IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DECENTRALIZED ONLINE VOTING SYSTEM USING ETHEREUM BLOCKCHAIN

[1]Kanchan Patil, [2]Ariyan Pujari, [3]Ramesh Taradi, [4]Ravikiran Dalawai, [5]Akash Vaggannavar

[1]Assistant Professor, Maratha Mandal's Engineering College, Belagavi, Karnataka, India

[2]UG Scholar , Maratha Mandal's Engineering College, Belagavi, Karnataka, India

[3] UG Scholar, Maratha Mandal's Engineering College, Belagavi, Karnataka, India

[4]UG Scholar, Maratha Mandal's Engineering College, Belagavi, Karnataka, India

[5] UG Scholar, Maratha Mandal's Engineering College, Belagavi, Karnataka, India

Abstract:

Electronic voting systems are widely used, but many of them still suffer from problems related to transparency, data integrity, and public trust. These issues are more common in systems where vote storage and verification are fully controlled by centralized authorities, leaving voters with no clear way to verify the process. To address this problem, this work presents a decentralized online voting system built on the Ethereum blockchain, where voting logic is handled through a smart contract rather than a central server.

The system uses MetaMask for voter authentication by verifying ownership of a blockchain address, while Web3.js is responsible for sending vote transactions to the blockchain. Node.js with Express is used to manage backend operations, and MySQL is included only for storing structured off-chain data that is not practical to keep on the blockchain. Voter registration is carried out using a one-time password sent to a mobile number, ensuring that each verified user can vote only once. Once a vote is recorded on the blockchain, it cannot be changed or submitted again.

The complete system was tested on a Ganache local blockchain to evaluate transaction behavior and system reliability. The results show that votes are processed correctly, duplicate voting is prevented, private keys remain secure, and vote counts are available immediately after the polling period ends. The study shows that using blockchain for online voting can improve trust and transparency while keeping the system simple enough for practical use.

Index Terms — Blockchain, Ethereum, Smart Contract, MetaMask, Decentralized Voting, Web3.js, Security, OTP Authentication.

I. INTRODUCTION

Electronic voting has been studied for many years as an alternative to paper-based elections. However, in most existing implementations, the system is still controlled by centralized servers. This centralized design creates several risks, such as data manipulation, unauthorized access, and complete system failure if a single component is compromised. As more public services move online and digital participation increases, these weaknesses have become harder to ignore. There is now a clear need for voting systems that people can trust, inspect, and verify rather than simply accept.

Blockchain technology offers a different approach by removing reliance on a single controlling authority. In a decentralized voting system, trust is distributed across a peer-to-peer network instead of being placed in one organization or server. Each vote recorded on the blockchain is protected using cryptographic techniques, time-stamped, and stored permanently. Once added, the vote cannot be altered or removed, which helps preserve the integrity of the election. At the same time, the transparent nature of the blockchain allows voters and auditors to verify results independently without exposing voter identities.

Based on these ideas, this work proposes a blockchain-based online voting platform built on the Ethereum network. The voting process is governed by smart contracts that automatically enforce election rules, removing the need for manual intervention. MetaMask is used to authenticate voters by confirming ownership of a blockchain address, while Web3.js connects the web application to the blockchain. Backend operations are handled using Node.js and Express, including OTP-based voter registration, session handling, and communication with a MySQL database that stores supporting voter information not kept on-chain.

The main objective of this study is to design a voting system that improves security without making the process complex for users. By combining decentralized execution, cryptographic signatures, and a simple user interface, the system aims to provide a voting experience that is both secure and practical. Experiments conducted on a Ganache test blockchain show that the decentralized approach offers clear improvements in transparency, integrity, and trust when compared to conventional electronic voting systems, making it suitable for use in academic institutions, organizations, and similar election environments.

II. LITERATURE REVIEW

The growth of digital governance has encouraged many researchers to study how elections can be conducted securely in online environments. While traditional electronic voting systems were introduced to replace paper-based elections, most of them still rely on centralized servers. This dependence has raised serious concerns, as centralized systems are more exposed to risks such as data manipulation, unauthorized access, and failure of a single controlling component. As these weaknesses became more visible over time, blockchain technology began to attract attention as a possible solution due to its decentralized and transparent nature.

Several studies have explored the use of blockchain for voting applications. Early research mainly focused on the immutability of blockchain records, showing that once votes are stored on the ledger, they cannot be altered or removed. This property significantly reduces the chances of election fraud. Researchers also highlighted the role of smart contracts in automating election processes such as candidate registration, vote validation, and result counting. By reducing human involvement in these critical steps, smart contracts help minimize bias and improve the overall reliability of the voting process.

Other works proposed end-to-end verifiable voting systems implemented using Ethereum smart contracts. These studies emphasized that decentralized consensus removes the need to place trust in a single administrator. Votes stored as blockchain transactions remain permanently available for verification, allowing auditors to review election results even after long periods. However, these systems also revealed practical issues, including high transaction costs, limited scalability, and difficulties faced by users when interacting directly with blockchain tools.

