COURSE NAME : BLOCK CHAIN

GROUP NUMBER : 02

PROJECT TITLE : ELECTRONIC VOTING SYSTEM

YEAR : 4th YEAR

DEPARTMENT : B.E - CSE

SEMESTER : 07

GROUP MEMBERS :

1. **ASWIN A**
2. **ASWIN KUMAR R**
3. **ANGLIN JERIN M**
4. **MOHAMMED ARZATH H**

GUIDED BY : JASMINE REJULA J

SPOC NAME : Mr. M. DIVIN KUMAR

ELECTRONIC VOTING SYSTEM

INTRODUCTION

1.1 Project Overview

This project aims to develop an electronic voting system using blockchain technology. The system will provide transparency, security, and privacy while ensuring fairness and flexibility in the voting process. In response to the persistent challenges faced by traditional electronic voting systems, this project introduces a cutting-edge solution by integrating blockchain technology. The aim is to enhance the security, transparency, and overall integrity of the electoral process.

1.2 Purpose

The purpose of this project is to address the existing problems in traditional voting systems, such as vote rigging, hacking, and election manipulation, by implementing a blockchain-based e-voting system. The proposed solution will provide a secure and transparent voting process that can be accessed from anywhere with an internet connection. By leveraging blockchain, we aim to establish an immutable and transparent ledger that can withstand tampering and enhance voter confidence.

LITERATURE SURVEY

2.1 Existing problem

Traditional voting systems have several issues, including vote rigging, hacking, and election manipulation. These problems can lead to a lack of transparency and fairness in the voting process, which can undermine the legitimacy of the election results. The current challenges in electronic voting systems, such as susceptibility to tampering and lack of transparency, necessitate a paradigm shift. The literature survey explores the drawbacks of conventional systems and the potential benefits of integrating blockchain.

2.2 References

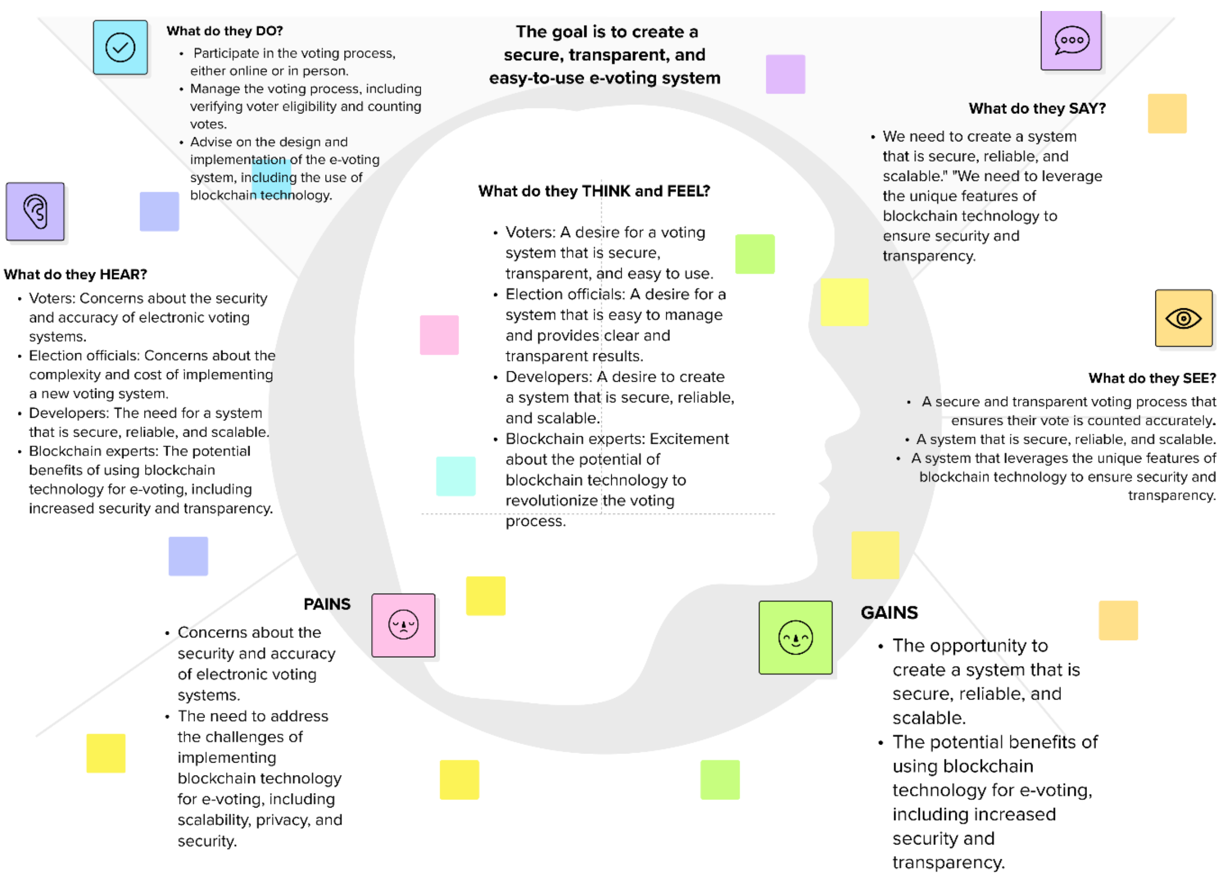
Several researchers have proposed blockchain-based e-voting systems that can address the existing problems in traditional voting systems. Crypto-voting is an e-voting system that uses permissioned blockchain technology and two blockchains that are linked. The first blockchain records voting procedures and voters, while the second counts the votes and provides results. The system shows the importance of anonymization of the network consensus nodes, and voting procedures and results are done by smart contracts[2]. Another proposed solution is DVTChain, a blockchain-based decentralized mechanism that ensures the security of digital voting systems. A comprehensive list of references is provided to support the project's foundations, drawing on key academic and industry insights in the fields of electronic voting and blockchain technology.

