NANN MUDHALVAN
PROJECT REPORT

ELECTRONIC VOTING SYSTEM

TEAM ID-NM2023TMID11945

TEAM LEADER

Gurubhavani-A -812020106009

TEAM MEMBER

Nisanthan.k- 812020106025
Nieeasarif.Y- 812020106024
Mohammed shakeel-812020106304

1. INTRODUCTION

1.1.Project Overview

The blockchain-based electronic voting system leverages the decentralized and tamper-resistant nature of blockchain technology to ensure secure and transparent elections. Each vote is recorded as a unique transaction on the blockchain, preventing manipulation or fraud. Smart contracts execute predefined voting rules, enhancing accuracy and reducing the need for intermediaries. Immutable ledger entries protect against data tampering, fostering trust in the electoral process. This innovative system enhances transparency, integrity, and the overall security of elections, paving the way for a more resilient and reliable democratic voting experience.

1.2.Purpose

TheThe purpose of a blockchain-based electronic voting system lies in enhancing the integrity and security of electoral processes. Utilizing blockchain ensures transparency, immutability, and decentralization, mitigating risks of fraud and manipulation. Each vote is recorded as a tamper-resistant transaction, creating an unalterable and publicly accessible ledger. This system enhances trust, reduces the potential for hacking, and provides a verifiable trail of votes. Additionally, it enables secure remote voting, improving accessibility. By leveraging blockchain technology, electronic voting systems aim to revolutionize democratic practices, fostering a more reliable and inclusive electoral environment.

2. LITERATURE

SURVEY2.1Existing

problem

While blockchain-based electronic voting systems offer potential benefits like transparency and tamper resistance, they also face challenges. Some existing problems include:

1. **Voter Privacy:** Blockchain transactions are transparent, which raises concerns about voter privacy. While the blockchain itself may not reveal individual votes, the linkage between a voter's identity and their blockchain address could compromise anonymity.

2. **Vulnerability to Cyber Attacks:** Despite blockchain's security features, electronic systems are still susceptible to cyber attacks. If attackers gain control over a significant number of nodes or the network as a whole, they could manipulate the voting process.

3. User Authentication: Ensuring secure and user-friendly methods for voter authentication is crucial. If private keys or authentication methods are compromised, it could lead to unauthorized access and manipulation of votes.

4. Usability Concerns: Blockchain technology can be complex, and ensuring a user-friendly experience for voters, especially those unfamiliar with blockchain, is a challenge. Complex systems may discourage participation and introduce the risk of user errors.

2.2References

1. Implementing an electronic voting system using blockchain technology can offer transparency and security. Some references and key papers on this topic include:

2. "A Blockchain-Based Approach Towards Overcoming Security Issues in E-Voting Systems" by S. Tople et al. (2018) - Discusses how blockchain can enhance the security of electronic voting systems.

3. "Securing Electronic Voting Systems with Blockchain" by N. Atzei et al. (2017) - Explores the application of blockchain for secure electronic voting, highlighting its potential advantages.

4. "Democracy Earth: Liquid Democracy on the Blockchain" by S. Benartzi (2015) - Focuses on the use of blockchain for liquid democracy, a form of governance that can be applied to electronic voting.

5. "A Survey of Blockchain Voting Systems" by M. Volkamer et al. (2020) - Provides an overview of different blockchain-based voting systems, their challenges, and potential solutions.

6. "E-Voting on Blockchain: An Initial Evaluation" by L. Buchman et al. (2018) - Examines the feasibility and challenges of implementing e-voting on a blockchain.

