20CS812 - PROJECT WORK

DECENTRALIZED VOTING SYSTEM USING BLOCKCHAIN

A PROJECT REPORT

Submitted by

HEMANTH SIVA SAI J	**111721102176**
VIGNARAJ G	**111721102166**
ADITHYA RAMESH SANKAR T	**111721102156**

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

R.M.K. ENGINEERING COLLEGE
(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

ANNA UNIVERSITY: CHENNAI 600 025

MARCH 2025

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

It is certified that this project report, **"DECENTRALIZED VOTING SYSTEM USING BLOCKCHAIN,"** is the bonafide work of **HEMANTH SIVA SAI J (111721102176), VIGNARAJ G (111721102166), ADITHYA RAMESH SANKAR T(111721102156),** who carried out the 20CS812 Project Work under my supervision.

SIGNATURE

Dr. T. Sethukarasi, M.E., M.S. Ph.D.,
Professor and Head
Computer Science and Engineering
R.M.K. Engineering College
R.S.M. Nagar, Kavaraipettai,
Tiruvallur District– 601206.

SIGNATURE

Dr. Ramesh T, M.E., Ph.D.,
Supervisor
Associate Professor
Computer Science and Engineering
R.M.K. Engineering College
R.S.M. Nagar, Kavaraipettai,
Tiruvallur District–601206.

Submitted for the Project Viva–Voce held on ..... at **R.M.K. Engineering College**, Kavaraipettai, Tiruvallur District– 601206.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We earnestly portray our sincere gratitude and regard to our beloved **Chairman Shri R. S. Munirathinam, our Vice Chairman, Shri. R. M. Kishore** and **our Director, Shri. R. Jyothi Naidu,** for the interest and affection shown towards us throughout the course.

We convey our sincere thanks to our **Principal**, **Dr. K. A. Mohamed Junaid,** for being the source of inspiration in this college.

We reveal our sincere thanks to our **Professor and Head of the Department of Computer Science and Engineering, Dr. T. Sethukarasi**, for her commendable support and encouragement for the completion of our project.

We would like to express our sincere gratitude to our project guide, **Dr. Ramesh T, Assistant Professor,** for their valuable suggestions regarding the successful completion of this project globally.

We take this opportunity to extend our thanks to all faculty members of the Department of Computer Science and Engineering, parents, and friends for all that they meant to us during the crucial times of the completion of our project.

TABLE OF CONTENTS

ABSTRACT

Traditional voting systems face challenges such as fraud, lack of transparency, and security risks. This project proposes a Decentralized Voting System that leverages blockchain technology and smart contracts to ensure secure, transparent, and tamper-proof elections. The immutability of blockchain protects votes, while smart contracts automate voter verification, vote casting, and tallying, thereby minimizing errors and fraud. The system features multi-factor authentication for secure access, cryptographic techniques for voter anonymity, and real-time auditing to foster trust. Scalability solutions like sidechains and sharding enable efficient management of large-scale elections. A user-friendly interface ensures voting accessibility for everyone, including remote and differently-abled voters. Furthermore, decentralization removes single points of failure, decreasing the risk of cyberattacks and election manipulation. The system enhances transparency by allowing public verification of votes without compromising voter privacy. By reinforcing electoral integrity, this approach boosts voter confidence, guarantees fair elections, and encourages greater participation in democratic processes.

Keywords: Decentralized Voting System, Blockchain, Smart Contracts, Election Security, Transparency, Voter Verification, Cryptographic Techniques, Real-Time Auditing, Scalability, Electoral Integrity.

List of Figures

LIST OF ABBREVIATIONS

S.NO	ACRONYM	EXPANDED FORM
1	EVM	Electronic Voting Machines
2	ZKP	Zero-Knowledge Proof
3	POS	Proof of Stake
4	POA	Proof of Authority
5	DID	Decentralized identity
6	HTTPS	Hypertext Transfer Protocol Secure

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Traditional voting systems are facing significant challenges that undermine the integrity, efficiency, and accessibility of the electoral process. These challenges not only compromise democracy but also discourage voter participation, leading to skepticism and mistrust in election outcomes. Below are some of the major issues associated with traditional voting systems:

1. Prone to Tampering, Fraud, and Lack of Transparency

Election fraud remains a serious concern in many countries, where traditional voting methods can be manipulated through ballot stuffing, vote-buying, and miscounting. The lack of transparency in vote tallying and election procedures further exacerbates doubts about the credibility of the process. Without a verifiable and tamper-proof system, election results can be contested, leading to political instability.

2. Time-Consuming and Error-Prone Manual Vote Counting

The process of manually counting votes is not only slow but also highly susceptible to human error. Mistakes in tallying can lead to misrepresentation of results, recount demands, and prolonged election disputes. In large-scale elections, delays in vote counting can create uncertainty and unrest among the public.

3. Vulnerability to Cyber Attacks and Data Breaches Centralized digital voting systems, such as electronic voting machines (EVMs) and online voting databases, are attractive targets for hackers. A single cyberattack can compromise sensitive voter data,

manipulate election results, and erode public trust. The security of these systems is crucial to prevent unauthorized access and ensure fair elections.

4. Lack of Trust Due to Opacity in the Electoral Process

Many voters feel disconnected from the electoral process due to its opaque nature. A lack of transparency in how votes are recorded, counted, and verified raises concerns about manipulation. Without a reliable method to independently verify their votes, citizens may feel uncertain about whether their votes were accurately counted.

5. High Costs of Deployment and Management

Organizing elections using traditional methods is expensive due to costs related to ballot printing, secure storage, transportation, staffing, and logistical arrangements. Governments must allocate substantial resources to ensure smooth elections, yet inefficiencies and mismanagement can still arise, leading to financial wastage.

6. Limited Accessibility for Remote and Differently-Abled Voters

Many voters, including those living in remote areas, overseas citizens, and individuals with disabilities, face difficulties in physically casting their votes. Traditional polling stations are not always accessible, which discourages participation and results in lower voter turnout. A more inclusive voting system is necessary to ensure that every eligible voter can exercise their right to vote conveniently.

1.2 LITERATURE REVIEW

1. **V Lalitha, S Samundeswari, R Roobinee, and Lakshme S Swetha**

   This study recognizes the critical role of security in voting systems, particularly in addressing the vulnerabilities of traditional voting methods. The authors highlight how conventional voting mechanisms suffer from transparency issues, data manipulation risks, and trust deficits among voters. They emphasize the need for a more robust and secure framework to mitigate these challenges. However, the secure voting system. Additionally, potential cybersecurity risks, such as study falls short in providing a detailed implementation strategy for deploying vulnerabilities in blockchain consensus mechanisms or attacks on cryptographic protocols, are not thoroughly addressed, leaving room for further exploration in securing decentralized voting systems.

2. **Sachin Kumar, Shoaib Akhtar, Soumalya Ghosh, and Kavita Saini**

   This research focuses on security and transparency, proposing blockchain as a viable solution to enhance electoral integrity. The study explores the advantages of using a decentralized ledger to prevent vote tampering and ensure immutability. While the authors provide strong theoretical support for blockchain integration, they also acknowledge the risks associated with cybersecurity threats. Issues such as phishing attacks, denial-of-service attacks, and vulnerabilities in smart contract execution are identified as potential risks. Furthermore, the study raises concerns about the digital divide, noting that limited internet access and technological literacy can create barriers to adoption, particularly in rural or underdeveloped regions. The legal and regulatory challenges associated with adopting blockchain in national elections

remain another key gap, requiring policymakers to establish clearer governance frameworks.

3. Rithvik Rao Rewatkar, Devansh Agarwal, Anmol Khandelwal, and Subho - Upadhyay

The authors recognize scalability as a major factor in the successful implementation of decentralized voting systems. The study highlights how blockchain-based voting needs to handle large-scale elections efficiently without compromising security and speed. While the authors propose solutions such as sharding and sidechains, they do not delve deeply into their practical feasibility or provide real-world testing results. Additionally, the study underscores the importance of privacy and security, stressing that decentralized voting systems must balance voter anonymity with verification mechanisms. However, the ethical and social implications of blockchain voting, including issues related to coercion, vote buying, and voter access, are not fully analyzed, necessitating further research in these areas.

