BLOCKCHAIN PROJECT

ELECTRONIC VOTING SYSTEM

The business problem in Electronic Voting Systems (EVS) is the need to ensure secure, reliable, and tamper-proof digital voting processes that maintain the integrity of elections while providing accessible and convenient voting options for citizens. This includes addressing issues related to voter identity verification, data security, fraud prevention, and ensuring equitable access to voting for all eligible individuals.

TEAM MEMBERS:

MEMBER 1(TL): RAJALAKSHMI M **(NM ID:** FFE09D11E50A888A0584481375D108E7)

MEMBER 2 : NANDHINI K.S **(NM ID:** EE383BFE44956B986174B8B691946A96)

MEMBER 3 : KANMANI S **(NM ID:** B57042E60B3423085CC4B15E3D2639F7)

MEMBER 4 : SETHANA SHREE S **(NM ID:** F26AD9635476D3DA7F14F8FD78D4ED0E)

NM TEAM ID : NM2023TMID00855

COLLEGE NAME: UNIVERSITY COLLEGE OF ENGINEERING KANCHEEPURAM

ELECTRONIC VOTING SYSTEM

1. INTRODUCTION:

India is a democratic country. As now all Indian citizen become a part of the growing digital India with a digital ID that is Aadhaar card. Voting schemes have evolved from counting hands in early days to systems that include paper, punch card and electronic voting machine.

The traditional methods of voting have long been the cornerstone of democratic societies around the world. However, these methods often face challenges such as voter fraud, security concerns, and logistical inefficiencies. In recent years, electronic voting systems have emerged as a potential solution to address these issues.

Among the various technological innovations that promise to revolutionize voting, blockchain technology has gained significant attention for its potential to enhance the transparency, security, and trustworthiness of electronic voting systems.

1.1 Project overview:

- ✓ The "Blockchain Electronic Voting System" project is geared towards the development of a cutting-edge and secure platform for conducting elections using blockchain technology.
- ✓ Our primary goals encompass creating a user-friendly electronic voting interface, implementing a robust blockchain infrastructure for security, enabling remote and electronic voting for enhanced accessibility, establishing rigorous identity verification methods to combat fraudulent voting, and providing real-time access to the blockchain ledger for transparency.

- ✓ Rigorous testing will be undertaken to ensure the system's reliability and security. This project is poised to revolutionize the electoral landscape by increasing voter participation, reducing the risk of election fraud, streamlining election management, and fostering greater trust in the democratic process.

- ✓ It makes sure your vote can't be messed with and is easy to check. This means you can vote from your computer or phone, which is really convenient. It's like a high-tech upgrade for voting that can make elections more trustworthy and could get more people to vote. But we need to be careful about making sure everyone can use it and that only eligible voters can participate.

1.2 Purpose ：

The purpose of a blockchain electronic voting system is to make the voting process more secure, convenient, and transparent. It aims to:

1) Identity Verification: To participate in electronic voting, citizens must first establish their identity securely. Blockchain-based identity verification systems can ensure that only eligible voters are granted access, reducing the risk of voter impersonation.

2) Casting Votes: Once a voter's identity is verified, they can cast their vote electronically. These votes are recorded as transactions on the blockchain, providing a transparent and auditable trail of votes.

3) Security: Blockchain's decentralized nature and cryptographic features make it extremely resistant to hacking and fraud. Votes recorded on the blockchain are immutable, and any attempts to alter the data would be immediately detected.

4) Transparency: All transactions on the blockchain are visible to anyone, providing transparency in the voting process. Voters can verify that their votes were counted correctly, and this transparency can increase trust in the electoral system.

5) Decentralization: Decentralized blockchain networks distribute the voting data across multiple nodes, reducing the risk of a single point of failure or manipulation.

2. LITERATURE SURVEY:

Currently increasing digital technology helped many people lives. In contrast to the electoral system, there are many conventional uses of paper in its implementation. The aspect of security and transparency is a threat from still widespread election with the conventional system .Block chain technology is one of solutions, because it embraces a decentralized system and the entire database are owned by many users. There is no doubt that the revolutionary concept of the blockchain, which is the underlying technology behind the famous crypto currency Bit coin and its successors, is triggering the start of a new era in the Internet and the online services. In this work, we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language.

E-voting is a potential solution to the lack of interest in voting amongst the young tech savvy population. For e-voting to become more open, transparent, and independently auditable, a potential solution would be base it on block chain technology. Block chain technology has a lot of promise; however, in its current state it might not reach its full potential.

2.1 EXISTING SYSTEM

The Existing System of Election is running manually. The Voter has to Visit to Booths to Vote a Candidate so there is wastage of Time. Due to this many people don't go out to cast their vote which is one of the most important and Worrying factor. In democracy Each and every vote is important. This Traditional system can be replaced by a new online system which will limit the voting frauds and make the voting as well as counting more efficient and transparent.

PROPOSED SYSTEM:

The current voting system requires some improvement in it because of the issues mentioned above. This can be achieved by replacing the existing system by the new system which will limit the voting frauds and make the voting as well as counting more efficient.