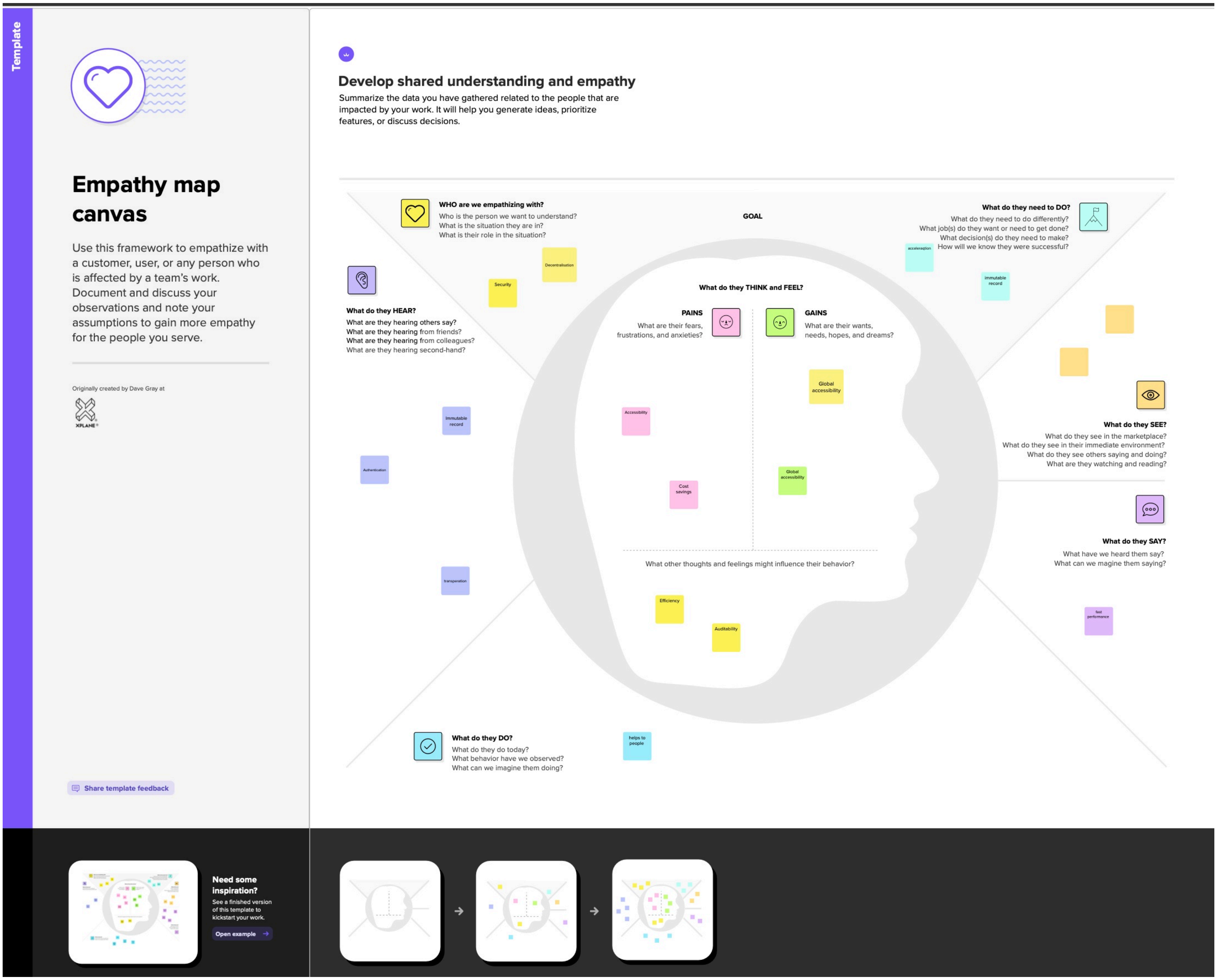
7. These references should give you a solid starting point to explore the integration of blockchain in electronic voting systems.eepak Panpatte, "Artificial Intelligence in Agriculture: An Emerging Era of Research", 2018.

2. Problem Statement Definition

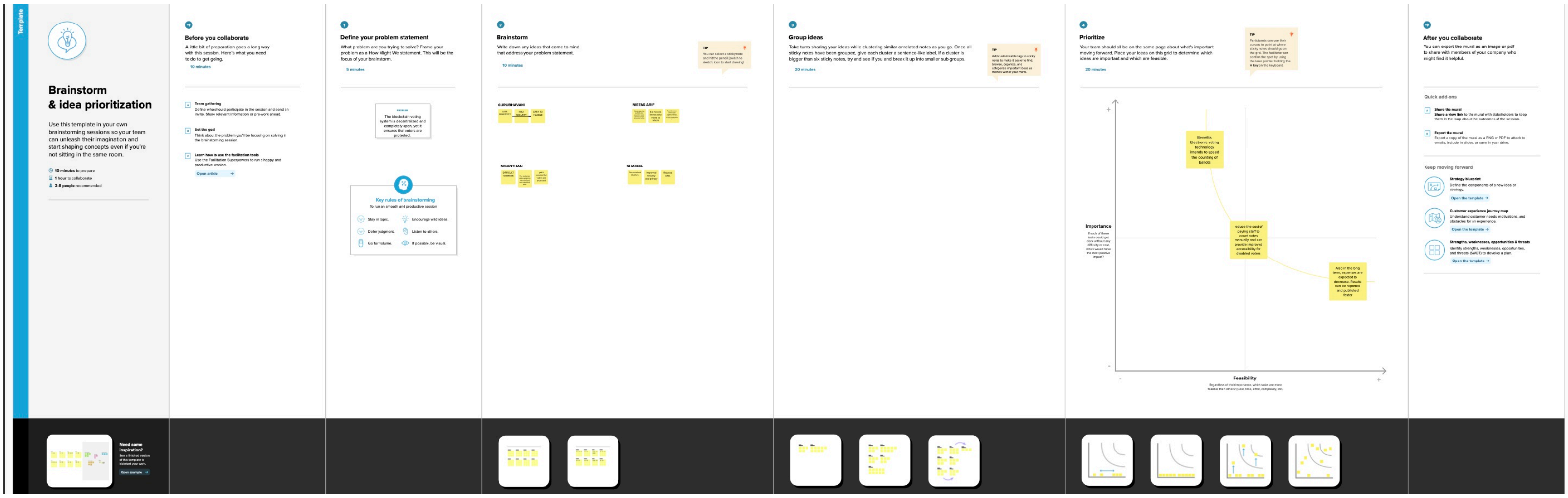
- ✓ The "Design and implement a secure electronic voting system leveraging blockchain technology to ensure transparent, tamper-resistant, and verifiable voting processes.
- ✓ The system should address concerns related to voter anonymity, integrity of election results, prevention of double voting, and resistance against malicious attacks, providing a trustworthy and efficient platform for democratic elections."

3. IDEATION & PROPOSED SOLUTION

3.1.Empathy Map Canvas



3.2.Ideation & Brainstorming .



