ELECTRONIC VOTING SYSTEM USING BLACKCHAIN TECHNOLOGY

PROJECT DOCUMENTATION

Submitted by

Team ID:	NM2023TMID03831
Team Leader:	RAJESH C
Team Member:	PREMNATH S
Team Member:	MAHESH S
Team Member:	PUGAZHARASAN D

INDEX

1. INTRODUCTION

1.1Project Overview:

1. Introduction:

The Blockchain-Based Electronic Voting System aims to revolutionize the traditional voting process by leveraging blockchain technology to enhance security, transparency, and accessibility in elections.

2. Security Measure:

- Implement encryption protocols to protect data in transit and at rest.
- Regularly audit and update smart contracts to address any potential vulnerabilities.
- Conduct penetration testing and security audits to identify and mitigate potential risks.

3. Objectives:

- Develop a secure and tamper-proof electronic voting system.
- Ensure transparency and auditability of the voting process.
- Enable remote voting for eligible participants.
- Eliminate the risk of double-voting and ensure the integrity of the election results.

4.User Interface:

- Design an intuitive, user-friendly interface for both voters and election administrators.
- Ensure accessibility and compliance with relevant standards.

5. Testing and Deployment:

- Conduct extensive testing, including unit tests, integration tests, and security audits.
- Deploy the system on a secure and reliable infrastructure.

6. Compliance and Legal Considerations:

- Ensure compliance with relevant electoral laws and regulations.
- Establish a clear process for handling disputes or challenges related to the electronic voting system.

7. Future Enhancements:

Explore options for integrating additional features like mobile voting, ranked choice voting, and real-time result tracking.

8. Conclusion:

The Blockchain-Based Electronic Voting System aims to provide a secure, transparent, and accessible platform for conducting elections. Through the integration of blockchain technology, this system addresses many of the challenges associated with traditional voting methods, ultimately enhancing the democratic process.

1.2Project Purpose:

Security and Integrity:

- **Immutable Record:**
  Blockchain creates a tamper-proof ledger of all transactions. Once a vote is recorded, it cannot be altered or deleted, ensuring the integrity of the voting process.

- **Cryptography:**
  Utilizes cryptographic techniques to secure data, making it extremely difficult for unauthorized parties to manipulate or forge votes.

Accessibility:

- **Remote Voting:**
  Enables remote or absentee voting, allowing citizens who might otherwise face barriers (e.g., physical disabilities, distance from polling stations) to participate.

Reduced Fraud:

- **Identity Verification:**
  Utilizes cryptographic techniques for identity verification, reducing the risk of voter impersonation or fraudulent votes.

- **Double Voting Prevention:**
  The decentralized nature of blockchain ensures that a single vote cannot be counted twice.

Efficiency and Speed:

- **Real-Time Updates:**
  Votes are recorded and tallied in real-time, eliminating the need for time-consuming manual counting.

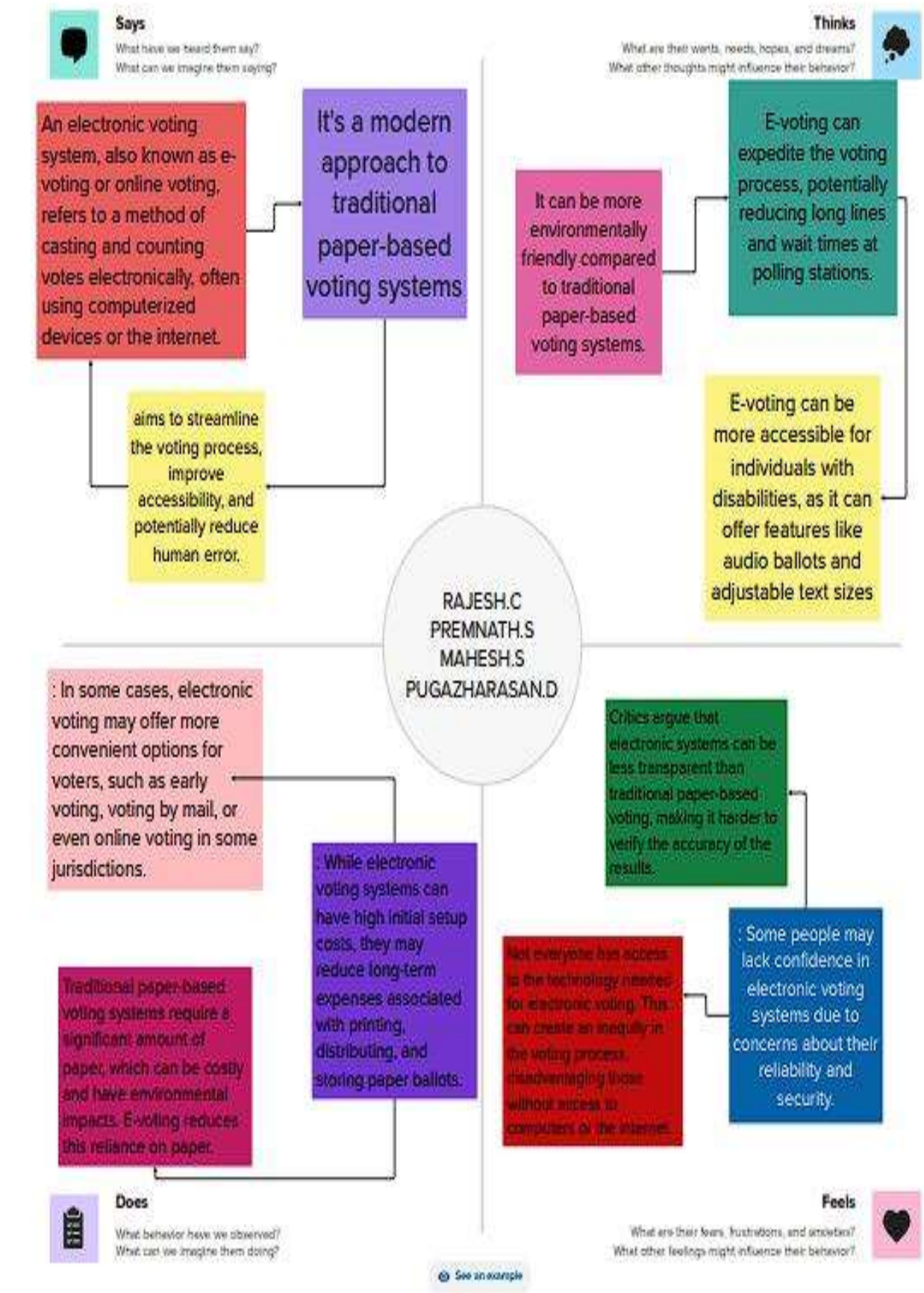
- **Faster Results:**
  Election results can be announced more quickly, providing timely information to the public.

2. IDEATION & PROPOSED SOLUTION

2.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user behavior and Attitudes.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the use

