Blockchain based Voting System

A Project Report Submitted
In Partial Fulfilment of the Requirements
for the Degree of

Bachelor of Technology
in
Computer Science & Engineering

by

Shruti Pandey (2100520100157)
Mudrika Pandey (2100520400037)

Under the Guidance of
Prof. Girish Chandra
Er. Diksha Sharma

Department of Computer Science and Engineering
Institute of Engineering & Technology
Dr. APJ Abdul Kalam Technical University, Uttar Pradesh

June, 2025

Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for requirement of any other degree.

Submitted by: - Date: 06/06/2025

(1) Name: Shruti Pandey
Roll No.: 2100520100157
Branch: CSE(SF)
Signature:

(2) Name: Mudrika Pandey
Roll No.: 2100520400037
Branch: CSE(SF)
Signature:

Certificate

This is to certify that the project report entitled **Blockchain based Voting System** presented by **Shruti Pandey** and **Mudrika Pandey** in the partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering, is a record of work carried out by them under my supervision and guidance at the department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

Prof. Girish Chandra

Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Er. Diksha Sharma

Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

We would like to express our sincere gratitude to the following individuals and organizations for their invaluable contributions to the completion of this project: First and foremost, We extend heartfelt thanks to our supervisor, **Prof. Girish Chandra**, for their unwavering support, guidance, and expertise throughout the project. His valuable insights and constructive feedback have greatly shaped the direction and outcomes of this work.

In addition, we are extremely grateful to our co-supervisor, Er. **Diksha Sharma**, for providing her valuable guidance which has contributed towards the successful implementation of our project.

We are grateful to the **Department of Computer Science Engineering, IET Lucknow** for providing the necessary resources and facilities to conduct this research. The access to Project Lab, High Speed Data Connection, Research Papers, significantly contributed to the success of this project. We would also like to extend our appreciation to our colleagues who provided assistance and encouragement during various stages of this project. Their collaboration and brainstorming sessions helped us overcome challenges and provided valuable insights.

We would like to thank the Project Co-Ordinator **Dr. Promila Bahadur** who volunteered their time and willingly shared their experiences and perspectives, and without whom this research would not have been possible.

We would like to express our gratitude to the Project Evaluation Committee Members for their continuous support and encouragement throughout this project, their timely feedback and suggestions helped us towards the improvement of our project quality. Their belief in our abilities and motivation have been instrumental in keeping us focused and determined. We would also like to acknowledge the authors of the resources, research papers, and publications that we consulted during this project. Their valuable work provided the foundation for our research and helped us gain a deeper understanding of the subject matter. Lastly, we are deeply thankful to our friends and family for their love and support during this process. They supported us a lot in the completion of this journey.

Shruti Pandey

Mudrika Pandey

Abstract

According to numerous international studies and democratic watchdog organizations, one of the most critical challenges in modern voting systems is ensuring secure, transparent, and tamper-proof elections. Traditional electronic voting systems are prone to vulnerabilities such as data manipulation, single points of failure, and limited auditability. In response to these limitations, our project introduces a **Blockchain-Based Voting System** designed to offer an end-to-end secure and transparent voting process for institutional and localized democratic Settings.

The existing solutions in blockchain voting often either lack integration with user identity verification systems or restrict their use to theoretical models with limited practical implementation. Our system addresses these gaps by combining SQLite-based user registration and authentication with Ethereum smart contracts, enabling only verified users to cast a single, immutable vote.

The backend, built using Node.js and Express.js, handles voter registration and login securely, storing critical information such as Aadhar number, name, and password in a lightweight SQLite database. Once authenticated, the voter can interact with a Solidity smart contract, deployed using Truffle and Ganache, to cast their vote directly on the blockchain.

This approach ensures that each vote is recorded transparently and is publicly verifiable without compromising voter identity. Our system prevents double voting, ensures voter anonymity, and resists tampering attempts. The blockchain records act as an immutable ledger of the election, which can be audited by any stakeholder, thereby enhancing trust in the electoral process.

Our project demonstrates a functional prototype of an end-to-end voting application that bridges traditional authentication with decentralized voting logic, laying the groundwork for a scalable and secure e-voting solution.

Contents

4: Experimental Results

5: Conclusion

Appendix

List of figures

List of tables

Chapter 1.Introduction

1.1 Introduction

Voting is the cornerstone of democracy, providing citizens with the means to express their choices and influence governance. However, traditional voting systems—whether manual or electronic—are often criticized for their vulnerability to fraud, lack of transparency, high costs, logistical challenges, and limited accessibility. In recent years, there has been a growing demand for a secure, verifiable, and efficient voting method that can restore trust in democratic processes. One of the most promising technologies for addressing these challenges is **blockchain**.

Blockchain is a **decentralized** and **distributed ledger technology** that enables the secure and transparent recording of transactions without the need for a central authority. Each block in the chain contains a cryptographic hash of the previous block, a timestamp, and transaction data. Once recorded, the data in any block cannot be altered retroactively without altering all subsequent blocks, which makes the system tamper-proof and immutable.

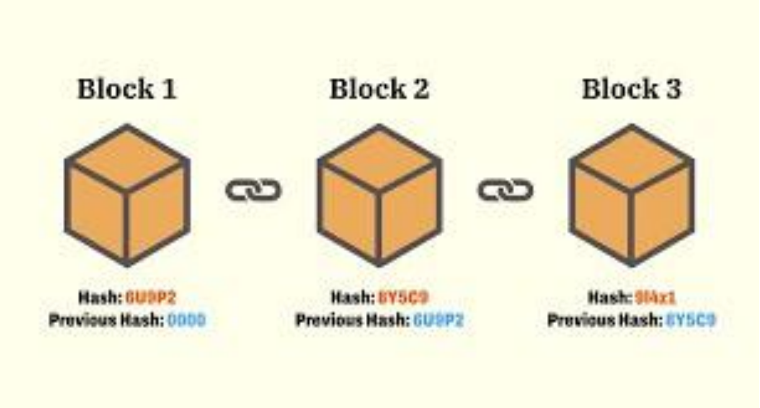


Figure 1.1: Blocks in blockchain

The core features of blockchain technology—decentralization, transparency, immutability, and cryptographic security—make it an ideal solution for voting systems. In a blockchain-based voting system, each vote is treated as a transaction and recorded on a distributed ledger. This ensures that votes cannot be altered, deleted, or duplicated, and that the entire election process is auditable by all participants, reducing the risk of electoral fraud and manipulation.

