Project Report

Team ID	NM2023TMID05253
Project Name	BIOMETRIC SECURITY SYSTEM FOR VOTING PLATFORM

Submitted by

TEAM LEADER : MANOKAR G

TEAM MEMBER 1 : DHINESHKUMAR A

TEAM MEMBER 2 : VASANTHAKUMAR D

PROJECT REPORT FORMAT

1.INTRODUCTION
1.1 Project Overview
1.2 Purpose
2.LITERATURE SURVEY
2.1 Existing problem
2.2 References
2.3 Problem Statement Definition
3.IDEATION & PROPOSED SOLUTION
3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
4.REQUIREMENT ANALYSIS
4.1 Functional requirement
4.2 Non-Functional requirements
5.PROJECT DESIGN
5.1 Data Flow Diagrams &User Stories
5.2 Solution Architecture
6.PROJECT PLANNING & SCHEDULING
6.1 Technical Architecture
6.2 Sprint Planning & Estimation
6.3 Sprint Delivery Schedule
7.CODING & SOLUTIONING
7.1 Feature 1
7.2 Feature 2
7.3 Database Schema (if Applicable)
8. PERFORMANCE TESTING
8.1 Performace Metrics
9. RESULTS
9.1 Output Screenshots
10.ADVANTAGES & DISADVANTAGES
11.CONCLUSION
12.FUTURE SCOPE
13.APPENDIX
13.1 Source Code
13.2 GitHub & Project Demo Link

1.INTRODUCTION

A biometric system is a technology that uses the physical or behavioural characteristics of a person to verify their identity. Biometric systems can be used for various purposes, such as access control, authentication, and identification. One of the applications of biometric systems is voting platforms, where biometric systems can help ensure the security, accuracy, and integrity of the electoral process.

Biometric systems for voting platforms can use different types of biometric modalities, such as fingerprint, face, iris, or voice recognition. These biometric modalities can be used to register, identify, and authenticate voters during the voting process. Biometric systems can also be integrated with other technologies, such as blockchain, encryption, or watermarking, to enhance the privacy, transparency, and reliability of the voting data.

1.1 Project overview :

Biometric systems can be used for various purposes, such as access control, authentication, and identification. One of the applications of biometric systems is voting platforms, where biometric systems can help ensure the security, accuracy, and integrity of the electoral process.

Biometric systems for voting platforms can use different types of biometric modalities, such as fingerprint, face, iris, or voice recognition. These biometric modalities can be used to register, identify, and authenticate voters during the voting process. Biometric systems can also be integrated with other technologies, such as blockchain, encryption, or watermarking, to enhance the privacy, transparency, and reliability of the voting data.

1.2 Purpose:

The purpose of biometric systems for voting platforms is to ensure the security, accuracy, and integrity of the electoral process. Biometric systems can use different types of biometric modalities, such as fingerprint, face, iris, or voice recognition, to register, identify, and authenticate voters during the voting process. Biometric systems can also be integrated with other technologies, such as blockchain, encryption, or watermarking, to enhance the privacy, transparency, and reliability of the voting data.

2.LITERATURE SURVEY

A literature survey for a biometric system for a voting platform in blockchain technology would involve reviewing relevant academic papers, articles, and research works that address the integration of biometrics and blockchain for secure and reliable voting systems. Here's a list of key topics and areas you might want to explore in your literature review:

1)Blockchain Technology in Voting:
Explore the foundational principles of blockchain technology in the context of voting systems.
Understand the benefits of using blockchain for transparent and secure voting.

2)Biometrics in Voting:
Investigate various biometric modalities such as fingerprint, iris, facial recognition, and voice recognition.
Examine the advantages and challenges of using biometrics for voter authentication.

3)Security and Trust in Voting:
Review how blockchain technology enhances security, transparency, and trust in voting platforms.
Analyze the potential threats and vulnerabilities in voting systems and how blockchain can mitigate them.

4)Privacy Concerns:
Explore the privacy implications of biometric data collection and storage in a voting system.
Review methods for protecting the privacy of voters while using biometrics.

5)Previous Research and Case Studies:
Identify previous research studies and real-world implementations of blockchain-based voting systems with biometrics.
Analyze the outcomes, challenges, and lessons learned from these projects.

6)Consensus Mechanisms and Smart Contracts:
Investigate the role of blockchain consensus mechanisms (e.g., Proof of Work, Proof of Stake) in ensuring the integrity of the voting process.
Understand how smart contracts can be used to automate and secure various aspects of the voting system.

7)Regulatory and Legal Aspects:
Review the regulatory and legal challenges associated with using biometrics and blockchain for voting.
Explore the compliance requirements and standards that need to be met.

8)User Experience and Accessibility:
Analyze the usability and accessibility of biometric-based voting systems for different user groups.
Discuss any issues related to inclusivity and accessibility.

9)Blockchain Scalability:
Investigate how blockchain scalability issues can impact the efficiency of a voting system.Explore solutions for improving scalability while maintaining security.

10)Future Directions and Challenges:
Identify emerging trends and future research directions in the field of biometric-based voting on blockchain.
Discuss the challenges and open issues that researchers are working to address.
11)Comparative Analysis:
Conduct a comparative analysis of different approaches and technologies used in similar systems.
Highlight the strengths and weaknesses of each approach.

2.1 Existing problem

Implementing a biometric system for a voting platform on the blockchain poses several challenges and concerns:

1. Security and Privacy: Biometric data is sensitive and permanent. Storing it on a blockchain can make it iмMutable but also increases the risk of data breaches, which could lead to identity theft and other privacy issues.

2. Accuracy and Reliability: Biometric systems мay not always be 100% accurate, which could lead to false positives or false negatives in the voting process. Ensuring the reliability of biometric data is crucial.

3. Voter Authentication: Ensuring that the person casting the vote is the rightful owner of the biometric data is a challenge. Identity theft and impersonation are potential risks.

4. Voter Anonymity: Blockchain's transparency Might conflict with the anonymity traditionally associated with voting. Balancing transparency and privacy is a Complex task.

5. Data Storage and Scalability: Storing biometric data on a blockchain can be resource-intensive. The scalability of blockchain networks can become a problem as the number of voters and their biometric data grows.

6. Regulatory and Legal Issues: The use of biometric data in voting may raise legal and regulatory concerns, as different countries have varying laws regarding the collection, storage, and use of such data.

7. Vulnerabilities: Blockchain technology itself may have vulnerabilities, and if exploited, could compromise the integrity of the voting system.

8. Cost and Accessibility: Developing and Maintaining a biometric system on the blockchain can be costly, potentially Limiting accessibility and inclusivity.

9. Double Voting Prevention: Ensuring that a voter cannot vote multiple times is challenging, as blockchain's immutability can make it difficult to correct errors.

10. Fallback Mechanisms: There should be backup mechanisms in case of biometric failures or disputes.

2.2 References

1.De Giusti A., Feierherd G., Pesado P., Depetris B. “Una aproximación a lo s requerimientos del software de voto electrónico de Argentina”. Congreso Argentino de Ciencias de la Computación. 2004.

2. Tula M. “Voto Electrónico”. Ariel Ciencias Políticas. 2005.

3. Cantijoch Cunill M. “El voto electrónico ¿Un temor justificado?”. Revista TEXTOS de la CiberSociedad, 7. http://www.cibersociedad.net. 2005.

4. Arsaute G. A., Tutores: Nasisi Óscar Herminio M. M. “Reconocimiento de características en huellas dactilares para la identificación humana”. Universidad Nacional de San Juan. Facultad de In geniería. Instituto de Automática. 1997.

5. Beavan Colin. “Huellas dactilares. Los orígenes de la dactiloscopía”. Ed. Alba. 1990.

6. Arrieta A., Marín J., Sánchez L. G., Romero L., Sánchez L. A., Batista V. “Gestión y Reconocimiento Óptico de los Puntos Caracter ísticos de Imágenes de Huellas Dactilares”. Universidad de Salamanca.

7. Reid P. “Biometrics for Network Security”. Prentice Hall. 2004.

8. Chirillo J. y otros. “Implementing Biometric Security”. Wiley Publishing. 2003.

2.3 Problem Statement Definition

Problem Statement:
In traditional voting systems, there exist significant challenges related to the security, transparency, and accessibility of the electoral process. Instances of identity fraud, tampering with voting records, and limited voter turnout continue to undermine the integrity of elections. Moreover, the need for secure, verifiable, and user-friendly voting solutions is paramount, particularly in an increasingly digital and interconnected world. To address these issues and enhance the electoral process, there is a compelling need for the development and implementation of a Biometric Security System for a Voting Platform using blockchain technology.

Key Problem Areas: Identity Verification and Security: Traditional voting systems often struggle to accurately verify the identity of voters, leading to concerns of fraud and impersonation. There is a need to develop a system that can reliably authenticate voters using biometric data while ensuring the security and privacy of their personal information.

Transparency and Trust: Maintaining trust in electoral processes is vital for a healthy democracy. Traditional systems lack the transparency and auditability required to verify the integrity of the voting process. A solution is needed to provide a verifiable and tamper-resistant record of votes and voter identity.

Accessibility: Many eligible voters face barriers to participation, such as physical disabilities, geographical distance, or lack of identification documents. An accessible voting platform is needed to ensure that all eligible citizens can exercise their right to vote.

Data Privacy and Compliance: With the collection of sensitive biometric and personal data, it is essential to establish robust data protection measures and compliance with privacy regulations to protect voter information.

