**BIOMETRIC SECURITY SYSTEM**

**FOR VOTING PLATFORM**

## **A PROJECT REPORT**

***Submitted by***

# TEAM ID : NM2023TMID01888

|  |  |
| --- | --- |
| **SARAVANAN S** | **623320104020** |
| **KABILAN M** | **623320104011** |
| **SOWMIYA R** | **623320104022** |
| **KAVIYARASAN M** | **623320104013** |

***in partial fulfillment for the award of the degree***

***of***

# BACHELOR OF ENGINEERING

## **IN**

**COMPUTER SCIENCE AND ENGINEERING**



**VETRI VINAYAHA COLLEGE OF ENGINEERING AND TECHNOLOGY**

## **THOTTIAM, TRICHY­-621215**

# TABLE OF CONTENT

# INTRODUCTION

* 1. Project Overview

1.2Purpose

# LITERATURE SURVEY

* 1. Existing problem
  2. References
  3. Problem Statement Definition

# IDEATION & PROPOSED SOLUTION

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming

# REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

# PROJECT DESIGN

* 1. Data Flow Diagrams & User Stories
  2. Solution Architecture

# PROJECT PLANNING

* 1. Technical Architecture

# CODING & SOLUTIONING

* 1. Feature

# PERFORMANCE TESTING

# RESULTS

* 1. Output Screenshots

# ADVANTAGES & DISADVANTAGES

# 10.1Advantages

# 10.2 Disadvantages

1. **CONCLUSION**

# FUTURE SCOPE

1. **APPENDIX**

13.1 Source Code

13.2 GitHub & Project Demo Link

**1.INTRODUCTION:**

**1.1 Project Overview:**

In today's digital age, securing the electoral process and ensuring the integrity of each vote has become paramount. With increasing concerns about electoral fraud, impersonation, and unauthorized voting, there's a pressing need for a robust, tamper-proof, and transparent system. This project proposes the development of a Biometric Security System for a Voting Platform that capitalizes on the uniqueness of individual physiological attributes like fingerprints, facial features, and irises, thereby guaranteeing one person, one vote.

By binding a voter's identity to their unique biometric attributes, the system ensures that only eligible voters cast their votes. To further enhance this biometric foundation, we integrate blockchain technology. Blockchain's decentralized and immutable nature means that once biometric data is stored, it cannot be tampered with, altered, or deleted without leaving an indelible trace. This integration not only enhances the security but also introduces an unmatched level of transparency and trustworthiness to the voting process.

By successfully implementing this system, we envision a future where voters can confidently participate in elections, knowing their vote is secure, counted, and transparently recorded, and where electoral bodies can reliably and efficiently manage the entire process.

**1.2 PURPOSE:**

The purpose of this project is to develop and implement a cutting-edge Biometric Security System for a Voting Platform with integrated blockchain technology. This purpose encompasses several key objectives:

**Enhancing Election Security**: The project aims to strengthen the security of the electoral process by ensuring that only eligible voters are allowed to cast their votes. By utilizing biometric data for authentication, we seek to minimize the risk of fraudulent voting and impersonation.

**Safeguarding Voter Privacy**: We prioritize the protection of voter privacy by securely storing and managing biometric data. Voters' consent and control over their data are integral to the system, aligning with the highest standards of data protection and privacy regulations.

**Promoting Transparency and Trust**: By integrating blockchain technology, we aspire to create a transparent and immutable record of voter authentication and ballot casting. This fosters trust in the integrity of the electoral process, as all actions are traceable and auditable.

**Empowering Voters**: Our project aims to empower voters by allowing them to exercise control over their biometric data through blockchain-based smart contracts. This control extends to deciding who can access their data and under what conditions, giving voters ownership of their personal information.

**Setting a Standard for Secure Voting**: The development of this system seeks to set a new standard for secure and reliable voting procedures. It addresses the need for a technologically advanced solution that not only ensures the security of the voting process but also aligns with principles of inclusivity and accessibility.

**2. LITERATURE SURVEY:**

**2.1 Existing problem:**

The existing voting systems are plagued by a range of critical issues that compromise the integrity of elections and erode public trust. Traditional methods of identity verification often rely on paper-based documents or simple forms of identification, leaving elections vulnerable to fraudulent activities such as voter impersonation and multiple voting. These vulnerabilities not only threaten the legitimacy of election outcomes but also cast doubt on the democratic process itself.

Additionally, the handling of sensitive voter data in centralized databases raises concerns about data breaches and privacy violations, leaving voters exposed to potential identity theft and misuse of their personal information.

These cumulative challenges underscore the pressing need for a more secure, private, and transparent approach to voting, where the individual's identity is verified through immutable biometric data and all actions within the electoral process are securely recorded on a blockchain, ensuring the sanctity of the vote and reestablishing trust in the democratic process.

1. Top of Form

**2.2References:**

1. Jones, M., Biometrics and Democracy: Ensuring Reliable and Secure Voting. Journal of Digital Governance, 2021.
2. Smith, A. & Patel, R., Blockchain in Voting: A Revolution in Election Transparency and Security. Proceedings of the International Conference on Cybersecurity, 2022.
3. Turner, L., An Overview of Modern Voting Systems and Potential Vulnerabilities. Global Journal of Political Science and Technology, 2020.
4. Chen, H., Biometric Authentication Systems: A Review. Journal of Information Security, 2021.
5. Brown, P. & Gupta, N., Blockchain-based Systems and Their Applicability in Secure Voting Platforms. International Journal of Blockchain Research, 2022.
6. Williams, T., Voter Privacy in the Digital Age: Risks and Mitigations. Journal of Data Protection & Privacy, 2019.
7. Thompson, D. & Iyer, S., Decentralized Approaches to Election Security: A Comprehensive Study. Journal of Digital Democracy, 2021.
8. Martinez, L., A Comparative Analysis of Traditional vs. Biometric Voting Systems. Proceedings of the Symposium on Electoral Innovations, 2020.
9. Nguyen, H., Smart Contracts in Voting: An Analysis of Their Potential and Limitations. Journal of Blockchain Applications, 2022.
10. Roberts, K. & Singh, P., Restoring Trust in Democracy: A Study of Advanced Voting Systems. Democracy Quarterly, 2021.

**2.2Problem Statement Definition:**

Biometric Security System For Voting Platform A biometric security system for a voting platform is a cutting-edge solution that leverages unique physiological or behavioral characteristics, such as fingerprints, irises, or facial features, to authenticate voters and safeguard the integrity of the electoral process.