To address these challenges, recent research has shifted attention toward improving usability and system efficiency. Many studies adopted browser-based wallets such as MetaMask to simplify voter authentication and transaction signing. This approach allows users to cast votes securely without directly handling private keys. Some researchers also explored hybrid system designs, where only essential voting data is stored on the blockchain, while user details and session-related information are managed off-chain using traditional databases. This design choice helps reduce cost and improve system performance while maintaining security.

In addition to system architecture, voter authentication has been an important topic in the literature. Various methods such as OTP-based verification, biometric authentication, and mobile registration have been discussed. Among these, OTP-based verification is often preferred due to its simplicity and ease of deployment. Existing studies indicate that OTP mechanisms can effectively reduce duplicate registrations and unauthorized access, especially when combined with cryptographic authentication methods.

Despite continuous improvements, prior research also points out several limitations. Many proposed systems remain at the prototype level and do not fully address real-world requirements such as session control,

administrative management, or user-friendly interfaces. Some studies also lack consideration of how blockchain-based voting can be integrated with existing legal and organizational election processes.

Overall, the literature suggests that an effective blockchain voting system should combine decentralized vote storage, secure authentication, transparent session handling, tamper-resistant result management, and an interface that is accessible to non-technical users. These observations form the foundation for the system proposed in this work, which integrates Ethereum smart contracts, MetaMask-based authentication, OTP verification, and a structured backend design to provide a secure and practical digital voting solution.

III. PROBLEM STATEMENT

Conventional electronic voting systems continue to struggle with issues related to security, transparency, and public trust. Most current e-voting platforms are built on centralized architectures, where a single authority is responsible for storing, verifying, and counting votes. This design introduces several risks, including data manipulation, unauthorized access, database tampering, server outages, and even insider attacks. In addition, many of these systems do not provide end-to-end verifiability, which means voters and auditors have no direct way to confirm whether votes were recorded and counted correctly.

Another major concern is voter authentication. Many existing e-voting solutions rely on basic login methods such as usernames and passwords, which are not sufficient for secure elections. Weak authentication increases the chances of impersonation, duplicate registrations, and unauthorized voting. Without a reliable and tamper-resistant identity verification process, it becomes difficult to ensure that each voter is legitimate and that only one vote is cast per person.

Practical deployment challenges further limit the effectiveness of earlier research prototypes. Several proposed systems focus mainly on technical concepts while ignoring real-world requirements. Features such as session control, prevention of double voting, real-time result availability, and user-friendly interfaces are often missing. As a result, many solutions remain difficult to scale and unsuitable for use by non-technical voters or election administrators.

These limitations highlight the need for a voting system that is secure, transparent, tamper-proof, and easy to use. Such a system should remove dependence on centralized control, provide strong voter authentication, prevent multiple voting, and ensure that all votes are stored in a permanent and verifiable manner. This research addresses these challenges by designing and implementing a complete blockchain-based voting platform using Ethereum smart contracts, MetaMask for authentication, Web3.js for blockchain interaction, Node.js for backend processing, and MySQL for managing supporting off-chain data.

IV. OBJECTIVES OF THE STUDY

The main objective of this research is to design and implement a secure, transparent, and decentralized online voting system using blockchain technology. The goal is to overcome the limitations of traditional and centralized e-voting systems by using smart contracts, cryptographic techniques, and a distributed ledger approach. The specific objectives of this study are outlined below.

1. **To design a decentralized voting architecture using the Ethereum blockchain**
   The system aims to store votes on an immutable blockchain ledger so that election data remains transparent, tamper-resistant, and verifiable by authorized parties.
2. **To implement smart contracts for automated election operations**
   Smart contracts are used to manage core election activities such as candidate registration, vote casting, session control, and result calculation without relying on manual intervention.
3. **To provide secure voter authentication using OTP verification**
   The system verifies voter identity through mobile-based OTP validation in order to reduce fake registrations, duplicate accounts, and unauthorized voting attempts.
4. **To enforce one-voter-one-vote using blockchain address tracking**
   Each voter is linked to a unique blockchain address, and smart-contract restrictions ensure that the same address cannot be used to cast multiple votes in a single election.

5. **To develop a simple and user-friendly interface for voters and administrators**
   The objective is to create an easy-to-use web interface with seamless MetaMask integration, allowing both voters and administrators to interact with the system without technical complexity.
6. **To use a hybrid on-chain and off-chain storage approach**
   Critical voting data is stored on the blockchain for security, while voter profiles, session information, and historical records are maintained in a MySQL database to improve efficiency and scalability.
7. **To evaluate system performance, security, and reliability**
   The system is tested using a Ganache environment to assess transaction handling, security enforcement, and the accuracy of vote counting under controlled conditions.

V. EXISTING SYSTEM VS PROPOSED SYSTEM

A. Existing System

Most traditional electronic voting systems, including web-based and kiosk-based solutions, are built on centralized architectures. In these systems, all voting data is stored and processed within a single server or a tightly controlled administrative environment. While such platforms offer convenience compared to paper-based elections, they also introduce several serious limitations that affect security, transparency, and trust.