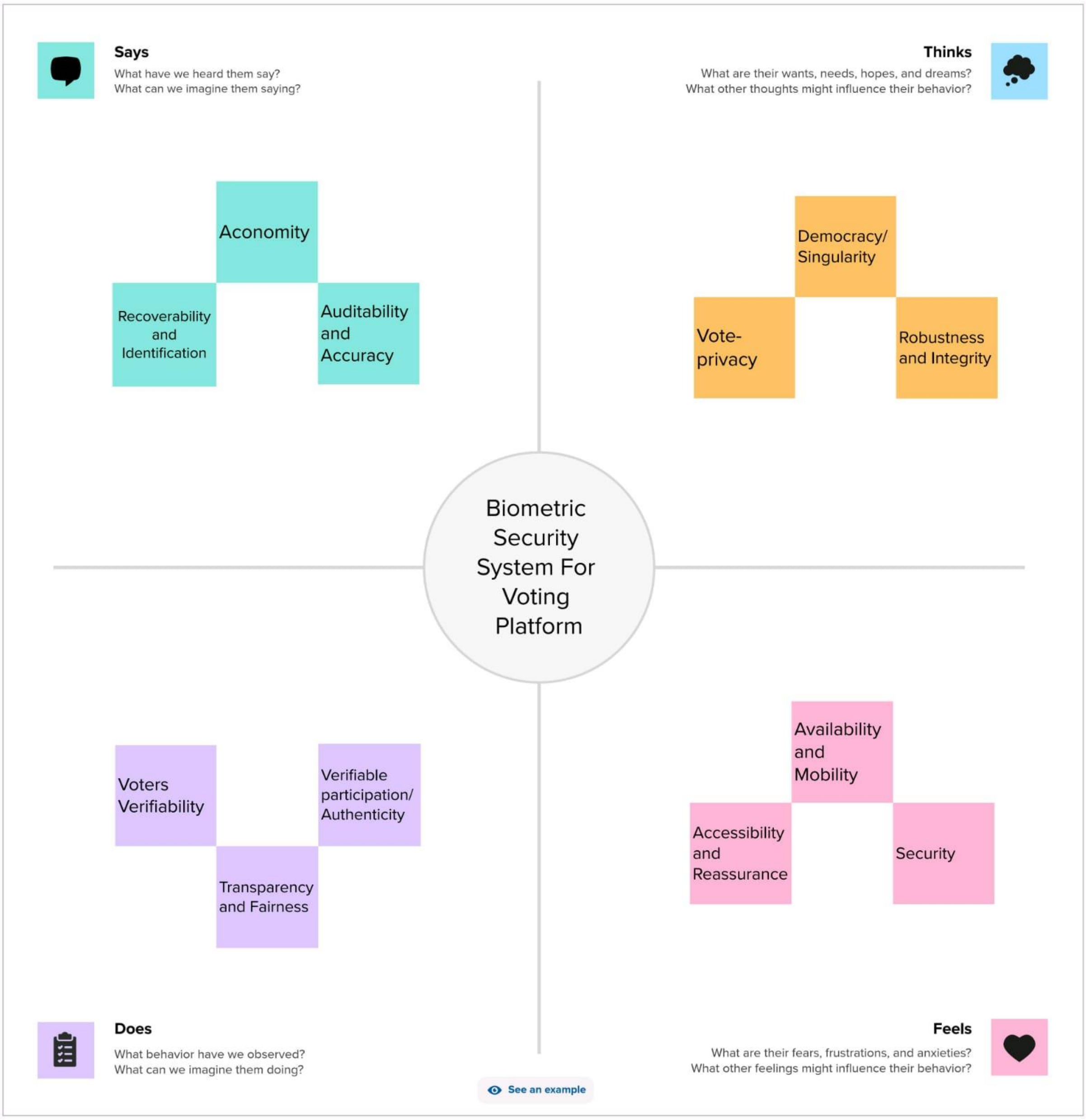
Scalability and Performance: The system must be capable of handling a large number of concurrent users during peak voting periods, ensuring smooth performance without interruptions.

Cross-Border Voting: The voting platform should support secure voting for citizens residing in different locations or countries, complying with eligibility criteria based on residency or citizenship.

Solution Objectives: The proposed Biometric Security System for a Voting Platform using blockchain aims to:
(i)Implement biometric authentication methods to enhance the accuracy and security of voter identity verification.
(ii)Utilize blockchain technology to provide a transparent and immutable ledger of voting transactions.
(iii)Improve accessibility for all eligible voters, including those with disabilities. Protect the privacy of voter data and ensure compliance with data protection regulations.
(iv)Enable the scalability and performance required for large-scale elections. Support cross-border voting to increase democratic participation.Uphold the principles of transparency, security, and trust in the electoral process.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Says

What have we heard them say?
What can we imagine them saying?

Acconomity

Recoverability and Identification

Auditability and Accuracy

Thinks

What are their wants, needs, hopes, and dreams?
What other thoughts might influence their behavior?

Democracy/ Singularity

Vote-privacy

Robustness and Integrity

Biometric Security System For Voting Platform

Voters Verifiability

Verifiable participation/ Authenticity

Transparency and Fairness

Availability and Mobility

Accessibility and Reassurance

Security

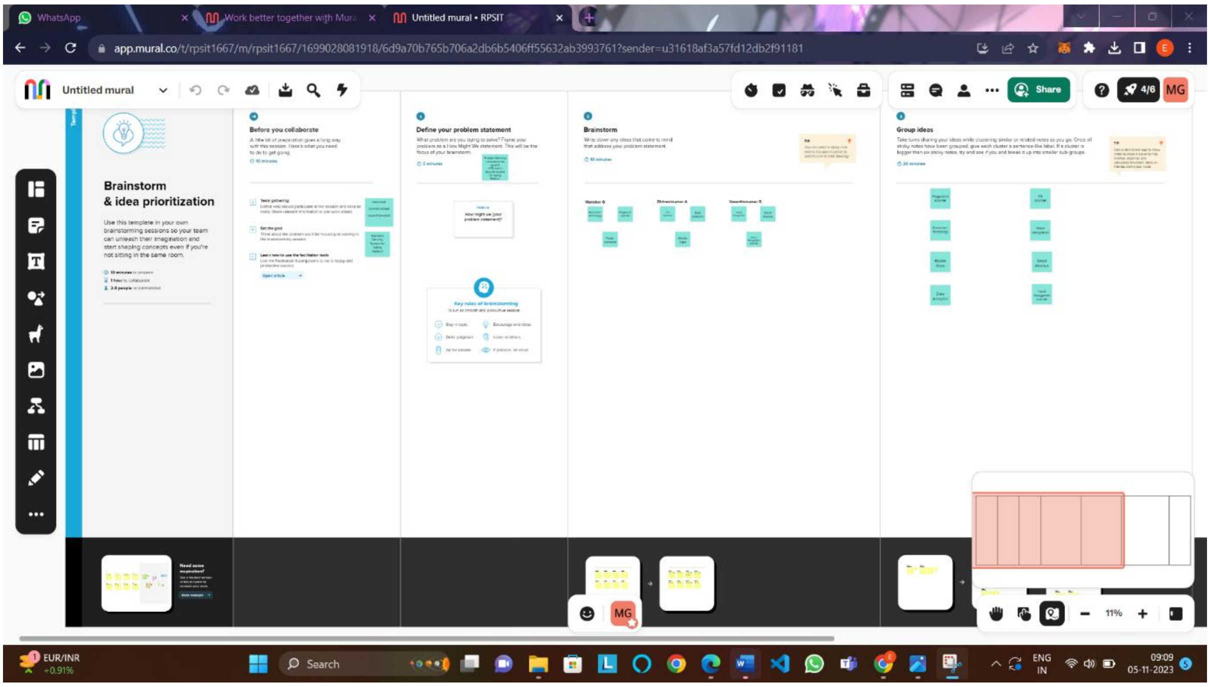
Does

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?

3.2 Ideation & Brainstorming



Brainstorm & idea prioritization

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.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

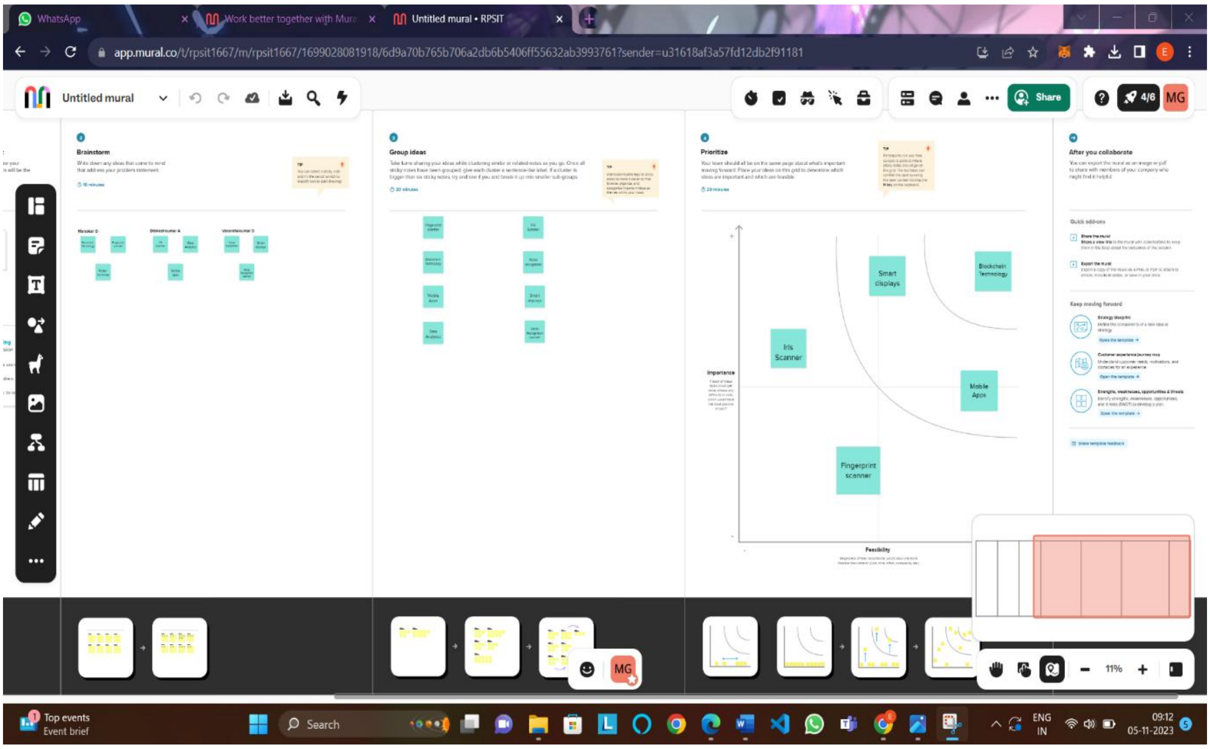
Brainstorm

Write down any ideas that come to mind that address your problem statement.

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.

WhatsApp
Work better together with Mura
Untitled mural • RPSIT
app.mural.co/t/rpsit1667/m/rpsit1667/1699028081918/6d9a70b765b706a2db6b5406ff55632ab3993761?sender=u31618af3a57fd12db2f91181
Untitled mural
Share
4/6
MG
Key rules of brainstorming
11%
EUR/INR
+0.91%
Search
ENG
IN
09:09
05-11-2023



4.REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement	Description
FR-1	Voter Registration	Capture and store biometric data (e.g., fingerprints, facial scans, iris scans) of eligible voters securely. Validate and verify the identity of voters during the registration process. Maintain an up-to-date and accurate voter database.
FR-2	Biometric Authentication	Authenticate voters using their biometric data, ensuring high accuracy. Support multiple biometric authentication methods to accommodate different voters. Implement anti-spoofing measures to prevent fraudulent attempts.
FR-3	Secure Voting	Provide a secure and tamper-resistant environment for casting votes.