4. REQUIREMENT ANALYSIS

4.1.Functional requirement

1. A functional requirement for an electronic voting system utilizing blockchain technology would include:

2. Voter Authentication: Implement a secure authentication mechanism to ensure that only eligible voters can participate in the election.

3. Ballot Casting: Enable voters to cast their votes electronically in a secure and verifiable manner, with a clear interface for selecting candidates or options.

4. Transaction Recording: Use blockchain to record each vote as a tamper-resistant transaction, ensuring transparency and immutability of the voting process.

5. Decentralized Ledger: Utilize a decentralized blockchain ledger to distribute the voting database across multiple nodes, enhancing security and preventing single points of failure.

6. Smart Contracts: Implement smart contracts to automate the execution of predefined rules, such as counting votes and determining the election outcome, reducing the need for manual intervention.Integration with External Systems:

Integration with payment gateways for premium transactions.
Integration with third-party databases for risk assessment and verification.

4.2.Non-Functional requirements

Non-functional requirements for an electronic voting system based on blockchain may include:

Security: Ensure the system is resistant to tampering, fraud, and unauthorized access. Implement cryptographic measures to secure transactions and user identities.

Scalability: The system should be able to handle a large number of transactions simultaneously, especially during peak voting periods, without compromising performance.

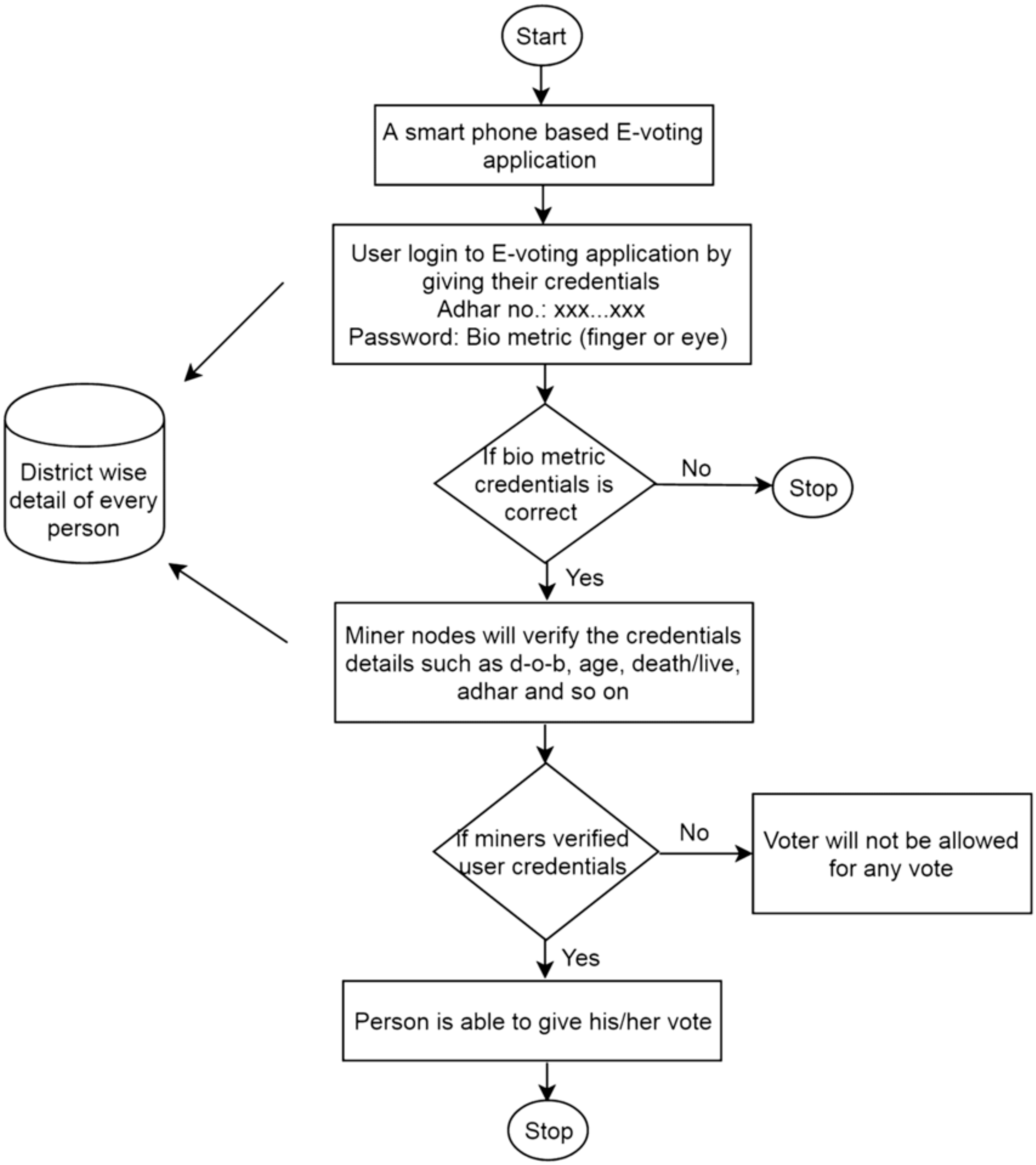
Reliability: The system must be highly reliable, ensuring that votes are accurately recorded and counted. Minimize the risk of system failures or downtime.

