

Naan Mudhalvan

Project Report

Biometric Security for Voting Platform

Submitted By

Team ID - NM2023TMID09503
Team Leader - SRISABARI S
Team Member 1 - VEDANAYAGAM L
Team Member 2 - SRUTHIE N
Team Member 3 - SHARUKESHAN R

Index

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 Requirement

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

8. PERFORMANCE TESTING

- 8.1 Performance Metrics

9. RESULTS

9.1 Output Screenshots

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview:

A biometric security system for a voting platform is a cutting-edge solution that leverages unique physiological or behavioural 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.

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.

1.2 Purpose:

The purpose of integrating a blockchain-based biometric security system into a voting platform is to revolutionize the electoral process by prioritizing enhanced election security, privacy protection, data integrity, and transparency. It seeks to bolster public trust and accountability by ensuring that only eligible voters can participate, safeguarding sensitive biometric data through decentralized, tamper-proof blockchain technology. User control is promoted through smart contracts, granting individuals agency over who can access their data. The system also maintains inclusivity by providing fallback mechanisms for authentication failures, ultimately fostering a new era of secure, reliable, and transparent voting procedures that empower citizens and uphold the democratic process.

2. LITERATURE SURVEY

2.1 Existing Problem:

Introducing a blockchain-based biometric security system into a voting platform is an innovative solution that aims to revolutionize the electoral process, but it comes with a host of existing problems and challenges. Firstly, the accuracy of biometric data is paramount, as even minor errors during data capture can lead to authentication failures, potentially disenfranchising eligible voters. Furthermore, concerns about data privacy persist, as securely storing biometric information in a decentralized ledger is a complex endeavor, demanding rigorous safeguards against unauthorized access. The system is also vulnerable to cyberattacks, requiring a robust cybersecurity infrastructure to ensure the system's integrity and protect sensitive biometric data. Overcoming user acceptance and accessibility barriers, especially for those with privacy concerns or limited technological proficiency, is vital. Additionally, the substantial costs and technical infrastructure demands, coupled with intricate legal and regulatory compliance, pose significant hurdles. Establishing reliable data recovery mechanisms, securing fallback options, and ensuring scalability for a growing number of voters are integral to addressing these challenges and making the blockchain-based biometric voting system a success.

2.2 References:

1. S. Shah, Q. Kanchwala, and H. Mi, "Block Chain Voting System," 2016.
2. Christian, "Desain Dan Implementasi Visual Cryptography Pada Sistem E-Voting Untuk Meningkatkan Anonymity," Institut Teknologi Bandung, 2017.
3. C. Dougherty, "[Vote Chain : Secure Democratic Voting]," 2016
4. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Wwww.Bitcoin.Org, p. 9, 2008.
5. D. A. Wijaya, Bitcoin Tingkat Lanjut. 2016.
6. H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. J. Kishigami, "Blockchain contract: A complete consensus using blockchain," 2015 IEEE 4th Glob. Conf. Consum. Electron. GCCE 2015, pp. 577–578, 2016.

2.3 Problem Statement Definition:

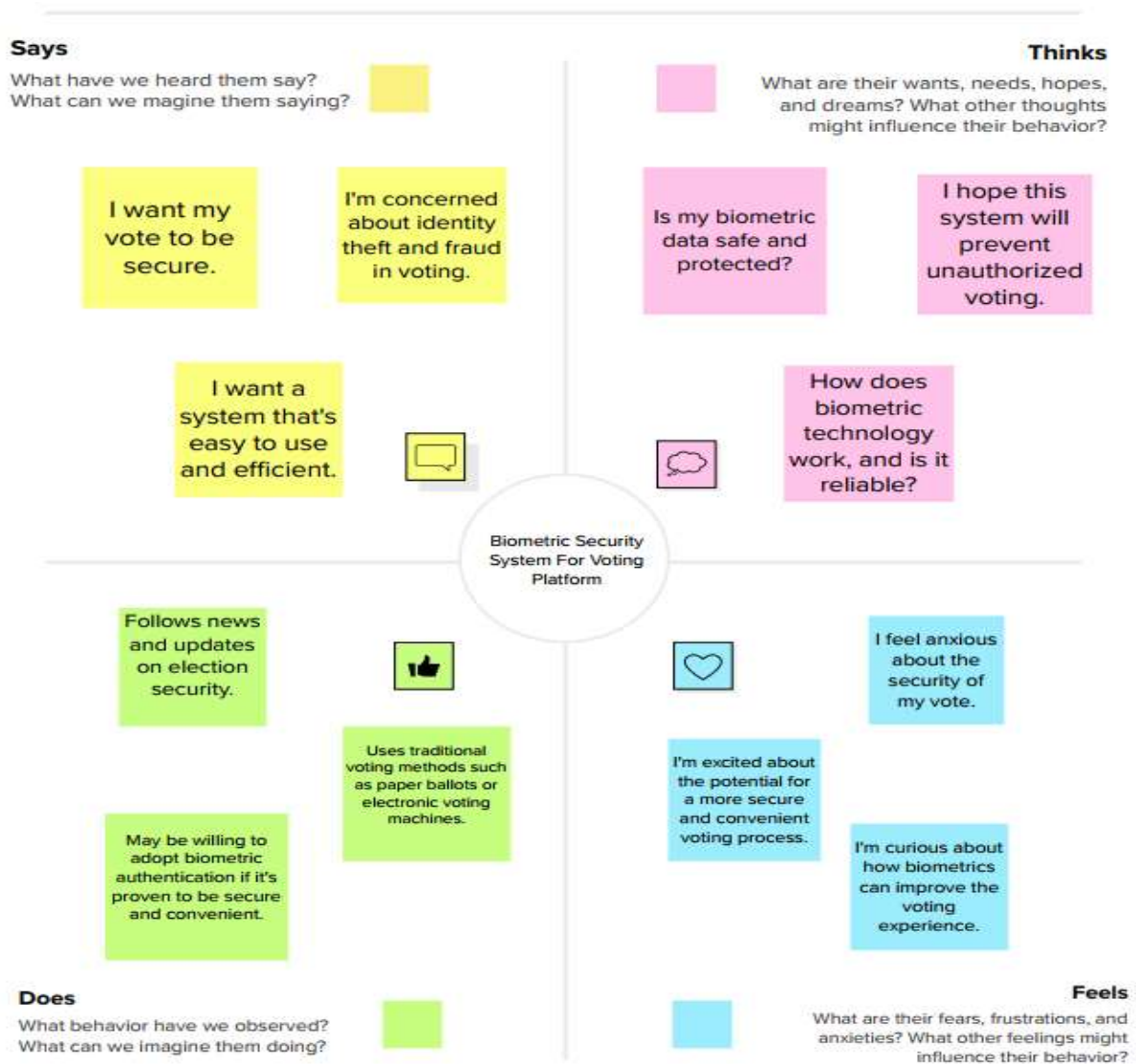
The challenge at hand is to design and implement a voting platform that combines biometric authentication with blockchain technology to address existing issues within the electoral process. These issues include ensuring the accuracy of biometric data, protecting voter privacy, defending against cyber threats, promoting user acceptance and accessibility, managing costs, adhering to legal and regulatory requirements, establishing data recovery mechanisms, and creating secure fallback options. The goal is to develop a comprehensive solution that enhances election security, increases transparency, and builds trust among voters, while simultaneously respecting individual privacy and maintaining inclusivity in the democratic process.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