4. Ashish Balti, Abhishek Prabhu, Sanskar Shahi, Shrutika Dahifale, and Vrajesh Maheta

This study explores the limitations of electronic democracy and advocates for mobile-based voting solutions. The authors emphasize how mobile accessibility can improve voter participation rates, especially among younger demographics and remote voters. However, while the study highlights mobile voting's advantages, it lacks a clear and structured approach to integrating blockchain for security and transparency. Concerns regarding security and privacy risks, such as malware threats, unauthorized access, and user authentication vulnerabilities, are raised but not addressed with definitive solutions. The absence of a detailed implementation.

1.3 SYSTEM REQUIREMENTS

1.3.1. Hardware Requirements

1. **Processor (CPU):** Multi-core processor with a clock speed of at least 2 GHz..
2. **Memory (RAM):** A minimum of 4GB RAM.
3. **Storage (SSD):** A minimum of 10GB.
4. **Network Infrastructure:** A moderate-speed, moderate-latency internet connection.

1.3.2. Software Requirements

1. **Operating System:**
    a. Linux-based OS (Ubuntu, Debian) for server-side deployment due to its security and efficiency in handling blockchain applications.
    b. Windows or macOS for front-end development and administrative tools
    c. Solidity for writing and deploying smart contracts.
2. **Blockchain Frameworks:**
    a. Ethereum for smart contract development and decentralized application execution.
    b. Hyperledger Fabric for permissioned blockchain networks in private elections.
    c. Solidity for writing and deploying smart contracts.
3. **Development Tools & Programming Languages:**
    a. NodeJS for backend logic, API development, and data processing.
    b. JavaScript (jQuery) for frontend development, ensuring a dynamic and user-friendly voting interface.
    c. Web3.js or Ethers.js for blockchain interaction with smart contracts.

4. **Database & Storage Solutions:**
    a. IPFS (InterPlanetary File System) for decentralized and secure storage of voting records.
5. **Cryptographic Libraries & Security Tools:**
    a. OpenSSL for encryption and secure transmission of voting data.
    b. MetaMask or WalletConnect for voter authentication and blockchain transactions.
    c. Zero-Knowledge Proof (ZKP) libraries for ensuring privacy in voter identity verification.
6. **Testing & Deployment Tools:**
    a. Ganache for testing Ethereum-based smart contracts in a local blockchain environment.
    b. Truffle Suite for smart contract development and testing automation

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing Model

The current centralized voting system suffers from several inefficiencies, leading to significant operational and security challenges. The primary challenges addressed by this project include:

- **Manual Vote Counting:** Traditional systems rely on manual vote counting, which is time-consuming and prone to human errors.
- **Susceptibility to Fraud:** Centralized databases create opportunities for tampering, ballot stuffing, and vote manipulation, undermining electoral integrity.
- **Lack of Transparency:** Voters and stakeholders often have limited visibility into the vote-counting process, leading to distrust in election outcomes.
- **Cybersecurity Risks:** Centralized databases are vulnerable to hacking, data breaches, and cyberattacks, increasing the risk of election interference.
- **Limited Accessibility:** Remote and differently-abled voters face significant barriers to participation due to physical and logistical constraints.
- **High Costs:** The deployment and maintenance of traditional voting systems require significant financial and human resources.

These challenges highlight the need for a more secure, efficient, and transparent voting system that leverages decentralized technology to address these shortcomings.

2.2 Proposed Model:

The proposed decentralized voting system introduces blockchain technology to create a transparent, secure, and efficient electoral process. Smart contracts ensure automated vote tallying, while decentralized storage eliminates the risk of single points of failure. Multi-factor authentication enhances voter security, and a user-friendly interface increases accessibility. The system is designed to scale, supporting national-level elections while maintaining integrity and auditability.

Key Components of the Project Scope

1. **Blockchain Infrastructure:**
    - Implementation of a decentralized ledger to ensure vote immutability and prevent unauthorized modifications.
    - Integration of a consensus mechanism such as Proof of Stake (PoS) or Proof of Authority (PoA) for secure and efficient vote validation.
2. **Smart Contract Implementation:**
    - Development of secure smart contracts to automate vote tallying, voter authentication, and fraud detection.
    - Ensuring that contract logic is immutable and publicly verifiable to enhance trust and transparency.

3. **User-Friendly Interface:**
    - Development of a responsive web and mobile-based interface for voters to cast their votes securely and intuitively.
    - Accessibility features such as language options and support for visually impaired users.

4. **Real-Time Vote Auditing and Transparency:**
    - Implementation of a public, auditable ledger that allows election observers to track vote counts in real-time.
    - Zero-knowledge proofs (ZKP) and encryption techniques to ensure voter anonymity while maintaining transparency.
5. **Scalability and Performance Optimization:**
    - Deployment of side chains or sharding techniques to manage large-scale elections efficiently.
    - Load balancing and distributed computing to handle high traffic volumes during peak election periods.
6. **Decentralized Storage and Security:**
    - Use of InterPlanetary File System (IPFS) or blockchain-based storage to secure voter and election data.
    - Encryption techniques such as AES-256 protect sensitive voter information from unauthorized access.

2.3 Benefits to the Society of Blockchain-Based Voting Systems

Blockchain technology plays a crucial role in enhancing electoral trust by ensuring that every vote cast is immutable and verifiable. Traditional voting systems often suffer from concerns about manipulation, ballot tampering, or human errors, but blockchain eliminates these vulnerabilities through its decentralized and transparent nature.

- **Immutability and Transparency:** Once a vote is recorded on the blockchain, it cannot be altered or deleted, eliminating the risk of post-election tampering. The ledger remains publicly accessible, allowing all stakeholders to verify votes independently.

- **Decentralization and Security:** Unlike traditional voting systems that rely on central authorities or intermediaries, blockchain distributes voting records across a decentralized network. This prevents single points of failure and safeguards the election process from cyberattacks or malicious interference.
- **Cryptographic Verification:** Advanced cryptographic techniques enable voters to verify their votes while maintaining privacy, fostering greater confidence in the integrity of the electoral process.

2.3.1 Increased Accessibility

Blockchain-based voting solutions significantly improve accessibility by providing a seamless and inclusive voting experience for diverse populations, including remote, disabled, and underserved voters.

- **Remote and Mobile Voting:** A blockchain-powered voting system enables individuals to participate in elections from anywhere in the world, eliminating geographical barriers and increasing voter turnout.
- **Support for Differently-Abled Voters:** User-friendly interfaces, voice-assisted navigation, and screen-reader compatibility enhance accessibility for individuals with disabilities, empowering them to exercise their right to vote independently and securely.
- **Inclusivity for Rural and Underserved Communities:** In many regions, physical polling stations are scarce or inaccessible. A decentralized, mobile-compatible voting platform ensures that people in rural and marginalized communities can participate in the democratic process without logistical hurdles.
- **Multi-Language Support:** Language barriers often discourage voter participation. Blockchain-based voting systems can integrate multiple language options, allowing

users to navigate and cast votes in their preferred language, thus ensuring greater inclusivity.

2.3.2 Elimination of Fraud

One of the most significant advantages of blockchain-based voting is its ability to eliminate electoral fraud, ensuring that elections remain fair and trustworthy.

- **Tamper-Proof Voting Records:** Since blockchain operates on a distributed ledger, no single entity can alter votes once they are recorded, eliminating risks of unauthorized vote changes or manipulation.
- **Smart Contract-Enabled Vote Validation:** Automated smart contracts enforce strict vote validation rules, preventing issues such as double voting, voter impersonation, and fake ballots.
- **Zero-Knowledge Proofs for Privacy and Transparency:** Blockchain allows for secure and anonymous voting using zero-knowledge proofs, ensuring that while votes remain private, the overall election process remains transparent and verifiable.
- **Reduced Influence of Malicious Actors:** The decentralized nature of blockchain prevents any central authority or bad actor from controlling or manipulating election results, fostering public trust in democratic institutions.

2.3.3 Faster and More Efficient Results Traditional vote counting and result declaration processes can be time-consuming, often leading to delays and disputes. Blockchain technology streamlines the electoral process, ensuring swift and accurate outcomes.

- **Automated Vote Counting:** Smart contracts instantly tally votes, eliminating the need for manual counting, reducing human errors, and ensuring rapid result announcements.

