



A Secure and Decentralized Blockchain-Based Electronic Voting Framework with Smart Contract Enforcement

Lokesh Kumar Garg¹, Sumit Kumar², Prashant Kumar Baheti³

^{1,3}Department of Computer Science Engineering, Engineering College Bharatpur, Rajasthan, India.

²Department of AI&DS, Engineering College Bharatpur, Rajasthan, India.

To Cite this Article: Lokesh Kumar Garg¹, Sumit Kumar², Prashant Kumar Baheti³, "A Secure and Decentralized Blockchain-Based Electronic Voting Framework with Smart Contract Enforcement", *Indian Journal of Computer Science and Technology*, Volume 05, Issue 01 (January-April 2026), PP: 169-179.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by-nc-nd/4.0/); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Electronic voting systems require satisfaction of security, transparency, and voter privacy to ensure fair and trustworthy election processes. Traditional centralized voting architectures suffer from limitations such as single points of failure, limited auditability, and vulnerability to data manipulation. This paper proposes a secure and decentralized electronic voting framework based on block chain technology to address these challenges. The proposed system integrates cryptographic authentication, role-based access control, smart contract automation, and distributed ledger storage to ensure tamper-resistant vote recording and transparent election management.

The architecture employs a hybrid design combining secure database management for authentication with block chain-based transaction storage for immutable vote recording. Smart contracts enforce election rules, including voter eligibility verification, single-vote constraints, and automated vote tallying. Off-chain storage mechanisms are incorporated to improve scalability while maintaining data integrity by cryptographic hashing.

Comprehensive testing, including unit, functional, integration, performance, and security evaluations, demonstrates reliable system operation and successful prevention of unauthorized access and duplicate voting attempts. Experimental results confirm that proposed framework provides secure vote handling, transparency, and auditability while preserving voter anonymity. The proposed approach offers a practical and scalable solution for next-generation decentralized electronic voting systems.

Key Words: Block chain, Electronic Voting, Decentralized Systems, E-Governance, and Privacy Preservation.

I. INTRODUCTION

Voting is a fundamental mechanism in democratic societies, enabling citizens to participate in decision-making processes and elect representatives. With rapid advancement of digital technologies, electronic voting (e-voting) systems have emerged as promising alternative to traditional paper-based voting methods due to their potential to improve accessibility, efficiency, and result processing speed. However, conventional electronic voting platforms are typically built on centralized architectures, which introduce critical challenges such as single points of failure, limited transparency, susceptibility to data manipulation, and reduced public trust in election outcomes.

Security and privacy remain primary concerns in electronic voting systems. Ensuring voter anonymity while maintaining vote integrity and verifiability is a complex requirement. Centralized databases can be vulnerable to unauthorized access, insider attacks, and data tampering, which may compromise election credibility. Also, lack of transparent auditing mechanisms in many existing systems makes independent verification difficult, thereby raising concerns regarding fairness and accountability [1].

Blockchain technology has gained attention as viable solution for addressing these challenges. As decentralized and tamper-resistant distributed ledger, blockchain enables transparent and immutable recording of transactions without depending on trusted central authority. The integration of cryptographic mechanisms, consensus protocols, and smart contracts allows automated enforcement of election rules while ensuring data integrity and auditability. These characteristics make blockchain particularly suitable for secure electronic voting applications [2].

Despite these advantages, existing blockchain-based voting solutions face several limitations, including scalability constraints, high computational overhead, complex deployment architectures, and insufficient integration of authentication and administrative control mechanisms. Some approaches emphasize advanced cryptographic privacy techniques which increase system complexity, while others focus primarily on decentralization without comprehensive validation of real-world operational requirements.

To address these challenges, this paper proposes a secure and decentralized blockchain-based electronic voting framework designed to ensure transparency, privacy preservation, and tamper-resistant vote recording. The proposed system integrates role-based access control, cryptographic authentication, and smart contract automation to enforce election policies such as voter eligibility verification, single-vote constraints, and voting period validation. A hybrid architecture combining secure off-chain database management with blockchain-based transaction storage is employed to improve scalability while maintaining security.

The proposed framework is implemented and evaluated by comprehensive testing, including unit, functional, integration, performance, and security assessments. Experimental results demonstrate reliable system behavior, secure transaction processing, and effective prevention of unauthorized access and duplicate voting attempts.

II. RELATED WORKS

Electronic voting systems have attracted research attention due to increasing demands for transparency, security, and trust in digital election processes. Blockchain technology has emerged as a solution to address limitations of traditional centralized voting systems, including single points of failure, lack of auditability, and vulnerability to manipulation. Recent studies have explored various blockchain-based architectures, cryptographic techniques, and scalability improvements to enhance electronic voting systems.

Li et al. [3] proposed S3 Voting framework, which introduces blockchain sharding to improve scalability in large-scale elections. By employing homomorphic time-lock puzzles and one-time ring signatures, the scheme ensures voter anonymity and ballot confidentiality while reducing computational overhead. Although the approach improves performance, its architectural complexity increases deployment challenges.

Zhang et al. [4] presented improved secure and efficient blockchain-assisted e-voting scheme incorporating secret sharing and identity-based ring signatures. Their system enables distributed vote counting without relying entirely on trusted third parties. However, inclusion of cloud service providers introduces partial dependency on external infrastructure.

