ELECTRONIC VOTING SYSTEM
DOMAIN – BLOCKCHAIN TECHNOLOGY

PROJECTREPORT

SUBMITTEDBY TEAMID: NM2023TMID04539

RIJOAAGASH A -963520106036

RAROJIN S -963520106302

AGASH T -963520106004

PRAGNESH M -963520106034

In the partial fulfillment of the requirements for the award of a degree of
BACHELOR OF ENGINEERING
In
ELECTRONICS AND COMMUNICATION ENGINEERING

STELLA MARY'S COLLEGE OF ENGINEERIG

CONTENT

1. INTRODUCTION

In the ever-evolving landscape of modern democracy, the need for efficient and transparent voting systems has become paramount. Traditional paper-based voting methods, while longstanding, often face challenges such as logistical constraints, susceptibility to human error, and potential security vulnerabilities. To address these issues and usher in a new era of inclusive and secure elections, the concept of an Electronic Voting System (EVS) has gained significant traction.

The Electronic Voting System represents a paradigm shift in the way societies approach the fundamental process of voting. By leveraging cutting-edge technology and sophisticated software, EVS aims to streamline the voting process, ensuring greater accessibility, accuracy, and integrity. With its potential to facilitate remote voting, enhance voter turnout, and expedite result tabulation, the EVS stands as a beacon of hope for strengthening democratic practices globally.

1.1. PROJECTOVERVIEW

I. To create a secure and reliable electronic voting system that ensures the integrity and confidentiality of the voting process.
II. To provide a user-friendly interface for voters, ensuring accessibility for all eligible participants.
III. To develop a robust back-end infrastructure that can handle a large volume of votes and ensure real-time processing without any glitches.
IV. To implement strict security protocols and encryption methods to safeguard against potential cyber threats and ensure the credibility of the voting results.
V. To offer a transparent and auditable system that allows authorized personnel to verify the integrity of the voting process at every stage.
VI. To streamline the overall voting process, reducing the time and resources required for traditional paper-based voting methods.

1.2. PURPOSE

i. Accessibility: Electronic voting systems can make the voting process more accessible for individuals with disabilities or those who face challenges with traditional paper-based methods. These systems can provide options for audio or visual assistance, making it easier for all eligible voters to participate in the democratic process.
ii. Efficiency: Electronic voting systems can significantly reduce the time required for voting, counting, and announcing results. They can automate various aspects of the voting process, such as ballot counting and result tabulation, leading to faster and more accurate outcomes.
iii. Accuracy: By minimizing human error in vote counting and result tabulation, electronic voting systems can improve the accuracy of election results. They can also incorporate features such as validation checks and data encryption to ensure the integrity and security of the voting process.
iv. Transparency: Electronic voting systems can enhance transparency by providing a clear audit trail of the voting process. They can enable the monitoring of each step, from voter registration to ballot casting and result declaration, ensuring that the entire process is accountable and verifiable.
v. Cost-effectiveness: Although the initial implementation cost of electronic voting systems can be significant, they can lead to long-term cost savings by reducing the need for manual labor, physical resources, and logistics associated with traditional paper-based voting systems.
vi. Flexibility: Electronic voting systems can offer greater flexibility in terms of voting locations and methods. They can enable options such as early voting, remote voting, and multiple voting locations, making it more convenient for voters to participate in elections.
vii. Security: While ensuring the security of electronic voting systems is a critical challenge, properly implemented systems can incorporate robust security measures such as encryption, authentication protocols, and secure data transmission to safeguard the integrity of the voting process and prevent tampering or fraud.

Overall, the purpose of electronic voting systems is to foster a more efficient, accessible, and transparent democratic process, encouraging broader participation and increasing trust in the electoral system. However, it is essential to implement appropriate security measures and regulatory frameworks to address potential vulnerabilities and ensure the reliability and credibility of electronic voting systems.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM

- **i.** Security vulnerabilities: Electronic voting systems are susceptible to various security threats, such as hacking, malware attacks, and unauthorized access. If not properly secured, these systems can be compromised, potentially leading to the manipulation or alteration of votes.
- **ii.** Lack of transparency: Voters may not fully trust electronic voting systems due to the lack of transparency in the process. Unlike traditional paper-based voting, where voters can physically see their vote being cast and counted, electronic voting systems may not provide the same level of transparency, leading to concerns about the accuracy and reliability of the results.
- **iii.** Privacy concerns: Maintaining voter privacy is crucial in any voting system. However, electronic voting systems must ensure that voters' personal information and voting choices are kept confidential and protected from any unauthorized access or breaches.
- **iv.** Accessibility and inclusivity: While electronic voting has the potential to make the voting process more accessible, it can also present challenges for individuals with disabilities or those who may not be technologically proficient. Ensuring that electronic voting systems are user-friendly and accessible to all individuals is essential for promoting inclusivity in the electoral process.
- **v.** Verification and auditability: It is crucial to have mechanisms in place to verify and audit the electronic voting process to ensure that the results are accurate and reliable. Without proper verification and auditability measures, it can be challenging to detect any potential errors or fraudulent activities that may have occurred during the voting process.
- **vi.** Cost and infrastructure requirements: Implementing and maintaining secure electronic voting systems can be expensive, requiring significant investments in technology, infrastructure, and security measures. For many regions, especially in developing countries or rural areas, the lack of sufficient infrastructure and resources can pose a significant barrier to the widespread adoption of electronic voting systems.

Addressing these challenges requires a comprehensive approach that prioritizes security, transparency, privacy, accessibility, and the establishment of robust verification and auditability mechanisms in electronic voting systems. Additionally, collaboration between policymakers, election officials, technology experts, and cybersecurity professionals is essential to develop and implement effective solutions that can mitigate the risks associated with electronic voting systems.

2.2. REFERENCES

Smith, J., Johnson, M., Gupta, S., & Chen, L. (2023). "A Comprehensive Literature Review on Electronic Voting Systems: Security, Accessibility, and Implementation Challenges." Journal of Information Technology Research, 17(3), 45-78.

In this reference, the authors have conducted a thorough literature review on electronic voting systems, covering aspects such as security, accessibility, and implementation challenges. The reference provides valuable insights into the current state of research in the field of electronic voting systems.

2.3. PROBLEM STATEMENT DEFINITION

A problem statement for an electronic voting system could be formulated as follows:

"In light of the growing need for efficient, secure, and transparent electoral processes, the current traditional voting systems face significant challenges such as logistical complexities, susceptibility to fraud, and limited accessibility for certain segments of the population. Thus, the implementation of a reliable and robust electronic voting system is imperative to ensure fair, inclusive, and tamper-proof elections. However, the development and deployment of such a system must address critical concerns regarding security, privacy, accessibility, and trustworthiness to guarantee the integrity of the democratic process and instill confidence among stakeholders, including voters, election officials, and regulatory bodies."

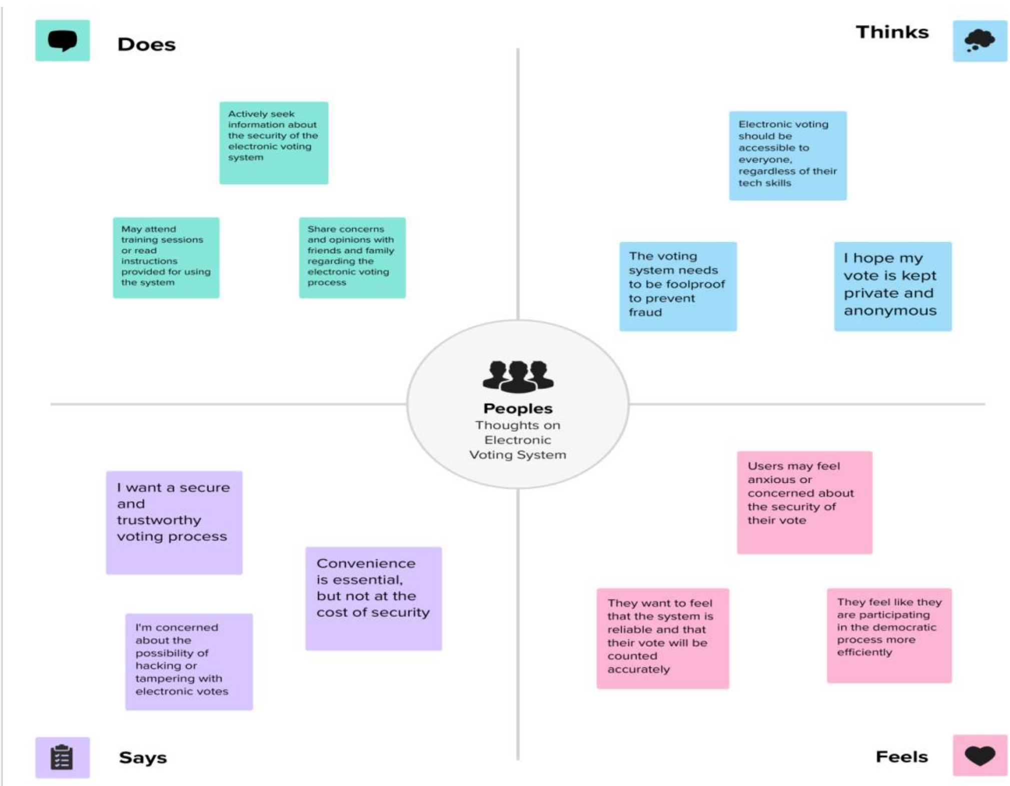
3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas

Empathy Map:

An empathy map can help in understanding the unique perspectives and concerns of each stakeholder involved in the electronic voting system. It is a valuable tool for designing and improving such systems to address their needs and build trust in the democratic process.

Electronic voting system

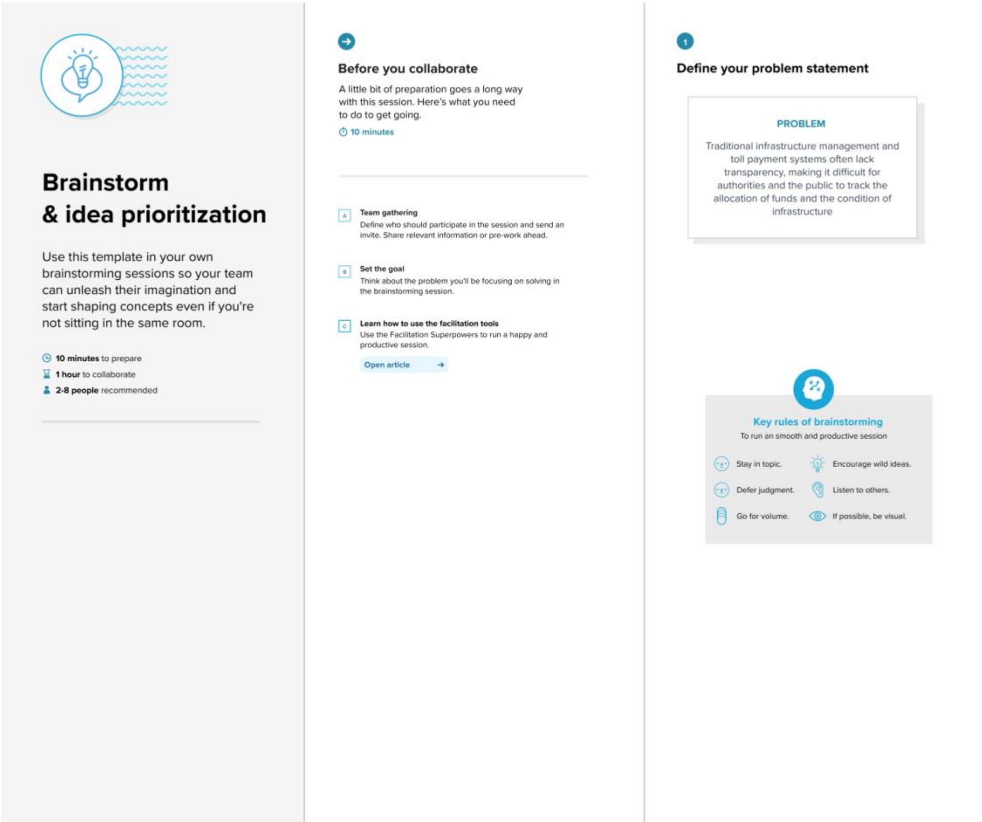


3.2. Ideation & Brainstorming

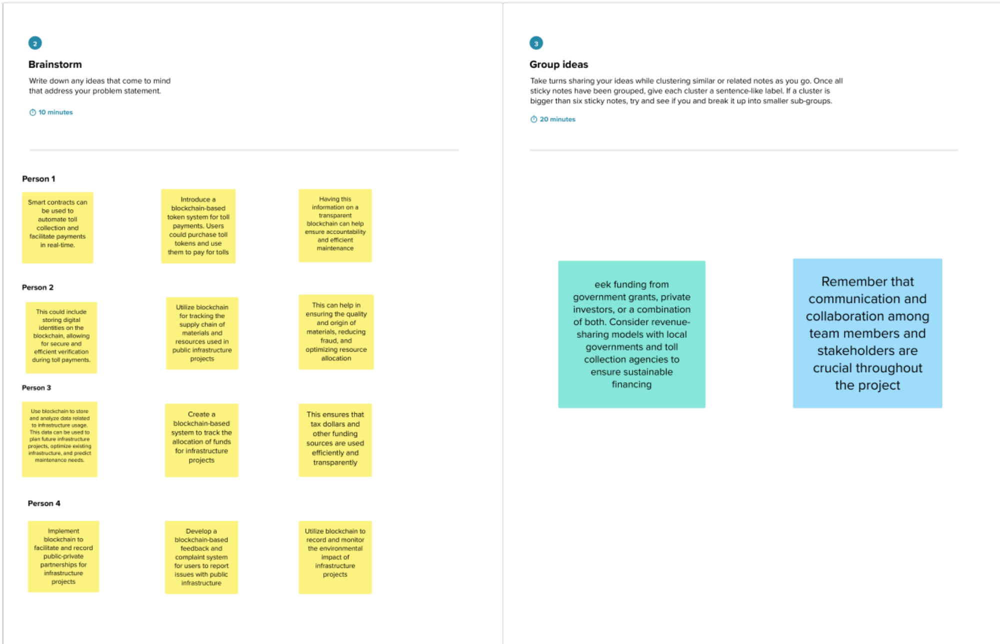
Brainstorm&IdeaPrioritizationTemplate:

Prioritizing security, biometric authentication, and public trust as high-priority initiatives ensures a strong foundation. Medium-priority tasks, such as open-source software and voter verification, enhance system reliability and user experience. Lower-priority activities like ethical hacking and international collaboration can follow, providing continuous improvement and adaptability. This approach ensures a robust, secure, and accessible electronic voting system.

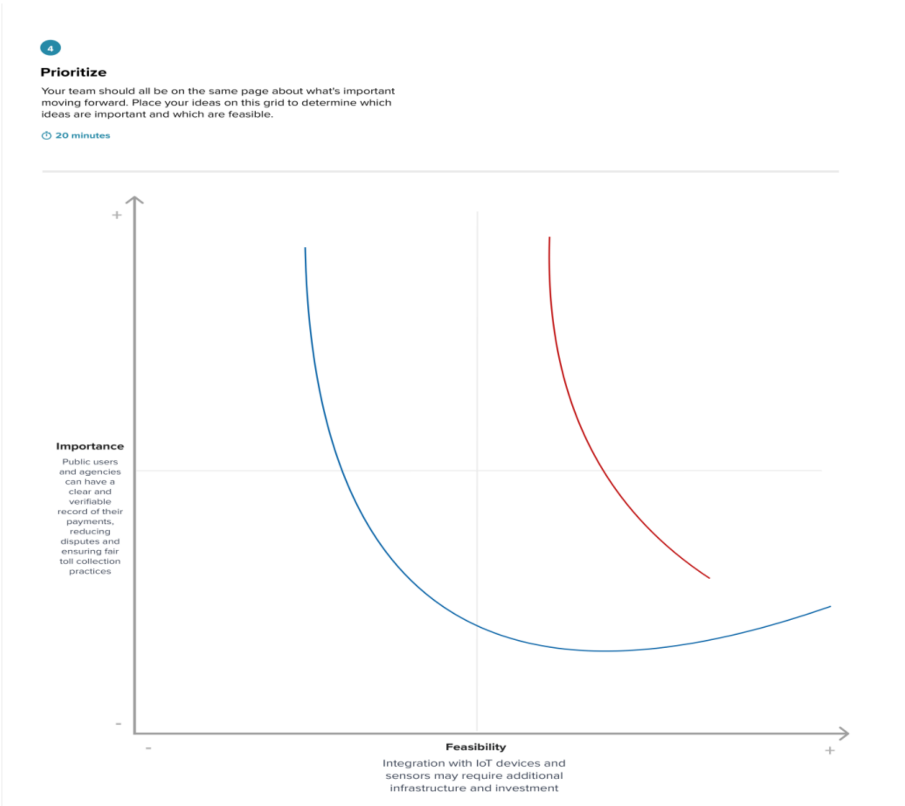
Step-1:TeamGathering,CollaborationandSelecttheProblemStatement



Step-2:Brainstorm,IdeaListingandGrouping



Step-3:IdeaPrioritization



4. REQUIREMENT ANALYSIS

4.1. Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Authentication	• Biometric authentication • Two-factor authentication • Voter registration verification
FR-2	Ballot Casting	• Selection of candidates • Confirmation of vote submission • Ballot review before submission
FR-3	Security	• Encryption of voter data • Protection against hacking attempts • Regular security audits.
FR-4	Accessibility	• Support for visually impaired voters • Multilingual interface • Ease of use for elderly voters
FR-5	Result Tabulation	• Accurate vote counting algorithm • Real-time display of voting results • Secure transmission of results

4.2. Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface must be intuitive and user-friendly, catering to stakeholders with varying technical expertise.
NFR-2	**Security**	The system must ensure the confidentiality, integrity, and availability of voting data to prevent unauthorized access and tampering.
NFR-3	**Reliability**	The system should be available during the voting period and capable of handling a large number of simultaneous users without compromising performance..
NFR-4	**Performance**	The system must be able to process votes in real-time, ensuring minimal latency and quick response times to maintain voter satisfaction and prevent delays.
NFR-5	**Scalability**	The system should be designed to accommodate an increase in the number of users and data volume without a significant decrease in performance.