1. **Centralized Data Storage**
   All votes, voter details, and election results are typically stored in a central database. This creates a single point of failure, where any system crash, internal misuse, or external attack can compromise the entire election process. Central storage also increases the risk of data corruption and unauthorized modifications.
2. **Lack of Transparency**
   In centralized voting systems, the internal operations are not visible to voters or independent observers. As a result, users must fully trust the authorities managing the system. There is usually no direct way for voters to verify whether their vote has been recorded or counted correctly.
3. **Vulnerability to Tampering and Cyber Attacks**
   Central servers are common targets for cyber attacks such as SQL injection, DDoS attacks, malware infections, and database manipulation. Unauthorized administrative access can also lead to intentional or accidental changes in voting data, which can affect election outcomes.
4. **Weak Voter Authentication**
   Many existing platforms rely on basic authentication methods such as usernames and passwords. These mechanisms are often insufficient for secure voting and can result in impersonation, duplicate registrations, and multiple votes cast by the same individual.
5. **Manual Election Management**
   Election-related tasks such as creating voting sessions, counting votes, and declaring results often require manual involvement by administrators. This not only slows down the process but also increases the possibility of human errors or biased handling.
6. **Limited Auditability**
   After an election is completed, centralized systems may not maintain a permanent and tamper-proof record of votes. Without an immutable audit trail, it becomes difficult to verify the integrity of the election data or resolve disputes at a later stage.

B. Proposed System

The proposed system introduces a decentralized, blockchain-based voting solution designed to overcome the limitations of centralized e-voting platforms. Instead of relying on a single authority, the system uses Ethereum smart contracts, distributed vote storage, OTP-based identity verification, and a hybrid on-chain and off-chain architecture to improve security, transparency, and reliability.

1. **Decentralized Vote Storage on the Blockchain**
   In the proposed model, votes are recorded directly on the Ethereum blockchain. Once a vote is submitted and confirmed, it becomes part of an immutable ledger that cannot be altered or deleted. This removes the possibility of vote manipulation by any central authority and ensures transparency throughout the election process.
2. **Smart Contract–Based Voting Logic**
   All election rules are implemented through smart contracts. Operations such as candidate registration, vote validation, session control, and vote counting are handled automatically by the contract itself. This removes the need for manual intervention and reduces the risk of bias or human error during election management.
3. **Secure Voter Authentication Using OTP**
   To ensure that only legitimate users participate in the election, voter identity is verified through mobile-based OTP authentication. This step helps prevent fake registrations and ensures that each voter is authenticated before being allowed to vote.
4. **Enforcement of One-Voter-One-Vote**
   The smart contract keeps track of blockchain addresses that have already voted in a particular session. Once an address is used, it is blocked from voting again in the same election. This guarantees strict enforcement of the one-voter-one-vote rule.
5. **MetaMask-Based Wallet Authentication**
   Instead of traditional passwords, the system uses MetaMask for authentication and transaction signing. Voters sign their vote using their wallet, and private keys remain securely stored within MetaMask at all times. This approach strengthens security and removes risks associated with password-based systems.
6. **Transparent and Verifiable Vote Counting**
   Since all votes are recorded as blockchain transactions, the counting process is fully transparent. Any authorized observer or auditor can independently verify the results by checking the blockchain, ensuring end-to-end verifiability without relying on administrators.
7. **Hybrid On-Chain and Off-Chain Storage**
   To maintain efficiency and scalability, only critical voting data is stored on the blockchain. Supporting information such as voter profiles, session records, and system logs are stored off-chain in a MySQL database. This hybrid approach balances security with performance.
8. **User-Friendly Interface**
   The system provides a simple and intuitive web interface for both voters and administrators. Voters can easily view elections and cast their votes, while administrators can create sessions, manage candidates, and monitor results without technical complexity.

VI. RESEARCH METHODOLOGY / SYSTEM DESIGN

The proposed decentralized voting system follows a step-by-step research methodology that combines secure authentication, blockchain-based vote recording, smart contract logic, and a hybrid data storage approach. The overall design focuses on transparency, tamper resistance, and system reliability. The methodology used in this study is divided into multiple phases, starting from requirement analysis and ending with testing and validation.

1.Requirement Analysis

The first phase involved identifying the major issues present in traditional electronic voting systems. These issues included centralized control, weak voter authentication, lack of transparency, and the possibility of vote tampering. Based on this analysis, system requirements were grouped into two categories.

Functional requirements included voter registration, OTP verification, vote casting, session handling, result calculation, and interaction with the blockchain. Non-functional requirements focused on system security, transparency, usability, immutability of votes, scalability, and overall performance.

