iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82819

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Design and Implementation of a Tamper-Evident Hybrid Blockchain Voting System with Batch Processing

Dhruv Kanojia[1], Aditya Unde[2], Narayan Raut[3], Dr. Shilpa Dange[4]
Department of Computer Science, Modern College of Arts, Commerce and Science, Shivajinagar, Pune
M.Sc. Computer Science Part 2, Pune, India

Abstract: ***A fair and reliable voting system is very important in any democratic country. In the past, voting was done using paper ballots and later through Electronic Voting Machines (EVMs). Although EVMs make the process faster, many people still do not fully trust them because they work like "black boxes" and do not provide enough transparency [2], [11]. To solve this problem, this paper presents a simple and secure voting system based on a hybrid blockchain approach. The system is built using Node.js and MySQL, where votes are stored in a decentralized and tamper-resistant ledger. To improve performance and reduce the load on the database, votes are collected in batches before being added to the blockchain. Each block is connected to the previous one using cryptographic hashing, so any change in data can be easily detected [1]. This design allows online voting without forcing users to completely trust the server. The results show that the system prevents tampering and increases trust by letting users view a public record of all stored blocks.***

Index Terms: Blockchain, Online Voting, Data Protection, SHA-256, Tamper Resistance, Batch Processing, Web Security.

I. INTRODUCTION

Voting methods have changed over time to make the process both easier and more secure. For a democracy to function well, it is important that voters trust their votes are recorded correctly and counted without any external or internal interference.

A. Background

Paper ballots were widely used as the main method of voting across the world. Although they are physical and easy to verify, they can be affected by issues such as tampering, ballot stuffing, and errors during counting [4]. With the growth of technology, many countries started using Electronic Voting Machines (EVMs) or Direct Recording Electronic (DRE) systems to make the voting process faster and reduce invalid votes [3].

Moving to digital systems also brought new challenges related to security and transparency. Many electronic voting systems are centralized, and only system administrators can access or check the internal code, making them less transparent. This setup creates a single point of failure and an easy target for attackers [2]. If the central database is compromised, a large number of votes could be changed without being noticed.

At the same time, internet technologies made remote online voting (i-voting) possible. While i-voting can help increase voter participation, it still has a trust issue on the server side, where people with higher access rights may be able to change stored votes [8].

B. Problem Statement

Most modern e-voting systems still depend on centralized databases (SQL or NoSQL) that are controlled by a single authority. In this type of system, a database administrator could potentially change vote counts or edit individual records [10]. Also, traditional databases do not automatically keep a clear and verifiable history of all changes, which makes it difficult to detect any modifications made later.

This study focuses on the problems of limited transparency and lack of immutability in centralized e-voting systems. The aim is to keep the ease of web-based voting while making sure that no one — including administrators — can change the results once they are recorded in the ledger [9].

©IJRASET: All Rights are Reserved

C. Objectives and Significance

The main goal is to design a secure, transparent, and verifiable voting system using a hybrid blockchain built with Node.js. The specific objectives are:

- Secure web interface: Provide an online platform that allows users to cast their votes in a safe way.
- Vote Batching: Use a batching approach to group multiple votes together for more efficient processing before creating a block.
- Custom blockchain ledger: Develop a ledger that secures vote records using SHA-256 hashing.
- Tamper detection: Make sure that any manual change in the database breaks the cryptographic chain and can be detected immediately.

This work presents a practical model that organizations can use to conduct secure internal elections without relying on expensive external auditors. The tamper-evident design helps prevent misuse, since any change in the data would break the blockchain sequence and be easily noticeable [9].

II. LITERATURE REVIEW

To understand why blockchain is used, it is necessary to look at how voting systems have developed over time and what limitations still exist.

A. Conventional and Electronic Systems

Paper ballots are easy to verify and widely trusted, but they require a lot of manual effort and are not very eco-friendly. The counting process can also lead to human errors, and moving ballot boxes from one place to another increases the risk of tampering [4].

EVMs solved some of the problems of paper ballots by reducing invalid votes and making the counting process faster [3]. However, they are often criticized for working like “black boxes”, since voters cannot directly verify whether their vote was recorded correctly. Security studies have also pointed out weak physical security and the possibility of insider attacks [2], [3].

B. Blockchain in Voting

To reduce the problems of centralized systems, researchers have studied Distributed Ledger Technology (DLT). Blockchain is a decentralized system where data is stored in a shared and immutable way. Once data is added, it cannot be changed, which helps solve major security issues found in traditional databases [6].