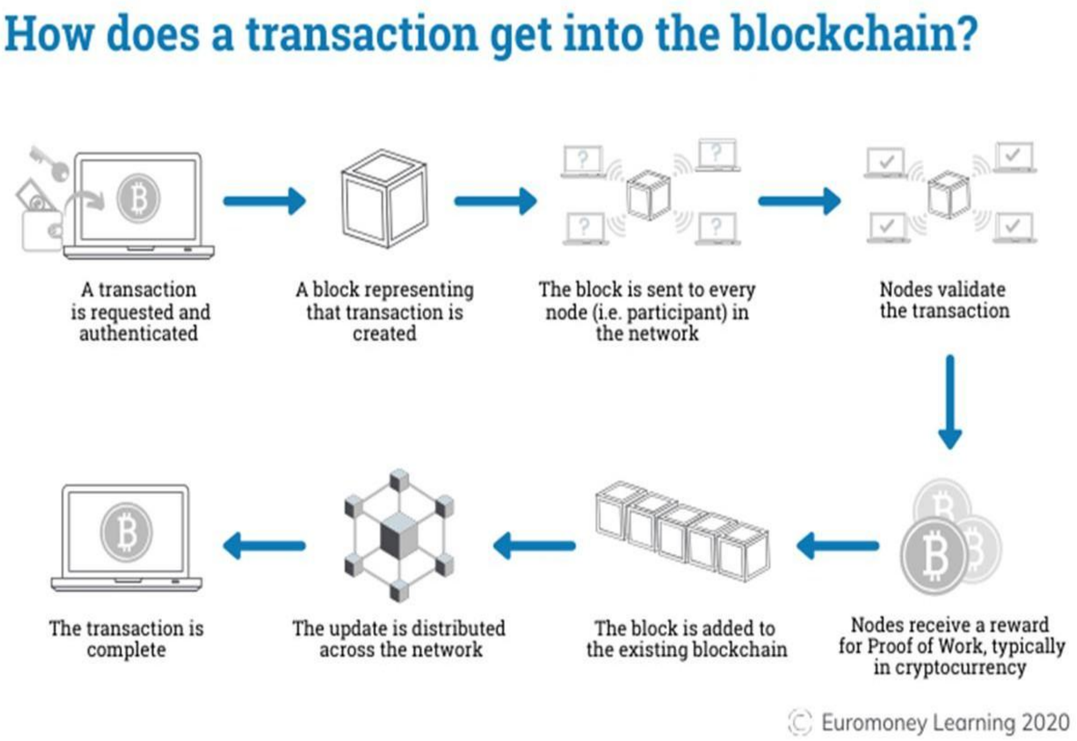

How does a transaction get into the blockchain?
A transaction is requested and authenticated
A block representing that transaction is created
The block is sent to every node (i.e. participant) in the network
Nodes validate the transaction
Nodes receive a reward for Proof of Work, typically in cryptocurrency
The block is added to the existing blockchain
The update is distributed across the network
The transaction is complete
© Euromoney Learning 2020

Figure 1.2: Transaction in a Blockchain

Our proposed project leverages blockchain technology using Ethereum smart contracts written in Solidity and executed on a local blockchain (Ganache). Voters are registered and authenticated using a secure Node.js backend and SQLite database. Once authenticated, a voter can cast a single vote which is permanently stored on the blockchain. This eliminates the need for central election authorities to count or validate votes, as the blockchain itself serves as the trusted source of truth.

This system is a prototype for how decentralized digital voting could be conducted in a secure, accessible, and verifiable manner, with potential applications ranging from university elections to national-scale referendums.

1.2 Current Voting Systems

The current methods of conducting elections in many countries involve manual or semi-automated systems, which often rely on centralized databases and authorities. These systems suffer from:

- Lack of transparency and traceability
- Risks of data manipulation or deletion
- Central points of failure
- Time-consuming and expensive logistics

Even in electronic voting systems, issues such as hardware failures, vote duplication, or software vulnerabilities can lead to fraudulent outcomes or mistrust among the electorate. Blockchain, by contrast, maintains a distributed ledger where every vote is a transaction recorded across a decentralized network. This guarantees integrity and enables real-time verification and auditability without depending on a central authority.

Vote once only in the box opposite the party or candidate of your choice using an ✗
A Party
B Party
C Party
D Party
E Party
F Party
G Party

Figure 1.3: Ballot Paper

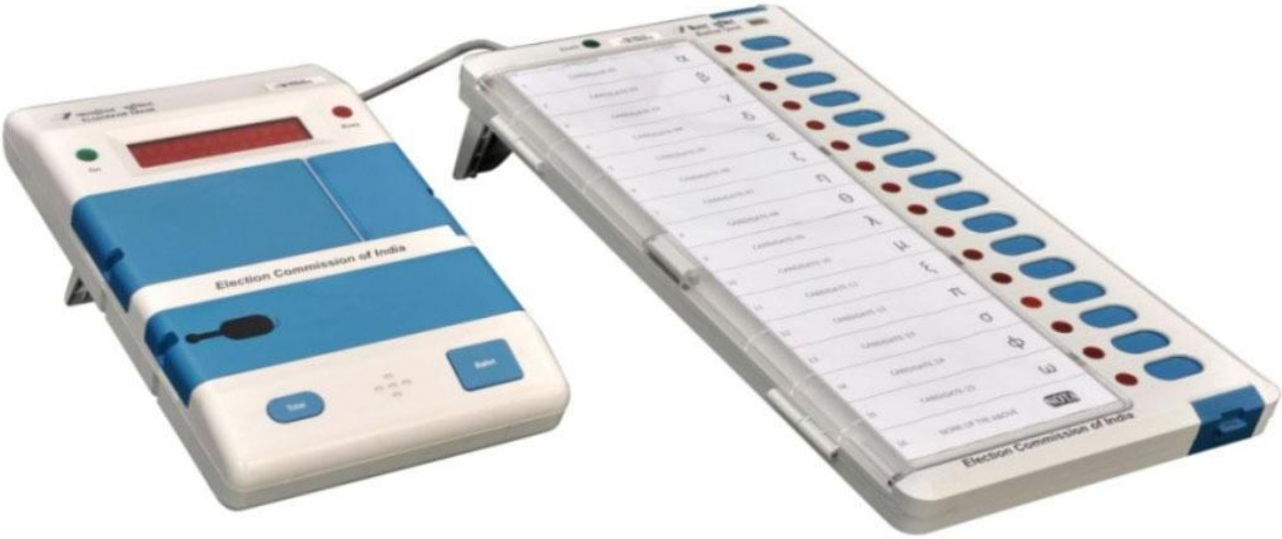


Figure 1.4: Electronic Voting Machine (EVM)

1.3 Current Work in the Field

As we reviewed the existing research and developments in the domain of digital and blockchain-based voting systems, several significant approaches and experimental implementations were found. Many of these aims to solve the challenges of traditional voting mechanisms, such as voter fraud, lack of transparency, centralized control, and logistical complexities during elections.

One of the earliest attempts to digitalize voting systems involved **electronic voting machines (EVMs)**, widely used in countries like India. Although EVMs have reduced ballot manipulation, they are still subject to hardware tampering and lack end-to-end verifiability for voters. Additionally, their reliance on central authorities for vote counting raises concerns regarding transparency.

To improve verifiability and trust, **online voting portals** were introduced in several small-scale settings (such as university elections and shareholder voting). These portals, while convenient, are vulnerable to **hacking, denial-of-service attacks, and insider manipulation**, especially because they typically run on centralized servers.

With the advent of **blockchain technology**, researchers have started exploring its application in electronic voting systems. A notable project titled “Follow My Vote” used blockchain to record votes as transactions, allowing for public auditability without revealing voter identities. Another project in Estonia tested blockchain voting for citizens residing abroad, achieving moderate success in terms of accessibility and data integrity.

A team from the **MIT Media Lab** analysed various blockchain voting prototypes and noted that while blockchain provides transparency and tamper resistance, issues such as voter authentication, internet accessibility, and vote privacy still require robust solutions.

In several university-level projects, **Ethereum smart contracts** were used to build decentralized voting apps (dApps). These systems typically allow voter registration through an admin panel, voting via MetaMask or other crypto wallets, and immutable storage of votes on the blockchain. While these systems show promise, scalability and legal acceptability in governmental elections remain areas of concern.

Thus, while blockchain-based voting systems are still in their experimental or prototype stages, the direction of current work clearly shows the potential of using decentralized technologies to create secure, transparent, and efficient electoral processes.

1.4 Problem Definition

Despite the technological advancements in many areas of public infrastructure, voting systems still suffer from inefficiency and security concerns. Some of the major challenges include:

1. Lack of transparency and auditability
2. Voter impersonation and fraud
3. Inefficient and delayed vote counting
4. Restricted accessibility for remote or disabled users
5. Over-dependence on centralized authorities

Therefore, the problem statement of our project is:
"To design and develop a decentralized, secure, and transparent blockchain-based voting system that allows only authenticated voters to cast a single immutable vote, and enables stakeholders to verify results without compromising voter privacy."