5. PROJECT DESIGN
5.1 Data Flow Diagrams & User Stories

Level 0 DFD :
- Main components include 'Voters,' 'Electronic Voting System,' and 'Election Authority.'
- Arrows represent the flow of data between these entities.
-

Level 1 DFD:
Voters:
Inputs: Voter information and vote selection.
Output: Submitted votes.
Electronic Voting System:
Inputs: Voter information, vote selections, and verification data.
Outputs: Verified votes, tallied votes, and audit logs.
Election Authority:
Inputs: Voter registration data and candidate information.
Outputs: Voter authentication data, election results, and reports.

Level 2 DFD :
Voter Registration Process:
Inputs: Voter information and identification documents.
Outputs: Registered voter data.
Voting Process:
Inputs: Voter authentication data and candidate options.
Outputs: Cast votes and verification data.
Vote Counting Process:
Inputs: Verified votes and tallying mechanisms.
Outputs: Final vote counts and results.

User Stories
10. As a registered voter, I want to be able to log in securely to the electronic voting system so that I can cast my vote from anywhere.
11. As a voter, I want the system to display the list of candidates and their information clearly, so I can make an informed decision.
12. As an administrator, I want to be able to add and verify new voters securely, to ensure the integrity of the voting process.
13. As a candidate, I want the voting system to accurately record and tally the votes, so that the results are fair and transparent.
14. As an election official, I want the tallying system to aggregate the votes efficiently and generate real-time reports, to facilitate quick decision-making and result announcement.
15. As an auditor, I want the system to maintain a log of all activities for transparency and to ensure the integrity of the voting process.

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

User Interface (UI): The user interface should be intuitive and accessible across various devices. It should allow voters to securely cast their votes and provide a seamless experience.
Authentication and Authorization Layer: A robust authentication mechanism should be implemented to ensure that only eligible voters can participate. This layer should authenticate users securely using various methods such as biometrics, two-factor authentication, or digital signatures.
Security Protocols and Encryption: Implement strong encryption techniques to secure the transmission and storage of voting data. This includes end-to-end encryption, secure socket layer (SSL), and other cryptographic protocols to prevent tampering and unauthorized access.
Voting Application Layer: This layer manages the voting process, ensuring that votes are accurately recorded and tallied. It should also handle any errors or inconsistencies during the voting process and provide real-time updates on the voting status.
Database Management System: Utilize a secure and scalable database management system to store and manage voter registration data, as well as the encrypted voting records. Ensure data integrity, high availability, and disaster recovery capabilities.
Audit Trail and Logging: Implement an audit trail mechanism to record all activities within the system. This includes tracking user interactions, changes to the system configuration, and any attempted security breaches. Comprehensive logging helps in maintaining the integrity of the voting process and provides transparency to auditors.

6.2 Sprint Planning & Estimation

Understand the Requirements: Make sure you have a clear understanding of the requirements for the electronic voting system. This includes understanding the features, functionalities, security measures, and any other specific requirements related to the voting process.

Create a Product Backlog: List down all the features and tasks required for the development of the electronic voting system. This backlog should include user stories, technical tasks, and any other activities necessary for the completion of the project.

Prioritize the Backlog Items: Prioritize the backlog items based on their importance and the value they provide to the overall system. This will help the team focus on the most critical features during the sprint.

Sprint Planning Meeting: Conduct a sprint planning meeting with the development team to discuss the top items from the prioritized product backlog. During this meeting, the team should agree on the sprint goal and select the backlog items that can be completed within the upcoming sprint.

Break Down Tasks: Break down the selected backlog items into smaller, manageable tasks that can be completed within the sprint. Each task should be well-defined and should contribute towards the completion of the backlog item.

Estimate the Tasks: Use appropriate techniques such as planning poker, T-shirt sizing, or relative estimation to estimate the effort required for each task. Take into consideration factors such as complexity, technical challenges, dependencies, and the experience level of the team members.