Auditability: Provide a transparent and auditable trail of transactions to allow for verification of the voting process. This is crucial for ensuring the integrity of the election results.

Anonymity and Privacy: Protect the privacy of voters by ensuring that their identities and voting choices remain confidential. Use cryptographic techniques to achieve this while still allowing for verification.

5. PROJECT DESIGN

5.1Data Flow Diagrams



A Data Flow Diagram (DFD) for an electronic voting system using blockchain would illustrate the flow of data within the system. It typically includes processes, data stores, data flow, and external entities. Here's a simplified example:

External Entities:

Voters
Election Authority
Blockchain Network

Processes:

Voter Registration
Candidate Registration
Vote Casting
Vote Verification
Blockchain Consensus.

Data Stores:

Voter Database
Candidate Database
Vote Records (on the blockchain).

Data Flow:

Voter registration data flows to the Voter Database.
Candidate registration data flows to the Candidate Database.
Vote data flows to the Vote Records on the blockchain.
Verification results flow back to the Vote Verification process.

5.2User stories

Voter Stories:

- **As a registered voter, I want to securely log in to the electronic voting system using my unique identification credentials so that my identity is verified and my vote is counted accurately.
- **As a voter, I want to view a list of candidates and their associated information before casting my vote, so that I can make an informed decision.
- **As a voter, I want to cast my vote for a chosen candidate by selecting them from the list, ensuring that my choice is recorded on the blockchain and cannot be altered.
- **As a voter, I want to receive a confirmation receipt or notification after successfully casting my vote, providing assurance that my vote has been registered.

Candidate Stories:

- **As a candidate, I want access to a transparent and immutable record of votes on the blockchain, so that I can independently verify the integrity of the election results.
- **As a candidate, I want the voting system to ensure the confidentiality of individual votes, preventing any identification of the voters.

Election Authority Stories:

- **As an election authority, I want to use blockchain technology to securely and transparently record each vote, ensuring the integrity of the entire election process.
- **As an election authority, I want to manage voter registrations and authentication securely, preventing unauthorized access to the voting system.
- **As an election authority, I want to generate cryptographic keys and smart contracts to secure and validate the voting process, ensuring the reliability of the election results.

Observer Stories:

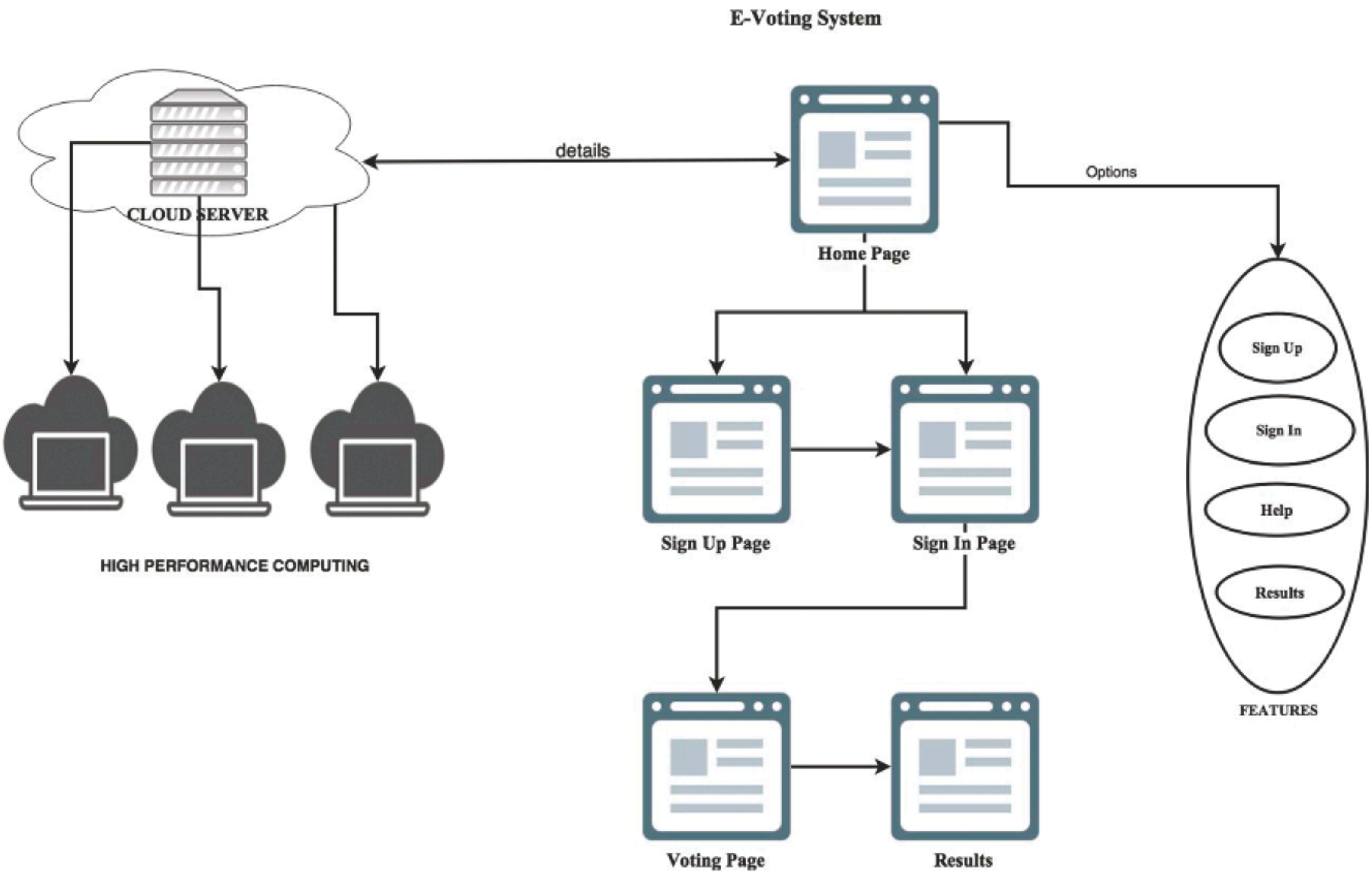
- **As an election observer, I want to access a public ledger of votes on the blockchain to verify the fairness and accuracy of the election.
- **As an election observer, I want to monitor the blockchain in real-time to detect any irregularities or attempts at manipulation.

