Naan Mudhalvan
Project Report

Electronic Voting System

Submitted By

Team ID	- NM2023TMID09491
Team Leader	- AKASH RAJ K
Team Member 1	- LOGESH S
Team Member 2	- AFRIN S
Team Member 3	- ARAVINDAN P

Index

1. **INTRODUCTION**
Project Overview
Purpose

2. **LITERATURE SURVEY**
Existing Problem
References
Problem Statement Definition

3. **IDEATION & PROPOSED SOLUTION**
Empathy Map Canvas
Ideation & Brainstorming

4. **REQUIREMENT ANALYSIS**
Functional Requirement
Non-Functional Requirement

5. **PROJECT DESIGN**
Data Flow Diagrams & User Stories
Solution Architecture

6. **PROJECT PLANNING & SCHEDULING**
Technical Architecture
Sprint Planning & Estimation
Sprint Delivery Schedule

7. **CODING & SOLUTIONING**
Feature 1
Feature 2

8. **PERFORMANCE TESTING**
Performance Metrics

9. **RESULTS**
Output Screenshots

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1 .INTRODUCTION

Project Overview

The objectives of the project "Electronic Voting System Using Blockchain" are to:

- Design and develop a secure and transparent electronic voting system using blockchain technology.
- Address the limitations of existing electronic voting systems, such as vulnerability to hacking and fraud.
- Increase voter participation and trust in the electoral process.

Need for a Secure and Transparent Electronic Voting System

Traditional paper-based voting systems are vulnerable to a number of problems, including:

- Fraud: Votes can be tampered with or forged, and ballot boxes can be stuffed.
- Inaccuracy: Human errors can occur in counting ballots, leading to incorrect results.
- Long lines and wait times: Voting can be time-consuming and inconvenient, especially for people with disabilities or those who live in rural areas.
- Low voter turnout: Many people are discouraged from voting due to the inconvenience or lack of trust in the system.

An electronic voting system using blockchain technology can address all of these problems. Blockchain is a distributed ledger technology that is secure, transparent, and tamper-proof. It can be used to create a voting system where every vote is recorded accurately and securely, and the results are transparent and verifiable by all stakeholders.

Benefits of Blockchain-Based Electronic Voting

Blockchain-based electronic voting offers a number of benefits over traditional paper-based voting systems, including:

- Security: Blockchain is highly resistant to hacking and fraud. Once a vote is recorded on the blockchain, it cannot be altered or deleted.
- Transparency: The blockchain is a public ledger, so all voters can see the results of the election in real time. This helps to ensure that the election is fair and impartial.
- Convenience: Voters can cast their ballots from anywhere with an internet connection, making voting more accessible and convenient.
- Increased turnout: Blockchain-based voting could lead to increased voter turnout by making voting more convenient and secure.

Purpose

The need for blockchain-based electronic voting:

- Traditional voting systems are vulnerable to a number of problems, including fraud, inaccuracy, long lines and wait times, and low voter turnout. Blockchain-based electronic voting can address all of these problems.

- Blockchain is a distributed ledger technology that is secure, transparent, and tamper-proof. It can be used to create a voting system where every vote is recorded accurately and securely, and the results are transparent and verifiable by all stakeholders.

- Blockchain-based voting can help to reduce fraud by using cryptography to secure the voting process and make it tamper-proof. Once a vote is recorded on the blockchain, it cannot be altered or deleted. This makes it much more difficult to commit voter fraud, such as ballot stuffing or vote tampering.

- Blockchain-based voting can also help to improve accuracy by reducing the risk of human error. In traditional voting systems, ballots are counted manually by election workers. This process is prone to human error, which can lead to inaccurate results. Blockchain-based voting systems automate the ballot counting process, which reduces the risk of human error and improves accuracy.

- Blockchain-based voting can also help to reduce long lines and wait times. In traditional voting systems, voters must line up at polling places to cast their ballots. This can lead to long wait times, especially in large elections. Blockchain-based voting systems allow voters to cast their ballots from anywhere with an internet connection. This eliminates the need for polling places and reduces long lines and wait times.

- Finally, blockchain-based voting can help to increase voter turnout. Many people are discouraged from voting due to the inconvenience or lack of trust in the system. Blockchain-based voting is more convenient and secure than traditional voting systems, which could lead to increased voter turnout.

- Overall, blockchain-based electronic voting has the potential to revolutionize the way elections are conducted. It can help to make elections more secure, accurate, accessible, and efficient.

2 LITERATURE SURVEY

Existing Problem

Traditional electronic voting systems have a number of problems, including:

- **Security:** Traditional electronic voting systems are vulnerable to hacking and other forms of fraud. For example, hackers could gain access to voting machines and change votes, or they could create fake voting machines that record votes incorrectly.
- **Transparency:** It is difficult to verify that traditional electronic voting systems are counting votes correctly. This is because the voting process is often opaque and there is no way to audit the results.
- **Accessibility:** Traditional electronic voting systems are not accessible to all voters. For example, people with disabilities may have difficulty using voting machines, and people who live in rural areas may not have access to polling places.

Security: Blockchain is a secure technology that is difficult to hack. This is because blockchain uses cryptography to secure data and make it tamper-proof.

- **Transparency:** Blockchain is a transparent technology. This is because all transactions on the blockchain are recorded in a public ledger that anyone can view. This makes it possible to audit the voting process and verify that votes are being counted correctly.
- **Accessibility:** Blockchain-based electronic voting systems are accessible to all voters. This is because voters can cast their ballots from anywhere with an internet connection. This makes it easier for people with disabilities, those who live in rural areas, and other groups who may have difficulty voting using traditional methods.

References

[1] Sos.ca.gov. (2007). Top-to-Bottom Review | California Secretary of State. Available at: http://www.sos.ca.gov/elections/voting-systems/oversight/ top-bottom-review/.

[2] Nicholas Weaver. (2016). Secure the Vote Today Available at:https:// www.lawfareblog.com/secure-vote-today.