- Online Election System would have user registration, user login and admin login.
- This Online Voting System will manage the Voter's information by which voter can login and use his voting rights.
- At the time of registration voter will be asked for this: Full name, age, Aadhaar card no, mobile no. email id and after being verified will be given the access.
- At the time of requesting vote, voter will be asked to enter his Aadhaar id. Then voter will be authenticated, and he can give vote from one of the candidate from the list .Voters can vote for a Candidate only once per Election.
- The software system allows the user to login in to their profiles and upload all their details including their previous milestone onto the system. The admin can check each Candidate details.
- The software system also allows Voters to view a list of Candidates in their area. The admin has overall rights over the system and can moderate and delete any details not pertaining to Election Rules

2.2 REFERENCES:

Coursera. [Online].

Available: *https://www.coursera.org/specializations/blockchaina*

[2] Edureka (How Blockchain Works) - Simply Explained. [Online].

Available: *https://youtu.be/9qfxLo1rt1Q?list=PL9ooVrP1hQOFJblZm3OdcVVH*

6Z8V7HP1

[3] Introduction to Blockchain by NPTEL IITM. [Online]

Available: *https://youtu.be/mzPoUjQC4WU*

[4] Blockchain in e-voting. [Online].

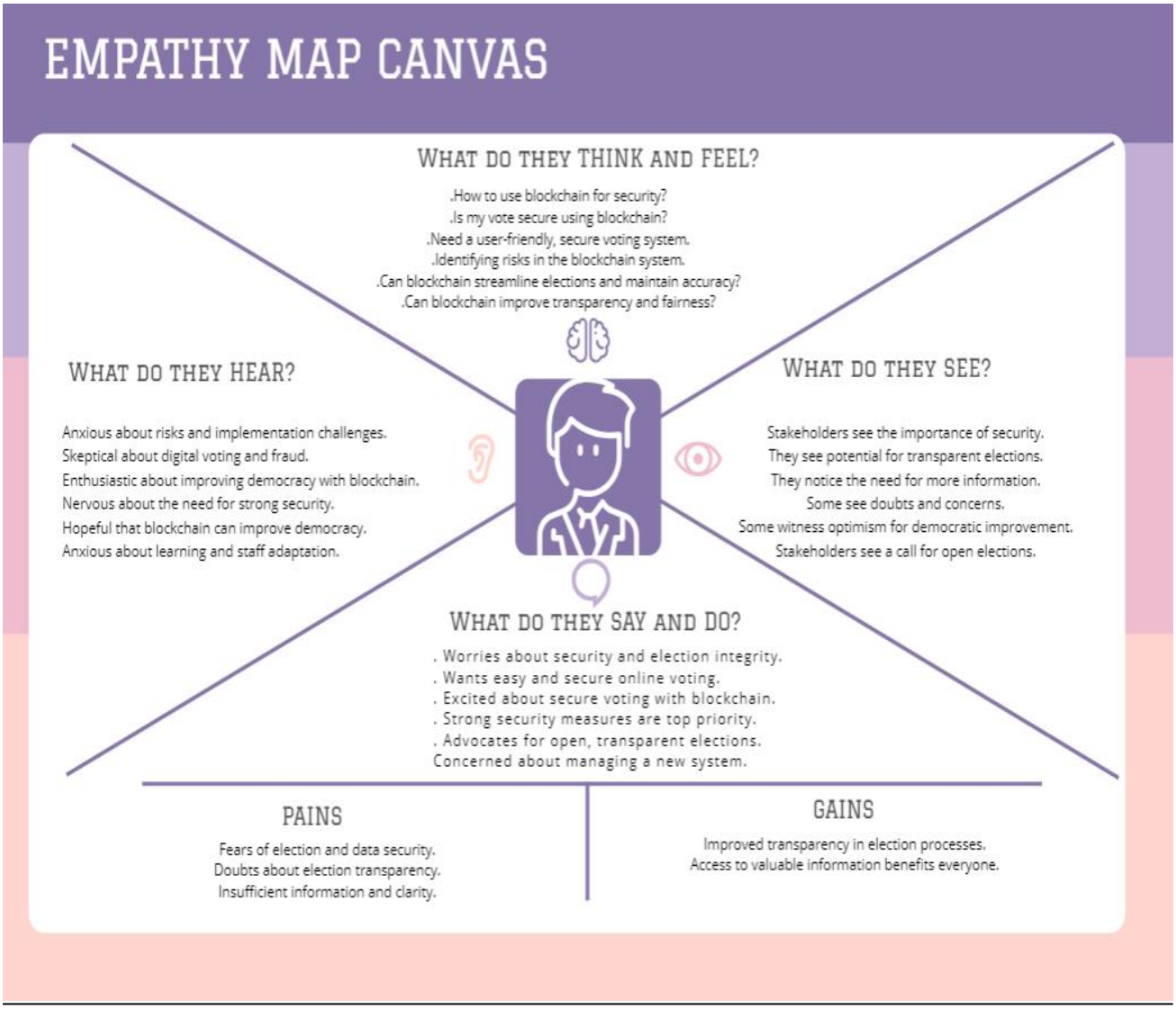
Available: https://www.youtube.com/watch? v=d0iLN8LDJ8g&feature=youtu.be

2.3 Problem Statement Definition:

The business problem in Electronic Voting Systems (EVS) is the need to ensure secure, reliable, and tamper-proof digital voting processes that maintain the integrity of elections while providing accessible and convenient voting options for citizens. This includes addressing issues related to voter identity verification, data security, fraud prevention, and ensuring equitable access to voting for all eligible individuals. Solving this problem requires robust cybersecurity measures, advanced authentication methods, and user-friendly interfaces to maintain trust in the democratic process.