Survey work by Ohize et al. [5] highlights recent architectural trends and challenges in blockchain-based voting systems, emphasizing issues related to scalability, infrastructure requirements, and cyber security risks. The study identifies the need for efficient consensus mechanisms and advanced cryptographic techniques to achieve practical large-scale deployment.

To strengthen voter anonymity and verifiability, Sangraula et al. [6] integrated zero-knowledge proofs with blockchain technology. Their approach employs zk-SNARK-based membership proofs and nullifiers to prevent double voting while preserving privacy. Although highly secure, zero-knowledge mechanisms introduce computational and gas costs.

Marouan et al. [7] proposed consortium blockchain-based voting platform designed for university elections using smart contracts and hybrid consensus mechanisms. The system improves execution efficiency and energy consumption; however, it primarily targets controlled institutional environments rather than large public elections.

Savaridassan et al. [8] implemented Solidity-based decentralized voting mechanism emphasizing smart contract security, encryption, and identity management.

Their simulation demonstrated improved transparency and reliability, though scalability considerations were limited. Similarly, Chafiq et al. [9] examined blockchain adoption for national elections by a multilayer distributed ledger approach, demonstrating enhanced electoral transparency but highlighting the importance of system design precision.

Li et al. [10] introduced ASEV, blockchain-based anonymous voting protocol supporting score-based voting using zero-knowledge proofs and advanced encryption schemes. While offering privacy and flexible voting models, the protocol increases computational complexity and implementation overhead.

Research Gap: Despite advancements, existing blockchain-based voting solutions still face challenges related to scalability, practical deployment complexity, authentication management, and balanced integration between off-chain and on-chain components. Many approaches either emphasize advanced cryptographic privacy at the expense of efficiency or focus on decentralization without comprehensive system validation.

Motivation of Work: To address these limitations, the proposed system introduces a secure and decentralized voting framework which integrates role-based authentication, smart contract-driven rule enforcement, and hybrid storage architecture. The design aims to achieve practical deployability while maintaining security, transparency, and efficient transaction processing validated by comprehensive system testing.

Main Contributions

The main contributions of this work are as follows:

- A decentralized blockchain-based electronic voting architecture which eliminates dependency on centralized authorities while ensuring transparency and trust.
- Integration of cryptographic authentication and role-based access control mechanisms to provide secure voter and administrator interaction.
- Design of smart contract-driven election management enabling automated enforcement of voting rules, including eligibility validation and single-vote constraints.
- A hybrid storage framework combining secure off-chain databases for authentication with immutable blockchain-based vote recording for enhanced scalability and security.
- Comprehensive system validation by unit, functional, integration, performance, and security testing, demonstrating reliable operation and resistance to unauthorized access and vote manipulation.

III. PROPOSED METHODOLOGY

This work proposes a secure, transparent, and decentralized electronic voting framework based on blockchain technology. The primary objective is to eliminate dependency on centralized authorities while ensuring vote integrity, voter anonymity, and resistance to tampering. By integrating cryptographic authentication, smart contracts, and distributed ledger mechanisms, the proposed system enables secure vote recording and verifiable election outcomes.

Unlike conventional electronic voting systems which depend on centralized databases susceptible to manipulation and

single-point failures, proposed framework distributes trust across blockchain nodes. Each voting operation is validated by decentralized consensus, ensuring immutable and publicly verifiable election records without exposing voter identities.

The workflow of proposed methodology consists of four major stages:

- Candidate Registration
- Data Preprocessing and Feature Extraction
- Secure Database Storage
- Block chain Transaction Processing

Candidate Registration

The candidate registration stage initializes election process by securely collecting candidate information, including identification details, address information, and bio-metric facial data. All collected information is encrypted prior to storage to ensure confidentiality and protection against unauthorized access.

Each candidate is assigned unique cryptographic identifier linking candidate metadata to block chain references. Once verified, registration records are anchored to block chain, ensuring immutability and auditability while preventing post-registration modification.

Data Preprocessing and Feature Extraction

The preprocessing stage prepares voter and candidate data for secure block chain integration. Raw data undergo validation, normalization, and duplicate detection to maintain consistency and eliminate fraudulent entries. Sensitive attributes are transformed using cryptographic hashing to preserve privacy while enabling secure verification.

Extracted features include voter wallet addresses, candidate identifiers, eligibility status, and election parameters. Public-private key cryptography is employed to authenticate users and digitally sign transactions, ensuring secure participation and non-repudiation.

Merkle tree structures are utilized to enable efficient integrity verification of voting records without exposing raw data. Smart contracts enforce election rules such as eligibility verification, voting time constraints, and automated vote counting. To address block chain storage limitations, large data elements are stored off-chain using IPFS, while cryptographic hash references are maintained on-chain.

Secure Database Storage

A secure auxiliary database manages encrypted authentication credentials and administrative configuration data required for system operation. Database supports login verification, election setup, candidate management, and poll configuration under access control policies.

This hybrid storage architecture combines efficiency of traditional databases with block chain immutability. Sensitive information remains encrypted at rest, improving confidentiality while maintaining system scalability and performance.