1.5 Scope of the Project

This project is aimed at building a secure, reliable, and easy-to-use electronic voting system leveraging blockchain. It includes:

- A frontend for voter registration and login
- A Node.js backend connected to an SQLite database for storing voter data
- A Solidity smart contract that manages candidate information and records votes on the Ethereum blockchain (via Ganache)

The project is suitable for use in:

- Student council elections
- Organizational board decisions

- Pilot programs for national elections

This prototype, though limited in scale, can be scaled and customized for more complex use cases.

1.6 Objectives

The main objectives of this project are:

1. To design a system that ensures only registered and authenticated voters can vote.
2. To store votes on a blockchain in a tamper-proof and transparent manner.
3. To ensure that no voter can vote more than once.
4. To enable real-time result visibility and auditability.
5. To build a system that can be reused and extended for real-world elections.

1.7 Relevance and Motivation of the Project

Free and fair elections are the cornerstone of any democracy. However, issues such as voter fraud, tampering, and lack of transparency continue to undermine the integrity of electoral processes worldwide. The emergence of blockchain technology presents a unique opportunity to address these challenges and build a more robust electoral system.

With real-time verification, immutable records, and decentralized control, blockchain can significantly reduce the risk of fraud and errors. Our motivation is to leverage this potential and create a secure and practical solution for modern voting needs, particularly in educational institutions, societies, and civic organizations.

This project is especially relevant in a digital-first world where remote participation, security, and trust in technology are critical components of future governance.

Chapter 2. Literature Review

The use of blockchain technology in voting systems has gained considerable attention in recent years due to its potential to provide transparency, security, and immutability—key attributes for a trusted electoral process. This literature review explores recent studies that propose various blockchain-based voting mechanisms, highlighting their architectures, methodologies, strengths, and limitations.

One of the earliest works in this field was presented by **Zhao and Chan**, who proposed a decentralized e-voting scheme using Ethereum smart contracts [1]. Their model focused on voter anonymity and vote integrity. By leveraging public-private key cryptography, each voter was issued a token, and votes were cast through smart contracts. The researchers emphasized the tamper-proof nature of blockchain and the possibility of verifying votes without revealing voter identity. However, the system faced scalability challenges due to Ethereum's gas fees and throughput limitations.

Swan (2015) introduced the idea of *Blockchain: Blueprint for a New Economy* [2], where blockchain was proposed not just for cryptocurrencies but as a foundational infrastructure for digital democracy. Swan discussed the application of blockchain in public governance, particularly for elections, and proposed a self-sovereign identity system where individuals control their own data. Although conceptual, the work laid theoretical groundwork for later blockchain voting platforms.

Kshetri and Voas (2018) explored the benefits and challenges of blockchain-based e-voting systems in government elections [3]. They argued that while blockchain could mitigate fraud, increase trust, and reduce administrative overhead, concerns around voter coercion, hardware security, and internet accessibility remain. Their analysis concluded that while blockchain enhances transparency, it must be complemented by robust authentication mechanisms and digital literacy among voters.

Hjalmarsson et al. (2018) presented a framework that utilizes Hyperledger Fabric for permissioned blockchain voting [4]. Their system included voter registration, vote casting, and result tallying phases. A permissioned blockchain was chosen to restrict access to trusted nodes, thereby improving scalability and control. This system demonstrated improved efficiency compared to public blockchain-based solutions, though it traded off some decentralization.

Another study by **Ali et al.** proposed a blockchain-based voting system integrated with biometric verification [5]. The system used fingerprint recognition to authenticate users before allowing them to vote via Ethereum smart contracts. Their system aimed to minimize impersonation and enhance accountability. Experimental results showed increased user confidence, but biometric integration raised concerns about privacy and the secure handling of sensitive data.

In **2019**, **Perrin and Ryan** introduced an innovative model using **Zero-Knowledge Proofs (ZKPs)** to maintain ballot secrecy while allowing vote verification [6]. Their protocol allowed voters to confirm that their vote was counted without disclosing its content. While promising, the computational complexity of ZKPs posed a barrier to large-scale implementation.

In a more practical deployment, the **Voatz platform** was piloted in the 2020 U.S. elections for overseas military voters [7]. Voatz combined blockchain with biometric security and mobile applications, offering a complete remote voting solution. However, it faced scrutiny from cybersecurity experts, who identified vulnerabilities related to vote privacy and device-level security. The case highlighted the real-world challenges of merging blockchain with mobile voting.

Patel et al. (2021) designed a voting system that utilized **IPFS (Interplanetary File System)** for decentralized vote storage and **Ethereum smart contracts** for logic execution [8]. Votes were encrypted and stored off-chain to improve scalability while maintaining integrity through blockchain hashes. Their hybrid architecture offered an optimal trade-off between cost, privacy, and performance.

Lastly, **Ghosh et al.** developed a **university-level voting application** using Solidity smart contracts and Web3.js [9]. Their project demonstrated end-to-end encryption, on-chain vote counting, and auditability. The platform was tested with student elections and showed positive feedback in terms of user experience and perceived trustworthiness.

The reviewed literature reveals a broad spectrum of blockchain-based voting applications ranging from conceptual models to real-world deployments. The core benefits offered by blockchain—immutability, decentralization, transparency, and cryptographic security—make it a compelling solution for enhancing electoral systems.

While early works like those of Zhao and Swan focused on the theoretical potential of blockchain, more recent efforts have advanced toward practical implementations. Smart contract platforms such as Ethereum and Hyperledger Fabric are popular choices for their programmable and verifiable nature. Integration with technologies like biometrics, IPFS, and ZKPs further enhances security and privacy, though at the cost of increased system complexity.

However, challenges remain. Issues such as voter authentication, resistance to coercion, vote buying, scalability, and digital divide need to be addressed before large-scale adoption. Moreover, blockchain alone does not guarantee a fully secure system; it must be supported by secure devices, privacy-preserving technologies, and legal frameworks.

In conclusion, while blockchain has shown immense promise in revolutionizing voting systems, further research, testing, and standardization are necessary to develop robust, scalable, and secure solutions that can gain public trust and governmental approval.

Chapter 3. Methodology

3.1 Introduction

In this chapter, we detail the step-by-step approach adopted to design, develop, and implement the blockchain-based voting system. The methodology includes the technologies used, system architecture, the design of the smart contract, the integration between the front-end and blockchain, and how the data is securely handled using MongoDB. This system addresses the major challenges in traditional voting systems such as vote tampering, multiple voting, lack of transparency, and centralization. Our methodology is centred around the core principles of decentralization, security, transparency, and verifiability, utilizing the Ethereum blockchain and smart contract capabilities.

Key features of the system include:

- **Decentralized Voting Logic**: The core voting mechanism, including vote recording and tallying, is managed by smart contracts deployed on a blockchain network. This removes reliance on a central authority and guarantees the integrity of the voting process.

- **User Authentication and Access Control**: Voters must register with valid credentials and be approved by an administrator before they can vote. The login and voter status management are handled by the backend using a MongoDB database.

- **Single-Vote Enforcement**: Each wallet address is permitted to vote only once. This is enforced at the smart contract level, where vote attempts from the same address are programmatically rejected after the first vote.

- **Real-Time Vote Count**: Votes are tallied instantly and can be viewed by users through the frontend. Since the vote data is stored on the blockchain, results can be verified independently by any party.

- **User-Friendly Web Interface**: The frontend provides voters with a simple and intuitive interface to register, log in, connect their wallet, and cast their vote. It also allows administrators to manage candidates and monitor the system.

- **Secure and Private**: Although the voting record is stored publicly on the blockchain, no personal information of the voter is recorded on-chain. The only identifiable marker is the wallet address, which ensures **anonymity and privacy**.

