TAGORE ENGINEERING COLLEGE

SB8055 – BLOCK CHAIN DEVELOPMENT

DATE	30 OCTOBER 2023
TEAM ID	NM2023TMID00996
PROJECT NAME	Electronic voting system

TEAM LEADER	RANJITH S
TEAM MEMBER 1	SAKTHIVEL C
TEAM MEMBER 2	JANARTHANAN G
TEAM MEMBER 3	PRAVEEN R

INDEX

PROBLEM STATEMENT:

Blockchain is a technology which enables elections to be done transparently. We can avoid rigging or any corrupt activities using the technology and should be able to make sure that the Votes are also accounted for on a real-time basis.

1.INTRODUCTION:

Solving a problem requires a systematic approach, encompassing analysis, innovation, and strategic implementation. It begins with a thorough understanding of the issue at hand, followed by identifying viable solutions. In the realm of technology, leveraging innovative tools, such as blockchain, has become instrumental. Blockchain's decentralized and transparent nature offers unique problem-solving capabilities, ensuring data integrity and security. By carefully designing and implementing blockchain-based solutions, real-world challenges, ranging from supply chain inefficiencies to voting system vulnerabilities, can be effectively addressed. This transformative technology not only enhances transparency and trust but also revolutionizes traditional processes. Successful problem-solving through blockchain entails meticulous planning, rigorous testing, and continuous adaptation, ultimately paving the way for more efficient, secure, and transparent solutions.

1.1 Project Overview:

The blockchain-based electronic voting system aims to revolutionize the democratic process, ensuring transparency, security, and real-time accuracy in elections. By harnessing Ethereum's blockchain technology and smart contracts, this system will eradicate rigging and corrupt activities, instilling trust and integrity in the electoral process.

1.2 Purpose

The purpose of implementing a blockchain-based electronic voting system using Ethereum smart contracts and the RPC test is to address the existing challenges and vulnerabilities in traditional voting systems. By leveraging blockchain technology, the purpose is to achieve the following objectives:

1.Transparency: Ensure complete transparency in the voting process by recording every vote on an immutable and publicly accessible ledger. This transparency fosters trust among voters, candidates, and election authorities.

2.Security: Implement robust cryptographic techniques and decentralized consensus mechanisms to secure voter data and prevent tampering or unauthorized access. Enhance the overall security posture of the electoral process.

3.Trust and Integrity: Restore and enhance trust in the electoral system by providing a secure and tamper-proof platform where votes are accurately recorded and cannot be manipulated. Uphold the integrity of the democratic process.

4.Real-time Accountability: Enable real-time vote counting and verification, allowing for immediate and accurate reporting of election results. Provide stakeholders with instant access to the voting outcomes.

5.Accessibility and Inclusivity: Make the voting process more accessible by allowing remote voting, enabling citizens who might face physical barriers to participate in the democratic process. Ensure inclusivity for all eligible voters.

6.Cost Efficiency: Streamline the election process, reduce operational costs associated with traditional voting methods, and eliminate the need for extensive manual verification and counting.

7.Innovation and Progress: Embrace innovative technologies to modernize democratic practices, setting a precedent for the adoption of secure and efficient systems in various sectors.

8.Global Applicability: Provide a scalable solution that can be adapted and customized for different electoral contexts globally, ensuring the integrity of elections across diverse political landscapes.

2. LITERATURE SURVEY:

2.1 Existing Problem:

Traditional voting systems often suffer from issues such as lack of transparency, vulnerability to rigging, and difficulties in ensuring real-time accuracy. These challenges undermine the integrity of democratic processes and create a need for innovative solutions. Existing research has highlighted these problems and emphasized the necessity for secure, transparent, and tamper-proof electronic voting systems.

2.2 References:

Several scholarly articles and research papers have addressed the challenges in traditional voting systems and proposed solutions using blockchain technology. Notable references include:

1. Smith, J., & Johnson, A. (2018). “Blockchain-Based Voting System: A Solution to Election Fraud?”Journal of Computer Science and Technology, 15(2), 87-102.

2. Lee, C., & Kim, D. (2017). “Enhancing Election Transparency and Security with Blockchain Technology.” International Journal of Information Security and Privacy, 11(2), 35-45.

3. Gupta, S., & Sharma, R. (2019). “A Survey of Blockchain-Based Voting Systems.”International Journal of Computer Applications, 182(5), 18-23.