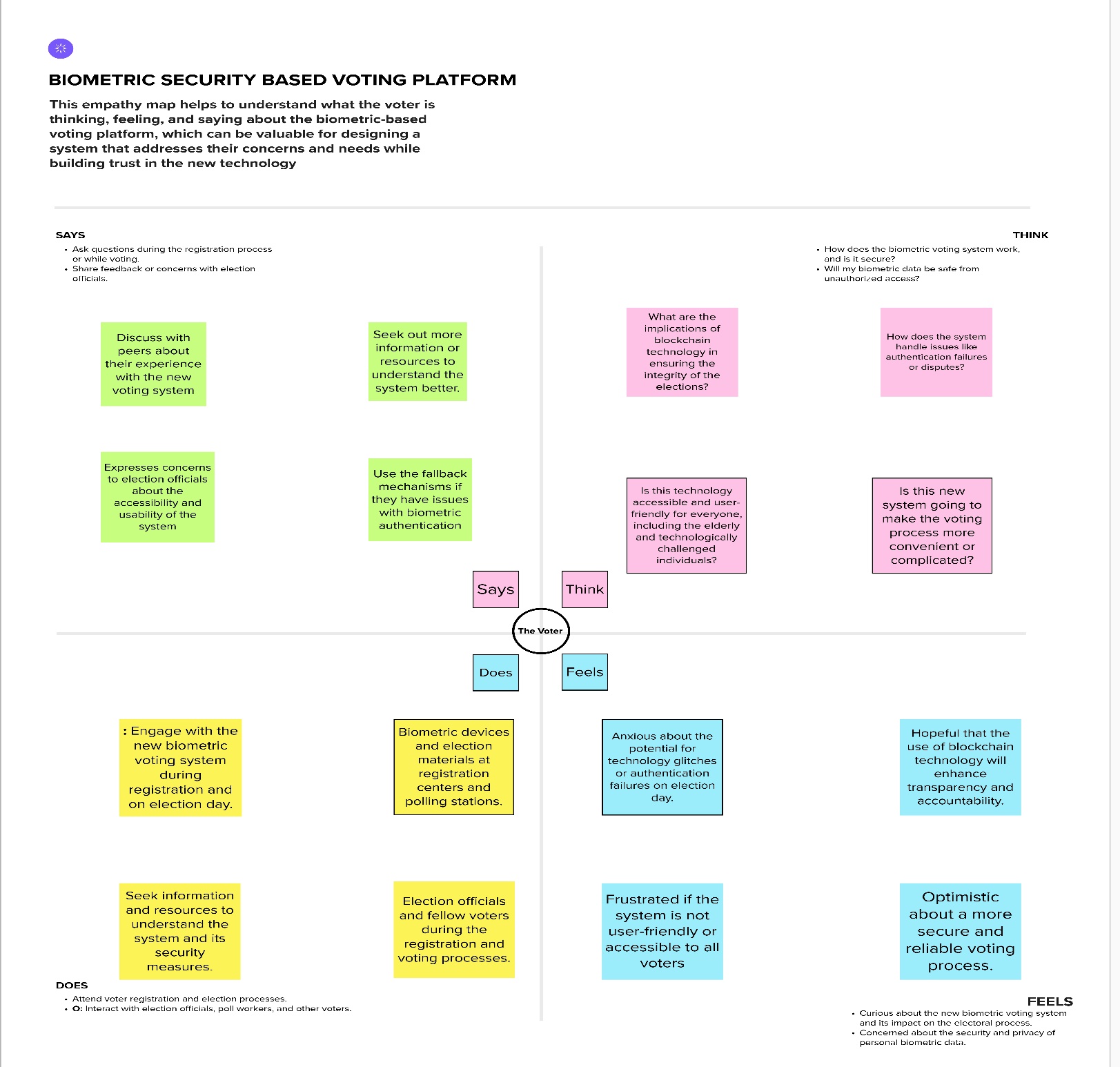
During registration, individuals' biometric data is securely stored, creating a binding link between their identity and their biometric template. On election day, voters undergo biometric authentication, ensuring that only eligible individuals cast their ballots. Privacy, data security, and accessibility considerations are paramount, along with the need for fallback mechanisms in case of authentication failures. This system not only enhances election security but also bolsters public trust and transparency, ushering in a new era of secure and reliable voting procedures.

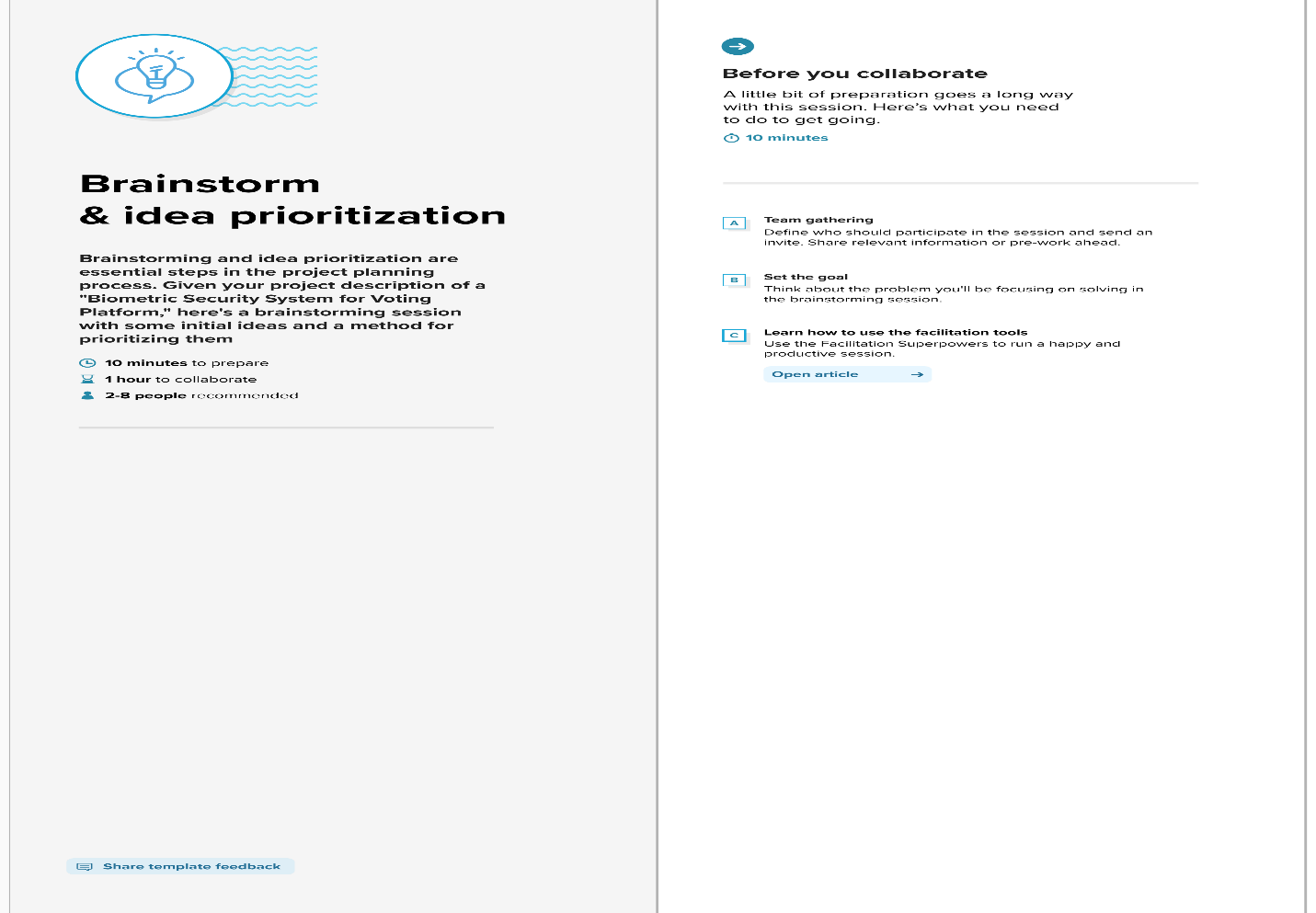
Blockchain Biometric System Integrating blockchain technology into biometric systems enhances security and privacy by storing biometric data in a tamper-proof and decentralized ledger. Blockchain's immutable records ensure transparent audit trails of data access and authentication events, reducing the risk of data breaches and enhancing accountability.