Blockchain Transaction Processing

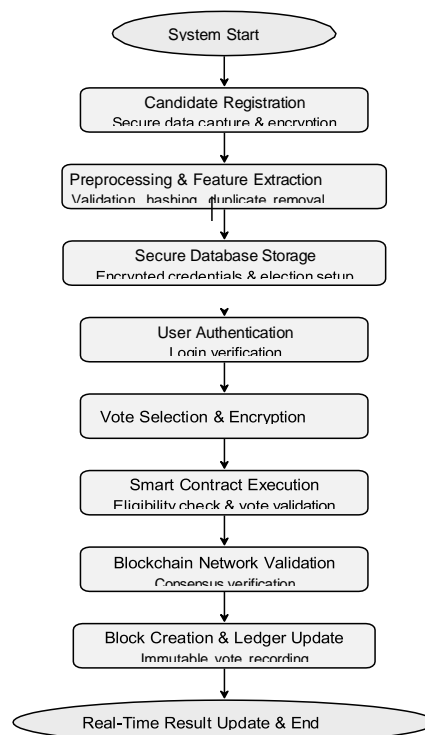


Fig. 1 Workflow of Proposed Blockchain-Based Voting Methodology

The voting phase begins when voter authenticates using secure credentials and cryptographic verification mechanisms. After authentication, voter accesses digital ballot interface and selects preferred candidate. Selected vote is encrypted and digitally signed before being converted into block chain transaction.

The transaction is broadcast to distributed block chain nodes, where consensus protocols verify transaction authenticity, voter eligibility, and compliance with election rules. Validated transactions are grouped into blocks and permanently appended to distributed ledger.

This decentralized validation process ensures immutability, transparency, and protection against tampering or double voting. Although voting records remain publicly verifiable, voter anonymity is preserved by cryptographic abstraction, ensuring privacy alongside trustworthiness.

Figure 1 illustrate multi-stage block chain voting pipeline integrating authentication, encryption, smart contract validation, and decentralized ledger recording to achieve secure and transparent electronic voting process.

IV.SYSTEM ARCHITECTURE

The proposed system architecture defines secure and decentralized blockchain-based electronic voting framework designed to ensure authentication, transparency, privacy preservation, and data integrity throughout election life cycle. The architecture integrates user authentication, role-based access control, smart contract automation, and decentralized blockchain storage to establish tamper-resistant and auditable voting environment.

The system adopts layered architecture consisting of User Interaction Layer, Application and Control Layer, and Block chain Layer. Each layer performs dedicated responsibilities to enable secure communication, controlled system operation, and reliable vote recording.

User Authentication and Access Control

The voting workflow begins when user submits login credentials comprising voter identification number and password. These credentials are verified against encrypted records stored in authentication database to ensure secure access.

Following successful verification, role-identification mechanism classifies authenticated entity as either voter or administrator. Based on classification, users are redirected to Admin Panel or Voting Interface. This role-based access control (RBAC) mechanism restricts operations according to authorization privileges and prevents unauthorized manipulation of election data.

Administrative Management Module

Administrators are responsible for election configuration and operational supervision. Authorized administrators can register candidates, define political parties, and configure election schedules, including voting start and end times.

Administrative operations generate system-level transactions which are validated by smart contracts to ensure accountability and traceability. Once election is activated, eligible voters obtain controlled access to voting interface, while administrative privileges remain restricted from altering recorded voting transactions.

Voting Interface and Secure Vote Casting

The voting interface serves as primary interaction platform for voters. After authentication, voters select preferred candidate using secure digital ballot. The selected vote is encrypted and digitally signed using cryptographic keys prior to transmission.

The encrypted vote is transformed into blockchain transaction and forwarded to smart contract layer, where election constraints such as voter eligibility, single- vote enforcement, and voting period validation are automatically verified. Automated validation minimizes manual intervention and reduces risk of electoral fraud.

Smart Contract and Blockchain Processing

Smart contracts function as core control logic of architecture by validating transactions and enforcing election rules. Verified transactions are broadcast to participating blockchain nodes for decentralized validation.

Consensus mechanisms ensure agreement among distributed nodes regarding transaction authenticity. After consensus is achieved, validated transactions are grouped into blocks and appended to blockchain ledger. This decentralized validation process satisfies immutability and prevents unauthorized modification or duplication of votes.

Decentralized Ledger and Result Update

All validated voting transactions are permanently stored within decentralized ledger, enabling transparent and verifiable election records. Administrators may monitor election progress and aggregated statistics; however, blockchain immutability prevents alteration of stored votes.

Real-time system updates notify voters of successful vote submission, improving usability and strengthening trust in electronic voting process. The integration of distributed storage and cryptographic verification ensures auditability while preserving voter anonymity.

Figure 2 illustrate layered interaction among authentication services, application control modules, smart contracts, and blockchain consensus mechanisms which enable secure and transparent electronic voting.

Security Properties of Proposed Architecture

The proposed blockchain-based voting framework is designed to satisfy fundamental security requirements of reliable electronic voting system. By integrating cryptographic primitives, decentralized ledger technology, and smart contract

enforcement, architecture ensures confidentiality, integrity, authentication, non-repudiation, transparency, and system availability throughout election life cycle.