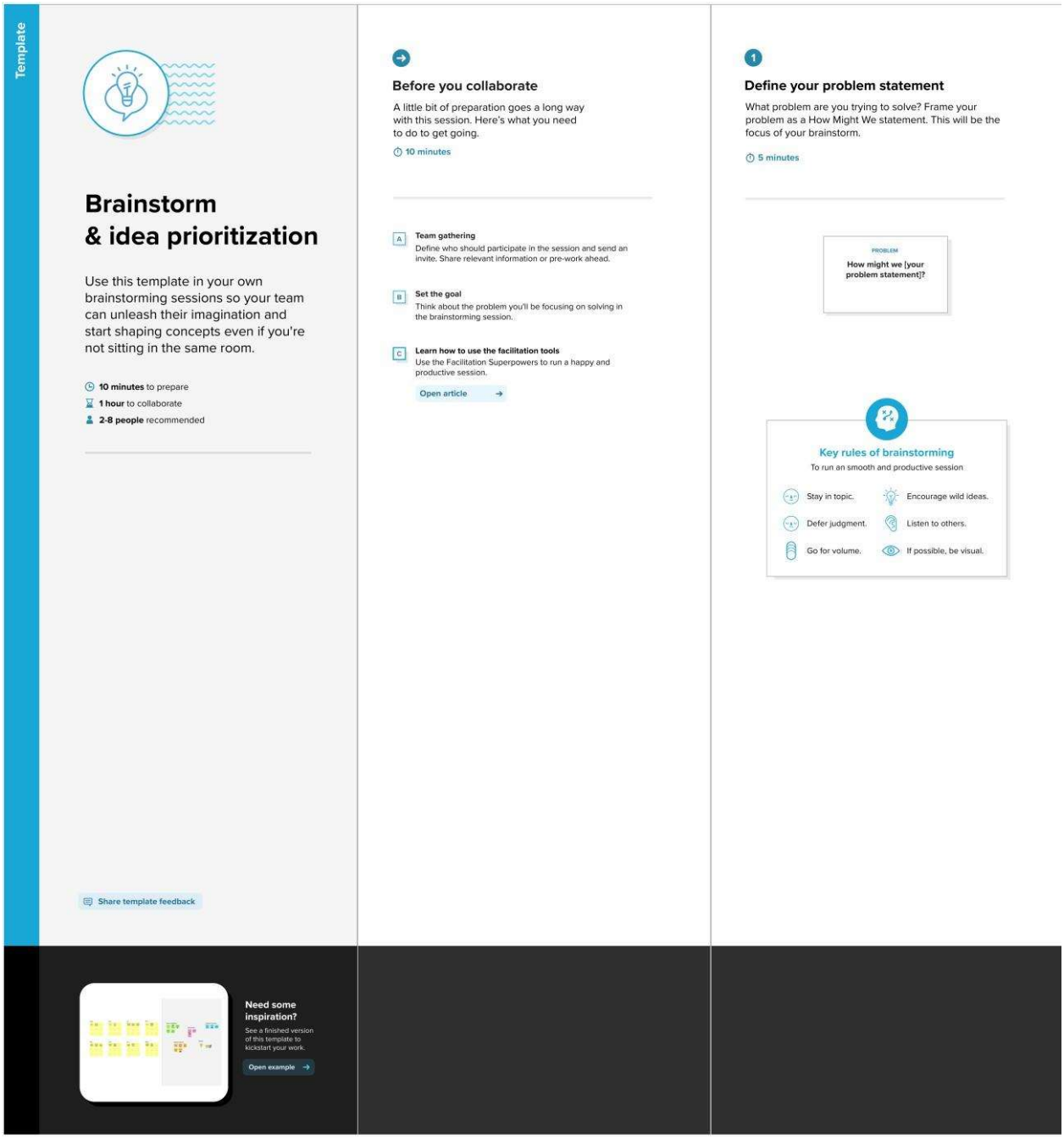

Says
What have we heard them say?
What can we imagine them saying?
Thinks
What are their wants, needs, hopes, and dreams?
What other thoughts might influence their behavior?
An electronic voting system, also known as e-voting or online voting, refers to a method of casting and counting votes electronically, often using computerized devices or the internet.
It's a modern approach to traditional paper-based voting systems
It can be more environmentally friendly compared to traditional paper-based voting systems.
E-voting can expedite the voting process, potentially reducing long lines and wait times at polling stations.
aims to streamline the voting process, improve accessibility, and potentially reduce human error.
E-voting can be more accessible for individuals with disabilities, as it can offer features like audio ballots and adjustable text sizes
RAJESH.C
PREMNATH.S
MAHESH.S
PUGAZHARASAN.D
: In some cases, electronic voting may offer more convenient options for voters, such as early voting, voting by mail, or even online voting in some jurisdictions.
Critics argue that electronic systems can be less transparent than traditional paper-based voting, making it harder to verify the accuracy of the results.
: While electronic voting systems can have high initial setup costs, they may reduce long-term expenses associated with printing, distributing, and storing paper ballots.
: Some people may lack confidence in electronic voting systems due to concerns about their reliability and security.
Traditional paper-based voting systems require a significant amount of paper, which can be costly and have environmental impacts. E-voting reduces this reliance on paper.
Not everyone has access to the technology needed for electronic voting. This can create an inequity in the voting process, disadvantaging those without access to computers or the internet.
Does
What behavior have we observed?
What can we imagine them doing?
Feels
What are their fears, frustrations, and anxieties?
What other feelings might influence their behavior?
See an example

2.2 Ideation & Brainstorming

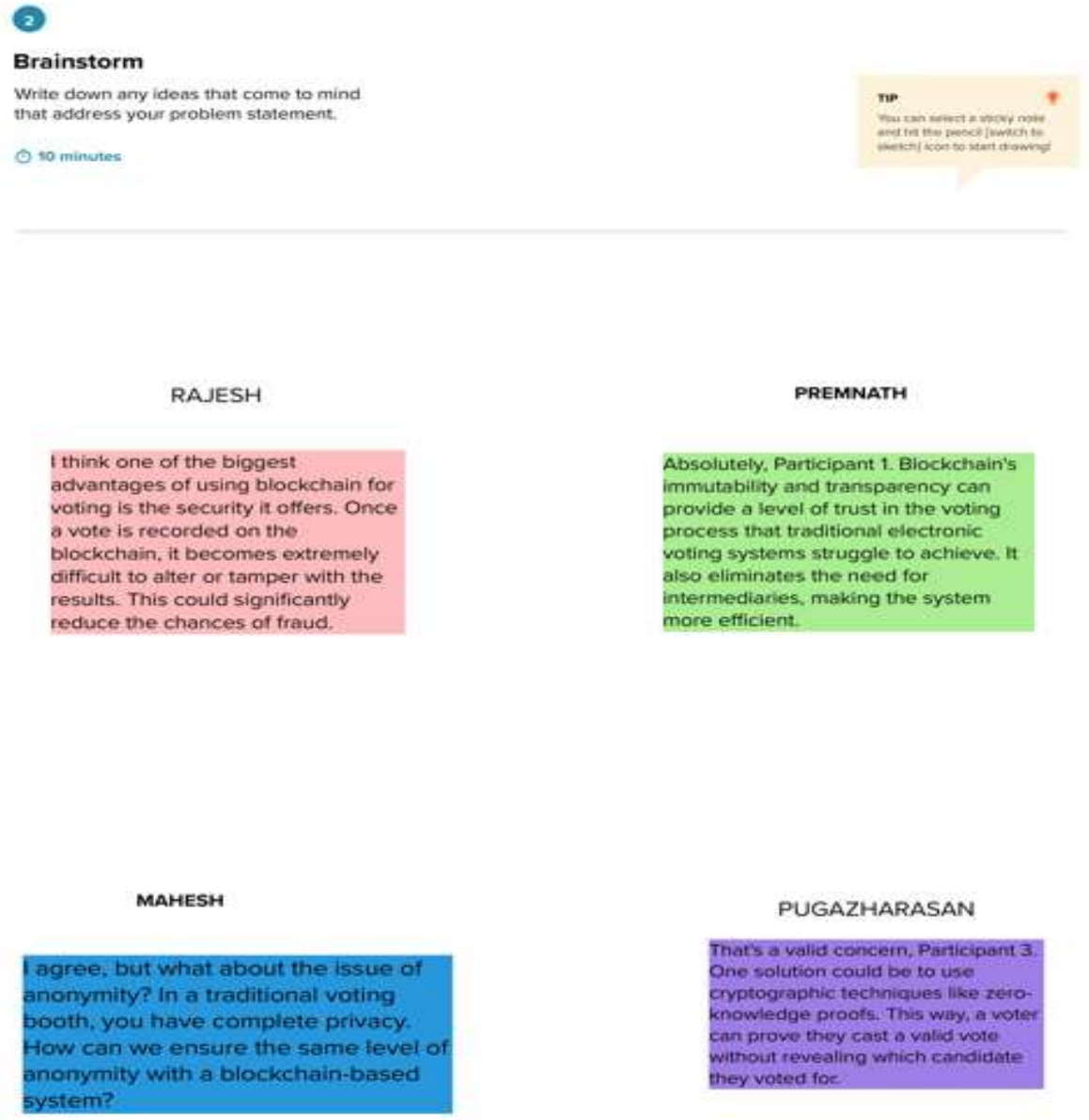
Brainstorming provides a free and open environment that encourages everyone within a team participate in the creative thinking process thatleads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping 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 dame room.