		Encrypt the voting data to protect voter privacy. Ensure one-person-one-vote by preventing multiple votes by the same voter.
FR-4	Blockchain Integration	Utilize a blockchain for recording and storing voting transactions securely and immutably. Implement smart contracts for vote counting and result verification. Enable transparent and auditable tracking of all voting activities on the blockchain.
FR-5	Voter Interface	Offer an intuitive and user-friendly interface for voters to cast their ballots. Provide clear instructions and guidance throughout the voting process. Ensure accessibility for voters with disabilities.
FR-6	Auditability and Transparency	Enable real-time monitoring of the voting process for election authorities. Support auditing and verification of votes and voter identity after the election. Ensure the system is resistant to fraud and manipulation.
FR-7	Election Management	Allow election administrators to set up and manage elections, including candidate registration and ballot creation. Provide tools for configuring election rules, such as eligibility criteria and voting deadlines.
FR-8	Results Reporting	Generate accurate and verifiable election results after voting is complete. Display results in a transparent and accessible manner for the public. Prevent unauthorized access to or manipulation of results.
FR-9	Security Measures	Implement robust cybersecurity measures to protect against attacks, data breaches, and unauthorized access. Ensure the privacy and confidentiality of voter biometric data. Implement disaster recovery and backup mechanisms to guarantee system availability.

FR-10	Cross-Border Voting	Support secure voting for citizens residing in different locations or countries. Verify voters' eligibility to participate in specific elections based on their residency or citizenship.
FR-11	User Management	Enable election officials to manage user accounts, roles, and permissions. Implement strict access controls to prevent unauthorized actions within the system.
FR-12	Scalability	Ensure the system can accommodate a high volume of concurrent users during peak voting periods. Be capable of handling large-scale elections with millions of voters.
FR-13	Compliance with Regulations	Comply with legal and regulatory requirements related to elections, privacy, and data protection. Support international standards for secure electronic voting systems.
FR-14	Usability and Training	Provide training and support materials for voters and election administrators. Ensure the system is user-friendly and easy to navigate.
FR-15	Redundancy and High Availability	Implement redundancy and failover mechanisms to prevent system downtime. Guarantee the system's availability during the entire election period.
FR-16	Integration	Integrate with external systems and databases, such as voter registries and identity verification services. Support interoperability with different blockchain networks and technologies.
FR-17	Accessibility	Ensure that the system is accessible to voters with disabilities, adhering to accessibility standards.
FR-18	Testing and Quality Assurance	Conduct extensive testing, including security, penetration, and usability testing. Ensure the system is free from vulnerabilities and bugs.

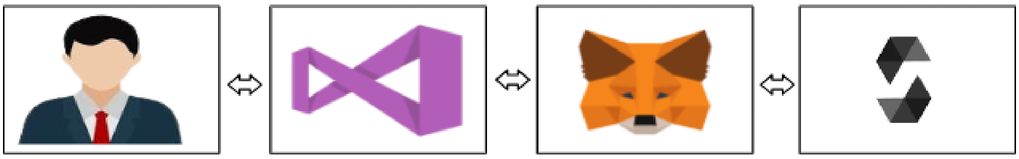
4.2 Non-Functional requirements

NFR No.	Functional Requirement	Description
NFR-1	Data Security	Ensure that all biometric data, voting records, and personal information are stored and transmitted securely using encryption and other protective measures.
NFR-2	Access Control	Implement robust access controls to prevent unauthorized access to the system, sensitive data, and administrative functions.
NFR-3	Authentication Security	Protect the biometric authentication process from spoofing, tampering, or other fraudulent activities.
NFR-4	Blockchain Security	Implement security measures to safeguard the blockchain network against attacks, including 51% attacks and double spending.
NFR-5	Scalability	Ensure that the system can scale horizontally to handle an increasing number of users and transactions, especially during peak voting periods. Support load balancing and resource provisioning to maintain performance as the system scales.
NFR-6	Performance	Define acceptable response times for user interactions, such as logging in, casting a vote, and accessing election results. Monitor system performance and optimize resource allocation to meet performance goals.
NFR-7	Reliability and Availability	Ensure high system availability during the entire election period, minimizing downtime or disruptions. Implement disaster recovery and backup mechanisms to guarantee the system's resilience.
NFR-8	Auditability and Traceability	Maintain a detailed audit trail of all voting activities and system operations. Ensure that all actions within the system can be traced and verified.

NFR-9	Compliance	Adhere to legal and regulatory requirements for electronic voting systems, privacy, and data protection. Support international standards for secure electronic voting systems.
NFR-10	Interoperability	Ensure compatibility with various biometric authentication devices, browsers, and blockchain networks. Support data exchange with external systems and government databases.
NFR-11	Usability	Provide a user-friendly interface for both voters and election administrators. Ensure that the system is easy to navigate and understand, even for non-technical users.
NFR-12	Accessibility	Adhere to accessibility standards to ensure that the system is usable by voters with disabilities. Provide alternative access methods for individuals with special needs.
NFR-13	Privacy	Protect the privacy of voters by anonymizing and securing their voting records and biometric data. Implement data retention and deletion policies in compliance with privacy laws.
NFR-14	Load Handling	Define the system's ability to handle a large number of simultaneous users and transactions without degrading performance. Test and optimize the system's capacity to ensure it can accommodate peak loads.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



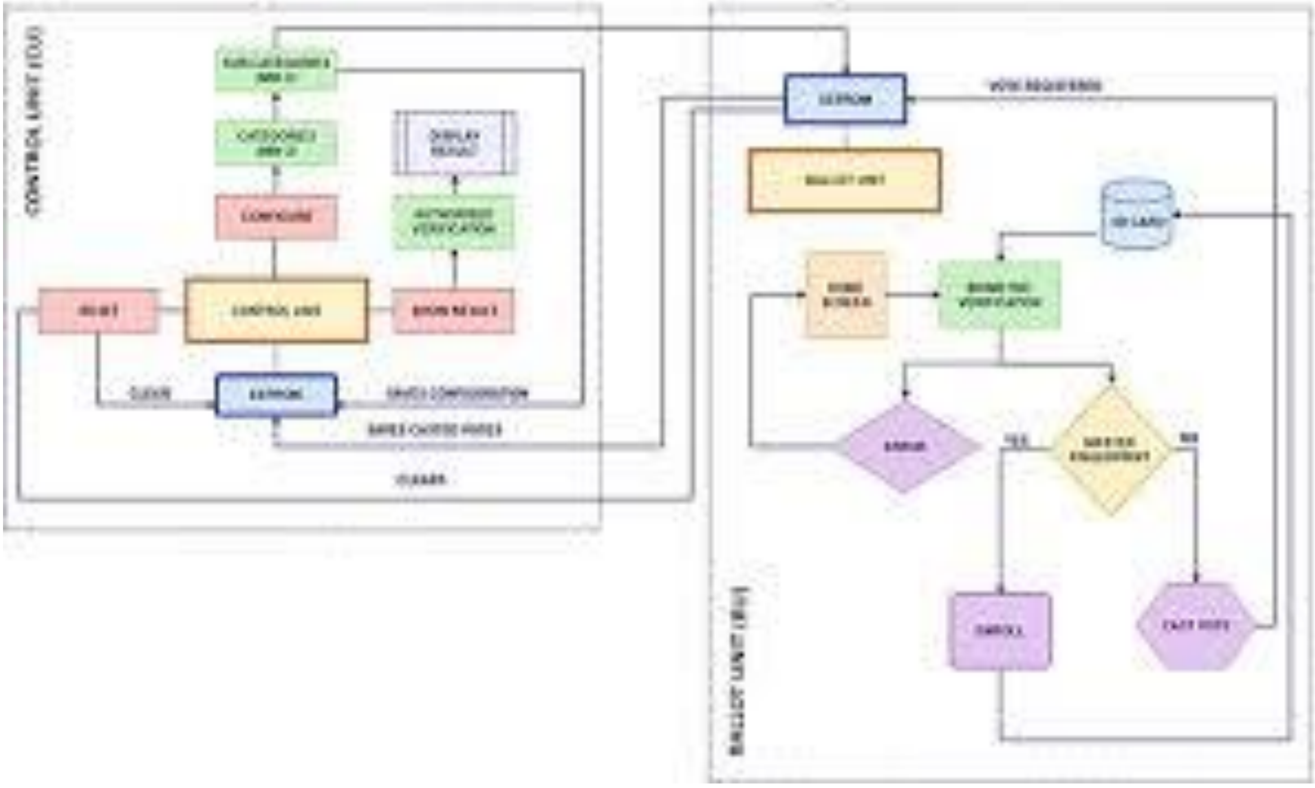
End User

Frontend (VS code)

Metamask

Solidity code (blockchain code)

Data Flow Diagrams



User Stories

Voter Registration:

As a new voter, I want to be able to register for the voting platform using my biometric data (e.g., fingerprint or facial scan) to ensure the security of my identity. As an election administrator, I want to verify and approve the voter registration requests securely and efficiently.

Authentication and Voting:

As a voter, I want to be able to authenticate myself using my biometric data to access the voting platform. As a voter, I want to cast my vote securely using my biometric authentication, ensuring that my vote is counted accurately. As a visually impaired voter, I want to use voice recognition for biometric authentication to ensure accessibility.

Security and Privacy:

As a voter, I want to be confident that my biometric data is securely stored and cannot be accessed by unauthorized individuals. As a voter, I want to be informed about the security measures in place to protect my biometric data and personal information.

Blockchain Integration:

As a voter, I want to be able to verify my vote on the blockchain to ensure that it was recorded accurately and cannot be altered. As an election observer, I want access to the blockchain to independently verify the voting results for transparency.

Accessibility:

As a voter with a physical disability, I want the option to use multiple biometric methods for authentication to ensure that I can vote easily. As an election administrator, I want to provide support for voters with disabilities to ensure they can participate independently.

Cross-Border Voting:

As an expatriate voter, I want to be able to securely vote from abroad using my biometric data to ensure that my voice is heard in my home country's elections.

Results Verification:

As a concerned citizen, I want to be able to verify that the election results on the blockchain match the actual votes cast to ensure the accuracy of the election.

Auditability and Transparency: As an election auditor, I want to access the blockchain to audit the voting process and confirm its integrity. As an election official, I want to maintain a clear audit trail of all activities and transactions related to the voting platform.

Compliance and Legal Requirements:

As a voter, I want assurance that the voting platform complies with local and international regulations to ensure the legality of the voting process.

