International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 3, 2025

Secure Electronic Voting System

S Revathi, Gajula Sadhika, Nelapatla Sahithi, Kanduri Sri Navya

[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women.

[2.3.4]B,tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women.

gajulasadhika18@gmail.com

ABSTRACT

This project presents a Secure Electronic Voting System using Blockchain technology to address the limitations of traditional and electronic voting systems, such as vote tampering, lack of transparency, and low voter trust. The system leverages the decentralized and tamper-proof nature of blockchain to ensure that every vote is securely recorded and cannot be altered once submitted [1][3].

By Using smart contracts, the voting process is automated, removing the need for third-party verification and reducing human error. Voter authentication is securely handled to prevent duplicate voting, while maintaining voter anonymity and data privacy. All votes are stored on a distributed ledger, making the results transparent, verifiable, and immutable [2].

The platform offers an easy-to-use interface for both voters and administrators, enabling smooth election setup, secure vote casting, and real-time result tracking. This system is ideal for secure digital elections in institutions, organizations, and government bodies, aiming to enhance security, trust, and voter participation through blockchain technology.

Keywords:

Blockchain, E-Voting, Smart Contracts, Secure Voting, Ethereum, MetaMask, Decentralized Systems, Immutable Ledger, Cryptographic Verification, Digital Election.

1-INTRODUCTION

In today's digital age, ensuring secure and transparent elections is more important than ever. Traditional voting methods often face challenges like vote tampering, fraud, and lack of transparency. To address these issues, blockchain technology offers a promising solution. Blockchain is a decentralized digital ledger that records transactions in a secure and unchangeable manner. By applying blockchain to electronic voting, each vote becomes a unique, encrypted transaction that is permanently recorded, making it nearly impossible to alter or delete. This system ensures that votes are counted accurately, maintains voter privacy, and allows for transparent auditing of results. Additionally, smart contracts—self-executing programs on the blockchain—can automate vote counting and validation processes, reducing human error and increasing efficiency. This project aims to develop a user-friendly and secure blockchain-based voting platform that enhances trust in the electoral process and makes voting more accessible, especially for remote or underserved populations.

Existing System:

Traditional voting systems, including paper ballots, Electronic Voting Machines (EVMs), and some online voting platforms, have several limitations. Paper-based voting is time-consuming, requires manual effort, and is prone to human errors, vote tampering, and mismanagement. EVMs, while faster, are centralized and can be vulnerable to hacking or manipulation. Online voting systems also depend on central servers, making them targets for cyberattacks and technical failures. In all these systems, there is limited transparenc,voters cannot verify if their vote was correctly recorded or counted. Voter authentication is often weak, increasing the chances of fake or duplicate votes. These issues raise serious concerns about the security, trust, privacy, and overall reliability of the current voting methods..

Proposed System:

The proposed system uses blockchain technology to create a secure, transparent, and tamper-proof electronic voting platform. Each vote is recorded as a transaction on a decentralized blockchain, making it impossible to modify or delete. Voter identity is verified through secure authentication, and smart contracts are used to automatically count and validate votes. This removes the need for manual counting and reduces the chance of fraud or errors. The system allows remote voting, protects voter privacy, and provides real-time transparency, ensuring trust in the overall election process..

2-RELATED WORK

We briefly review the recent works in related surveys on existing blockchain-based voting systems.Jafar et al. [1] explore the integration of blockchain technology in electronic voting systems, examining its potential to enhance the accuracy, security, and convenience of e-voting processes. The paper discusses the progression of e-voting, addressing how blockchain can resolve prevalent issues. It also discusses various current implementations of blockchain in e-voting, highlighting both general e-voting challenges and specific technical hurdles in blockchain applications. Common e-voting challenges discussed include secure digital identity management, ensuring anonymous vote-casting, personalized ballot processing, and enabling voters to verify their own ballot casting. Moreover, the paper explores technical difficulties such as high initial setup costs, escalating security concerns, prevention of evidence tampering, and balancing transparency with voter privacy. Another significant issue is the existing lack of transparency and trust in e-voting systems. By integrating blockchain technology, many of these longstanding issues in traditional e-voting systems can be largely mitigated. Nonetheless, this work identifies unique challenges that emerge with blockchain-based e-voting systems. These include scalability and processing overheads, constraints due to scalability, block size limitations, and sensitivity to varying demands in voter participation. These challenges underscore the complexity of implementing blockchain in e-voting systems and highlight the need for ongoing research and development in this evolving field.Abuidris et al. [2] offer an in-depth look on various companies that use blockchain-based e-voting systems. A system called Follow My Vote [3] aims to offer a secure online voting platform based on blockchain. The system allows voters to audit the ballot box and see election progress in real-time. They also use a webcam and user ID for remote, secure voting, which enables voters to confirm their votes. Agora [4] proposes a blockchain-based digital voting platform which features immediate recording of votes on multiple blockchain layers and offers tamper-proof results and full auditability while being able to maintain voter privacy.

