Blockchain-Based Voting System for Academic Elections

Abstract

This document presents the design and implementation of a **blockchain-based electronic voting system** developed for academic and demonstrative purposes. The system aims to showcase the core principles of blockchain technology—immutability, transparency, and verifiability—within a simplified and understandable voting application. Unlike cryptocurrency-focused blockchains, this project emphasizes auditability and data integrity rather than mining, tokens, or smart contracts.

---

1. Introduction

Electronic voting systems face long-standing challenges related to trust, transparency, and manipulation. Traditional centralized systems require voters and auditors to trust a single authority. Blockchain technology introduces a distributed, tamper-evident data structure that can mitigate these concerns.

This project explores how blockchain concepts can be applied to an election scenario in a **clear, explainable, and academic** manner. The system is not intended for real national elections but as a proof-of-concept demonstrating how blockchain principles enhance trust in digital voting.

---

2. Design Goals

The main objectives of the system are:

- **Immutability:** Once a vote is recorded, it cannot be altered without detection.
- **Transparency:** Votes can be audited and verified by anyone.
- **Verifiability:** Each vote can be independently verified using cryptographic hashes.
- **Privacy:** Voter identities are never stored in plain form.
- **Simplicity:** The system avoids unnecessary complexity such as mining or smart contracts.

---

3. System Architecture

3.1 High-Level Overview

The application consists of:

- A **frontend web interface** for voting and auditing
- A **blockchain data structure** stored locally (demo mode)
- An **audit modal** for inspecting and verifying the chain

Each vote corresponds to a single blockchain block.

---

4. Blockchain Data Model

4.1 Block Structure

Each block in the chain has the following structure:

```
{
  "index": 1,
  "timestamp": "2025-12-27T10:03:03Z",
  "data": {
    "voterHash": "...",
    "candidateId": "left",
    "txId": "..."
  },
  "previousHash": "...",
  "hash": "...",
  "nextHash": "..."
}
```

4.2 Genesis Block

The first block in the chain is the **Genesis Block**:

- Index = 0
- Previous hash = "0"
- Contains no vote data

It serves as the immutable starting point of the chain.

---

5. Hashing and Integrity

5.1 Hash Function

The system uses **SHA-256**, a cryptographic hash function, to compute block hashes.

Each block hash is computed as:

```
SHA256(index + timestamp + data + previousHash)
```

5.2 Chain Verification

The chain is considered valid if:

- Each block's stored hash matches its recomputed hash
- Each block's `previousHash` matches the hash of the previous block

Any modification to block data immediately invalidates the chain.

---

6. Transaction ID (TxID)

Each vote generates a **Transaction ID (TxID)**:

- Unique per vote
- Stored inside block data
- Provided to the voter as a receipt

TxID enables voters to verify:

- Whether their vote exists on the chain
- Which candidate received the vote

without revealing voter identity.

---

7. Audit and Verification Features

7.1 Hash Lookup

Users can input a block hash to:

- Locate the block in the chain
- Verify its hash integrity
- Check consistency with neighboring blocks

7.2 TxID Lookup

Users can input a TxID to:

- Confirm the vote exists
- Identify the candidate
- View timestamp and block index

This mimics real-world blockchain explorers such as Etherscan or Tronscan.

---

8. Tamper Demonstration

To demonstrate immutability, the system includes a **Tamper Demo**:

- A vote block’s data is intentionally modified
- Block hash is NOT recomputed
- Chain verification fails immediately

This visually proves that:

Any unauthorized modification is detectable.

The chain can be restored using a backup mechanism.

---

9. Export and Import

The blockchain can be exported as a JSON file:

- Enables offline verification
- Demonstrates portability and auditability

Imported chains are verified before acceptance.

This feature emphasizes that:

> Trust is placed in mathematics, not in a single system.

---

10. Privacy Considerations

- Voter identifiers are hashed using SHA-256
- No personal information is stored on-chain
- TxID-based verification preserves secret ballot principles

---

11. Limitations

- Local storage (demo mode)
- No distributed consensus
- No cryptographic signatures
- Not suitable for real-world national elections

These limitations are intentional to maintain clarity and educational value.

---

12. Conclusion

This project demonstrates that blockchain concepts can be applied to electronic voting systems in a clear and auditable way. By focusing on immutability, transparency, and verifiability—while avoiding unnecessary complexity—the system serves as a strong academic example of blockchain-based trust mechanisms.

Future work may include backend integration, distributed nodes, and advanced cryptographic techniques.

Keywords

Blockchain, Electronic Voting, Immutability, Hash Chain, Auditability, Academic Project