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



Step-2: Brainstorm, Idea Listing and grouping



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

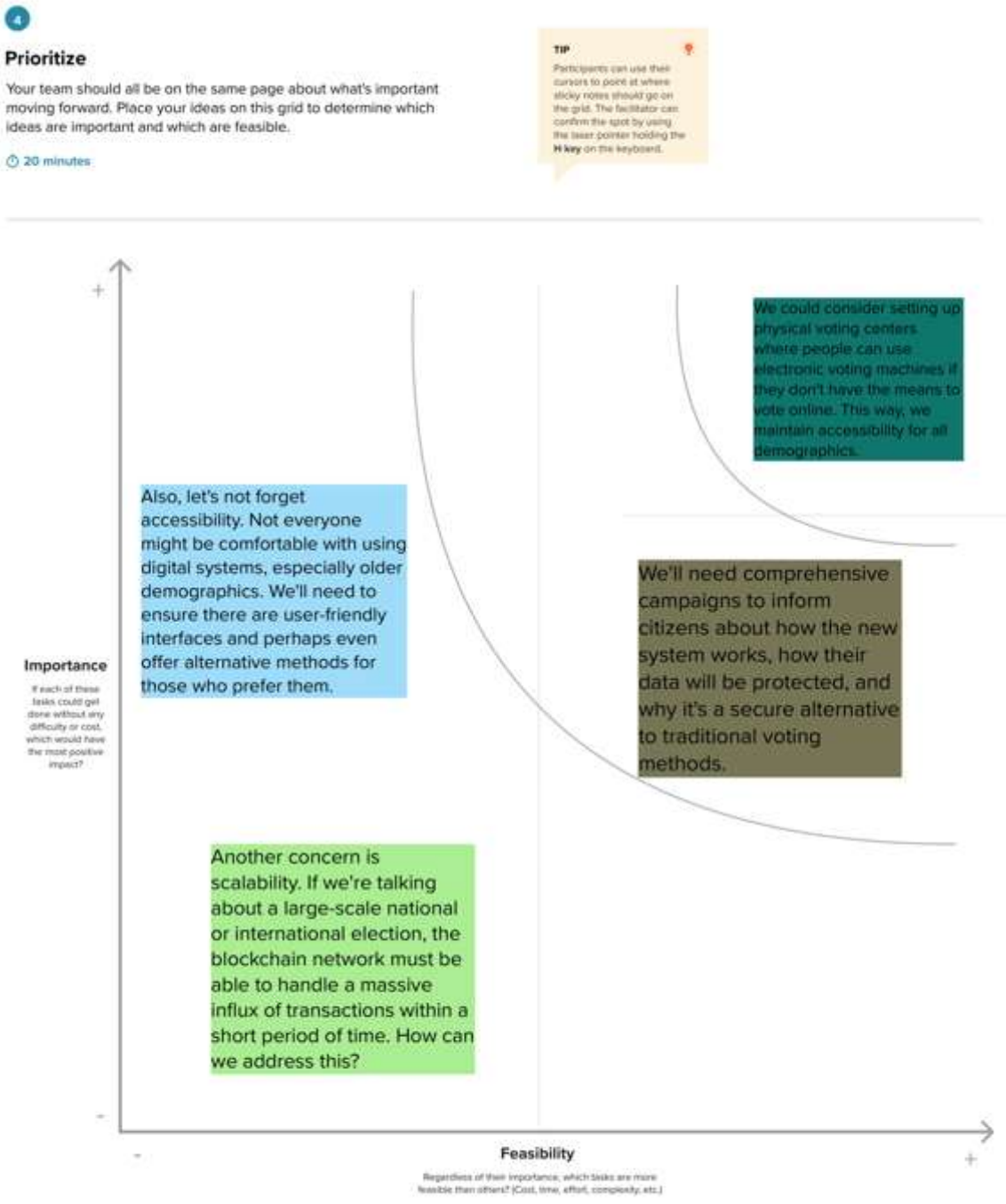
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Different countries have varying laws and regulations regarding elections. We'll need to work closely with legal experts to ensure compliance and address any potential conflicts.

We could consider setting up physical voting centers where people can use electronic voting machines if they don't have the means to vote online. This way, we maintain accessibility for all demographics. Participant 10:

It's clear that implementing an electronic voting system using blockchain technology is a complex task that requires careful consideration of various factors. Transparency, security, accessibility, anonymity, and legal compliance are all critical aspects that need to be addressed. Thank you all for your valuable contributions to this discussion.

Step-3: Idea Prioritization



3. REQUIREMENT ANALYSIS

3.1Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Voter Registration:	The system should allow eligible voters to register securely. Verify the eligibility of voters based on legal criteria.
FR-2	Authentication and Authorization:	Authenticate voters securely using biometric data, digital signatures, or secure tokens. Authorize eligible voters to cast their vote.
FR-3	Ballot Creation and Management:	Generate electronic ballots for each election. Manage different types of elections (e.g., local, national, referendums). Include candidate information and party affiliations.
FR-4	Vote Casting:	Enable voters to cast their votes electronically. Ensure one vote per eligible voter.
FR-5	Vote Verification and Transparency:	Allow voters to verify that their vote has been recorded correctly. Provide a transparent view of the entire voting process.
FR-6	Security and Privacy:	Implement strong encryption and cryptographic protocols to protect data in transit and at rest. Prevent tampering, fraud, and unauthorized access.

3.2Non-functional Requirements:

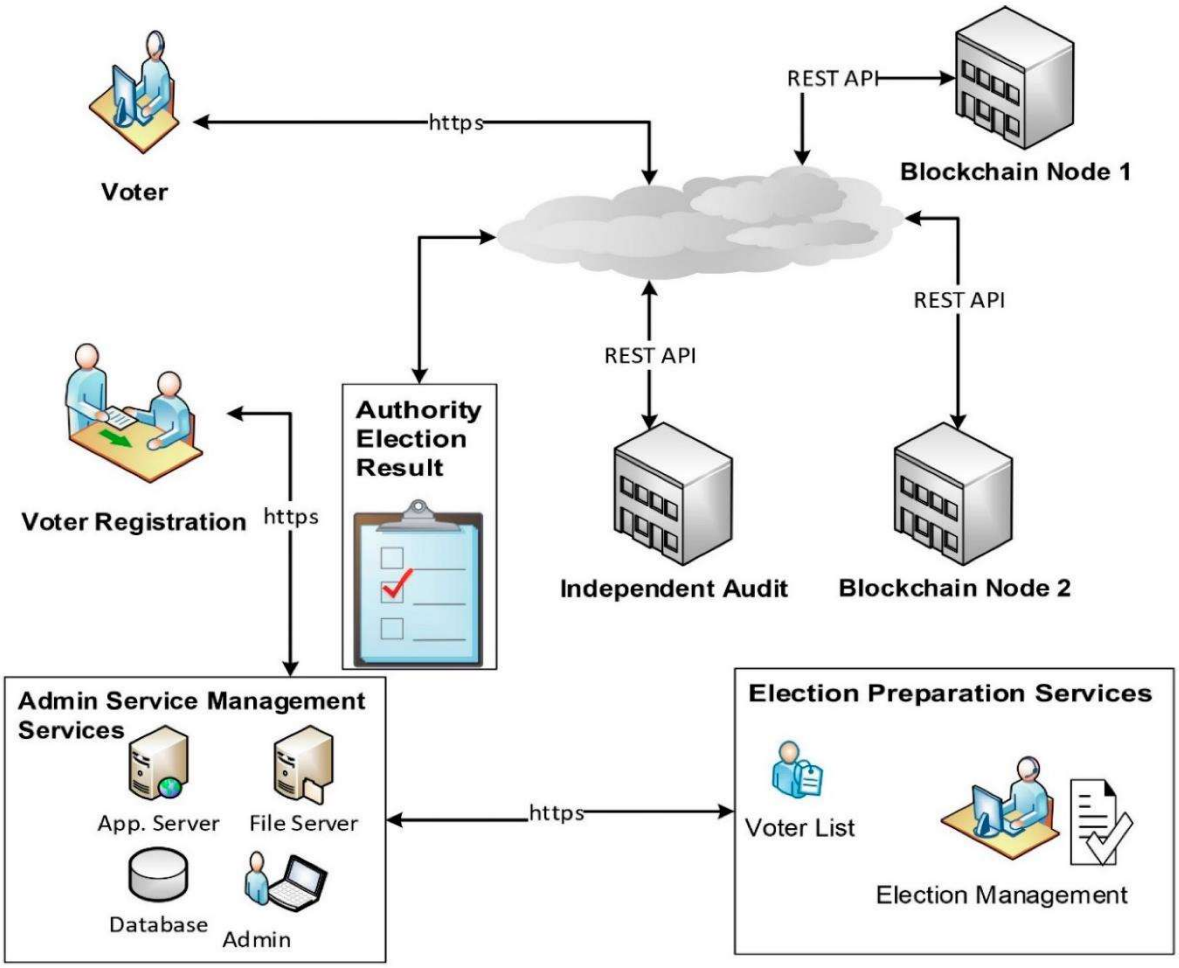
Following are the non-functional requirements of the proposed solution.

