Decentralized Voting system using Blockchain

A dissertation submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted
by

Kodam Akhiranandha (21B81A05D2)

Deepak Reddy Bondugula (21B81A05E0)

Dinesh Reddy Valipi (21B81A05E1)

Under the Guidance of

Ms G.Sushma

Sr. Asst. Professor

Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.) - 501510, Telangana State

2024-2025

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,
Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"Decentralized Voting system using Blockchain"** is being submitted by **Kodam Akhiranandha (21B81A05D2), Deepak Reddy Bondugula (21B81A05E0), and Dinesh Reddy Valipi (21B81A05E1)** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering,** during the academic year 2024-2025.

Project Guide
Ms. G. Sushma
Sr. Asst. Professor

Professor-in-charge
Dr. V. Dattatreya

External Examiner

Professor and Head, CSE
Dr. A. Vani Vasthala

DECLARATION

We hereby declare that this Project report titled "**Decentralized Voting system using Blockchain**" submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by us under the guidance of **Ms.G.Sushma**. The information and data given in the report is authentic to the best of our knowledge. This project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

Kodam Akhiranandha (21B81A05D2)

Deepak Reddy Bondugula (21B81A05E0)

Dinesh Reddy Valipi (21B81A05E1)

Date: 27-03-2025

Place: Ibrahimpatnam, Hyderabad

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Decentralized voting using Ethereum blockchain is a secure, transparent, and tamper-proof way of conducting online voting. It is a decentralized application built on the Ethereum blockchain network, which allows participants to cast their votes and view the voting results without the need for intermediaries. In this system, votes are recorded on the blockchain, making it impossible for anyone to manipulate or alter the results. The use of smart contracts ensures that the voting process is automated, transparent, and secure. The use of the blockchain technology and the implementation of a decentralized system provide a reliable and cost-effective solution for conducting trustworthy and fair elections.

CHAPTER 1
INTRODUCTION

Elections are a crucial part of democracy, yet traditional voting systems often face challenges such as fraud, lack of transparency, and security vulnerabilities. To address these issues, this project introduces a blockchain-based voting system that ensures security, transparency, and immutability in the voting process. By integrating blockchain technology with an Electronic Voting Machine (EVM) interface, the system leverages Ethereum blockchain, smart contracts, and cryptographic hashing to securely record and verify votes, preventing tampering or manipulation. The decentralized nature of blockchain eliminates single points of failure, ensuring a more reliable voting mechanism. Additionally, the system features an intuitive React and Bootstrap-based user interface, making it accessible and easy to use. Automated vote counting through smart contracts ensures real-time results with accuracy, reducing human errors and delays. This project aims to revolutionize electoral processes by providing a secure, transparent, and efficient alternative to traditional voting methods, enhancing trust and integrity in democratic elections.

1.1 Motivation

The motivation behind this project stems from the growing concerns over election security, vote tampering, and lack of transparency in traditional voting systems. Blockchain technology offers a decentralized, immutable, and transparent solution to these challenges, ensuring trust and fairness in the electoral process. By integrating smart contracts for automated vote counting and verification, this system eliminates human errors and delays. This project aims to modernize elections by leveraging blockchain to create a secure and trustworthy voting mechanism.

1.2 Problem Statement

Traditional voting systems face challenges such as vote tampering, lack of transparency, security vulnerabilities, and inefficiencies in result processing. These issues can lead to election fraud, distrust among voters, and delays in announcing results. This project aims to develop a blockchain-based voting system that ensures security, transparency, and efficiency in the electoral process.

1.3 Objective

The objective of this project is to develop **a** secure, transparent, and efficient voting system using blockchain technology. By leveraging Ethereum blockchain and smart contracts**,** the system ensures that votes are immutable, tamper-proof, and verifiable. It aims to eliminate election fraud, enhance voter trust, and provide a decentralized platform for conducting elections. The system also focuses on automating vote counting, reducing human errors, and ensuring real-time result processing. Overall, this project seeks to revolutionize traditional voting methods by integrating cutting-edge blockchain security into the electoral process.

CHAPTER 2

LITERATURE REVIEW

2.1 A Survey of Blockchain Based on E-voting Systems

Authors: Yousif Osman Abuidris, Rajesh Kumar and Wang Wenyong

ABSTRACT: Blockchain technology as a decentralized and distributed public ledger in a P2P network has recently gained much attention. In this technology, a linked block structure is applied, and a trusted consensus mechanism is established to synchronize data modifications, making it possible to develop a tamper-proof digital platform for data storage and sharing. We think that blockchain could be used in various interactive online systems, such as the Internet of Things, supply chain systems, voting systems, etc. The scope of this survey is to shed light on some recent contributions of the security and privacy issues associated with e-voting based on blockchain. At the end of this paper, we provided a comparison for the security and privacy requirements of the existing e-voting systems based on blockchain.

2.2 A Framework to Make Voting System Transparent Using Blockchain Technology

Authors: Muhammad Shoaib Farooq, Usman Iftikhar and Adel Khelifi

ABSTRACT: Traditional and digital voting systems face widespread mistrust, causing fundamental rights violations and a lack of transparency. Blockchain technology can help address these issues and ensure fair elections. The article presents a blockchain-based platform that provides transparency and reliability, allowing voters to conduct voting without physical polling stations. The framework supports a scalable blockchain using flexible consensus algorithms and Chain Security Algorithm. Smart contracts provide secure connections between users and the network. The system also includes cryptographic hash encryption and 51% attack prevention. Performance evaluations show the system can be implemented on a large-scale population.

2.3 A Survey on Smart Electronic Voting System Using Blockchain Technology

Authors: Naina Nagesh Dhepe and Dr. Pathan Mohd Shafi

ABSTRACT: India is the world's largest democracy with over one billion in population; India has over 668 million voters and 543 parliamentary constituencies. India is the largest democracy in the world. Voting is the bridge of government and governance. In recent years, the technology used in the voting process has been given a renewed focus. There are many security problems in the current voting system, and even simple security features are difficult to prove. There are many concerns about a voting system that can be proven right. There are some reasons why an electronic system is being used by the government to increase elections and reduce electoral expenses. There is still some scope for electronic voting systems, because there is no way to identify whether or not the user is authentic and to secure electronic voting machines from misconceptions by the electronic voting scheme. In order to increase safety and transparency between the users, the proposed system will develop a compatible high security voting machine with the help of Block-chain technology.

2.4 Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal

Authors: Matthew Sharp, Laurent Njilla and Chin-Tser Huang

ABSTRACT: Advancements in blockchain technology and network technology are bringing in a new era in electronic voting systems. These systems are characterized by enhanced security, efficiency, and accessibility. In this paper, we compose a comparative analysis of blockchain-based electronic voting (e-voting) systems using blockchain technology, cryptographic techniques, counting methods, and security requirements. The core of the analysis involves a detailed examination of blockchain-based electronic voting systems, focusing on the variations in architecture, cryptographic techniques, vote counting methods, and security. We also introduce a novel blockchain-based e-voting system, which integrates advanced methodologies, including the Borda count and Condorcet method, into e-voting systems for improved accuracy and representation in vote tallying. The system's design features a flexible and amendable blockchain structure, ensuring robustness and security. Practical implementation on a Raspberry Pi 3 Model B+ demonstrates the system's feasibility and adaptability in diverse environments. Our study of the evolution of e-voting systems and the incorporation of blockchain technology contributes to the development of secure, transparent, and efficient solutions for modern democratic governance.

2.5 Survey on Voting System using Blockchain Technology

Authors: Mayur Shirsath, Mohit Zade, Riteshkumar Talke, Praful Wake and Maya Shelke.

ABSTRACT: E-voting, a symbol of modern democracy, is being explored for its potential to improve voting processes. Despite its potential, research has primarily focused on technical and legal issues, neglecting the practicality of e-voting. The proposed internet-based online voting system, supported by blockchain technology, aims to address issues like distance and improper voting tools. The system uses encryption and hashing techniques to secure voting, storing transactions on a private blockchain. This innovative solution, which does not require technical skills, is expected to increase voter turnout in online voting.

2.6 Existing System

The current voting system, which has been in place for decades, operates through both traditional paper-based methods and electronic alternatives. While these methods have served as the foundation of democratic elections worldwide, they come with significant challenges that hinder their efficiency, transparency, and security. In this discussion, we explore the key aspects of existing voting systems, their limitations, and the critical need for modernization.

Traditional Paper-Based Voting

Paper ballots have been the cornerstone of electoral processes for many countries. Voters physically visit polling stations, where they are handed paper ballots that they mark and deposit into a secure ballot box. After voting ends, these ballots are manually counted and recorded to determine election results. While this method ensures a tangible voting record, it presents several problems, including long wait times, susceptibility to ballot tampering, human counting errors, and logistical difficulties in securing and transporting ballots.