3.REQUIREMENT ANALYSIS

Functional Requirements:
A blockchain-based electronic voting system is implemented to ensure secure, transparent, and tamper-proof elections. The system uses Solidity smart contracts deployed on the Ethereum blockchain to record all election activities immutably. MetaMask is used for wallet-based login, providing decentralized and secure voter authentication.
The system has separate interfaces and functionalities for admin and voters. Admins manage elections by adding voters and candidates and viewing results. Voters authenticate using their wallet address and can vote only once. The entire process is powered by smart contracts to maintain transparency and prevent manipulation.
Functional requirements for implementing a secure electronic voting system using blockchain include:

- MetaMask Integration
- Candidate Management
- Secure Voting
- Voter Management
- Immutability and Transparency
- Decentralized Identity (Wallet-Based)
- Election Scheduling
- Smart Contract Deployment
- Role-Based Access Handling
- Vote Counting and Result Display

Non-Functional Requirements:
The Secure Electronic Voting System ensure the system's overall quality and usability.non-functional requirements describe how the system should perform under various conditions. These include performance factors such as system speed, reliability, and availability, especially under high user load. While writing the code the software's performance and ability is also considered. Some other non-functional requirements are speed, reliability and the security of the system.

Scalability:

- The system should be capable of handling a growing number of users and transactions without performance degradation.
- Future elections involving larger populations or more complex voting scenarios should be supported without major architectural changes.

Performance:
The system must process and confirm votes quickly, providing real-time updates and instant feedback to the user.

- It should maintain low response times even when many users are accessing the system concurrently.

Software Requirements:
Operating Systems

: Windows
Front-End

: HTML ,CSS, JS
Blockchain Framework

: Ethereum using Hardhat
Environment

: Visual Studio Code
Smart Contract Lang.

: Solidity

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

Hardware Requirements:
Processor

: Intel i5 or higher
Ram

: 8GB.
Hard Disk

: 1TB

4.DESIGN

System Architecture:
It describes the structure and behavior of technology infrastructure of an enterprise, solution or system. In other words, System architecture can be described as the flow of application which is represented below in the pictorial form. He purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission/business and stake holders requirements).
System architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements. It addresses the architectural principles, concepts, properties, and characteristics of the system-of-interest. It may also applied to more than one system, in some cases forming the common structure, pattern, and set of requirements for classes or families of similar or related systems.
The SEBoK considers systems engineering to cover all aspects of the creation of a system, including system architecture.
The majority of interpretations of system architecture are based on the fairly intangible notion of structure (i.e relationships between elements). Some authors limit the types of structure considered to be architectural: for example, restricting themselves to functional and physical structure. Recent practice has extended consideration to include behavioral, temporal and other dimensions of structure.
ISO/IEC/IEEE 42010 Systems and Software Engineering – Architecture Description (ISO 2011) provides a useful description of the architecture considering the stakeholder concerns, architecture viewpoints, architecture views, architecture models, architecture descriptions, and architecting throughout the life cycle.

A discussion of the features of systems architectures can be found in(Maier and Rechtin 2009). An attempt to develop and apply a systematic approach to characterizing architecture belief systems in systems engineering has been described by the INCOSE UK Architecture Working Group (Wilkinson et al.2010, Wilkinson 2010).