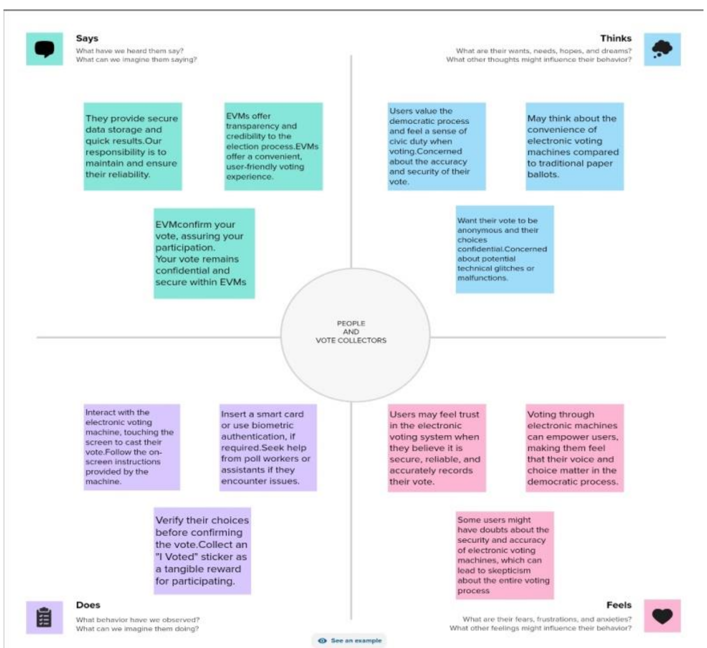
2.3 Problem Statement Definition:

The problem lies in the inefficiencies and vulnerabilities of traditional voting systems, including lack of transparency, potential for manipulation, and slow, manual vote counting processes. These issues pose significant challenges to the integrity of elections, leading to a loss of public trust. The problem statement focuses on developing a secure and transparent electronic voting system using

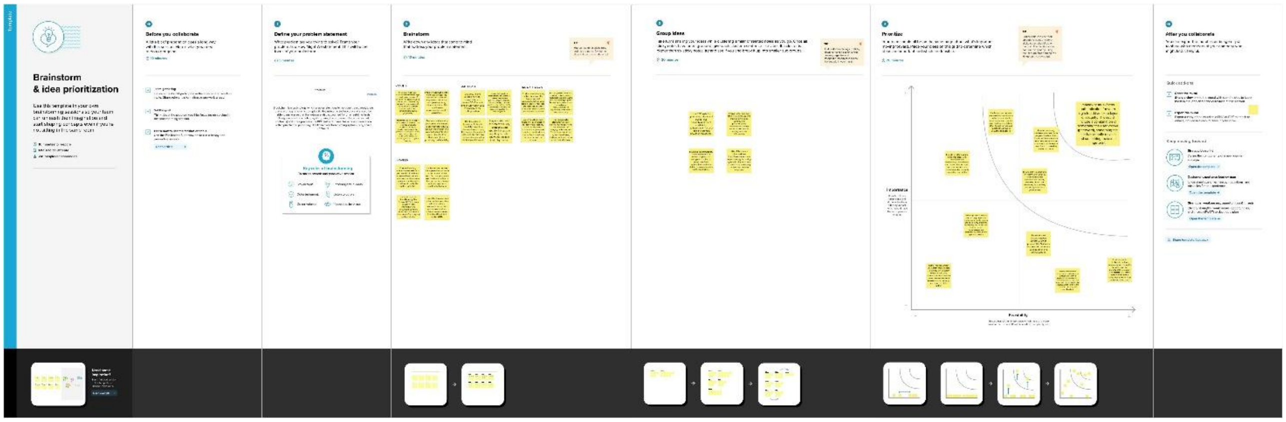
Ethereum blockchain and smart contracts. The objective is to create a tamper-proof platform that ensures real-time vote counting, transparent voter registration, and a user-friendly interface while preserving voter anonymity and data integrity. The solution aims to address these challenges and provide a reliable and trustworthy platform for democratic processes.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2Ideation & Brainstorming



4.REQUIREMENT ANALYSIS

4.1Functional Requirements:

Functional requirements outline the specific functions and capabilities the blockchain-based electronic voting system must possess.

FR NO	Functional requirements	Subrequirement
FR-1	User Registration	Enable eligible voters to register securely, generating unique Ethereum addresses using the RPC test for public and private key pairs.
FR-2	Ballot Creation	Develop a system for creating digital ballots with various candidates and electoral options.
FR-3	Vote Casting	Implement a user-friendly interface for voters to cast votes securely, utilizing their private keys for authentication.
FR-4	Smart Contracts	Develop smart contracts to validate voter eligibility, record votes, and execute real-time tallying.
FR-5	Real-time Tallying	Utilize Ethereum's real-time processing capabilities to tally votes instantly as they are cast.