- **What does the user Think and Feel?**
 - I hope this system will prevent unauthorized voting.
 - How does biometric technology work, and is it reliable?
 - Is my biometric data safe and protected?
- **What does the user Hear?**
 - Are there any rumors or misinformation circulating about the use of biometrics in voting, and how do these impact user perception?
 - Are there any expert opinions or statements from cybersecurity professionals or voting system experts that users have come across?
- **What does the user Say and Do?**
 - I want my vote to be secure.
 - I'm concerned about identity theft and fraud in voting
 - I want a system that's easy to use and efficient.
- **What does the user Pain?**
 - Users may be concerned about the security of their personal information.
 - Users may struggle with a product that is difficult to navigate, has a confusing user interface, or lacks clear instructions.
- **What does the user Gain?**
 - Biometric systems can reduce the risk of errors and voter fraud, providing users with more accurate election results and ensuring their votes are counted correctly.
 - Biometric authentication can make the voting process more accessible for individuals with disabilities, ensuring that a wider range of users can participate in elections.


Example: Biometric Security System for Voting Platform



3.2 Ideation & Brainstorming:

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

Template



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.

🕒 10 minutes to prepare

🕒 1 hour to collaborate

👤 2-8 people recommended

➔

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.

🕒 10 minutes

A

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

C

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article ➔

1

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.

🕒 5 minutes

PROBLEM

Biometric data is unique to each individual, making it difficult for anyone to fraudulently cast multiple votes or impersonate others. This strengthens the integrity of elections by minimizing voter fraud. Ensuring the security of voting machines, electronic voting systems, and online voting platforms is vital.

🧠

Key rules of brainstorming

To run a smooth and productive session

🗣️ Stay in topic.

💡 Encourage wild ideas.

🕒 Defer judgment.

👂 Listen to others.

🗣️ Go for volume.

👁️ If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

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

🕒 10 minutes

TIP

You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

SRISABARI

Implement a multi-modal biometric system that combines multiple biometric identifiers, such as fingerprints, iris scans, and facial recognition. This increases the accuracy of identity verification and makes it more difficult for fraudulent attempts.

SHARUKESHAN

Utilize blockchain technology to create a secure and transparent ledger of votes. Each vote is encrypted and added to the blockchain, making it tamper-proof and easily auditable.

VEDANAYAGAM

Use biometric data during voter registration to ensure that each voter's identity is accurately verified. This can be done during the initial registration process to create a secure database of eligible voters.

SRUTHIE

Encrypt and protect biometric data using state-of-the-art encryption techniques to prevent unauthorized access or data breaches. Data should only be accessible for authentication purposes.

Implement anti-spoofing measures to detect and prevent attempts to use fake or replicated biometric data for authentication.

Ensure the system is accessible to all voters, including those with disabilities. Provide options for voice commands, braille displays, or other accessibility features to ensure inclusivity.

Develop a system that ensures the anonymity of voters. Votes are tied to biometric data for authentication but separated from personal information to protect privacy.

Create a hybrid voting system that combines traditional paper ballots with biometric authentication. Voters can choose their preferred method while still benefiting from biometric security.

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 can 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.

Collaborate with national ID systems to streamline the voter registration process, using biometric data already collected for other purposes, such as ID cards or passports.

These ideas can be integrated into a comprehensive Biometric Security System for a Voting Platform to enhance security, transparency, and accessibility in the electoral process. The specific features and technologies used should align with the legal and regulatory requirements of the region in which the system is deployed.

Establish clear and strict policies for the retention and deletion of biometric data to address privacy concerns and data protection regulations.