- **Tamper-Resistance**: Since all votes are stored on the blockchain in an immutable ledger, the data is tamper-proof and transparent. These builds trust among participants in the voting process.

- **Hybrid Architecture**: The combination of blockchain (for voting logic) and a traditional backend (for user management) results in a balanced, efficient, and secure system that is both practical for real-world use and theoretically robust.

3.2 System Overview

The proposed **Blockchain-Based Voting System** is designed to leverage the unique strengths of blockchain technology to address common issues in traditional and electronic voting systems, such as lack of transparency, vote manipulation, multiple voting by the same user, and centralized control. The system is built as a hybrid solution that combines **decentralized blockchain smart contracts** with a **centralized backend** to manage user data and authentication securely.

The overall system functions as a secure, transparent, and tamper-proof platform where authorized users can cast their votes for registered candidates through a user-friendly web interface. Each vote is permanently recorded on a blockchain ledger using **Ethereum smart contracts**, ensuring immutability and preventing any form of manipulation.

To simplify access to blockchain features and ensure security, **MetaMask** is used to manage digital wallets and sign blockchain transactions. **Web3.js**, a JavaScript library for interacting with the Ethereum blockchain, connects the frontend with smart contracts deployed on a **local Ethereum test network** (simulated using **Ganache**).

While the blockchain handles vote transactions and results tallying, a Node.js/Express **backend** paired with SQLite is responsible for managing voter registration, storing candidate information, handling login credentials, and maintaining the status of whether a voter has already voted. This modular structure ensures that the voting logic and user data management are separated, enhancing both **scalability** and **security**.

3.3 System Architecture

The system architecture follows a modular approach consisting of:

1. **Client Layer (Frontend Web Application)**
   - Provides the user interface for voters and administrators.
   - Interacts with MetaMask and Web3.js for blockchain communication.
2. **Blockchain Layer (Smart Contract on Ethereum)**
   - Handles core voting logic including candidate management, voting, and result tallying.

- Deployed locally using Ganache.

3. **Backend Layer**
   - Stores non-sensitive data such as voter registration info and login credentials.
   - Logs vote events and ensures one-time voting via metadata.
4. **Wallet Layer (MetaMask)**
   - Used for authenticating voters and signing transactions on the blockchain.
   - Ensures only authorized users cast votes.

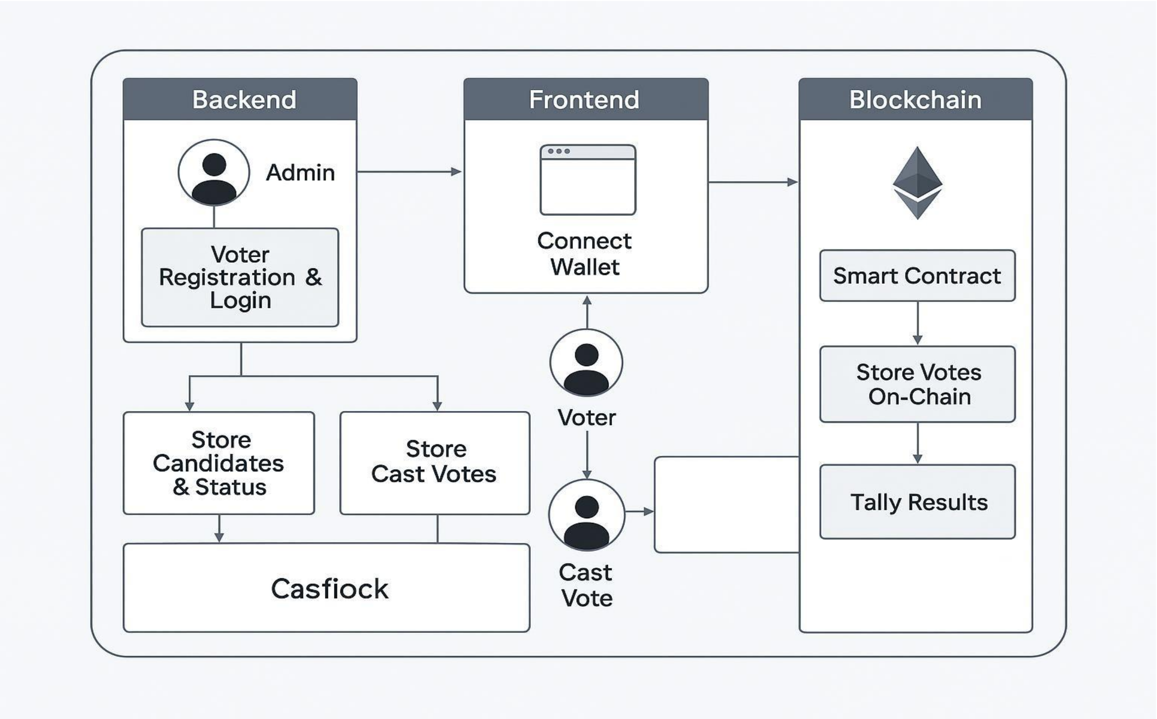


Figure 3.1: System architecture

3.4 Modules Description

3.4.1 Voter Registration and Authentication

- Voters must register with their unique credentials such as email and wallet address.
- An admin verifies voter identity before enabling voting permissions.
- Upon successful login, voters connect their MetaMask wallet.

3.4.2 Candidate Management

- Admin can add, update, or remove candidates.
- Candidate data is stored in MongoDB for UI display and is also registered on the blockchain.
- This dual storage ensures fast UI rendering and tamper-proof vote counting.

3.4.3 Vote Casting Mechanism

- Once logged in, voters view the list of candidates and cast their vote by interacting with the blockchain via Web3.js.
- Upon vote submission:
  - MetaMask prompts for transaction confirmation.
  - The smart contract records the vote securely.
- Vote values are stored in the smart contract, ensuring integrity and immutability.

3.4.4 Vote Counting and Result Display

- The smart contract has a getVotes() function that can be publicly called to retrieve vote counts.
- Results are fetched and shown in real time on the frontend.
- The decentralized nature ensures that these results are not manipulable by any single party.

3.5 Technology Stack

Component	Technology/Tool
Smart Contract	Solidity
Blockchain Network	Ethereum (via Ganache)
IDE for Smart Contracts	Truffle
Frontend	HTML, CSS, JavaScript
Blockchain Interface	Web3.js
Wallet	MetaMask
Backend	Node.js, Express.js
Database	SQLite
Development Tools	VS Code, Postman

Table 3.1: Technology stack

3.6 System Design

3.6.1 Requirement Analysis

To effectively design and develop a system, it is important to understand and document the requirements of the system. The process of gathering and documenting the requirements of a system is known as requirement analysis. It helps to identify the goals of the system, the stakeholders and the constraints within which the system will be developed. The requirements serve as a blueprint for the development of the system and provide a reference point for testing and validation.