In blockchain-based voting, each vote is treated as a transaction. After verification, it is added to a block and connected to the previous block using cryptographic hashing [9]. Recent approaches include:

- Smart Contracts: Voting systems built on Ethereum use smart contracts to automate the process. However, public blockchains often have high transaction costs and scalability issues, which makes them less suitable for small-scale elections [1], [5].
- Permissioned Blockchains: Systems like Hyperledger allow only approved participants to join the network, which improves privacy and control in institutional voting systems [10].
- Hybrid Models: These systems combine a normal web interface with a blockchain backend, helping to maintain both ease of use and data security [7].

C. Gaps in Existing Research

Many existing blockchain voting systems depend on complex platforms that require cryptocurrency wallets and high computing power [5]. There are only a few lightweight solutions that are suitable for small and medium-sized elections. In addition, most systems do not provide a simple real-time public ledger that allows regular users to verify that the data has not been tampered with [2]. This work tries to fill these gaps by building a lightweight permissioned blockchain using Node.js along with a batching mechanism to improve efficiency.

III. METHODOLOGY AND SYSTEM DESIGN

This study uses a hybrid approach that combines the ease of a centralized web application with the immutability of blockchain-based logging [7].

A. System Architecture

The system architecture follows a layered Model-ViewController (MVC) design, with an additional blockchain layer integrated into it.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

1) *Presentation Layer:* The front end is developed using HTML5 and Bootstrap 5. It includes separate portals for Voters and Admins, providing a responsive interface that works well on different devices.
2) *Application Layer (Node.js):* The application layer uses Node.js with Express.js to manage HTTP requests, handle user sessions, and control the voting process.
3) *Data Persistence Layer:* MySQL is used with a dualstorage setup. Standard tables store user login details and election information, while a separate blockchain table is used as an immutable log of votes. Each block is saved in serialized JSON format and connected using previous_hash and hash fields, forming a linked-list style structure inside the relational database [7].

IV. IMPLEMENTATION

The project uses a code-first approach to convert the design into a working web application.

A. Database Schema Design

The system uses a relational database called voting_db. This hybrid SQL design allows efficient querying while also maintaining a linear, blockchain-like structure for storing blocks [10].

- Users: This table stores user authentication data. Passwords are securely stored after being hashed using Bcrypt [8].
- Blockchain: This is the main ledger of the system. It stores fields such as block_index, timestamp, vote_data, previous_hash, and hash.
- User _votes: This table is used to ensure that each user can vote only once per election. It uses a composite unique key (user id, election id) to enforce the “One Person, One Vote” rule [3].

B. Vote Batching Mechanism

To improve database performance and scalability, the system does not create a new block for every single vote right away. Instead, it uses a Vote Batching mechanism.

- Vote Collection: When a user submits a vote, it is first checked for validity and then stored in a temporary memory area called the “pending pool.”
- Batch Trigger: The system keeps track of the number of pending votes. Once a set limit (for example, 5 or 10 votes) is reached, the batching process starts automatically.
- Block Creation: All votes in the batch are combined into a single data structure. This data is then hashed and saved as a new block in the blockchain.

This approach reduces the number of cryptographic operations and database writes needed, which makes the system more efficient during high-traffic elections.

C. Block Generation and Linking

Since this system is a prototype for internal elections, there is no need for a resource-heavy consensus method like Proof of Work (PoW). Instead, cryptographic linking is used to maintain data integrity.

When a batch of votes is ready to be added:

- Sanitization: All inputs are cleaned to avoid injection attacks.
- Linking: The system fetches the hash of the most recent block from the database and assigns it as the previous_hash for the new block [9].
- Hashing: The new block data (vote batch, timestamp, and previous hash) is processed using the SHA-256 algorithm to create a unique hash value.
- Storage: The generated block is then permanently stored in the blockchain table.

D. Tamper-Evidence Mechanism

A “Tampering Test” module was added to check the security of the system. This feature allows an authorized admin to manually change a stored hash in the database for testing purposes. After that, the system recalculates the hash using the stored data and compares it with the existing value. If there is any difference, it is immediately detected and the blockchain is marked as compromised [2]..

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. RESULTS AND ANALYSIS

The system was tested for both functionality and security in a local Node.js environment.

A. User Interface Results

The user interface worked well and adapted smoothly to different screen sizes. The Admin Dashboard was used for creating and handling elections.

Once an election is active, voters can securely cast their ballots. Fig. 1 shows the voting interface where the blockchain transaction begins.

B. Public Ledger and Transparency

Transparency was one of the key goals of this system. Fig. 2 shows the public ledger view, which displays the cryptographic connections (previous hash and hash) between blocks. This allows even non-technical users to verify that the blockchain has not been modified.

