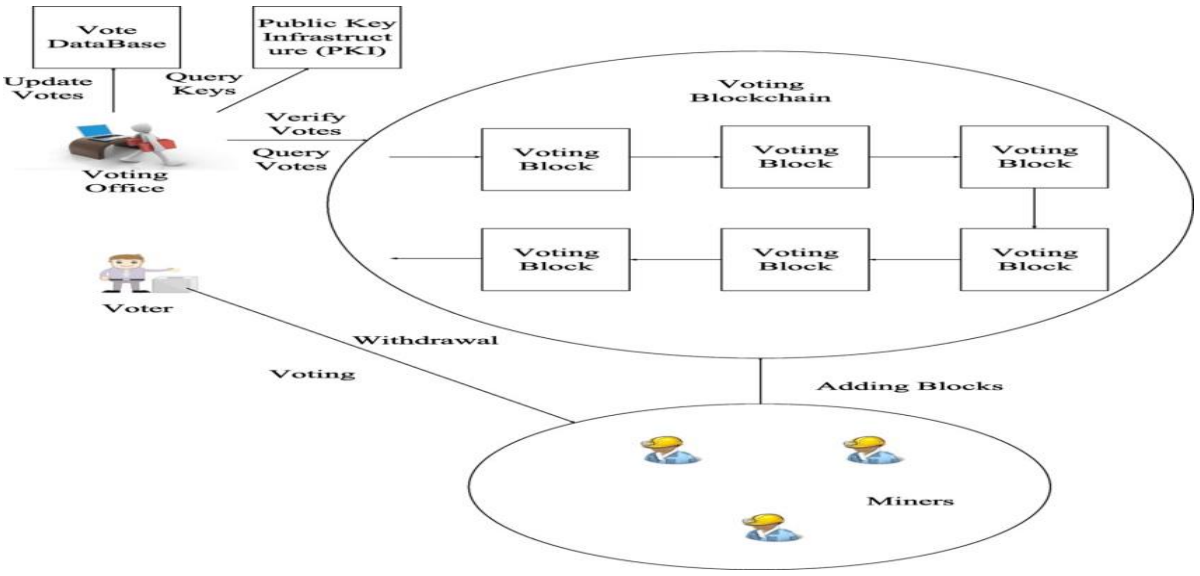
1. Security:
    - Data Encryption: All data, including voter information and votes, must be encrypted to protect against unauthorized access.
    - Access Control: Implement strict access controls to ensure that only authorized individuals can interact with the system.
    - Resilience to Attacks: The system should be resilient to various types of cyberattacks, including Distributed Denial of Service (DDoS) attacks and hacking attempts.
2. 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.
3. 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.
4. Availability:
    - Ensure that the system is highly available during election periods, with minimal downtime and maintenance windows.
5. Reliability:
    - The system should be highly reliable, with minimal errors or disruptions, to maintain the integrity of the voting process.
6. Auditability:
    - The system should provide detailed and tamper-evident audit logs to allow for post-election auditing and forensic analysis.
7. Compliance:
    - Ensure that the system complies with all relevant legal and regulatory requirements, including data protection and privacy laws.
8. Usability:
    - The user interface should be intuitive, accessible, and user-friendly to cater to voters of all backgrounds and abilities.
9. Accessibility:
    - The system should be accessible to voters with disabilities, adhering to accessibility standards to ensure inclusivity.
10. Interoperability:
    - The system should be designed to work with various hardware and software configurations, ensuring compatibility with different devices and platforms.