Confidentiality:

Voter privacy is preserved by encryption and cryptographic hashing applied during data transmission and storage. Votes are encrypted prior to block chain submission, ensuring sensitive voter information and vote selections remain inaccessible to unauthorized entities. Off-chain storage mechanisms enhance privacy by maintaining only cryptographic hash references on block chain.

Data Integrity:

Voting data integrity is ensured by block chain immutability and secure hash chaining. Each transaction is cryptographically linked to previous blocks, forming append-only ledger. Any unauthorized modification produces hash mismatches detectable across distributed nodes, thereby preventing tampering or vote alteration.

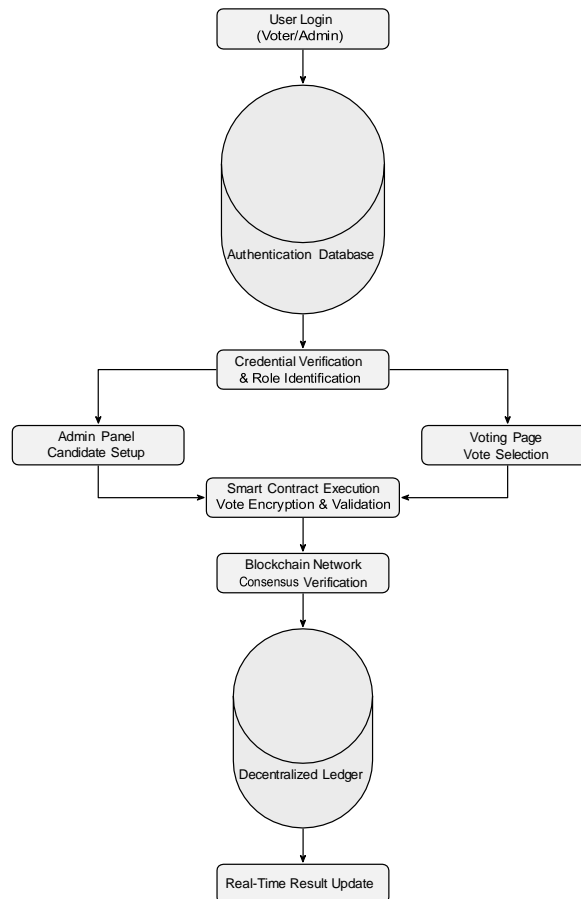


Fig. 2 System architecture of proposed blockchain-based decentralized voting platform

Authentication and Authorization: Secure authentication is achieved by credential verification combined with role-based access control (RBAC). Only verified users are granted access to permitted system functions based on assigned roles. Public-private key cryptography authenticates transaction origin and prevents impersonation or unauthorized participation.

Non-Repudiation: Digital signatures attached to blockchain transactions provide non-repudiation to ensure submitted votes cannot be denied by their originators. Smart contracts validate signed transactions before acceptance, satisfying accountability while preserving voter anonymity by cryptographic abstraction.

Transparency and Auditability: The decentralized ledger enables independent verification of election activities without revealing voter identities. Authorized entities can audit transaction histories and validate aggregated results, thereby improving trust and verifiability of election outcomes.

Availability and Fault Tolerance: The distributed architecture eliminates single points of failure present in centralized voting systems. Replicated ledger copies across multiple blockchain nodes ensure continuous availability and resilience against node failures, network disruptions, or targeted attacks.

Security properties establish resilient voting infrastructure capable of resisting manipulation, unauthorized access, and data compromise while maintaining fairness, privacy preservation, and transparency within decentralized electoral process.

V.MODULES

The proposed blockchain-based voting framework is organized into modular components to ensure scalability, security, and efficient election management. Modular decomposition enables clear separation of system responsibilities while supporting secure interaction among application services, blockchain network, and supporting database infrastructure. The system consists of two primary functional modules: Voter Module and Admin Module. Access to each module is regulated by role-based access control (RBAC) mechanisms to enforce authorization policies.

Voter Module

The Voter Module provides secure interface by which eligible users participate in election process. The module is designed to preserve voter anonymity, prevent unauthorized participation, and ensure vote integrity by cryptographic authentication and block chain validation mechanisms. These mechanisms satisfies, only verified voters can submit vote and each voter participates only once per election.

The primary functionalities of voter module include:

- **Secure Authentication:** Voters authenticate using unique credentials verified against encrypted database records, preventing unauthorized access and impersonation.
- **Candidate Information Access:** The module presents structured candidate and election information to support informed decision-making.
- **Secure Vote Casting:** Votes are submitted by encrypted digital ballot interface. Each vote is digitally signed and converted into blockchain transaction prior to transmission.
- **Vote Confirmation and Verification:** Upon successful submission, voters receive confirmation which indicate transaction is recorded on blockchain, enabling verification without revealing voting preferences.
- **Duplicate Voting Prevention:** Smart contracts enforce single-vote constraints by validating voter eligibility and checking participation records maintained on blockchain ledger.

Admin Module

The Admin Module manages election configuration and operational supervision. Administrative access is restricted to authorized users and governed by authorization policies. Although administrators control election setup and monitoring, blockchain immutability prevents modification or deletion of recorded votes.