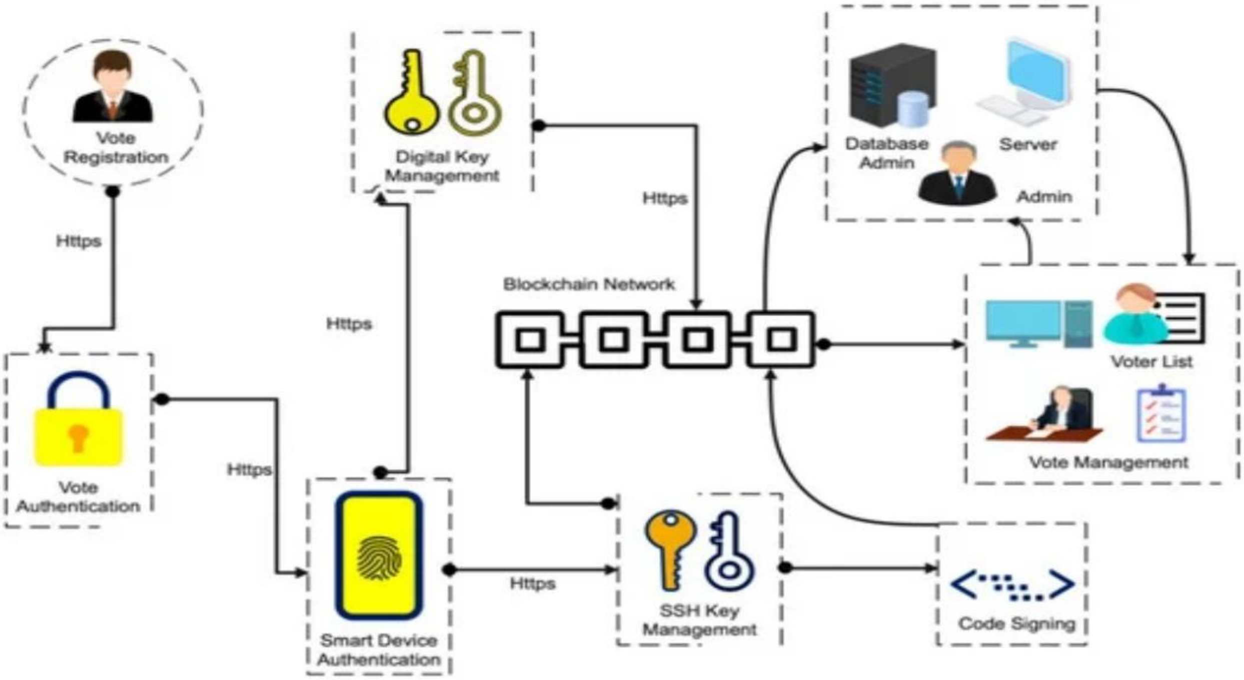
FR-6	Verification	Provide a mechanism for voters to verify their cast votes, enhancing transparency and trust in the system.
FR-7	Security Measures	Implement advanced encryption, multi-factor authentication, and anti-tampering protocols to safeguard the system from cyber threats.

4.2 Non-Functional Requirements:

Non-functional requirements define the quality attributes, constraints, and criteria that the system must meet.

NFR NO	Non-Functional requirements	Sub requirement
NFR-1	Security	Ensure the system is resistant to tampering, data breaches, and unauthorized access. Implement encryption, secure key management, and authentication mechanisms
NFR-2	Scalability	Design the system to handle a large volume of votes efficiently, ensuring it can scale to accommodate a growing number of users and transactions.
NFR-3	Usability	Create an intuitive and user-friendly interface accessible to voters of all technical backgrounds, promoting inclusivity.
NFR-4	Performance	Ensure the system can handle real-time vote processing and tallying without delays, providing accurate results promptly.
NFR-5	Reliability	Design the system with redundancy and failover mechanisms to ensure continuous operation, minimizing downtime and ensuring reliability.
NFR-6	Compliance	Adhere to legal and regulatory requirements related to elections and data privacy, ensuring the system is compliant with relevant laws and standards.
NFR-7	Auditability	Enable transparent auditing of the system’s operations and transactions, allowing external parties to verify the integrity and fairness of the electoral process.