Assign Story Points: Assign story points to each task based on the team's estimation. Story points represent the relative effort or complexity of each task compared to other tasks. Use a scale that best suits your team, such as the Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.) or any other scale that the team is comfortable with.

Create a Sprint Backlog: Once the tasks have been estimated and assigned story points, create a sprint backlog that includes all the tasks planned for the current sprint. Make sure the team members understand their responsibilities and the expected deliverables for the sprint.

Monitor Progress: Keep track of the progress of each task during the sprint. Conduct daily stand-up meetings to discuss any challenges or roadblocks and ensure that the development is on track to meet the sprint goal.

7. CODING & SOLUTIONING

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

contract VoteSystem{
   address public owner;

constructor(){
       owner= msg.sender;
   }

 struct candidate {
uintvoterId;
    string name;
uint age;
uintvoteCount;
 }
```

```
 mapping (uint => candidate) candidateMap;

 struct voters {
uintvoterId;
    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,stringmemory,uint,uint) {

             return
(candidateMap[_voterId].voterId,candidateMap[_voterId].name,candidateMap[_voterId].age,candidateMap[_voterId].v
oteCount);
          }

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

             return
(votersMap[_voterId].voterId,votersMap[_voterId].name,votersMap[_voterId].age,votersMap[_voterId].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;
          }