2. System Study and Technology Selection

A detailed study of available blockchain platforms was carried out to select the most suitable technology for implementation. Ethereum was chosen because of its mature smart contract support, strong compatibility with Web3-based applications, active developer community, and availability of a predictable testing environment through Ganache. MetaMask was selected for wallet-based authentication, while Node.js and MySQL were chosen to handle backend operations and off-chain data storage efficiently.

3. Architectural Design

The system is designed using a hybrid, multi-layer architecture. The frontend layer consists of HTML, CSS, JavaScript, and MetaMask for user interaction. The backend layer uses Node.js with Express to handle application logic, OTP verification, and communication with the database. The blockchain layer consists of an Ethereum smart contract that manages vote storage and enforcement of voting rules. MySQL is used as the database layer to store user metadata and session history.

4. Smart Contract Design

A dedicated smart contract was developed to manage core election operations. The contract handles candidate registration, enforces voting session time limits, tracks voter addresses for each session, updates vote counts immutably, and prevents double voting at the blockchain level. All contract functions were tested using Ganache test accounts to ensure correct behavior before integration.

5. Authentication Methodology

To ensure that only legitimate voters participate in elections, an OTP-based mobile verification mechanism was implemented. The authentication process includes mobile number validation, OTP generation with expiration control, OTP verification before account creation, and binding of the voter account with a MetaMask wallet for blockchain-based voting. This approach strengthens identity verification and prevents unauthorized access.

6. Hybrid Data Storage Strategy

Since storing large volumes of data on the blockchain is costly and inefficient, a hybrid storage strategy was adopted. Critical voting data such as vote transactions, candidate information, and voter address tracking are stored on-chain. Supporting data, including user details, session metadata, archived results, and OTP records, are stored off-chain in a MySQL database. This separation improves scalability while maintaining transparency and security.

7. System Implementation

Each system component was implemented independently and later integrated into a complete application. The smart contract was deployed using Truffle and Ganache. Backend services were developed using Express.js, while Web3.js was used to connect the application with the blockchain. Database schemas were created in MySQL for managing voters and election sessions. Separate frontend interfaces were designed for voters and administrators to simplify interaction.

8. Testing and Validation

The system was tested and validated using a Ganache local blockchain to simulate real voting scenarios. MetaMask accounts were used to test wallet-based authentication and transaction signing. Various test cases were performed, including double voting attempts, voting outside allowed session times, unauthorized access, and OTP expiration or mismatch scenarios. The results confirmed that the system reliably enforces voting rules and maintains tamper-proof vote records.

B. System Design

1. Overall System Architecture

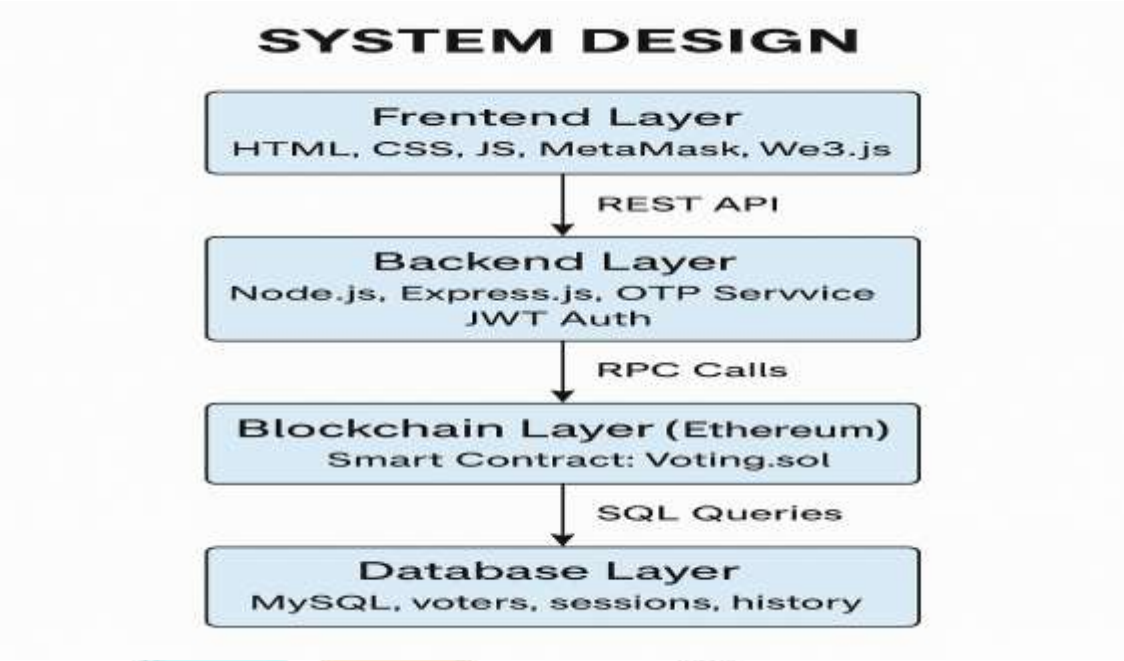

Figure 1: system architecture

2. Module Design

a. Voter Registration Module

- Validates mobile number
- Sends OTP (in development mode, displayed on console)
- Verifies OTP
- Allows password creation
- Generates unique Voter ID
- Stores voter details in MySQL