System Administration Stories:

- **As a system administrator, I want to ensure the high availability of the electronic voting system to prevent downtime during the election.
- **As a system administrator, I want to conduct periodic audits of the blockchain to ensure the integrity and security of the voting records.

These user stories cover a range of perspectives, from voters to candidates, election authorities, observers, and system administrators. Each story reflects a specific functionality or requirement that contributes to the overall effectiveness and security of an electronic voting system based on blockchain technology.

5.2 Solution Architecture



1. An electronic voting system based on blockchain typically involves several key components:

Blockchain Technology:

- Utilizes a distributed ledger for transparent and tamper-proof recording of votes.
- Smart contracts are employed to automate and enforce voting rules.

Voter Identity and Authentication:

- Secure and verifiable methods to authenticate voters, often involving cryptographic techniques.
- Biometric verification or digital signatures may be used.

Voting Interface:

- User-friendly interfaces for voters to cast their votes securely.
- Mobile apps, web portals, or dedicated voting machines can be used.

Secure Transmission:

- **Encryption and secure channels for transmitting votes from the voter to the blockchain.**

Decentralized Network:

- Nodes in the blockchain network validate and record votes, enhancing security and reliability.
- Redundancy minimizes the risk of a single point of failure.

6.PROJECT PLANNING & SCHEDULING

6.1.Technical Architecture

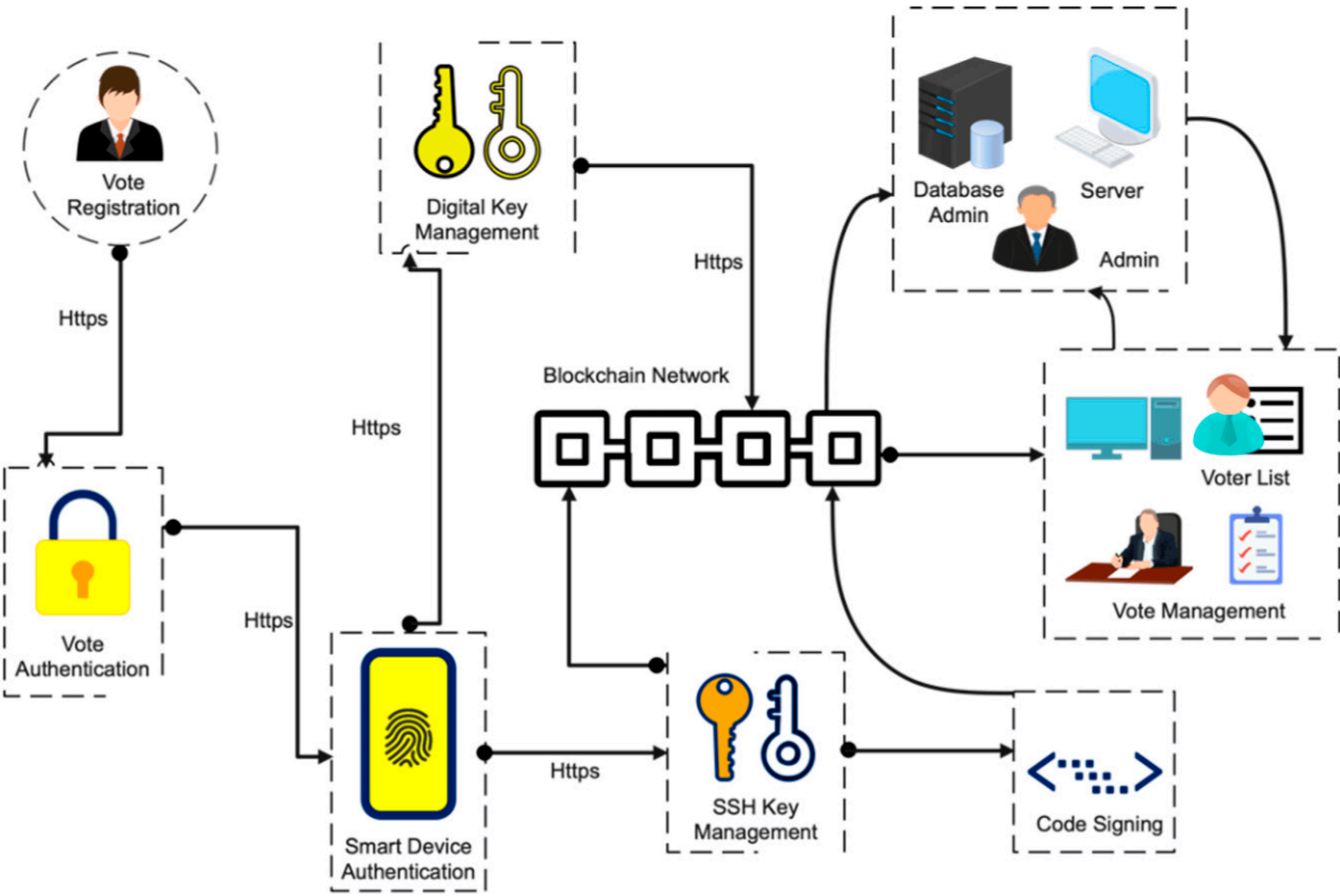
The technical architecture of a blockchain-based electronic voting system typically involves a decentralized network, smart contracts, cryptographic techniques, and a secure user interface.