Electronic Voting Machines (EVMs)

In response to the inefficiencies of paper ballots, many nations have introduced Electronic Voting Machines (EVMs). These machines allow voters to select their preferred candidates digitally, reducing the need for manual counting and aiming to streamline the voting process. However, despite their technological advancements, EVMs have faced criticism due to their vulnerability to hacking, programming errors, and a lack of voter-verifiable audit trails in some systems. Additionally, concerns have been raised about the proprietary nature of EVM software, limiting public and expert scrutiny of their security.

Online Voting Systems

Online voting, or internet-based voting, has been proposed as a modern solution to increase voter participation and accessibility. In theory, it allows voters to cast their ballots remotely using secure credentials. However, this method faces significant security risks, including the possibility of cyberattacks, malware, and voter impersonation. Furthermore, digital literacy and access to stable internet connections pose barriers to equitable participation.

2.7 Limitations of Existing Voting Systems

1. **Lack of Transparency:** In both paper-based and electronic systems, verifying whether a vote has been counted accurately remains a challenge. Many election processes lack an effective mechanism for voters to track their votes, creating distrust in election outcomes.

2. **Vulnerability to Fraud and Manipulation:** Traditional voting methods are prone to various types of election fraud, including ballot stuffing, vote buying, and tampering with physical or electronic votes. Even electronic systems are not immune, as they can be hacked, leading to altered vote counts or system failures.

3. **Slow Vote Counting and Result Compilation:** In many cases, manually counting paper ballots is a labor-intensive process, causing delays in announcing results. Even with electronic voting, technical malfunctions can slow down the process, leading to disputes and allegations of manipulation.

4. **High Costs and Resource Demands:** Conducting an election involves significant expenditures on ballot printing, electronic voting machines, polling station setup, security personnel, and election monitoring. The financial burden can be overwhelming, particularly for developing nations.

5. **Centralized Control and Authority Risks:** Most traditional voting systems operate under centralized authorities, increasing the risk of electoral manipulation. If the process is managed by a small number of officials without adequate oversight, concerns arise regarding the integrity of the election.

6. **Limited Accessibility and Voter Disenfranchisement:** Physical polling stations often require voters to travel, which can be challenging for elderly individuals, persons with disabilities, and those living in remote areas. Additionally, certain voter ID requirements and bureaucratic registration processes can disenfranchise eligible voters.

7. **System Reliability and Technical Failures:** Electronic voting machines and online platforms are not immune to malfunctions. Software bugs, power outages, and server crashes can disrupt the voting process, delaying elections or invalidating votes.
8. **Legal and Regulatory Complexities:** Many jurisdictions struggle to develop comprehensive legal frameworks for integrating new voting technologies. Regulations may not adequately address emerging threats such as cybersecurity risks and digital voter authentication.

While traditional voting systems have facilitated democratic participation for generations, their numerous challenges highlight the urgent need for innovation. The ideal voting system should be secure, transparent, efficient, and accessible to all eligible voters. Emerging technologies, such as blockchain-based voting, offer promising solutions to many of these issues, but careful implementation and rigorous security measures are necessary to ensure their effectiveness. As election systems continue to evolve, governments must prioritize voter trust, security, and inclusivity to uphold the fundamental principles of democracy.

CHAPTER 3

REQUIREMENTS

3.1 Software Requirements

To develop and run the decentralized voting system, the following software tools and frameworks are required:

Backend Development:

- **Node.js** – JavaScript runtime for executing backend logic and handling API requests.
- **Express.js** – Web framework for building RESTful APIs to facilitate communication between the frontend and backend.
- **MongoDB Atlas** – Cloud-based database to store user and election-related information (Port: 5000).

Blockchain Development:

- **Solidity** – Smart contract programming language for writing secure voting contracts.
- **Truffle** – Development framework for compiling, testing, and deploying smart contracts.
- **Ganache** – Local Ethereum blockchain for testing smart contracts before deployment.
- **Web3.js** – JavaScript library for interacting with the Ethereum blockchain.
- **MetaMask** – Browser extension for managing Ethereum wallets and connecting to blockchain applications.

Frontend Development:

- **React.js** – JavaScript library for building the user interface of the voting system.

Additional Tools:

- **Postman** – API testing tool for validating backend services.

Git/GitHub – Version control system for tracking changes and collaborating on the project.

3.2 Hardware Requirements

For optimal performance and reliability, the system requires the following hardware components:

Minimum Hardware Requirements:

- **Processor** – 2 GHz or higher (Multi-core recommended for better performance).
- **RAM** – 4 GB (Sufficient for development, but 8 GB is recommended for smoother operation).
- **Storage** – Minimum **100 GB** of available disk space to accommodate system files, smart contracts, and database storage.
- **Internet Connectivity** – A **stable internet connection** is required to interact with the blockchain network and ensure real-time transactions.

Recommended Hardware for Efficient Performance:

- **Processor** – Intel i5/i7 or AMD Ryzen 5/7 (Multi-threading support for faster execution).
- **RAM** – 8 GB or more (Improves response time and allows seamless multitasking).
- **Storage** – SSD with at least 256 GB (Faster read/write speeds for database interactions).

3.3 User Requirements

- **Basic Computer Skills:** Users should be familiar with web-based applications.
- **MetaMask Wallet:** Voters must have a MetaMask wallet to interact with the blockchain.
- **Authentication Credentials:** Users should have valid login credentials for voter verification.

CHAPTER 4

SYSTEM DESIGN

PROPOSED SYSTEM

The proposed decentralized voting system utilizing the Ethereum blockchain presents an innovative solution to the challenges faced by traditional and electronic voting systems. By harnessing the power of blockchain technology, this system ensures that the election process is not only secure and transparent but also resistant to fraud and manipulation. Unlike conventional voting systems that rely on centralized authorities to manage and validate election results, the proposed system eliminates intermediaries by using smart contracts to facilitate, record, and tally votes in a fully automated and verifiable manner.

This blockchain-based voting system guarantees the integrity of election data by making each vote immutable and publicly verifiable while maintaining the anonymity of the voter. Every transaction within the system is permanently stored on a distributed ledger, making it impossible for any single entity to alter or tamper with the results. The decentralized nature of the Ethereum network ensures that the system remains operational and resistant to cyber threats, providing a trustworthy platform for democratic elections.

Advantages of the Proposed System

1. **Decentralization** – Unlike traditional voting systems that are controlled by a central authority, the proposed system distributes control across multiple nodes in the Ethereum network. This ensures that no single entity has the power to manipulate election results.
2. **Transparency** – Every vote cast is recorded on the blockchain, allowing all participants to verify the election results in real-time. This enhances public confidence in the electoral process.
3. **Immutability** – Once a vote is recorded on the blockchain, it cannot be altered or deleted. This eliminates the risk of vote tampering and fraudulent activities.

4. **Remote Voting Capability** – The system allows eligible voters to cast their ballots from anywhere in the world, increasing accessibility and voter participation.
5. **Cost Efficiency** – By eliminating the need for physical polling stations, paper ballots, and manual vote counting, the proposed system significantly reduces the cost of conducting elections.
6. **Real-Time Results** – Since votes are recorded and counted instantly on the blockchain, election results can be generated in real-time, minimizing delays and disputes.
7. **Enhanced Security** – The cryptographic security mechanisms inherent in blockchain technology prevent unauthorized access, ensuring that votes remain confidential and protected against cyber threats.

Objectives of the Proposed Research

1. **Enhanced Security** – One of the primary goals of this system is to establish an election platform that guarantees the security of voter data and ensures that votes remain untampered. By leveraging blockchain's cryptographic security, the system will prevent unauthorized access and manipulation.
2. **Ensuring Transparency** – The system aims to provide complete transparency to voters by allowing them to independently verify their votes while maintaining voter anonymity. This will help eliminate suspicions of election fraud and encourage trust in the system.
3. **Improving Accessibility** – By allowing remote voting, the system aims to increase voter participation, particularly among individuals who may face difficulties reaching physical polling stations due to location, disability, or other constraints.
4. **Strengthening Trust** – Trust in the electoral process is fundamental to democracy. This system seeks to enhance voter confidence by offering an immutable, tamper-proof voting mechanism that guarantees accurate results.
5. **Scalability and Efficiency** – The research also aims to develop a scalable system capable of handling elections at various levels, from small-scale

organizational polls to national elections, without compromising performance or security.