- **Hardware Requirements**
  - Processor – 2 GHz or more
  - RAM – 4 GB or more
  - Disk Space – 100 GB or more

- **Software Requirements**
  - Node.js (version – 18.14.0)
  - Web3.js (version – 1.8.2)
  - Truffle (version – 5.7.6)
  - Solidity (version – 0.5.16)
  - Ganache (version – 7.7.3)
  - MetaMask
  - Python (version – 3.9)

3.6.2 Data Flow Diagram

- **Level 0 data flow diagram-**

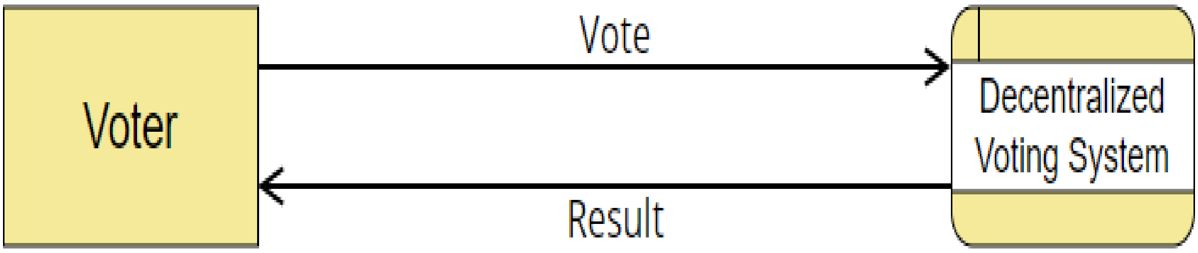


Figure 3.2: Level 0 data flow diagram

- **Level 1 data flow diagram-**

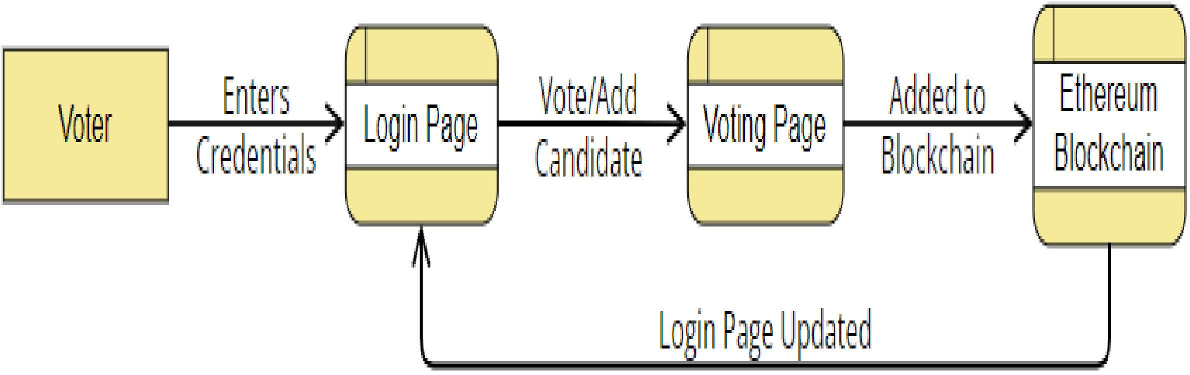


Figure 3.3: Level 1 data flow diagram

- **Level 2 data flow diagram-**

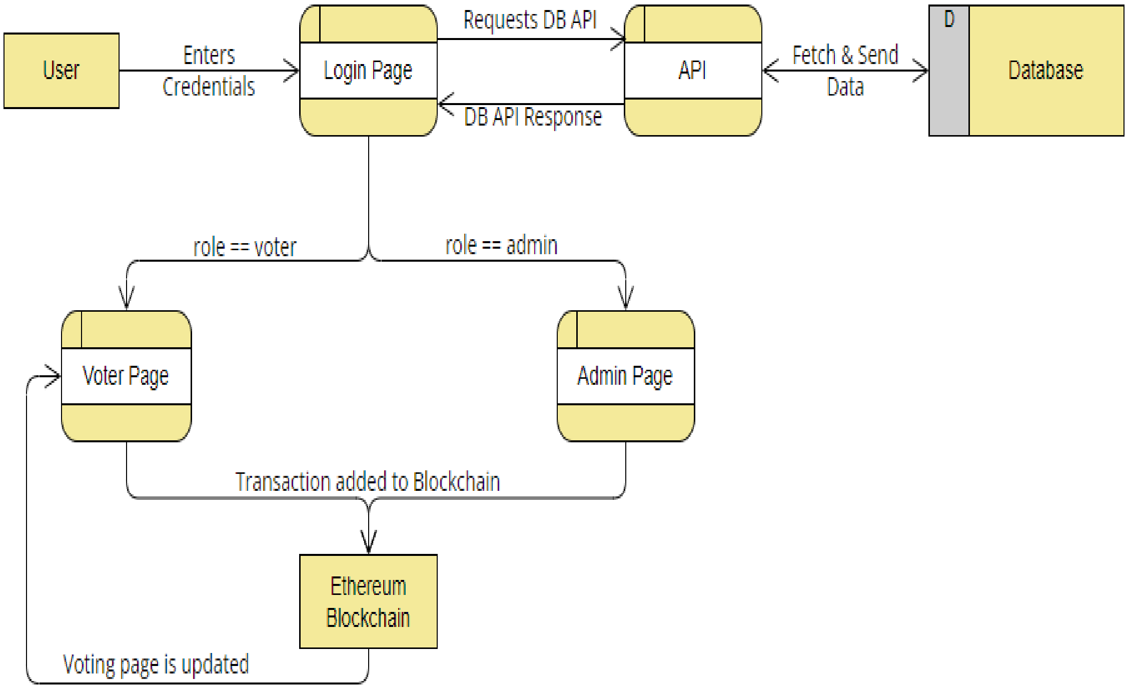


Figure3.4: Level 2 data flow diagram.

3.6.3 Use case diagram

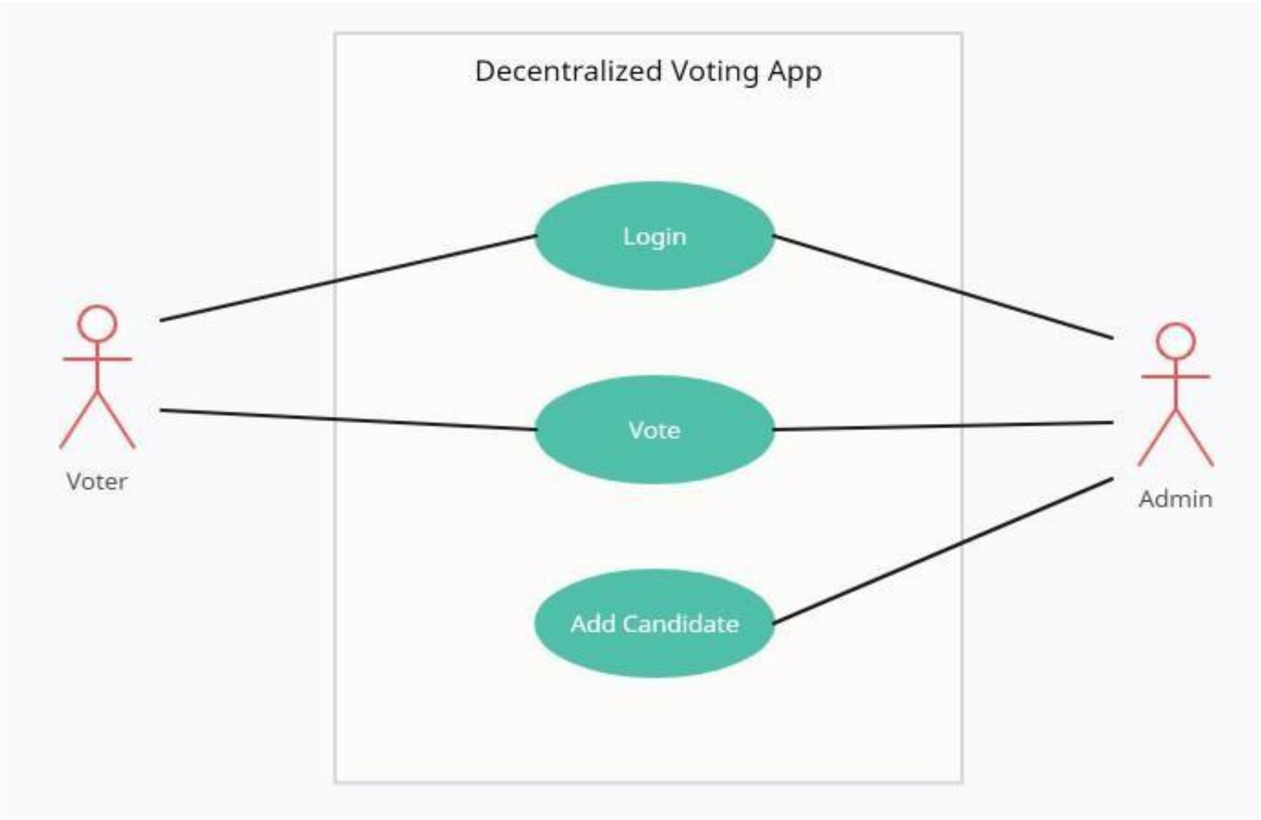


Figure 3.5: Use case diagram

Chapter 4: Experimental Results

E-voting systems based on blockchains use a variety of concepts and technologies to enable secure and trustworthy elections. Blockchain frameworks like Ethereum and Hyperledger Fabric, consensus algorithms like Proof of Work, Proof of Stake, and Practical Byzantine Fault Tolerance, and privacy-enhancing techniques like homomorphic encryption and zero-knowledge proofs are among these technologies.