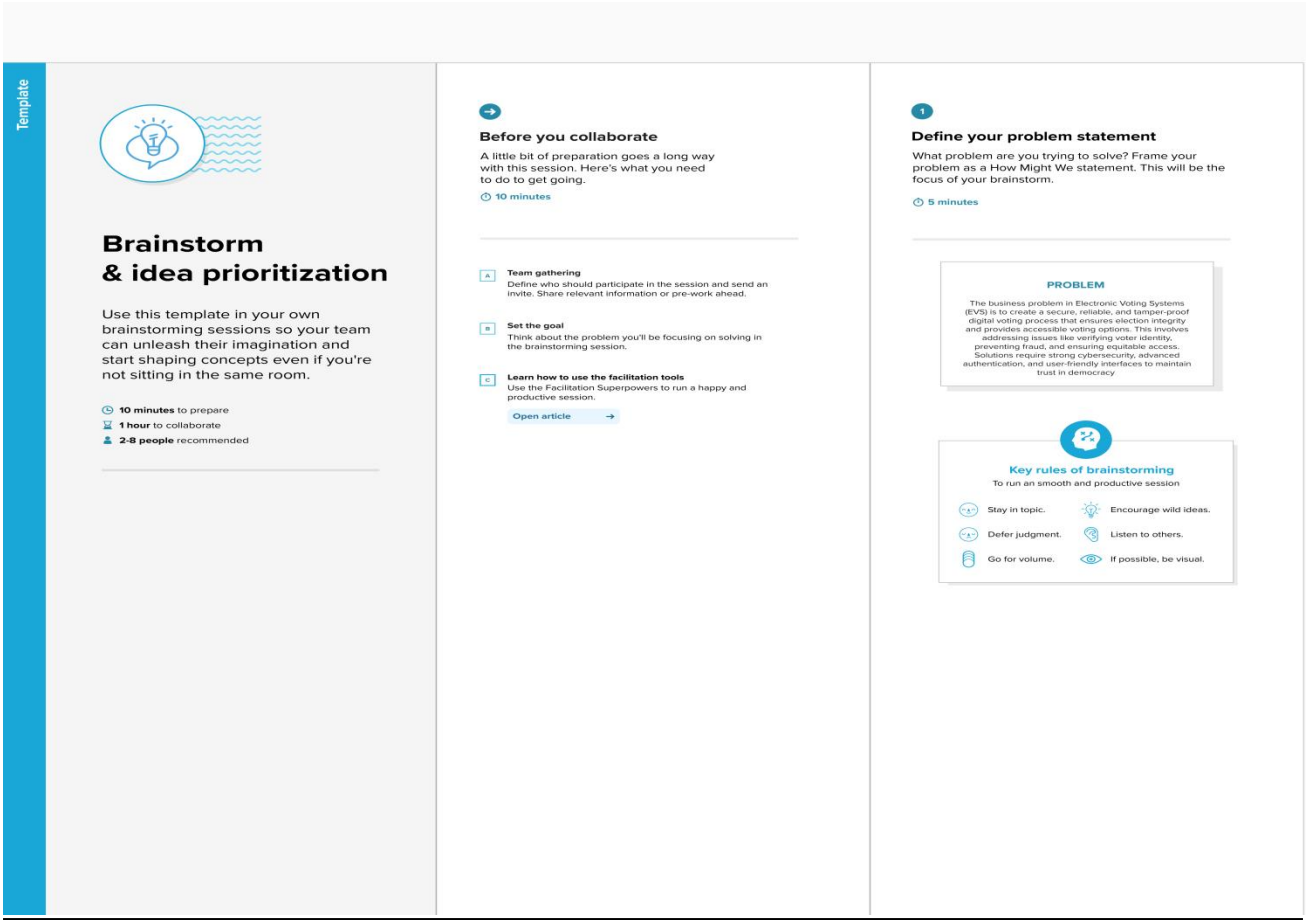
3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS

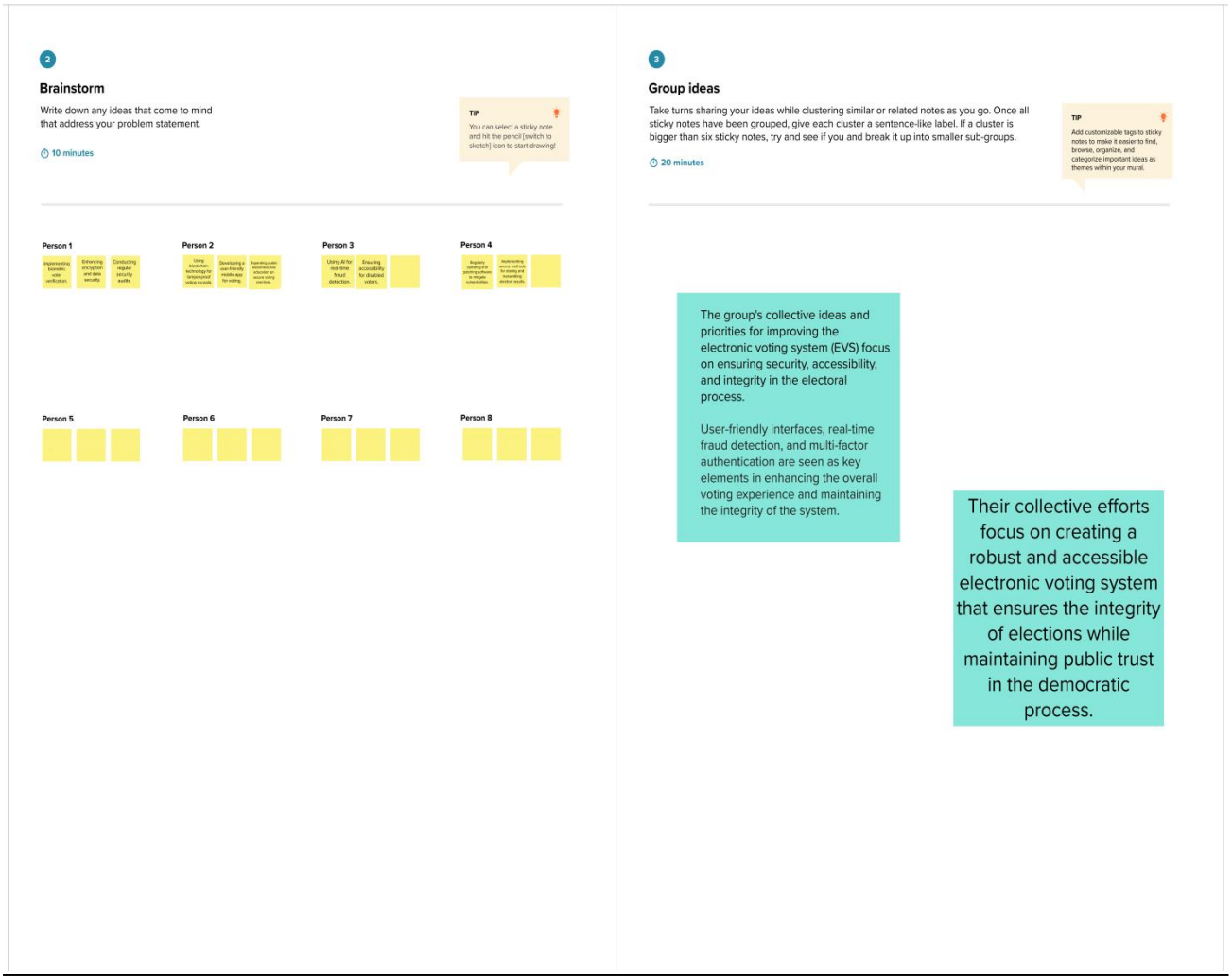


3.2 IDEATION AND BRAINSTORMING:

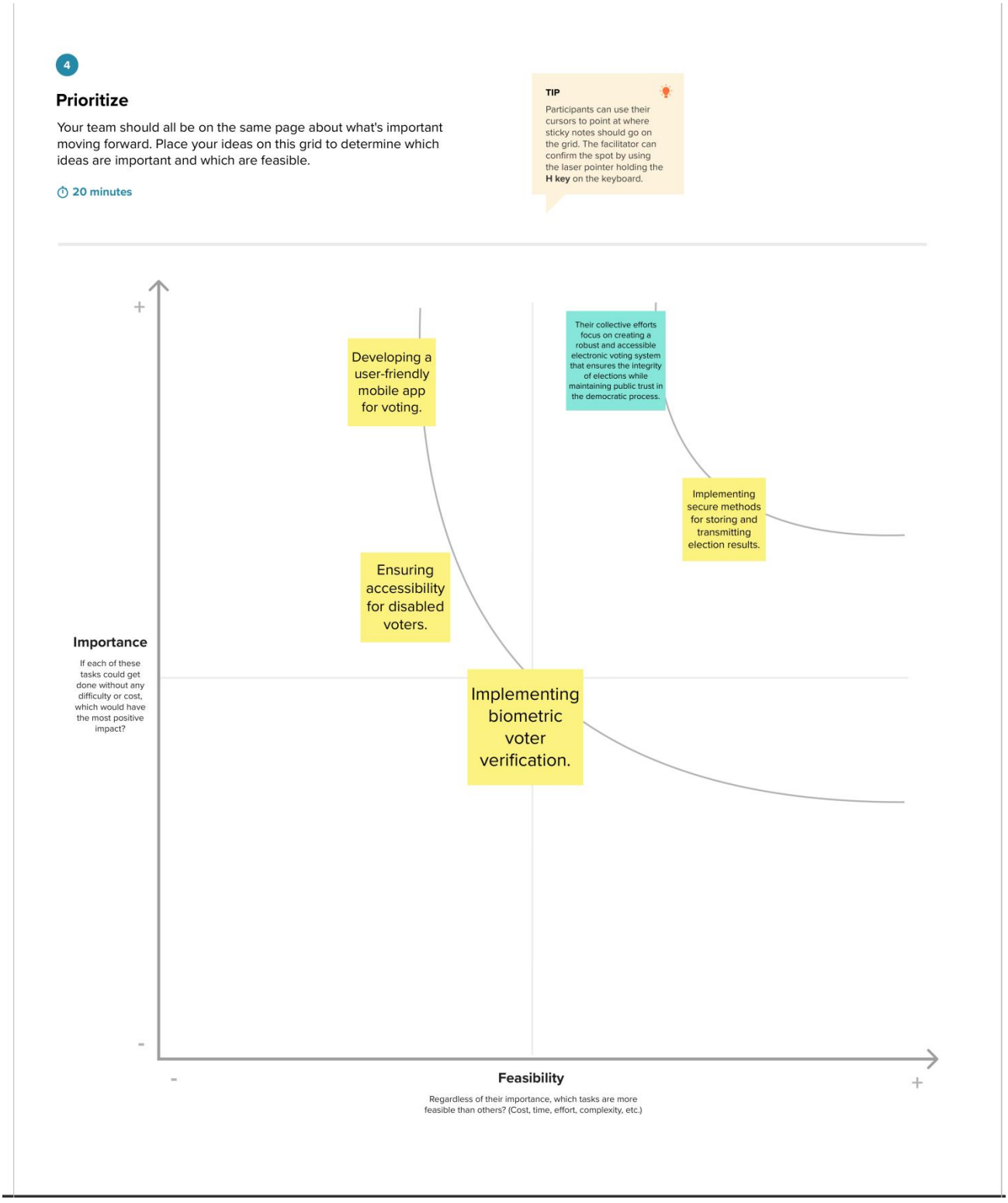
STEP 1:



STEP 2:


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
Person 1
Person 2
Person 3
Person 4
Person 5
Person 6
Person 7
Person 8
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
The group's collective ideas and priorities for improving the electronic voting system (EVS) focus on ensuring security, accessibility, and integrity in the electoral process.
User-friendly interfaces, real-time fraud detection, and multi-factor authentication are seen as key elements in enhancing the overall voting experience and maintaining the integrity of the system.
Their collective efforts focus on creating a robust and accessible electronic voting system that ensures the integrity of elections while maintaining public trust in the democratic process.

STEP 3:


4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
+
Developing a user-friendly mobile app for voting.
Their collective efforts focus on creating a robust and accessible electronic voting system that ensures the integrity of elections while maintaining public trust in the democratic process.
Implementing secure methods for storing and transmitting election results.
Ensuring accessibility for disabled voters.
Implementing biometric voter verification.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
-
-
Feasibility
+
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Business Requirements for An Electronic Voting System (EVS) Using Blockchain Technology Should Focus on Key Objectives. Here Are the Requirements In A Concise Format:

Security: Ensure End-To-End Data Security and Voter Anonymity Through Cryptographic Techniques.