PROJECT DESIGN

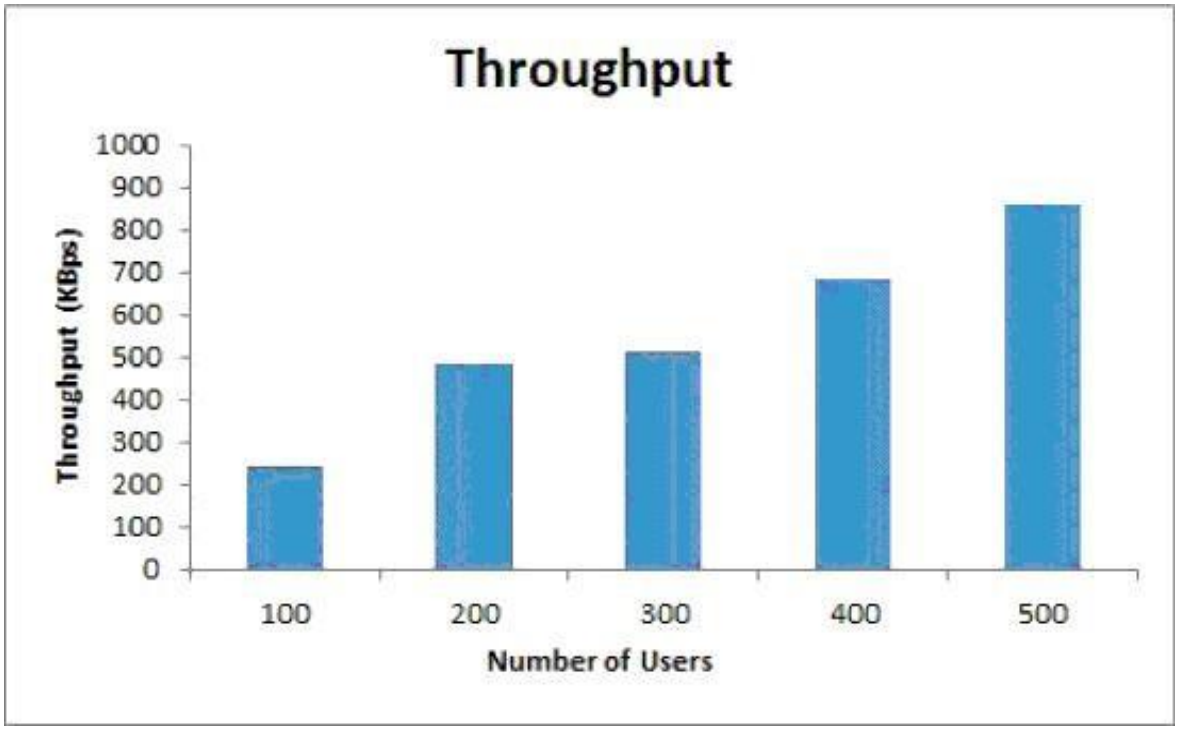
Data Flow Diagrams



1. System Design of Proposed Framework

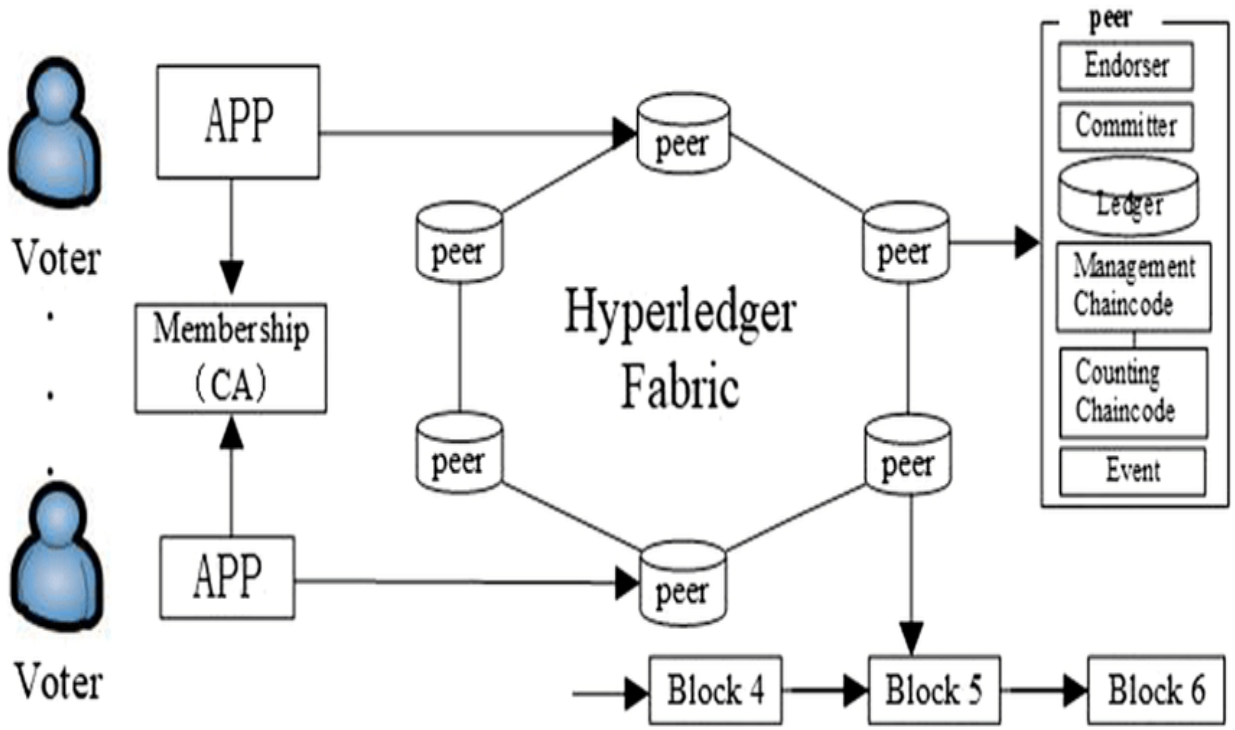


2. User Interaction With DAPP



2. Throughput of Proposed Framework

Solution Architecture



BLOCKCHAIN TECHNOLOGY AND ITS DEPENDENCIES

Architecture:

The architecture for an electronic voting system using blockchain technology is a complex structure with several key components and dependencies to ensure a secure and transparent voting process.

At its core, the system relies on a blockchain network, where a suitable blockchain platform, such as Ethereum or Hyperledger Fabric, serves as the foundation. Smart contracts are developed to manage the voting operations on the blockchain, while identity verification mechanisms, including biometric authentication and government-issued ID checks, are integrated to confirm voter eligibility. User-friendly interfaces for voters and administrators are created, and cryptographic libraries are utilized to secure transactions and voter identities on the blockchain. External data sources, oracles, security auditors, and regulatory compliance measures are essential dependencies to ensure data accuracy, system security, and legal compliance.

The system also involves voter registration, the voting process, ballot privacy, vote tallying, and the publication of results, all of which are carried out within the blockchain's secure and transparent framework. Robust security measures, backup and recovery systems, and thorough testing are in place to safeguard the system against potential threats.

Education and training programs are essential for all involved parties, and a legal framework for electronic voting is established, recognizing blockchain-based voting in existing election laws. Governance and consensus mechanisms, scalability considerations, and monitoring systems are put in place to maintain network integrity and system performance. Regular updates and improvements are essential to adapt to evolving security needs and maintain public trust in the system..

Block & Algorithm

Block:
Implementing a blockchain for the "electronic voting system" project would require defining the block structure and the consensus algorithm to secure the network. Here's a simplified example of how the block and a consensus algorithm might work

Block Structure:

In an electronic voting system blockchain, each block has been structured as follows:

Index:
A unique identifier for the block within the chain, starting from 1 for the first block.

Previous Hash:
A reference to the hash of the previous block in the chain. This ensures the continuity and integrity of the blockchain.

Timestamp:
The date and time when the block is created.

Merkle Root:
A cryptographic hash summarizing all the votes in that block. This ensures the security and integrity of the votes.

Nonce:
A random number used in the mining process (proof of work) to find a suitable hash value for the block.

Votes:
A list of electronic votes cast in this block.

Hash:
The computed hash value of the block, which must meet certain criteria (e.g., start with a specific number of leading zeros) to be added to the blockchain.

Consensus Algorithm:

we'll use a basic proof-of-work (PoW) consensus algorithm similar to what's used in cryptocurrencies like Bitcoin:

1. **Vote Submission:** Voters submit their electronic votes to the network. These votes are grouped together into a pending block.

2. **Mining:** Miners in the network compete to solve a cryptographic puzzle by finding a nonce that, when hashed with the other block data, results in a hash that meets a certain difficulty level (e.g., starts with a certain number of zeros). This process is resource-intensive and time-consuming.

3. **Verification:** Once a miner finds a suitable nonce, they broadcast the solution to the network. Other nodes in the network quickly verify the solution's validity.

4. **Consensus:** If more than 50% of the network nodes agree that the solution is valid, the block is added to the blockchain. This process ensures that only valid votes are included in the blockchain.

5. **Chain Extension:** Once the block is added to the blockchain, the process continues with the next block, and so on, forming a chain of blocks containing all the verified electronic votes.