C. Security Validation

To test the security of the system, an administrator tried to manually modify a block's stored hash in the database. Fig. 3 shows that the system quickly detected the mismatch and immediately flagged it as tampering.

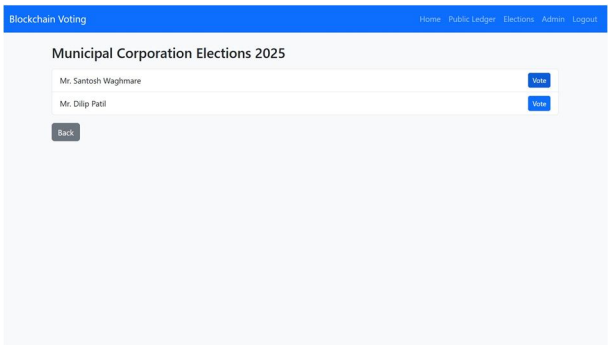


Fig. 1. Voting interface where users cast their votes, which are then processed into blockchain blocks.

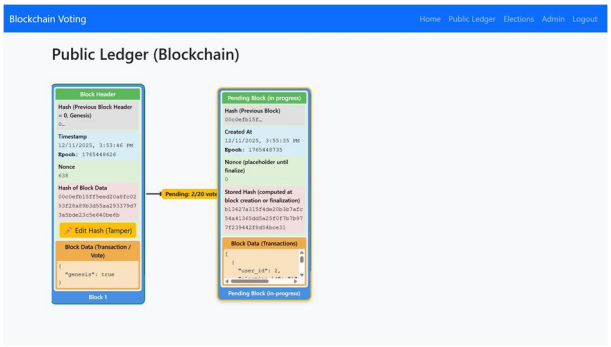


Fig. 2. Public ledger view showing the structure of the blockchain

D. Performance Evaluation

Performance testing was carried out to measure the efficiency of the batching approach:

- Processing Speed: By removing heavy consensus mechanisms like Proof of Work (PoW) and using batching, the system is able to create blocks almost instantly.
- Throughput: Combining multiple votes into batches greatly reduces the number of database write operations compared to creating one block per vote.
- Analysis: The SHA-256 hashing process takes very little time (only a few milliseconds), which helps maintain a smooth user experience even under load [7].

©IJRASET: All Rights are Reserved

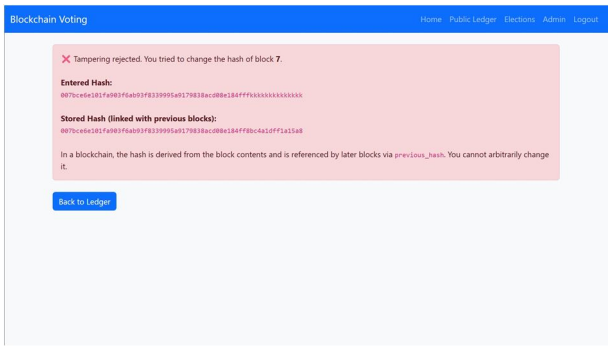


Fig. 3. System displaying an alert when a tampering attempt is detected in the database.

VI. CONCLUSION AND FUTURE WORK

This project presents a working web-based voting system that uses basic blockchain principles to reduce trust issues in electronic voting. The hybrid design using Node.js and MySQL shows that data immutability can be achieved without the complexity and high cost of public blockchain networks.