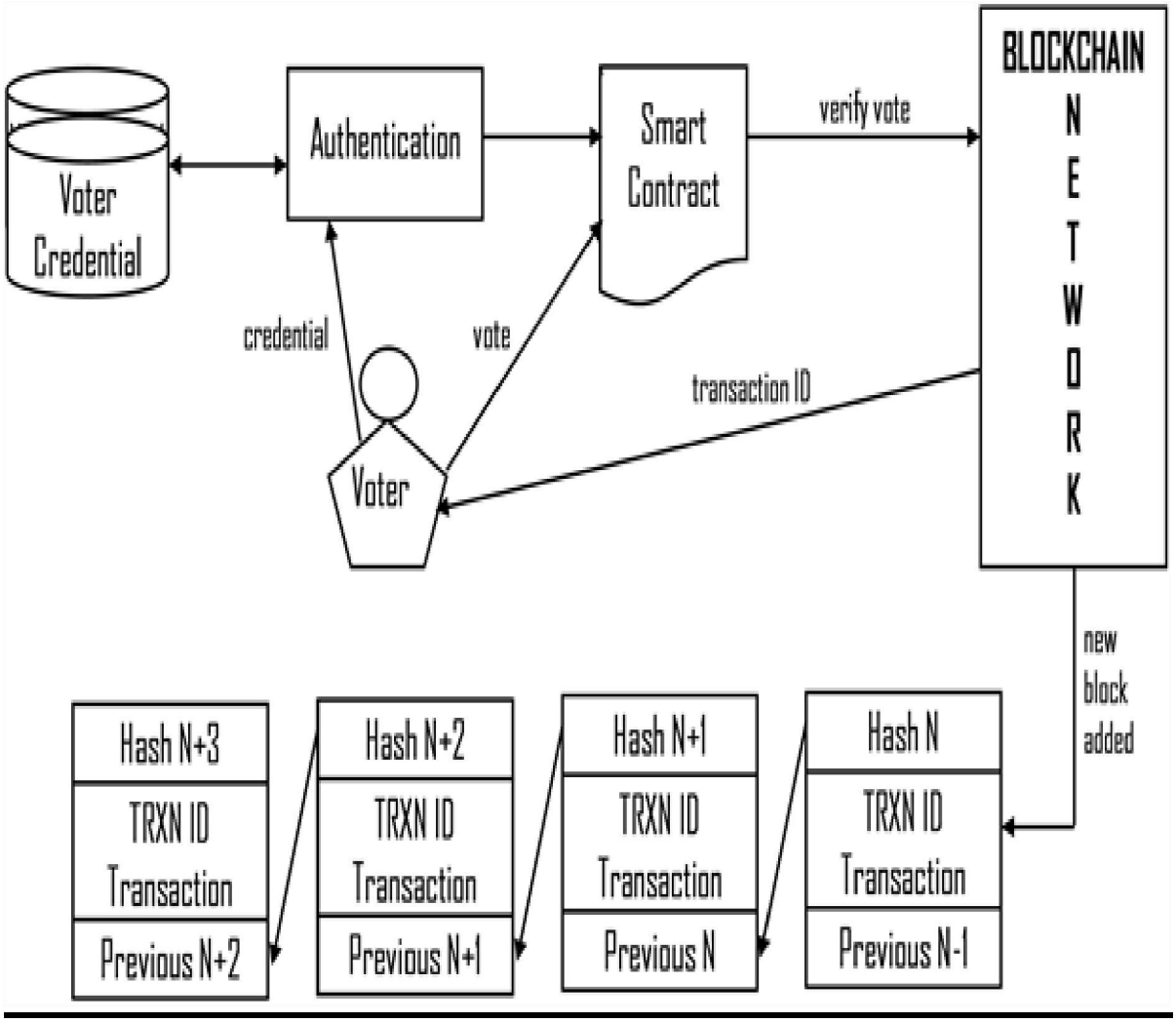
NFR No.	Non-Functional Requirement	Description
NFR-1	Security:	Ensure that the system is resistant to hacking, fraud, and other cyber threats. Implement multi-factor authentication and encryption techniques.
NFR-2	Reliability:	The system should be available and operational at all times, even under high load.
NFR-3	Performance:	The system should process votes efficiently and in real-time. Minimize latency in vote submission and result retrieval.
NFR-4	Scalability:	The system should be able to handle an increasing number of users and votes without compromising performance.
NFR-5	Usability:	The user interface should be intuitive and user-friendly, catering to a diverse range of users.
NFR-6	Maintainability:	The system should be easy to maintain, update, and debug.

4. PROJECT DESIGN

4.1Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of thesystem requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



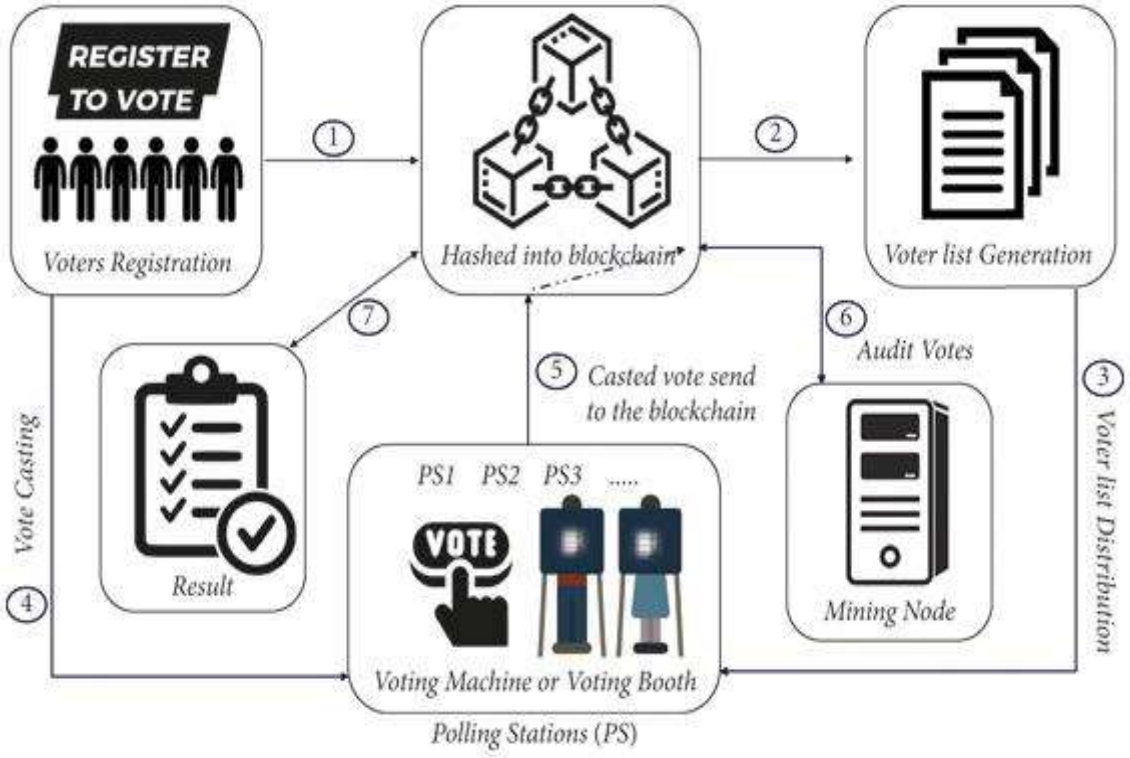

Voter Credential
Authentication
Smart Contract
verify vote
BLOCKCHAIN NETWORK
credential
vote
transaction ID
Voter
new block added
Hash N+3
TRXN ID Transaction
Previous N+2
Hash N+2
TRXN ID Transaction
Previous N+1
Hash N+1
TRXN ID Transaction
Previous N
Hash N
TRXN ID Transaction
Previous N-1