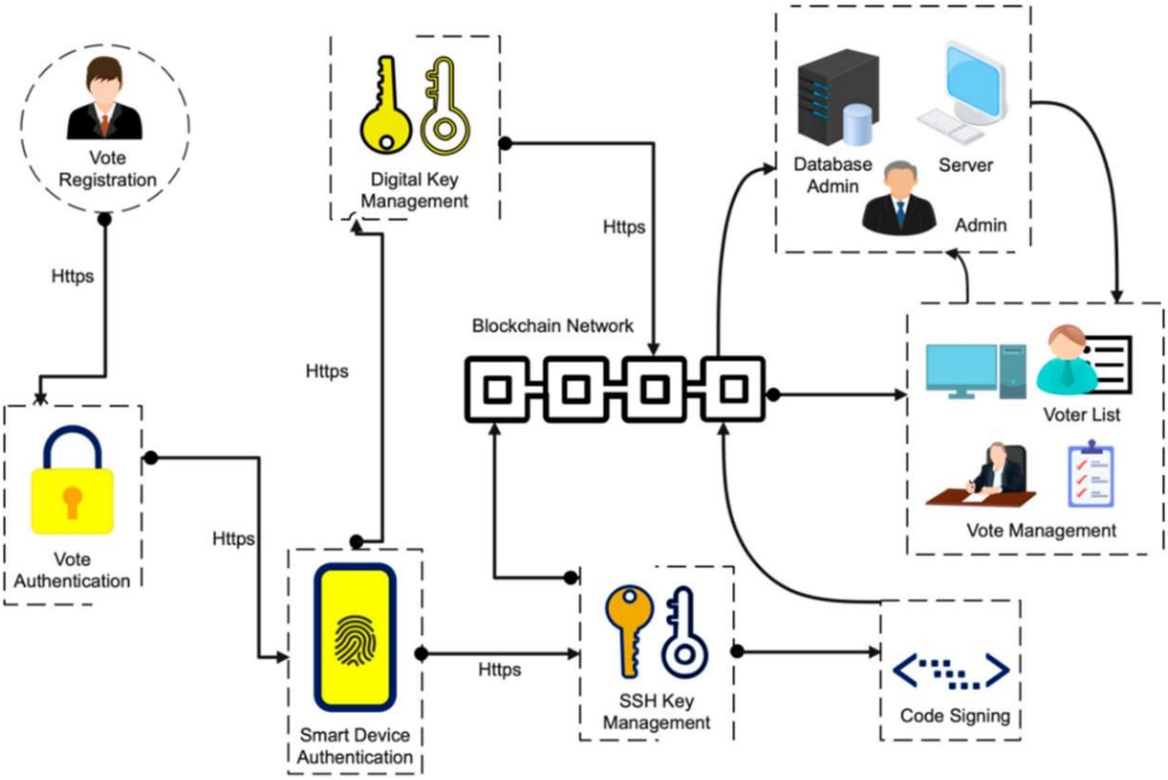
Training and Support:

As a voter, I want access to clear and concise instructions on how to use the biometric voting system. As an election administrator, I want to provide training and support materials for voters to ensure a smooth voting experience.

Load Handling and Scalability:

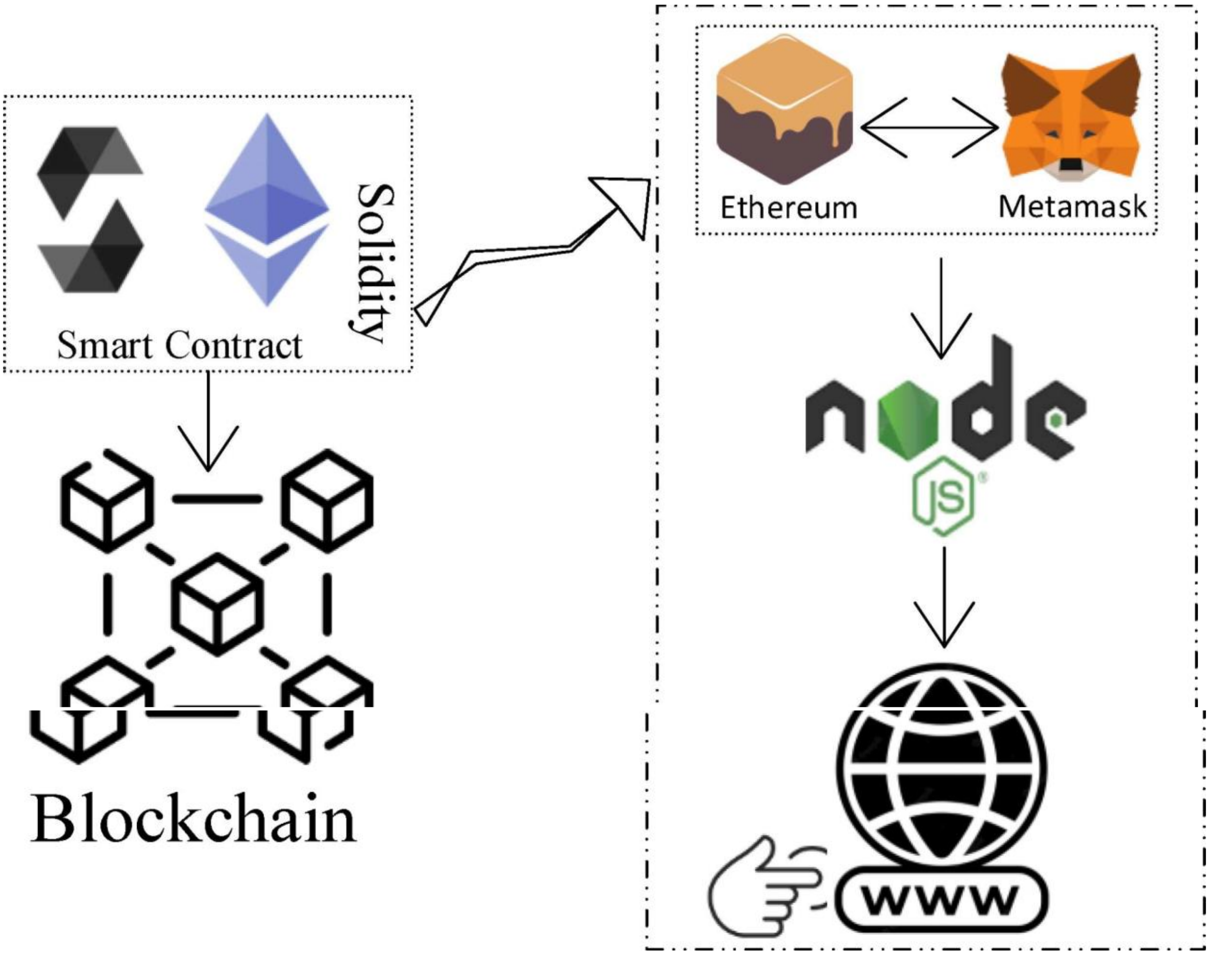
As an election administrator, I want the system to handle a high volume of voters during peak voting periods without performance degradation.

5.2 Solution Architecture



6.PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Groom the Backlog:

Before the sprint planning meeting, ensure that the product backlog is well-groomed, with user stories and tasks well-defined and prioritized.

Sprint Goal:

Establish a clear sprint goal. In this context, it might be related to implementing specific biometric authentication features, enhancing blockchain security, or improving user experience.

Select User Stories:

Based on the sprint goal, the team selects user stories and tasks from the product backlog that can be realistically completed during the sprint.

Break Down User Stories:

Break down selected user stories into smaller, actionable tasks. For example, if the goal is to enhance biometric security, tasks might include integrating a new biometric sensor or improving the encryption of biometric data.

Estimate Tasks:

Use estimation techniques (e.g., story points, hours, or ideal days) to estimate the effort required for each task. The team's estimation should consider factors like complexity, dependencies, and technical challenges.

Capacity Planning:

Assess the team's capacity for the upcoming sprint based on past performance and team availability.

Task Assignment:

Assign tasks to team members, considering their skills and expertise.

Definition of Done:

Define clear acceptance criteria for each user story to ensure it's considered "done" at the end of the sprint.

Sprint Review and Retrospective:

Discuss the outcomes and lessons learned from the previous sprint in the planning meeting and incorporate improvements in the new sprint.

Sprint Estimation:

There are several methods for estimating tasks within a sprint:

Story Points:

Use a relative estimation technique to assign story points to tasks. Each story point represents the effort required, and tasks are estimated in relation to one another. This helps prioritize tasks based on their relative size and complexity.

Hours or Ideal Days:

Some teams prefer to estimate tasks in hours or ideal days. This provides a more concrete estimate of time, which can be useful for planning.

Planning Poker:

Teams can use the Planning Poker technique to collectively estimate tasks. Team members assign points to tasks, discuss any discrepancies, and re-estimate until a consensus is reached.

T-shirt Sizing:

Use T-shirt sizes (S, M, L, XL) to estimate tasks based on their relative complexity. This is a quick and straightforward way to estimate tasks.

Expert Judgment:

In some cases, experts on the team may provide estimates based on their experience and knowledge.

After sprint planning and estimation, the team can start the sprint, work on the tasks, and track progress during daily stand-up meetings. Continuous communication and adjustments are key to successful sprint execution in an Agile development environment.

6.3 Sprint Delivery Schedule

Sprint 1: Project Initiation (2 weeks)

Define the project scope, objectives, and requirements.
Create a project team and assign roles and responsibilities.
Conduct initial research on biometric security, blockchain technology, and voting platform requirements.
Develop a high-level project plan and Sprint schedule.
Set up communication channels and tools for the project team.

Sprint 2: Requirements and Design (2-4 weeks)

Gather detailed requirements for the voting platform and biometric security system.Create user stories and prioritize them based on importance.
Develop an architectural design for the biometric security system and its integration with the blockchain-based voting platform.
Define the technology stack and tools you'll use.
Create wireframes and prototypes of the user interface.
Begin building the development environment.

Sprint 3-6: Development (4-8 weeks)

Implement the biometric security system components, including fingerprint, iris, or other biometric authentication methods. Develop the blockchain-based voting platform, including smart contract logic. Integrate the biometric system with the voting platform. Conduct ongoing testing and quality assurance. Address any technical issues and bugs.
Begin documenting the code and system functionality.

Sprint 7: Testing and QA (2-4 weeks)

Conduct comprehensive testing of the entire system.
Perform security testing and vulnerability assessments. Engage in user acceptance testing (UAT) to ensure the platform meets stakeholders' requirements. Perform load testing to ensure scalability. Refine and fix any issues discovered during testing.

Sprint 8: Deployment (1-2 weeks)

Prepare for the production deployment. Deploy the biometric security system and voting platform to a production environment. Monitor the system closely after deployment to ensure stability.

Sprint 9: Training and Documentation (1-2 weeks)

Train election officials and system administrators on how to use the system. Create user manuals and documentation for voters and system users.

Sprint 10: Final Testing and Bug Fixes (1-2 weeks)

Perform a final round of testing and address any last-minute issues.
Conduct a security audit and review the entire system.

Sprint 11: Go-Live and Launch (1 week)

Coordinate with election authorities to plan and execute the system's use in a real election or voting event.

Sprint 12: Post-Launch Evaluation and Maintenance (Ongoing)

Continuously monitor and maintain the system to ensure its reliability and security. Gather feedback from users and make necessary improvements.

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

7.1 Feature 1

Smart Contract (Solidity):

```
// 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);
  }
}
```

7.2 Feature 2

Trasfor Ownership

Transferring ownership of a biometric system for a voting platform is a complex process that involves legal, technical, and administrative considerations. Here are the general steps you should take when transferring ownership:

Legal and Regulatory Compliance:

Review the legal and regulatory framework governing the ownership and operation of biometric voting systems in your jurisdiction. Ensure that any applicable laws and regulations allow for the transfer of ownership.