Transparency: Guarantee Transparency in The Voting Process, With Auditable and Immutable Records on The Blockchain.

Accessibility: Facilitate Convenient and Inclusive Voting for All Eligible Citizens, Regardless of Location or Physical Abilities.

User Registration and Authentication:

The system must allow eligible voters to register using their Ethereum wallet addresses via Metamask. It should authenticate voters through Metamask to ensure secure identity verification.

Secure Voting Process:

The EVS should enable users to cast their votes securely and anonymously using their Ethereum wallet. Votes should be stored as transactions on the Ethereum blockchain, ensuring transparency.

Real-Time Vote Tracking and Verification:

The system must provide real-time tracking of votes cast, allowing voters to confirm the status of their votes. Voters should have the ability to verify their votes on the blockchain to ensure authenticity.

User-Friendly Voting Interface and Accessibility:

Design and implement a user-friendly voting application that is easy to use for all voters. Ensure accessibility features for individuals with disabilities, such as screen reader compatibility and optimized user interfaces.

Security and Audit Trail:

Implement robust security measures to protect the system against vulnerabilities and maintain the integrity of the voting process. Develop and maintain a blockchain-based audit trail for all votes to ensure transparency and traceability in the election process.

Non-Functional Requirements For Your Electronic Voting System (EVS) Project:

Performance:

The system must provide high performance, with low latency and the ability to handle a large number of concurrent users during peak voting periods.