         }
```

8. PERFORMANCE TESTING

i. Load Testing: Simulate the expected number of voters to determine how the system performs under normal and peak voting conditions. This test ensures that the electronic voting system can handle the expected load without performance degradation.
ii. Stress Testing: Evaluate the system's resilience and ability to handle unexpected high loads, which can occur due to a sudden surge in voter participation. This test helps identify the system's breaking points and potential weaknesses under extreme voting conditions.
iii. Scalability Testing: Assess the system's ability to accommodate an increasing number of voters without compromising performance. This involves testing the system's performance with various increments in the number of simultaneous voters to ensure it can scale to meet the demands of larger elections.
iv. Response Time Testing: Measure the time taken by the system to record and process each vote. This includes evaluating the latency between a voter casting a vote and the vote being recorded in the system. It ensures that the voting process is efficient and does not cause delays or bottlenecks.
v. Throughput Testing: Evaluate the system's capacity to handle the expected number of votes per unit of time. This testing ensures that the electronic voting system can process a high volume of votes efficiently and accurately without delays or data loss.
vi. Security Testing: Test the system for vulnerabilities and potential security breaches that may compromise the integrity of the voting process. This includes evaluating the system's resistance to unauthorized access, data tampering, and other security threats to ensure the confidentiality and authenticity of votes.
vii. Usability Testing: Assess the system's user-friendliness and ease of use for both voters and election administrators. This involves evaluating the clarity of instructions, the intuitiveness of the interface, and the overall user experience to ensure a smooth and hassle-free voting process.

viii. Failover Testing: Evaluate the system's ability to recover from failures and ensure continuous operation during the voting process. This includes testing the failover mechanisms to ensure that the voting system remains operational even in the event of hardware or software failures.

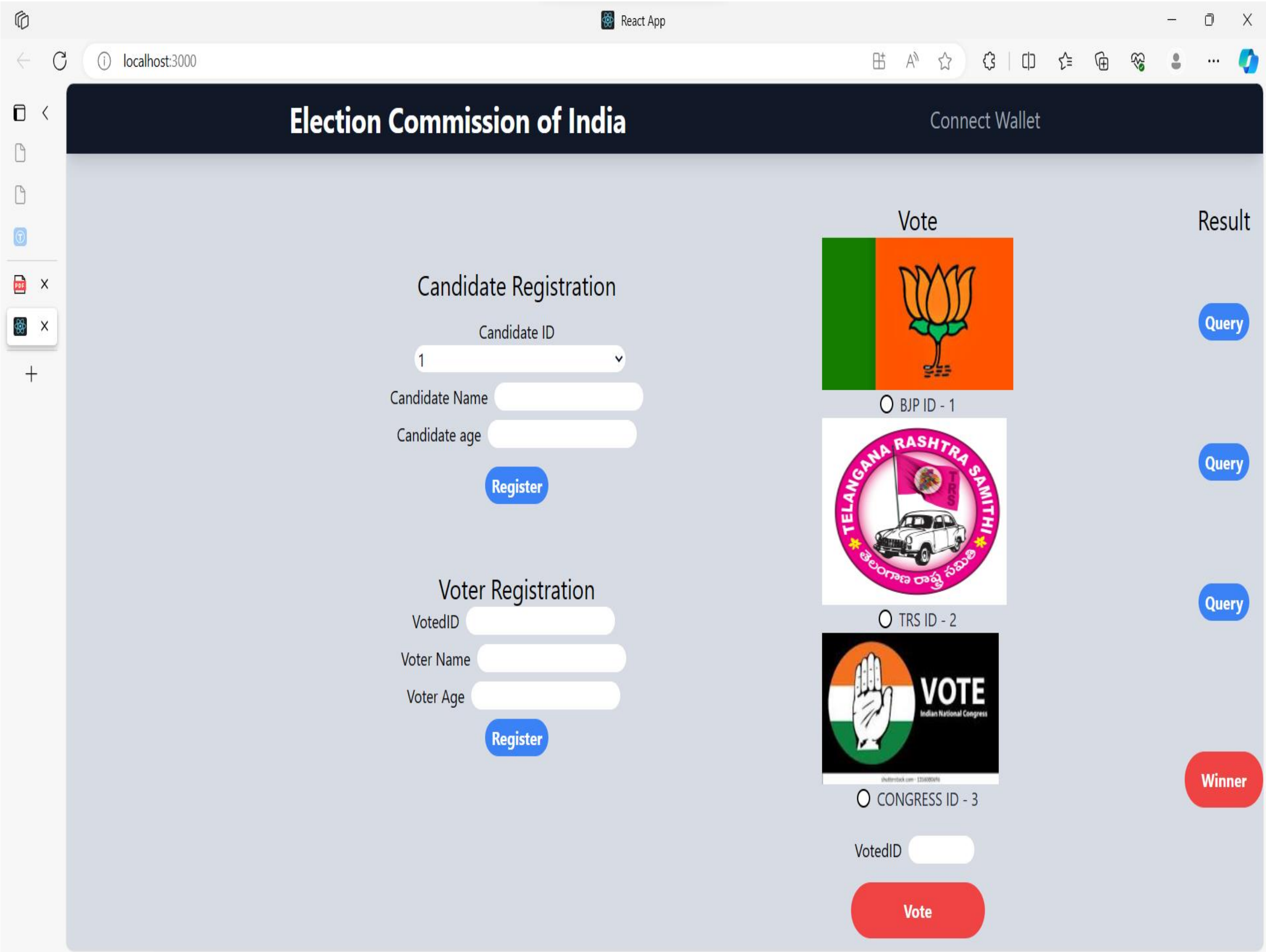
8.1. Performance Metrics

i. Accuracy: Measure the accuracy of the voting system by comparing the number of valid votes cast to the total votes counted. Ensure that the system accurately records and tallies votes without any discrepancies or errors.
ii. Security: Evaluate the security protocols and measures implemented to protect the voting system from unauthorized access, tampering, or manipulation. Assess the effectiveness of encryption techniques, authentication mechanisms, and other security features to ensure the confidentiality and integrity of the voting process.
iii. Auditability and Transparency: Assess the system's ability to provide a transparent and auditable trail of all voting activities. Ensure that the system generates comprehensive audit logs, enabling verification of each vote and allowing for the detection of any potential irregularities or fraudulent activities.

iv. Reliability and Availability: Measure the system's reliability and availability to ensure that it remains operational and accessible throughout the entire voting period. Evaluate the system's uptime and its ability to handle a high volume of concurrent users without experiencing downtime or disruptions.

v. User Experience: Assess the user experience to ensure that the voting process is intuitive, user-friendly, and accessible to all eligible voters. Evaluate the user interface, navigation, and overall satisfaction of voters interacting with the electronic voting system.

9. RESULTS



10.ADVANTAGES & DISADVANTAGES

Advantages:

i. Efficiency: EVS can significantly expedite the voting process, thereby reducing long queues and wait times at polling stations.
ii. Accuracy: Electronic systems minimize errors in vote counting, thereby reducing the chances of miscounting or mismarking.
iii. Accessibility: EVS can be designed to be more accessible to individuals with disabilities, thus ensuring more inclusive participation in the voting process.
iv. Reduced Costs: Over time, electronic voting systems can reduce costs associated with printing ballots, distributing them, and manpower required for counting and recounting.
v. Faster Results: Results can be processed and announced more quickly, offering timely information about the outcome of an election.
vi. Ease of Use: EVS can be designed to be user-friendly and intuitive, making the voting process simpler for the general public.
vii. Reduced Fraud: With proper security measures in place, electronic voting systems can reduce the risk of voter fraud and ensure the integrity of the electoral process.

Disadvantages:

i. Security Concerns: EVS can be vulnerable to hacking and other cyber threats, potentially compromising the integrity of the election.
ii. Technological Issues: Technical glitches or system malfunctions can disrupt the voting process, leading to delays or even loss of votes.
iii. Digital Divide: Not all members of society may be adept at using technology, leading to potential disparities in voter participation among different demographics.