- **Real-Time Vote Auditing:** Election monitors, candidates, and stakeholders can track vote progression in real-time, minimizing uncertainty and post-election conflicts.
- **Efficient Election Cycles:** By integrating decentralized technology, the entire election process—from voter registration to result declaration—becomes more streamlined, reducing administrative burdens and costs associated with traditional elections.
- **Minimized Post-Election Disputes:** With a transparent and verifiable voting record, blockchain helps mitigate disputes related to vote discrepancies, recounts, or allegations of fraud, ensuring smoother transitions of power.

2.3.4 Strengthening Democratic Institutions

The adoption of blockchain-based voting solutions marks a revolutionary step in reinforcing democracy by addressing critical concerns of security, accessibility, and efficiency.

- **Enhanced Voter Participation:** The ease of access, remote voting capabilities, and improved security encourage greater voter engagement, leading to higher participation rates in elections

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

The decentralized voting system is designed to provide a secure, transparent, and tamper-proof electoral process by leveraging blockchain technology. The system architecture integrates multiple components that work together to ensure efficient voting operations while maintaining voter privacy and election integrity.

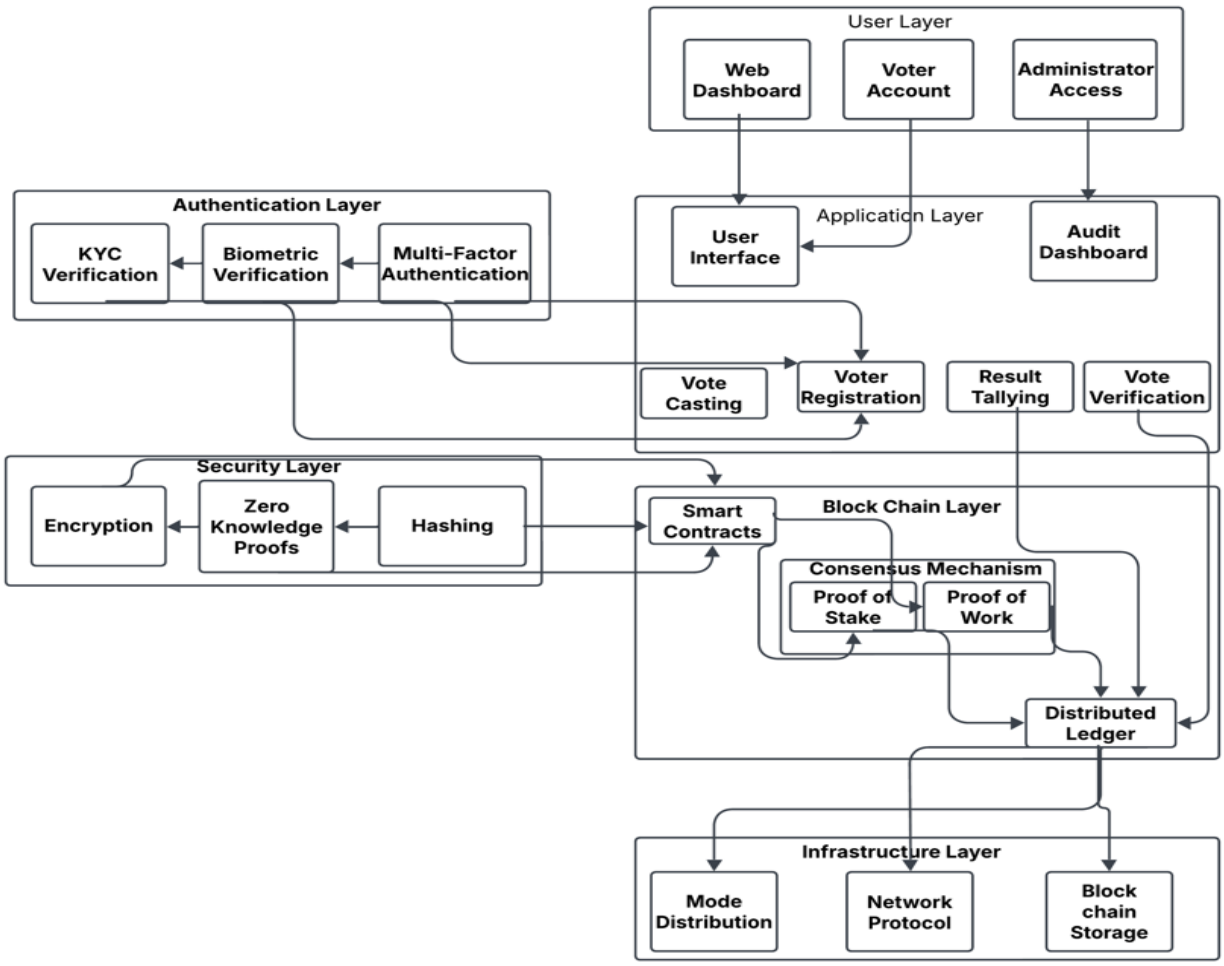

FIG 3.1 System Architecture Diagram

Key Components of the System

1. Blockchain-Based Infrastructure

- The foundation of the system is a blockchain ledger that records votes in an immutable and transparent manner.
- Smart contracts automate vote validation and tallying, eliminating the risk of human errors and unauthorized modifications.
- Consensus mechanisms such as Proof of Stake (PoS) or Byzantine Fault Tolerance (BFT) ensure secure and decentralized vote verification.

2. Secure Voter Authentication

- Multi-factor authentication (MFA) ensures that only eligible voters can participate in the election.
- Voter verification methods include biometric authentication, cryptographic key verification, and digital signatures.
- Decentralized Identity (DID) solutions are implemented to preserve voter anonymity while preventing voter fraud.

3. User-Friendly Voting Interface

- A web-based and mobile-compatible interface allows voters to cast their votes conveniently and securely.
- The interface is designed for accessibility, supporting multiple languages and features for differently-abled users.
- Real-time confirmation of vote submission ensures transparency and voter confidence.

4. Vote Encryption and Privacy Mechanisms

- Zero-knowledge proofs (ZKP) and homomorphic encryption are used to ensure that votes remain private and tamper-proof.
- End-to-end encryption protects voter data from potential cyber threats and unauthorized access.

5. Real-Time Vote Auditing and Transparency

- Every vote is recorded on a public ledger that can be audited by election monitors without compromising voter identity.
- Blockchain explorers allow independent verification of election results, increasing transparency and trust.
- Election observers and regulators can track voting trends without interfering with the electoral process.

6. Scalable and Distributed Network Infrastructure

- The system employs distributed nodes to process and verify transactions efficiently.
- Sharding and sidechains optimize network scalability, enabling the system to handle large-scale elections seamlessly.
- Load balancing ensures smooth operations even during peak voting periods.

7. Secure Data Storage and Backup

- The voting data is stored in decentralized solutions such as the InterPlanetary File System (IPFS) to prevent data loss or tampering.
- Regular data backups and redundancy mechanisms are in place to ensure election continuity even in the event of a system failure.