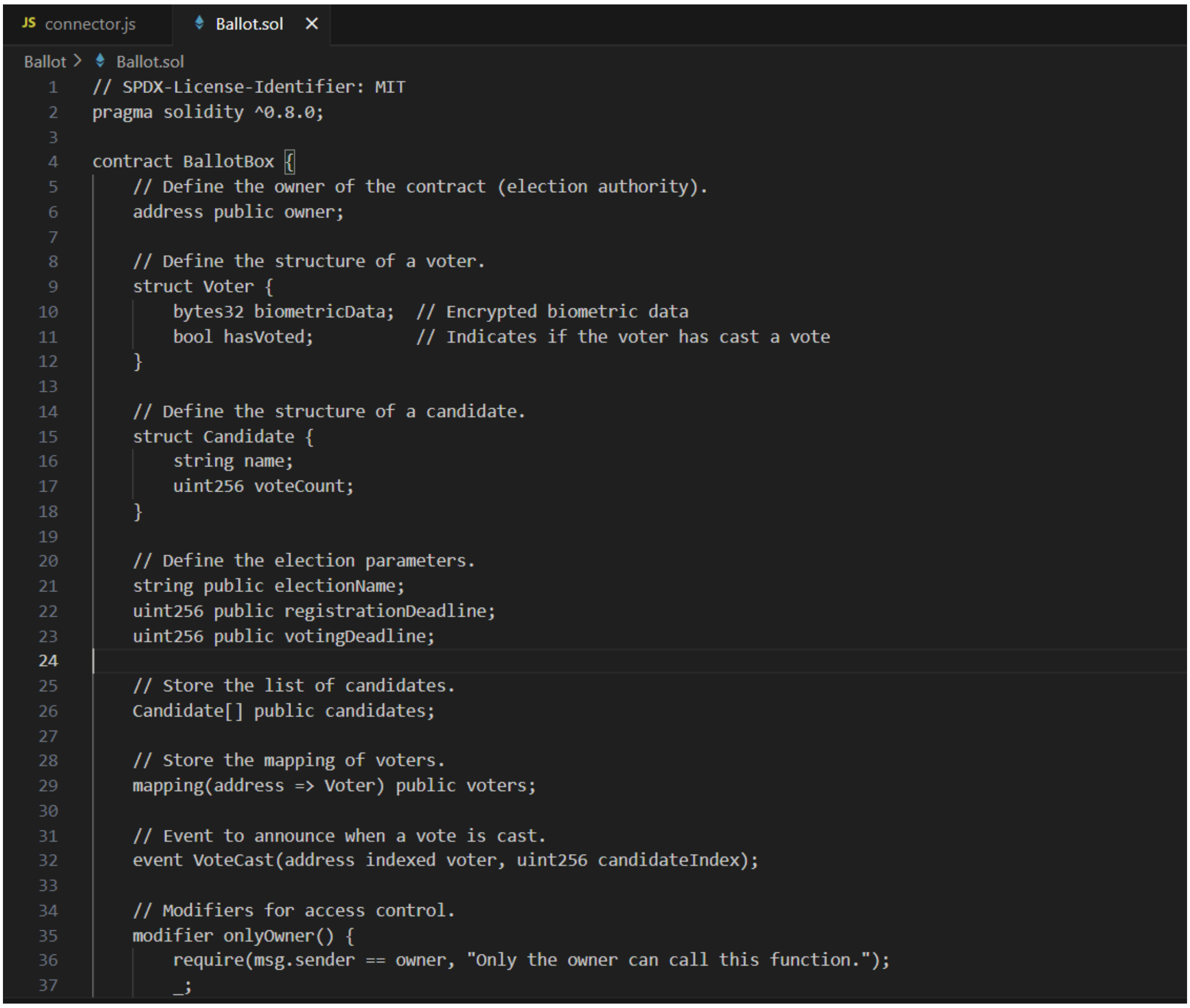
Contractual Agreements:

Review any existing contracts, licenses, or agreements related to the biometric system. Determine if these agreements allow for a transfer of ownership.

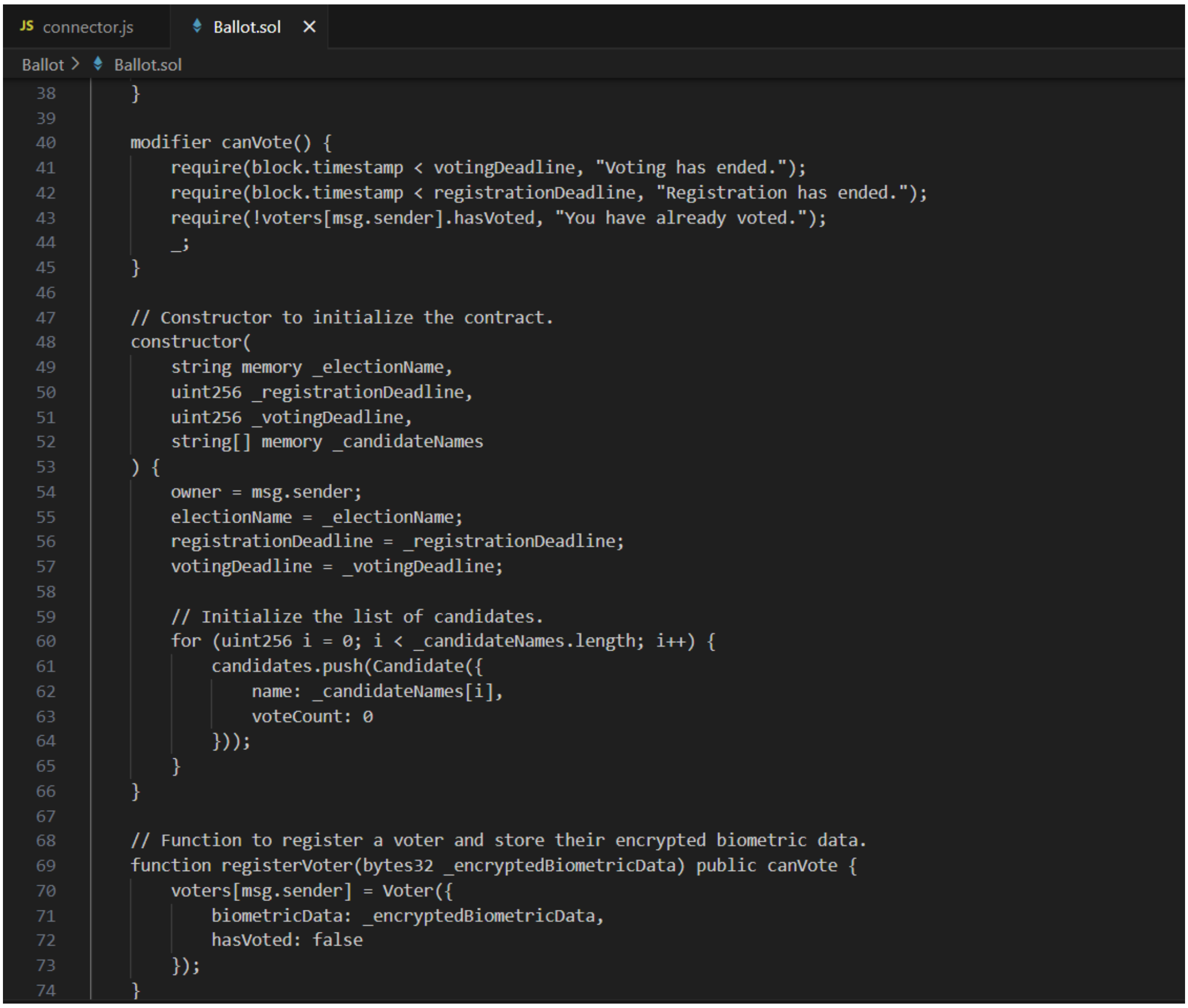
Due Diligence:

Conduct a thorough audit and evaluation of the existing biometric system to ensure it is in good working condition. Verify the accuracy, security, and integrity of the biometric data stored in the system.

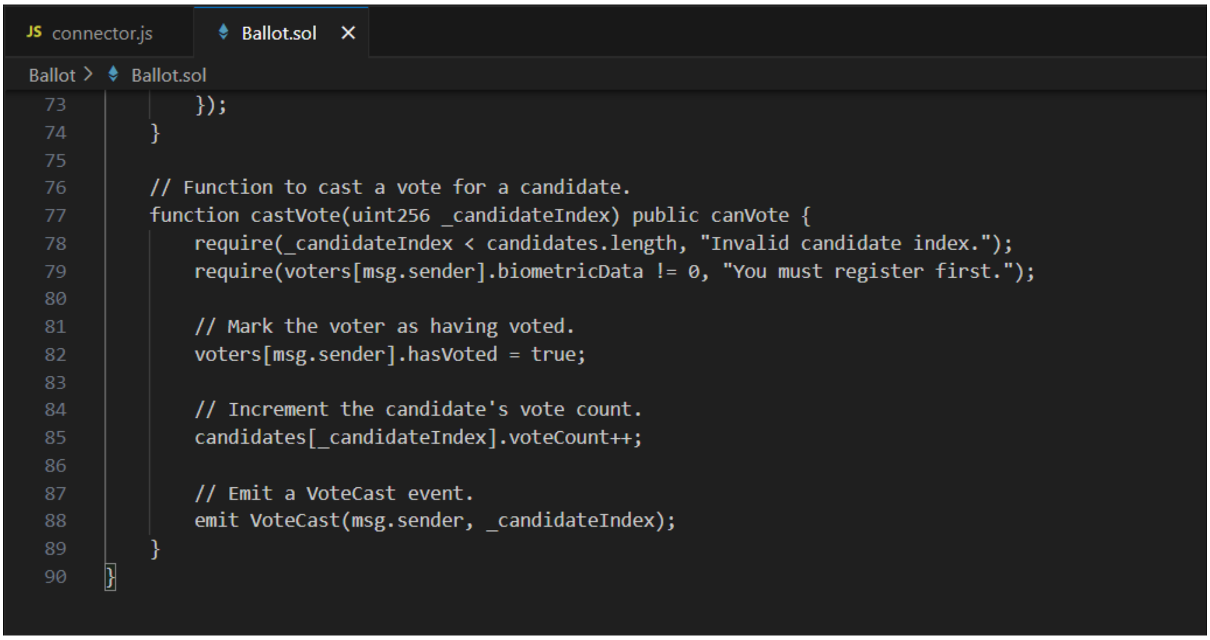
Smart Contract (Solidity):



```
// 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);
    }
}
```

7.3 Database Schema (if Applicable)

On-Chain Ethereum Smart Contracts:

Implementing on-chain Ethereum smart contracts in a biometric system for a voting platform can enhance transparency, security, and trust in the voting process. Here are the key considerations for using Ethereum smart contracts in this context:

Voter Registration and Verification:

Use Ethereum smart contracts to manage the voter registration process. Each eligible voter can have a unique Ethereum address associated with their identity.

Voter Authentication:

Integrate biometric verification within the smart contract to verify voters' identities. Biometric data can be hashed and stored on-chain or off-chain and compared during the voting process.