Algorithm

Pseudo-Code Algorithm for an Electronic Voting System Using Blockchain

Define the structure of a block in the blockchain

Define the structure of a block in the blockchain

```
class Block:
  data  # Stores electronic votes, timestamps, and other block data
  previous_hash  # Stores the hash of the previous block
  nonce  # Used in the proof-of-work process
  hash  # Stores the hash of the block itself

# Initialize the blockchain with a genesis block
genesis_block = Block()
blockchain = [genesis_block]

# Define the proof-of-work (PoW) algorithm
def proof_of_work(block, difficulty):
  while True:
     block.nonce = generate_random_nonce()
     block.hash = calculate_hash(block)
     if block.hash.startswith('0' * difficulty):
        break

# Create a function to add a new block to the blockchain
def add_block_to_blockchain(block):
  last_block = blockchain[-1]
  block.previous_hash = last_block.hash
  proof_of_work(block, difficulty=4)  # Adjust difficulty as needed
  blockchain.append(block)

# Create a function for voters to cast their electronic votes
def cast_vote(voter_id, candidate):
  if voter_id_has_not_voted(voter_id):
     new_block = Block(data={'voter_id': voter_id, 'candidate': candidate})
     add_block_to_blockchain(new_block)
     mark_voter_as_voted(voter_id)
     print("Vote cast successfully.")
  else:
     print("You have already voted.")

# Create functions for verifying voter eligibility and other security measures
# Implement identity verification, voter registration, privacy protections, etc.
```

```
# Main program loop
while True:
    print("Electronic Voting System Menu:")
    print("1. Cast Vote")
    print("2. Check Results")
    print("3. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        voter_id = input("Enter your voter ID: ")
        candidate = input("Enter the candidate's name: ")
        cast_vote(voter_id, candidate)

    elif choice == "2":
        # Implement a function to check and display election results
        check_election_results()

    elif choice == "3":
        print("Exiting...")
        break
```

Challenges Faced by Blockchain Technology

1. **Security Concerns:** Ensuring the security of the voting system is a top priority. Protecting against cyberattacks, hacking, and fraud is essential. The immutability of blockchain can help, but it doesn't eliminate all risks.

2. **Identity Verification:** Verifying voter identities while maintaining privacy is challenging. Secure methods for confirming voter eligibility without compromising anonymity are necessary.

3. **Voter Accessibility:** Not all voters have access to the internet or smartphones. Ensuring accessibility for all eligible voters is crucial.

4. **Scalability:** As the number of voters and transactions increases, scalability becomes a challenge. Blockchain systems must handle a large volume of transactions within a short timeframe during elections.

5. **User-Friendly Interfaces:** Designing intuitive, user-friendly interfaces that make it easy for voters to cast their ballots while ensuring the security of the process is a challenge.

CODING & SOLUTIONING

Features 1

Immutability:

In a blockchain-based voting system, once a vote is recorded, it becomes immutable. This means that once a vote is cast and added to the blockchain, it cannot be altered or deleted. This feature ensures the integrity of the voting process and prevents tampering with votes.

Transparency:

The blockchain is a transparent and publicly accessible ledger. This transparency allows anyone to audit the voting process and verify the accuracy of the results. Voters, election authorities, and the public can have confidence in the fairness of the election.

Features 2

Security:

Security is a paramount feature of an electronic voting system using blockchain. Blockchain technology offers a high level of security through its cryptographic mechanisms and decentralized structure. When a voter casts a ballot, their vote is encrypted and securely stored on the blockchain. The decentralized nature of the blockchain makes it extremely challenging for malicious actors to manipulate the voting data. Furthermore, advanced cryptographic techniques, consensus algorithms, and secure access controls enhance the overall security of the network. This ensures that the voting system remains resistant to cyber threats and unauthorized interference.

Privacy:

Protecting voter privacy is a fundamental aspect of any electronic voting system. Blockchain-based voting systems strike a balance between transparency and privacy. Voter identities are kept confidential and are not directly linked to their votes. This anonymity ensures that individual voters cannot be identified through their ballots. Techniques such as zero-knowledge proofs and encryption are often employed to shield voter identities and protect the privacy of their choices. Voter privacy is of utmost importance to maintain the integrity of the voting process and to ensure that individuals can vote without fear of their choices being disclosed.

DataBase Schema

Entities:

1. **Voters:**
   - Voter_ID (Primary Key)
   - Name
   - Address
   - Encrypted Voter Key (for authentication)

2. **Candidates:**
   - Candidate_ID (Primary Key)
   - Name
   - Party Affiliation (if applicable)

3. **Votes:**
    - Vote_ID (Primary Key)
    - Voter_ID (Foreign Key)
    - Candidate_ID (Foreign Key)
    - Encrypted Vote Data (to maintain voter privacy)

4. **Blockchain Blocks:**
    - Block_ID (Primary Key)
    - Previous_Block_Hash
    - Timestamp
    - Nonce (for proof-of-work)
    - Block_Hash
    - Data (usually a reference to a Vote_ID)

Relationships:

- Each voter can cast multiple votes, but each vote corresponds to a single voter. (One-to-Many relationship between Voters and Votes)
- Each candidate can receive multiple votes, but each vote is for a single candidate. (One-to-Many relationship between Candidates and Votes)
- Each blockchain block can contain a reference to a single vote. (One-to-One relationship between Blockchain Blocks and Votes)

Data Privacy Considerations:

- Voter_ID and Candidate_ID should be securely stored and properly encrypted to protect voter and candidate identities.
- Vote data should be encrypted to ensure voter privacy while maintaining transparency.