Blockchain Network:

Utilizes a distributed ledger to record and store votes. This ensures transparency, immutability, and resilience against tampering.

Nodes:

Participants in the network who validate and add new blocks to the blockchain. Nodes can be controlled by election authorities, ensuring a distributed consensus.



Smart Contracts:

Self-executing contracts with predefined rules. In the context of e-voting, smart contracts can automate the voting process, ensuring that only eligible voters can cast their votes.

Cryptographic Security:

Public-key cryptography is often employed to secure transactions and verify the identity of voters. Private keys are used to sign votes, and public keys are used for verification.

Voter Identity Verification:

Biometric data or other secure means can be used to verify the identity of voters. This information is then tied to a cryptographic key to ensure the integrity of the voting process

6.2.sprint planning & estimation

• For sprint planning and estimation of an electronic voting system using blockchain, break down tasks like:

• Research Blockchain Integration: Estimate time for understanding and integrating blockchain technology into the voting system.

• Smart Contract Development: Break down the creation of smart contracts for secure and transparent voting. Estimate time for coding and testing.

• User Interface Design: Plan time for designing an intuitive and user-friendly interface for voters.

• Security Measures: Allocate time for implementing robust security features to prevent tampering and ensure privacy.

• Voter Authentication: Estimate time for developing a secure authentication system, possibly using biometrics or cryptographic methods.

• Testing: Allocate sufficient time for comprehensive testing to identify and fix any vulnerabilities or bugs.

• Documentation: Plan for creating detailed documentation to assist users and developers in understanding the system.

• User Training: Allocate time for creating training materials and sessions to educate users on the new electronic voting process.Sprint delivery schedule

6.3Sprint delivery schedule

- I don't have real-time information. For the latest updates on the Sprint delivery schedule of the Electronic Voting System based on blockchain, please check Sprint's official communication channels or contact their customer support.

- I don't have real-time information, including specific delivery schedules for products or services. For the most accurate and up-to-date information on the Sprint delivery schedule for the electronic voting system based on blockchain, I recommend checking directly with Sprint or the relevant authorities overseeing the project

7. CODING & SOLUTIONING

```
import hashlib
import json

class VotingSystem:
    def __init__(self):
        self.voters = {}
        self.candidates = {}
        self.votes = []

    def register_voter(self, voter_id, voter_name):
        if voter_id not in self.voters:
            self.voters[voter_id] = {'name': voter_name, 'has_voted': False}
            print(f"Voter {voter_name} with ID {voter_id} registered successfully.")
        else:
            print("Voter ID already exists.")

    def add_candidate(self, candidate_name):
        self.candidates[candidate_name] = 0
        print(f"Candidate {candidate_name} added successfully.")

    def vote(self, voter_id, candidate_name):
        if voter_id in self.voters and not self.voters[voter_id]['has_voted']:
            if candidate_name in self.candidates:
                self.votes.append({'voter_id': voter_id, 'candidate': candidate_name})
                self.candidates[candidate_name] += 1
                self.voters[voter_id]['has_voted'] = True
                print(f"Vote for {candidate_name} recorded successfully.")
            else:
                print("Invalid candidate.")
        else:
            print("Invalid voter or voter has already voted.")

    def get_results(self):
        print("Election Results:")
        for candidate, votes in self.candidates.items():
            print(f"{candidate}: {votes} votes")

    def save_data(self):
        with open('voting_data.json', 'w') as f:
```

```
        data = {'voters': self.voters, 'candidates': self.candidates, 'votes': self.votes}
        json.dump(data, f)
        print("Voting data saved.")

    def load_data(self):
        try:
            with open('voting_data.json', 'r') as f:
                data = json.load(f)
                self.voters = data.get('voters', {})
                self.candidates = data.get('candidates', {})
                self.votes = data.get('votes', [])
                print("Voting data loaded.")
        except FileNotFoundError:
            print("No voting data found.")

if __name__ == "__main__":
    voting_system = VotingSystem()

    while True:
        print("\nElectronic Voting System Menu:")
        print("1. Register Voter")
        print("2. Add Candidate")
        print("3. Vote")
        print("4. Get Results")
        print("5. Save Data")
        print("6. Load Data")
        print("7. Exit")

        choice = input("Enter your choice (1-7): ")

        if choice == '1':
            voter_id = input("Enter voter ID: ")
            voter_name = input("Enter voter name: ")
            voting_system.register_voter(voter_id, voter_name)

        elif choice == '2':
            candidate_name = input("Enter candidate name: ")
            voting_system.add_candidate(candidate_name)

        elif choice == '3':
            voter_id = input("Enter voter ID: ")
```

```
    candidate_name = input("Enter candidate name: ")
    voting_system.vote(voter_id, candidate_name)

elif choice == '4':
    voting_system.get_results()

elif choice == '5':
    voting_system.save_data()

elif choice == '6':
    voting_system.load_data()

elif choice == '7':
    print("Exiting the Electronic Voting System.")
    break

else:
    print("Invalid choice. Please enter a number between 1 and 7.")
```