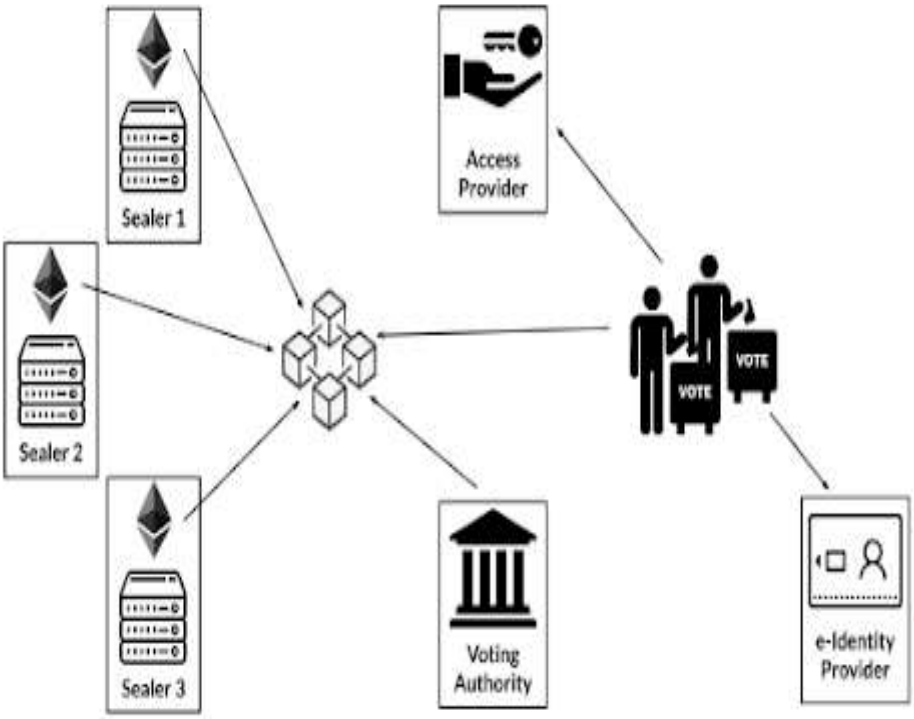
4.2Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gapbetween business problems and technology solutions.

Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed and delivered.




Sealer 1
Sealer 2
Sealer 3
Access Provider
Voting Authority
VOTE
VOTE
e-Identity Provider

4.3User Stories:

- As a registered voter, I want to securely cast my vote from the convenience of my own home, ensuring that my vote is counted accurately and anonymously.
- As a government official, I want to ensure that the electronic voting system maintains the integrity of the voting process by preventing any tampering or manipulation of the votes.
- As a voter, I want to receive a unique digital identity and authentication method to ensure that only eligible individuals can participate in the voting process.
- As a voter, I want to receive a confirmation receipt after submitting my vote, providing me with a verifiable record of my participation in the election.
- As an election administrator, I want to be able to easily verify the authenticity of a voter's identity and eligibility to prevent fraudulent votes from being counted.
- As a government official, I want to have a transparent and immutable ledger of all votes cast, accessible to authorized parties for auditing and verification purposes.
- As a voter, I want to be able to verify that my vote was correctly recorded and counted in the final tally without compromising the anonymity of my vote.
- As a government official, I want to ensure that the blockchain-based electronic voting system is resistant to DDoS attacks and other forms of cyber threats to maintain the integrity of the election process.
- As a voter, I want the electronic voting system to be user-friendly and accessible to individuals with disabilities, ensuring that everyone has an equal opportunity to participate.
- As an election administrator, I want to be able to efficiently manage and update the voter registration database to reflect changes in eligibility and prevent duplicate registrations.
- As a voter, I want to have the option to change my vote before the election deadline, providing flexibility in case I change my mind.
- As a government official, I want the blockchain-based electronic voting system to be interoperable with existing election infrastructure and standards to facilitate a smooth transition.
- As a voter, I want to receive notifications and updates about the election process, including important deadlines and instructions for participating in the electronic voting system.
- As a government official, I want to ensure that the electronic voting system complies with privacy regulations and data protection laws to safeguard voter information.
- As a voter, I want to have confidence in the security measures implemented in the electronic voting system, knowing that my vote is protected from any unauthorized access or manipulation.
- These user stories represent various perspectives and needs of stakeholders involved in an electronic voting system built on blockchain technology. They serve as a foundation for designing, developing, and testing the system to meet the requirements of all parties involved.

5.CODING & SOLUTIONING

Coding refers to the process of translating a software design into a functional program using programming languages like JavaScript, Python, or Java. It involves writing code, debugging, and optimizing algorithms to create a solution for a specific problem or requirement.

Solutioning involves designing a comprehensive solution strategy before coding. It includes problem analysis, architectural design, selecting appropriate technologies, and planning for scalability and security. Solutioning ensures that the software addresses the problem effectively and aligns with the project's goals.

5.1 FEATURE 1

Feature 1: User Authentication

```
function authenticateUser(username, password) {
// Code to validate username and password
        if (isValidCredentials(username, password)) {
                return "Authentication successful";
        }
        else {
                return "Authentication failed";
        }
}
function isValidCredentials(username, password) {
// Code to check credentials against database or backend
system
// Return true if valid, false otherwise
}
```

Explanation:

The code snippet checks the provided username and password against a database or backend system. If the credentials are valid, the function returns "Authentication successful"; otherwise, it returns "Authentication failed." This feature ensures secure user access to the system.

5.2FEATURE 2

Feature 2: Data Encryption

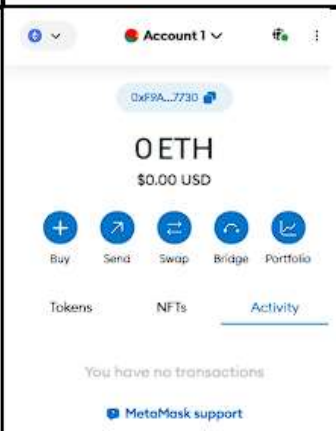
```
const crypto = require('crypto');
function encryptData(data, key) {
    const cipher = crypto.createCipher('aes-256-cbc', key);
    let encryptedData = cipher.update(data, 'utf-8', 'hex');
    encryptedData += cipher.final('hex');
    return encryptedData;
}
function decryptData(encryptedData, key) {
    const decipher = crypto.createDecipher('aes-256-cbc', key);
    let decryptedData = decipher.update(encryptedData, 'hex',
    'utf-8');
    decryptedData += decipher.final('utf-8');
    return decryptedData;
}
```

Explanation:

This code snippet demonstrates data encryption and decryption using the AES encryption algorithm. encryptData function takes data and a key, encrypts it, and returns the encrypted data in hexadecimal format. decryptData function reverses the process, decrypting the data using the same key.

6.RESULTS

6.1 Performance Metrics:

S.No.	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite:	
2.	Extract the zip files	Open to vs code	

3.	Remix Ide platform exploring	Deploy the smart contract code Deploy and run the transaction. By selecting the environment - inject the MetaMask.	
4	Open file explorer	Open the extracted file and click on the folder. Open src, and search for utiles. Open cmd enter commands 1.npm install 2.npm bootstrap 3. npm start	
5	{LOCALHOST IP ADDRESS	copy the address and open it to chrome so you can see the front end of your project.	

7. ADVANTAGES & DISADVANTAGES

Advantages:

1.Security and Transparency:

Immutable Ledger: Once a vote is recorded on the blockchain, it cannot be altered or deleted. This ensures the integrity of the voting process.

Transparency: The entire voting process, from registration to tallying, can be viewed in real-time by anyone with access to the blockchain.

2. **Reduced Fraud:**

Identity Verification: Blockchain can facilitate secure and transparent identity verification, making it more difficult for individuals to vote multiple times or impersonate others.

3.Accessibility:

Remote Voting: It can potentially enable remote voting from any location with an internet connection, increasing accessibility for people who might have difficulty physically reaching a polling place.

4.Efficiency:

Faster Results:

Vote counting and result declaration can be instantaneous since the process is automated and doesn't rely on manual counting.