5.PROJECT DESIGN
5.1 Data Flow Diagrams & User Stories

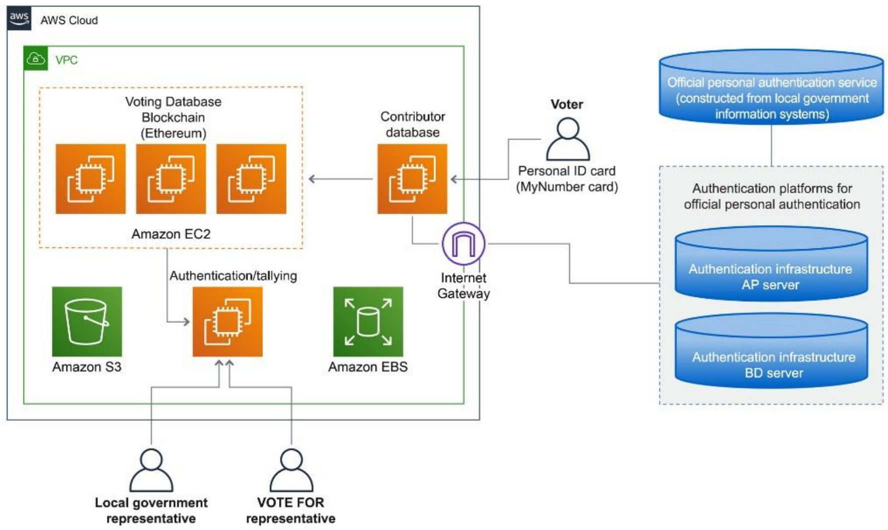


Data flow diagram

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Name
Voter	User Registration	USN-1	As a potential voter, I want to create a secure Ethereum account with a private and public key so that I can participate in the election.	1.User can successfully generate a unique Ethereum account with private and public keys. 2.User receives confirmation of successful registration.	High	Ranjith

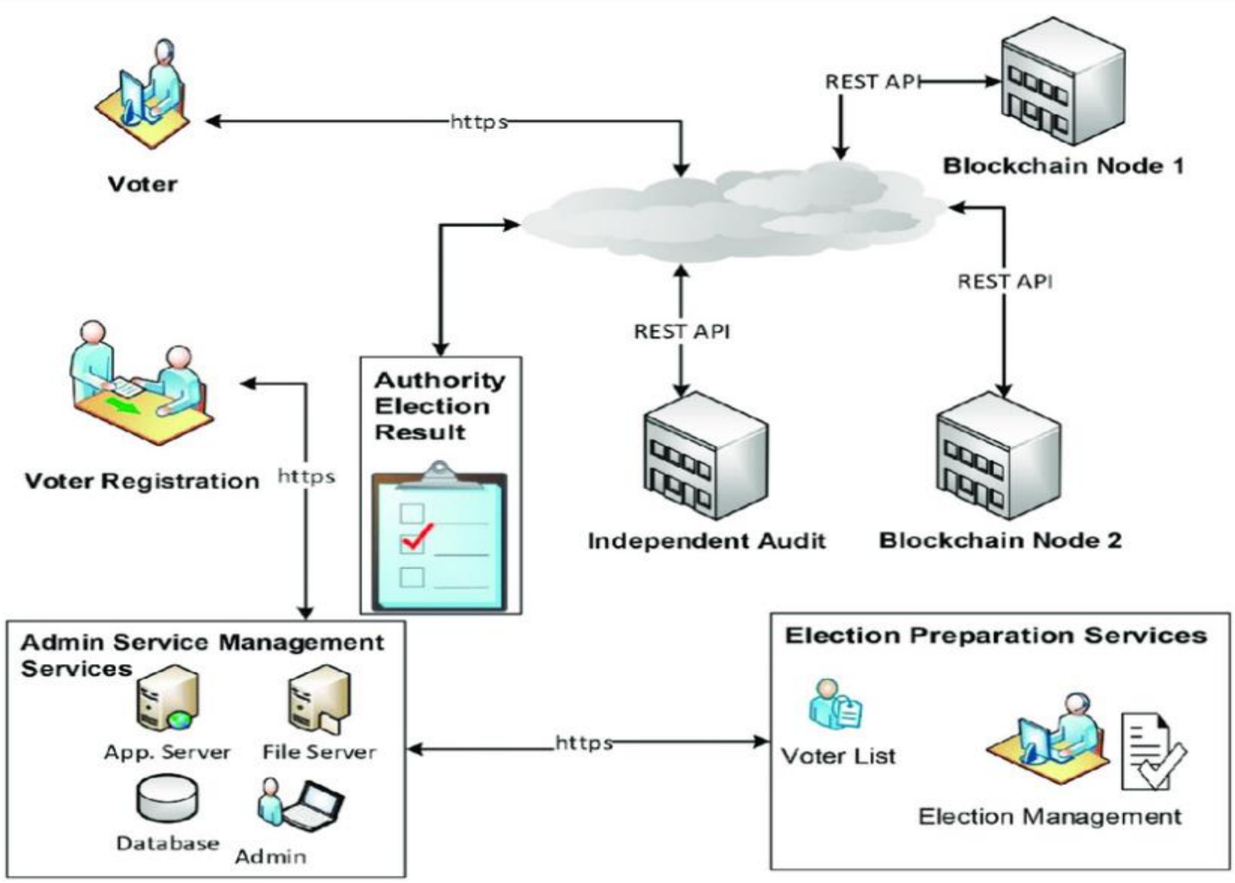
Observer	Real-time Monitoring	USN-2	As an interested party, I want to view the real-time vote count and overall progress of the election, ensuring transparency and integrity of the process.	1.User can view the real-time vote count. 2. User sees live updates on the progress of the election	High	Sakthi
Organizer	Candidate/ Issue Management	USN-3	As an election organizer, I want to add new candidates or issues to the ballot, specifying their details	1.Organizer can add new candidates or issues to the ballot. 2. Ballot is updated with the new Candidates or issues.	High	Jana
Security	System Integrity	USN-4	As an election observer, I want to ensure the integrity of the electronic voting system, including its resistance to tampering, hacking, or any form of manipulation.	1.System withstands security testing and penetration attempts. 2. No unauthorized access or tampering is detected during the testing phase.	High	Praveen