Step-3: Idea Prioritization

4

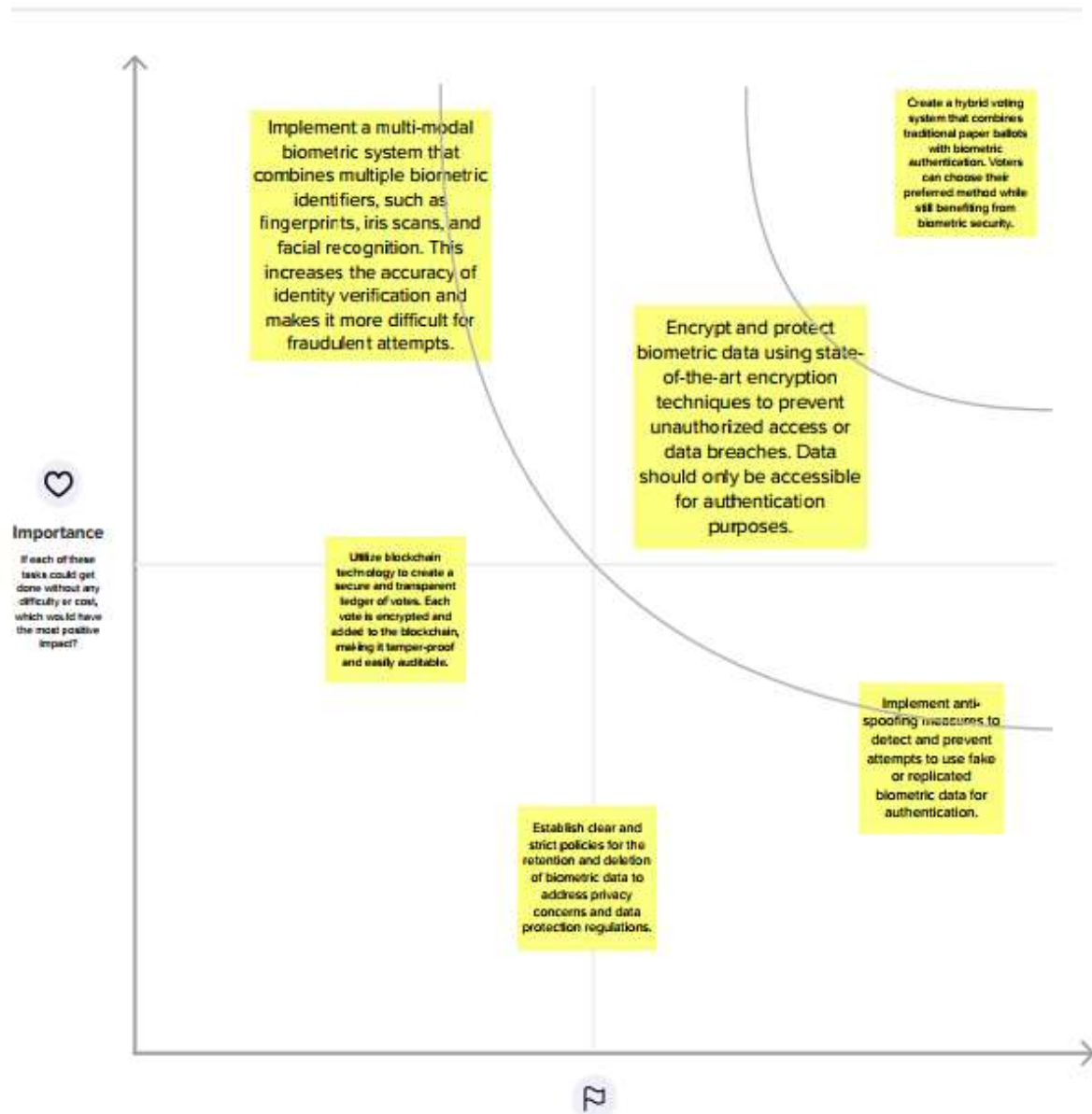
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.



4.REQUIREMENT ANALYSIS

4.1 Functional Requirement:

1. Security and Privacy:

- The system must implement robust security measures to protect against unauthorized access, data breaches, and tampering.
- It should ensure the privacy and anonymity of voters by separating voter identity from their actual vote.

2. Election Monitoring and Auditing:

- The platform should log all activities, including voter interactions and system events.
- It must allow for auditing and verification of election results, ensuring the integrity of the process.

3. Accessibility and Usability:

- The system should be user-friendly and accessible to all eligible voters, including those with disabilities.
- It must support multiple languages and provide clear instructions for the voting process.

4. Real-time Reporting and Results:

- The platform should provide real-time updates on the progress of the election and the results as they are recorded.
- It must have reporting mechanisms for election officials, candidates, and the public.

5. Redundancy and Failover:

- The system should have backup mechanisms and failover capabilities to ensure uninterrupted voting in case of system failures or network issues.

6. Compliance and Legal Requirements:

- The platform must adhere to all relevant laws, regulations, and standards regarding voting processes and data protection.

7. Integration with Biometric Hardware:

- The system should be compatible with biometric devices such as fingerprint scanners and facial recognition cameras.

8. Data Retention and Purging:

- The platform should specify data retention policies and automatically purge voter data that is no longer needed in compliance with privacy laws.

9. Scalability:

- The system should be designed to handle a scalable number of users and increased voting load during peak election periods.

10. User Support and Training:

- The platform should provide user support and training materials to election officials, voters, and administrators.

4.2 Non Functional Requirement:

1. Security:

- The system should employ the highest level of security to prevent unauthorized access and data breaches.
- It must comply with industry standards and best practices for securing biometric data.

2. Performance:

- The platform should be highly responsive, with minimal latency in user interactions and result reporting.
- It should support a large number of concurrent users without performance.

3. Reliability:

- The system should be available 24/7, with a high level of uptime.
- It must be resilient to system failures, including hardware, software, and network issues.

4. Scalability:

- The platform should be designed to scale horizontally to accommodate an increasing number of voters without performance degradation.
- It must handle peak loads during elections effectively.

5. Compatibility:

- The system should work across various devices, browsers, and operating systems.
- It must support a range of biometric hardware devices for user authentication.

6. Usability:

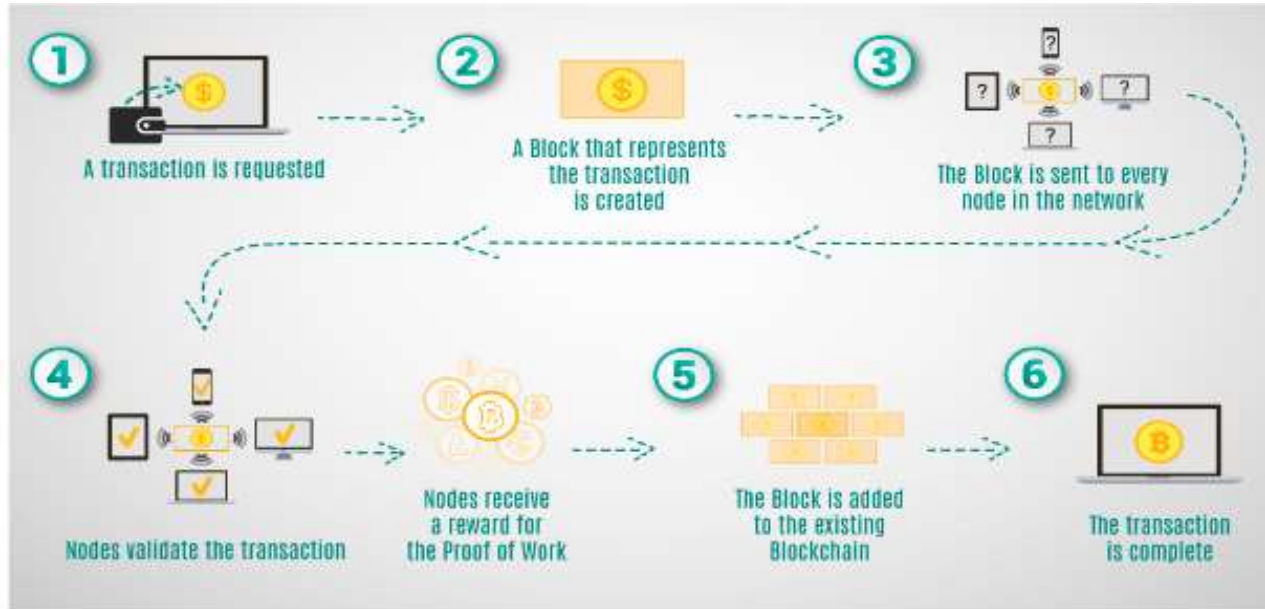
- The platform should have an intuitive and user-friendly interface to ensure that voters of all demographics can easily navigate the system.
- It must be designed with accessibility features for individuals with disabilities.