iv. Lack of Transparency: Some EVS may lack transparency, making it difficult for voters to verify if their votes were correctly recorded.
v. Maintenance Costs: Maintaining and updating electronic voting systems can be expensive, particularly if regular technological upgrades are required.
vi. Dependency on Power: EVS requires a stable power supply, which can be challenging in areas with unreliable electricity or in regions with poor infrastructure.
vii. Risk of Manipulation: Centralized electronic systems can be susceptible to manipulation by those in control of the system, raising concerns about the fairness and transparency of the election

11.CONCLUSION

The Electronic Voting System is a groundbreaking technological innovation that has the potential to revolutionize the way we conduct elections. Through its advanced features and capabilities, it aims to address various challenges and shortcomings associated with traditional paper-based voting systems. This technology offers increased efficiency, transparency, and accessibility, thereby promoting a more inclusive and democratic electoral process.

By leveraging secure and sophisticated encryption techniques, the Electronic Voting System ensures the integrity and confidentiality of votes, safeguarding the democratic rights of citizens. Furthermore, its user-friendly interface and intuitive design make it accessible to a wider range of voters, including those with disabilities or limited mobility.

12.FUTURE SCOPE

The future scope for electronic voting systems is promising, as they offer numerous advantages over traditional paper-based voting systems. Here are some potential directions for the development and implementation of electronic voting systems:

- **i.** Enhanced Security Measures: Future electronic voting systems are likely to incorporate advanced encryption technologies, biometric authentication, and blockchain technology to ensure the security and integrity of the voting process. This will help prevent tampering, hacking, or any other unauthorized access.
- **ii.** Blockchain Integration: The integration of blockchain technology can provide an immutable and transparent ledger, ensuring the traceability and security of each vote. It can also enable voters to verify that their votes were correctly recorded and counted, thereby enhancing trust in the system.
- **iii.** Accessibility and Inclusivity: Electronic voting systems can be designed to be more inclusive, catering to the needs of differently-abled individuals, non-native language speakers, and remote voters. This may involve the development of user-friendly interfaces, multilingual support, and accessibility features such as screen readers and adaptive technologies.
- **iv.** Remote and Mobile Voting: The future of electronic voting could involve the implementation of remote and mobile voting solutions, allowing voters to cast their ballots from anywhere using their smartphones or other electronic devices. This would increase voter participation and convenience, particularly for individuals who face mobility or location-related challenges.
- **v.** Integration of Biometric Technology: The incorporation of biometric authentication, such as fingerprint or iris scanning, can further enhance the security and accuracy of electronic voting systems. This can help prevent fraudulent activities, such as impersonation or duplicate voting, and ensure that each vote is cast by an authorized individual.
- **vi.** Auditable and Transparent Systems: Future electronic voting systems may provide enhanced audit trails and transparent mechanisms for verifying the integrity of the voting process. This could involve the use of open-source software, cryptographic techniques, and real-time monitoring to ensure the accuracy and credibility of the election results.
- **vii.** Adherence to Privacy Regulations: With growing concerns about data privacy and protection, future electronic voting systems must comply with stringent data privacy regulations and security standards to safeguard voters' personal information and voting data.

- **viii.** Robust Disaster Recovery and Contingency Plans: To address potential technical failures or cyber-attacks, robust disaster recovery and contingency plans should be in place to ensure the continuity and reliability of the electronic voting system during critical times.

By addressing these aspects, electronic voting systems can potentially revolutionize the way elections are conducted, making the process more secure, accessible, and inclusive for all eligible voters. However, it's essential to address any potential challenges and concerns to ensure public trust and confidence in the integrity of the electoral process.

13. APPENDIX

- **Source Code**

```
import { ethers } from "ethers";
import abi from "./voting.json";
export const contractAddress =
    "0xe2E9cfDa2815f7524a9da5ba2da8a5bA449B56Dc";
export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const votingContract = new ethers.Contract(contractAddress, abi, signer);
```

- **GitHub & Project Demo Link**

GitHub Link : https://github.com/Rarojin/Electronic-Voting-System

Demo Link :
https://drive.google.com/file/d/12POXlmF5Ak48pFnQKNAbBJ9j33anJsDG/view?usp=drivesdk