3.2 UML Diagrams

3.2.1 CLASS DIAGRAM: - Illustrates the structure of the system by showing its classes, attributes, methods, and relationships between them

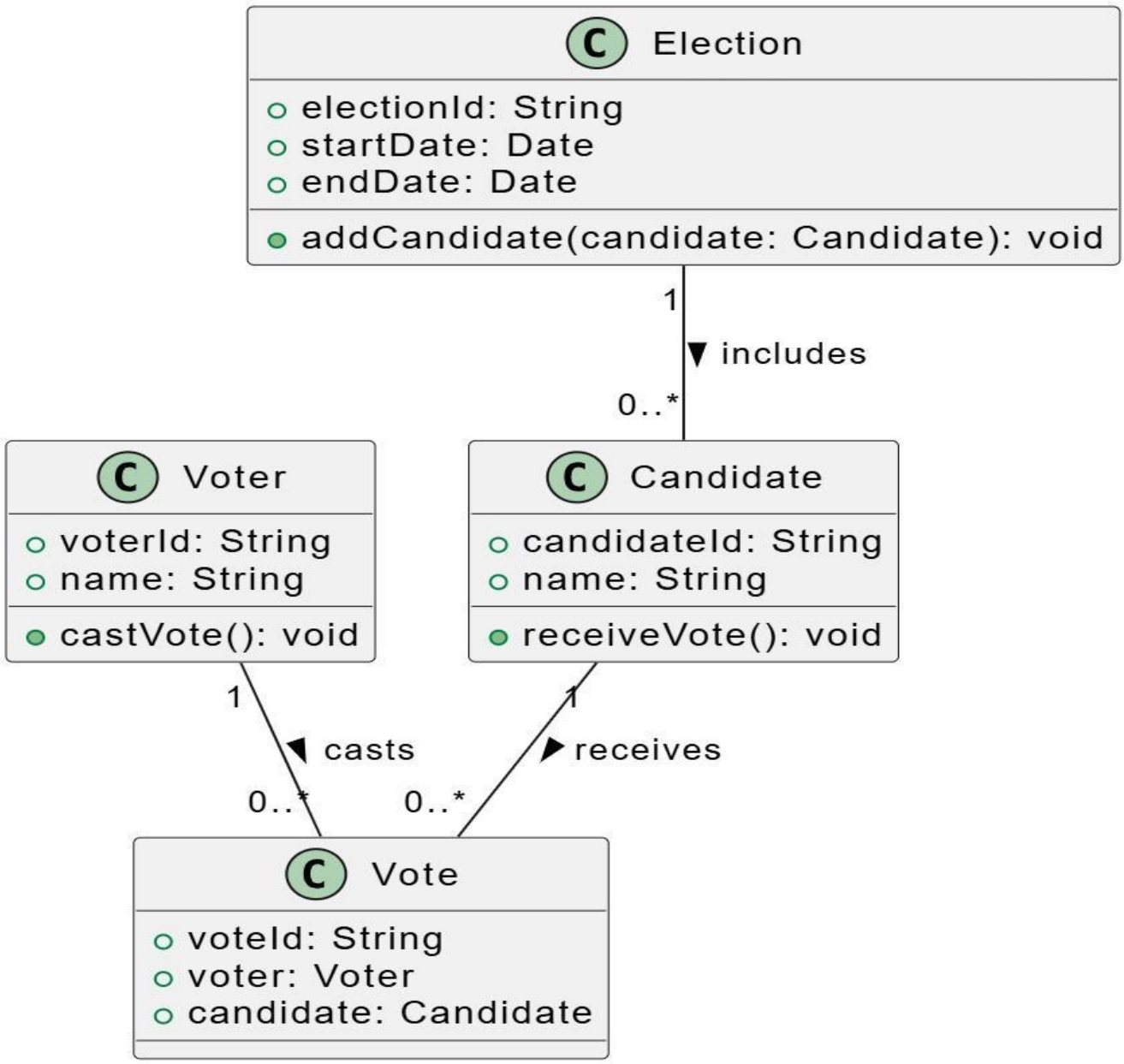

FIG 3.2.1 CLASS DIAGRAM

3.2.2 SEQUENCE DIAGRAM:

Describes the workflow for a specific process, like detection and alert generation, detailing the interaction between various components.

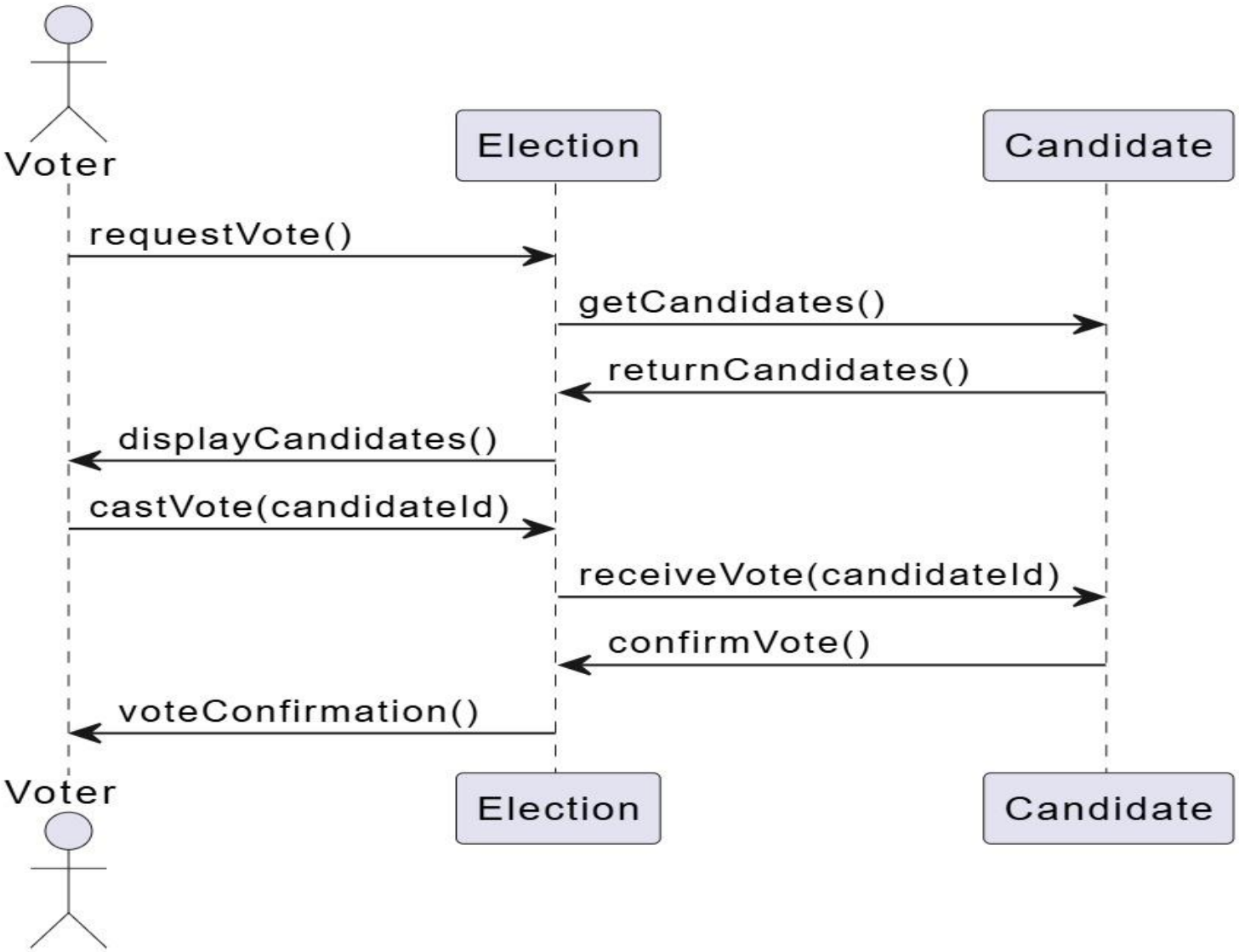

FIG 3.2.2 SEQUENCE DIAGRAM

3.2.3 COMPONENT DIAGRAM

Shows the high-level architecture, including the system's main components and their interactions.