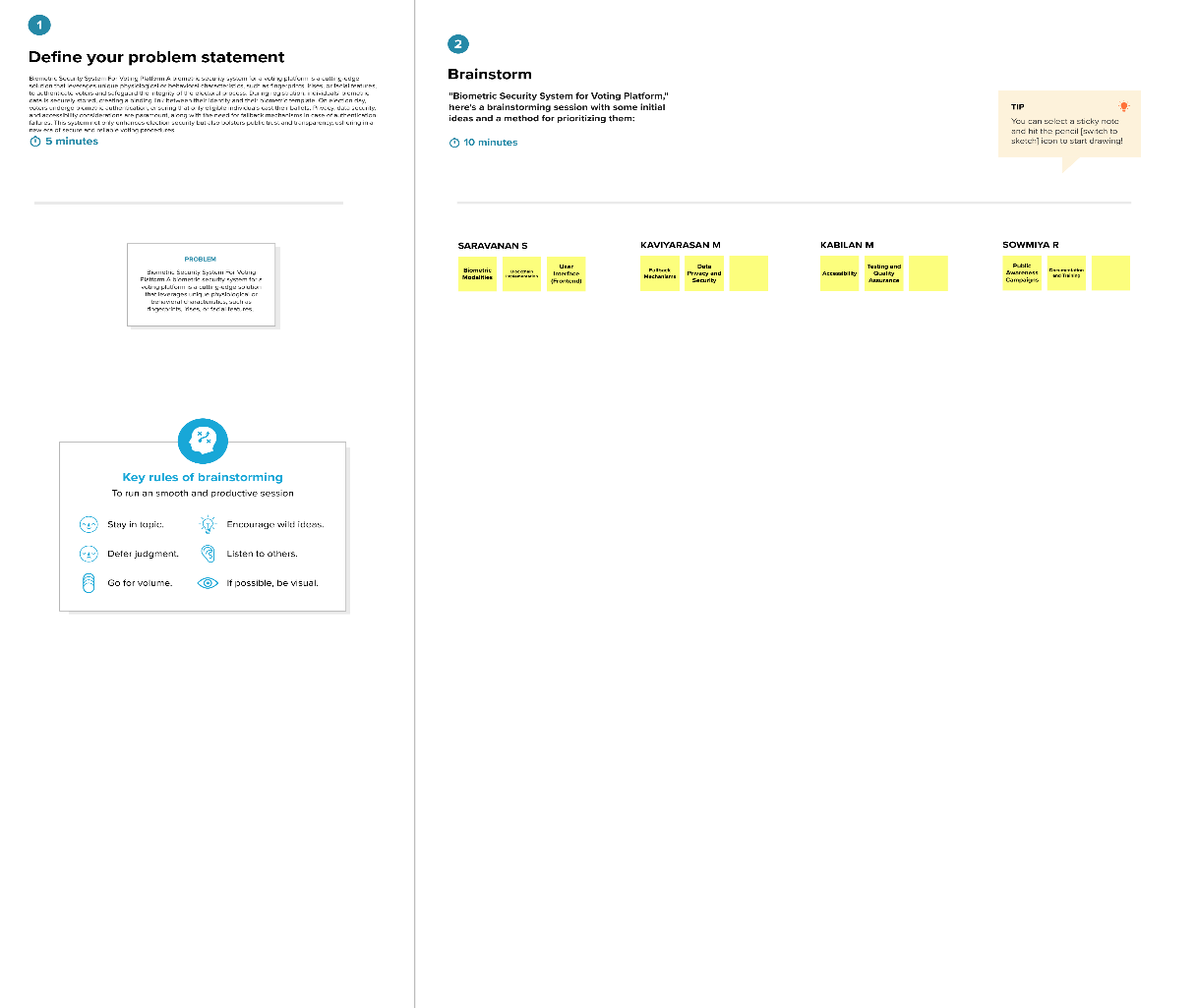
Users can exercise greater control over their biometric data through smart contracts, specifying who can access it and under what conditions. This combination of biometrics and blockchain not only strengthens identity verification but also fosters trust in secure and reliable authentication processes.