[3] TechCrunch, (2018). Liquid democracy uses blockchain to fix politics, and now you can vote for it. Available at: https://techcrunch.com/2018/ 02/24/liquid-democracy-uses-blockchain/

[4] Ajit Kulkarni, (2018), "How To Choose Between Public And Permissioned Blockchain For Your Project", Chronicled, 2018.

[5] "What Are Smart Contracts? A Beginner's Guide to Smart Contracts", Blockgeeks, 2016. Available at: https://blockgeeks.com/guides/ smart-contracts/

[6] Salanfe, Setup your own private Proof-of-Authority Ethereum network with Geth, Hacker Noon, 2018. Available at: https://tinyurl.com/ y7g362kd

Problem Statement Definition

- Hacking: Voting machines and election systems can be hacked, allowing attackers to change votes or prevent people from voting.
- Voter fraud: People can vote multiple times, impersonate other voters, or cast ballots on behalf of deceased voters.
- Inaccuracy: Human errors in counting ballots can lead to inaccurate results.
- Lack of transparency: It is difficult to verify that votes are being counted correctly and that the election results are accurate.

However, there are still a number of challenges that need to be addressed before blockchain-based electronic voting systems can be widely deployed, including:

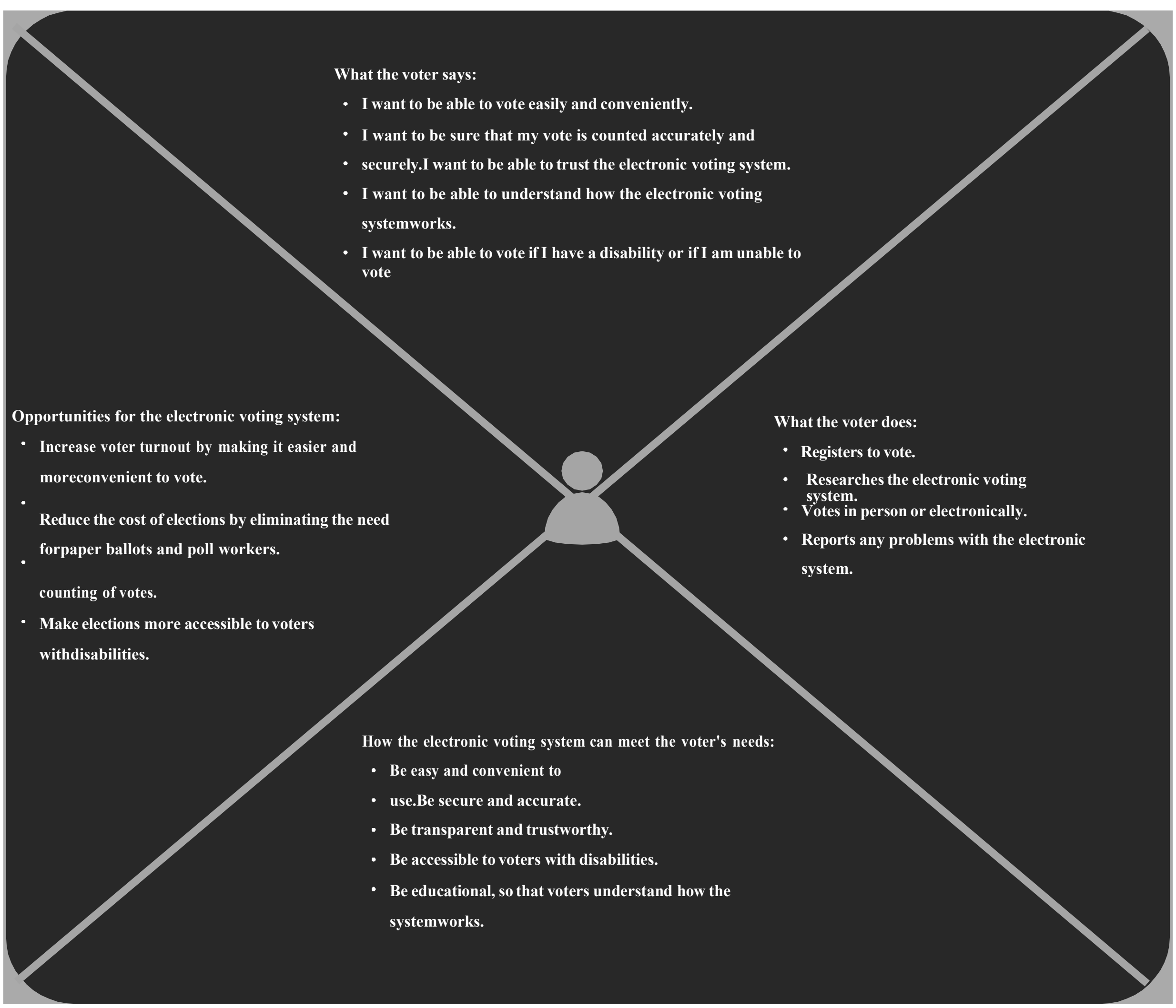
- Security: Blockchain-based voting systems need to be designed to be secure against a variety of attacks, including hacking, denial-of-service attacks, and Sybil attacks.
- Privacy: Blockchain-based voting systems need to protect the privacy of voters, ensuring that their votes cannot be traced back to them.
- Scalability: Blockchain-based voting systems need to be able to handle a large number of voters and ballots without sacrificing security or performance.
- Usability: Blockchain-based voting systems need to be easy to use for voters, election officials, and auditors.

- Secure voting protocols: We are developing new voting protocols that are secure against a variety of attacks.
- Privacy-preserving voting: We are developing new voting mechanisms that protect the privacy of voters.

- Scalable voting systems: We are developing new blockchain-based voting systems that can handle a large number of voters and ballots without sacrificing security or performance.
- User-friendly voting systems: We are developing new voting systems that are easy to use for voters, election officials, and auditors.