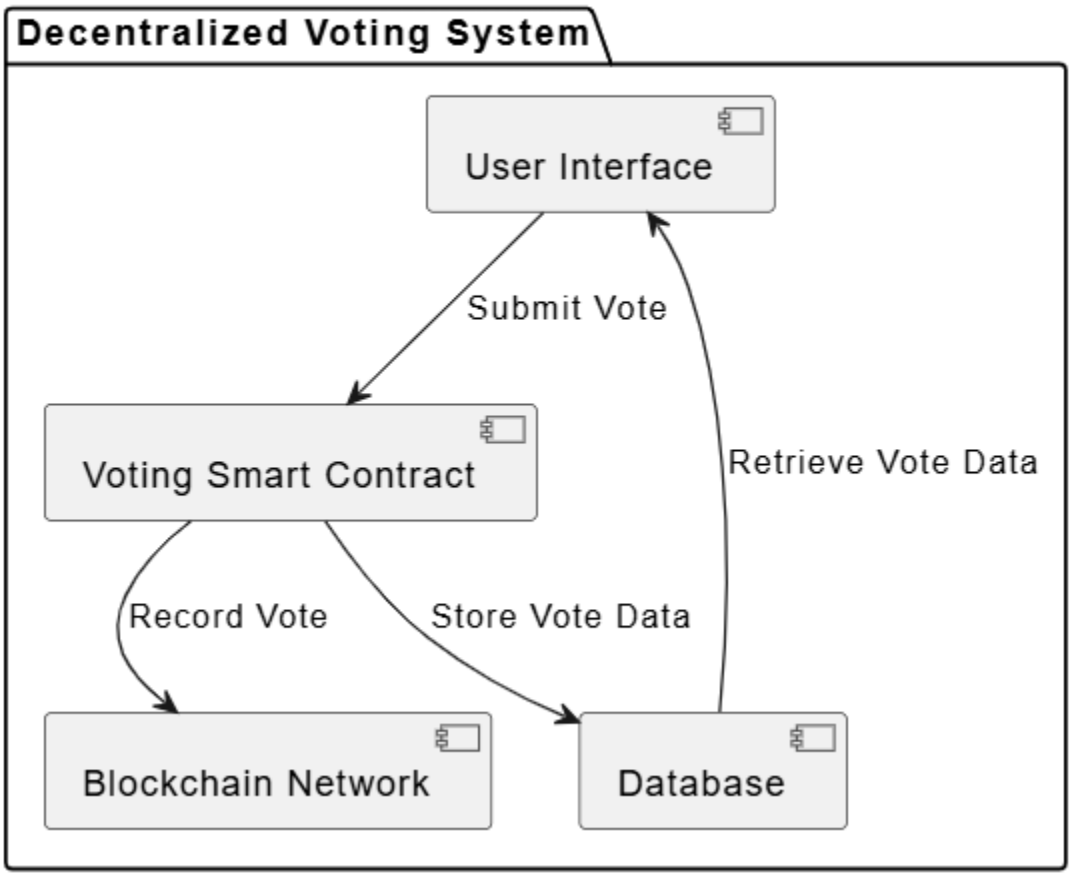

FIG 3.2.3 COMPONENT DIAGRAM

3.2.4 ACTIVITY DIAGRAM

Shows the high-level architecture, including the system's main components and their interactions.

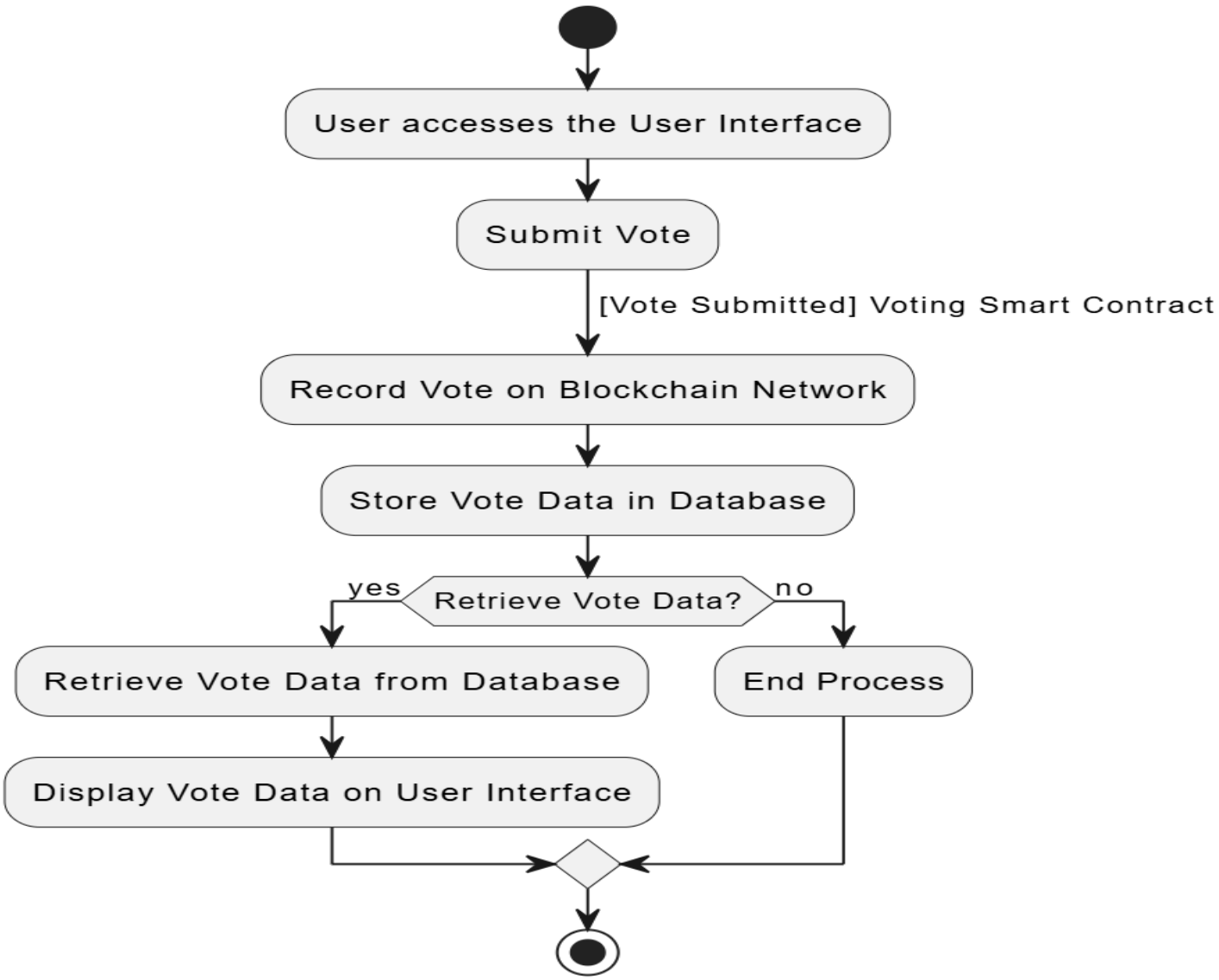

FIG 3.2.4 ACTIVITY DIAGRAM

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULES:

1. User Authentication and Access Control

- Uses cryptographic key-based authentication to ensure only eligible voters participate.
- Ensures decentralized identity verification to protect voter anonymity while preventing fraud.

2. Blockchain Network and Smart Contracts

- Establishes a blockchain network for recording and verifying votes.
- Deploys Ethereum or Hyperledger Fabric smart contracts to automate vote validation and tallying.
- Configures consensus mechanisms like Proof of Stake (PoS) to ensure decentralization and security.

3. Frontend and User Interface Module

- Develop a user-friendly web and mobile voting platform using React.
- Implements an intuitive design for easy vote casting, real-time tracking, and confirmation.
- Provides multilingual support and accessibility features for an inclusive voting experience.

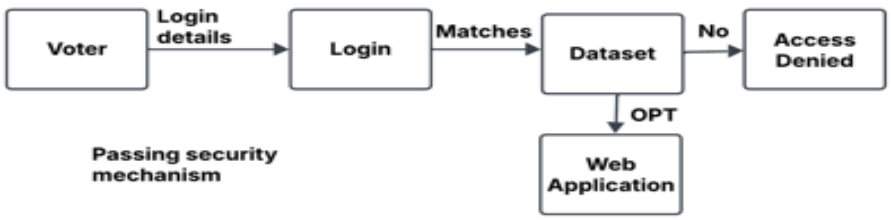

FIG 4.1.3 Passing security mechanic

4. Backend Processing and Database Management

- Uses Flask to manage API requests, validate user inputs, and facilitate blockchain interactions.
- Stores encrypted metadata of votes in PostgreSQL while ensuring immutability in the blockchain.
- Employs Inter Planetary File System (IPFS) for secure and decentralized vote storage.

5. Encryption and Security Module

Implements end-to-end encryption techniques such as AES-256 to protect voter data.

Uses Zero-Knowledge Proofs (ZKP) to ensure anonymous yet verifiable voting.

6. Vote Auditing and Transparency Module

- Provides real-time vote auditing via a public blockchain ledger.
- Allows election observers and regulators to verify election integrity without compromising voter privacy.Implements blockchain explorers for independent vote verification and public transparency.