6. **User-Friendly Interface** – The proposed system will be designed to be user-friendly and accessible even to individuals with minimal technical knowledge, ensuring that every eligible voter can participate without difficulty.
7. **Auditability and Verifiability** – The system will ensure that election data can be audited by independent observers while preserving voter privacy, making post-election verification simple and reliable.

By implementing these objectives, the proposed decentralized voting system has the potential to modernize the electoral process, eliminating many of the risks associated with traditional voting methods while promoting a more democratic and accessible system for all voters.

4.1 System Architecture

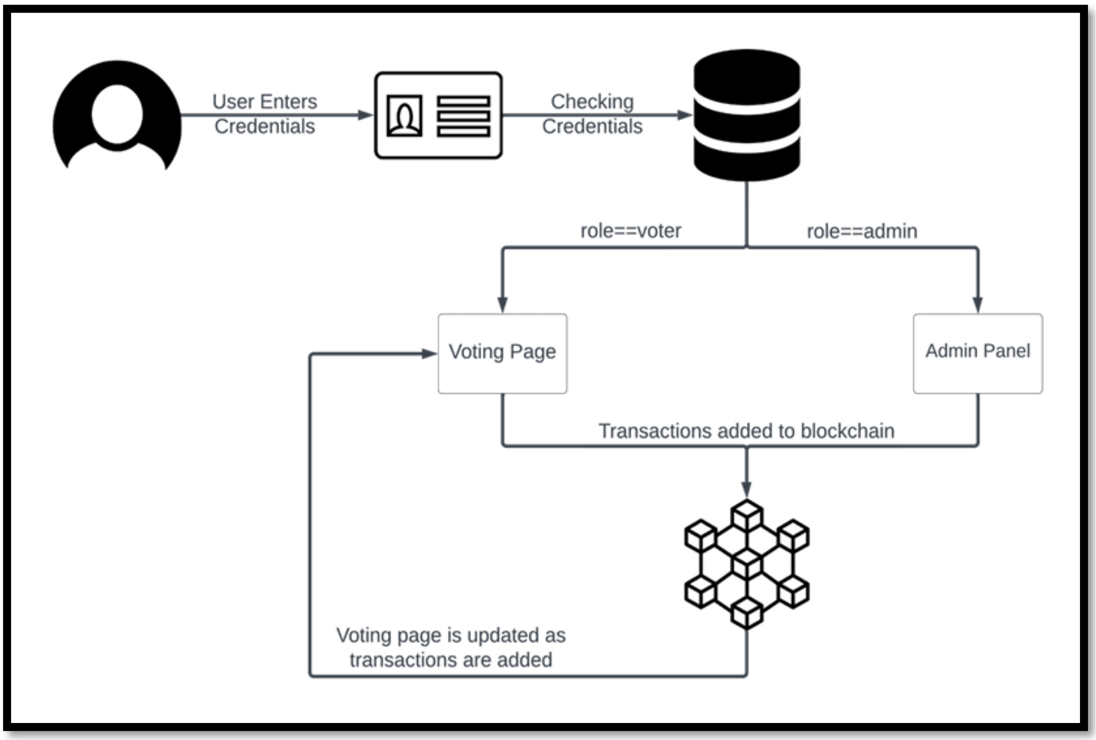

Figure 4.1 System Architecture

System Flow

User enters the credentials (voter id & password) and they are matched with the database. If the match is found user is either redirected to admin page or voter page as per their role corresponding to the credentials in the database. Once the admin is logged in he/she can start the voting process by adding candidates and defining dates. Voter can vote once the voting process has been started. Once the voter has voted the transaction is recorded to the blockchain and the voting page is updated with real-time votes.

4.2 Proposed Modules

1. Voter Module

The **Voter Module** is designed for individuals who are eligible to participate in the election process. It provides an easy-to-use interface for voters to securely cast their votes while ensuring their privacy and the integrity of the election.
Key features of the **Voter Module** include:

- **User Authentication** – Voters need to log in using their unique credentials to access the voting system. This ensures that only registered voters can participate.
- **Candidate Information** – Before casting their vote, voters can view details about all the candidates running for election, including their names, political affiliations (if applicable), and other relevant information.
- **Casting a Vote** – Once authenticated, voters can select their preferred candidate and submit their vote. The vote is then recorded securely on the blockchain.
- **Vote Verification** – After submitting a vote, the voter can verify that their vote has been successfully recorded without revealing their identity.
- **Anonymity & Security** – The system ensures that votes remain anonymous, preventing anyone from linking a specific vote to a particular voter.
- **Remote Voting** – Since the system is online, eligible voters can cast their votes from anywhere, eliminating the need to visit a physical polling station.

2. Admin Module

The **Admin Module** is designed for election officials responsible for managing the voting process. This module allows administrators to set up and oversee elections, ensuring that everything runs smoothly.

Key features of the **Admin Module** include:

- **Election Setup** – Admins can configure election details such as start and end dates, candidate registration, and voter eligibility requirements.
- **Candidate Management** – The admin is responsible for verifying and approving candidates who wish to participate in the election.
- **Voter Registration & Management** – The admin ensures that only eligible voters are registered in the system. If needed, they can add, modify, or remove voter details before the election begins.
- **Monitoring the Voting Process** – Admins can track voter participation and system performance throughout the election period to ensure smooth operations.
- **Vote Counting & Result Declaration** – Once the voting period ends, the system automatically tallies the votes, and the admin can verify and publish the final election results.
- **Security & Fraud Prevention** – The admin ensures that the voting process is secure, preventing any unauthorized access or duplicate voting.