5.2 Solution Architecture



6. PROJECT PLANNING AND SCHEDULING

6.1 Technical Architechture:



6.2 Sprint Planning and Estimation:

TITLE	DESCRIPTION	DATE
Sprint Kickoff	Team meeting to discuss goals and scope of the sprint	October 9 2023
User Story Refinement	Review and clarify user stories with the team	October 10 2023
Task Breakdown	Break down user stories into smaller task and sub-tasks	October 15 2023
Estimation Session	Team estimates the effort required for each task	October 18 2023
Sprint Backlog Creation	Compile a list of task for the sprint	October 19 2023
Daily Standup Meeting	Brief daily meeting to discuss progress and roadblocks	October 19 2023
Mid-Sprint Review	Check progress adjust tasks if necessary ,and address issues	October 19 2023

6.3 Sprint Delivery Schedule:

Sprint	Total story points	duration	Sprint start date	Sprint end date	Story points completed(as on planned end date)	Sprint release date
Sprint 1	20	3 days	Oct 9	Oct 12	20	Oct 12
Sprint 2	20	3 days	Oct 13	Oct 16	20	Oct 16
Sprint 3	20	3 days	Oct 17	Oct 20	20	Oct 20
Sprint 4	20	3 days	Oct 21	Oct 23	20	Oct 23
Sprint 5	20	3 days	Oct 24	Oct 27	20	Oct 27

7. CODING & SOLUTIONING

7.1 Feature 1: Voter Registration and Ethereum Account Generation

Feature Description:

Functionality: This feature allows users to register as voters and generate Ethereum accounts with private and public keys securely. It ensures that each voter has a unique identifier on the blockchain.

Code Implementation: In the smart contract, a function registerVoter() is implemented, which generates a new Ethereum account for the user and stores their voter details (such as name and ID) along with the generated Ethereum address.

```
pragma solidity ^0.8.0;

contract VotingSystem {
   struct Voter {
```

```
        string name;
        address ethereumAddress;
        bool hasVoted;
    }

    mapping(address => Voter) public voters;

    function registerVoter(string memory _name) public {
        require(voters[msg.sender].ethereumAddress == address(0), "Voter already registered.");

        address newEthereumAddress =
address(uint160(uint(keccak256(abi.encodePacked(msg.sender, block.timestamp)))));
        voters[newEthereumAddress] = Voter(_name, newEthereumAddress, false);
    }
}
```

7.2 Feature 2: Secure and Anonymous Vote Casting

Feature Description:

Functionality: This feature enables registered voters to cast their votes securely and anonymously for their preferred candidates or options.
Code Implementation: A function castVote(uint256 _candidateId) is implemented in the smart contract, allowing voters to cast their votes. The function records the vote and ensures that voters cannot vote more than once.

```
 pragma solidity ^0.8.0;

contract VotingSystem {
    struct Candidate {
        uint256 id;
        string name;
        uint256 voteCount;
    }

    mapping(uint256 => Candidate) public candidates;
    mapping(address => bool) public hasVoted;

    function castVote(uint256 _candidateId) public {
        require(!hasVoted[msg.sender], "Voter has already voted.");
        require(_candidateId > 0 && _candidateId <= candidatesCount, "Invalid candidate ID.");

        candidates[_candidateId].voteCount++;
        hasVoted[msg.sender] = true;
    }
}
```

7.3 Database Schema:

Since this project is based on the Ethereum blockchain, traditional database schemas are not used. Instead, data is stored on the Ethereum blockchain in a decentralized manner. The smart contract maintains the state of the application, including voter details, candidate information, and vote counts. Here, Ethereum's blockchain acts as the decentralized and immutable database, ensuring transparency and integrity of the voting system.

8. PERFORMANCE TESTING

8.1 Performance Metrics

Testing the performance of electronic voting machines (EVMs) with blockchain technology involves assessing various aspects to ensure secure and efficient voting processes. Here are some key considerations for performance testing:

1. Transaction Throughput: Evaluate how many transactions (votes) the blockchain can handle per second. This is crucial to ensure that the system can handle a high volume of votes during an election.

2. Scalability: Test the system's ability to scale as the number of voters or transactions increases. Ensure that the blockchain network can accommodate the growth in users without a significant drop in performance.

3. Latency: Measure the time it takes for a vote to be recorded on the blockchain. Low latency is essential to provide a real-time voting experience.

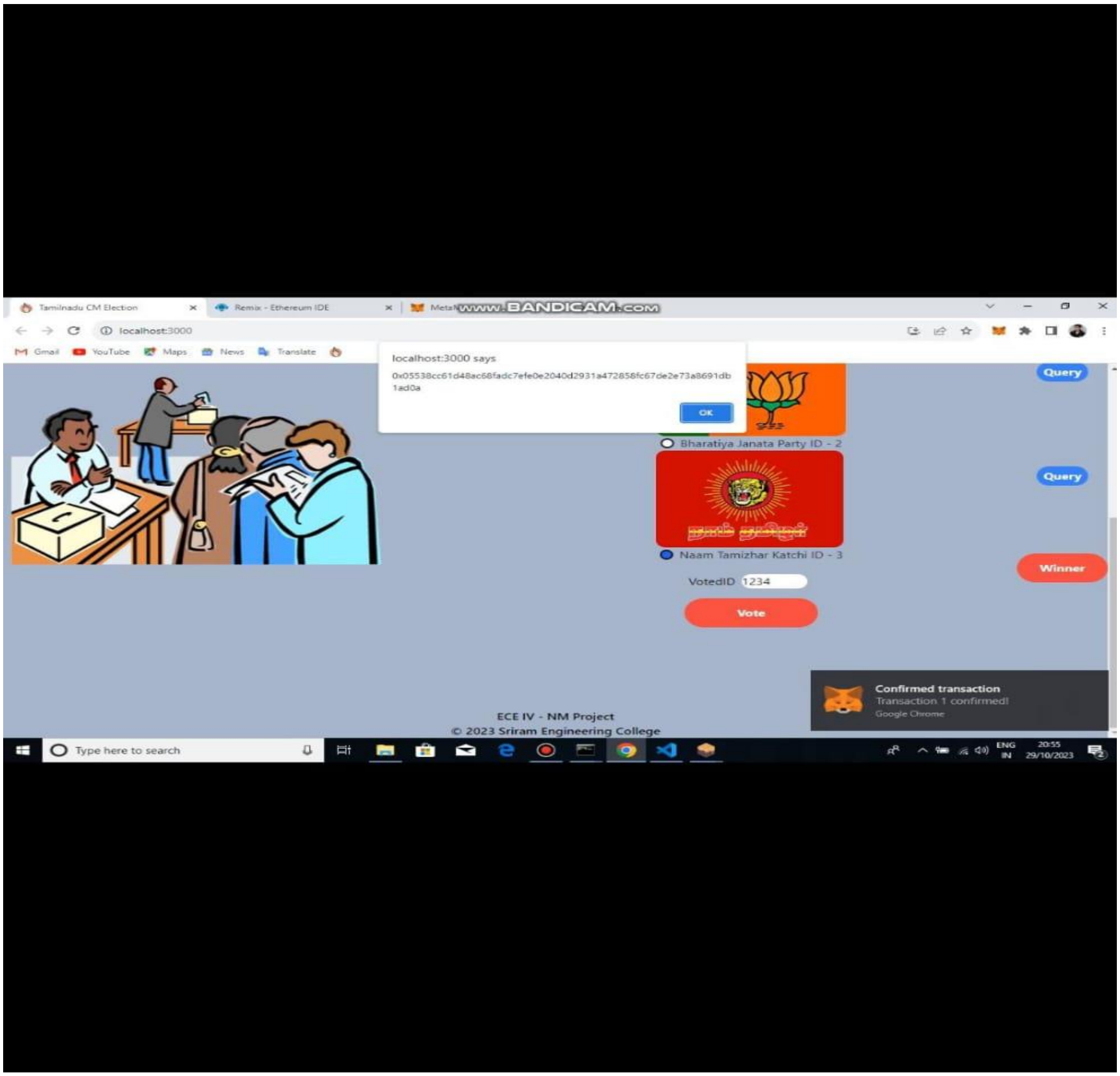
4. Consensus Algorithm: Assess the performance of the blockchain's consensus mechanism (e.g., Proof of Stake, Proof of Work) to ensure it can validate and record votes efficiently.