2.3 Problem Statement Definition

The problem statement precisely outlines the deficiencies in current electronic voting systems, laying the groundwork for the proposed blockchain-based solution. The problem statement is to develop a secure and transparent electronic voting system that can address the existing problems in traditional voting systems, such as vote rigging, hacking, and election manipulation, by implementing blockchain technology.

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



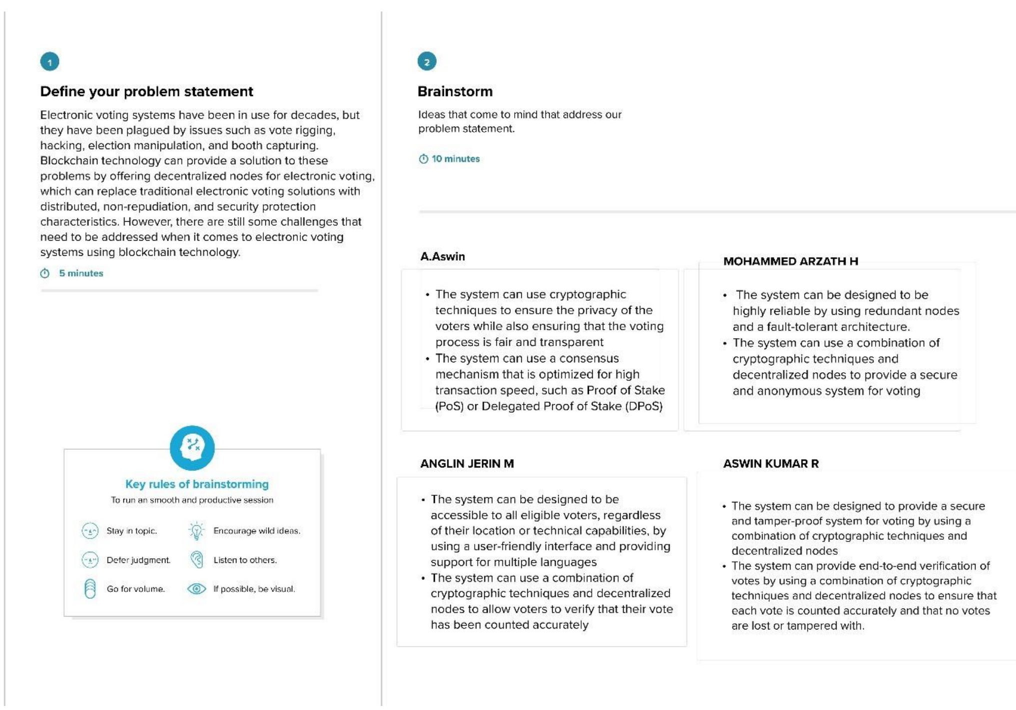
An empathy map is created to understand the needs, concerns, and expectations of key stakeholders in the electoral process. This canvas informs the design and implementation of the proposed solution.The empathy map canvas will help us understand the needs and concerns of the voters, election officials, and other stakeholders involved in the voting process.