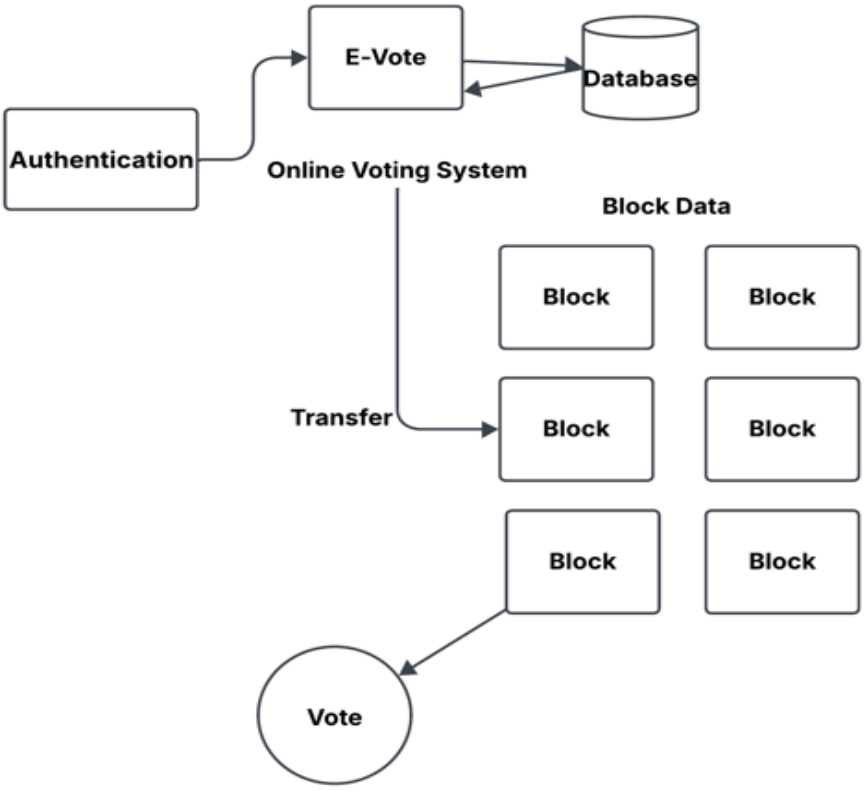

FIG 4.1.5 Encryption

7. Scalability and Performance Optimization Module

- Uses sharding and sidechains to enable the system to handle large-scale elections efficiently.
- Implements load balancing and distributed computing to optimize performance during peak voting periods.
- Ensures that vote processing remains efficient and does not overload network resources.

4.2 DEPLOYMENT:

4.2.1 System Deployment and Setting Up the Blockchain Network

- Deploying the blockchain infrastructure on a secure and scalable network (Ethereum, Hyperledger Fabric, or a private blockchain).
- Configuring smart contracts to handle vote transactions and ensure automated validation.
- Establishing nodes and validators to maintain decentralization and prevent single points of failure.

4.2.2 Server and Database Configuration

- Hosting backend services using cloud platforms such as AWS, Google Cloud, or on-premises secure servers.
- Configuring PostgreSQL for metadata storage and IPFS for decentralized vote storage.
- Implementing security policies for data encryption and access control.

4.2.3. Frontend Deployment

- Deploying the user interface on secure web servers, ensuring accessibility across devices.
- Implementing HTTPS protocols and secure authentication mechanisms.
- Testing frontend responsiveness across different browsers and mobile platforms.

4.2.4. Security Implementation

- Conducting vulnerability assessments and penetration testing to identify security gaps.

- Implementing firewalls, intrusion detection systems (IDS), and anti-DDoS mechanisms.
- Ensuring that cryptographic security layers (AES-256, Zero-Knowledge Proofs) are properly configured.

4.3 Testing and Quality Assurance

4.3.1 Unit Testing

- Verifying individual components such as smart contracts, authentication modules, and transaction handlers.
- Ensuring that each module functions correctly in isolation.

4.3.2. Integration Testing

- Testing interactions between the frontend, backend, blockchain network, and database.
- Ensuring seamless communication between smart contracts and API services.

CHAPTER 5

RESULTS

5.1 Results

The results highlight that the decentralized voting system is a robust tool for modern electoral processes, particularly in systems where vote verification and fraud detection are crucial. The combination of high accuracy and efficient similarity analysis enables real-time validation of votes, ensuring election integrity.

Strengths:

- High accuracy in vote validation and fraud detection.
- Efficient processing times, making it suitable for real-world elections.
- Transparent and verifiable voting records are maintained on the blockchain.

Limitations:

- Slightly lower accuracy in low-turnout elections due to variability in voting patterns.
- Dependence on network performance, as blockchain processing times can be affected by transaction congestion.

Future Enhancements:

- Expanding the dataset to include diverse election formats and voting scenarios.
- Enhancing AI models to detect voting anomalies with greater accuracy.
- Integrating real-time analytics dashboards for better election monitoring. These improvements will further strengthen the system’s reliability and effectiveness in decentralized voting environments.

CHAPTER 6

CONCLUSION

The decentralized voting system transforms electoral processes by incorporating blockchain technology, smart contracts, and cryptographic security. This guarantees transparency, prevents tampering, and eliminates the risks of election fraud and vote manipulation. By decentralizing control, the system diminishes the influence of a single entity, fostering public trust in democratic institutions. A key advantage of this system is real-time vote auditing and automated tallying, which improve efficiency and accuracy. Unlike traditional methods that involve manual counting and delays, blockchain-based voting delivers immediate, verifiable results. Advanced security measures such as multi-factor authentication and decentralized identity frameworks further safeguard voter privacy and deter fraud, ensuring a secure and reliable process. Beyond security, this system enhances voter accessibility, especially for individuals in remote areas, while significantly lowering election costs. Despite its advantages, challenges such as scalability, regulatory compliance, and public acceptance persist. Continued research, pilot testing, and collaboration with policymakers are vital for successful implementation. With ongoing advancements, decentralized voting can provide a secure, inclusive, and fraud-resistant electoral system.

CHAPTER 7

APPENDIX I- CODE

Election.sol

```
pragma solidity ^0.8.0;
contract Election {
    address public manager;
    struct Candidate {
        uint id;
        string CfirstName;
        string ClastName;
        string CidNumber;
        uint voteCount;
    }
    mapping (address => bool) public voters;
    mapping (uint => Candidate) public candidates;
    uint public candidatesCount;
    event votedEvent (
        uint indexed candidateId
    );
    constructor () {
        manager = msg.sender;
    }
    function addCandidate (string memory _CfirstName, string memory _ClastName, string memory _CidNumber)
public restricted {
        candidatesCount++;
        candidates[candidatesCount] = Candidate(candidatesCount, _CfirstName, _ClastName, _CidNumber, 0);
    }
    modifier restricted () {
        require(msg.sender == manager, "Access denied. Only manager can perform this action.");
        _;
    }
    function vote (uint _candidateId) public {
        require(!voters[msg.sender], "You have already voted.");
```

```
        require(_candidateId > 0 && _candidateId <= candidatesCount, "Invalid candidate ID.");
        voters[msg.sender] = true;
        candidates[_candidateId].voteCount++;
        uint candidateId = _candidateId;
        emit votedEvent(_candidateId);
    }
    // Users
    // Register
    struct User {
        string firstName;
        string lastName;
        string idNumber;
        string email;
        string password;
        address add;
    }
    mapping (uint => User) public users;
    uint public usersCount;
    function addUser (string memory _firstName, string memory _lastName, string memory _idNumber, string
memory _email, string memory _password) public {
        usersCount++;
        users[usersCount] = User(_firstName, _lastName, _idNumber, _email, _password, msg.sender);
    }
}
```

Migrations.sol

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

contract Migrations {
    address public owner;
    uint public last_completed_migration;
    modifier restricted() {
        if (msg.sender == owner) _;
    }
```

```
    constructor() {
        owner = msg.sender;
    }
    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }
    function upgrade(address new_address) public restricted {
        Migrations upgraded = Migrations(new_address);
        upgraded.setCompleted(last_completed_migration);
    }
}
```

Truffleconfig.js

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,
      network_id: "*" // Match any network id
    }
  },
  compilers: {
    solc: {
      version: "^0.8.0" // Use a specific version of Solidity
    }
  }
};
```