b. Login & Authentication Module

- Verifies credentials using bcrypt-hashed passwords
- Generates JWT for session management
- Validates MetaMask connection

c. Voting Module

- Checks blockchain session timings
- Ensures voter has not already voted
- Processes vote transaction via MetaMask
- Smart contract increments vote count

d. Admin Module

- Creates and manages voting sessions
- Adds or removes candidates
- Views real-time results
- Resets session and archives results into MySQL

e. Blockchain Smart Contract Module

- Contains all election rules
- Maintains vote counts
- Prevents duplicate voting
- Ensures immutable result storage

3. Data Flow Diagram (DFD – Level 0)

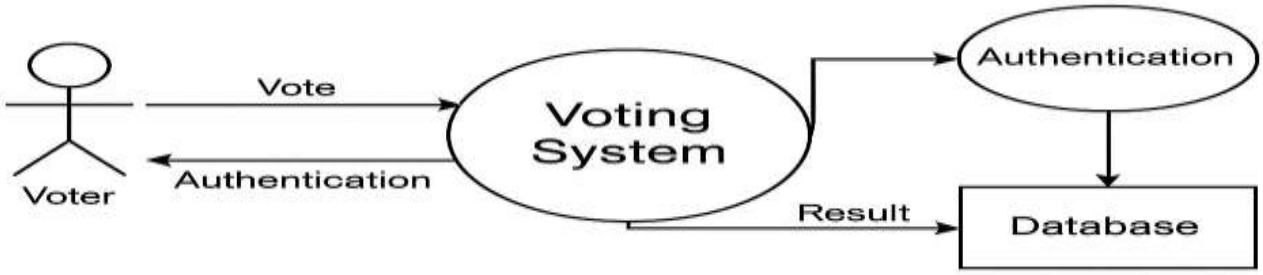

Figure 2: DFD -Level 0

Data Flow Diagram (DFD – Level 1)