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

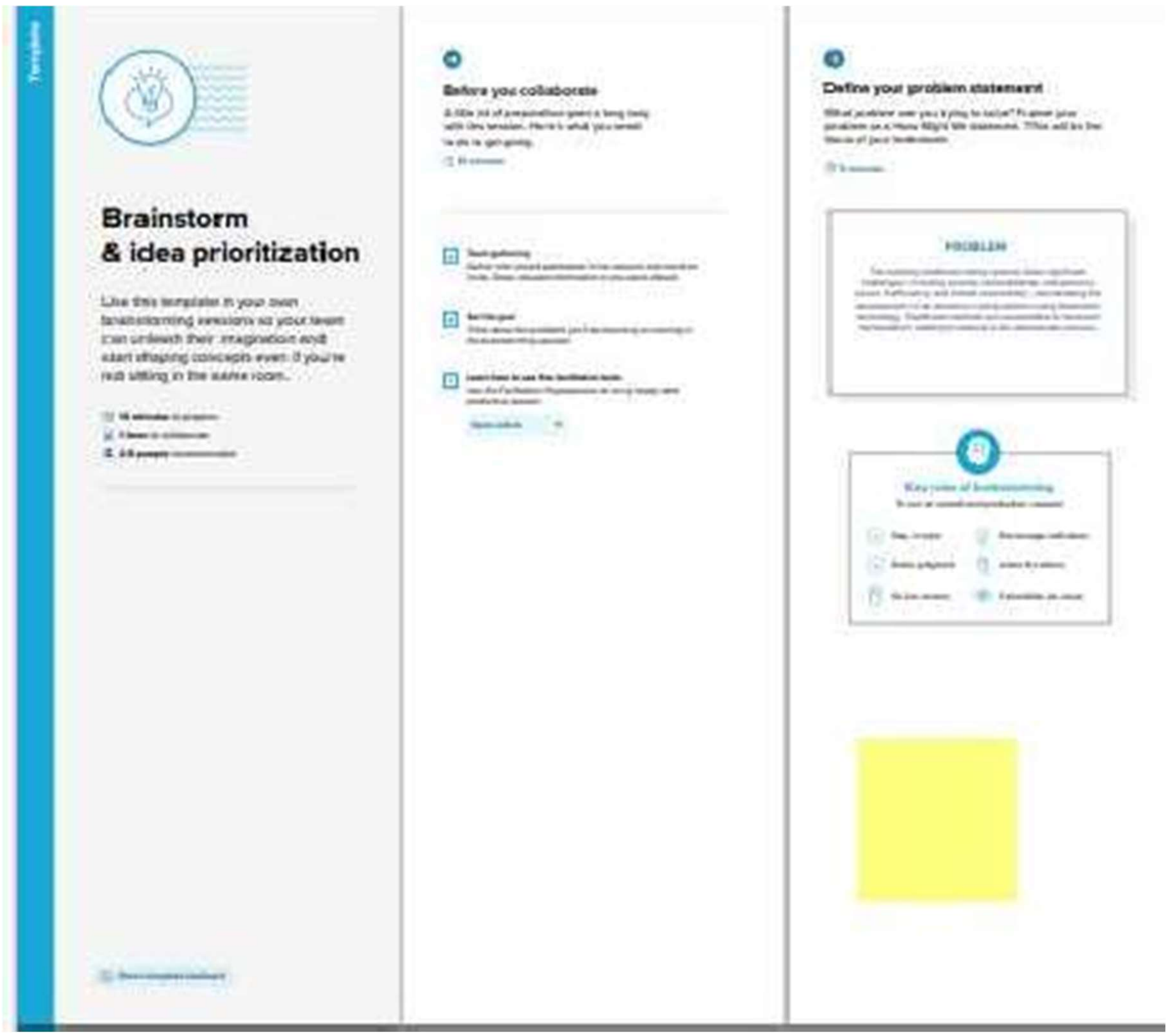



I'm worried about the security of electronic voting systems.
I'm not sure how to use electronic voting systems.
I'm worried that my vote won't be counted accurately.
I'm not sure if I can trust the electronic voting system.
Be easy and convenient to use.
Be secure and accurate.
Be educational, so that voters understand how the system works.
Be accessible to voters with disabilities.
Be transparent and trustworthy.
Voter think
Needs
Feels
Says & Do
Frustrated if they are unable to vote easily or conveniently.
Anxious if they are worried about the security of the electronic voting system.
Confused if they don't understand how to use the electronic voting system.
Empowered if they are able to vote easily and securely.
I want to be able to vote easily and conveniently.
I want to be able to vote if I have a disability or if I am unable to vote in person.
Registers to vote.
Researches the electronic voting system.
Votes in person or electronically.