7. Interoperability:

- The system should integrate with other election-related systems, such as voter registration databases and result aggregation platforms.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories:

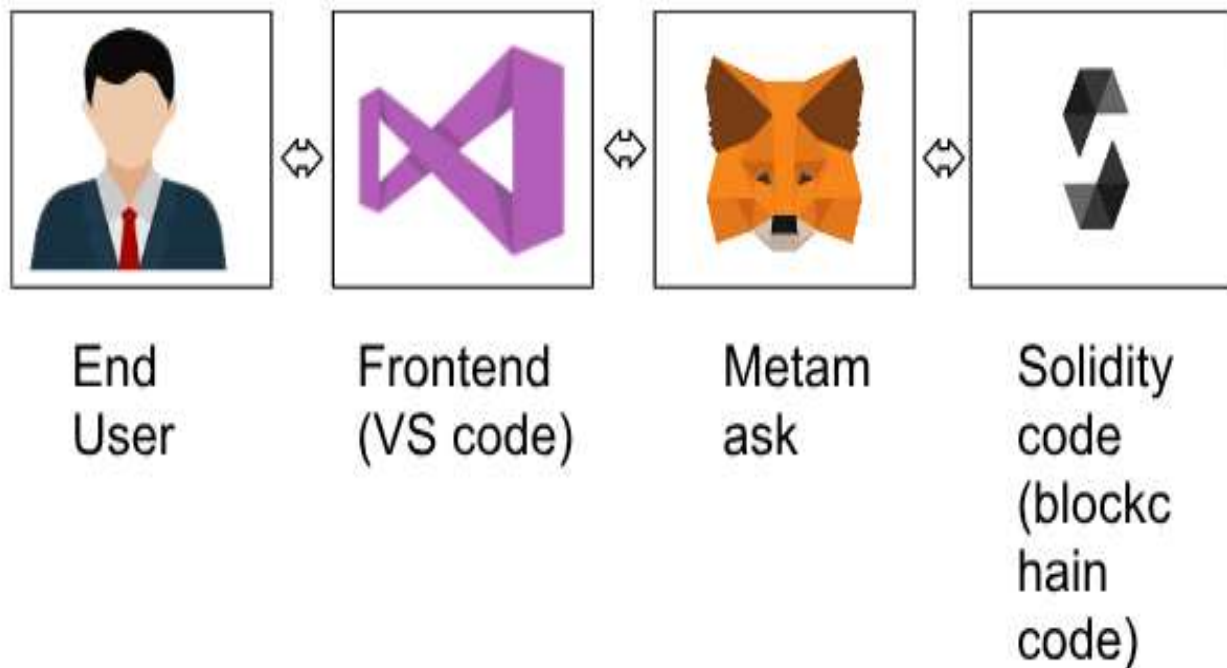


5.2 Solution Architecture:

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.



Tech Stack:

- REMIX IDE
- VS CODE
- NODE JS
- METAMASK

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture:

Designing the technical architecture for a biometric security voting platform requires a scalable, secure, and resilient system that can handle a large number of concurrent users while protecting the integrity of the voting process. Here is a high-level technical architecture for such a platform:

1. User Interface Layer:

Web Application: The user interface can be a web-based application accessible from various devices, including desktops, tablets, and smartphones. It should offer a user-friendly and accessible interface for voters.

2. Application Layer:

Authentication and Authorization: This layer handles user authentication and authorization, ensuring that voters are eligible and authorized to participate in the election. It also manages the biometric authentication process.

Voter Information Management: Manages voter registration, updates, and deletions. It also maintains the database of registered voters, including their biometric data.

Ballot Generation and Management: Generates electronic ballots based on the specific election or referendum. It ensures the correct ballot is presented to each voter.

Vote Casting and Recording: Handles the secure casting of votes and records each vote, linking it to the respective voter's identity.

Security Services: Implements security mechanisms to protect the system from cyber threats and unauthorized access.

Audit and Logging: Manages detailed audit logs to track all system activities, including voter interactions and administrative actions.

Communication Services: Provides notification services for election-related information and real-time updates to voters, candidates, and election officials.

Integration Services: Facilitates integration with external systems, such as voter registration databases and result aggregation platforms.

3. Business Logic Layer:

Voter Eligibility: Ensures voters meet eligibility criteria and validates their identity using biometric data.

Ballot Processing: Manages the presentation of the correct ballot to voters and ensures the proper recording of votes.

Election Monitoring and Reporting: Handles real-time reporting of election progress and results.

Privacy and Anonymity: Implements mechanisms to separate voter identity from vote choice while protecting voter privacy.

4. Data Layer:

Voter Database: Stores voter information, including personal details and biometric data. This database should be highly secure, with encryption and access control measures in place.

Ballot and Vote Data: Stores electronic ballots, cast votes, and results securely. Data encryption and backup procedures are crucial.

Audit Log Database: Records and stores detailed audit logs of all system activities.

External Systems Data: Integrates with external databases and systems, such as voter registration databases.

5. Security Layer:

Biometric Data Security: Ensures the highest level of security for biometric data, including encryption and secure storage.

Network Security: Implements network security measures to protect data in transit.

Authentication and Authorization: Manages user access, ensuring only authorized users can interact with the system.

Data Protection: Implements data protection mechanisms to safeguard voter information and voting records.

6. Infrastructure Layer:

Servers and Cloud Services: The platform can be hosted on a combination of on-premises servers and cloud services to ensure scalability and redundancy.