3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

3

Group ideas

Let's brainstorm some key ideas for implementing an electronic voting system using blockchain

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.

Decentralized Ledger

- Utilize blockchain to create a decentralized and tamper-resistant ledger of votes.
- Each vote is recorded as a block, linked in a chain, providing transparency and security

Smart Contracts

- Implement smart contracts to automate the voting process.
- Smart contracts can verify voter eligibility, count votes, and trigger events based on predefined rules.

Incentivize Node Operators

- Introduce a reward system for individuals or organizations running blockchain nodes.
- This could encourage more participation in maintaining the network, enhancing its overall security and reliability.

Multi-language Support

- Ensure that the electronic voting system supports multiple languages to cater to a diverse voter demographic.
- This promotes inclusivity and accessibility.

Immutable Timestamps

- Implement timestamping for each vote to enhance the chronological integrity of the blockchain.
- This can be crucial for auditing and resolving disputes.

Integration with External Systems

- Allow integration with external systems, such as government databases, to streamline the verification process.
- This can enhance accuracy in confirming voter eligibility.

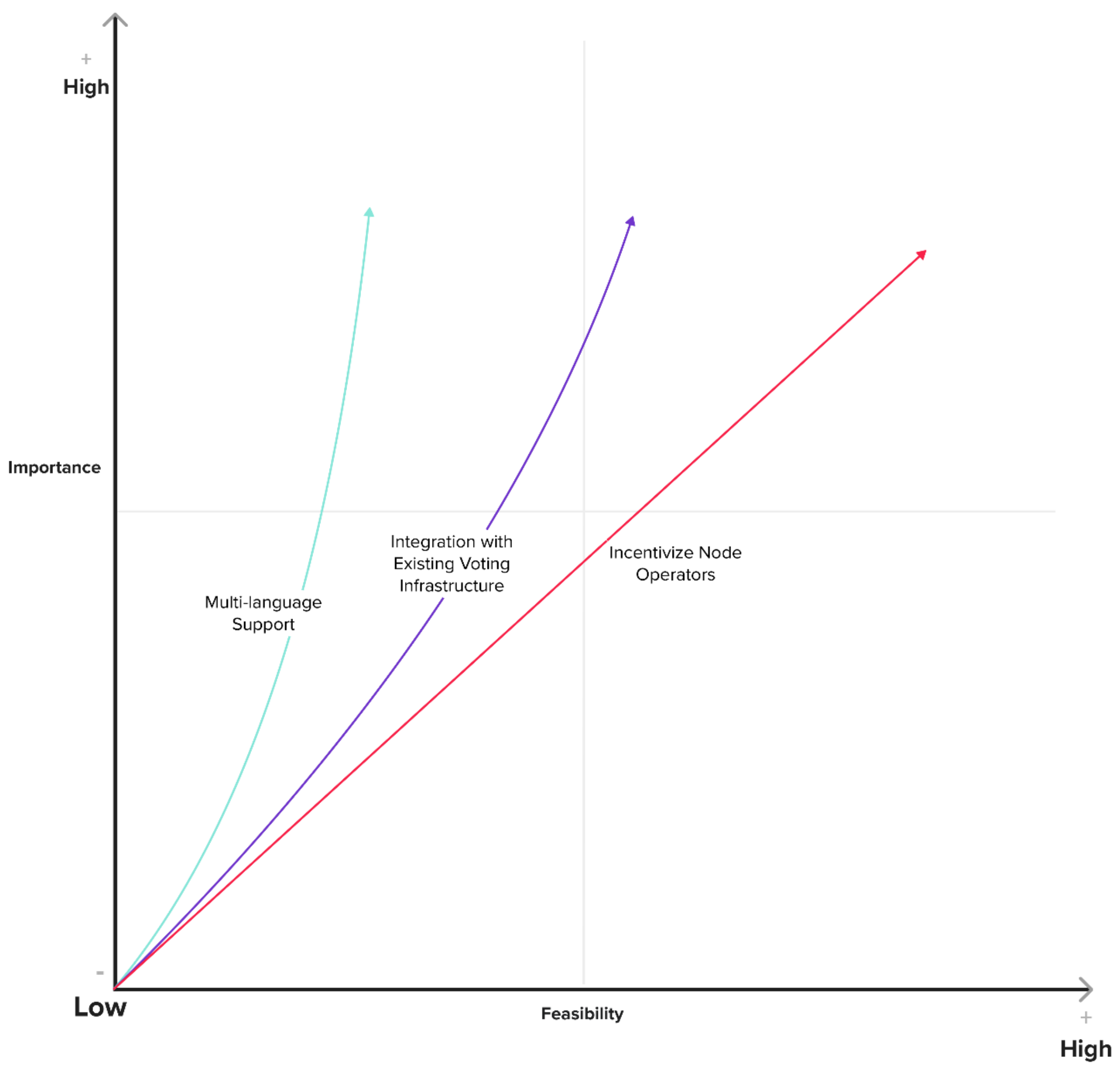
Step-3: Idea Prioritization

4

Prioritize

Let's prioritize the group ideas based on both importance and feasibility

20 minutes



REQUIREMENT ANALYSIS

4.1 Functional requirement