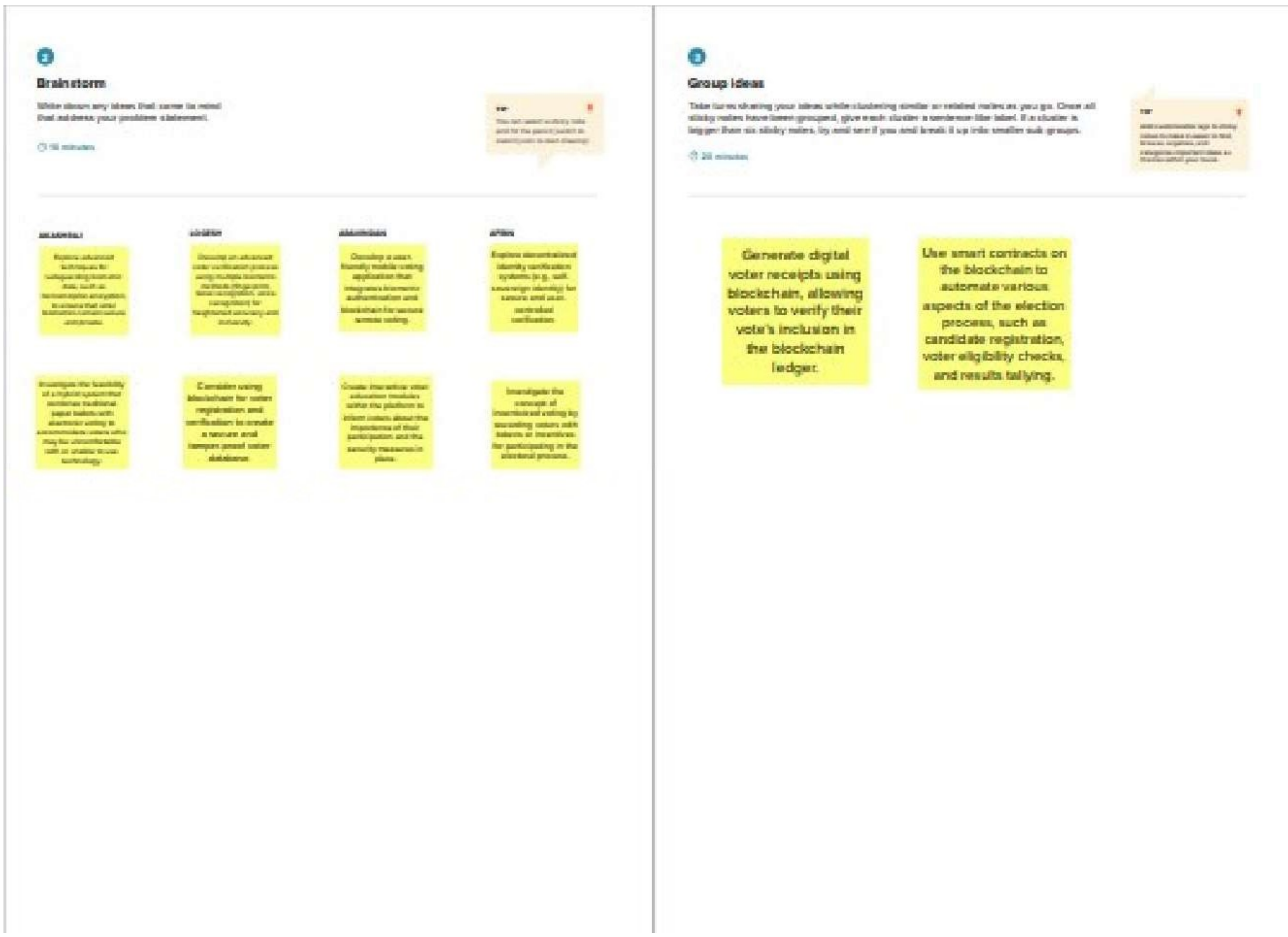
Ideation & Brainstorming

Step 1:-

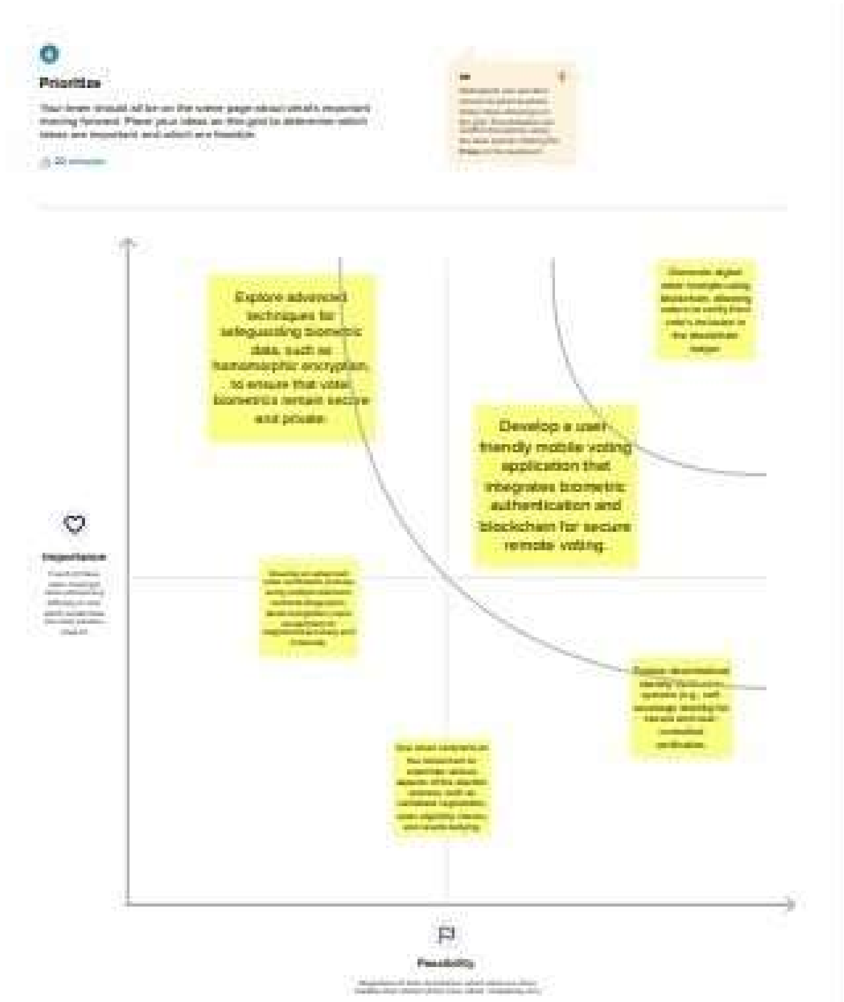
Team Gathering, Collaboration and Select the Problem Statement



Step 2:-
Brainstorm, Idea Listing and Grouping



Step 3:-
Idea Prioritization



4. REQUIREMENT ANALYSIS

Functional Requirements

- Voter registration: Voters should be able to register to vote using a secure and user-friendly process. The system should verify the voter's eligibility and generate a unique voter ID.
- Ballot creation: Election officials should be able to create ballots and publish them to the blockchain. The system should ensure that the ballots are valid and tamper-proof.
- Voting: Voters should be able to cast their votes securely and anonymously. The system should prevent voters from voting multiple times or impersonating other voters.
- Result tabulation: Once the election is over, the system should automatically tabulate the votes and generate the results. The results should be transparent and verifiable by all stakeholders.