Security:

Data on the blockchain must be secure, with strong encryption and protection against unauthorized access, tampering, and fraud.

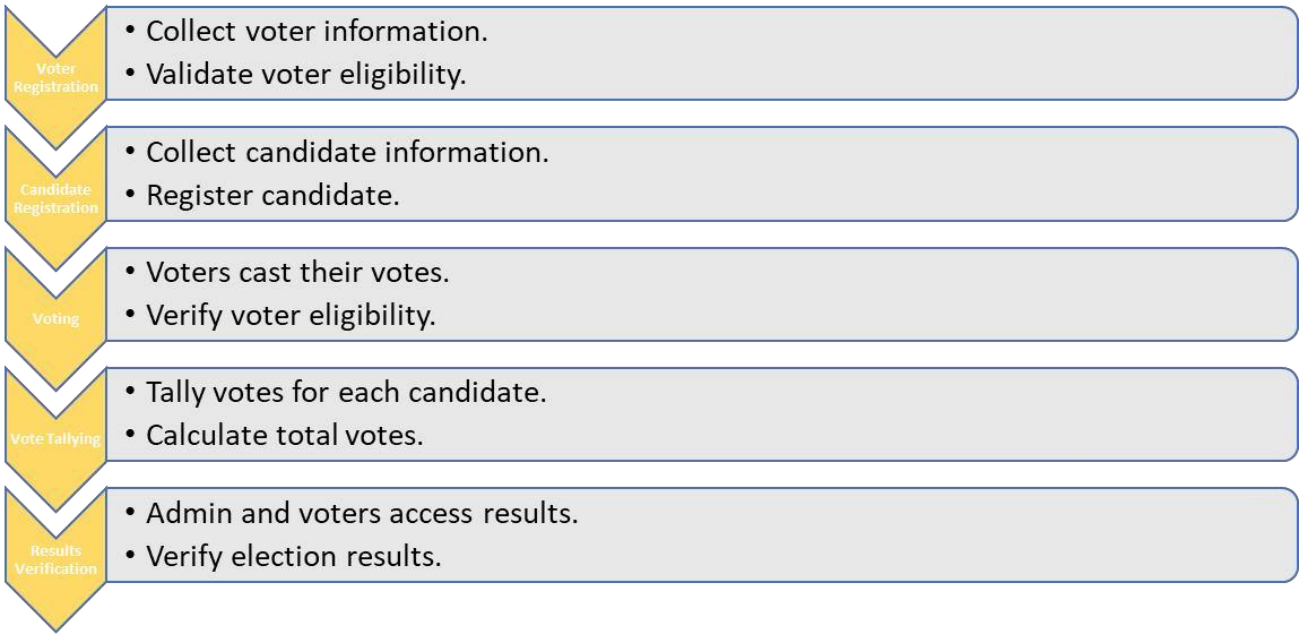
Reliability and Availability:

The system should be highly reliable and available during the voting period, with minimal downtime or service interruptions. It should have contingency plans in place.

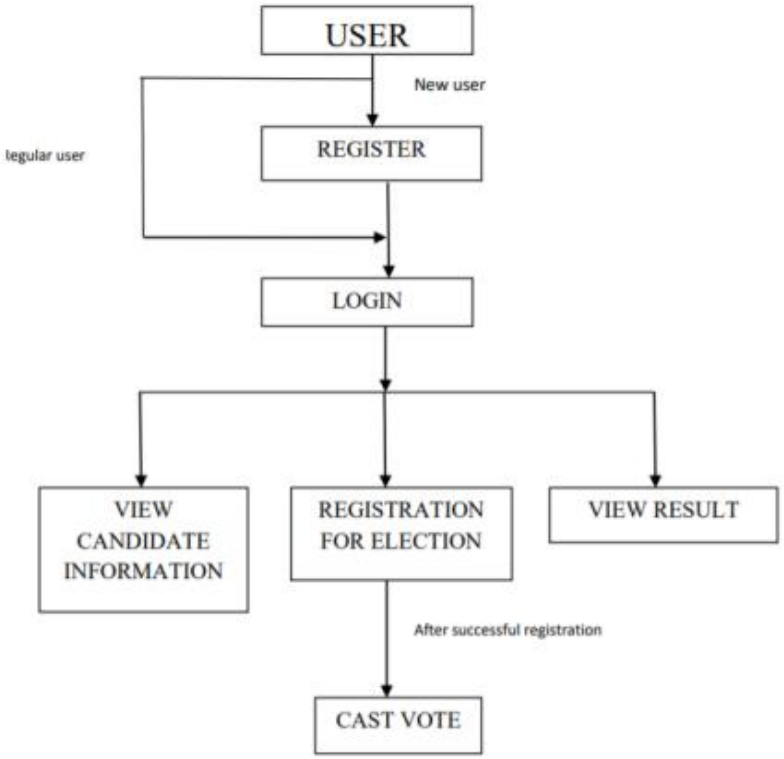
Compliance: The EVS must adhere to legal and regulatory requirements for elections and data privacy, ensuring fairness and accuracy. It should also comply with established election standards and practices.