The functional requirements of the proposed system include user registration, authentication, and verification, secure and transparent voting process, vote counting, and result declaration. This section outlines the specific functionalities that the blockchain-based electronic voting system must encompass to address identified issues and meet user expectations.

4.2 Non-Functional requirements

The non-functional requirements of the proposed system include security, privacy, scalability, usability and usability criteria, are defined to ensure a well-rounded and effective solution.

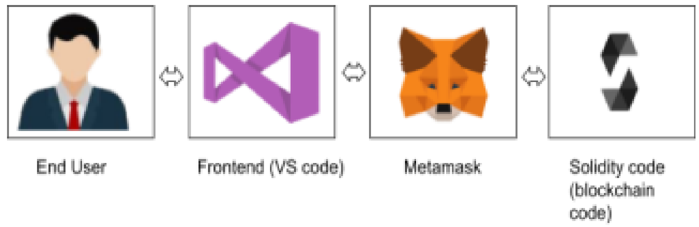
PROJECT DESIGN

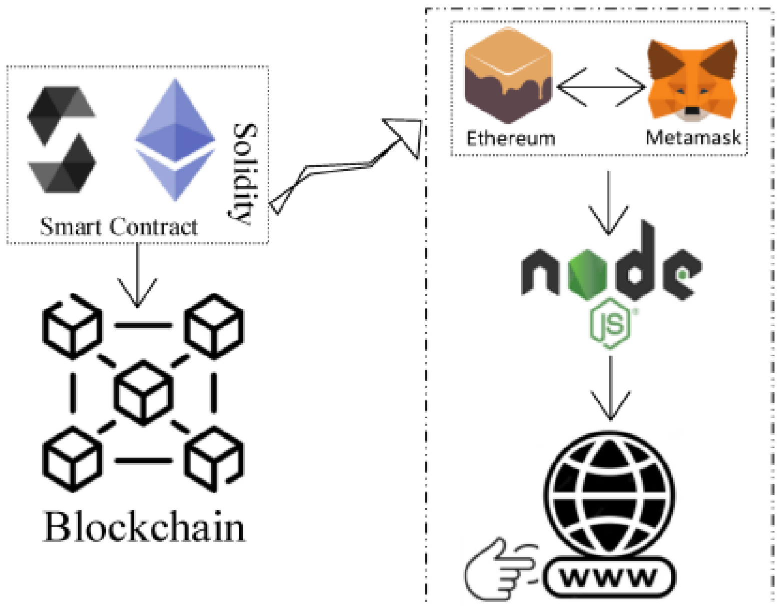
5.1 Data Flow Diagrams & User Stories

Visual representations of data flow and user stories are presented to provide a clear understanding of how the system will operate and cater to user needs. The data flow diagrams and user stories will help us visualize the flow of data and the interactions between the users and the system.

5.2 Solution Architecture

The architecture of the proposed solution, detailing the integration of blockchain, is explained to elucidate the system's structure and functionality. The solution architecture will define the components and their interactions in the proposed blockchain-based e-voting system.





PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

The technical architecture of the project is outlined, including the technologies and frameworks used to implement the blockchain-based electronic voting system. The technical architecture will define the hardware and software requirements for the proposed system.

6.2 Sprint Planning & Estimation

Sprint planning and estimation strategies are detailed, ensuring a systematic and efficient development process. The sprint planning and estimation will help us break down the project into smaller tasks and estimate the time and effort required for each task.