App.js

```
App = {
    web3Provider: null,
    contracts: {},
    account: '0x0',
    hasVoted: false,
    votedForID: 0,
    finishElection: 0,
    mins: 0,

    // web3 connects our client side application to the blockchain.
    // metamask gives us an instance of web3 that we will use to connect to the blockchain
    // if this doesn't happen we will set a default web3 provider from our local blockchain instance 'localhost 7545'
    init: function () {
        return App.initWeb3();
    },
```

```
  initWeb3: async function () {
    // Modern dapp browsers...
    if (window.ethereum) {
      App.web3Provider = window.ethereum;
      try {
        // Request account access
        await window.ethereum.enable();
      } catch (error) {
        // User denied account access...
        console.error("User denied account access")
      }
    }
// Legacy dapp browsers...
    else if (window.web3) {
      App.web3Provider = window.web3.currentProvider;
    }
// If no injected web3 instance is detected, fall back to Ganache
    else {
      App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
    }
    web3 = new Web3(App.web3Provider);
    web3.eth.defaultAccount=web3.eth.accounts[0]
    return App.initContract();
  },


  // we then initailze our contract
  // this function loads up our contract to our front end application
  initContract: function () {
    $.getJSON("Election.json", function (election) {
      // Instantiate a new truffle contract from the artifact
      App.contracts.Election = TruffleContract(election);
      // Connect provider to interact with contract
      App.contracts.Election.setProvider(App.web3Provider);

      App.listenForEvents();

      return App.render();
    });
  },


  // Listen for events emitted from the contract
  listenForEvents: function () {
    App.contracts.Election.deployed().then(function (instance) {
      // Restart Chrome if you are unable to receive this event
      // This is a known issue with Metamask
      // https://github.com/MetaMask/metamask-extension/issues/2393
      instance.votedEvent({}, {
        fromBlock: 0,
        toBlock: 'latest'
      }).watch(function (error, event) {
        console.log("event triggered", event)
        // Reload when a new vote is recorded
        // App.render();
      });
```

```
    });
  },


  // render function which is what will layout all the content on the page
  render: function () {
    var electionInstance;
    var loader = $("#loader");
    var content = $("#content");
    var register = $("#register");

    loader.show();
    content.hide();

    // Load account data
    web3.eth.getCoinbase(function (err, account) {
      if (err === null) {
        App.account = account;
        $("#accountAddress").html("Your Account: " + account);
      }
    });


    App.contracts.Election.deployed().then(function(instance) {
      electionInstance = instance;
      // document.querySelector('.buy-tickets').style.display = 'none';

      return electionInstance.manager();
    }).then(function (manager) {
      if (manager !== App.account){
        document.querySelector('.buy-tickets').style.display = 'none';
      }

     return electionInstance.candidatesCount();
    }).then(function(candidatesCount) {
     var candidatesResults = $("#candidatesResults");
     candidatesResults.empty();

     var candidatesSelect = $('#candidatesSelect');
     candidatesSelect.empty();

     for (var i = 1; i <= candidatesCount; i++) {
      electionInstance.candidates(i).then(function(candidate) {
       var id = candidate[0];
       var fname = candidate[1];
       var lname = candidate[2];
       var idNumber = candidate[3];
       var voteCount = candidate[4];

       // Render candidate Result
       var candidateTemplate = "<tr><th>" + id + "</th><td>" + fname+ " " + lname + "</td><td>" +
idNumber  + "</td><td>" + voteCount + "</td></tr>"
       candidatesResults.append(candidateTemplate);

       // Render candidate ballot option
```

```
            if (lidNumber === idNumber) {
                if(lpassword === password)
                {
                    location.href='results.html';
                }
                else {
                    prompt("Incorrect login details, Please try again");
                }
                break;
            }
        }
    },

    startElection: function () {
        localStorage.setItem("finishElection", "0");
        location.href='index.html';
    },
    endElection: function () {
        localStorage.setItem("finishElection", "1");
        location.href='results.html';
    }
};
$(function () {
    $(window).load(function () {
        App.init();
    });
});
```