5. Cost-Effective:

Reduced Administrative Costs:

Traditional voting systems involve significant administrative costs for printing, staffing, and physical infrastructure. Blockchain-based systems could potentially reduce these expenses.

6.Reduced Voter Suppression:

Accessibility for All:

It can potentially overcome barriers like long lines, transportation issues, or other logistical challenges that can discourage some individuals from voting. Increased Trust: Trust in the System: Knowing that their votes are securely recorded and cannot be tampered with can increase public trust in the electoral process.

Disadvantages:

1.Technical Challenges:

Technological Barriers: Not everyone has access to the internet or is comfortable using digital technology. Implementing a blockchain-based voting system might exclude some demographics.

2. **Security Concerns:** Private Key Management: If voters don't secure their private keys properly, it could lead to hacking or unauthorized access to their votes. 51% Attack: If a malicious entity gains control of the majority of the blockchain's computing power, it could potentially manipulate the results.

3.Voter Anonymity:

Balancing Transparency and Privacy: Striking the right balance between transparent voting and maintaining voter privacy can be challenging.

Legal and Regulatory Challenges: Compliance with Election Laws: Adapting blockchain voting to comply with existing election laws and regulations could be complex and time-consuming.

4.System Reliability:

Downtime and Failures: Technical glitches, network outages, or cyberattacks could disrupt the voting process.

5.Lack of Paper Trail:

Auditability: Unlike traditional paper-based systems, blockchain-based voting may lack a physical paper trail, which can be important for auditing and recounts.

6.Resistance to Change:

Trust in New Technology: Convincing the public and authorities to trust a new, relatively untested technology for something as critical as elections could be a significant challenge.

8. CONCLUSION

1.An electronic voting system utilizing blockchain technology holds significant promise for revolutionizing the way we conduct elections. This innovative approach addresses many of the challenges that traditional voting systems face, including concerns about security, transparency, and trustworthiness.

2.Blockchain's inherent properties, such as decentralization, immutability, and transparency, provide a robust foundation for building a secure and tamper-proof voting system. By distributing the ledger across a network of nodes, it becomes extremely difficult for any single entity to manipulate or alter the voting records.

3. Moreover, cryptographic techniques ensure that individual votes remain anonymous, protecting the privacy of voters. The use of smart contracts can also automate various aspects of the voting process, streamlining the verification and tabulation of results.

4.Despite these advantages, it's important to acknowledge that implementing an electronic voting system using blockchain is not without its challenges. Technical issues, such as scalability, user interface design, and accessibility for all demographics, need to be carefully considered and addressed.

5.Additionally, ensuring widespread adoption and trust in such a system requires extensive education and outreach efforts to build confidence among voters, political stakeholders, and the public at large.

6.In conclusion, an electronic voting system utilizing blockchain technology has the potential to enhance the integrity, security, and efficiency of electoral processes. With careful planning, rigorous testing, and transparent deployment, it could pave the way for a more inclusive and democratic future. However, it's imperative to approach the implementation with caution, taking into account the complexities and potential pitfalls associated with such a transformative technology.

9. FUTURE SCOPE

- **Enhanced Security and Transparency:** Blockchain provides a highly secure and transparent way to record and verify transactions. Each vote would be recorded as a unique transaction, making it extremely difficult for malicious actors to tamper with the results.
- **Immutable Voting Records:** Once a vote is recorded on the blockchain, it becomes immutable, meaning it cannot be altered or deleted. This ensures the integrity of the voting process.
- **Elimination of Voter Fraud**: Blockchain's cryptographic algorithms and consensus mechanisms make it very difficult for anyone to manipulate or forge votes. This can significantly reduce the risk of voter fraud.
- **Increased Accessibility:** Electronic voting systems can potentially make it easier for people to vote, especially those who may face challenges with physical access to polling places.
- **Remote Voting:** With proper security measures in place, blockchain-based electronic voting systems could enable remote voting, allowing people to vote from the comfort of their own homes.
- **Real-Time Results:** Because blockchain transactions are recorded in real-time, election results could be available almost instantly after the voting period concludes, providing faster and more efficient reporting.

10. APPENDIX

1.Introduction

- Background Provide an overview of the need for electronic voting systems and the benefits of using blockchain technology.
- Objectives List the specific goals of the electronic voting system, such as enhancing security, transparency, and accessibility.

2.System Architecture

- Components Describe the key components of the system, including: User Interface Blockchain Network Smart Contracts Backend Servers
- Interaction Flow Explain how each component interacts with the others in the system.

3.Blockchain Technology

- Overview Briefly explain what blockchain is and how it works.
- Blockchain for Voting Describe why blockchain is suitable for electronic voting, emphasizing features like immutability, decentralization, and transparency.

4.System Design

- User Registration Explain the process of registering voters and associating their identities with a unique identifier on the blockchain.
- Casting Votes Detail how a voter casts their vote, how it is encrypted, and how it is recorded on the blockchain.
- Counting Votes Describe how the votes are counted, emphasizing the

transparency and security measures in place.

- Results Verification Explain how anyone can independently verify the election results using the blockchain.

5.Security Measures

- Encryption and Decryption Explain the use of cryptographic techniques to secure the voting process.
- Identity Verification Detail how the system ensures that only eligible voters can participate.
- Double Voting Prevention Describe mechanisms in place to prevent a voter from casting multiple votes.

6.Privacy Considerations

- Voter Anonymity Explain how the system maintains voter privacy while still ensuring the integrity of the process.
- Transparency vs. Privacy Balance Discuss the trade-off between transparency and voter privacy.

7.Appendix

- Smart Contract Code Provide sample code snippets for the smart contracts used in the system.
- Data Structures Explain the data structures used in the blockchain for storing voter information and votes.
- Scalability Considerations Discuss how the system can handle a large number of votes and voters.
- Legal and Regulatory Compliance Outline the legal and regulatory considerations that need to be addressed when implementing the system.
- Testing and Deployment Detail the testing procedures and steps for deploying the system in a real-world election.

11.SOURCE CODE

Folder Structure:

Name	Date modified	Type	Size
voting-app	25-10-2023 18:39	File folder	

Name	Date modified	Type	Size
.git	25-10-2023 18:39	File folder	
node_modules	26-10-2023 14:41	File folder	
public	25-10-2023 18:39	File folder	
src	25-10-2023 18:27	File folder	
.gitignore	25-10-2023 18:27	Git Ignore Source ...	1 KB
package.json	25-10-2023 18:27	JSON Source File	1 KB
package-lock.json	26-10-2023 14:41	JSON Source File	1,222 KB
postcss.config.js	25-10-2023 18:27	JavaScript Source ...	1 KB
README.md	25-10-2023 18:27	Markdown Source ...	1 KB
tailwind.config.js	25-10-2023 18:27	JavaScript Source ...	1 KB

Name	Date modified	Type	Size
components	25-10-2023 18:27	File folder	
images	25-10-2023 18:27	File folder	
pages	25-10-2023 18:27	File folder	
utils	25-10-2023 18:27	File folder	
App.css	25-10-2023 18:27	CSS Source File	1 KB
App.js	25-10-2023 18:27	JavaScript Source ...	1 KB
App.test.js	25-10-2023 18:27	JavaScript Source ...	1 KB
index.css	25-10-2023 18:27	CSS Source File	1 KB
index.js	25-10-2023 18:27	JavaScript Source ...	1 KB
logo.svg	25-10-2023 18:27	Microsoft Edge HT...	3 KB
reportWebVitals.js	25-10-2023 18:27	JavaScript Source ...	1 KB
setupTests.js	25-10-2023 18:27	JavaScript Source ...	1 KB

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the
build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

Manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

App.css

```css
.App {
  text-align: center;

}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }

  to {
    transform: rotate(360deg);
  }
}
```

App.js

```
import "./App.css";
import Home from "./pages/Home";
import { Navbar } from "./components/Navbar";

function App() {
    return (
        <div className="App">
            <Navbar />
            <Home />
        </div>
    );
}

export default App;
```

Home.js

```
import React from "react";
import Voting from "../components/Voting";

function Home() {
    return <Voting />;
}

export default Home;
```

App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Constans.js

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

export const contractAddress = "0xd9145CCE52D386f254917e481eB44e9943F39138";

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

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

Index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
    background-color: #d6dbe4;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