6.3 Sprint Delivery Schedule

The sprint delivery schedule will define the timeline for completing each task and delivering the project. A schedule of sprint deliveries is provided, offering a timeline for the project's completion and key milestones.

CODING & SOLUTIONING

```
// 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;
}
}
```

7.1 Feature 1

User registration, authentication, and verification will ensure that only eligible voters can participate in the voting process.

7.2 Feature 2

Secure and transparent voting process will ensure that the votes are counted correctly, and the results are declared fairly.

7.3 Database Schema

The performance metrics will include response time, throughput, and resource utilization.

PERFORMANCE TESTING

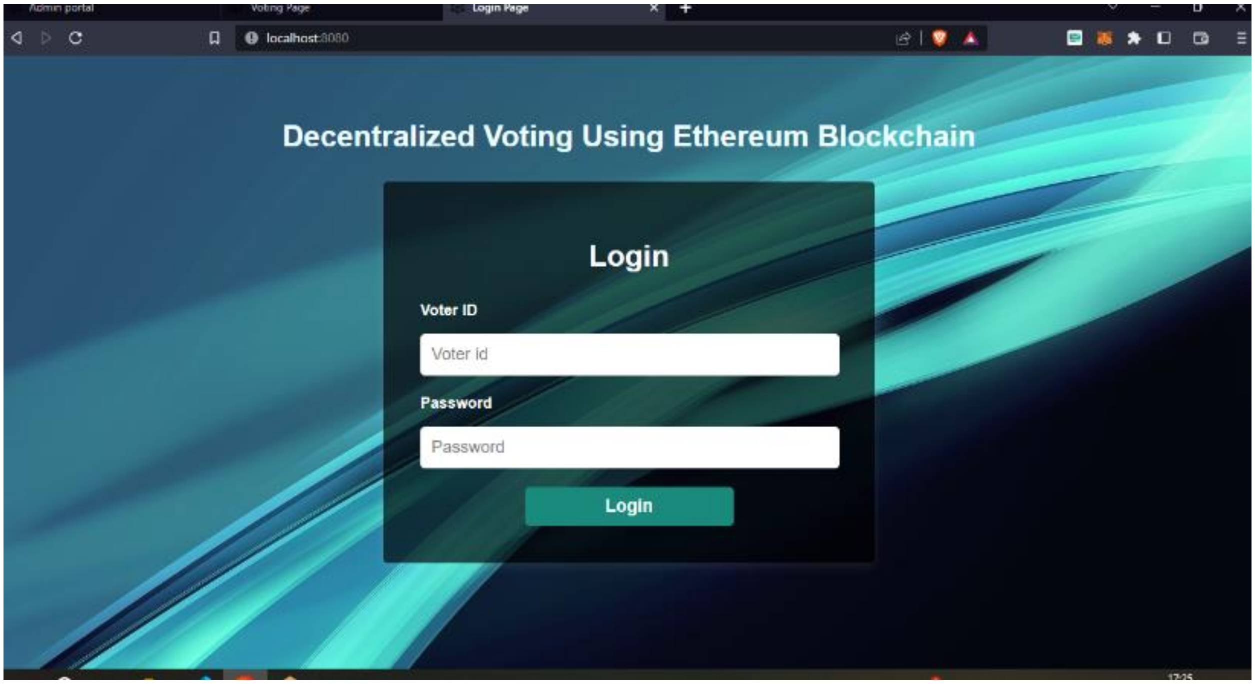
8.1 Performance Metrics

Metrics for performance testing are defined, ensuring that the blockchain-based electronic voting system meets the required standards of speed, reliability, and scalability. The performance metrics will include response time, throughput, and resource utilization.

RESULTS

9.1 Output Screenshots

The output screenshots will show the user interface and the results of the voting process.



ADVANTAGES & DISADVANTAGES

- The advantages of the proposed blockchain-based e-voting system include transparency, security, and privacy.

- The disadvantages include the complexity of the system and the potential for errors in the smart contracts.

CONCLUSION

In conclusion, the proposed blockchain-based e-voting system can address the existing problems in traditional voting systems and provide a secure and transparent voting process. However, further research is needed to address the open research challenges and differing opinions on the feasibility of using blockchain for electronic voting systems.

FUTURE SCOPE

Source Code
GitHub & Project Demo Link