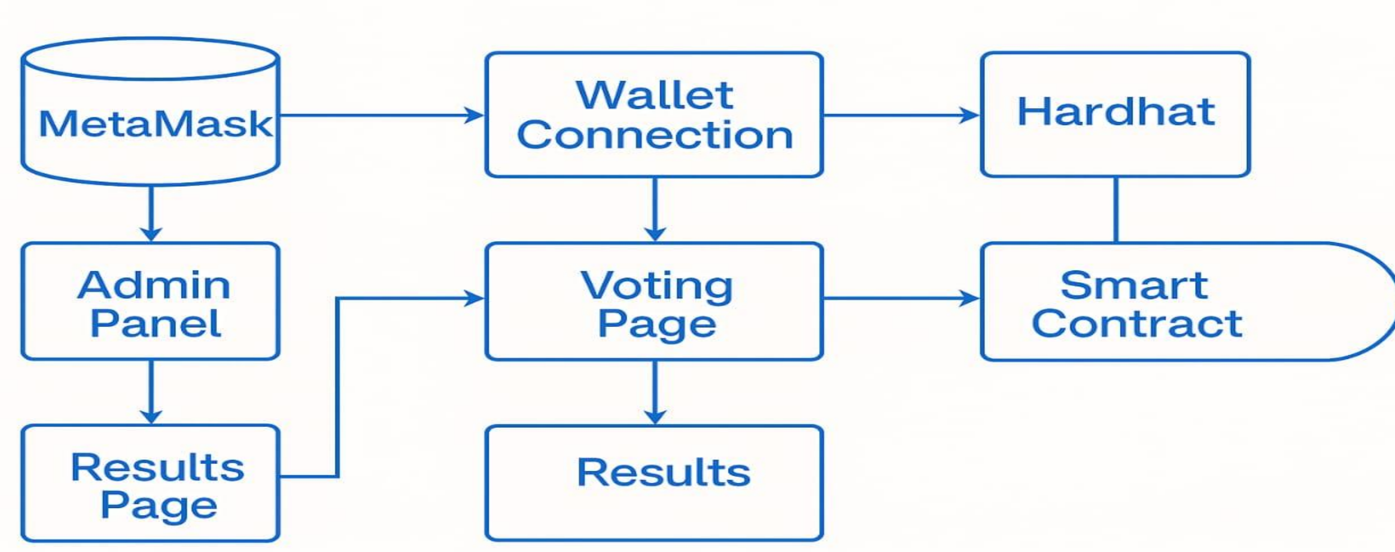


Fig. 4.1.1.1 System Architecture

Technical Architecture:

Technical Architecture refers to the structural

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

process of designing and building system's architecture with focus on the users and sponsors view of the environment. Technology architecture associates application components from application architecture with technology components representing software and hardware components. Its components are generally acquired in the market place and can be assembled and configured to constitute the enterprise's technological infrastructure. A technical architecture diagram provide a bird's eye view of the infrastructure of our project. The diagram illustrates how components in a system interact with one another in the large scale of things. Technical Architecture (TA) is a form of IT architecture that ids used to design computer system. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

Throughout the past decade, architecture has become a broadly used term in the context of information technology. This doesn't come as a surprise considering how most companies had to redesign their IT landscape to adopt digital trends like cloud computing software as service (SaaS). This digital transition required not only skilled developing teams but first and foremost IT architects. In their roles as IT strategists and planners, they map out a target architecture and make sure that all IT decisions align with business goals and requirements.

But IT architecture encompasses a variety o different roles and disciplines that are sometimes difficult to tell apart. This is largely due to highly dynamic nature of IT, its widespread adoption throughout all industries and business that have developed their own practices. In general, there's differentiation between enterprise architecture, solution architecture and technology architecture. In order to understand what technology architecture means, it's helpful to examine the term architecture on its own.

At its core, the term architecture describes the formation of a structure by strategically assembling single components. In this process of assembling, the architect has to adhere to certain rules or requirements like legal constraints, financial constraints or scientific laws. In the world technology architecture design, the focus lies on technology limitations, meaning that a technology architect makes sure that a new application is compatible with the existing technology at a company by specifying things like the communications network or hardware that it uses.