@tailwind base;
@tailwind components;
@tailwind utilities;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Reportwebvitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) =>
{
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

Setuptest.js

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

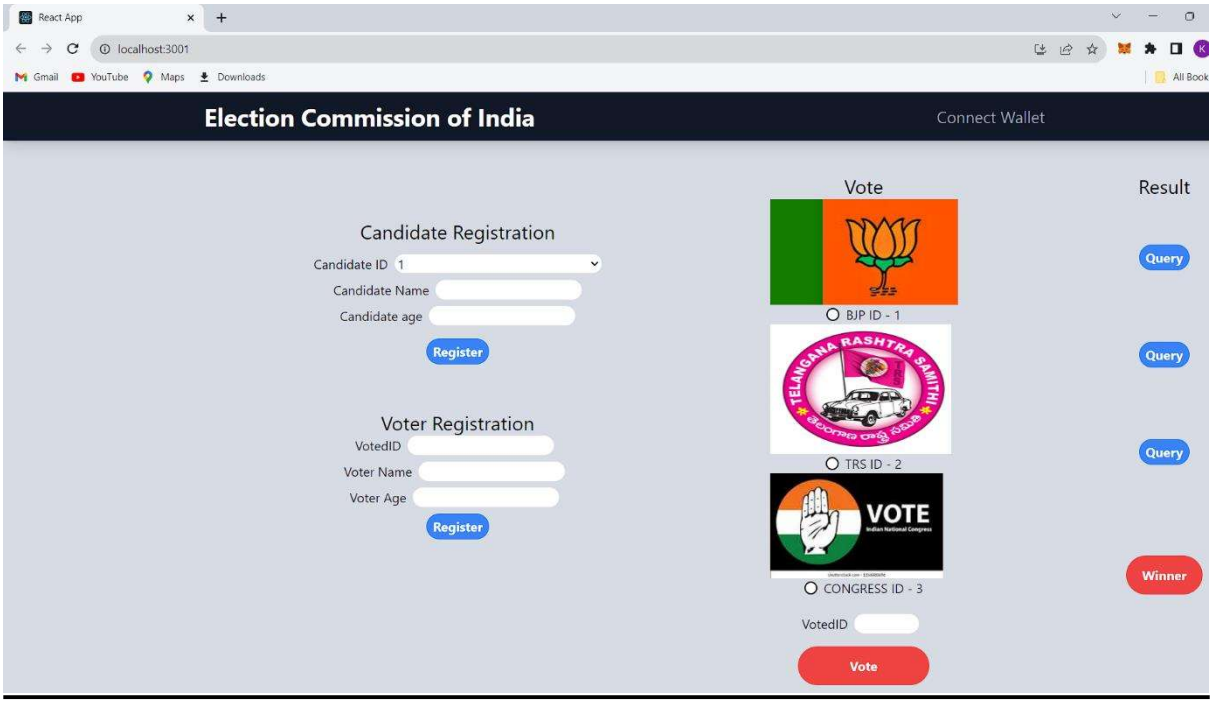
Package.json

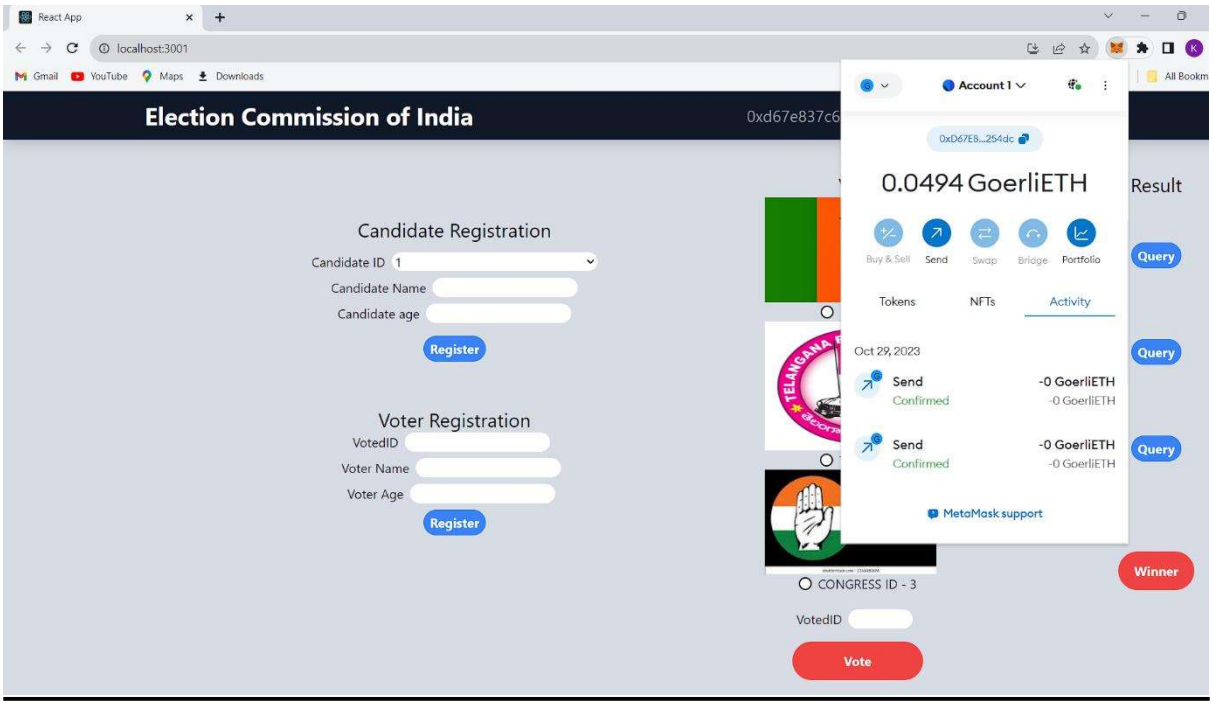
```
{
  "name": "voting-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "ethers": "^5.7.2",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "autoprefixer": "^10.4.13",
    "postcss": "^8.4.19",
    "tailwindcss": "^3.2.4"
  }
}
```

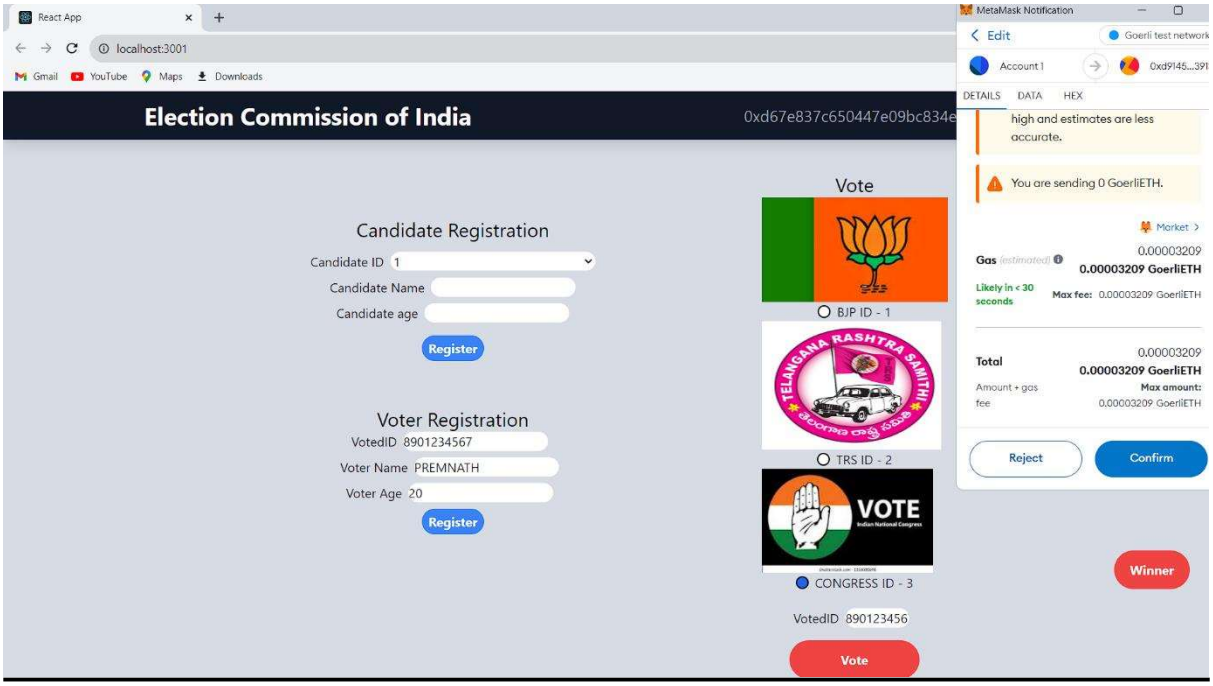
GitHub Link:
https://github.com/Rajesh12345qwe/Electronic-Voteing-System