The primary functionalities of admin module include:

- **Election Configuration:** Initialization of elections by definition of voting schedules, election parameters, and operational settings.
- **Candidate Management:** Registration, verification, approval, and association of candidates with political parties prior to election activation.
- **Election Control:** Activation and termination of voting sessions according to predefined timelines enforced by smart contracts.
- **Monitoring and Supervision:** Real-time monitoring of blockchain transactions, voter participation statistics, and election progress without altering stored voting data.
- **Policy Enforcement and System Maintenance:** Administrative oversight ensures compliance with election rules and maintains system reliability and operational continuity.

The modular architecture enhances maintainability, scalability, and security by separating voter interaction from administrative control. Integration of blockchain validation with role-based access mechanisms ensures transparency, fairness, and resistance to unauthorized manipulation within decentralized voting platform.

VI.DATA FLOW DIAGRAM

Level 0 Data Flow Diagram

The Level 0 Data Flow Diagram (DFD), shown in Fig. 3, illustrate high-level representation of data exchange between external users and proposed blockchain-based voting system. At abstraction level, entire voting platform is modeled as single processing entity, emphasizing primary data interactions while omitting internal processing details.

The process begins when voter submits voting information by user interface. The received data are forwarded to decentralized voting system, where eligibility verification, authentication, and encryption procedures are performed to ensure secure participation. The system validates voter credentials and checks prior participation records to prevent duplicate voting.

Following successful validation, encrypted vote is transformed into blockchain transaction and transmitted to blockchain network. The transaction is verified by consensus mechanisms and permanently recorded in distributed ledger, ensuring immutability and protection against unauthorized modification.

After confirmation from blockchain network, acknowledgment message is returned to voter indicating successful vote submission. In addition, aggregated election results can be retrieved from blockchain and presented in real time without exposing individual voter identities.

The Level 0 DFD highlights secure data exchange, decentralized transaction validation, and transparent record management enabled by blockchain integration.

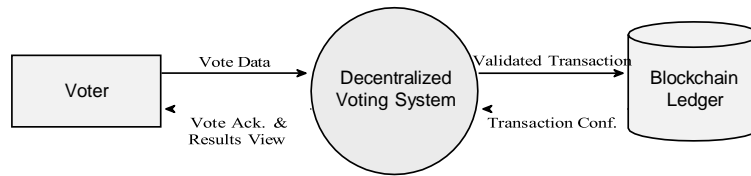


Fig. 3 Level 0 Data Flow Diagram of blockchain-based decentralized voting system

Level 1 Data Flow Diagram

The Level 1 Data Flow Diagram (DFD), shown in Fig. 4, presents detailed decomposition of decentralized voting system introduced in Level 0 diagram. It illustrate internal functional processes responsible for authentication, vote casting, and blockchain transaction management.

The workflow begins when voter submits login credentials by login interface. The authentication module verifies these credentials by interacting with user database containing securely stored records. Upon successful verification, voter is granted authorized access to voting interface.

Within voting process, voter reviews candidate information and submits selected vote by secure digital ballot. The voting data are encrypted and forwarded to transaction processing module, where vote is converted into blockchain transaction.

The generated transaction is transmitted to Ethereum blockchain network for validation by consensus mechanisms. Once validated, transaction is permanently recorded in distributed ledger, ensuring immutability and protection against unauthorized modification.

Following confirmation from blockchain network, system updates voter participation status to prevent duplicate voting attempts. A vote acknowledgment message is then returned to voter, confirming successful submission without revealing voting preferences.

The Level 1 DFD demonstrates internal data processing workflow, highlighting secure authentication, controlled vote handling, and decentralized transaction recording enabled by blockchain integration.

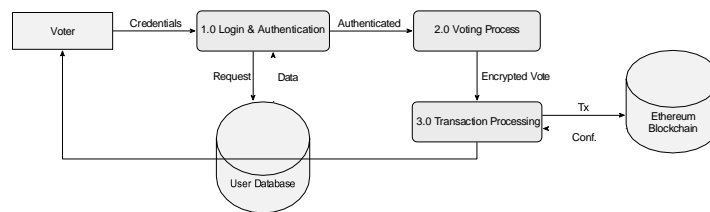


Fig. 4 Level 1 Data Flow Diagram of blockchain-based voting system

Level 2 Data Flow Diagram

The Level 2 Data Flow Diagram (DFD), shown in Fig. 5, provides detailed decomposition of internal processes associated with authentication, role management, and blockchain transaction handling within proposed decentralized voting system. This level expands functional modules introduced in Level 1 DFD by illustrating fine-grained data interactions among system components.

The workflow begins when user submits login credentials by login interface. The credentials are forwarded to database application programming interface (API), which communicates with user database to perform authentication and retrieve role information. After successful verification, role identification process determines whether authenticated entity is voter or administrator.

For authenticated voters, access is granted to voter interface, where candidate information is displayed and vote selection is performed. The selected vote under- goes encryption and is forwarded to transaction handling module, which converts voting data into blockchain transaction. The transaction is transmitted to Ethereum blockchain network for consensus validation and permanent storage in distributed ledger. Following confirmation, system updates voter participation status to prevent duplicate voting and generates acknowledgment response.