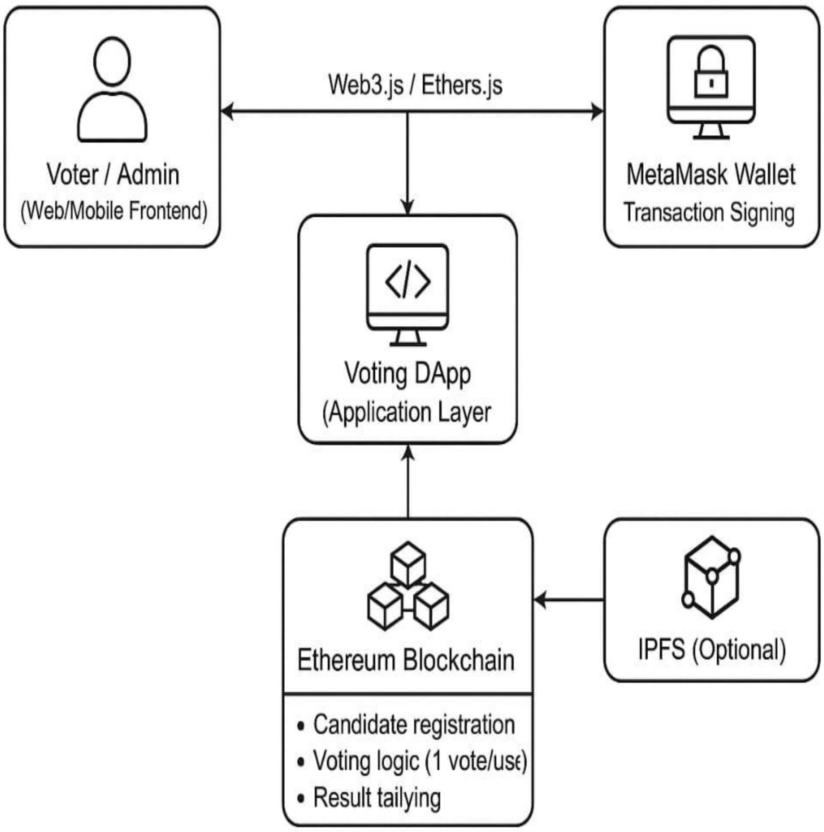


Fig. 4.1.2.1 Technical Architecture

5.IMPLEMENTATION

5.1 Technologies & Libraries

• *HTML/CSS/JavaScript:*
These are the core frontend technologies used to build the user interface. HTML is used for structuring the web content, CSS is used for styling and layout, and JavaScript enables dynamic interactions on the webpage such as candidate selection and real-time updates..

• *MetaMask:*
MetaMask is a browser extension used to manage Ethereum wallets. It allows users to authenticate, sign transactions, and interact securely with the blockchain directly from the browser. It acts as a bridge between the frontend and the Ethereum network.

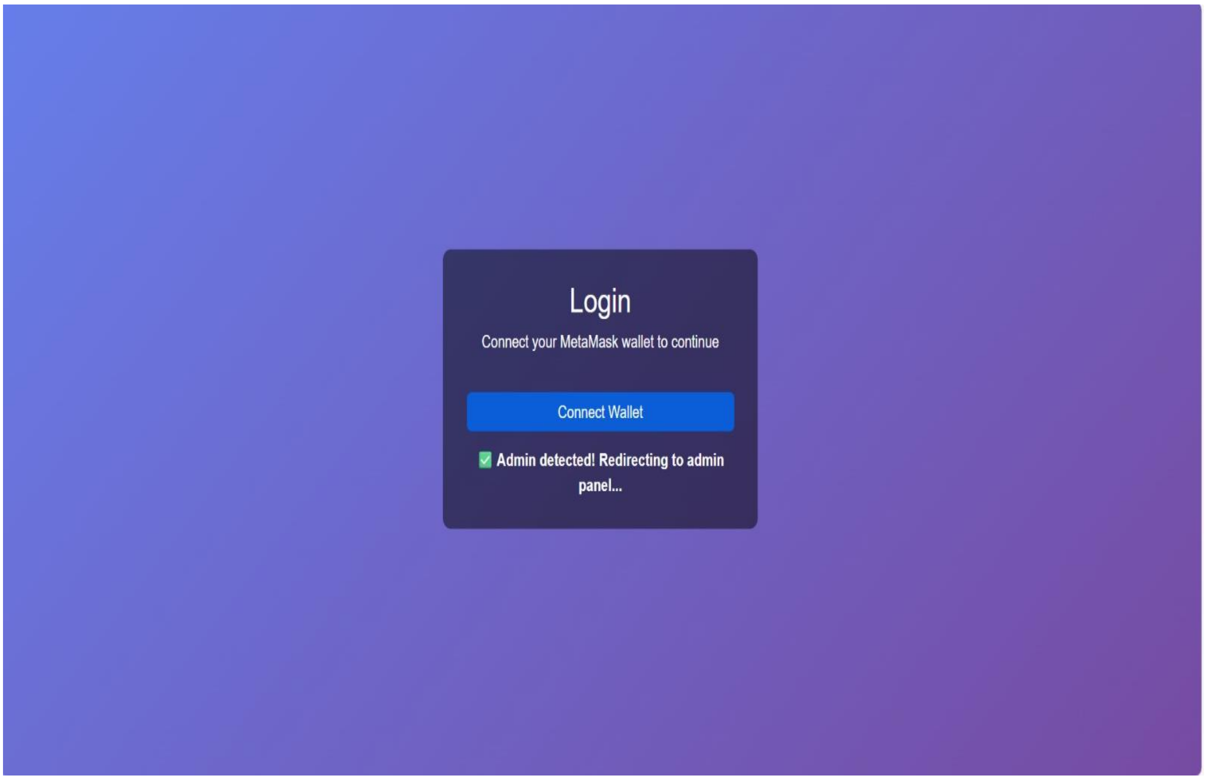
• *Ethers.js:*
Ethers.js is a JavaScript library used for interacting with the Ethereum blockchain. It provides functionalities to connect the frontend with smart contracts, send transactions, and read blockchain data. It is lightweight, secure, and works well with MetaMask.