4.3 Data Flow Diagrams

- **Level 0 data flow diagram**

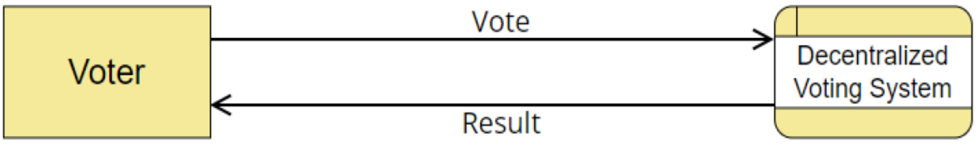

Figure 4.3(a) Level 0 Data Flow Diagram

- **Level 1 data flow diagram**

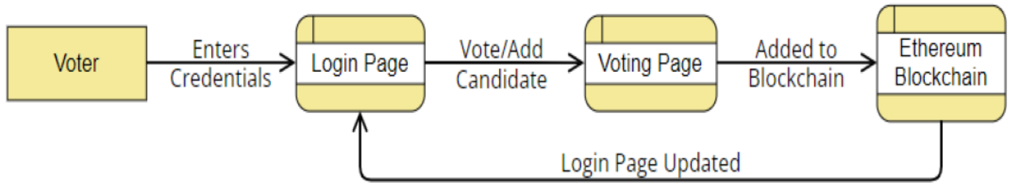

Figure 4.3(b) Level 1 Data Flow Diagram

- **Level 2 data flow diagram**

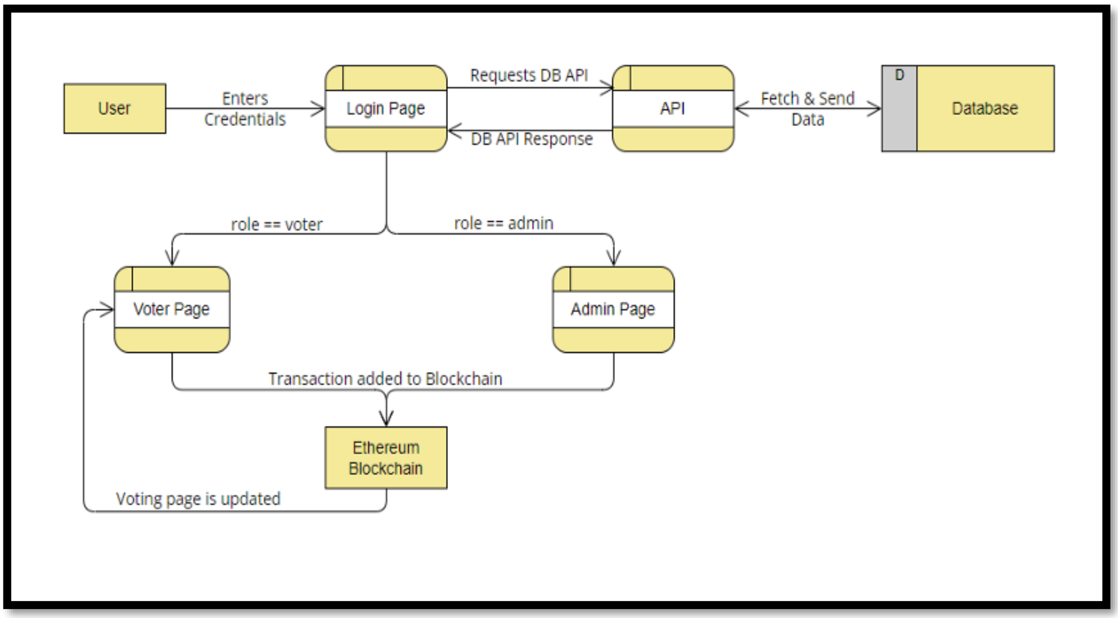

Figure 4.3(c) Level 2 Data Flow Diagram

4.4 Use Case Diagram

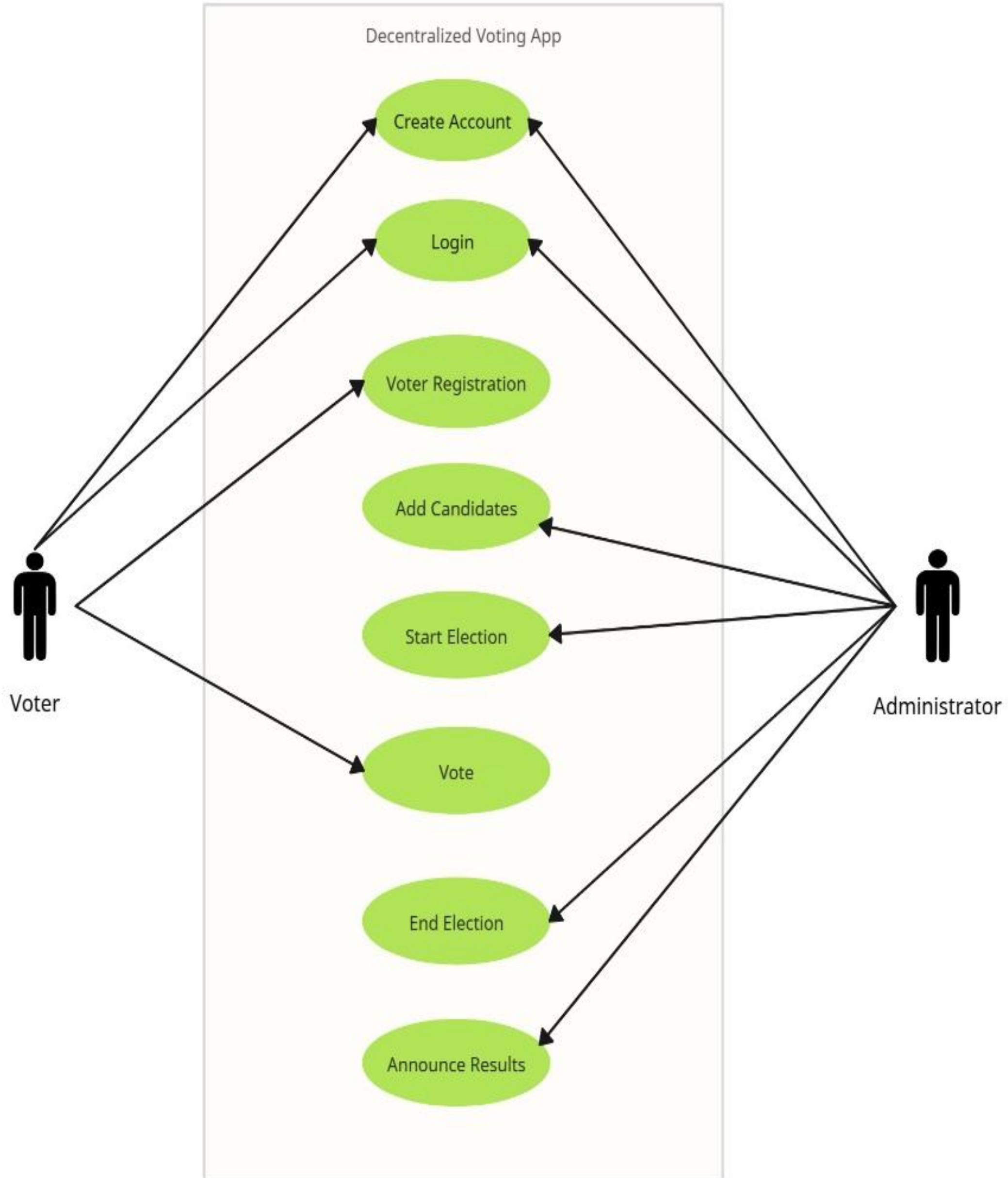

Figure 4.4(a) Use Case Diagram

4.4 Class Diagram

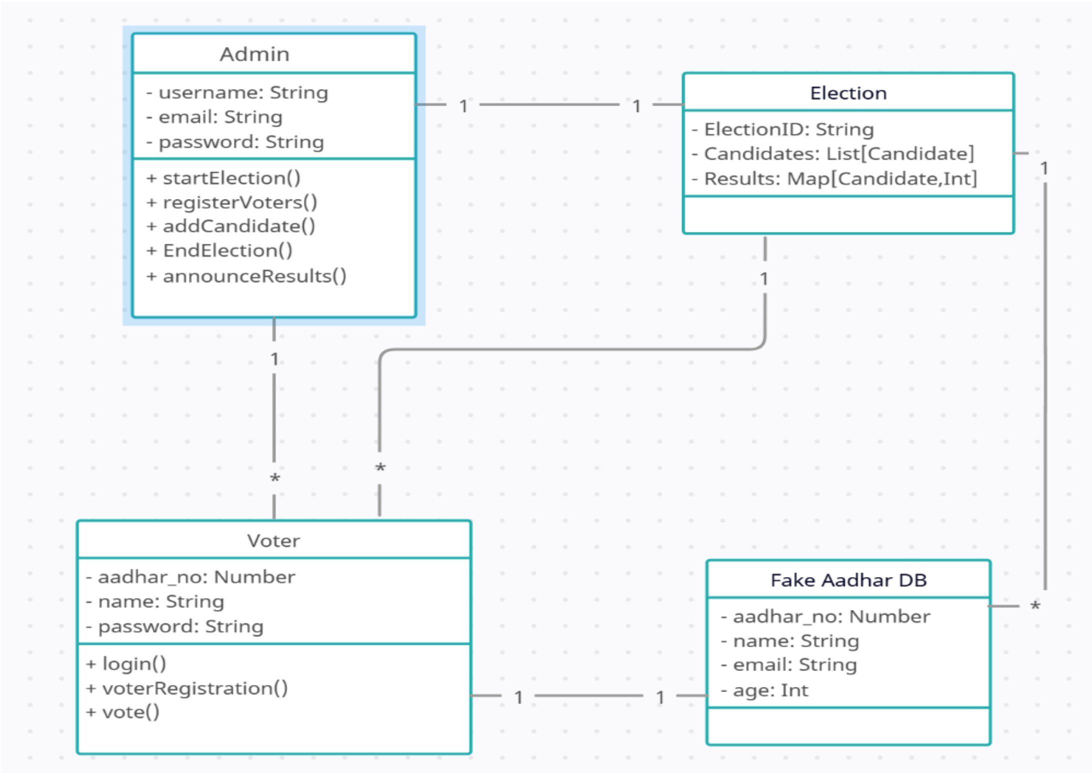

Figure 4.4(b) Class Diagram

4.4 Activity Diagram

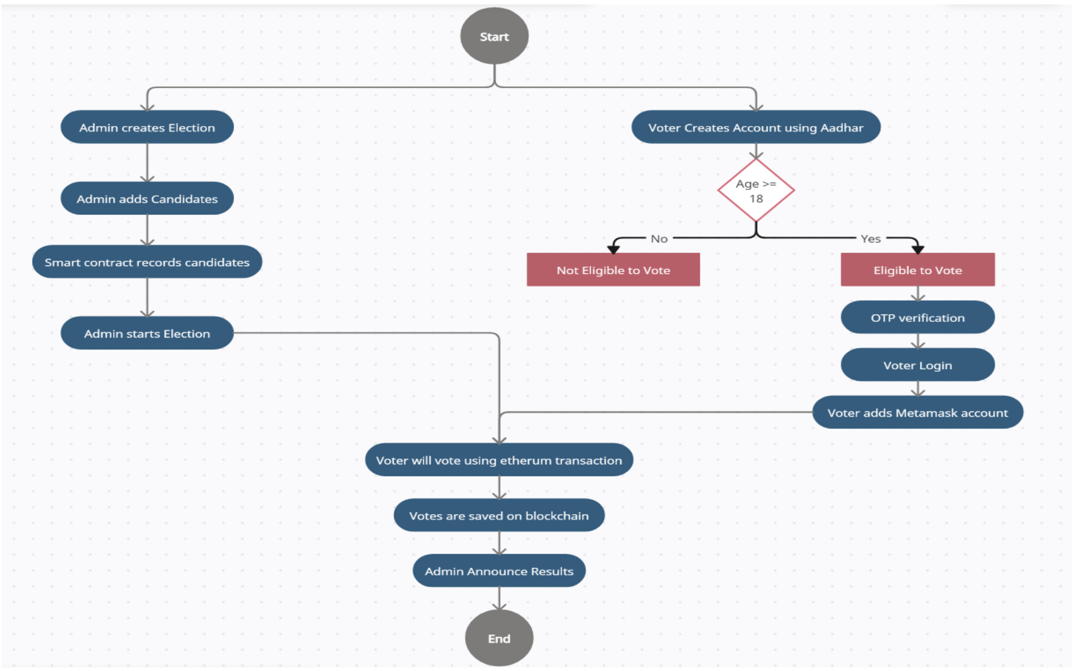

Figure 4.4(c) Activity Diagram

4.4 Sequence Diagram

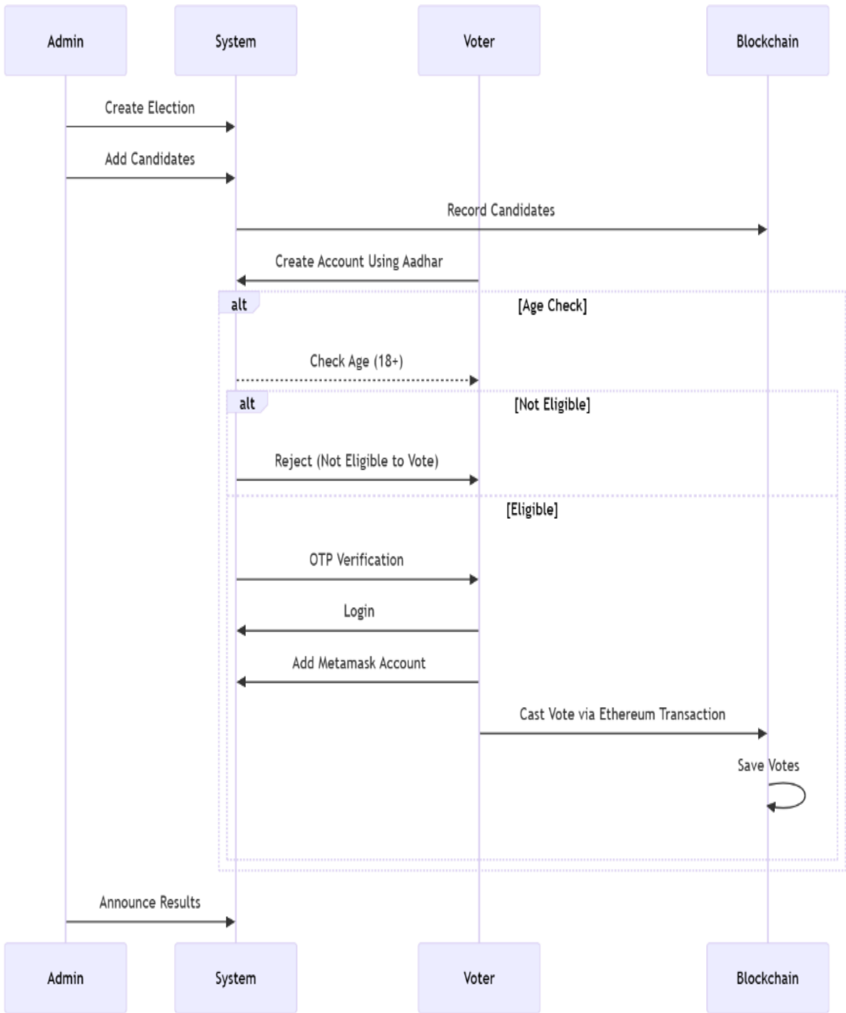

Figure 4.4(d) Sequence Diagram

CHAPTER 5

IMPLEMENTATION

5.1 Code Snippets

Smart Contract Code:

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

contract Voting {
    address public admin;
    uint256 public candidateCount;
    uint256 public voterCount;
    uint256 public verifiedVoterCount;
    uint256 public votedVoterCount;
    bool public start;
    bool public end;

    enum Phase { Registration, Voting, Results, Ended }
    Phase public currentPhase;

    struct Candidate {
        uint256 candidateId;
        string candidateName;
        string description;
        uint256 voteCount;
    }
    mapping(uint256 => Candidate) public candidateDetails;

    struct ElectionDetails {
        string electionName;
        string description;
    }
    ElectionDetails public electionDetails;

    struct Voter {
        address voterAddress;
        string name;
        bytes32 aadhaarHash;
        bool isVerified;
        bool hasVoted;
        bool isRegistered;
    }
    mapping(address => Voter) public voterDetails;
    address[] public voters;
```

```
    event ElectionCreated(string name, string description);
    event PhaseChanged(Phase newPhase);
    event CandidateAdded(uint256 candidateId, string name);
    event VoterRegistered(address voter, string name);
    event VoterVerified(address voter, bool verified);
    event VoteCast(address voter, uint256 candidateId);
    event ElectionEnded();

    modifier onlyAdmin() {
        require(msg.sender == admin, "Only admin can perform this action");
        _;
    }

    modifier onlyVerifiedVoter() {
        require(voterDetails[msg.sender].isVerified, "Only verified voters can vote.");
        _;
    }

    constructor() {
        admin = msg.sender;
        currentPhase = Phase.Registration;
    }

    function createElection(string memory _name, string memory _description) public
onlyAdmin {
        electionDetails = ElectionDetails(_name, _description);
        start = true;
        end = false;