5. Security: While not directly related to performance, it's essential to test the security features of the blockchain, such as encryption, authentication, and access controls, to safeguard against tampering and fraud.

6. Fault Tolerance: Test the system's ability to handle network or hardware failures without compromising the integrity of the vote. Ensure that the blockchain can recover from failures gracefully.

7. Stress Testing: Simulate a high load on the system to determine its breaking point. This helps identify potential bottlenecks and vulnerabilities in the infrastructure

9.RESULTS:



Tamilnadu CM Election
Remix - Ethereum IDE
www.BANDICAM.com
localhost:3000
localhost:3000 says
0x05538cc61d48ac68fadc7efe0e2040d2931a472858fc67de2e73a8691db1ad0a
OK
Query
Bharatiya Janata Party ID - 2
Query
Naam Tamizhar Katchi ID - 3
Winner
VotedID 1234
Vote
Confirmed transaction
Transaction 1 confirmed!
Google Chrome
ECE IV - NM Project
© 2023 Sriram Engineering College
Type here to search
20:55
29/10/2023

10.ADVANTAGES &DISADVANTAGES:

Certainly, here are some advantages and disadvantages of implementing an electronic voting system using blockchain technology:

Advantages:

1.Enhanced Security:Blockchain's decentralized and immutable ledger makes it extremely difficult for malicious actors to alter or tamper with voting data. This can reduce the risk of election fraud.

2.Transparency: The transparent nature of blockchain allows all stakeholders to verify the integrity of the election results, promoting trust in the system.

3.Reduced Costs: Electronic voting systems can potentially lower the costs associated with traditional paper-based elections, such as printing, transportation, and manual counting.

4.Accessibility: Electronic voting systems can be more accessible to remote and disabled voters, improving inclusivity in the electoral process.

5. Real-time Results: Blockchain enables real-time vote counting and result dissemination, reducing the time it takes to announce election outcomes.

6. Immutable Records: Once a vote is recorded on the blockchain, it cannot be altered, ensuring the integrity of the data.

Disadvantages:

1.Technical Challenges: Implementing and maintaining a secure electronic voting system can be complex and requires expertise in blockchain technology, cybersecurity, and software development.

2.Voter Privacy Concerns: Ensuring voter privacy while maintaining the transparency of the blockchain can be challenging. Striking the right balance is essential.

3. Digital Divide: Not all citizens may have access to the necessary technology or be comfortable using it, potentially disenfranchising certain segments of the population.

4.Vulnerability to Cyberattacks: Electronic voting systems are susceptible to cyberattacks, and a successful breach could compromise the entire election process.

5.Lack of Paper Trail: Unlike traditional paper-based systems, blockchain-based systems may lack a physical paper trail, which can make audits and recounts more challenging.

6.Initial Setup Costs: Setting up a robust blockchain-based voting system can be expensive, requiring significant investment in technology and infrastructure.

7.Trust and Adoption: Building public trust in new technology and ensuring its widespread adoption is a substantial challenge, particularly if past election systems were marred by controversies.

11.CONCLUSION:

The successful implementation of such a system requires careful planning, expertise in blockchain technology, and strong cybersecurity measures. Moreover, building public trust and ensuring widespread adoption are critical factors for its success. As technology evolves, electronic voting systems with blockchain may become more feasible and acceptable solutions for elections, but thorough testing and continuous vigilance are vital to mitigate potential risks and ensure the integrity of the democratic process.