Furthermore, authentication mechanisms such as biometric verification and identity management systems are critical in confirming voter legitimacy and maintaining the voting system's integrity.

In this section, we present a technology summary in five broader categories:

- Blockchain platforms;
- Consensus algorithms;
- Security and privacy techniques;
- Authentication and identity verification techniques;
- Other techniques (cryptography, development, testing).

4.1 Blockchain Platforms

The blockchain frameworks and technologies domain includes a variety of platforms and tools used in the design and implementation of blockchain-based systems. Blockchain frameworks such as Ethereum, Hyperledger Fabric, Bitcoin, and Multichain provide the foundation required for developers to create decentralized apps.

Figure below includes a range of widely used blockchain frameworks, including the proposed blockchain e-voting systems context.
In all the frameworks mentioned, Ethereum is the most popular choice, as evidenced by the 34.91% portion of utilized frameworks.
Although papers mentioned specific frameworks, there are further studies, and no specific blockchain framework is explicitly stated. Instead, they proposed customized systems that are based on the general concept of blockchain technology.
We identified several key concepts that deserve further consideration during the development and implementation of blockchain-based e-voting systems. These concepts address areas such as

- Cryptography techniques;
- Choice of development environments for smart contracts;
- Utilization of testing and benchmarking tools.

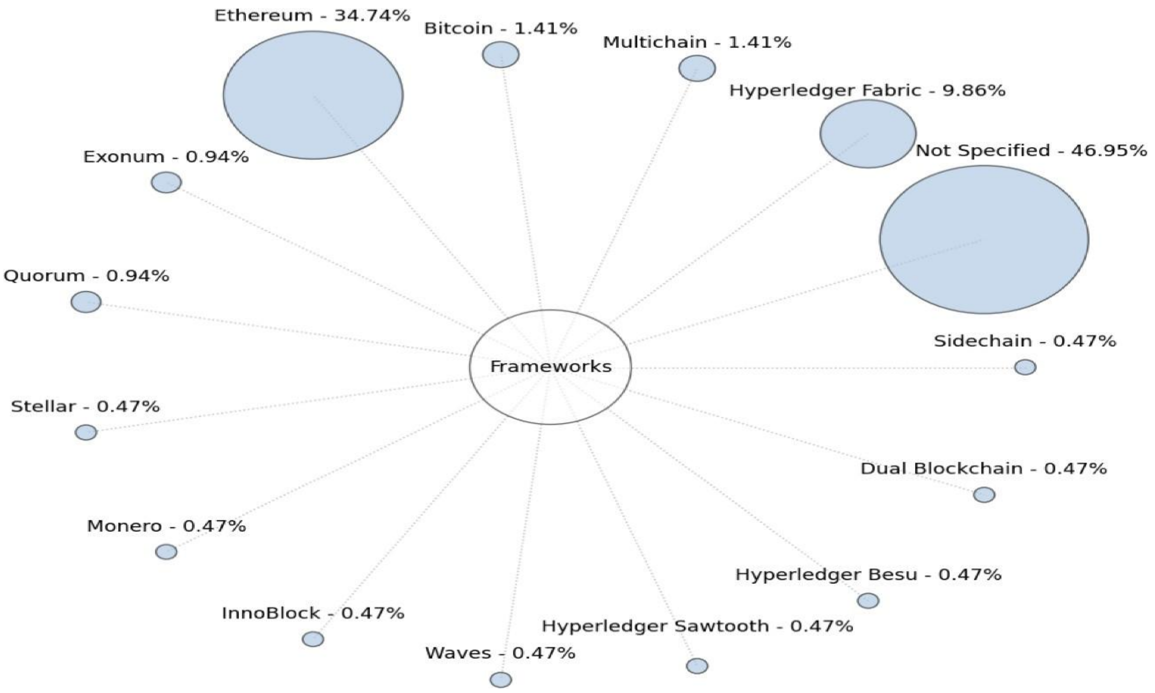


Figure 4.1: Blockchain frameworks distribution of proposed blockchain-based e-voting systems.

4.2 Consensus Algorithms

The consensus algorithms that were mentioned are illustrated in table below. Although a substantial number of papers do not explicitly mention the consensus algorithm used, it is reasonable to assume that for most proposed systems that use Ethereum as their framework, the consensus algorithm can be considered as Proof of Work (PoW). The following and most substantial protocol is referred to as “Proof of Work (PoW)”, resulting in approximately 5.2% portion of used consensus algorithms. In the following, we provide a brief definition for each of these consensus algorithms:

Technique	No. of Papers	Normalized (%)
Biometric Authentication	27	100
Aadhaar ID Verification	7	25.93
OTP (One-Time Password)	6	22.22
Multifactor Authentication	3	11.11
Multi-Step Authentication	3	11.11
PKI-based X.509	2	7.41
Unique Hash IDs	1	3.70

Table 4.1: Distribution of authentication and identity verification techniques in blockchain-based e-voting papers (if mentioned)

4.3 Test Cases

Test case	1
Name of the test	Input Aadhar
Input	Valid unique ID
Expected output	Input Aadhar feed by the user
Actual output	Valid Aadhar number is accepted as enrolled in the database
Result	Successful

Table 4.2: Input Aadhar Test Case

Test case	1
Name of the test	Email OTP authentication
Input	Valid/enrolled email-id
Expected output	Obtain OTP to the registered email
Actual output	Receiving unique OTP from the enrolled email ID
Result	Successful

Table 4.3: Email OTP Authentication Test Case

Test case	1
Name of the test	Private key verification
Input	Valid/enrolled email ID
Expected output	Obtaining unique hash value
Actual output	Receiving unique hash value using blockchain from the enrolled email ID
Result	Successful

Table 4.4: Private Key Verification Test Case

4.4 Observation

Aadhar Verification Module:

- Validated unique Aadhar IDs from the database.
- Ensured only enrolled voters could proceed to the voting stage.
- Output matched expected results – Status: Successful.

Email OTP Authentication:

- OTPs were correctly generated and sent to registered email addresses.
- Verified user identity through real-time OTP validation.
- Authentication passed without discrepancies – Status: Successful.

Private Key Verification:

- Unique hash values securely generated using blockchain mechanisms.
- Ensured tamper-proof and verifiable user identity.
- Test case met all cryptographic expectations – Status: Successful.

Security Assurance:

- No unauthorized access or data inconsistency observed.
- Cryptographic primitives (hashing, OTP) provided secure voter verification.

System Reliability:

- All modules performed as intended under test scenarios.