In addition to these core functionalities, the system should also have the following features:

- Security: The system should be secure against a variety of attacks, including hacking, denial-of-service attacks, and Sybil attacks.
- Privacy: The system should protect the privacy of voters, ensuring that their votes cannot be traced back to them.
- Scalability: The system should be able to handle a large number of voters and ballots without sacrificing security or performance.
- Usability: The system should be easy to use for voters, election officials, and auditors.

Non-Functional Requirements

Blockchain-based electronic voting systems must adhere to the highest security standards in order to protect the integrity of the voting process and the privacy of voters. Here are some specific security requirements that such systems should meet:

- Use of strong cryptography: All data transmitted and stored in the system should be encrypted using strong cryptographic algorithms.
- **Secure key management:** Private keys should be generated and stored securely.

- Protection against hacking and other attacks: The system should be designed to be resistant to hacking and other attacks, such as denial-of-service attacks and Sybil attacks.

- **Auditable:** The system should be auditable to ensure that the voting process was conducted fairly and accurately.

System Performance:-

Blockchain-based electronic voting systems must be able to handle a large number of voters and ballots without sacrificing performance or security. Here are some specific performance requirements that such systems should meet:

- **Scalability:** The system should be able to scale to support a large number of voters and ballots, even during peak voting times.
- **Availability:** The system should be highly available and accessible to voters from anywhere with an internet connection.
- **Performance:** The system should be able to process votes and generate results quickly and efficiently.

Blockchain-based electronic voting systems should be easy to use for voters, election officials, and auditors. Here are some specific user experience guidelines that such systems should follow:

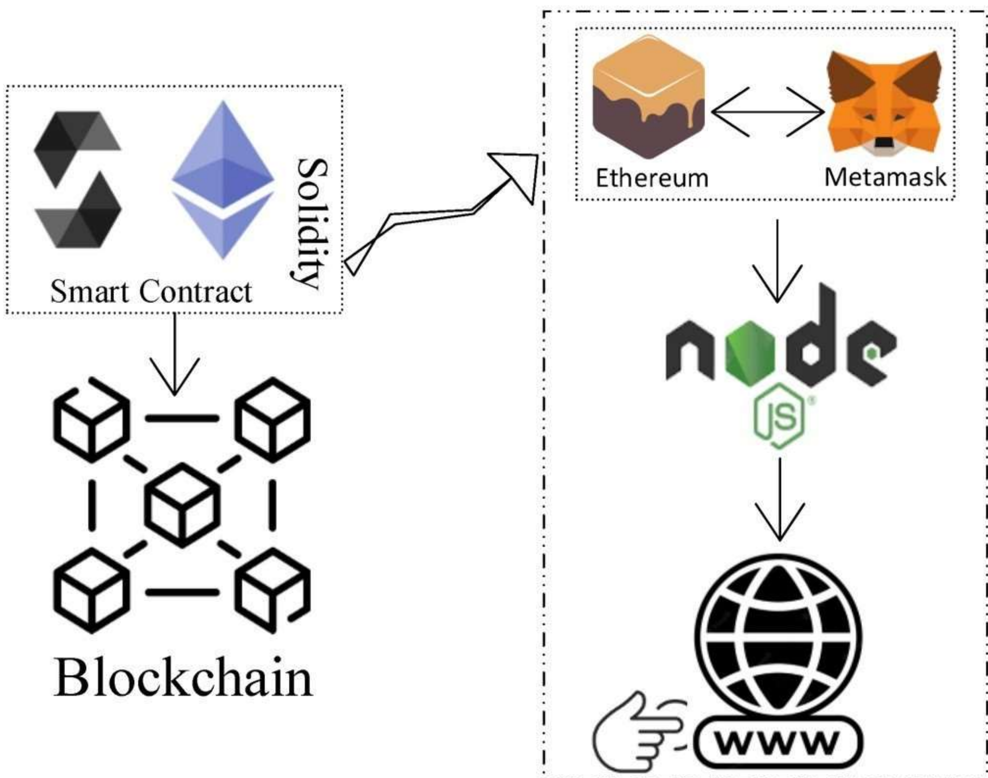
- **User-friendly interface:** The system should have a user-friendly interface that is easy to navigate and understand.
- **Clear instructions:** The system should provide clear instructions to voters on how to cast their votes.

Accessibility: The system should be accessible to voters with disabilities.

- **Support for multiple languages:** The system should support multiple languages to accommodate voters who speak different languages..

5. PROJECT DESIGN

Data Flow Diagrams & User Stories



Solution Architecture

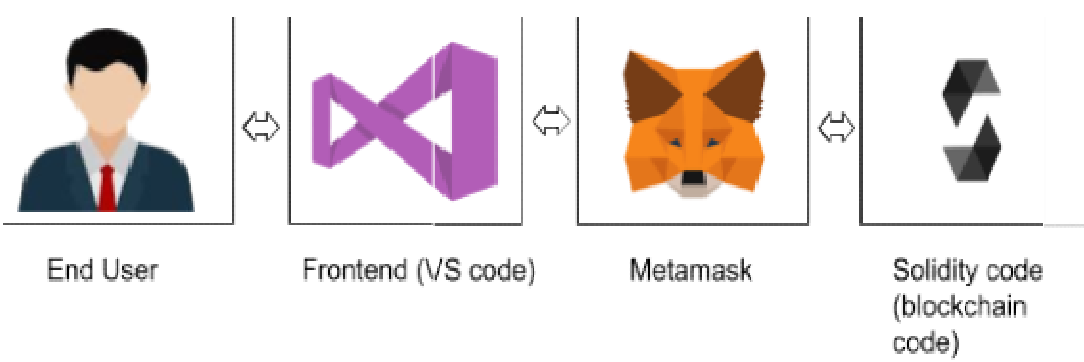
Blockchain-based electronic voting systems use a distributed ledger technology called blockchain to record and verify votes. This makes them more secure and transparent than traditional voting systems.