Voting Process:

Create a smart contract that handles the voting process, ensuring that each voter can cast only one vote. Ethereum's immutable ledger ensures the integrity of the voting data.

Privacy:

Consider privacy concerns. While Ethereum is transparent, it's possible to implement privacy solutions such as zero-knowledge proofs or sidechains for confidential voting.

Voting Tokens:

Issue voting tokens or tokens representing the right to vote to eligible voters. These tokens can be used to interact with the voting smart contract.

Endorsement and Validation:

Use a decentralized network of validators to endorse the voting results, enhancing trust in the process. Validators could be nodes on the Ethereum network or a consortium of trusted entities.

Security:

Implement robust security measures to protect the Ethereum private keys used for voter authentication. Biometric data should be securely stored and transmitted.

Auditability:

Take advantage of Ethereum's immutability to ensure that the entire voting process is auditable. All interactions with the smart contract are recorded on the blockchain.

Token Transfers:

Enable voters to transfer their voting tokens to others (if allowed by the rules) while ensuring that the transferred tokens still represent the right to vote.

Off-Chain Metadata Storage (Traditional Database or Decentralized Storage):

Traditional Database:

Relational Databases: Traditional databases like MySQL, PostgreSQL, or SQL Server can be used to store metadata related to the biometric system. They are suitable for structured data and provide robust querying capabilities.

Advantages: Data consistency, ACID (Atomicity, Consistency, Isolation, Durability) compliance, strong data relationships, and well-established security features.

Considerations: Scaling can be more complex, and centralized databases can be vulnerable to single points of failure and security breaches.

Decentralized Storage:

Blockchain: Utilizing a blockchain for metadata storage provides transparency, immutability, and distributed consensus. Systems like Ethereum, Hyperledger Fabric, or bespoke blockchain solutions can be considered.

Distributed File Systems: Solutions like IPFS (InterPlanetary File System) or Storj offer decentralized and peer-to-peer data storage and retrieval capabilities.

Advantages: Immutability, transparency, resistance to single points of failure, enhanced security, and data availability.

Considerations: Complex setup and management, potential scalability issues, and the need for participants to maintain blockchain nodes.

Hybrid Approach:

A combination of traditional and decentralized storage can be used to balance the advantages of both. For instance, storing critical data and historical records on a blockchain while using traditional databases for faster access to frequently changing data. This approach can provide data integrity, security, and performance.

Additional details

8. PERFORMANCE TESTING

8.1 Performace Metrics

Measuring the performance of a Biometric Security System for a Voting Platform using blockchain is crucial to ensure its effectiveness, security, and reliability. Here are some key performance metrics to consider:

Authentication Accuracy: This metric evaluates the accuracy of biometric authentication methods (e.g., fingerprint, facial recognition, iris scan) in correctly identifying voters. It's essential to minimize false positives and false negatives.

False Acceptance Rate (FAR): FAR measures the percentage of unauthorized users who are incorrectly granted access to the system. A lower FAR is desirable for security.

False Rejection Rate (FRR): FRR measures the percentage of authorized users who are incorrectly denied access. A lower FRR indicates a more user-friendly system.

Biometric Template Matching Time: This metric assesses the time it takes to compare a voter's biometric data against the stored templates. Faster matching times enhance the user experience.

Transaction Speed: Evaluate the speed at which transactions are processed on the blockchain. Slow transaction speeds can lead to delays in recording votes and verifying identities.

Throughput: Measure the system's ability to handle a high volume of transactions within a given time frame. High throughput is essential to accommodate a large number of voters during peak voting periods.

Scalability: Assess how the system performs as the number of users and transactions increases. Scalability is crucial to ensure the system can handle growing demand during elections.

Security: Evaluate the system's ability to protect against various forms of cyberattacks, including DDoS attacks, data breaches, and unauthorized access.

Blockchain Transaction Confirmation Time: Measure the time it takes for a transaction to be confirmed on the blockchain. Faster confirmation times ensure the transparency and immutability of the voting data.

Privacy: Assess the level of privacy protection provided to voters. Ensure that biometric data is securely stored and that voters' personal information is anonymized and protected.

Voter Turnout: Track the percentage of eligible voters who participate in the election. A well-performing system should encourage higher voter turnout due to improved accessibility and security.

Auditability: Measure the ease of auditing the blockchain to verify the accuracy and integrity of voting results. The system should make it easy to track and verify all transactions.

Accessibility: Evaluate the accessibility of the voting platform to different user groups, including individuals with disabilities. Ensure that the system complies with accessibility standards.

Usability: Gather user feedback to assess the system's user-friendliness. User satisfaction surveys can provide insights into the ease of use and overall experience.

Uptime and Reliability: Measure the system's availability and reliability during the election period. High uptime is critical to ensure that voters can access the platform without interruptions.

Compliance with Regulations: Ensure that the system complies with relevant legal and regulatory requirements, such as data protection laws and election regulations.

Cost-effectiveness: Assess the cost-effectiveness of implementing and maintaining the system, taking into account hardware, software, and operational expenses.

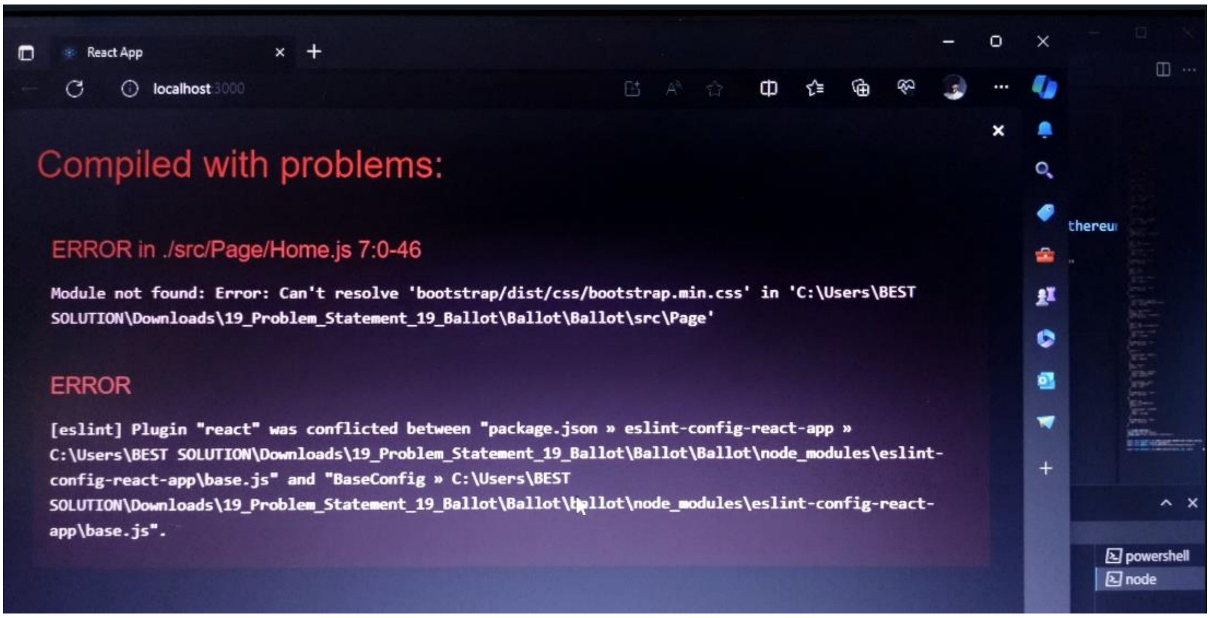
Environmental Impact: Evaluate the environmental impact of the system, including energy consumption and carbon footprint, especially if it relies on energy-intensive blockchain networks.

Response Time: Measure the response time of the system when voters interact with it. Fast response times contribute to a smooth voting experience.

Disaster Recovery and Redundancy: Assess the system's ability to recover from failures and the presence of redundancy mechanisms to ensure continuous operation.

9. RESULTS

9.1 Output Screenshots



```
Compiled with problems:

ERROR in ./src/Page/Home.js 7:0-46

Module not found: Error: Can't resolve 'bootstrap/dist/css/bootstrap.min.css' in 'C:\Users\BEST
SOLUTION\Downloads\19_Problem_Statement_19_Ballot\Ballot\Ballot\src\Page'

ERROR

[eslint] Plugin "react" was conflicted between "package.json » eslint-config-react-app »
C:\Users\BEST SOLUTION\Downloads\19_Problem_Statement_19_Ballot\Ballot\Ballot\node_modules\eslint-
config-react-app\base.js" and "BaseConfig » C:\Users\BEST
SOLUTION\Downloads\19_Problem_Statement_19_Ballot\Ballot\Ballot\node_modules\eslint-config-react-
app\base.js".
```

10. ADVANTAGES & DISADVANTAGES

Biometric systems in voting platforms have both advantages and disadvantages, which must be carefully considered when implementing such systems. Here are some of the key points to keep in mind:

Advantages of Biometric Systems for Voting:

Enhanced Security:

Biometrics, such as fingerprint or iris scans, provide a high level of security and accuracy in voter authentication, reducing the risk of identity fraud and multiple voting.

Reduced Fraud:

Biometric data is unique to each individual, making it difficult for unauthorized individuals to impersonate eligible voters.

Improved Voter Verification:

Biometrics can streamline the voter verification process, making it faster and more efficient, especially in large-scale elections.

Data Accuracy: Biometric data is highly accurate, minimizing the likelihood of errors associated with manual data entry.

Increased Voter Turnout:

Biometric systems can make voting more accessible and convenient, potentially increasing voter turnout.

Transparency and Trust:

Biometric systems can enhance the transparency and trustworthiness of the voting process, as they are difficult to tamper with or manipulate.

Accessibility:

Biometric systems can be inclusive, as they can be designed to accommodate individuals with disabilities or those who have difficulty with traditional voting methods.

Disadvantages of Biometric Systems for Voting:

Cost and Infrastructure:

Implementing a biometric voting system can be expensive, requiring significant investment in hardware, software, and infrastructure.

Privacy Concerns:

Collecting and storing biometric data raises privacy concerns. Voters may be apprehensive about the security and misuse of their biometric information.

Technical Challenges:

Biometric systems can be prone to technical issues such as sensor malfunctions, false positives, or false negatives, which could disenfranchise eligible voters.

Vulnerability to Cyberattacks:

Biometric data, if not properly secured, can be vulnerable to data breaches and cyberattacks.

Exclusivity:

Biometric systems may exclude individuals who do not have biometric data, or it may not be feasible to obtain biometric data for every voter.

Ethical Concerns: The collection and use of biometric data raise ethical questions related to consent and the potential misuse of such data.

Complexity for Voters:

Some voters, especially those who are not familiar with technology or biometrics, may find the system too complex to use.