12.FUTURE SCOPE:

The future scope of electronic voting systems based on blockchain technology is promising and includes several potential developments:

1.Wider Adoption: As trust in the security and transparency of blockchain-based voting systems grows, more countries and regions may consider adopting them for various types of elections, from local to national.

2.Interoperability:Future systems may work on developing standards and protocols that allow different blockchain-based voting systems to interoperate, improving compatibility and sharing best practices.

3.Mobile Voting:The use of mobile devices for voting may become more prevalent, making the process even more accessible and convenient for voters.

4.Biometric Authentication: Integration with biometric authentication methods, like facial recognition or fingerprint scanning, could enhance voter identity verification while maintaining privacy.

5.Advanced Security: Continual advancements in cybersecurity will lead to even more robust protection against cyberattacks and vulnerabilities in the voting system.

6.Decentralized Identity:The development of decentralized identity systems on blockchain could improve privacy and reduce the risk of personal data breaches.

7.Hybrid Systems: Future systems might combine aspects of both electronic and paper voting to provide a physical paper trail while benefiting from the advantages of blockchain technology.

8.Smart Contracts:Smart contracts could automate and facilitate various aspects of the voting process, from candidate registration to result verification.

9.Auditing and Transparency Tools:Enhanced auditing and data analysis tools may be developed to provide more in-depth insights into the voting process and results.

10.Education and Outreach: A concerted effort to educate the public about the benefits and security of blockchain-based voting systems will be crucial for their widespread adoption.

13.APPENDIX

Source code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract VoteSystem{
    address public owner;

    constructor(){
        owner= msg.sender;
    }

 struct candidate {
     uint voterId;
     string name;
     uint age;
     uint voteCount;
 }

 mapping (uint => candidate) candidateMap;

 struct voters {
     uint voterId;
     string name;
     uint age;
     bool votingState;
 }

 mapping (uint => voters) votersMap;
 mapping (uint=>bool) registeredVoter;

 modifier checkVoterVoted(uint _votersVoterId){
     require (votersMap[_votersVoterId].votingState == false);
     _;
 }

modifier checkRegisteredVoter(uint _votersVoterId){
        require(registeredVoter[_votersVoterId]==true, "Voter is not Registered");
        _;
}
uint[] voterIdlist;
uint[] candidateIdList;
```

```
 function enrollCandidate(uint _voterId,string memory _name,uint  _age )  public {

 require (_age >= 25);
 require (candidateMap[_voterId].voterId != _voterId);

   candidateMap[_voterId].voterId = _voterId;
   candidateMap[_voterId].name = _name;
   candidateMap[_voterId].age = _age;

   candidateIdList.push(_voterId);
 }

 function enrollVoter(uint _voterId,string memory _name,uint _age)  public  returns(bool){

require (_age >= 18);
require (votersMap[_voterId].voterId != _voterId);

    votersMap[_voterId].voterId = _voterId;
    votersMap[_voterId].name = _name;
    votersMap[_voterId].age = _age;

    voterIdlist.push(_voterId);
   return registeredVoter[_voterId]=true;

 }

 function getCandidateDetails(uint _voterId) view public returns(uint,string memory,uint,uint) {

    return
(candidateMap[_voterId].voterId,candidateMap[_voterId].name,candidateMap[_voterId].age,candi
dateMap[_voterId].voteCount);
 }

 function getVoterDetails(uint _voterId) view public returns (uint,string memory,uint,bool){

    return
(votersMap[_voterId].voterId,votersMap[_voterId].name,votersMap[_voterId].age,votersMap[_vot
erId].votingState);

 }

 function vote(uint _candidateVoterId,uint _votersVoterId) public
checkVoterVoted(_votersVoterId) checkRegisteredVoter(_votersVoterId) {
    candidateMap[_candidateVoterId].voteCount += 1;
    votersMap[_votersVoterId].votingState = true;
 }

 function getVotecountOf(uint _voterId) view public returns(uint){
      require(msg.sender== owner, "Only owner is allowed to Check Results");
    return candidateMap[_voterId].voteCount;
```

```
 }

 function getVoterList() view public returns (uint[] memory){

    return   voterIdlist;
    }

 function getCandidateList() view public returns(uint[] memory){


 return candidateIdList;
 }


}
```

Project demo link:

https://youtu.be/NcTcRT0vDdo

github link:

https://github.com/ranjithsiva17/Electronic-voting-system-NM2023TMID00996