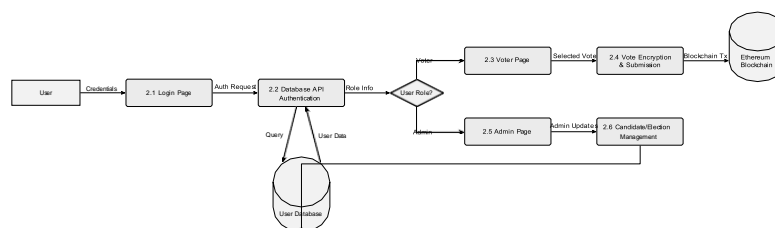


Fig. 5 Level 2 Data Flow Diagram showing detailed role-based operations in blockchain voting system

For administrative users, requests are directed to admin interface, where operations such as candidate registration and election configuration are executed. Administrative updates are securely stored in database by controlled access mechanisms while maintaining separation from blockchain-recorded voting transactions.

The Level 2 DFD illustrate detailed data exchanges between authentication services, role-based processing modules, database services, and blockchain infrastructure, demonstrating secure access control and decentralized vote recording within proposed architecture.

VII.ER DIAGRAM

Figure 6 presents Entity–Relationship (ER) diagram describing database schema used for authentication and election management in proposed blockchain-based voting system. ER model defines structural relationships among system entities required for secure user access, role-based authorization, and election configuration.

The schema consists of three primary entities: Voter, Candidate, and Election. These entities support authentication operations and election management, while voting transactions themselves are recorded on blockchain to ensure immutability.

Voter Entity: The Voter entity stores authentication-related information required for secure system access. It includes attributes voter id, role, and password. The voter id acts as primary key, uniquely identifying each registered user and preventing duplication. The role attribute defines access privileges (e.g., voter or administrator), enabling role-based access control within system. User passwords are stored in hashed or encrypted form to enhance protection against credential-based attacks.

Candidate Entity: The Candidate entity maintains information related to election participants. It includes attributes such as candidate id (primary key), candidate name, and affiliated political party. This entity allows structured storage and retrieval of candidate data presented to voters during voting process.

Election Entity: The Election entity represents election-specific configuration parameters, including election id and election scheduling information. This entity enables administrators to manage multiple election instances while maintaining consistent data organization.

Relationship Modeling: The relationship Casts Vote connects Voter, Candidate, and Election entities, representing voter participation within specific election context. Logical constraints enforced by application logic and smart contracts ensure, each voter can cast only one vote per election, preventing duplicate participation.

The ER model separates authentication and administrative data stored in database from voting transactions maintained on blockchain ledger. This hybrid design improves system efficiency while preserving security, auditability, and data integrity within decentralized voting framework.

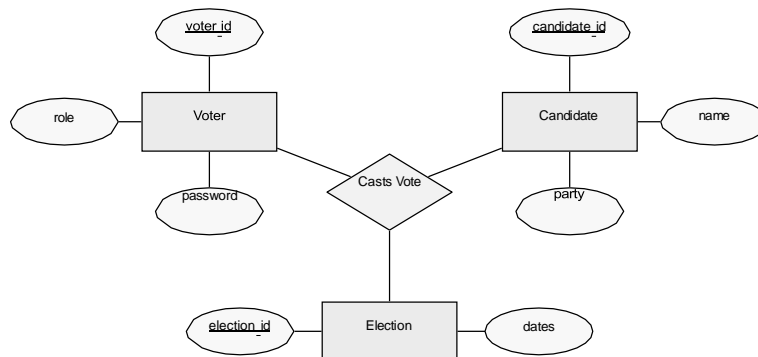


Fig. 6 Entity–Relationship diagram of blockchain-based voting system

VIII.USE CASE DIAGRAM

Figure 7 illustrate Use Case Diagram of proposed decentralized blockchain-based voting system. The diagram represents functional requirements from user interaction perspective and identifies how system actors access core services within defined system boundary.

The system includes two primary actors: Voter and Admin. Both actors interact with system by common Login use case, which performs authentication and role verification prior to granting access to system functionalities.

After successful authentication, Voter actor is authorized to access voting-related operations. The voter can view candidate information and participate in election by Cast Vote use case. During process, selected vote is securely encrypted and recorded as blockchain transaction, ensuring integrity, transparency, and voter anonymity.

The Admin actor is granted privileged access to administrative functionalities required for election management. These include candidate registration and election configuration by Add Candidate and related administrative operations. Administrative actions are restricted to system configuration tasks and do not permit modification of votes already stored on blockchain ledger.

The Use Case Diagram demonstrates role-based access control by clearly separating voter participation functions from administrative management operations. This separation ensures controlled system interaction while maintaining security, transparency, and operational integrity within decentralized voting platform.