• *Solidity:*
Solidity is a contract-oriented programming language used to write smart contracts. In this project, it is used to define logic for voting, candidate registration, and result computation. These smart contracts ensure security, transparency, and immutability.

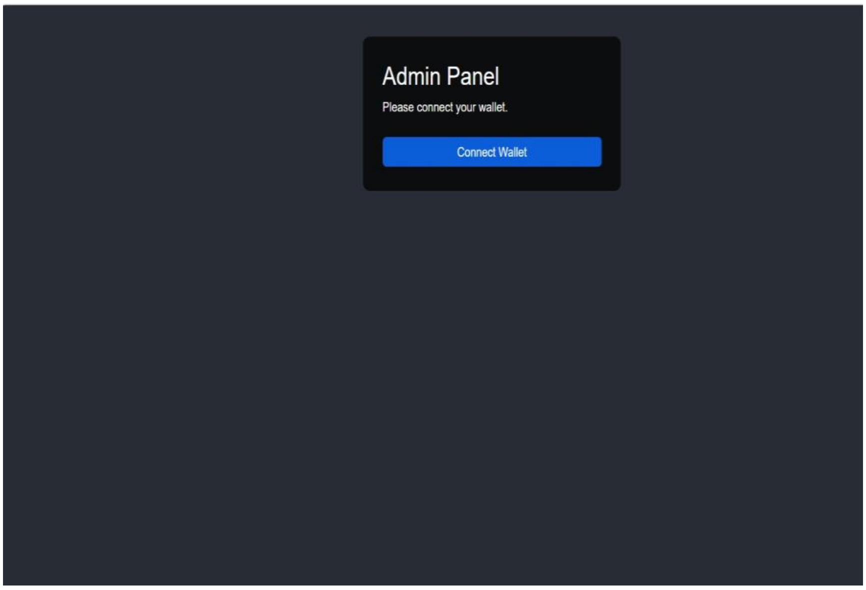
• **Hardhat:**
Hardhat is a development environment used to compile, deploy, and test smart contracts. It replaces Ganache in this project and provides local blockchain simulation, debugging tools, and seamless integration with Ethers.js.

• *Node.js & npm:*
Node.js is used as the runtime environment to run JavaScript outside the browser, and npm (Node Package Manager) is used to install dependencies like Ethers.js and Hardhat necessary for smart contract development and frontend-backend integration.