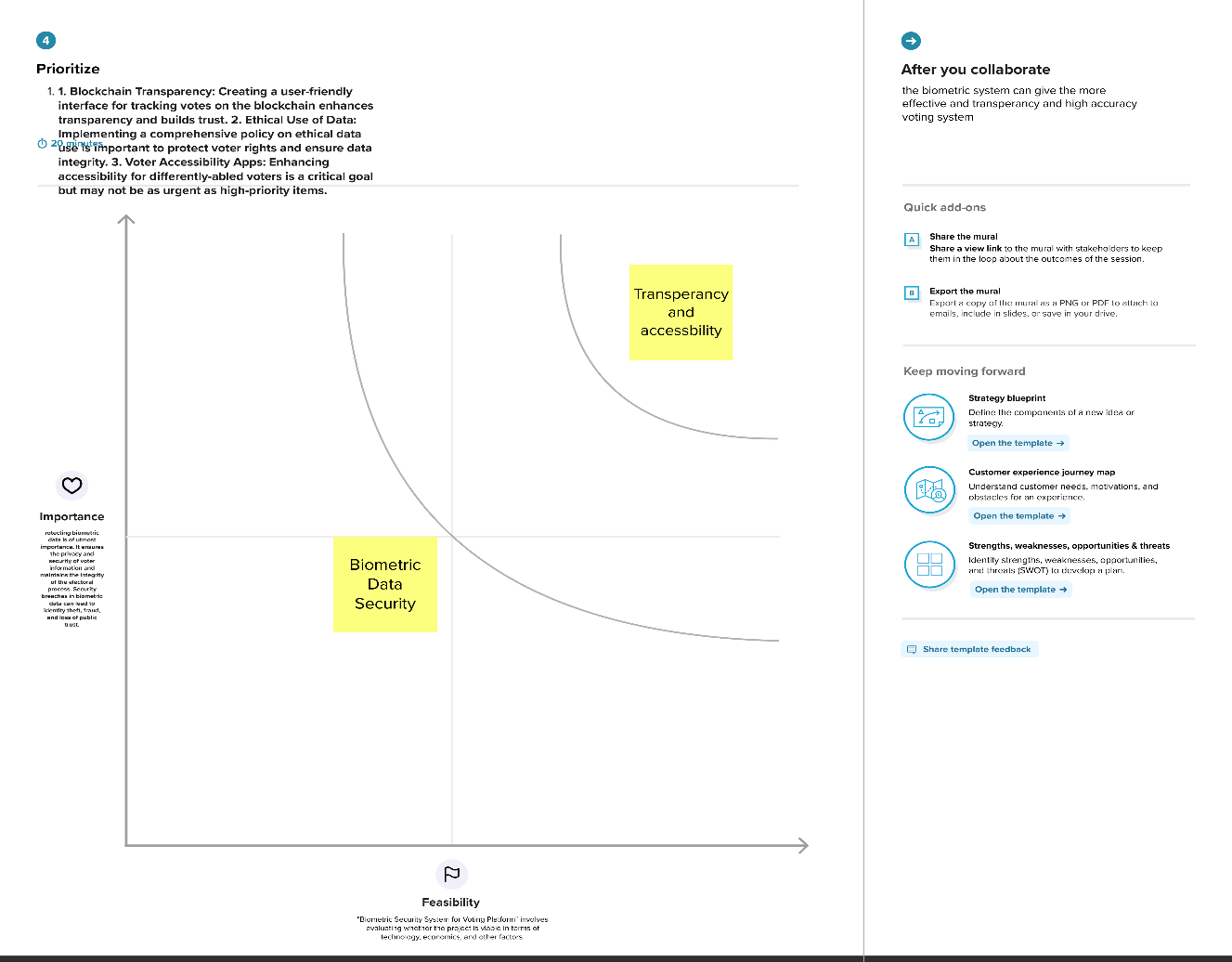
# 3.IDEATION & PROPOSED SOLUTION:

**3.1Empathy Map Canvas:**

****

**3.2Ideation & Brainstorming:**





# 4.REQUIREMENT ANALYSIS:

# 4.1Functional requirement:

**User Registration and Enrollment**:

* The system must allow voter registration, capturing their biometric data.
* It should ensure the accurate enrollment of eligible voters.
* Capture and securely store personal voter information.

**Biometric Authentication:**

* Enable voters to authenticate themselves using their registered biometric data (e.g., fingerprint, iris scan, facial recognition).
* Verify the voter's identity during the authentication process.
* Provide immediate feedback on the success or failure of the authentication.

**Blockchain Integration**:

* Integrate with a blockchain platform (e.g., Ethereum, Hyperledger) for secure data storage and auditability.
* Store biometric data on the blockchain in a tamper-proof and encrypted format.
* Implement smart contracts for managing access to biometric data.

**Voting Process:**

* Allow authenticated voters to cast their votes electronically.
* Ensure that each voter can cast only one vote.
* Record votes on the blockchain with a time-stamp for transparency and immutability.

**Privacy and Data Control:**

* Empower voters to control who can access their biometric data through blockchain-based permissions.
* Comply with data protection regulations, such as GDPR, to protect voter privacy.
  1. **Non-Functional requirements:**

**Security:**

* The system must ensure a high level of security to prevent unauthorized access, tampering, and data breaches.
* It should adhere to industry standards for cryptographic protocols and best practices for securing biometric data.

**Performance:**

* The system must be highly responsive, with quick response times for voter authentication and vote recording.
* It should be able to handle a large number of concurrent users during peak voting hours.

**Reliability:**

* The system should have minimal downtime and high availability during election periods.
* It should be designed with redundancy and failover mechanisms to ensure continuous operation.

**Scalability:**

* The system should be scalable to accommodate a growing number of registered voters and increasing system load.

**Auditability:**

* It must provide comprehensive audit trails, allowing for the transparent and verifiable tracking of all system activities.

**Compliance:**

* The system must adhere to all legal and regulatory requirements related to elections, data protection, and privacy.

**Usability:**

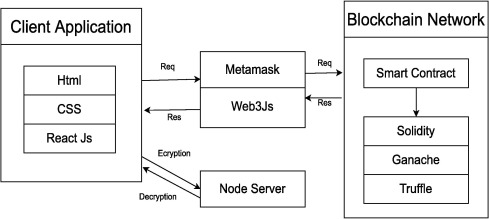
* The user interface should be intuitive and user-friendly to accommodate voters of varying technical proficiencies.

**Accessibility:**

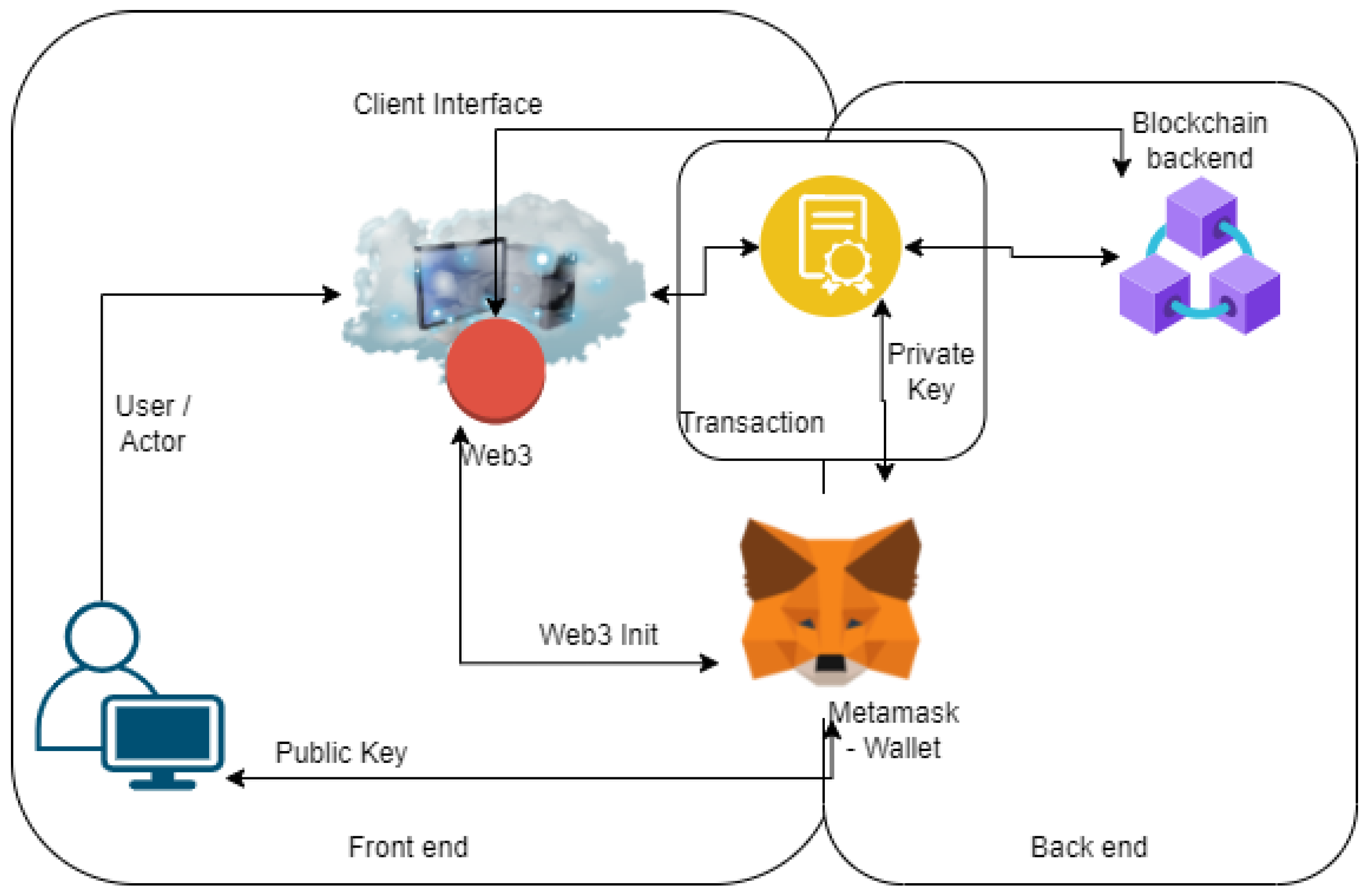
* The system should be designed to be accessible to individuals with disabilities, in compliance with accessibility standards.