5.PROJECT DESIGN:

5.1 Data Flow Diagrams & User Stories



5.2 Solution Architecture



6.PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture:

- Define the technical architecture for the blockchain-based electronic voting system.
- Select an appropriate blockchain platform such as Ethereum or Hyperledger.
- Specify the data storage, encryption, and security measures to ensure the integrity and confidentiality of votes. Plan for scalability to accommodate a growing number of voters and adaptability for potential system updates.
- Implement protocols for identity verification and access control to prevent fraudulent voting.

6.2 Sprint Planning & Estimation

Sprint Number	Sprint Focus	Duration (Weeks)	Goals and Tasks
Sprint 1	Blockchain Architecture Design	2	- Research and design blockchain architecture. Define data structures and encryption methods. Select a suitable blockchain platform.
Sprint 2	Voter Registration & Identity Verification	3	- Develop smart contracts for voter registration. Implement identity verification mechanisms. Create a user-friendly registration interface.
Sprint 3	Secure Voting Process	3	-Integrate blockchain into the voting process. Develop smart contracts for vote casting and recording. Implement real-time vote tracking and verification.
Sprint 4	Ethereum Setup and Smart Contract Design	2	- Set up Ethereum development environment with Visual Studio Code. Design the core smart contracts for secure voting on the blockchain.
Sprint 5	MetaMask Integration and Identity Verification	3	- Integrate MetaMask for Ethereum wallet functionality in the voting application. Develop the user interface for voter registration, interacting with MetaMask.
Sprint 6	Deployment & Final Optimization	2	- Deploy and Conduct final testing, validation, and performance optimization. Prepare for live implementation and ongoing system monitoring.

6.3 SPRINT DELIVERY SCHEDULE:

Sprint Number	Sprint Focus
Sprint 1	Blockchain Architecture Design
Sprint 2	Voter Registration on Blockchain
Sprint 3	Voting Process Integration
Sprint 4	Blockchain-Based Verification
Sprint 5	User-Friendly Interface
Sprint 6	Security and Privacy Features
Sprint 7	Accessibility Enhancements
Sprint 8	Testing and Quality Assurance
Sprint 9	Deployment and Final Testing

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature: Blockchain Integration for Security

Description:

Blockchain integration enhances the security and transparency of the electronic voting system by recording votes in a tamper-proof ledger.

Smart Contract Features:

Owner Privileges: The contract allows an owner to have special privileges, such as checking the results, making the contract more secure and controlling access.

Candidate and Voter Registration: The contract allows the registration of both candidates and voters. Candidates need to be at least 25 years old, while voters need to be at least 18 years old to register.

Enrollment of Candidates: Candidates can enroll by providing their voter ID, name, and age. This data is stored in the candidateMap and allows for tracking candidate details.

Enrollment of Voters: Voters can enroll by providing their voter ID, name, and age. Their registration status is stored in the registeredVoter mapping, which helps in checking whether a voter is registered.

Voting Mechanism: The contract enables registered voters to cast their votes. It checks if the voter has not voted before and is a registered voter before allowing them to vote. Vote Counting: When a voter casts their vote for a candidate, the candidate's voteCount is incremented, and the voter's votingState is set to true to prevent multiple votes.

Access Control: The owner has the exclusive right to check the results, as indicated by the require statement in the getVotecountOf function.

FEATURE 2:

Accessibility and Ethereum Smart Contract Interaction:

Offers a user-friendly interface to make the voting process accessible to all eligible voters, including those with disabilities.

Provides options for remote or in-person voting to accommodate diverse voter needs.

Description:

This code facilitates interaction with an Ethereum smart contract for a voting system. It connects to the contract, retrieves the contract's ABI (Application Binary Interface) from a file named "voting.json," and uses the Ethereum provider and signer for managing interactions with the contract.

8.PERFORMANCE TESTING:

8.1 PERFORMANCE METRICS:

Transaction Speed: Measure how quickly the system processes votes. Faster transaction speeds ensure that votes are counted in a timely manner.

Error Rate: Keep track of the number of errors or failed transactions. A low error rate indicates a more reliable voting system.