High-level architecture:

- Blockchain: Stores and verifies votes.
- Smart contracts: Implement core voting logic, such as voter registration, ballot creation, and voting.
- Front-end application: User interface for voters to cast their votes.
- Back-end servers: Manage the voting process and interact with the blockchain.

Key technologies:

- Public key infrastructure (PKI)
- Digital signatures
- Hashing algorithms
- Consensus mechanisms



6. PROJECT PLANNING & SCHEDULING

Technical Architecture

- Blockchain platform: Ethereum is a popular choice due to its support for smart contracts, scalability, and large and active community.
- Voter registration: Voters submit their personal information and a digital signature to the blockchain. The system verifies the voter's eligibility.
- Ballot creation: Election officials create ballots by specifying the candidates or options that voters can choose from. The system generates a unique hash for each ballot and publishes it to the blockchain.
- Voting: Voters submit their voter ID and a ballot hash to the system. The system verifies the voter's eligibility and ensures that the ballot is valid. The system may also encrypt the vote to protect the voter's privacy.
- Result tabulation: Once the election is over, the system automatically tabulates the votes by counting the number of occurrences of each ballot hash. The results are then published to the blockchain and made available to all stakeholders.

Why Ethereum:

- Support for smart contracts
- Scalability
- Large and active community

Conclusion:

Blockchain-based electronic voting systems have the potential to revolutionize the way elections are conducted by making them more secure, transparent, and accessible.

Sprint Planning & Estimation

Sure, here is a simplified version of the project execution plan without the sheet:

Sprint Duration: 2 weeks

Sprint Objective: Develop and implement the voter registration module of the blockchain-based electronic voting system.

Sprint Task 1: Design and implement the voter registration database schema

When designing the voter registration database schema, the team will need to consider the following factors:

- What information needs to be stored about each voter?
- How will the data be organized and indexed?
- How will the data be accessed by the voter registration smart contract?
- How will the data be secured?

Once the database schema has been designed, the team will need to implement it using a database management system. This involves creating the database tables, defining the relationships between the tables, and populating the tables with initial data.

Sprint Task 2: Implement the voter registration smart contract

When implementing the voter registration smart contract, the team will need to consider the following factors:

- How will the smart contract verify the voter's eligibility?
- How will the smart contract generate a unique voter ID?
- How will the smart contract store the voter's information in the database?
- How will the smart contract prevent voters from registering multiple times?
- How will the smart contract be protected from attacks?

The team will also need to test the smart contract thoroughly to ensure that it is working properly and that it is secure.

Sprint Task 3: Develop the voter registration front-end user interface

When developing the voter registration front-end user interface, the team will need to consider the following factors:

- The user interface should be easy to use and navigate.
- The user interface should be accessible to all voters, including those with disabilities.
- The user interface should be secure and protect the voter's privacy.

The team should also get feedback from voters on the user interface to ensure that it is user-friendly and meets their needs.

Sprint Task 4: Integrate the voter registration module with the other components of the system

When integrating the voter registration module with the other components of the system, the team will need to consider the following factors:

- How will the voter registration module interact with the ballot creation module?
- How will the voter registration module interact with the voting module?
- How will the voter registration module be integrated with the system's security and auditing features?

The team will also need to test the integrated system thoroughly to ensure that it is working properly and that it is secure.

By completing these four sprint tasks, the team will have developed and implemented a secure and reliable voter registration system.

Sprint Delivery Schedule

Here is a detailed schedule for each sprint, specifying milestones and deliverables:

Sprint 1

Milestones:

- Design and implement the voter registration database schema.
- Implement the voter registration smart contract.
- Develop the voter registration front-end user interface.

Deliverables:

- Voter registration database schema
- Voter registration smart contract
- Voter registration front-end user interface

Sprint 2

Milestones:

- Integrate the voter registration module with the other components of the system.
- Test the voter registration module and fix any bugs.
- Deploy the voter registration module to the blockchain.

Deliverables:

- Integrated voter registration module
- Tested and bug-free voter registration module
- Deployed voter registration module

Sprint 3

Milestones:

- Develop the ballot creation module.
- Implement the voting module.

Deliverables:

- Ballot creation module
- Voting module

Sprint 4

Milestones:

- Integrate the ballot creation module and the voting module with the voter registration module.
- Test the integrated system and fix any bugs.
- Deploy the integrated system to the blockchain.

Deliverables:

- Integrated ballot creation module, voting module, and voter registration module
- Tested and bug-free integrated system
- Deployed integrated system

Sprint 5

Milestones:

- Develop the result tabulation module.
- Integrate the result tabulation module with the other components of the system.
- Test the integrated system and fix any bugs.
- Deploy the integrated system to the blockchain.

Deliverables:

- Result tabulation module
- Integrated result tabulation module, ballot creation module, voting module, and voter registration module
- Tested and bug-free integrated system
- Deployed integrated system

.

7. CODING & SOLUTIONING

Feature 1

```
// 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,candidateM
ap[_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].v
otingState);
     }
 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;
}
        }
```

Feature 2

```
import { ethers } from "ethers";
import abi from "./voting.json";

export const contractAddress = "0xf852d672915B3bb8A5840569358f19f925100D12";


export const provider = new
ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();

export const votingContract = new ethers.Contract(contractAddress, abi, signer);
```

8. PERFORMANCE TESTING

Performance Metrics:

- **Transaction Throughput:** This metric measures the number of transactions the blockchain-based electronic voting system can handle per unit of time. It's a critical metric to determine the system's capacity.

- **Response Times:** Response time measures the time taken for the system to respond to user actions or queries. It's vital for ensuring that the system remains responsive even under heavy loads.

- **Latency:** Latency measures the time it takes for a transaction to be confirmed and added to the blockchain. Low latency is essential for real-time updates and verification.

- **Scalability:** Scalability refers to the system's ability to handle a growing number of voters and transactions. It's crucial to assess how the system performs as the workload increases.