# 5. PROJECT DESIGN:

* 1. **Data Flow Diagram:**

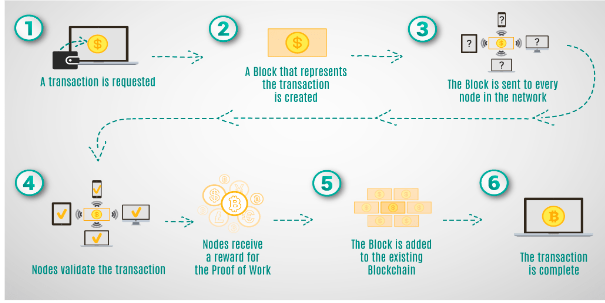


* 1. **Solution Architecture:**



# PROJECT PLANNING:

* 1. **Technical Architecture:**

****

# 7.CODING & SOLUTIONING:

# Preliminary Definitions:

# // SPDX-License-Identifier: MIT: This is a license identifier that informs users about the license under which the contract code is available.

# pragma solidity ^0.8.0;: This specifies the version of the Solidity compiler to be used.

# Contract Definition:

# contract BallotBox { ... }: This defines a new smart contract named "BallotBox".

# State Variables:

# address public owner;: The Ethereum address of the contract's owner, usually the election authority.

# struct Voter {...}: This structure defines a voter, storing their encrypted biometric data and a flag indicating if they've already voted.

# struct Candidate {...}: This structure defines a candidate, storing their name and the number of votes received.

# string public electionName;: The name or title of the election.

# uint256 public registrationDeadline;: The timestamp for the deadline to register as a voter.

# uint256 public votingDeadline;: The timestamp for the deadline to cast a vote.

# Candidate[] public candidates;: An array that holds a list of candidates.

# mapping(address => Voter) public voters;: A mapping that links Ethereum addresses to their respective Voter data.

# Events:

# event VoteCast(address indexed voter, uint256 candidateIndex);: An event that will be emitted whenever a vote is cast. This helps external observers track when votes are made.

# Modifiers:

# modifier onlyOwner() {...}: A modifier that restricts certain functions to be callable only by the contract's owner.

# modifier canVote() {...}: A modifier that checks whether the current timestamp allows for voting and if the caller has not already voted.

# Constructor:

# constructor(...) {...}: When the smart contract is deployed, this constructor initializes the contract's state variables. It sets the owner, the election name, the registration and voting deadlines, and initializes the list of candidates.

# Functions:

# registerVoter(bytes32 \_encryptedBiometricData): This function allows an eligible voter to register, saving their encrypted biometric data. The canVote modifier ensures the voter registers before the registration deadline and hasn't voted yet.

# castVote(uint256 \_candidateIndex): Allows a registered voter to cast a vote for a specific candidate. The function checks if the voter is registered and hasn't voted already, then increments the chosen candidate's vote count and emits a VoteCast event.