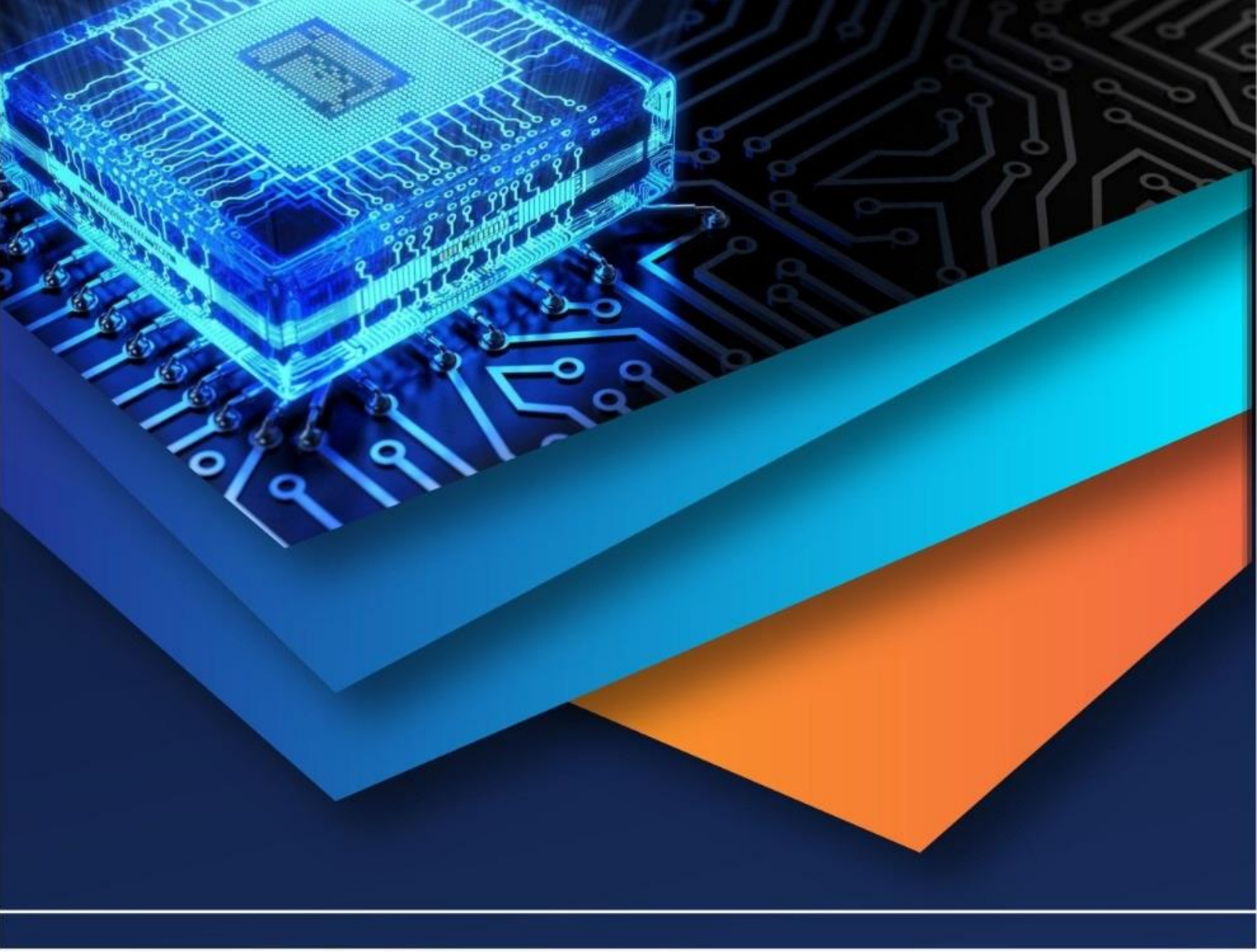
The system replaces a traditional "black box" database with a transparent, linked-chain structure [1]. The use of vote batching also improves efficiency, making the system suitable for medium-scale elections. In addition, the Public Ledger interface lets users view the blockchain directly, which helps improve overall trust in the system [4].

A. Future Work

Future improvements can focus on making the ledger more decentralized by distributing it across multiple peer-to-peer nodes, which would reduce the risk of single-point data loss [9]. Another possible enhancement is the use of Homomorphic Encryption, which would allow votes to be counted while still remaining encrypted, thereby improving voter privacy [8].

REFERENCES

[1] H. R. H. Al-Jawaheri, "Blockchain-based e-voting systems: A survey," arXiv preprint arXiv:1906.11078, 2019.
[2] K. M. Khan, J. Arshad, and M. M. Khan, "A Review of Electronic Voting Systems," International Journal of Information Engineering and Electronic Business, vol. 12, no. 1, 2020.
[3] E. Walia and A. Walia, "Analysis of Electronic Voting System in Various Countries," International Journal of Computer Applications, 2015.
[4] S. Al-Maaitah, "The Evolution of Voting: Analysis of Conventional and Electronic Voting Systems," 2021.
[5] P. Pawlak, "Blockchain-based e-voting system," PeerJ Computer Science, 2022.
[6] I. A. Khan, "Understanding Blockchain Technology," SSRN Electronic Journal, 2019.
[7] T. K. Das, "A secure electronic voting system using blockchain technology," Symmetry, 2020.
[8] M. H. Al-Adhaileh, "E-Voting System based on Blockchain Technology," International Journal of Advanced Computer Science and Applications, 2020.
[9] R. Gupta, "Blockchain Based E-Voting System," Journal of Physics: Conference Series, 2022.
[10] A. B. Smith, "Blockchain voting and its potential in election security," Procedia Computer Science, 2020.
[11] The Hindu, "Burden of proof: On India and election integrity," The Hindu, Nov. 10, 2025. Available: https://www.thehindu.com/opinion/editorial/burden-of-proof-on-india-and-election-integrity/ article70251994.ece



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)