6.SCREENSHOTS

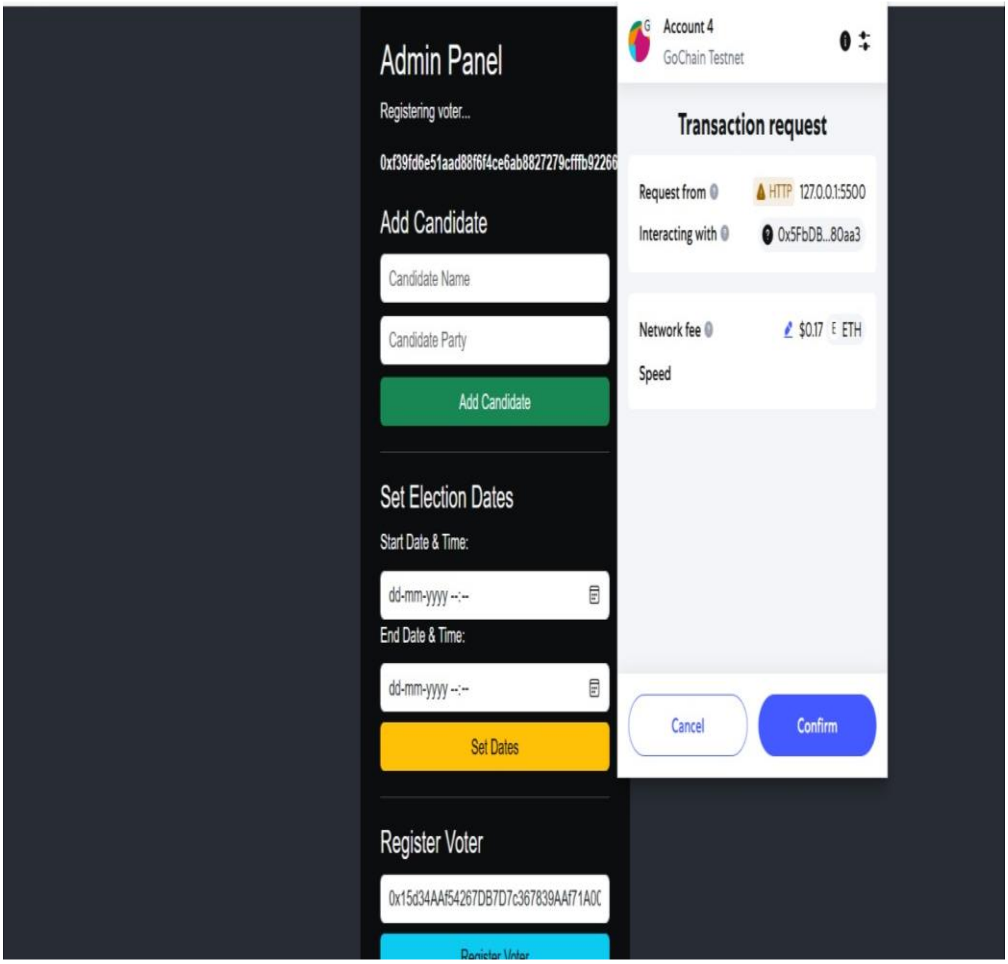


Screenshot 6.1 Secure Electronic Login Page

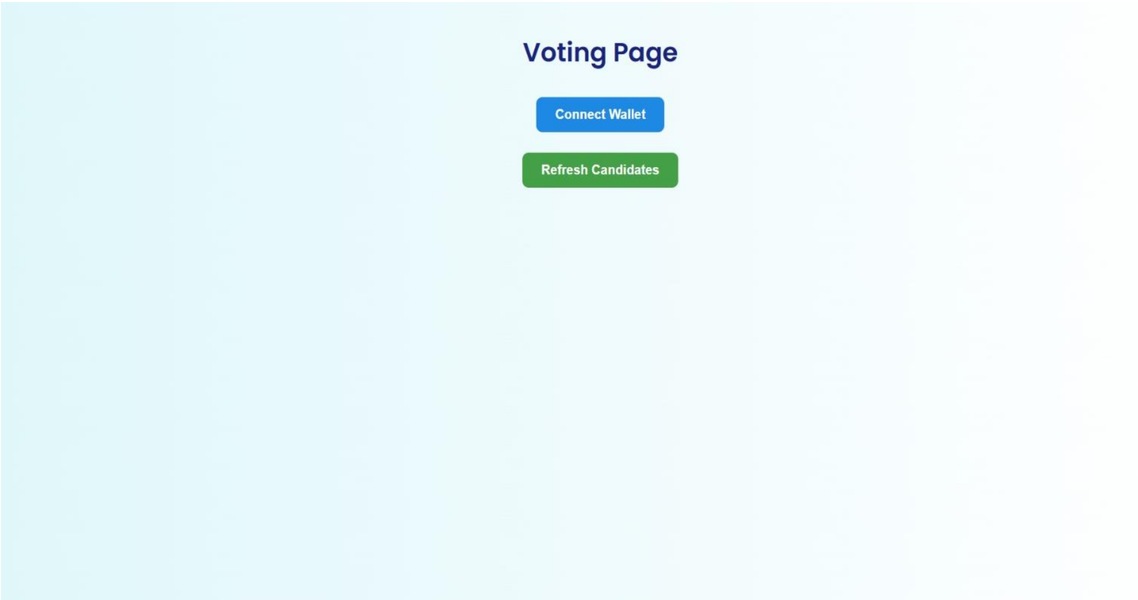


Screenshot 6.2 Secure Electronic Voting System Admin panel

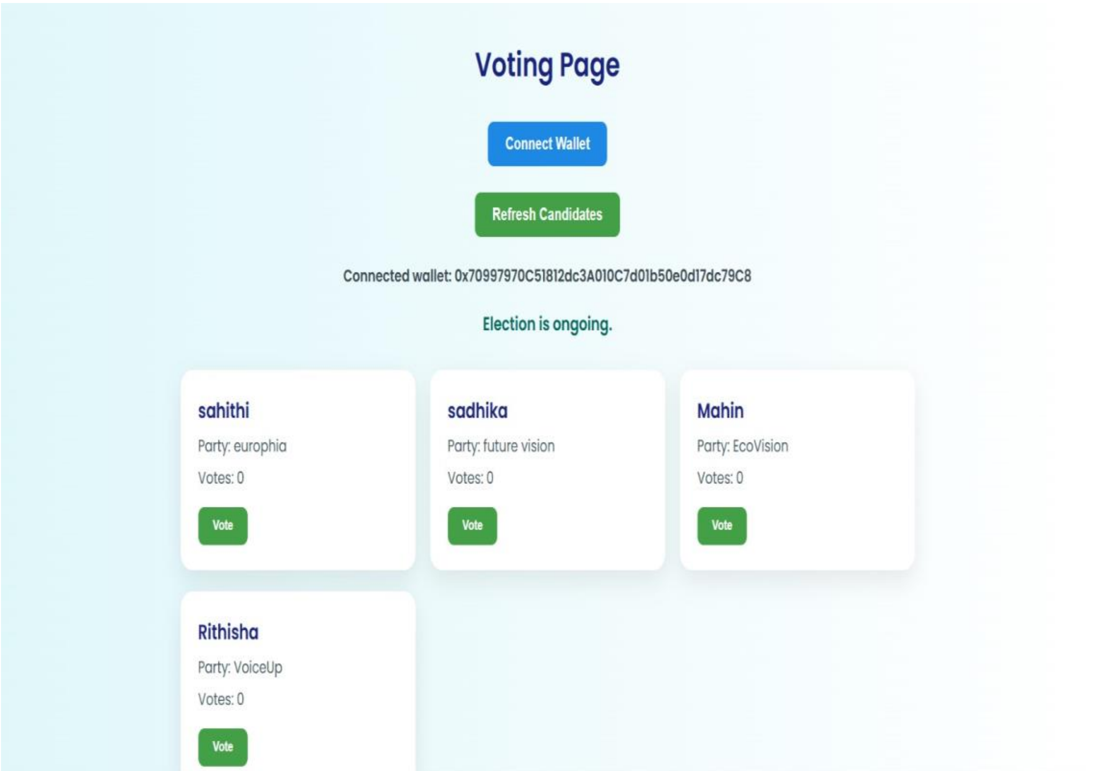
\



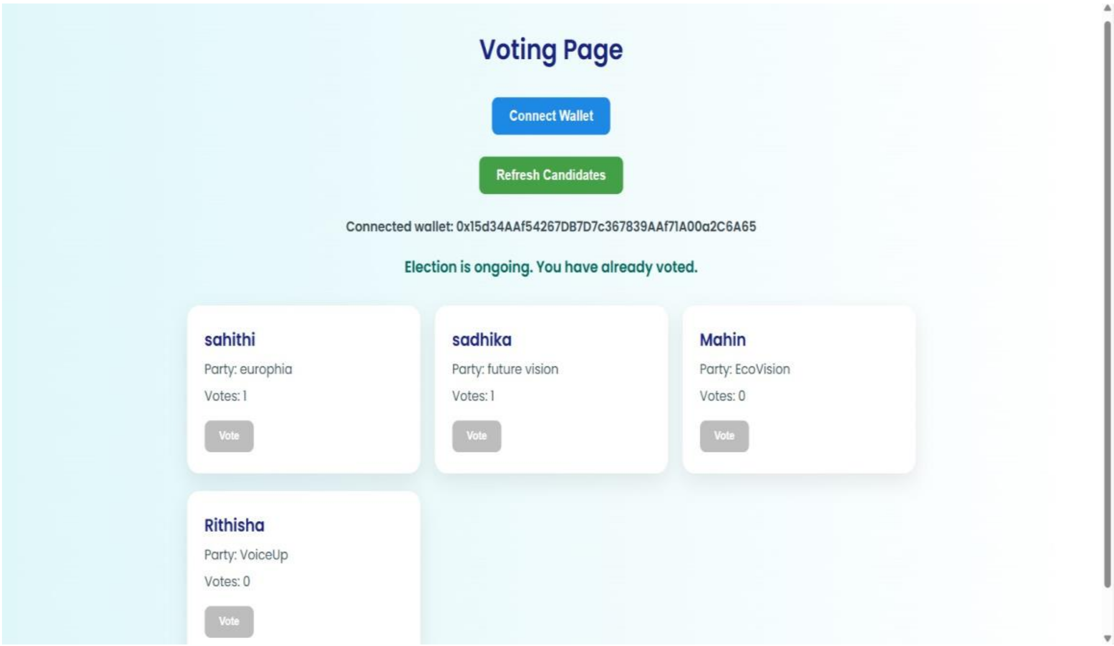
Screenshot 6.3 Secure Electronic Voting System Admin panel



Screenshot 6.4 Voting Page



Screenshot 6.5 Voting Page



Screenshot 6.6 Voting Page

7-CONCLUSION

The Secure Electronic Voting System using Blockchain offers a transparent, tamper-proof, and decentralized solution to address the major limitations of traditional voting systems. By leveraging blockchain technology, this system ensures the integrity and immutability of every vote cast. The integration of smart contracts, MetaMask, and Ethereum blockchain enhances trust, transparency, and automation in the voting process. Additionally, using tools like Hardhat and React provides a smooth development and user experience. This system not only prevents vote manipulation and fraud but also empowers remote and real-time