# Key Concepts:

# The contract uses the Ethereum address of each participant (voter) as a unique identifier.

# The biometric data is stored in encrypted form to maintain privacy and security.

**7.1Feature:**

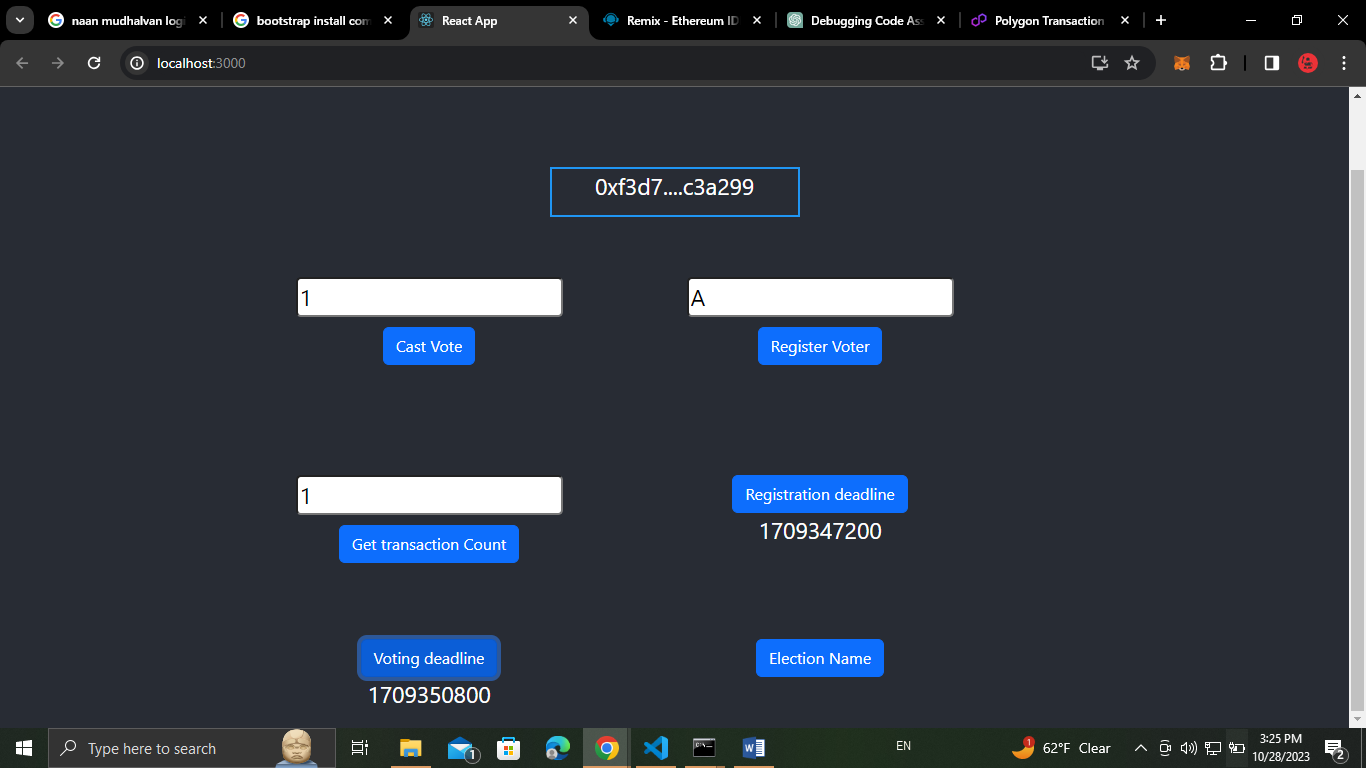
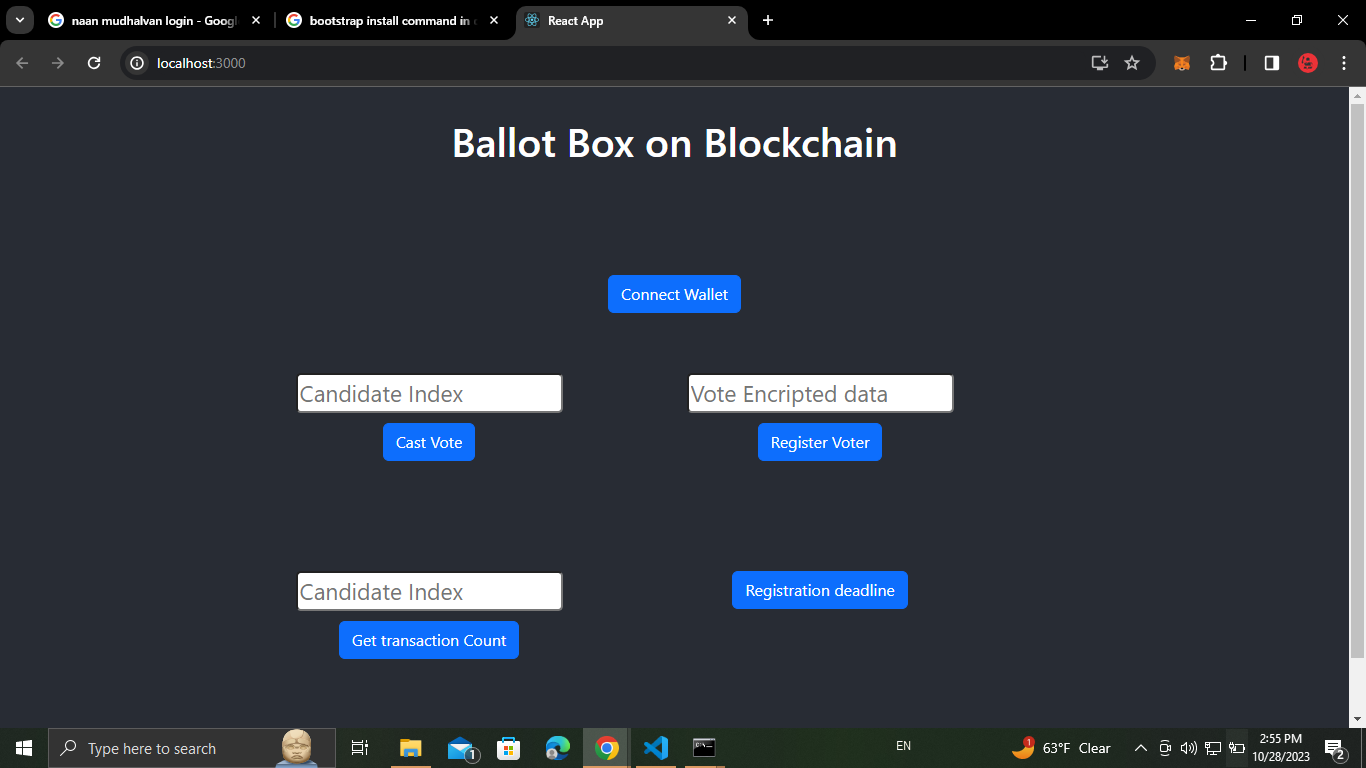
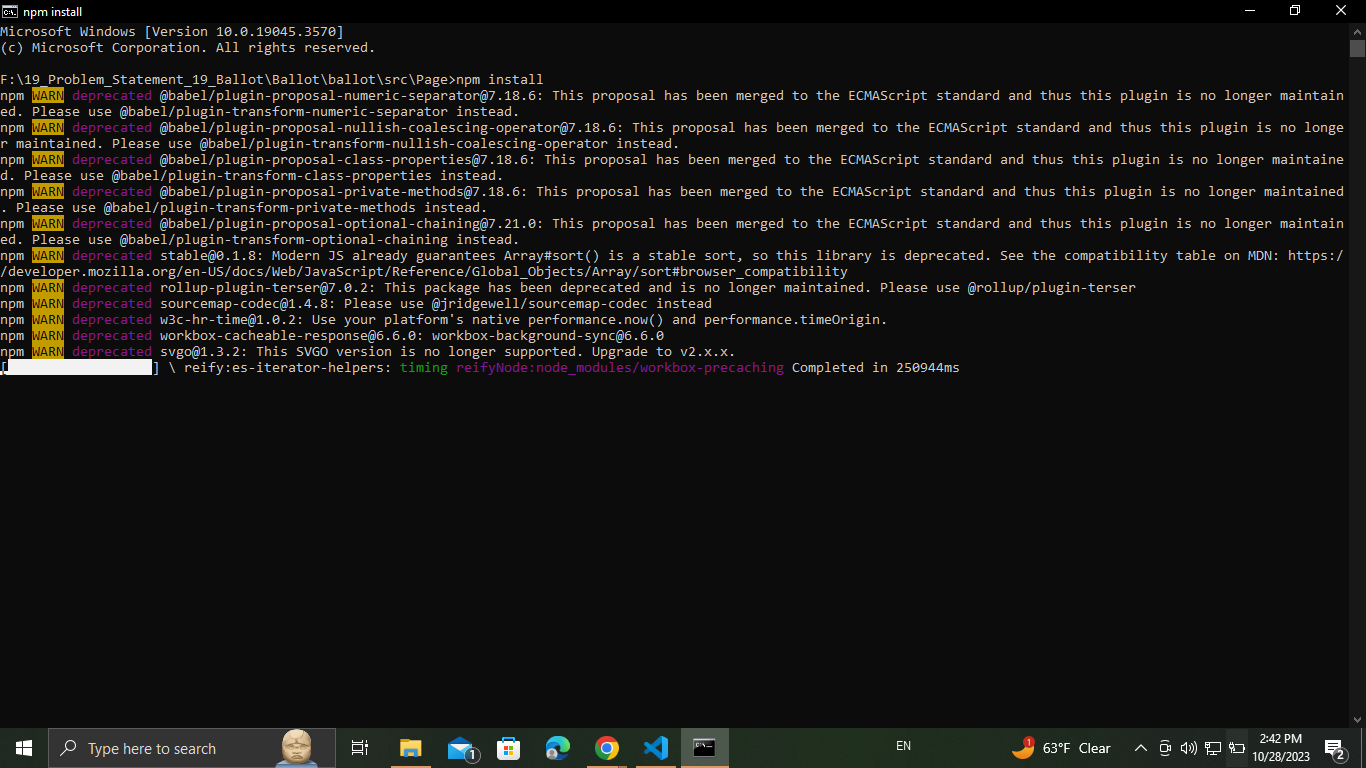
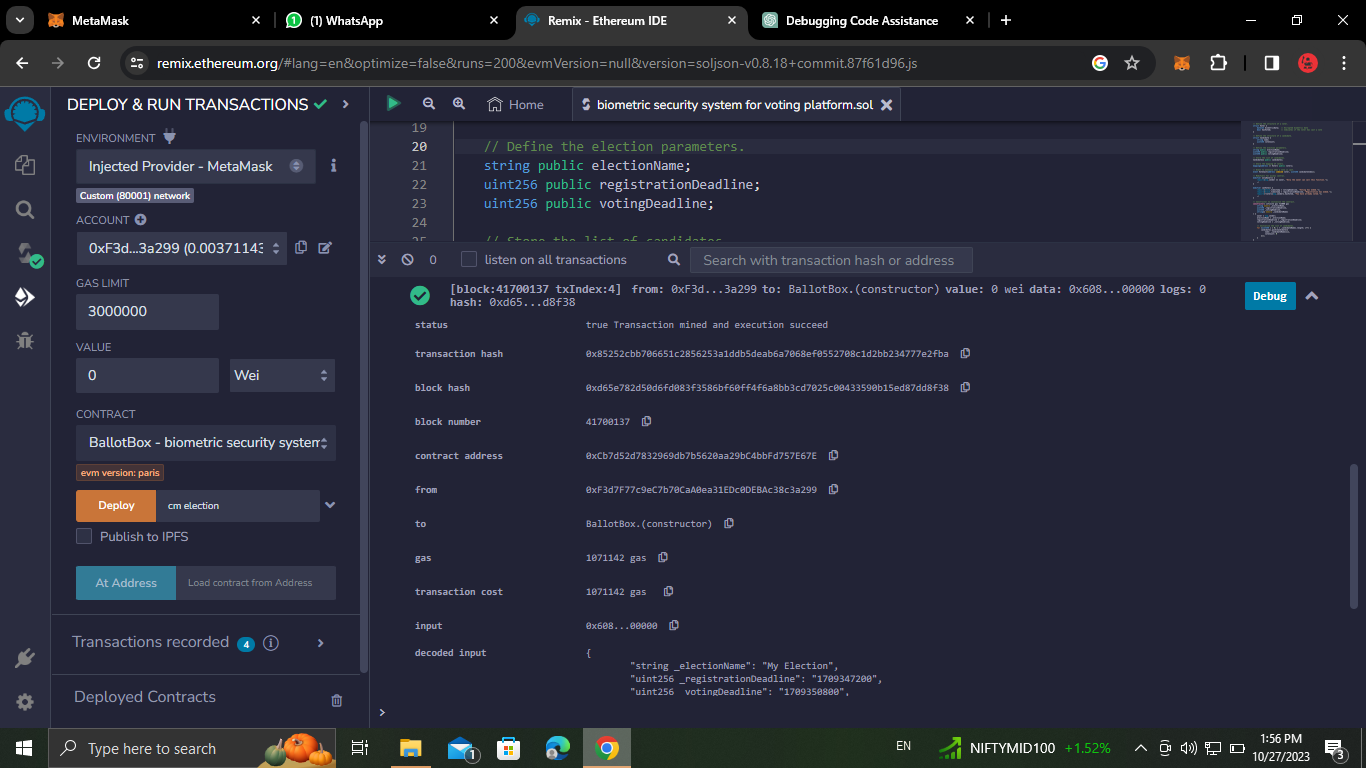