Load Balancers: Distributes incoming traffic across multiple servers to ensure optimal performance.

Backup and Redundancy: Implements backup and disaster recovery solutions to maintain system availability.

Network Infrastructure: Provides the underlying network infrastructure to connect all system components securely.

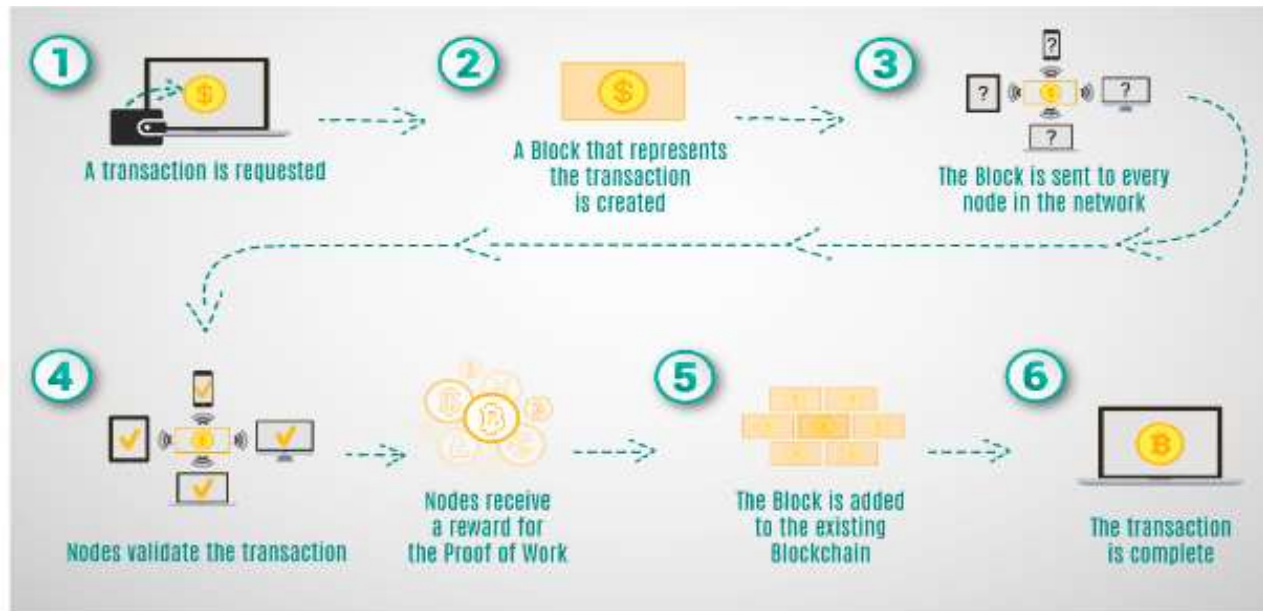
7. Biometric Hardware Integration:

Interfaces with biometric devices like fingerprint scanners, facial recognition cameras, and other biometric sensors for voter authentication.

8. Monitoring and Management:

Incorporates tools and processes for monitoring system performance, health, and security.

Enables system administrators to manage and maintain the platform effectively.



6.2 Sprint Planning & Estimation:

1. Product Backlog Refinement:

Begin by ensuring that your product backlog is well-defined, with a prioritized list of features and user stories that need to be implemented in the platform.

2. Define Sprint Goals:

Each sprint should have a clear goal or objective, such as implementing specific features, enhancing security, or improving system performance.

3. Sprint Duration:

Determine the sprint duration, typically 2-4 weeks, based on the project's complexity and the team's velocity.

4. Sprint Planning Meeting:

Hold a sprint planning meeting at the beginning of each sprint. During this meeting, the team will select a set of user stories or backlog items to work on during the sprint.

5. User Story Selection:

Based on the sprint goal and priority, the team should select user stories from the product backlog to work on during the sprint. These stories should be well-defined

and meet the "Definition of Ready" criteria.

6. Break Down User Stories:

For each selected user story, the team should break it down into smaller tasks or sub-tasks. These tasks should be granular enough to be estimated and tracked.

7. Estimation Techniques:

Use estimation techniques such as story points, t-shirt sizing, or hours to estimate the effort required for each task or user story. Story points are commonly used in Agile for relative estimation.

8. Team Involvement:

Involve the entire development team in the estimation process. Team members with different skills and perspectives can provide valuable input.

9. Velocity and Capacity:

Calculate the team's velocity, which is the average number of story points or tasks completed in previous sprints. Use this velocity to determine the team's capacity for the upcoming sprint.

10. Commitment:

Based on the team's capacity and the estimated effort for user stories and tasks, the team commits to completing a certain amount of work during the sprint. This commitment should be realistic and achievable.

11. Sprint Backlog:

Create a sprint backlog that lists all the selected user stories and their associated tasks for the sprint. This serves as the plan for the sprint.

12. Daily Stand-ups:

Conduct daily stand-up meetings to review progress, address any obstacles, and make adjustments as needed.

13. Review and Retrospective:

At the end of the sprint, hold a sprint review to demonstrate the completed work to stakeholders. Also, conduct a sprint retrospective to reflect on the sprint's process and identify areas for improvement.

14. Adjustments:

Use the insights from the sprint retrospective to make adjustments to future sprint planning and execution to continuously improve the project's efficiency and quality.

15. Documentation:

Document all sprint planning and estimation results, including the sprint backlog and any changes made during the sprint.

6.3 Sprint Delivery Schedule:

Sprint 1: Project Kick-off and Setup

Duration: 2 weeks

Goals:

Establish the project team and roles.

Set up the development environment.

Define the project scope and requirements.

Deliverables:

Project team roles and responsibilities documented.

Development environment set up.

Initial project plan and scope document.

Sprint 2: Voter Registration and Authentication

Duration: 2 weeks

Goals:

Implement voter registration functionality.

Develop user authentication using biometric data.

Deliverables:

Voter registration feature.

Biometric authentication module.

Sprint 3: Ballot Generation and Management

Duration: 2 weeks

Goals:

Create the module for generating electronic ballots.

Implement the management of different types of ballots.

Deliverables:

Ballot generation functionality.

Ballot management system.

Sprint 4: Vote Casting and Recording

Duration: 2 weeks

Goals:

Develop the vote casting interface.

Implement secure vote recording and time stamping.