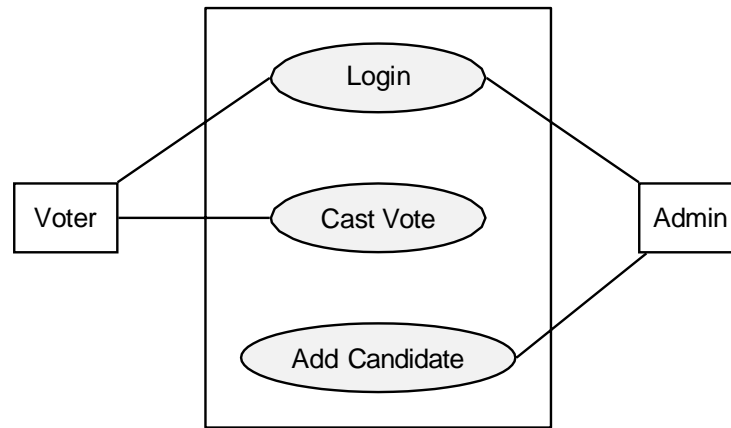


Fig. 7 Use case diagram of decentralized blockchain-based voting system

IX. TESTING AND RESULTS

The testing phase evaluates ability of proposed blockchain-based voting system to satisfy functional correctness, security requirements, and operational performance objectives. The evaluation aims to validate system reliability, ensure secure vote processing, and confirm correct interaction among authentication services, application modules, and blockchain components. Considering security-critical nature of electronic voting systems, comprehensive validation was performed to ensure data integrity, voter privacy, and accurate vote recording.

Testing activities were conducted across authentication, voting operations, administrative management, and blockchain transaction processing modules. Multiple testing strategies were employed to assess system behavior under different operational conditions are as follows.

- **Unit Testing:** Individual software components and smart contract functions were tested independently to verify logical correctness. Key operations, including voter authentication, candidate registration, vote casting, and result recording, were validated using automated test scripts executed within Truffle and Hardhat development environments. All tested contract functions produced correct outputs under predefined input conditions.
- **Integration Testing:** Integration testing evaluated interoperability between front-end application and blockchain smart contracts using Web3.js and MetaMask. The tests confirmed reliable end-to-end interaction during authentication, vote submission, transaction confirmation, and result retrieval processes.
- **Performance Testing:** System scalability was assessed by simulating concurrent voting requests. Performance evaluation focused on transaction latency and throughput under increased workload conditions. Experimental observations indicated stable transaction processing with acceptable response times during peak voting scenarios, demonstrating practicality of decentralized vote handling.
- **Security Testing:** Security testing examined system resilience against unauthorized access, replay attacks, and duplicate voting attempts. Smart contract validation mechanisms successfully enforced single-vote constraints, while blockchain immutability prevented modification of recorded transactions after confirmation.
- **Functional Testing:** Functional validation ensured correct enforcement of election rules, including voter eligibility verification, voting period constraints, and accurate vote tallying. The system consistently generated results which matched blockchain-recorded transactions, confirming functional correctness.

Experimental results demonstrate proposed framework provides reliable operation, secure vote management, and transparent transaction recording while preserving voter privacy and resistance to tampering.

Test Results

Table 1 presents results of unit test performed to validate JSON Web Token (JWT) authorization mechanism implemented in proposed voting system. The purpose of test was to verify unauthorized users are prevented from accessing restricted system functionalities.

During evaluation, login credentials were supplied to authentication module to determine whether access control policies were correctly enforced. The expected behavior was denial of access when valid authorization tokens were absent. Experimental observations confirm unauthorized users were unable to access protected resources, including voting and administrative interfaces.

The successful completion of test confirms correct implementation of authentication and authorization mechanism and demonstrates effective enforcement of role-based access control within system.

Test Case No.	1
Test Type	Unit Testing
Test Name	JWT Authorization Validation
Description	Verification of JWT-based authentication preventing unauthorized system access.
Inputs	Login credentials (username and password)

Expected Output	Unauthorized users are denied access to protected pages.
Actual Output	Access to voting and admin pages was denied without valid authorization.
Results	Pass
Remarks	Authorization mechanism operates as intended.

Table 1 Unit test results for JWT-based authorization mechanism

In Table 2, Test Case 2 presents functional test performed to validate login functionality of proposed voting portal. The test evaluates whether users with valid credentials can successfully authenticate while unauthorized access attempts are rejected.

During evaluation, Voter ID and password were provided as input to authentication module. The expected behavior was access would be granted only when submitted credentials matched records stored in database, whereas invalid credentials would generate unauthorized access response. Experimental observations confirms authentication was successful exclusively for valid users. The test case was therefore marked as passed, demonstrating correct implementation of login authentication process.

Test Case No.	2
Test Type	Functional Testing
Test Name	User Login Verification
Description	Validation of login functionality ensuring access only with valid credentials.
Inputs	Voter ID and password
Expected Output	Access granted for valid credentials; unauthorized error for invalid credentials.
Actual Output	Login successful only with correct credentials.
Results	Pass
Remarks	Authentication module operates as intended.

Table 2 Functional test results for user login authentication in voting portal

In Table 3, Test Case 3 presents unit test performed to validate candidate registration functionality of proposed voting system. The objective of test is to verify authorized administrators can successfully register candidates within system.

During evaluation, candidate name and political party information were provided as input by administrative interface. The expected outcome was successful execution of registration transaction, confirming correct processing and storage of candidate details. Experimental observations confirms registration transaction completed successfully without errors. The test case was therefore marked as passed, demonstrating correct implementation of candidate registration module.

Test Case No.	3
Test Type	Unit Testing
Test Name	Candidate Registration Verification
Description	Validation of administrator-controlled candidate registration process.
Inputs	Candidate name and political party
Expected Output	Successful candidate registration transaction.
Actual Output	Candidate registration transaction completed successfully.
Results	Pass
Remarks	Candidate registration module operates as intended.