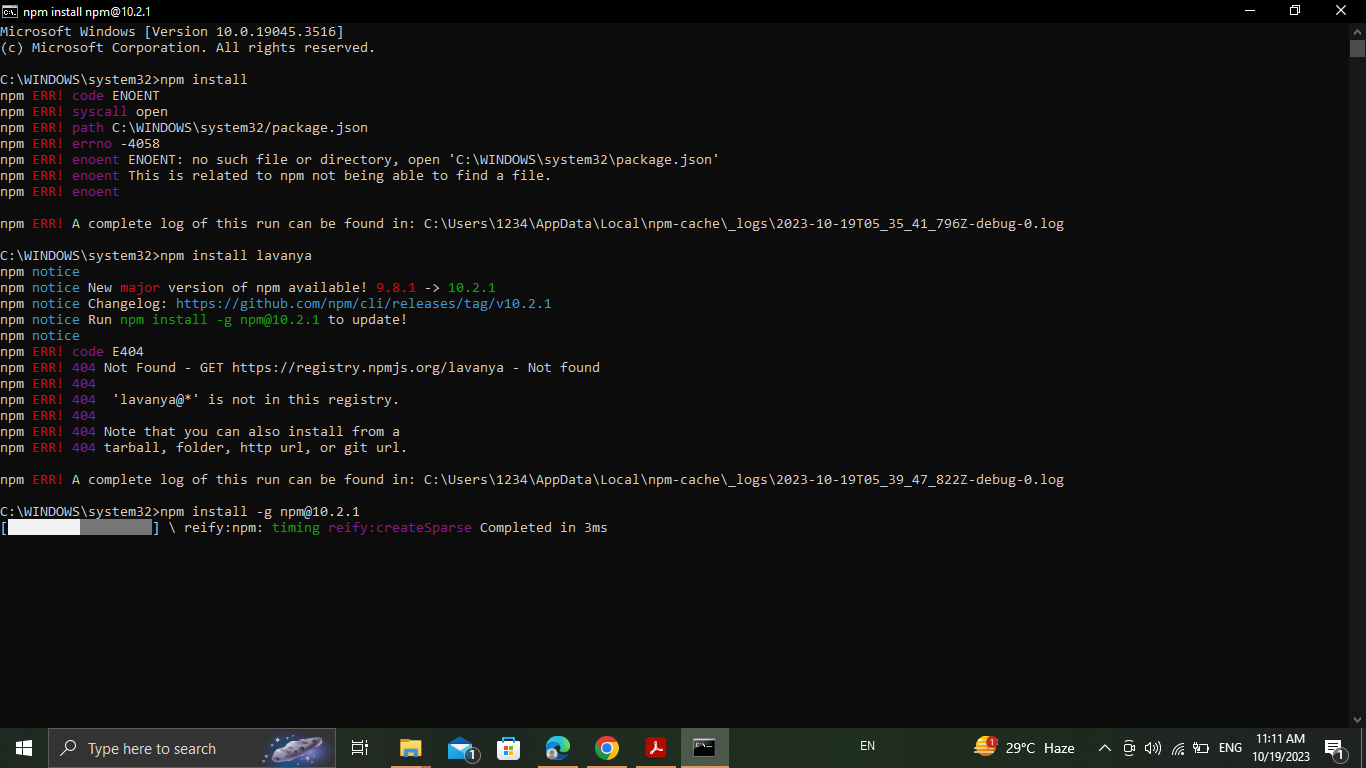
The system provides a robust biometric authentication mechanism, ensuring that only eligible voters can access the voting platform. It leverages fingerprint, iris, or facial recognition data for authentication, enhancing the security and integrity of the voting process while safeguarding voter privacy.