        currentPhase = Phase.Registration;
        emit ElectionCreated(_name, _description);
    }

    function addCandidate(string memory _candidateName, string memory
_description) public onlyAdmin {
        require(currentPhase == Phase.Registration, "Can only add candidates during
Registration phase.");
        candidateDetails[candidateCount] = Candidate(candidateCount,
_candidateName, _description, 0);
        emit CandidateAdded(candidateCount, _candidateName);
        candidateCount++;
    }

    mapping(bytes32 => address) public aadhaarToAddress;
    function registerVoter(string memory _name, string memory _aadhaar) public {
        bytes32 aadhaarHash = keccak256(abi.encodePacked(_aadhaar));

        require(voterDetails[msg.sender].isRegistered == false, "Voter already
registered.");
```

```
        require(aadhaarToAddress[aadhaarHash] == address(0), "This Aadhaar is
already registered with another account.");

        voterDetails[msg.sender] = Voter(msg.sender, _name, aadhaarHash, false, false,
true);
        aadhaarToAddress[aadhaarHash] = msg.sender; // Link Aadhaar to Ethereum
address

        voters.push(msg.sender);
        voterCount++;

        emit VoterRegistered(msg.sender, _name);
    }

    function verifyVoter(address voterAddress, bool _verifiedStatus) public
onlyAdmin {
        require(voterDetails[voterAddress].isRegistered, "Voter not registered");
        require(!voterDetails[voterAddress].isVerified, "Voter is already verified");

        voterDetails[voterAddress].isVerified = _verifiedStatus;

        if (_verifiedStatus) {
            verifiedVoterCount++; // Increment verified voter count
        }

        emit VoterVerified(voterAddress, _verifiedStatus);
    }

    function vote(uint256 candidateId) public onlyVerifiedVoter {
        require(currentPhase == Phase.Voting, "Voting phase has not started.");
        require(!voterDetails[msg.sender].hasVoted, "Already voted");
        require(candidateId < candidateCount, "Invalid candidate ID");

        candidateDetails[candidateId].voteCount++;
        voterDetails[msg.sender].hasVoted = true;
        votedVoterCount++; // Increment voted voter count

        emit VoteCast(msg.sender, candidateId);
    }

    function changePhase() public onlyAdmin {
        require(currentPhase != Phase.Ended, "Election already ended.");

        if (currentPhase == Phase.Registration) {
            require(candidateCount > 1, "Add minimum of 2 candidates before starting
voting phase.");
            currentPhase = Phase.Voting;
        } else if (currentPhase == Phase.Voting) {
            currentPhase = Phase.Results;
        } else if (currentPhase == Phase.Results) {
```