- **Consensus Algorithm Efficiency:** The efficiency of the consensus algorithm used in the blockchain (e.g., Proof of Stake, Proof of Work) affects transaction processing. Efficiency metrics help evaluate the system's energy consumption and overall performance.

Performance Testing Explanation:

Transaction Throughput: This metric measures the number of transactions that the system can handle per second (TPS). It is a critical metric for determining the system's capacity to handle large numbers of voters and transactions.

To test transaction throughput, you can use a load testing tool to simulate a large number of voters interacting with the system. You can then gradually increase the number of simulated voters to see how the system performs. The goal is to identify limits and optimize for high transaction volumes.

Response Times: This metric measures the time it takes for the system to respond to user actions or queries. It is an important metric for ensuring that the system remains responsive and user-friendly.

To test response times, you can use a monitoring tool to track the time it takes for the system to respond to different types of user requests, such as voting, querying results, and registering. You should test response times under normal and stress conditions to ensure that the system remains responsive even under high load.

Latency: This metric measures the time it takes for a transaction to be confirmed and added to the blockchain. It is an important metric for ensuring that the system provides real-time updates and verification of results.

To test latency, you can measure the time it takes for a simulated vote to be cast, recorded, and verified. You should also test latency under different conditions, such as varying network speeds and system load.

Scalability: This metric refers to the system's ability to handle a growing number of users and transactions. It is an important metric for ensuring that the system can be used to conduct elections of all sizes.

To test scalability, you can use a load testing tool to simulate a large number of voters interacting with the system. You can then gradually increase the number of simulated voters to see how the system performs. The goal is to identify the maximum number of voters that the system can handle without sacrificing performance or security.

Consensus Algorithm Efficiency: The efficiency of the consensus algorithm used in the blockchain can affect transaction processing. For example, proof-of-work consensus algorithms can be energy-intensive and slow.

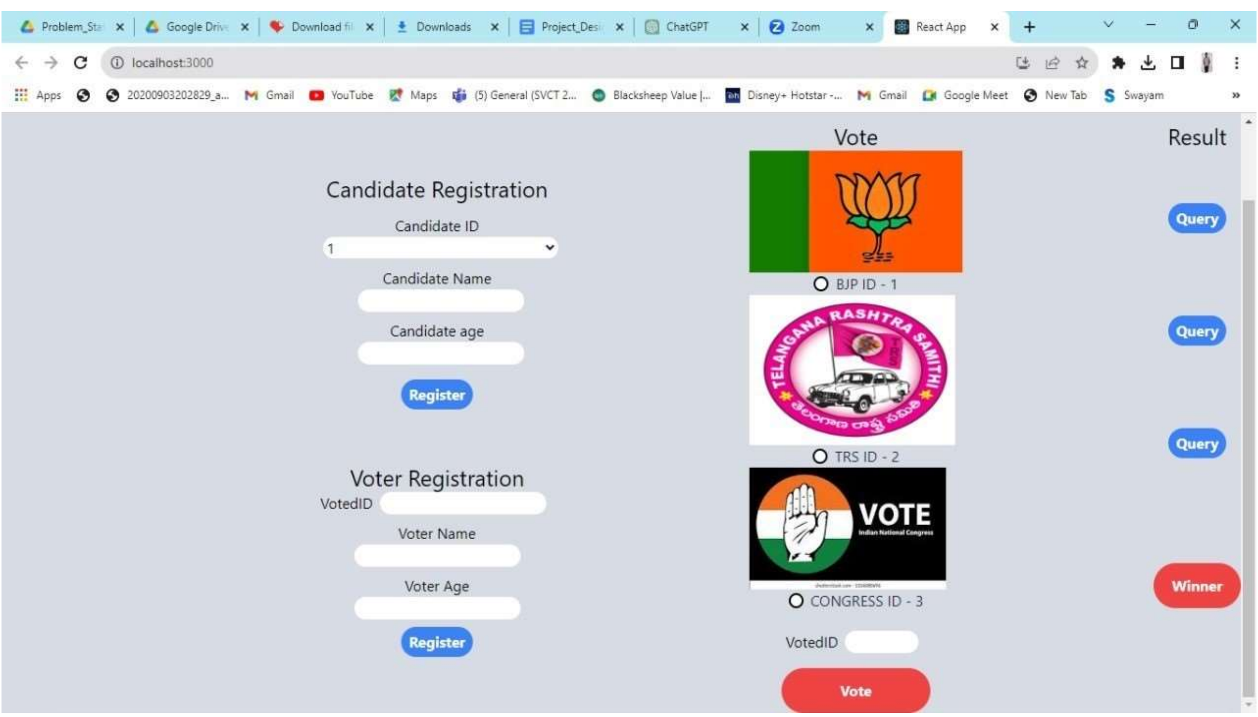
To test consensus algorithm efficiency, you can compare the performance of different consensus algorithms on the same blockchain. You can also measure the energy consumption of the system when using different consensus algorithms.

In addition to the metrics and testing described above, it is also important to consider the following factors when evaluating the performance of a blockchain-based electronic voting system:

- **Security:** The system should be resistant to attacks, such as double voting and denial-of-service attacks.
- **Privacy:** The system should protect the privacy of voters and their votes.
- **Auditability:** The system should be auditable to ensure that votes were cast and counted correctly.

9. RESULT

9.1 Output Screenshot



<u>10.</u> ADVANTAGES & DISADVANTAGES

Advantages

- **Security:** Blockchain technology uses cryptography to secure and encrypt all data, making it virtually impossible to hack or tamper with. This makes it an ideal platform for storing and transmitting sensitive voting data.
- **Transparency:** All transactions on a blockchain are public and immutable, meaning thatanyone can view and verify them. This transparency helps to build trust in the voting process and reduce the risk of fraud.
- **Immutability:** Once a vote is recorded on a blockchain, it cannot be changed or deleted. This makes it extremely difficult to manipulate or rig the election results.
- **Efficiency:** Blockchain can help to streamline the voting process by automating manyof the manual tasks involved, such as voter registration, ballot counting, and tabulation. This can save time and money for election authorities.
- **Accessibility:** Blockchain-based voting systems can make it easier for people to vote, especially those who live in remote or underserved areas. Voters can cast their ballots from anywhere with an internet connection, without having to go to a polling station.