Deliverables:

Vote casting feature.

Secure vote recording mechanism.

Sprint 5: Security Enhancements

Duration: 2 weeks

Goals:

Enhance platform security, including data protection and access control.

Perform a security audit.

Deliverables:

Improved security measures.

Results of the security audit with recommended actions.

Sprint 6: Election Monitoring and Reporting

Duration: 2 weeks

Goals:

Implement real-time reporting of election progress.

Develop a results reporting system.

Deliverables:

Real-time election progress reporting.

Results reporting feature.

Sprint 7: Integration and Testing

Duration: 2 weeks

Goals:

Integrate the platform with external systems and databases.

Deliverables:

Successful integration with external systems.

Comprehensive test reports.

Sprint 8: Final Testing and Deployment

Duration: 2 weeks

Goals:

Finalize testing and quality assurance activities.

Deliverables:

Completed testing and quality assurance reports.

Sprint 9: Deployment and Post-Deployment Activities

Duration: 2 weeks

Goals:

Deploy the system for an actual election.

Monitor the system's performance during the election.

Deliverables:

Deployed and operational system.

Sprint 10: Post-Election Analysis and Improvements

Duration: 2 weeks

Goals:

Analyze the election results and system performance.

Identify areas for improvement based on the election experience.

Deliverables:

Post-election analysis report.

Action plan for system improvements.

7.CODING & SOLUTIONING

7.1 Feature 1:

Solidity Code of Our Project:

```
// 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:

Javascript Code for our Project:

```

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

const abi = [
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_electionName",
                "type": "string"
            },
            {
                "internalType": "uint256",
                "name": "_registrationDeadline",
                "type": "uint256"
            }
        ],
        "name": "register",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "nonpayable"
    }
]

```

```

{
  "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/")
}

```

```

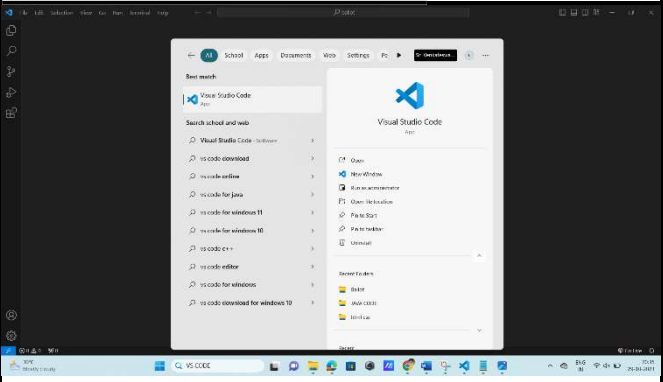
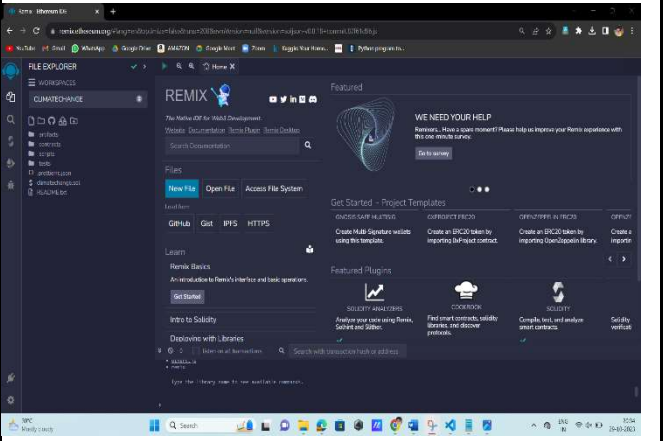
export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "0xD52c1b477B072d5A9cA2f73690Ba14334d0D8Ce0"

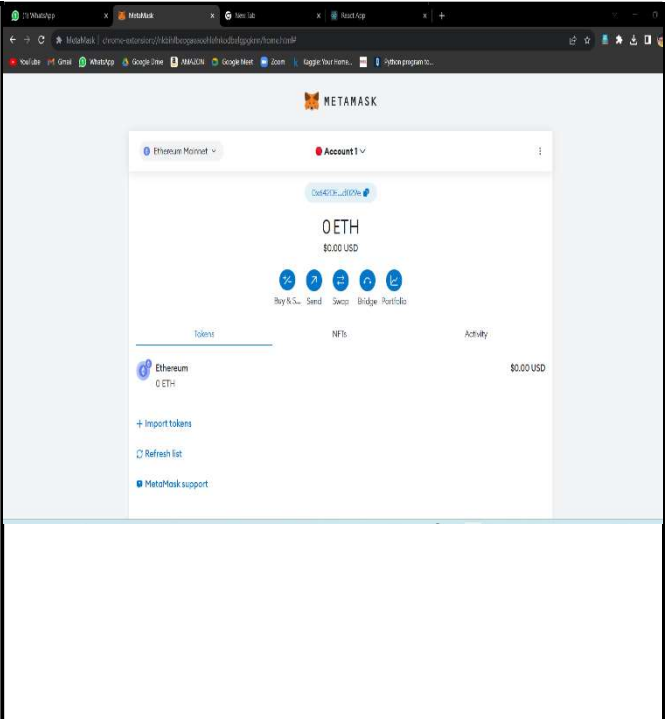
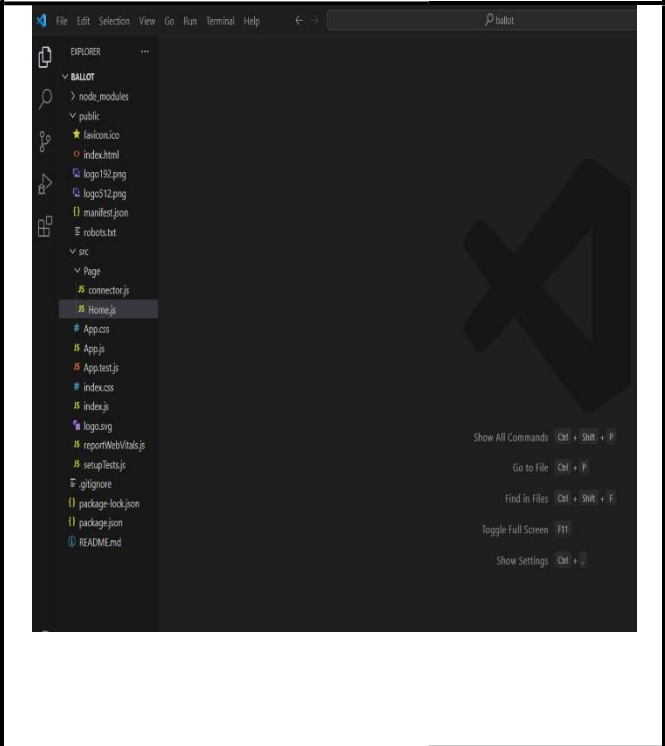
export const contract = new ethers.Contract(address, abi, signer)