```
      currentPhase = Phase.Ended;
      start = false;
      end = true;
      emit ElectionEnded();
    }

    emit PhaseChanged(currentPhase);
  }

  function getCurrentPhase() public view returns (string memory) {
    if (currentPhase == Phase.Registration) return "Registration";
    if (currentPhase == Phase.Voting) return "Voting";
    if (currentPhase == Phase.Results) return "Results";
    return "Election Ended";
  }

  function getCandidates() public view returns (Candidate[] memory) {
    Candidate[] memory candidates = new Candidate[](candidateCount);
    for (uint256 i = 0; i < candidateCount; i++) {
      candidates[i] = candidateDetails[i];
    }
    return candidates;
  }

  function getVoters() public view returns (Voter[] memory) {
    Voter[] memory voterList = new Voter[](voterCount);
    for (uint256 i = 0; i < voterCount; i++) {
      voterList[i] = voterDetails[voters[i]];
    }
    return voterList;
  }

  function getTotalVoter() public view returns (uint256) {
    return voterCount;
  }

  function getRegisteredVoterCount() public view returns (uint256) {
    return voterCount;
  }

  function getVerifiedVoterCount() public view returns (uint256) {
    return verifiedVoterCount;
  }

  function getVotedVoterCount() public view returns (uint256) {
    return votedVoterCount;
  }

  function getResults() public view returns (Candidate[] memory) {
```

```
        require(currentPhase == Phase.Results || currentPhase == Phase.Ended, "Results
are not available yet.");
        Candidate[] memory results = new Candidate[](candidateCount);
        for (uint256 i = 0; i < candidateCount; i++) {
            results[i] = candidateDetails[i];
        }
        return results;
    }
}
```

Blockchain Context Provider

a React Context Provider for managing blockchain interactions in a React application that uses Web3.js and a Voting smart contract deployed on Ethereum.

Code:

```
import React, { createContext, useState, useEffect } from "react";
import Web3 from "web3";
import VotingContract from "../contracts/Voting.json"; // Ensure correct path

export const BlockchainContext = createContext();

const BlockchainProvider = ({ children }) => {
    const [web3, setWeb3] = useState(null);
    const [account, setAccount] = useState("");
    const [contractInstance, setContractInstance] = useState(null);
    const [errorMessage, setErrorMessage] = useState("");

    // Initialize Web3 and Contract
    const initWeb3 = async () => {
        console.log("Web3 initialization")
        if (window.ethereum) {
            try {
                const web3Instance = new Web3(window.ethereum);
                await window.ethereum.request({ method: "eth_requestAccounts" });

                const accounts = await web3Instance.eth.getAccounts();
                const networkId = await web3Instance.eth.net.getId();
                const deployedNetwork = VotingContract.networks[networkId];

                if (!deployedNetwork) {
                    setErrorMessage("Smart contract not deployed on this network.");
                    return;
                }

                const instance = new web3Instance.eth.Contract(
```

```
                VotingContract.abi,
                deployedNetwork.address
            );

            setWeb3(web3Instance);
            setAccount(accounts[0]);
            setContractInstance(instance);
        } catch (error) {
            console.error("Web3 Initialization Error:", error);
            setErrorMessage("Failed to connect to blockchain.");
        }
    } else {
        setErrorMessage("Please install MetaMask.");
    }
};

useEffect(() => {
    initWeb3(); // Call initWeb3 when provider mounts
}, []);

return (
    <BlockchainContext.Provider
        value={{
            web3,
            account,
            contractInstance,
            errorMessage,
            initWeb3
        }}
    >
        {children}
    </BlockchainContext.Provider>
);
};