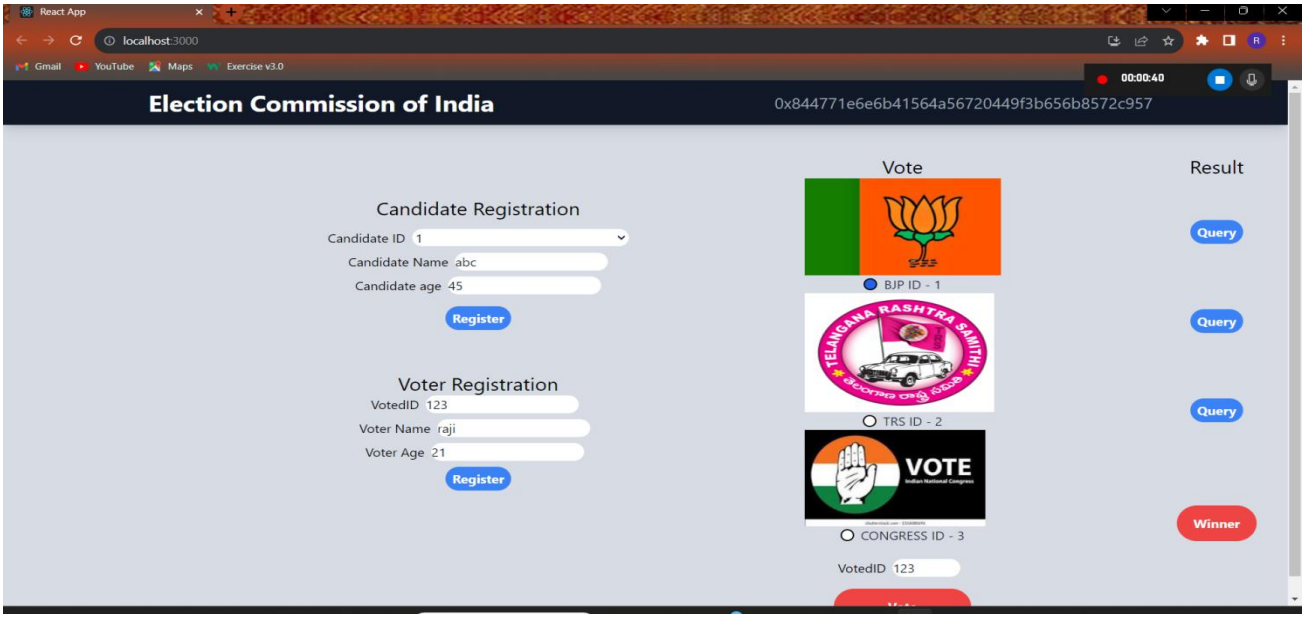
Resource Usage: Monitor the utilization of resources (e.g., server resources, blockchain capacity) to ensure the system can handle high demand without slowing down.

Security Audits: Regularly review the system for vulnerabilities and security threats to maintain the integrity of the voting process.

User Feedback: Collect feedback from voters about the system's ease of use and responsiveness to improve the user experience.

RESULTS:

9.1 OUTPUT SCREENSHOTS



10.ADVANTAGES:

Enhanced Security: Blockchain's decentralized and immutable nature makes it extremely secure. Once a vote is recorded, it is nearly impossible to alter or tamper with the data, reducing the risk of fraud.

Transparency: The open and transparent nature of blockchain allows anyone to independently verify the validity of election results. This promotes trust in the electoral process.

Accessibility: Electronic voting systems can be accessible from any location with an internet connection, making it easier for citizens to vote. This is particularly beneficial for people with mobility or transportation challenges.

Efficiency: Digital voting can streamline the voting process, reducing administrative overhead and potentially minimizing long queues at polling station.

DISADVANTAGES:

Complex Implementation: Implementing a blockchain-based electronic voting system is complex and requires a skilled technical team. It may be challenging for some regions with limited resources.

Voter Authentication: Ensuring the identity of voters in an online environment can be challenging. Reliable methods of identity verification are essential to prevent fraud.

System Failures: Technical issues, including server failures or connectivity problems, could disrupt the voting process and erode confidence in the system.

Public Trust: Some people may be skeptical about the reliability of electronic voting systems, potentially eroding trust in the electoral process.

11 . CONCLUSION

In this project, we introduced a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient election while guaranteeing voters privacy. Blockchain technology offers a new possibility to overcome the limitations and adoption barriers of electronic voting systems which ensures the election security and integrity and lays the ground for transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain.

To achieve this, we have divided this overall work into three modules.

1. The front-end module
2. The back-end module
3. Connection of two different module along with the testing.

We have also done some research work related to the implementation of Blockchain in decentralized application. And in the future to make the voting process more secure and to correctly identify the person who is voting we can use ML and AI concepts. Using these concepts, we can verify whether the person voting is the same as the person who has registered during the registration process.

12. FUTURE SCOPE

The future scope of this project, developing an electronic voting system with blockchain, promises heightened security, global adoption, and inclusivity. Advancements in blockchain security will ensure the integrity of the voting process. Bridging the digital divide aims to make electronic voting universally accessible. Smart contracts and robust identity verification will streamline voting and enhance privacy. Real-time auditing fosters trust and transparency.

Decentralized decision-making could extend to various sectors beyond elections. Cost reductions and international collaboration may promote wider adoption. Ongoing research and education efforts will address challenges. This project envisions a future of secure, transparent, and globally accessible electronic voting systems, transforming democratic processes.

13. APPEDIX

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,
candidateMap[_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[
_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;

 }

}
```

GITHUB LINK:

https://github.com/RAJALAKSHMI-07/Blockchain-Electronic-Voting-System-.git

PROJECT DEMO LINK:

https://drive.google.com/file/d/1b256XdHSeAI9emHQhuPPfZmVtnKKOKvb/view?usp=drivesdk