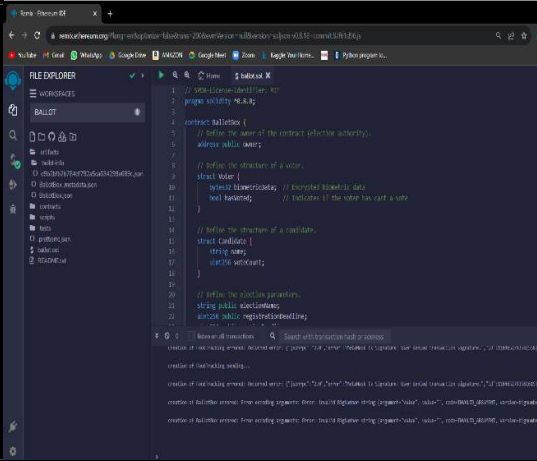
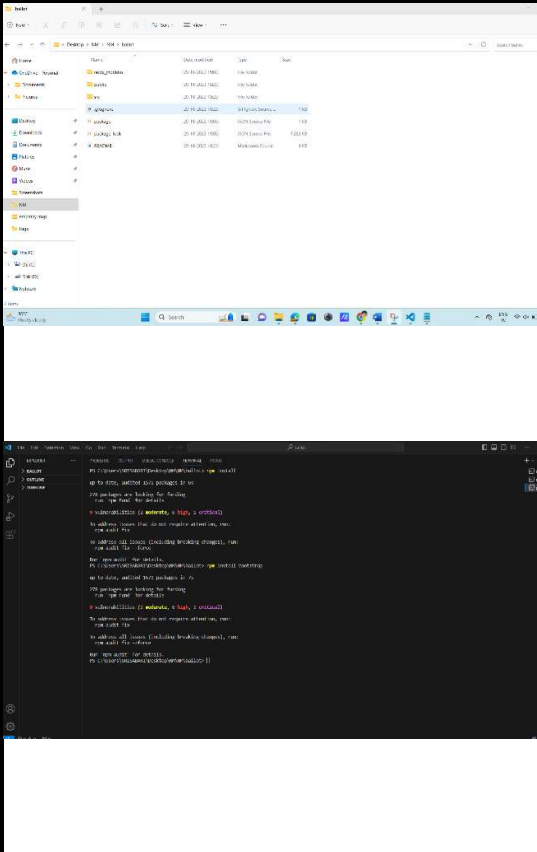
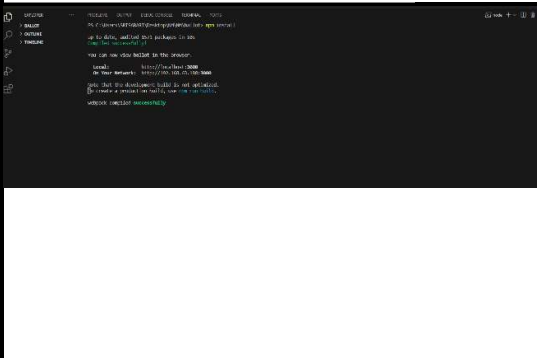
```

8.PERFORMANCE TESTING

8.1 Performance Metrics:

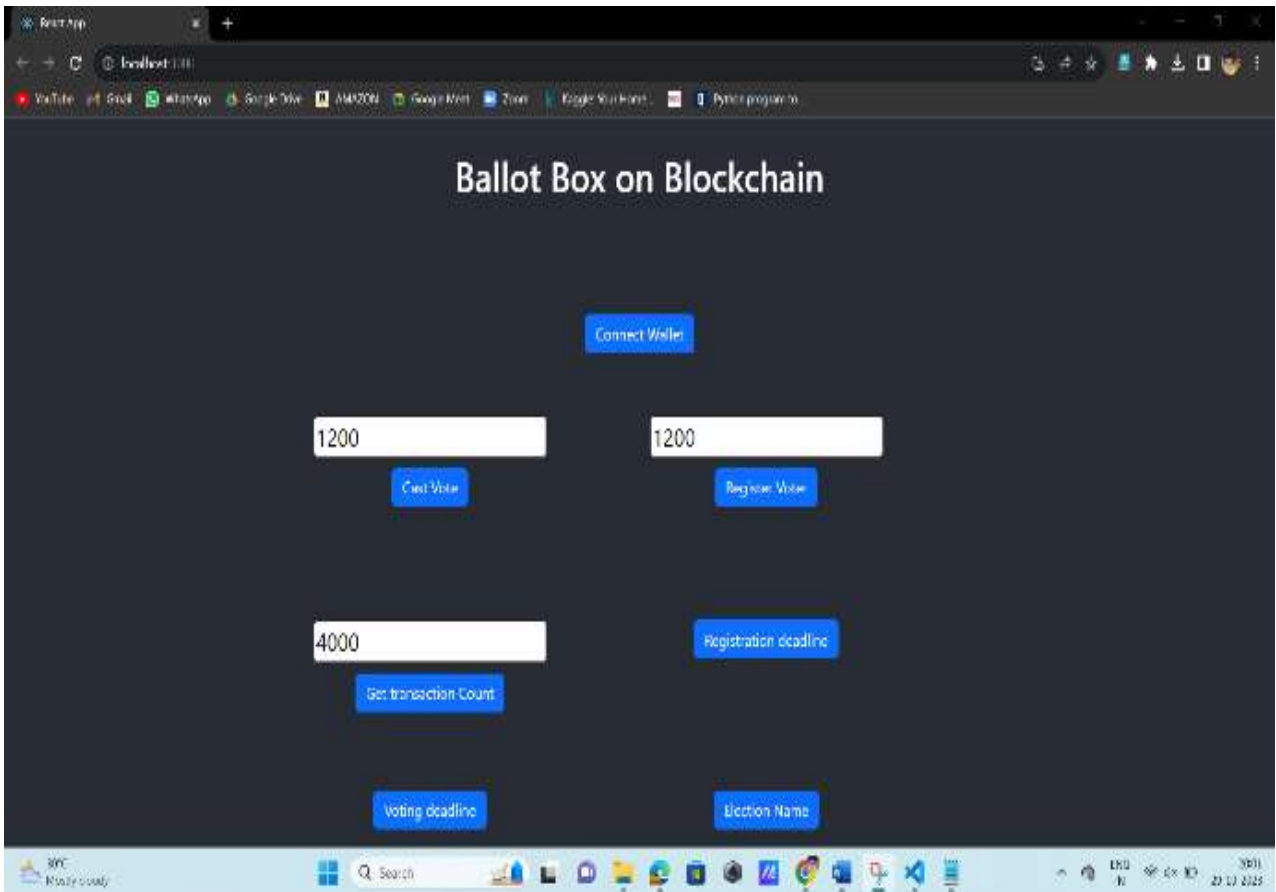
S.No	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite	<div>A screenshot of the Visual Studio Code application. The search bar at the top contains the text "Visual Studio Code". Below the search bar, a list of search results is displayed, including "Visual Studio Code - Software", "vs code download", "vs code online", "vs code for java", "vs code for windows 11", "vs code for windows 10", "vs code c++", "vs code editor", "vs code for windows", and "vs code download for windows 10". The right sidebar shows the "Visual Studio Code" extension page with a "Get" button and a list of features.</div> <div>A screenshot of the Remix IDE homepage. The page features a dark theme and a sidebar on the left with a "FILE EXPLORER" and "WORKSPACES" section. The main content area displays the "REMIX" logo, a "Get Started" button, and a "WE NEED YOUR HELP" section. Below these, there are sections for "Get Started - Project Templates", "Learn Remix Basics", "Intro to Solcity", and "Deploy with Libraries". The bottom of the page shows a footer with the Remix logo and a "Get Started" button.</div>

			
2.	Extract the zip files	Open to vs code	

3.	Remix Ide platform exporting	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 utilities. Open cmd enter commands 1.npminstall 2.npmbootstrap 3..npmstart	
5	{LOCALHOSTI PADDRESS}	Copy the address and open it to chrome so you can see the frontend of your project.	

9.RESULTS

9.1 Output Screenshots:



10.ADVANTAGES & DISADVANTAGES

Advantages:

Enhanced Security:

Biometric authentication adds an extra layer of security, making it more difficult for unauthorized individuals to impersonate voters.

Reduced Voter Fraud:

Biometric verification helps prevent voter fraud by ensuring that each voter can only vote once, reducing the risk of double voting or identity theft.

Accessibility:

Electronic voting platforms can be designed to be more accessible to voters with disabilities, improving inclusivity.

Efficiency and Speed:

Voting can be conducted more efficiently, potentially reducing waiting times at polling stations and expediting the vote counting process.

Reduced Administrative Costs:

Automated processes can reduce administrative overhead associated with traditional paper-based voting systems.

Real-time Reporting:

Election results can be reported in real-time, providing transparency and reducing the time required to announce outcomes.

Disadvantages:

Technical Challenges:

Implementing biometric authentication and ensuring its reliability can be technically challenging. False negatives or positives can occur.

Transparency Concerns:

Some stakeholders may be concerned about the transparency and verifiability of electronic voting systems, as it can be challenging to ensure that votes are accurately recorded and counted.

11.CONCLUSION

In conclusion, the development and implementation of a biometric security voting platform represent a significant step toward modernizing and enhancing the electoral process. This project offers several advantages, including increased security, reduced voter fraud, improved efficiency, and the potential for real-time reporting. It can also promote inclusivity and accessibility, reduce administrative costs, and have a positive environmental impact.