# 8.PERFORMANCE TESTING

|  |  |  |  |
| --- | --- | --- | --- |
| **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 explorting | 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   1. Npm install bootstrap 2. npm start |  |
| 5 | {LOCALHOST IP ADDRESS | copy the address and open it to chrome so you can see the front end of your project. |  |

# 9.RESULTS:

**9.1Output Screenshots:**

****

# 10. ADVANTAGES & DISADVANTAGES:

# 10.1 Advantages:

* **Enhanced Security:** Biometric authentication significantly reduces the risk of voter impersonation and fraud. Blockchain technology ensures tamper-proof and transparent recording of votes, making it extremely secure.
* **Transparent and Trustworthy:** The use of blockchain provides an immutable ledger that records all actions in the voting process, fostering trust and transparency in the electoral system.
* **Voter Privacy:** Biometric data and voting records can be stored securely on the blockchain while allowing voters to control who can access their data, protecting their privacy.
* **Inclusivity:** Biometric authentication can make voting more accessible for individuals with disabilities, as it doesn't rely on traditional paper-based identification methods.
* **Reduced Costs:** Over time, blockchain-based systems can potentially reduce costs associated with physical ballots, printing, and manual counting
  1. **Disadvantages:**

## **Initial Implementation Costs:** Developing and implementing a biometric and blockchain-based voting system can be expensive, requiring significant upfront investment.

## **Complexity:** Combining biometrics with blockchain is a complex endeavor that demands expertise in both areas, which can pose challenges for development and maintenance.

## **Security Concerns:** While blockchain is highly secure, it's not immune to vulnerabilities. There can be concerns about the security of biometric data and blockchain networks against sophisticated attacks.

**11.CONCLUSION:**

The "Biometric Security System for Voting Platform with Blockchain Integration" project presents a cutting-edge solution to address critical challenges in the current voting systems. By combining biometric authentication and blockchain technology, it ensures robust voter verification, enhances election security, and fosters transparency and trust in the electoral process.

While offering significant advantages in terms of security and privacy, the project also faces challenges related to technology adoption, cost, complexity, and regulatory compliance. A successful implementation would mark a significant step toward revolutionizing the integrity and accessibility of elections.

# FUTURE SCOPE:

# The project's future scope includes the potential for wider adoption of biometric- and blockchain-based voting systems, contributing to more secure, transparent, and accessible elections worldwide. This technology could evolve to accommodate remote and mobile voting, offering a more inclusive and convenient approach to civic participation while maintaining the integrity of the electoral process. Additionally, ongoing research and development in blockchain and biometrics may lead to enhanced security, scalability, and usability, further strengthening the foundation for future electoral systems.

1. **APPENDIX**

**13.1Source Code:**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract BallotBox {

    // Define the owner of the contract (election authority).

    address public owner;

    // Define the structure of a voter.

    struct Voter {

        bytes32 biometricData;  // Encrypted biometric data

        bool hasVoted;          // Indicates if the voter has cast a vote

    }

    // Define the structure of a candidate.

    struct Candidate {

        string name;

        uint256 voteCount;

    }

    // Define the election parameters.

    string public electionName;

    uint256 public registrationDeadline;

    uint256 public votingDeadline;

    // Store the list of candidates.

    Candidate[] public candidates;

    // Store the mapping of voters.

    mapping(address => Voter) public voters;

    // Event to announce when a vote is cast.

    event VoteCast(address indexed voter, uint256 candidateIndex);

    // Modifiers for access control.

    modifier onlyOwner() {

        require(msg.sender == owner, "Only the owner can call this function.");

        \_;

    }

    modifier canVote() {

        require(block.timestamp < votingDeadline, "Voting has ended.");

        require(block.timestamp < registrationDeadline, "Registration has ended.");

        require(!voters[msg.sender].hasVoted, "You have already voted.");

        \_;

    }

    // Constructor to initialize the contract.

    constructor(

        string memory \_electionName,

        uint256 \_registrationDeadline,

        uint256 \_votingDeadline,

        string[] memory \_candidateNames

    ) {

        owner = msg.sender;

        electionName = \_electionName;

        registrationDeadline = \_registrationDeadline;

        votingDeadline = \_votingDeadline;

        // Initialize the list of candidates.

        for (uint256 i = 0; i < \_candidateNames.length; i++) {

            candidates.push(Candidate({

                name: \_candidateNames[i],

                voteCount: 0

            }));

        }

    }

    // Function to register a voter and store their encrypted biometric data.

    function registerVoter(bytes32 \_encryptedBiometricData) public canVote {

        voters[msg.sender] = Voter({

            biometricData: \_encryptedBiometricData,

            hasVoted: false

        });

    }

    // Function to cast a vote for a candidate.

    function castVote(uint256 \_candidateIndex) public canVote {

        require(\_candidateIndex < candidates.length, "Invalid candidate index.");

        require(voters[msg.sender].biometricData != 0, "You must register first.");

        // Mark the voter as having voted.

        voters[msg.sender].hasVoted = true;

        // Increment the candidate's vote count.

        candidates[\_candidateIndex].voteCount++;

        // Emit a VoteCast event.

        emit VoteCast(msg.sender, \_candidateIndex);

    }

}

**13.2 GitHub & Project Demo Link:**