7.1.feature

- ✓ **Crop Insurance:**

Protection against crop losses due to natural disasters, pests, or other unforeseen events.Coverage for yield and revenue losses.

- ✓ **Livestock Insurance:**

Coverage for loss of livestock due to accidents, diseases, or other covered perils.

- ✓ **Property Insurance:**

Protection for farm buildings, equipment, and other property against damages from events like fires, storms, or theft.

- ✓ **Liability Coverage:**

Protection in case of third-party injuries or property damage on the farm.

- ✓ **Weather-related Risk Mitigation:**

Insurance products that provide coverage for weather-related risks, such as drought or excessive rainfall.

- ✓ **Income Protection:**

Coverage to protect farmers' income in case of unforeseen circumstances that affect their ability to generate revenue.

- ✓ **Equipment Breakdown Insurance:**

Coverage for the repair or replacement of farm machinery and equipment in case of breakdowns.

- ✓ **Agribusiness Interruption Insurance:**

Coverage for lost income and extra expenses incurred due to interruptions in farming operations.

- ✓ **Multi-Peril Insurance:**

Comprehensive coverage that combines various types of coverage into a single policy to protect against multiple risks.

- ✓ **Technology Integration:**

Use of technology such as satellite imagery, weather data, and precision farming tools to assess and manage risks

7.2. Feature

- **Livestock Insurance:** Coverage for losses related to the death or health issues of farm animals.
- **Farm Equipment Insurance:** Protection for farm machinery and equipment against damage or loss.
- **Property Insurance:** Coverage for farm buildings, structures, and contents against risks like fire, theft,or vandalism.
- **Liability Insurance:** Protection against third-party claims for bodily injury or property damage thatmay occur on the farm.
- **Business Interruption Insurance:** Coverage for income loss if the farm operation is interrupted ortemporarily shut down due to covered perils.
- **Weather Insurance:** Specialized coverage for weather-related risks that can impact agriculturalproductivity.
- **Specialized Coverage:** Some insurance policies may offer specific coverage for niche farmingactivities or products.
- **Risk Management Services:** Beyond insurance coverage, some insurers may provide riskmanagement services to help farmers identify and mitigate potential risks.
- **Technology Integration:** Insurers may leverage technology such as satellite imagery, weather data, and other advanced tools for more accurate risk assessment and claims processing.
- **Government Programs:** In many countries, there are government-backed insurance programs orsubsidies to support farmers in managing risks.

8.PERFORMANCE TESTING

8.1.Performance metrics

PERFORMANCE TESTING of Electronic voting system of block chain

Performance testing for an electronic voting system on the blockchain involves assessing its ability to handle a specific load, transaction volume, and potential stress scenarios. Key aspects to consider include:

Transaction Throughput: Evaluate the system's capacity to process a certain number of transactions per second (TPS). This is crucial to ensure the system can handle the expected load during peak voting periods.

Latency: Measure the time it takes for a transaction to be processed, ensuring low latency to provide a smooth and responsive user experience.