CHAPTER 8
APPENDIX II- SCREENSHOTS

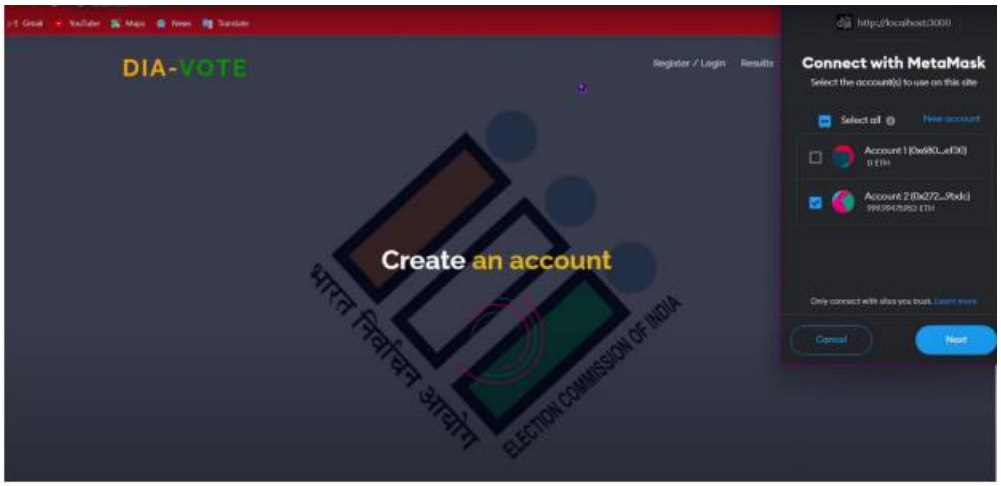

Fig 01 Wallet Linking Page

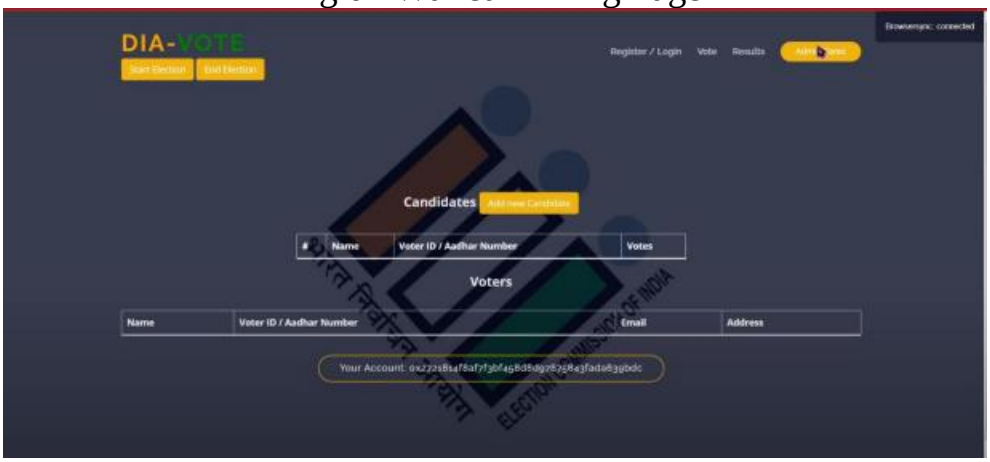

Fig 02 Home Page

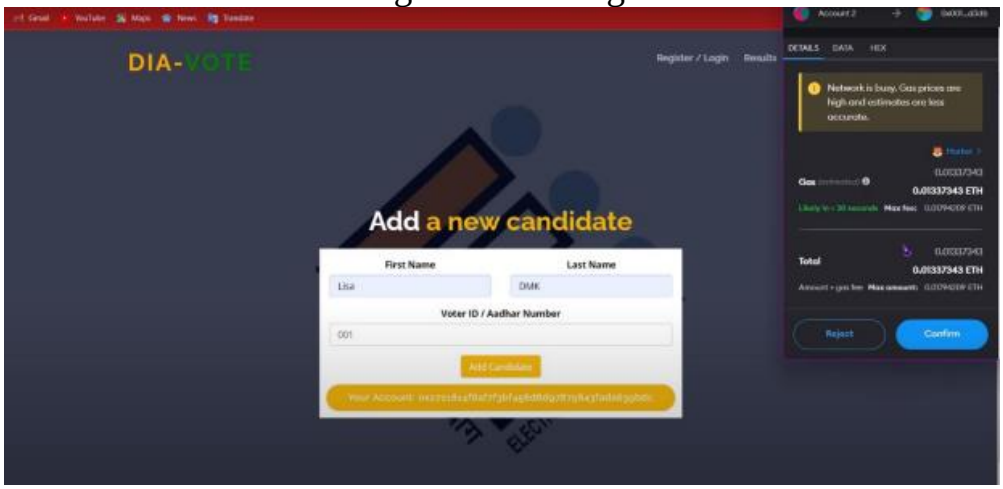

Fig 03 Add new candidate Page

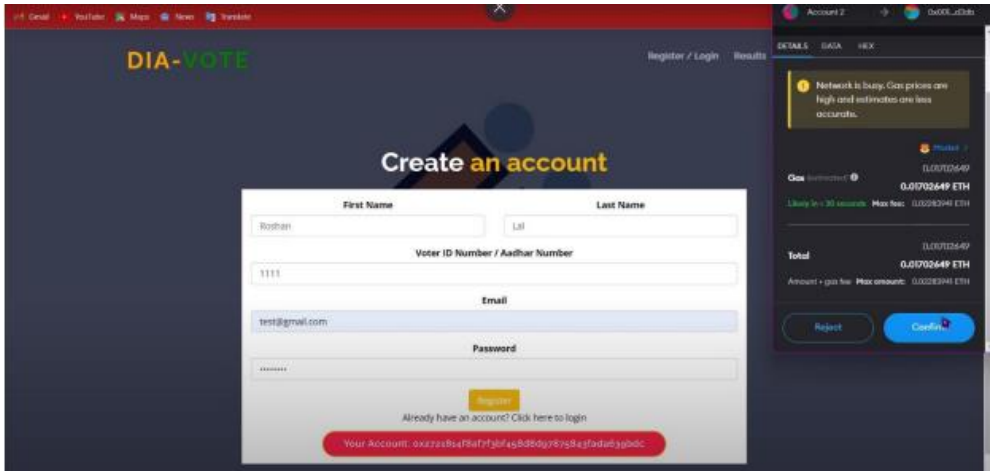

Fig 04 Voter Registration Page

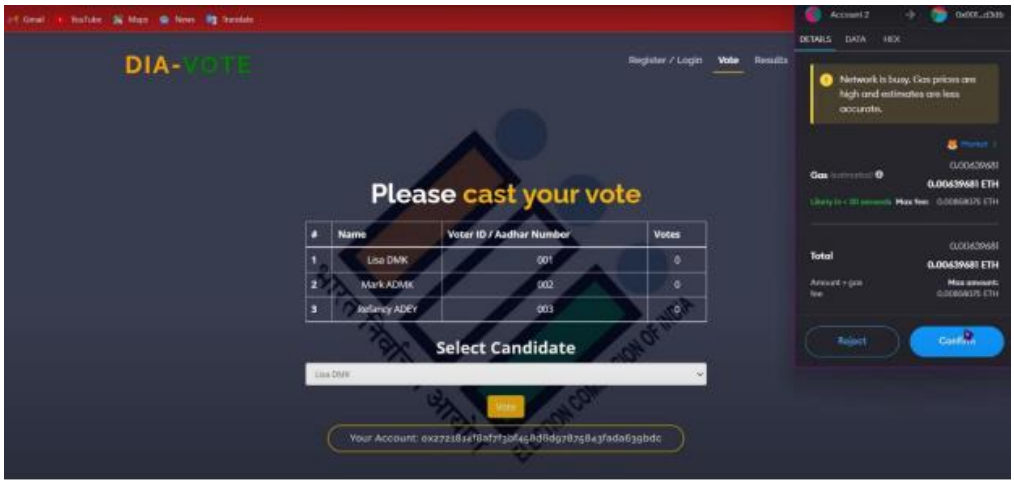

Fig 05 Vote Casting Page

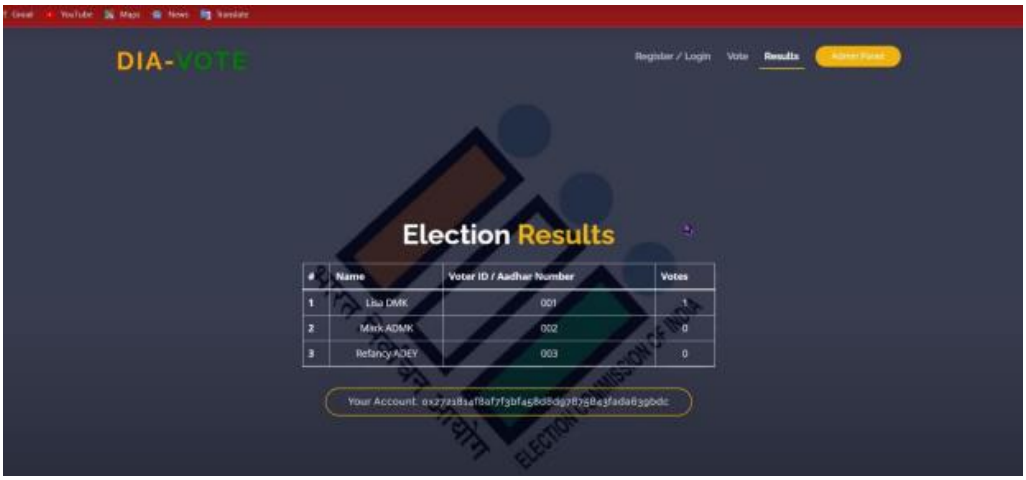

Fig 06 Results Page

REFERENCES

[1] Sos.ca.gov. (2007). Top-to-Bottom Review | California Secretary of State. Available at: http://www.sos.ca.gov/elections/voting-systems/oversight/ top-bottom-review/.

[2] Nicholas Weaver. (2016). Secure the Vote Today Available at:https:// www.lawfareblog.com/secure-vote-today.

[3] TechCrunch, (2018). Liquid democracy uses blockchain to fix politics, and now you can vote for it. Available at: https://techcrunch.com/2018/ 02/24/liquid-democracy-uses blockchain.

[4] Ajit Kulkarni, (2018), "How To Choose Between Public And Permissioned Blockchain For Your Project", Chronicled, 2018.

[5]Zhang S, Wang L, Xiong H. Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. Int J Inf Secur. 2020;19:323–41.

[6]Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE International Congress on big data (BigData congress); 2017 (pp. 557–564). IEEE.

[7]Hyvärinen H, Risius M, Friis G. A blockchain-based approach towards overcoming financial fraud in public sector services. Bus Inform Syst Eng. 2017;59:441–56.

[8]Tanwar S, Gupta N, Kumar P, Hu YC. Implementation of blockchain-based e-voting system. Multimedia Tools Appl. 2024;83(1):1449–80.

[9] Stach C, Gritti C, Przytarski D, Mitschang B. Assessment and treatment of privacy issues in blockchain systems.

[10] Gautam Srivastava1, Ashutosh Dhar Dwivedi2 and Rajani Singh2(2018); Crypto-democracy: A Decentralized Voting Scheme using Blockchain Technology.

[11]Friðrik Þ.Hjálmarsson,Gunnlaugur K.Hreiðarsson(2018);Blockchain-Based E-Voting System.

[12] Nir Kshetri and Jeffrey Vote (2018); Blockchain-Enabled E-Voting; www.computer.org/software. AUTHORS

[13] Umut Can Çabuk1, Eylül Adıgüzel2, Enis Karaarslan2(2018); A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems; International Journal of Advanced Research in Computer and Communication Engineering.

[14] Madise, Ü. & Martens, T. (2006). E-voting in Estonia 2005. The first practice of countrywide binding Internet voting in the world. Electronic Voting, 86.

[15] S. Raval, "Decentralized Applications: Harnessing Bitcoin's Blockchain Technology." O'Reilly Media, Inc. Sebastopol, California (2016).

[16] Jason Paul Cruz1, a) Yuichi Kaji2,b)(2017); E-voting System Based on the Bitcoin Protocol and Blind Signatures; IPSJ Transactions on Mathematical Modeling and Its Applications Vol.10 No.1 14–22.