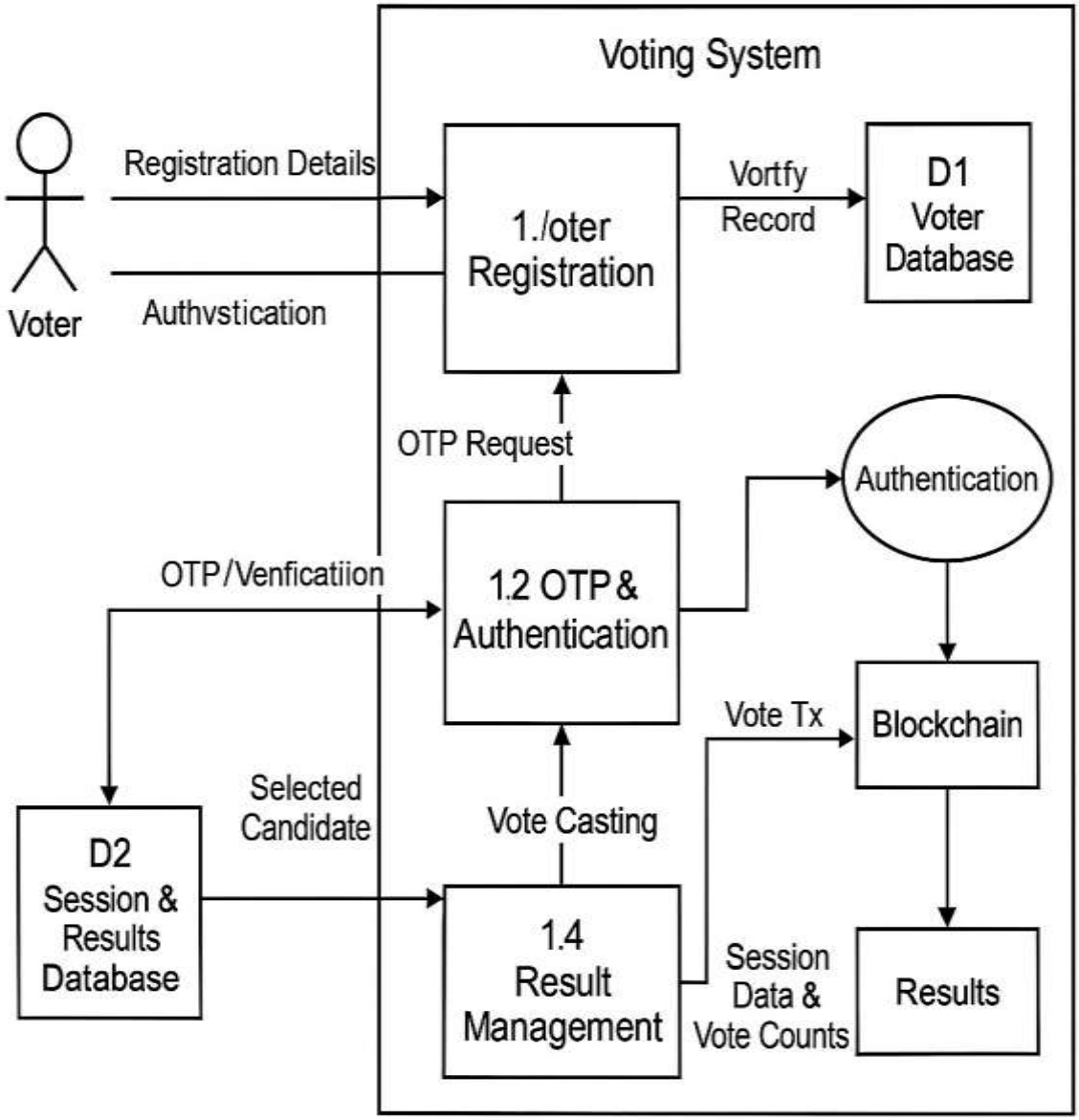

Figure 3: DFD - Level 1

4. Security Design

- Blockchain immutability for vote protection
- OTP verification to prevent fake registrations
- Wallet-based authentication via MetaMask
- Bcrypt hashing for password security
- JWT-based request authorization
- Session-wise voter tracking in smart contracts

VII. SYSTEM ARCHITECTURE

The proposed decentralized voting system follows a layered architecture that separates user interaction, application logic, blockchain operations, and data persistence. This design improves maintainability, security, and scalability while enabling seamless integration between traditional web technologies and Ethereum smart contracts.

A. Architecture Overview

- The system is organized into four main layers:
- Frontend Layer (Client / Browser)
- Backend Layer (Application Server)
- Blockchain Layer (Ethereum Network)
- Database Layer (Relational Storage)

These layers communicate through well-defined interfaces: HTTP/REST between the browser and server, JSON-RPC calls between the server and blockchain, and SQL queries between the server and the MySQL database.

B. Frontend Layer

The frontend is implemented using HTML, CSS, and JavaScript, and is accessed through a standard web browser. It provides separate views for:

- Voter Interfaces: registration, login, voting page, and session history.
- Admin Interfaces: candidate management, session configuration, and result viewing.
- Key responsibilities of the frontend layer include:
- Capturing user inputs such as voter details, OTP codes, login credentials, and selected candidates.
- Displaying countdown timers, candidate lists, and real-time results.
- Integrating with MetaMask, which injects the window.ethereum object and allows users to sign blockchain transactions securely.
- Using Web3.js to initiate voting transactions and query smart-contract state when necessary.

The frontend layer remains stateless; all security-sensitive operations (authentication, voting, and data validation) are delegated to the backend and the smart contract.

C. Backend Layer

The backend is built using Node.js and Express.js and acts as the central coordinator between the client, blockchain network, and database. Its main functions are:

API Endpoints:

- Exposes RESTful routes for OTP generation, OTP verification, voter registration, login, session/result management, and auxiliary operations.

OTP and Authentication Service:

- Generates and validates one-time passwords, verifies login credentials using bcrypt-hashed passwords, and issues JSON Web Tokens (JWT) for authenticated requests.

Business Logic and Validation:

- Enforces server-side validation rules (mobile number format, password strength, duplicate accounts), checks authorization based on user roles (admin/user), and ensures that only eligible voters can access blockchain voting.

Blockchain Gateway:

- Uses Web3.js on the server side to interact with the deployed Voting.sol smart contract for administrative operations such as setting voting periods, resetting sessions, and reading result summaries.

Database Access:

- Manages persistent data through MySQL queries, logging important events such as registrations, sessions, and final results.

- By centralizing these responsibilities, the backend layer provides a secure, controlled environment for all interactions while shielding the blockchain from malformed or unauthorized requests.

D. Blockchain Layer

The blockchain layer is implemented using the Ethereum platform, typically running on a Ganache local network during development and testing. It hosts the Voting smart contract, which encapsulates the core election rules:

- Storage of candidates, vote counts, and session identifiers.

- Enforcement of voting periods through votingStart and votingEnd timestamps.

- One-vote-per-address-per-session logic using mappings from session numbers to voter addresses.

- Immutable recording of each successful vote, making tampering virtually impossible after confirmation.

All transactions that modify election state (such as casting a vote or configuring dates) are submitted as signed Ethereum transactions through MetaMask, ensuring that the user's private key never leaves the wallet.

E. Database Layer

The database layer uses MySQL to store structured, off-chain information that is not practical or cost-effective to keep on the blockchain. The core entities are:

- Voters Table: holds voter IDs, full names, mobile numbers, bcrypt-hashed passwords, and roles (admin/user).
- Voting Sessions Table: stores session numbers, start and end timestamps, total votes, candidate summaries, winners, and historical results in JSON or relational form.