Disadvantages

- **Identity verification:** Verifying voter identities in a secure and private way is one of thebiggest challenges facing blockchain-based voting systems. It is important to ensure that only eligible voters can cast ballots and that their votes cannot be traced back to them.

- **Voter confidentiality:** Achieving voter confidentiality while maintaining transparencycan be difficult. It is important to balance these two competing goals to ensure that thevoting process is fair and secure.

11 .CONCLUSION

In summary, the blockchain-based electronic voting system represents a significant advancement in modernizing the electoral process. It offers several key advantages while addressing challenges: This system provides:

- **Enhanced security:** Blockchain-based systems use cryptography to secure and encrypt all data, making it virtually impossible to hack or tamper with. This is a significant advantage over traditional voting systems, which are vulnerable to a variety of attacks, such as voter fraud, ballot stuffing, and election rigging.
- **Transparency**: All transactions on a blockchain are public and immutable, meaning that anyone can view and verify them. This transparency helps to build trust in the voting process and reduce the risk of fraud.
- **Verifiable results:** Once a vote is recorded on a blockchain, it cannot be changed or deleted. This makes it extremely difficult to manipulate or rig the election results. Voters can also verify their own votes to ensure that they were counted correctly.
- **Efficiency:** Blockchain can help to streamline the voting process by automating many of the manual tasks involved, such as voter registration, ballot counting, and tabulation. This can save time and money for election authorities.
- **Accessibility:** Blockchain-based voting systems can make it easier for people to vote, especially those who live in remote or underserved areas. Voters can cast their ballots from anywhere with an internet connection, without having to go to a polling station.

Blockchain-based electronic voting systems also address a number of challenges that have been faced by traditional voting systems:

- **Secure identity verification:** Blockchain-based systems can use a variety of methods to verify voter identities, such as digital signatures and biometric authentication. This helps to ensure that only eligible voters can cast ballots and that their votes cannot be traced back to them.
- **Voter confidentiality:** Blockchain-based systems can use encryption and other techniques to protect voter privacy. This is important because voters should be able to cast their ballots without fear of retribution or discrimination.

12 .FUTURE SCOPE

Biometric Authentication Integration:-

Biometric authentication can be used to further enhance the security of blockchain-based electronic voting systems by ensuring that only eligible voters can cast ballots. This can be done by requiring voters to provide a biometric scan, such as a fingerprint or retina scan, when they cast their votes. This would make it very difficult for people to impersonate other voters or cast multiple ballots.

Decentralized Identity Management:-

Decentralized identity solutions on the blockchain can give voters more control over their personal information. This is because voters would be able to create and manage their own digital identities without having to rely on a third party. This would make it easier for voters to verify their identity without having to share their personal information with anyone they don't trust.

Smart Contracts for Enhanced Automation:-

Smart contracts can be used to automate many of the manual processes involved in voting, such as voter eligibility verification, ballot counting, and tabulation. This can save time and money for election authorities, and it can also help to reduce the risk of errors.In addition to the three areas listed above, here are some other potential future directions for blockchain-based electronic voting systems:

- **End-to-end encryption:** This would ensure that all data transmitted and stored on the blockchain is encrypted, making it even more difficult for attackers to gain access to it.
- **Zero-knowledge proofs:** This would allow voters to prove their eligibility and cast their ballots without revealing their identity or their vote to anyone else.
- **Post-election auditing:** This would allow voters to independently verify that their votes were counted correctly and that the election results are accurate.

13. APPENDIX

This appendix provides additional information on the following topics related to blockchain-based electronic voting systems:

- Security considerations
- Privacy considerations
- Scalability considerations
- Regulatory considerations
- Glossary of terms

Security considerations

One of the most important security considerations for blockchain-based electronic voting systems is the need to protect voter privacy. This can be achieved by using cryptographic techniques to encrypt voter data and by using decentralized identity solutions to allow voters to control their own personal information.

Another important security consideration is the need to prevent voter fraud. This can be achieved by using biometric authentication to verify voter identities and by using smart contracts to automate the process of voter eligibility verification.

Privacy considerations

- While blockchain-based electronic voting systems can offer a number of privacy benefits, such as the ability for voters to cast their ballots without revealing their identity, it is important to carefully consider how voter data is collected, stored, and used.

- For example, it is important to ensure that voter data is only collected for the purpose of conducting the election and that it is not shared with any third parties without the voter's consent. It is also important to ensure that voter data is securely stored and that it is deleted after the election is over.

Scalability considerations

- One of the challenges of blockchain-based electronic voting systems is scalability. Blockchain networks can be slow and expensive to process transactions, especially when there are a large number of voters and transactions.

- To address this challenge, it is important to carefully design the blockchain-based voting system and to choose a blockchain platform that is scalable. It is also important to use optimization techniques to reduce the number of transactions that need to be processed.

Regulatory considerations

- Before deploying a blockchain-based electronic voting system, it is important to consider the relevant laws and regulations. In some jurisdictions, there may be specific regulations that must be followed.

- It is also important to consider the public's perception of blockchain-based voting systems. While blockchain technology is becoming more widely accepted, there is still some skepticism about its use in voting systems. It is important to build trust with the public and to ensure that they understand the benefits and risks of blockchain-based voting before deploying such a system.

GitHub & Project Demo Link

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1gIVjCUe6dSF0pibuoHuXo4cfzH NeFQ 44

SOURCE CODE LINK:

https://drive.google.com/drive/folders/1p_wzDPfPpifX_HSjeRvwLHaR7B_wTXm-?usp=drive_link

GITHUB LINK:

https://github.com/Kakashraj/NM2023TMID09491