voting with enhanced security and anonymity. Through this project, we demonstrate how modern technologies can transform democratic processes, paving the way for a more secure, scalable, and efficient electoral system in the future.

REFERENCES

[1] Wood, G. (2014). *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum Yellow Paper. https://ethereum.github.io/yellowpaper/paper.pdf

[2] Christidis, K., & Devetsikiotis, M. (2016). *Blockchains and Smart Contracts for the Internet of Things*. IEEE Access, 4, 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339

[3] Bhargava, B., Doss, R., & Yavuz, A. (2020). *A Blockchain-based Secure Voting System*. IEEE International Conference on Decentralized Applications and Infrastructures.

[4] Hardhat Documentation. (2024). *Ethereum development environment for professionals*. https://hardhat.org/docs

[5] MetaMask. (2024). *MetaMask: Ethereum Wallet and Gateway to Blockchain Apps*. https://metamask.io/

[6] Ethers.js Documentation. (2024). *A complete Ethereum library in JavaScript*. https://docs.ethers.io/

[7] Solidity Documentation. (2024). *Solidity – Smart Contract Programming Language*. https://docs.soliditylang.org/

[8] Wüst, K., & Gervais, A. (2018). *Do you need a Blockchain?*. IACR Cryptology ePrint Archive. https://eprint.iacr.org/2017/375.pdf

[9] Al-Bassam, M. (2017). *Scalable, Transparent, and Cryptographically Verifiable Voting System*. https://arxiv.org/abs/1705.11001

[10] Chaum, D., Rivest, R. L., & Sherman, A. T. (2009). *Scantegrity: End-to-end voter-verifiable optical-scan voting*. IEEE Security & Privacy. https://doi.org/10.1109/MSP.2009.39

[11] Buterin, V. (2013). *Ethereum White Paper*. https://ethereum.org/en/whitepaper/

[12] GitHub – ethers-io/ethers.js. (2024). https://github.com/ethers-io/ethers.js

[13] GitHub – MetaMask Extension. (2024). https://github.com/MetaMask/metamask-extension

[14] National Institute of Standards and Technology (NIST). (2021). *Security Considerations for E-Voting*. https://www.nist.gov/

[15] Truffle Suite Documentation. (2024). *Truffle: Smart Contract Development Framework*. https://trufflesuite.com/docs/.