This layer supports:

- Efficient querying for login, administration, and reporting.
- Long-term archival of results and audit logs.
- Integration with external tools (e.g., analytics dashboards or reporting scripts).
- The database never stores raw votes; it only stores aggregated and historical information derived from blockchain state, preserving integrity while enabling efficient analysis.

F. Inter-Layer Interaction

The interaction among layers follows a clear sequence:

- A voter accesses the web interface and performs registration or login.
- The backend validates credentials, interacts with the MySQL database, and returns JWT tokens for subsequent requests.
- When the voter casts a vote, the frontend triggers a MetaMask transaction, which is signed locally and sent to the Ethereum network.
- The smart contract validates the voting conditions and updates on-chain state if the transaction is valid.
- The backend periodically or on-demand reads the smart contract state (candidate vote counts, session info) and updates the MySQL session records to maintain a synchronized historical view.
- Administrators and voters can view results through the frontend, which fetches processed data from the backend and, if required, cross-checks with blockchain data.
- This architecture ensures that security-critical operations are handled either by the blockchain or the authenticated backend, while the database and frontend remain focused on usability, persistence, and presentation.

Block Diagram

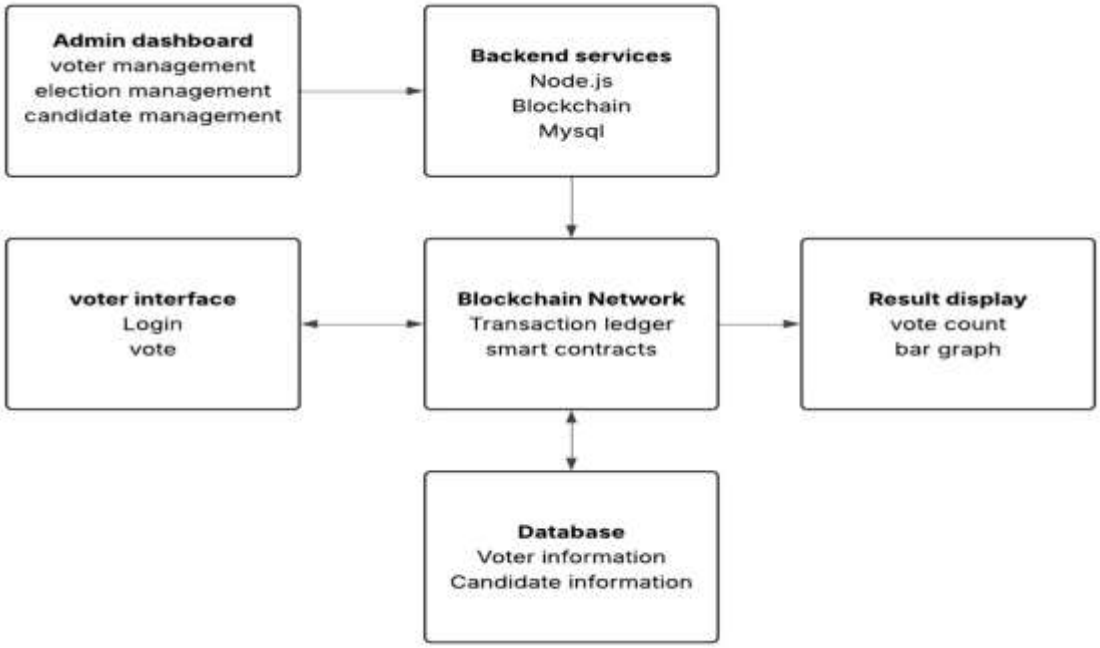

Figure 4: Block Diagram

VII.SNAPSHOTS

Figure 5: Voting Interface

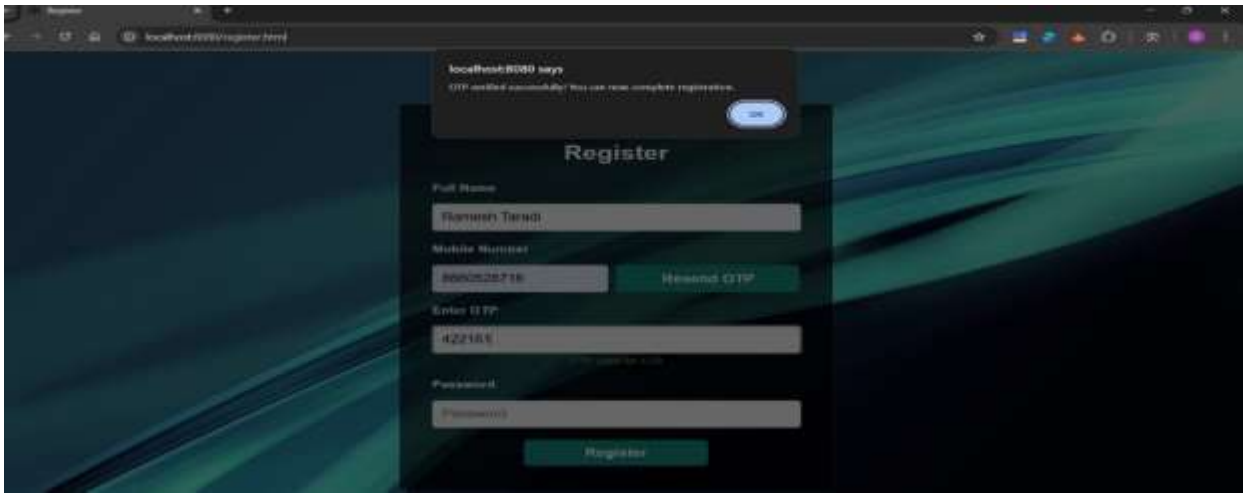

Figure 6: Voter Register Page

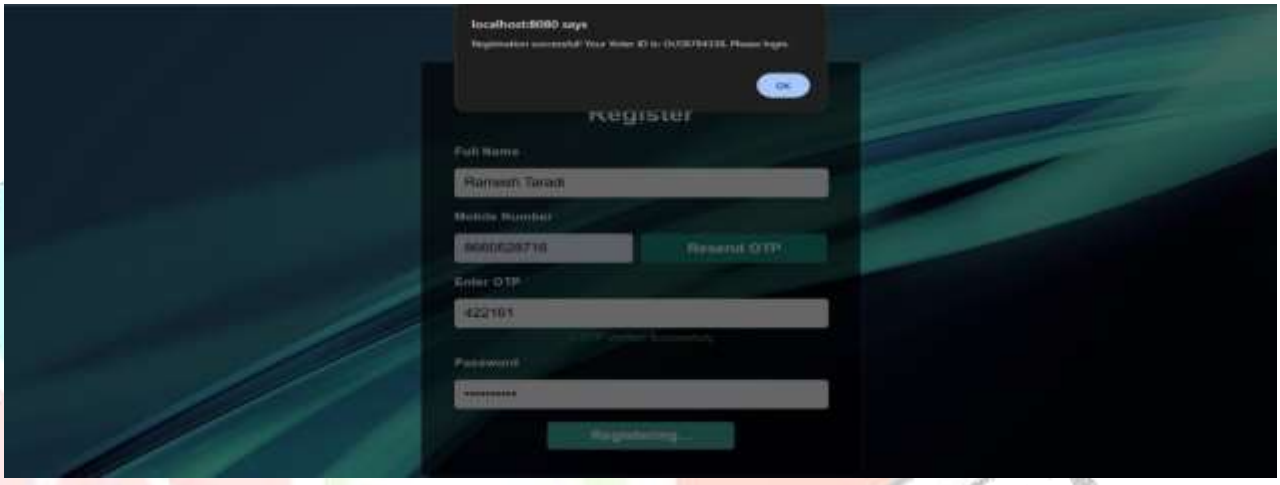

Figure 7: OTP Verification

Figure 8: Login Page

Figure 9: Admin Page

Figure 10: Session History

IX. CONCLUSION

This study demonstrates that a decentralized voting system built on blockchain technology can effectively address many of the weaknesses found in traditional electronic voting models. By using Ethereum smart contracts along with MetaMask-based authentication and a secure backend setup, the proposed system provides a transparent and tamper-resistant voting process. Votes are recorded in an immutable manner, while the hybrid design—storing critical voting data on-chain and managing user and session information off-chain—offers a practical balance between security and system efficiency.