export default BlockchainProvider;
```

5.1 Front Page Screenshot

The front page of the decentralized voting system serves as the landing page for users. It consists of two main options: **Admin Login** and **Voter Login**.

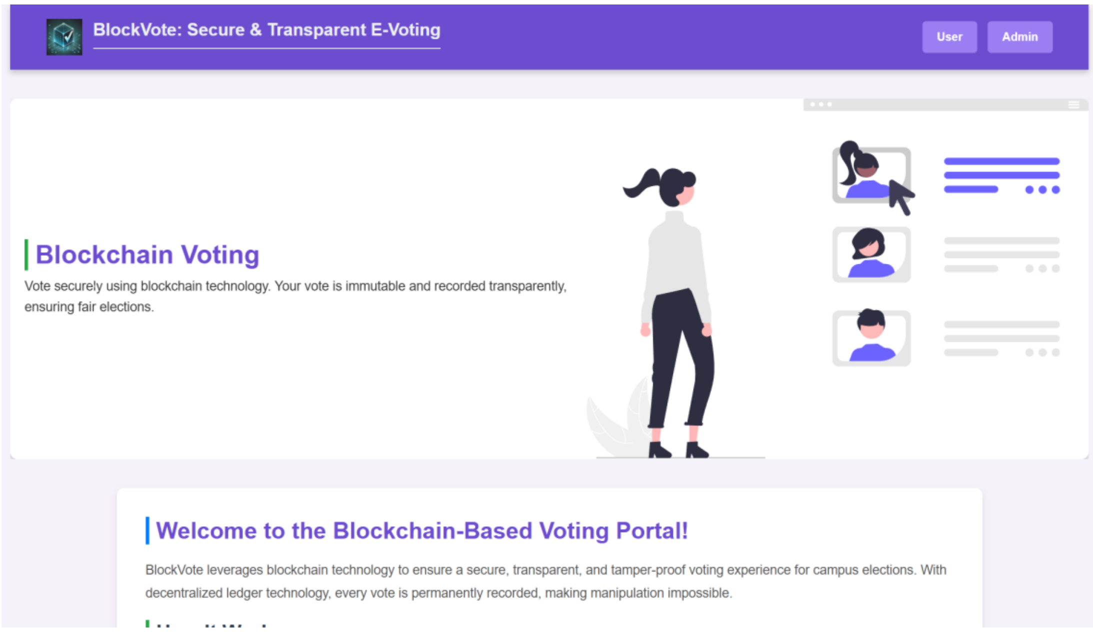

Figure 5.1 Landing Page

The page has a user-friendly interface with a simple navigation system. The admin can log in to manage candidates and voting sessions, while the voter logs in to cast their vote securely.

Voter Login Page

The Voter Login Page provides a secure authentication process for eligible voters to access the voting system. To log in, the voter must enter their Aadhar ID and password. Upon submission, the system retrieves the registered email associated with the Aadhar ID from the Aadhar database. An OTP (One-Time Password) is then sent to the voter's email for verification. Once the voter successfully enters the OTP, they are granted access to the Voter Dashboard, where they can view election details and cast their vote securely.

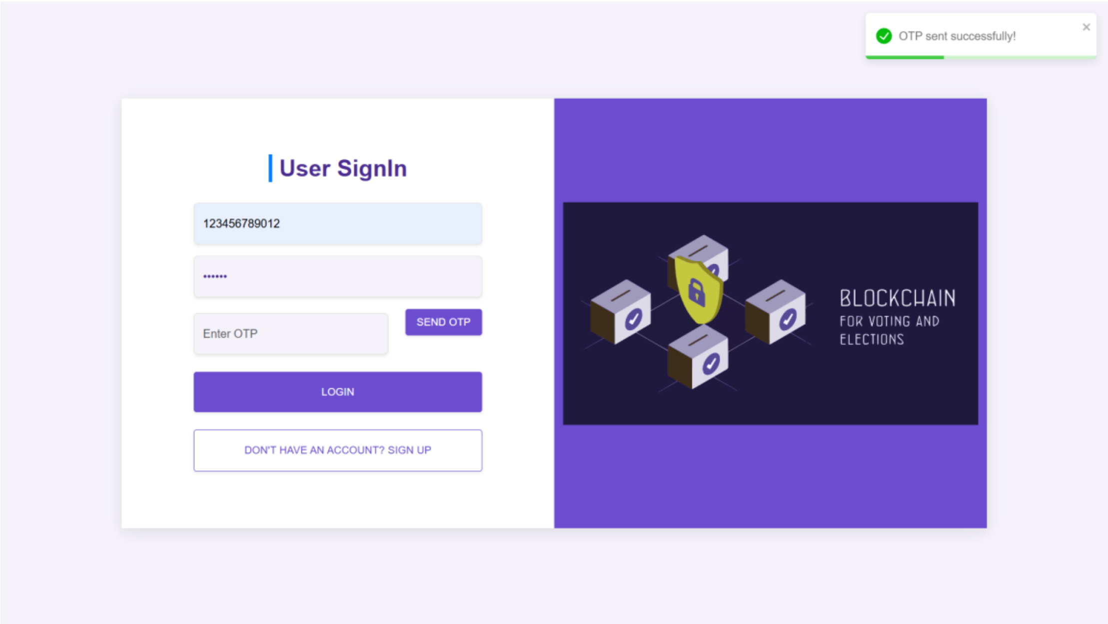

Figure 5.2 Voter Login/Signup page

This multi-step authentication process ensures that only authorized users can participate in the voting process, enhancing security and preventing unauthorized access.

Voter Dashboard

The **Voter Dashboard** is the main interface for voters to participate in the election process. It provides clear instructions and access to key features:

- **Voter Registration** – Enables eligible voters to register before casting their vote.
- **Voting Section** – Allows voters to view candidates and securely cast their votes.
- **View Results** – Displays election results transparently once voting concludes.

The dashboard ensures a smooth and secure voting experience with a user-friendly design.

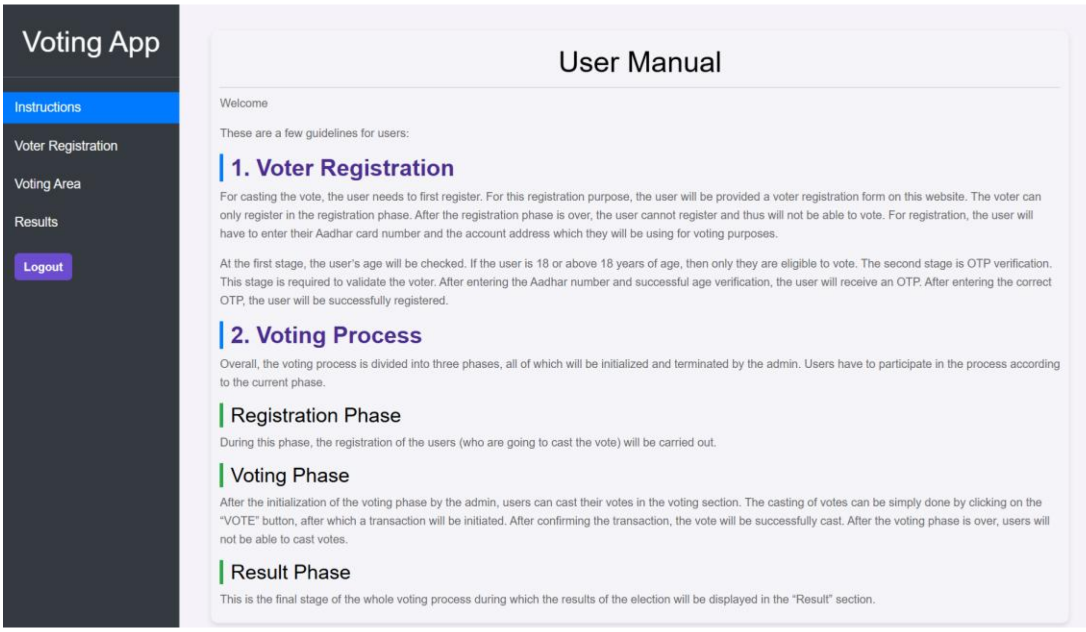

Figure 5.3 Voter Dashboard

Voting Page

The **Voting Page** is the core part of the system where registered voters can securely cast their votes. It is designed to be simple and easy to use while ensuring accuracy and transparency.

Features:

- **Candidate List:** Displays the names, party affiliations, and other relevant details of the candidates.
- **Secure Authentication:** Only verified users can access the voting page after OTP verification.
- **One-Time Voting:** Voters can select their preferred candidate and confirm their choice. Once submitted, the vote cannot be changed.
- **Vote Confirmation:** A confirmation message is displayed after a successful vote submission.
- **Transparency:** The voter can later check whether their vote was successfully recorded (without revealing their choice).

The page ensures a smooth and secure voting process while maintaining voter anonymity.

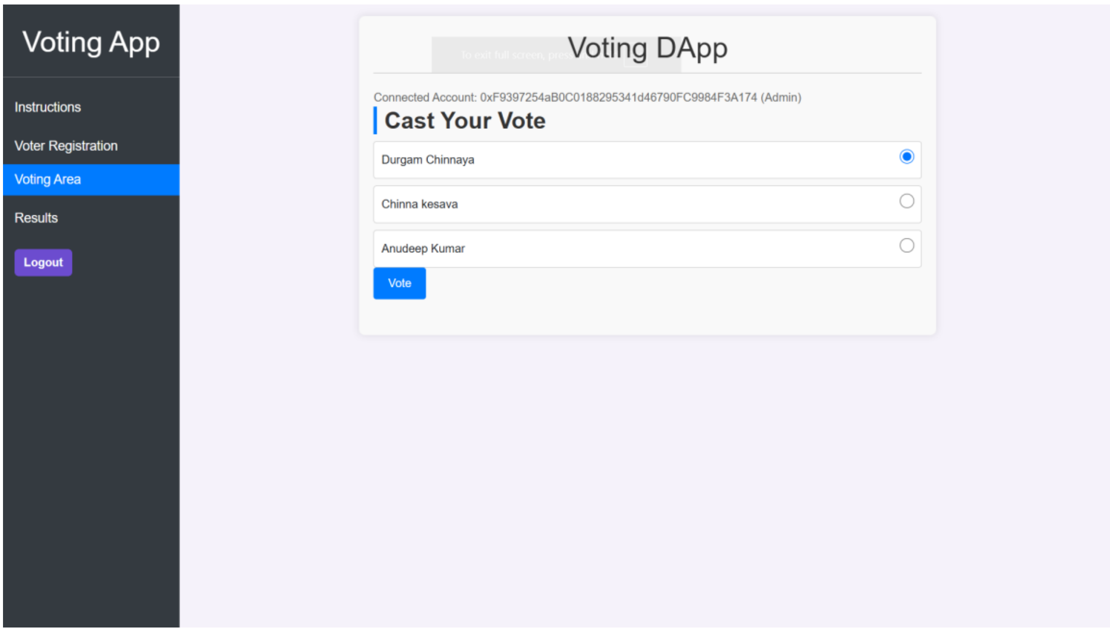

Figure 5.4 Voting Page

Admin Login Page

The Admin Login Page allows election officials to securely access the admin dashboard. The admin enters their email and password for authentication.

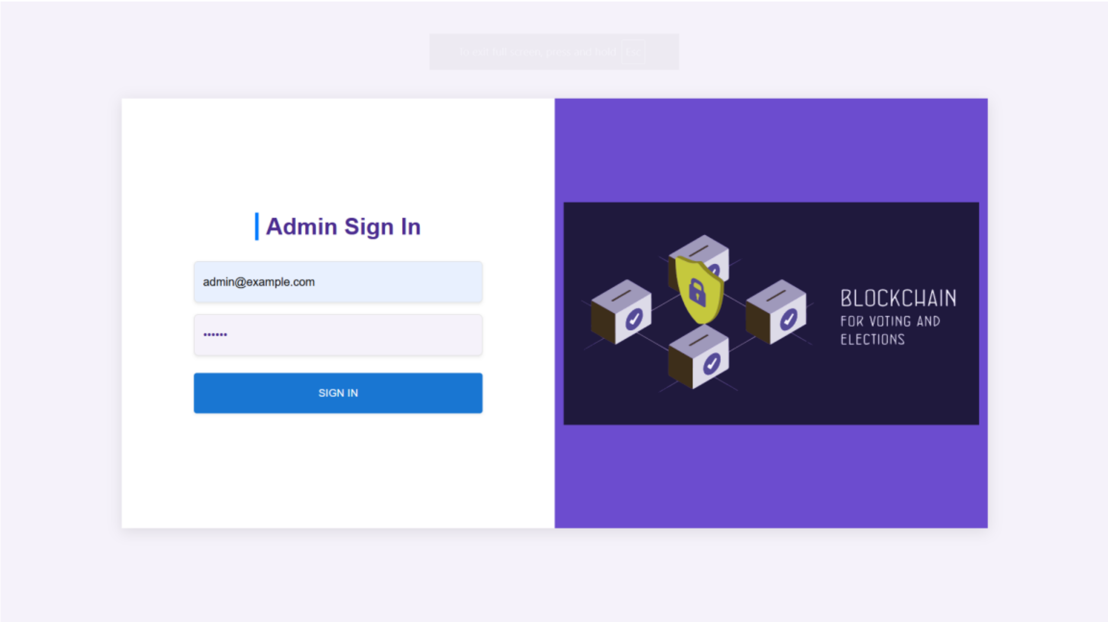

Figure 5.5 Admin Login page

Upon successful login, they are redirected to the Admin Dashboard, where they can manage the election process, including candidate verification, voter management, and monitoring the voting system.

Admin Dashboard

The Admin Dashboard serves as the control center for managing the election process. It provides access to essential administrative functionalities:

- **Candidate Management** – Allows admins to verify and approve candidate registrations.
- **Voter Management** – Enables monitoring and managing registered voters.
- **Election Configuration** – Provides options to set up election parameters like start and end dates.
- **Voting Monitoring** – Tracks live voting progress and ensures smooth operations.
- **Results Overview** – Displays real-time election results after voting ends.
- **System Management** – Maintains logs and security settings for audit and control.

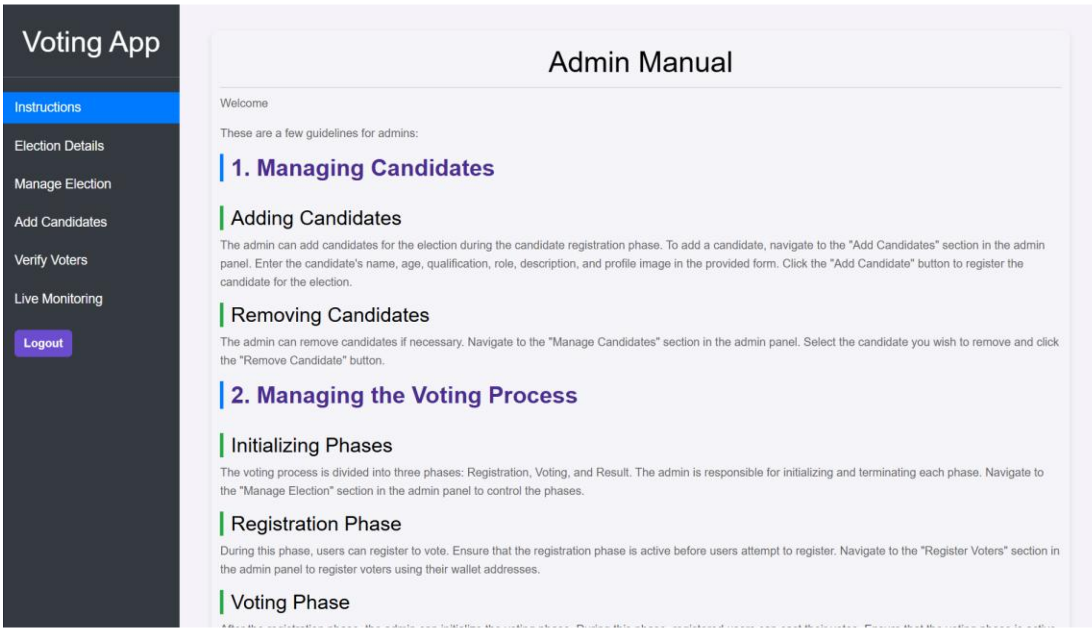

Figure 5.6 Admin Dashboard

The dashboard ensures secure and efficient election management with an intuitive interface.

5.2 Results and Discussions

Results of the Decentralized Voting System

After implementing the decentralized voting system using blockchain, the following results were observed:

1. **Security Enhancement:** The system successfully prevents any unauthorized modification of votes, ensuring transparency.
2. **Decentralization:** The voting data is stored on a blockchain network, eliminating the risk of single-point failures.
3. **Real-time Voting Updates:** The votes are recorded and reflected in real time on the blockchain.
4. **Anonymity and Privacy:** The identity of voters remains confidential while ensuring that each vote is counted only once.
5. **Ease of Access:** The system enables voters to cast votes from any location with an internet connection.

Discussions on Observations

- **Transparency and Trust:** Voters can verify their vote on the blockchain, ensuring no tampering occurred.
- **Scalability Concerns:** Blockchain transactions may take time if there is high network traffic. Optimizations in the consensus mechanism (e.g., PoA instead of PoW) can address this.
- **Gas Fees:** Ethereum transactions require gas fees, which need to be minimized for cost-effectiveness.