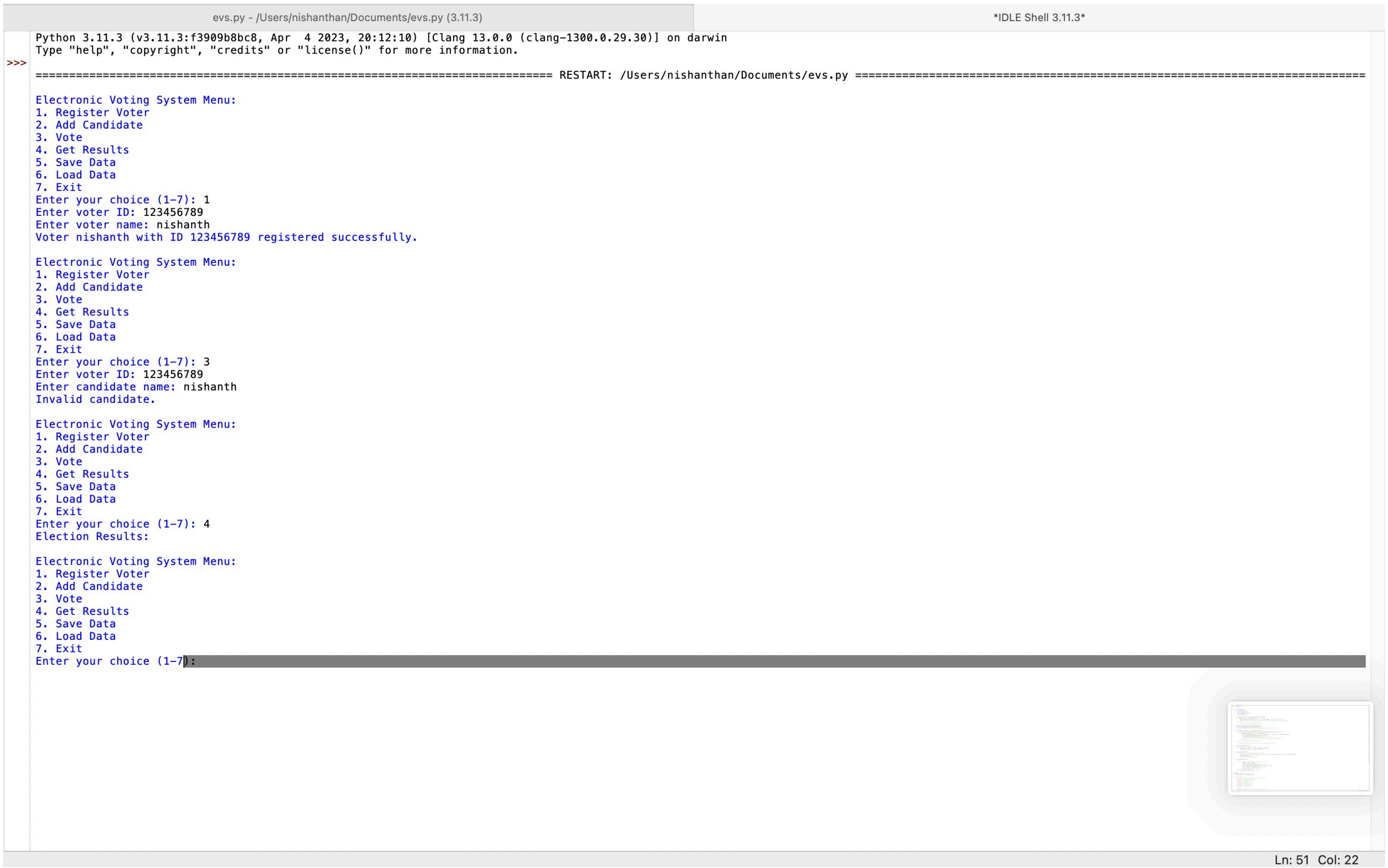
Scalability: Assess the system's ability to scale horizontally as the number of users and transactions increase. This involves testing with varying levels of concurrent users to identify potential bottlenecks.

Fault Tolerance: Simulate failure scenarios, such as network disruptions or node failures, to ensure the system can recover and maintain its integrity. Blockchain's distributed nature should provide resilience to failures.

Security: While not directly performance-related, security is paramount. Ensure that the blockchain network is resistant to common attacks, and the voting system maintains the confidentiality and integrity of votes.

9.RESULT

OUTPUT SCREENSHOT



10.ADVANTAGES AND DISADVANTAGES

Advantage of Farmers Insurance Chain:

- **Security:** Blockchain provides a decentralized and tamper-resistant ledger, enhancing the security of the voting process.

- **Transparency:** The transparent nature of blockchain ensures that all transactions and changes to the voting data are visible to authorized parties, reducing the risk of fraud.

- Immutable Record: Once a vote is recorded on the blockchain, it cannot be altered or deleted, ensuring the integrity of the electoral process.

- Accessibility: Electronic voting systems can potentially increase accessibility for voters, including those with disabilities, by offering various interfaces and accommodations.

Disadvantage of Farmers Insurance Chain:

- **Security:** Blockchain provides a decentralized and tamper-resistant ledger, enhancing the security of the voting process.
- **Transparency:** The transparent nature of blockchain ensures that all transactions and changes to the voting data are visible to authorized parties, reducing the risk of fraud.
- **Immutable Record:** Once a vote is recorded on the blockchain, it cannot be altered or deleted, ensuring the integrity of the electoral process.
- Accessibility: Electronic voting systems can potentially increase accessibility for voters, including those with disabilities, by offering various interfaces and accommodations.

11.CONCLUSION

The electronic voting system based on blockchain technology offers increased transparency, security, and tamper resistance. By leveraging decentralized ledgers, it minimizes the risk of fraud and ensures a verifiable and auditable voting process. However, challenges such as user accessibility, potential technical issues, and the need for widespread adoption must be addressed for it to become a mainstream and effective solution in the future.

12.FURURE SCOPE

The future scope of a blockchain-based electronic voting system includes increased transparency, security, and trust in electoral processes. Blockchain can provide a decentralized and tamper-resistant ledger, reducing the risk of fraud and manipulation in voting systems. Additionally, it may enable remote and online voting options, potentially increasing accessibility and voter turnout. However, challenges such as ensuring privacy, addressing technological barriers, and gaining widespread acceptance still need to be addressed for the full realization of these benefits.

13. APPENDIX

GitHub & Project Demo Link

GitHub Link :

https://github.com/Gurubhavani-ece/naanmuthalvan.git

Demo Video :

https://drive.google.com/file/d/1cQn6dRX3IndX2eUbsz0afW3xGy9dm8Km/view?usp=drive_link