Testing conducted in a controlled Ganache environment showed that the system consistently enforced the one-voter-one-vote rule and protected the integrity of each ballot. The absence of centralized control reduced the risk of manipulation, while OTP-based registration added an additional layer of security by verifying voter identity before participation. These results indicate that blockchain-based voting systems can improve trust, transparency, and reliability without making the voting process complex for users. Overall, the proposed solution presents a scalable and secure approach that can be adapted for use in academic institutions, organizations, and future digital election environments.

X. REFERENCES

1. Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
2. Vitalik Buterin, “Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform,” 2014.
3. Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). “Blockchain Technology: Beyond Bitcoin.” Applied Innovation Review, 2, 6–19.
4. Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends.” IEEE International Congress on Big Data.
5. Kshetri, N. (2017). “Blockchain’s Roles in Strengthening Cybersecurity and Protecting Privacy.” Telecommunications Policy, 41(10), 1027–1038.
6. Ali, A., & Pal, S. (2020). “Blockchain-Based Electronic Voting System.” International Journal of Computer Applications, 176(30), 12–18.
7. Sharma, T., & Gupta, N. (2021). “Secure Online Voting Using Ethereum Blockchain.” International Journal of Engineering Research & Technology, 10(4), 256–261.
8. Wood, G. (2014). “Ethereum: A Secure Decentralised Generalised Transaction Ledger.” Ethereum Yellow Paper.
9. Alzahrani, B., & Bulusu, N. (2021). “Analysis of Blockchain-Based Voting Systems.” IEEE Access, 9, 10343–10356.
10. IJCRT Template, “Paper Format Guidelines,” International Journal of Creative Research Thoughts (IJCRT).