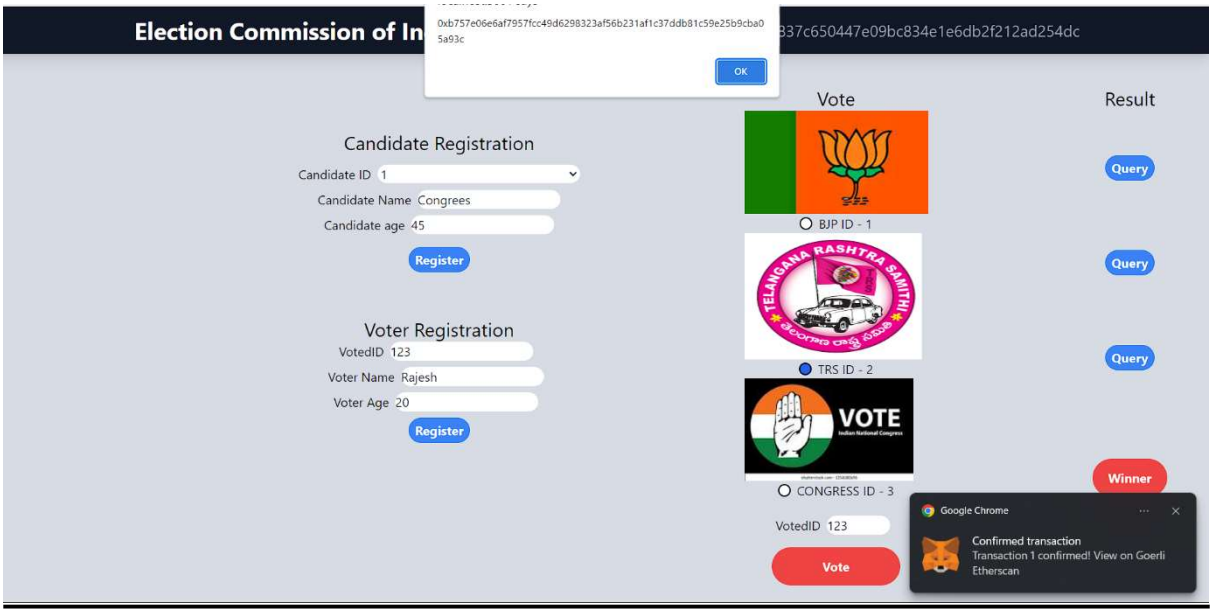
Demo Link: https://drive.google.com/file/d/1-ljEqFcLNvz87QE3EYJfA6ydqmOVJiWz/view?usp=drive_link

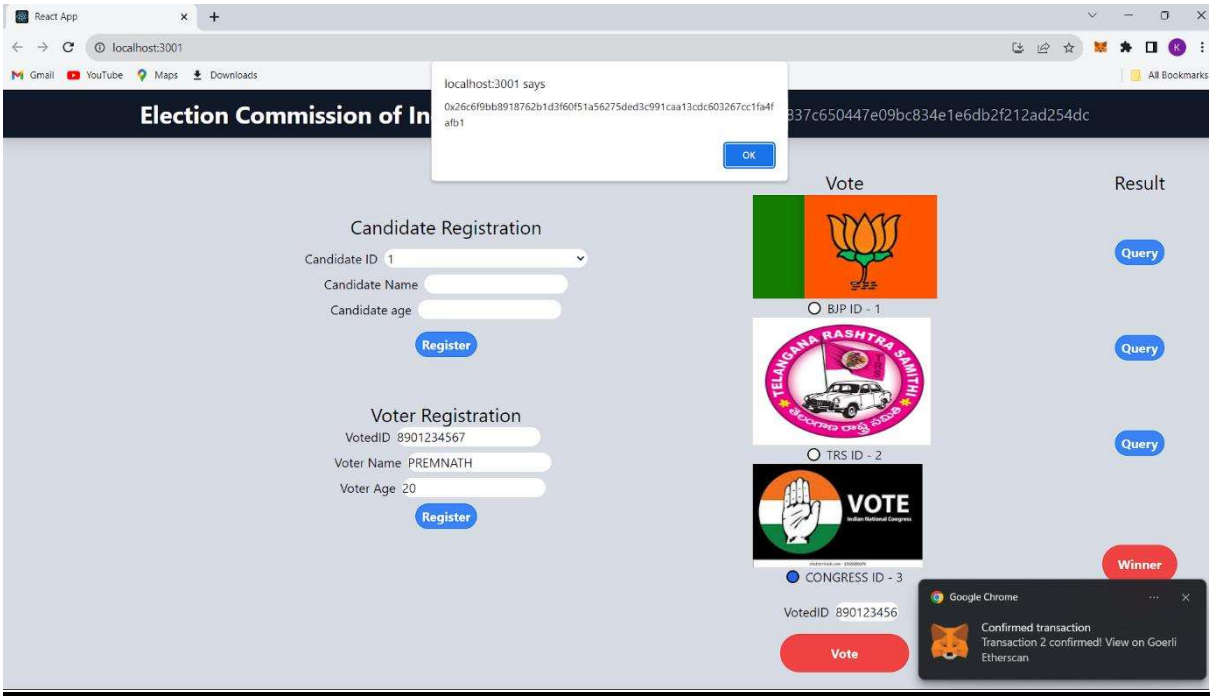
11.OUTPUT SCREENSHOTS






React App
localhost:3001
Gmail
YouTube
Maps
Downloads
Election Commission of India
0xd67e837c650447e09bc834e
Candidate Registration
Candidate ID 1
Candidate Name
Candidate age
Register
Voter Registration
VotedID 8901234567
Voter Name PREMNATH
Voter Age 20
Register
Vote
BJP ID - 1
TRS ID - 2
CONGRESS ID - 3
VotedID 890123456
Vote
MetaMask Notification
Edit
Goerli test network
Account 1
0xd9145...391
DETAILS
DATA
HEX
high and estimates are less accurate.
You are sending 0 GoerliETH.
Market
Gas (estimated)
0.00003209
0.00003209 GoerliETH
Likely in < 30 seconds
Max fee: 0.00003209 GoerliETH
Total
0.00003209
0.00003209 GoerliETH
Amount + gas fee
Max amount:
0.00003209 GoerliETH
Reject
Confirm
Winner


Election Commission of In
0xb757e06e6af7957fcc49d6298323af56b231af1c37ddb81c59e25b9cba05a93c
OK
337c650447e09bc834e1e6db2f212ad254dc
Candidate Registration
Candidate ID 1
Candidate Name Congrees
Candidate age 45
Register
Voter Registration
VotedID 123
Voter Name Rajesh
Voter Age 20
Register
Vote
BJP ID - 1
TRS ID - 2
CONGRESS ID - 3
VotedID 123
Vote
Result
Query
Query
Query
Winner
Google Chrome
Confirmed transaction
Transaction 1 confirmed! View on Goerli Etherscan


React App
localhost:3001
Gmail
YouTube
Maps
Downloads
All Bookmarks
localhost:3001 says
0x26c6f9bb8918762b1d3f60f51a56275ded3c991caa13cdc603267cc1fa4fafb1
OK
Election Commission of In
837c650447e09bc834e1e6db2f212ad254dc
Vote
Result
Candidate Registration
Candidate ID 1
Candidate Name
Candidate age
Register
BJP ID - 1
Query
TELANGANA RASHTRA SAMITHI
TRS ID - 2
Query
Voter Registration
VotedID 8901234567
Voter Name PREMNATH
Voter Age 20
Register
VOTE
CONGRESS ID - 3
Winner
VotedID 890123456
Vote
Google Chrome
Confirmed transaction
Transaction 2 confirmed! View on Goerli Etherscan