However, it is crucial to acknowledge and address the potential disadvantages and challenges associated with this project. These include concerns about biometric privacy, technical difficulties, access and inclusivity issues, voter authentication challenges, system vulnerabilities, and high costs. Overcoming these challenges requires a comprehensive approach, encompassing robust security measures, public trust-building, and compliance with legal and regulatory requirements.

Ultimately, the success of a biometric security voting platform project hinges on striking a balance between the advantages it offers and the challenges it presents. The project should be guided by a commitment to transparency, security, inclusivity, and the protection of voters' privacy. As technology continues to evolve, the continued improvement and adaptation of such platforms will be necessary to ensure that they meet the evolving needs of the electoral process while safeguarding its integrity.

12.FUTURE SCOPE

The future scope for a biometric security voting platform project is dynamic and promising, with the potential to further revolutionize the electoral process. Here are some areas where the project can expand and evolve:

Enhanced Security: Continuously improving security measures to safeguard against evolving cyber threats and to protect biometric data is essential. Incorporating advanced encryption and multi-factor authentication methods can enhance the platform's resilience.

Blockchain Integration: Incorporating blockchain technology can add an extra layer of transparency and security to the voting process. Blockchain can create immutable and verifiable records of votes while ensuring anonymity.

Mobile Voting: Developing mobile applications for voting can increase accessibility and convenience for voters. This could include secure mobile biometric authentication and voting.

AI and Machine Learning: Integrating AI and machine learning can help in fraud detection, anomaly identification, and improving the overall performance and security of the system.

Post-election Auditing: Implementing more advanced post-election auditing and verification processes can enhance the transparency and accountability of election results.

Public Engagement: Developing features for public engagement, such as real-time result tracking, can foster trust and participation in the electoral process.

Continual Compliance: Adhering to evolving data protection and privacy regulations while ensuring legal and regulatory compliance at all times.

The future scope for a biometric security voting platform is vast, with the potential to transform the way elections are conducted, enhancing security, accessibility, and transparency. As technology evolves, it is essential to adapt and stay ahead of emerging challenges and opportunities, while always prioritizing the integrity of the democratic process.

13.APPENDIX

- **Source Code :**

<https://drive.google.com/file/d/1AVCY6YsajfMYbybEzp45wWWJDbh3-1qh/view>

- **Github Link :**

<https://github.com/SRISABARI46/NM2023TMID09503>

- **Demo Link :**

https://drive.google.com/file/d/1fAsbqV_fm6F6MEgSc-37GQcTw8pOCTNd/view