11. CONCLUSION

In conclusion, the implementation of a Biometric Security System for a Voting Platform using blockchain technology holds the potential to revolutionize the way we conduct elections. This innovative approach combines the strengths of biometric authentication and blockchain's security and transparency to create a robust and trustworthy voting system. Here are some key points to summarize the concept:

Enhanced Security: The integration of biometric authentication methods, such as fingerprint, facial recognition, or iris scans, provides a higher level of security and ensures the identity of the voter, reducing the risk of impersonation or fraud.

Blockchain's Transparency: The use of blockchain technology enables transparent and tamper-resistant recording of voting transactions. This transparency fosters trust among voters and stakeholders by allowing them to verify the integrity of the election process.

Accessibility: Biometric systems can make voting more accessible to a wider range of individuals, including those with disabilities, by providing user-friendly interfaces and accommodating various authentication methods.

Audibility and Verification: Blockchain-based systems enable real-time auditing of the voting process and post-election verification of results. This feature ensures the accuracy and fairness of elections.

Cross-Border Voting: Such systems can enable secure voting for citizens living abroad, increasing participation in elections and improving democracy.

Data Privacy: Robust measures must be in place to protect the privacy of voters. Biometric data and personal information must be securely stored and managed.

Compliance: The system must comply with local and international regulations related to elections, data protection, and privacy to ensure its legality and trustworthiness.

Non-Functional Requirements: Considerations such as security, scalability, performance, reliability, and usability are crucial to the success of the system.

12. FUTURE SCOPE

The future scope for a Biometric Security System for a Voting Platform using blockchain is promising, with the potential to enhance the security, transparency, and accessibility of voting processes. Here are some areas of future development and opportunities:

Widespread Adoption: As more countries and regions recognize the benefits of secure and transparent electronic voting, the adoption of biometric security systems with blockchain integration is likely to grow. This could lead to more extensive use in national and local elections.

Improved Accessibility: Future developments in biometric technology may make it more accessible and inclusive for a broader range of voters, including those with disabilities. Innovations such as facial recognition, voice recognition, and other biometric methods may be explored to make the voting process more inclusive.

Enhanced Security: Security will continue to be a top priority. Blockchain technology, with its inherent immutability and cryptographic security, will be used to secure the integrity of the voting process. Ongoing research and development will focus on making the system resistant to attacks and ensuring the privacy of voter data.

User-Friendly Interfaces: User interfaces for voting platforms will continue to evolve to be more intuitive and user-friendly. This is crucial to encourage voter participation, especially among older or less tech-savvy populations.

Mobile Voting: Future voting platforms may increasingly support mobile voting applications. Voters could use their smartphones for secure biometric authentication and voting, making the process more convenient and accessible.

Cross-Border Voting: Biometric security systems and blockchain technology could enable secure cross-border voting for expatriates and citizens living abroad, allowing them to participate in their home country's elections.

Secure Identity Verification: Beyond elections, biometric systems and blockchain can be used for secure identity verification in various applications, including government services, financial transactions, and healthcare.

Research and Development: Continuous research and development will be essential to stay ahead of emerging threats and vulnerabilities. Cybersecurity experts and researchers will play a crucial role in ensuring the ongoing security of these systems.

Regulatory Frameworks: Governments and international organizations will need to develop clear and robust regulatory frameworks to govern the use of biometric security systems in voting. These frameworks should address issues like data privacy, consent, and auditability.

Blockchain Scalability: Scalability remains a challenge for blockchain technology. Future solutions, such as sharding and layer 2 solutions, will need to be explored to handle the large volumes of transactions that occur during an election.

Post-Election Audits: The use of blockchain in voting can facilitate post-election audits and transparency. Future developments may focus on making the auditing process more automated and user-friendly.

Interoperability: Ensuring interoperability between different voting systems and blockchain networks is important, especially in international or cross-border voting scenarios. Developing standardized protocols and interfaces will be essential.

Public Awareness and Education: Building public trust in the security and reliability of biometric voting systems will require extensive education and awareness campaigns to inform voters about the technology, its benefits, and safeguards in place to protect their data and privacy.

13.APPENDIX

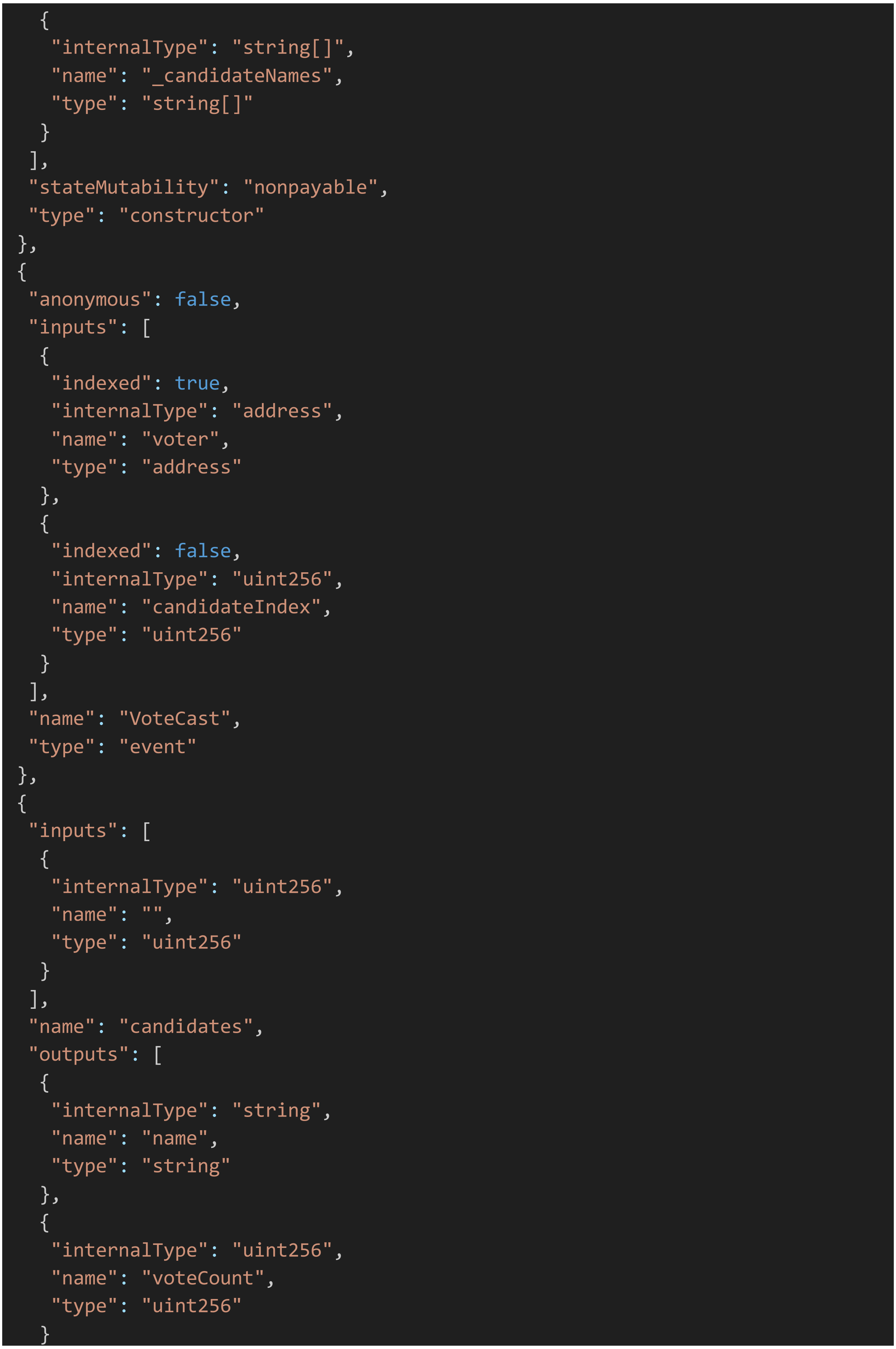
Source Code

Solidity coding :

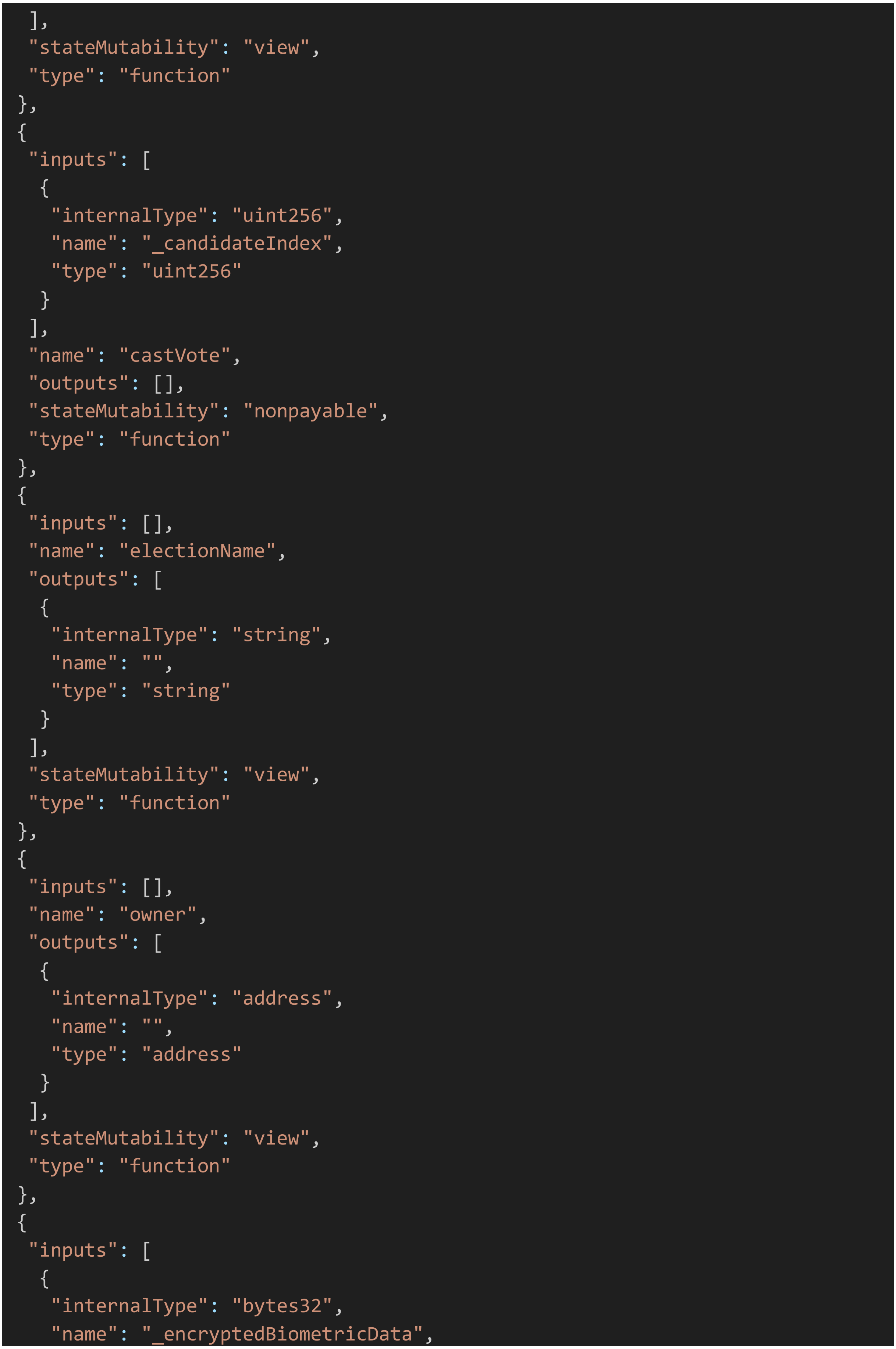
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