Chapter 5: Conclusion

5.1 Conclusion

The proposed framework provides complete security to the e-voting system, with the usage of Ethereum blockchain and smart contracts to provide added security to the system. Blockchain implementation prevents vote manipulation and provides privacy, integrity for voters to cast their vote. Smart contracts ensure that the voter can vote only once using his/her unique ID (Aadhar number). With the convention of different security algorithms like SHA-256, Merkel hash, and SMTP prototyping, the security of the system is enhanced. As a result, the voter is authorized to cast his/her vote from wherever they are; providing high security standards to the system and convenient and easier ways to vote.

5.2 Future Work

1. To the proposed existing system, additional biometrics (fingerprint, face authentication) can be added to enhance the security of the system.

2. Three-step authentications can also be used to provide more security to the system.

}

Appendix

Appendix A

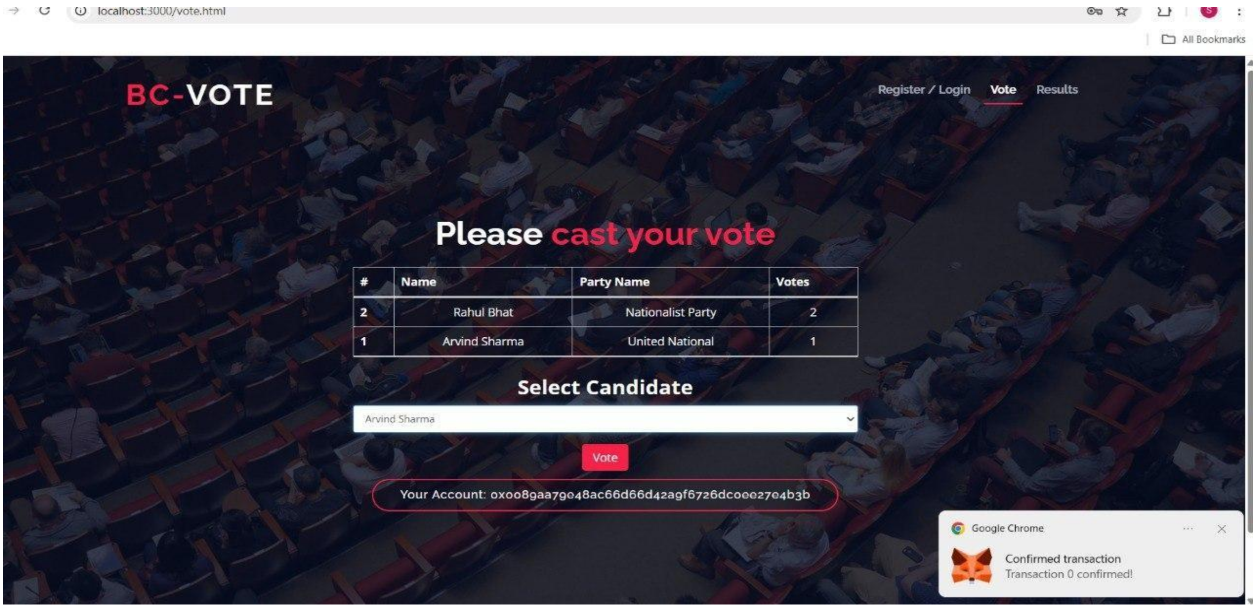
Code Snippets

A.1 Smart Contract

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

// Smart contract for a basic blockchain-based voting system contract VotingSystem {

// Address of the contract deployer (admin)
address public admin;

// Flags to track the election state
bool public electionStarted;
bool public electionEnded;

// Structure to represent a candidate
struct Candidate {
   uint id;
   string name;
   uint voteCount;
}

// Structure to represent a voter
struct Voter {
   bool authorized; // whether the voter is allowed to vote
   bool voted;      // whether the voter has voted
   uint vote;       // the candidateId voted for
}

// Mappings to store voters and candidates
mapping(address => Voter) public voters;
mapping(uint => Candidate) public candidates;

// Total number of candidates and votes
uint public totalCandidates;
uint public totalVotes;

// Constructor sets the admin to the contract deployer
constructor() {
   admin = msg.sender;
}

// Modifier to restrict functions to only admin
modifier onlyAdmin() {
    require(msg.sender == admin, "Only admin can perform this.");
    _;
}

// Function to add a candidate (only admin can do this)
function addCandidate(string memory _name) public onlyAdmin {
    candidates[totalCandidates] = Candidate(totalCandidates, _name, 0);
    totalCandidates++;
}

// Function to authorize a voter (only admin)
function authorizeVoter(address _voter) public onlyAdmin {
    voters[_voter].authorized = true;
}

// Admin starts the election
function startElection() public onlyAdmin {
    electionStarted = true;
    electionEnded = false;
}

// Admin ends the election
function endElection() public onlyAdmin {
    electionEnded = true;
    electionStarted = false;
}

// Function to vote for a candidate by ID
function vote(uint _candidateId) public {
    require(electionStarted, "Election has not started.");
    require(!electionEnded, "Election has ended.");
    require(!voters[msg.sender].voted, "You have already voted.");
    require(voters[msg.sender].authorized, "You are not authorized to vote.");

    voters[msg.sender].voted = true;
    voters[msg.sender].vote = _candidateId;

    candidates[_candidateId].voteCount++;
    totalVotes++;
}

// Function to get candidate details
function getCandidate(uint _candidateId) public view returns (string memory name, uint voteCount) {
    return (candidates[_candidateId].name, candidates[_candidateId].voteCount);
}

// Function to get the winner after the election ends
function getWinner() public view returns (string memory winnerName) {
    require(electionEnded, "Election not ended yet.");

    uint winningVoteCount = 0;
    uint winningCandidateId;

    for (uint i = 0; i < totalCandidates; i++) {
        if (candidates[i].voteCount > winningVoteCount) {
            winningVoteCount = candidates[i].voteCount;
            winningCandidateId = i;
        }
    }