The proposed system outperforms traditional voting methods in terms of security, accessibility, and trust, making it a viable solution for future elections.

5.3 Testing

To ensure the system functions correctly and meets user expectations, several manual tests were conducted.

1. Functionality Testing

- Checked if voters can log in using their Aadhaar ID and receive an OTP for verification.
- Verified that only eligible voters can access the system.
- Ensured that votes are cast successfully and counted correctly.

2. User Interface Testing

- Manually tested the navigation flow to ensure a smooth user experience.
- Checked if all buttons, forms, and links work properly.
- Verified that error messages are displayed for invalid inputs.

3. Usability Testing

- Observed if users can easily register, vote, and check results without confusion.
- Ensured that instructions on the dashboard are clear and easy to follow.
- Tested the system with different users to check ease of access.

4. Security & Access Control Testing

- Tried logging in with incorrect credentials to confirm that unauthorized users cannot access the system.
- Verified that admins and voters can only perform actions assigned to them.
- Checked if sensitive data, such as passwords, are not visible or accessible.

5. Election Process Testing

- Simulated an election from candidate registration to vote counting.
- Verified that votes are recorded correctly and results are displayed properly.
- Ensured that once a vote is cast, it cannot be changed or duplicated.

5.4 Validation

The validation phase involved comparing the expected output with the actual system behavior.

Functional Validation

The system was tested against the functional requirements such as:

- Successful registration of voters and candidates
- Secure vote casting and immutability of transactions
- Real-time vote count updates
- Role-based access control for voters and admins

Blockchain Validation

We used **Ganache** to validate the transactions and ensure correct deployment of smart contracts. The vote transaction hash was checked on the blockchain explorer to confirm successful execution.

CHAPTER 6

CONCLUSION

6.1 Conclusion

The decentralized voting system using blockchain enhances transparency, security, and accessibility in elections. By eliminating intermediaries and leveraging **smart contracts**, it ensures that every vote is securely recorded, counted accurately, and remains immutable.

The system provides a viable alternative to traditional voting methods by addressing key issues such as:

- Fraud Prevention
- Real-time Vote Tallying
- Voter Anonymity
- Remote Accessibility

With further improvements in blockchain scalability and cost-effectiveness, this system can be adopted for large-scale elections worldwide.

6.2 Future Scope

The decentralized voting system can be enhanced with additional features in the future:

1. **Layer-2 Solutions for Cost Reduction** – Implementing Polygon or Optimistic Rollups can reduce Ethereum gas fees.
2. **Biometric Authentication** – Adding fingerprint or facial recognition for voter authentication.
3. **Multi-blockchain Compatibility** – Deploying on Hyperledger Fabric or Polkadot for better interoperability.
4. **Decentralized Identity Verification** – Using Self-Sovereign Identity (SSI) to ensure voter authenticity.

5. **Mobile App Integration** – Developing a cross-platform mobile app for easier voting.

By integrating these enhancements, blockchain-based voting can become a global standard for secure and transparent elections.

REFERENCES

[1] "A Survey on Smart Electronic Voting System Using Block-Chain Technology" https://www.jetir.org/papers/JETIR2101083.pdf

[2] Y. Zhang, Y. Li, L. Fang, P. Chen, and X. Dong, "Survey on Voting System using Blockchain Technology" 2023, doi: 10.1109/ACCESS.2023.3253682.

[3] W. Zhang et al., "A Privacy-Preserving Voting Protocol on Blockchain," in IEEE International Conference on Cloud Computing, CLOUD, 2018, doi: 10.1109/CLOUD.2018.00057.

[4] Uzma Jafar, Mohd Juzaidin Ab Aziz, Hafiz Adnan Hussain, "A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems," DOI:10.3390/s22197585.

[5] Amitabh Bhargava, "Blockchain-Based Digital Voting System: A Secure and Decentralized Electoral Process" Published in: 2022 5th International Conference on Contemporary Computing and Informatics (IC3I)

[6] VaibhavAnasune, "Literature survey- Online Voting: Voting System Using Blockchain," https://www.irjet.net/archives/V6/i6/IRJET-V6I6119.pdf.

[7] M. Tawfik, A. Almadani, and A. A. Alharbi, "A Survey of Blockchain Based on E-voting Systems," SSRN Electron. J., 2020, doi: 10.1145/3376044.3376060.

[8] D. Khader, B. Smyth, P. Y. Ryan, and F. Hao, "A fair and robust voting system by broadcast", in 5th International Conference on Electronic Voting, Vol. 205, pp 285-299, 2012.

[9] Muhammad Shoaib Farooq, Usman Iftikhar, Adel Khelifi, "A Framework to Make Voting System Transparent Using Blockchain Technology" 2022, doi: 10.1109/ACCESS.2022.3180168.

[10] by Matthew Sharp, Laurent Njilla, Chin-Tser Huang, Tieming Geng, "Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal" 2024 doi: doi.org/10.3390/network4040021.