Table 3 Unit test results for candidate registration functionality

In Table 4, Test Case 4 presents functional test performed to validate voting process within proposed blockchain-based voting system. The objective of test is to verify where authenticated voters can successfully cast votes and each vote is correctly recorded as blockchain transaction.

During evaluation, voter selected candidate by voting interface and initiated process by clicking Vote button. The expected outcome was successful execution of voting transaction, confirming selected vote was securely processed and stored on blockchain ledger. Experimental observations confirms transaction completed successfully with- out errors. The test case was therefore marked as passed, demonstrating correct implementation of voting functionality.

Test Case No.	4
Test Type	Functional Testing
Test Name	Vote Casting Verification
Description	Validation of voter ability to cast vote and record transaction on blockchain.
Inputs	Candidate selection followed by clicking Vote button

Expected Output	Successful vote transaction recorded on blockchain.
Actual Output	Vote transaction completed successfully and recorded.
Results	Pass
Remarks	Voting functionality operates as intended.

Table 4 Functional test results for vote casting and blockchain transaction recording

X.CONCLUSION

This paper presented secure and decentralized blockchain-based electronic voting system designed to enhance transparency, integrity, and voter privacy in digital election processes. By leveraging blockchain technology, smart contracts, and cryptographic authentication mechanisms, proposed framework eliminates dependence on centralized authorities while ensuring tamper-resistant vote recording and verifiable election outcomes.

The system architecture integrates role-based access control, secure authentication, and automated smart contract validation to enforce election rules and prevent unauthorized participation or duplicate voting. A hybrid storage model combining traditional databases with blockchain-based ledgers improves scalability while maintaining security. Experimental evaluation by unit, functional, integration, performance, and security testing confirmed reliable system behavior, secure transaction processing, and correct enforcement of voting policies.

The obtained results demonstrates proposed framework provides practical and trustworthy solution for decentralized electronic voting. Future work will focus on improving scalability for large-scale national elections, optimizing transaction latency, and exploring advanced privacy-preserving techniques such as zero-knowledge proofs and layer-2 blockchain solutions to enhance efficiency and voter anonymity.

References

1. Naik, A.C., Prajapati, A.M., Pandey, S.N., Mishra, A.C.: Blockchain based e- voting system. In: 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 316–320 (2023). <https://doi.org/10.1109/ICOEI56765.2023.10125883>. IEEE
2. Singathala, H., Narayansetty, S., Kata, H.: Blockchain based e-voting system. In: 2024 Second International Conference on Emerging Trends in Information Tech- nology and Engineering (ICETITE), pp. 1–6 (2024). <https://doi.org/10.1109/ic-ETITE58242.2024.10493789>. IEEE
3. Li, M., Xue, K., Luo, X., Sun, W., Wei, D.S., Sun, Q., Lu, J.: s3 voting: A blockchain sharding based e-voting approach with security and scalability. *IEEE Transactions on Dependable and Secure Computing* 22(2), 1596–1611 (2024) <https://doi.org/10.1109/TDSC.2024.3446392>
4. Zhang, J., Wu, C., Sherratt, R.S., Wang, J.: An improved secure and efficient e-voting scheme based on blockchain systems. *IEEE Internet of Things Journal* 12(7), 8626–8637 (2024) <https://doi.org/10.1109/IIOT.2024.3507366>
5. Ohize, H.O., Onumanyi, A.J., Umar, B.U., Ajao, L.A., Isah, R.O., Dogo, E.M., Nuhu, B.K., Olaniyi, O.M., Ambafi, J.G., Sheidu, V.B., et al.: Blockchain for securing electronic voting systems: a survey of architectures, trends, solutions, and challenges. *Cluster Computing* 28(2), 132 (2025) <https://doi.org/10.1007/s10586-024-04709-8>
6. Sangraula, P., Adhikari, N.B.: Zero knowledge proof on top of blockchain for anonymous and verifiable e-voting system. *Journal of The Institution of Engineers (India): Series B*, 1–12 (2025) <https://doi.org/10.1007/s40031-025-01198-0>
7. Marouan, A., Badrani, M., Zannou, A., Kannouf, N., Chetouani, A.: E-voting sys- tem based on blockchain for enhanced university elections. *SN Computer Science* 6(3), 204 (2025) <https://doi.org/10.1007/s42979-025-03671-5>
8. Savaridassan, P., Lohani, S., Gupta, H.: A block-chain based decentralized mechanism to ensure the security of electronic voting system using solidity language. In: *International Conference on Deep Sciences for Computing and Communications*, pp. 44–51 (2023). https://doi.org/10.1007/978-3-031-68905-5_6. Springer
9. Chafiq, T., Azmi, R., Mohammed, O.: Blockchain-based electronic voting systems: A case study in morocco. *International Journal of Intelligent Networks* 5, 38–48 (2024) <https://doi.org/10.1016/j.ijin.2024.01.004>
10. Li, F., Wang, X., Chen, T., Li, L., Huang, H.: Asev: Anonymous and scored- based e-voting protocol on blockchain. In: *International Conference on Frontiers in Cyber Security*, pp. 309–322 (2023). https://doi.org/10.1007/978-981-99-9331-4_21. Springer