    winnerName = candidates[winningCandidateId].name;
}


}
```

Appendix B

B.1 Screen Shots



Please cast your vote

#	Name	Party Name	Votes
2	Rahul Bhat	Nationalist Party	2
1	Arvind Sharma	United National	1

Select Candidate

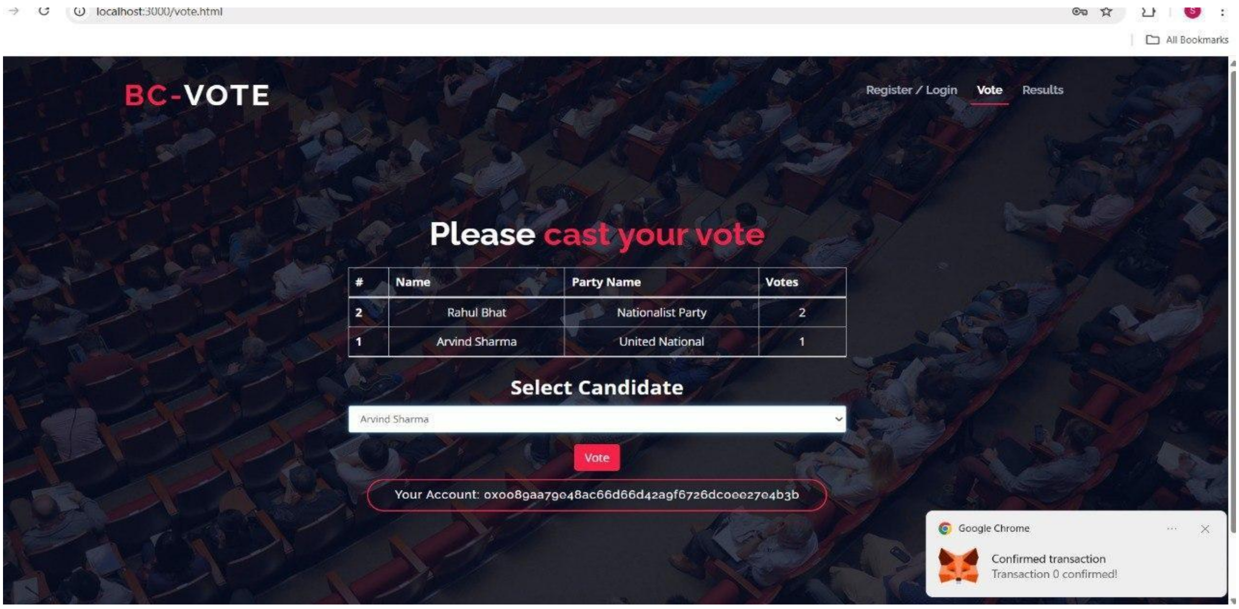
Arvind Sharma

Your Account: 0x0089aa79e48ac66d66d42a9f6726dc0ee27e4b3b

Google Chrome

Confirmed transaction
Transaction 0 confirmed!

Figure B.1: Vote Casting Page



Please cast your vote

#	Name	Party Name	Votes
2	Rahul Bhat	Nationalist Party	2
1	Arvind Sharma	United National	1

Select Candidate

Arvind Sharma

Your Account: 0x0089aa79e48ac66d66d42a9f6726dc0ee27e4b3b

Google Chrome

Confirmed transaction
Transaction 0 confirmed!

Figure B.2: Result showing page

References

[1] H. Agarwal and G. N. Pandey, "Online Voting System for India Based on AADHAR ID," *2013*. [Online]. Available: https://ieeexplore.ieee.org/document/6756265

[2] S. Chakraborty and S. Mukherjee, "Biometric voting system using Aadhaar card in India," *2016*. [Online]. Available: https://dl.acm.org/doi/10.1145/1866307.1866309

[3] B. S. Raju and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," *2019*. [Online]. Available: https://www.researchgate.net/publication/331731568_Trustworthy_Electronic_Voting_Using_Adjusted_Blockchain_Technology

[4] H. K. Prasad, A. Kankipati, and S. K. Sakhamuri, "Security Analysis of India's Voting Machine," *2010*. [Online]. Available: https://ieeexplore.ieee.org/document/8675008

[5] A. Spanos and I. Kantzavelou, "EtherVote: A Blockchain-based Electronic Voting System," *2023*. [Online]. Available: https://ar5iv.org/html/2307.10726

[6] A. Singh and K. Chatterjee, "SecEVS: Secure Electronic Voting System Using Blockchain Technology," *2018*. [Online]. Available: https://ieeexplore.ieee.org/document/9387645

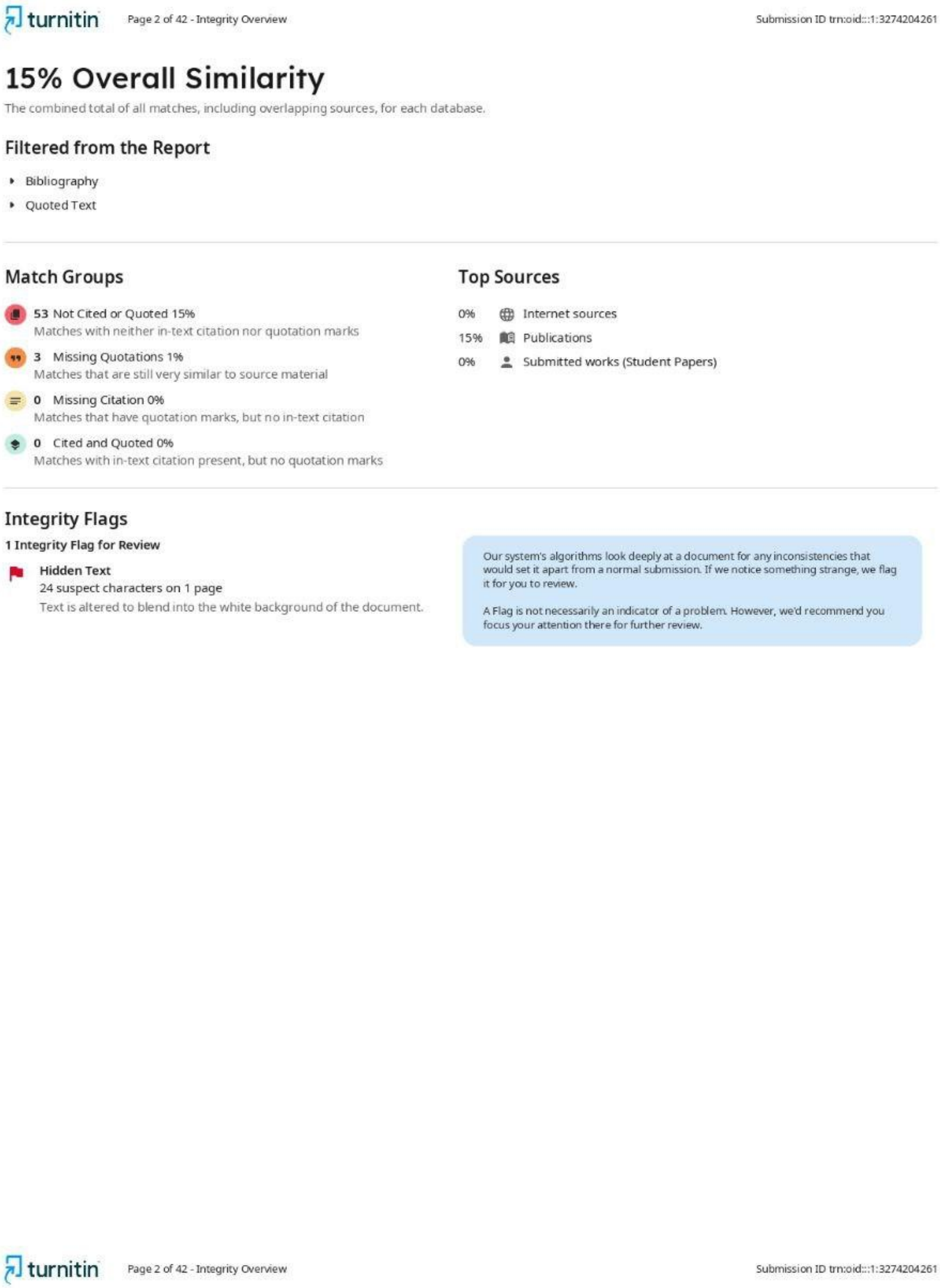
[7] C. Angsuchotmetee, P. Setthawong, and S. Udomviriyalanon, "BlockVOTE: An Architecture of a Blockchain-based Electronic Voting System," *2019*. [Online]. Available: https://ieeexplore.ieee.org/document/9183185

[8] S. K. Vivek, R. S. Yashank, Y. Prashanth, N. Yashas, and M. Namratha, "E-Voting Systems using Blockchain: An Exploratory Literature Survey," *2020*. [Online]. Available: https://ieeexplore.ieee.org/document/8974826

[9] F. D. Giraldo, B. M. C., and C. E. Gamboa, "Electronic Voting Using Blockchain And Smart Contracts: Proof Of Concept," *2020*. [Online]. Available: https://www.mdpi.com/2673-8732/4/4/21

[10] M. Sharp, L. Njilla, C.-T. Huang, and T. Geng, "Blockchain-Based E-Voting Mechanisms: A Survey and a Proposal," *2024*. [Online]. Available: https://www.mdpi.com/2673-8732/4/4/21

Plagiarism Report



turnitin Page 2 of 42 - Integrity Overview Submission ID trn:oid:::1:3274204261

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

53 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0%	Internet sources
15%	Publications
0%	Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Hidden Text
24 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 42 - Integrity Overview Submission ID trn:oid:::1:3274204261