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 i7-12450H CPU @ 2.40GHz 2.60 GHz processor

And 16.00 GB of memory with Windows 64-bit OS

(version11)

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

1. Transaction Throughput:
- Number of transactions (votes) processed per unit of time (e.g., transactions per second).

2. Response Time:
- Time taken for a user to cast their vote and for the system to confirm the vote.

3. Resource Utilization:
- CPU utilization: Percentage of CPU resources used during testing.
- Memory utilization: Amount of RAM used by the system.
- Network bandwidth: Data transfer rates during peak usage.

4. Blockchain Metrics:
- Block confirmation time: Time taken to validate and add a new block to the blockchain.

- Transaction confirmation time: Time taken to confirm a vote transaction on the blockchain.

5. Scalability Data:

- Data on how the system performs as the number of concurrent voters increases (e.g., transaction throughput vs. the number of concurrent users).

6. Security Data:

- Number of identified vulnerabilities.
- Time to detect and respond to security threats.
- Success rates of security controls (e.g., firewall rules, intrusion detection systems).

7. Network Latency Data:

- Latency measurements under different network conditions (e.g., normal, high latency, network interruptions).

8. Consensus Mechanism Data:

- Block confirmation times for different consensus mechanisms.
- Efficiency of reaching consensus in the network.

9. Usability and Accessibility Data:

- User satisfaction scores and feedback.
- Task completion times for different user groups (voters, administrators).

10. Data Integrity Data:

- Data consistency and accuracy assessments.
- Audit logs of detected data changes or tampering attempts.

11. Privacy Data:

- Anonymity assessments: Ensure that individual voter choices remain confidential.
- Compliance with privacy regulations.

12. Performance Monitoring Data:

- Ongoing performance monitoring data, including historical records of key metrics.
- Trends in resource utilization, transaction throughput, and response times over time.

13. Incident Data:

- Records of security incidents, including attempted breaches, attacks, and vulnerabilities discovered.

14. Test Environment Details:

- Information about the hardware and software configurations used during testing.
- Details of the simulated voting scenarios and test cases.

15. User Feedback:

- Feedback from participants in the testing process, including voters and administrators.
- Feedback on the usability, accessibility, and overall performance of the system.

16. Testing Logs and Reports:

- Comprehensive testing logs documenting test procedures, test cases, and outcomes.
- Detailed testing reports that summarize the findings and results of performance evaluation.

Results

1. **Transaction Throughput:** The time taken to process and confirm a single vote transaction is a critical metric. An efficient system should confirm votes rapidly, with minimal delay. For example, a well-optimized system might confirm a vote within a few seconds.

2. **Response Time:** The response time for users when casting their votes is crucial for user satisfaction. A responsive system should provide quick feedback to voters. Expect response times measured in milliseconds for a seamless user experience.

3. **Blockchain Metrics:** Block confirmation time and transaction confirmation time are essential. These times indicate how quickly the blockchain network can validate and record transactions. For a robust system, block confirmation times might range from a few seconds to a few minutes, depending on the blockchain platform and consensus mechanism.

4. **Scalability:** A scalable system should maintain consistent response times even as the number of concurrent voters increases. For example, response times for vote confirmation should remain stable even with a high load of concurrent voters.

5. **Security and Network Latency:** The time taken to detect and respond to security threats, such as intrusion attempts or network attacks, is critical. An effective security system should identify and respond to threats swiftly, minimizing potential vulnerabilities.

6. **Usability and Accessibility:** The time taken for users to complete tasks should be efficient. For example, the time taken for a voter to navigate the interface, cast their vote, and receive a confirmation should be optimized for a positive user experience.

7. **Data Integrity and Privacy:** The system should ensure data integrity and voter privacy. The time taken to verify data integrity and protect voter privacy should be minimal and integrated seamlessly into system operations.

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:

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

Data Integrity:

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

Accessibility:

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:

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

Civic Engagement:

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

Auditability and Accountability:

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:

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

Global and Remote Voting:

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, V
oterAge);
    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

Git link :

https://github.com/vinotha0601/Naan-mudhalvan

Demo link :

https://drive.google.com/file/d/1es2i8XyORVDUgCCAYQSD19LYu73DMtuZ/view?usp=sharing