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);
    }
}
```

Java script :

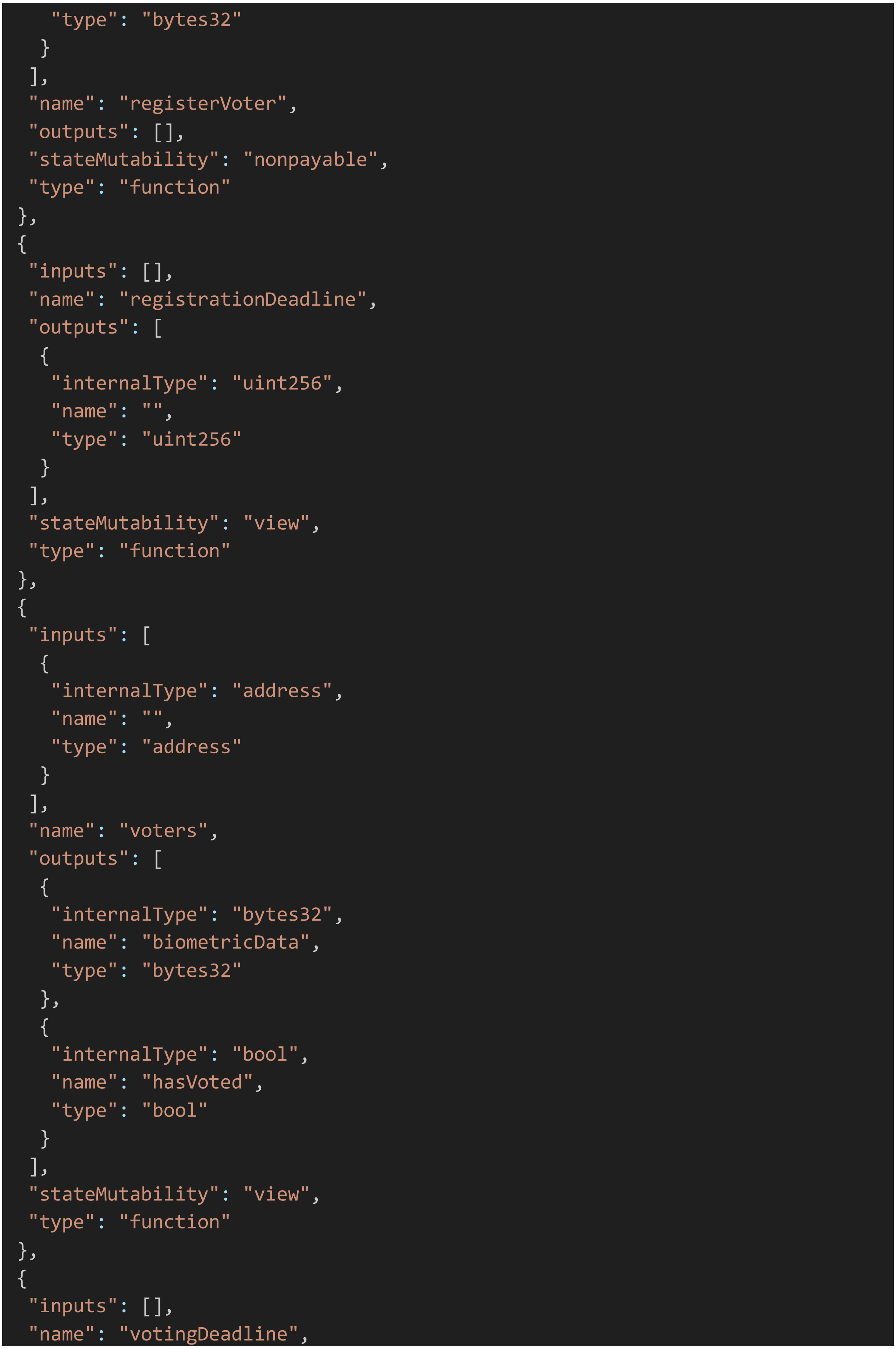
```
const { ethers } = require("ethers");

const abi = [
 {
  "inputs": [
   {
    "internalType": "string",
    "name": "_electionName",
    "type": "string"
   },
   {
    "internalType": "uint256",
    "name": "_registrationDeadline",
    "type": "uint256"
   },
   {
    "internalType": "uint256",
    "name": "_votingDeadline",
    "type": "uint256"
   },
```



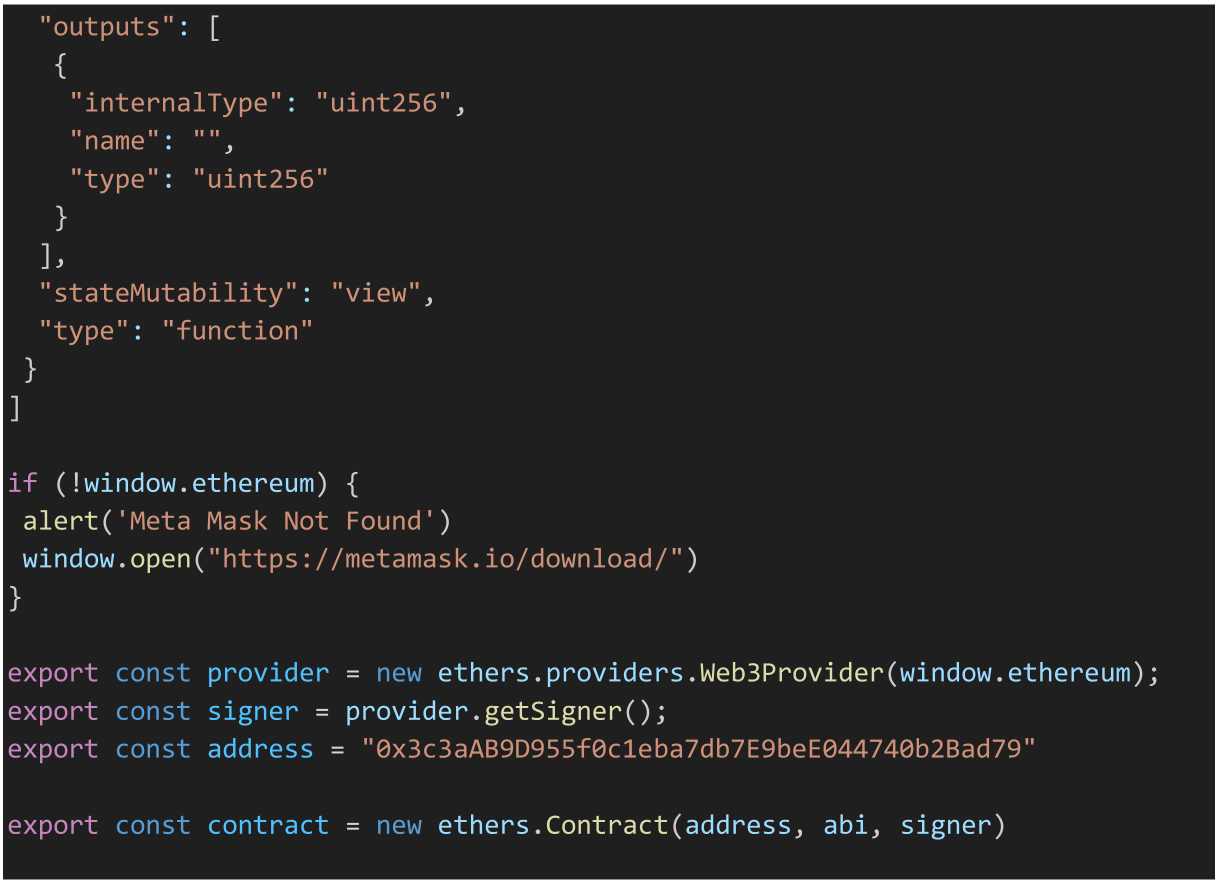
```
  {
   "internalType": "string[]",
   "name": "_candidateNames",
   "type": "string[]"
  }
 ],
 "stateMutability": "nonpayable",
 "type": "constructor"
},
{
 "anonymous": false,
 "inputs": [
  {
   "indexed": true,
   "internalType": "address",
   "name": "voter",
   "type": "address"
  },
  {
   "indexed": false,
   "internalType": "uint256",
   "name": "candidateIndex",
   "type": "uint256"
  }
 ],
 "name": "VoteCast",
 "type": "event"
},
{
 "inputs": [
  {
   "internalType": "uint256",
   "name": "",
   "type": "uint256"
  }
 ],
 "name": "candidates",
 "outputs": [
  {
   "internalType": "string",
   "name": "name",
   "type": "string"
  },
  {
   "internalType": "uint256",
   "name": "voteCount",
   "type": "uint256"
  }
```



```
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [
   {
    "internalType": "uint256",
    "name": "_candidateIndex",
    "type": "uint256"
   }
  ],
  "name": "castVote",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "electionName",
  "outputs": [
   {
    "internalType": "string",
    "name": "",
    "type": "string"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "owner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [
   {
    "internalType": "bytes32",
    "name": "_encryptedBiometricData",
```



```
   "type": "bytes32"
  }
 ],
 "name": "registerVoter",
 "outputs": [],
 "stateMutability": "nonpayable",
 "type": "function"
},
{
 "inputs": [],
 "name": "registrationDeadline",
 "outputs": [
  {
   "internalType": "uint256",
   "name": "",
   "type": "uint256"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "address",
   "name": "",
   "type": "address"
  }
 ],
 "name": "voters",
 "outputs": [
  {
   "internalType": "bytes32",
   "name": "biometricData",
   "type": "bytes32"
  },
  {
   "internalType": "bool",
   "name": "hasVoted",
   "type": "bool"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [],
 "name": "votingDeadline",
```



```
  "outputs": [
   {
    "internalType": "uint256",
    "name": "",
    "type": "uint256"
   }
  ],
  "stateMutability": "view",
  "type": "function"
 }
]

if (!window.ethereum) {
 alert('Meta Mask Not Found')
 window.open("https://metamask.io/download/")
}

export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "0x3c3aAB9D955f0c1eba7db7E9beE044740b2Bad79"

export const contract = new ethers.Contract(address, abi, signer)
```

GitHub :

https://github.com/Manokarece/naanmudhalvan

PROJECT DEMO LINK:

https://youtu.be/jsOrp5a_XSI?si=RV1JMiUI4PMrXdFc