IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DECENTRALIZED VOTING SYSTEM

Prachit Kanwar, Parv Jain, Mukund Khandelwal, Naman Acharya

Department Of Computer Science and Engineering, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India.

I. ABSTRACT

This project explains about decentralized voting system for conducting elections is a revolutionary approach designed to ensure accessibility, transparency, and impartiality in the electoral process. By leveraging blockchain technology and secure cryptographic techniques, this system allows voters to cast their ballots remotely, eliminating the need for physical polling locations. Each vote is securely recorded on an immutable and transparent ledger, preventing tampering or fraud. Furthermore, the decentralized nature of the system means that it is not controlled by any single entity, reducing the potential for bias or manipulation. This innovative voting system holds the promise of greater inclusivity, integrity, and trust in the democratic process, making it a vital tool for modernizing elections and ensuring fair representation.

II. INTRODUCTION

A decentralized voting system, implemented as a decentralized application (Dapp) using web technologies, represents a pioneering solution for conducting fair and secure elections online. Unlike traditional voting systems, which often rely on centralized authorities and paper ballots, a decentralized voting Dapp leverages the power of blockchain technology to create a transparent, tamper-resistant, and trustless voting process. Key features that set it apart from its conventional counterpart include cryptographic security to protect voter privacy, immutable ledger for transparent and auditable results, elimination of intermediaries, and accessibility from anywhere with an internet connection. This innovative approach not only enhances the integrity of the electoral process but also increases inclusivity and convenience for voters, potentially revolutionizing the way we participate in democratic decision-making.

Key features:

1. Trust and Transparency
2. Accessibility
3. Anonymity
4. Immutable records
5. Real-time results

6. Decentralization

III. METHODOLOGY

This application is implemented via visual studio code platform or remix IDE (open-source IDE) to write smart contracts and deploy. With NodeJS (NPM packet manager) for server-side development and using solidity (smart contract language) for storing records on blockchain, for frontend we will use HTML and CSS for designing the layer of interaction for the voter and we will use react framework to make it more interactive and the other dependencies are Web3.js, hardhat and meta mask wallet.

1. Node js: - Node.js is an open-source runtime environment that allows developers to execute JavaScript code outside of a web browser. It is built on the V8 JavaScript engine and provides a server-side platform for building scalable and high-performance applications. Node.js is known for its non-blocking, event-driven architecture, which makes it particularly well-suited for building real-time, data-intensive applications like web servers, APIs, and microservices. Its extensive package ecosystem, powered by npm (Node Package Manager), offers a wide range of libraries and modules, making it a popular choice for web development and other server-side applications. Node.js has gained widespread adoption and popularity for its ability to streamline the development process by enabling both server-side and client-side code to be written in the same programming language, JavaScript.

2. Hardhat: - Hardhat is a popular development framework and toolset for Ethereum developers. It is designed to simplify the process of building and deploying smart contracts on the Ethereum blockchain. Hardhat offers a range of features and capabilities, including a built-in Ethereum development network, automated testing, and a comprehensive set of development tasks. It also supports integration with other Ethereum development tools and libraries, making it a versatile choice for building decentralized applications (Dapps) and interacting with the Ethereum blockchain. With its robust functionality and developer-friendly environment, Hardhat has become a valuable tool for those looking to streamline the Ethereum development process and ensure the reliability and security of their smart contracts.

3. Remix IDE: - Remix IDE is a popular integrated development environment (IDE) for Ethereum blockchain development. It provides a user-friendly and web-based platform for creating, testing, and deploying smart contracts and decentralized applications (Dapps) on the Ethereum network. Remix offers a range of features, including a code editor, a Solidity compiler, a debugger, and a web3 provider, making it a comprehensive tool for Ethereum developers. It simplifies the development process by offering a real-time development environment with built-in testing capabilities and easy deployment options, ultimately streamlining the creation of Ethereum-based projects.

4. MetaMask: - To use blockchain we must connect to it. We'll have to install a special browser extension to use Ethereum blockchain. we'll be able to connect our local Ethereum blockchain with our personal account and interact with smart contract.

IV. MODELING AND ANALYSIS

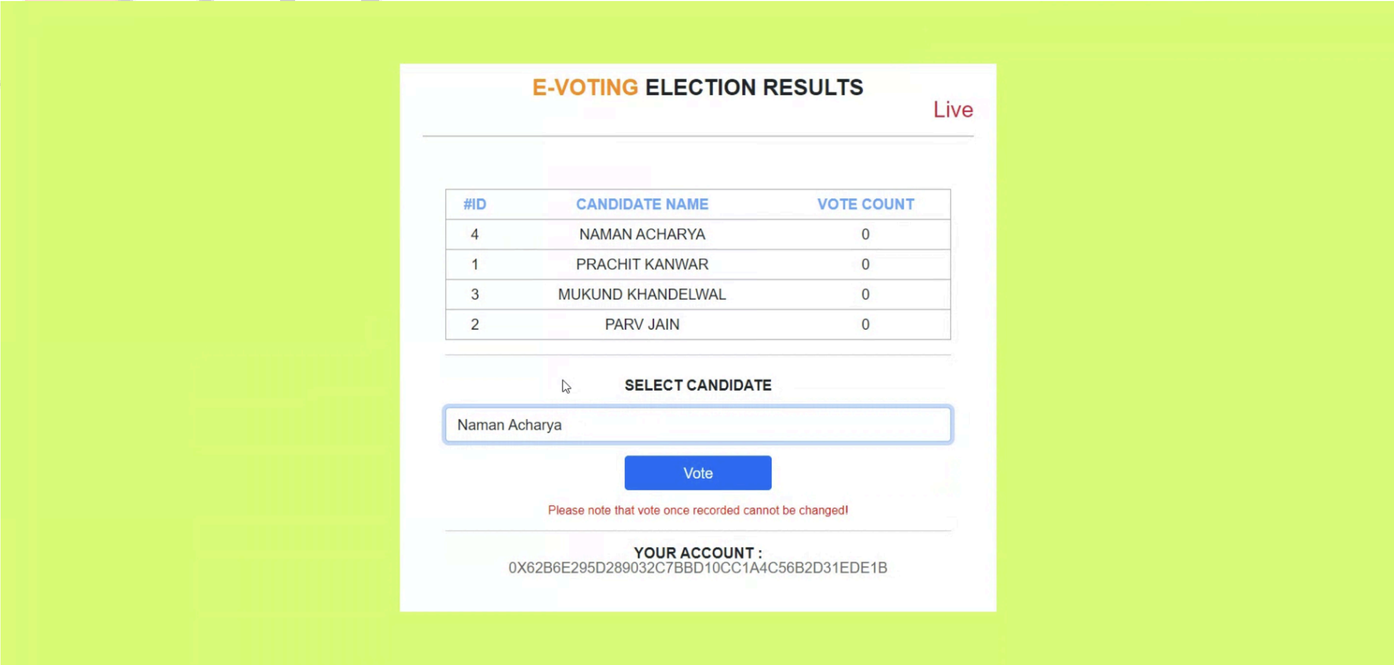

FIGURE 1: INITIAL INTERFACE

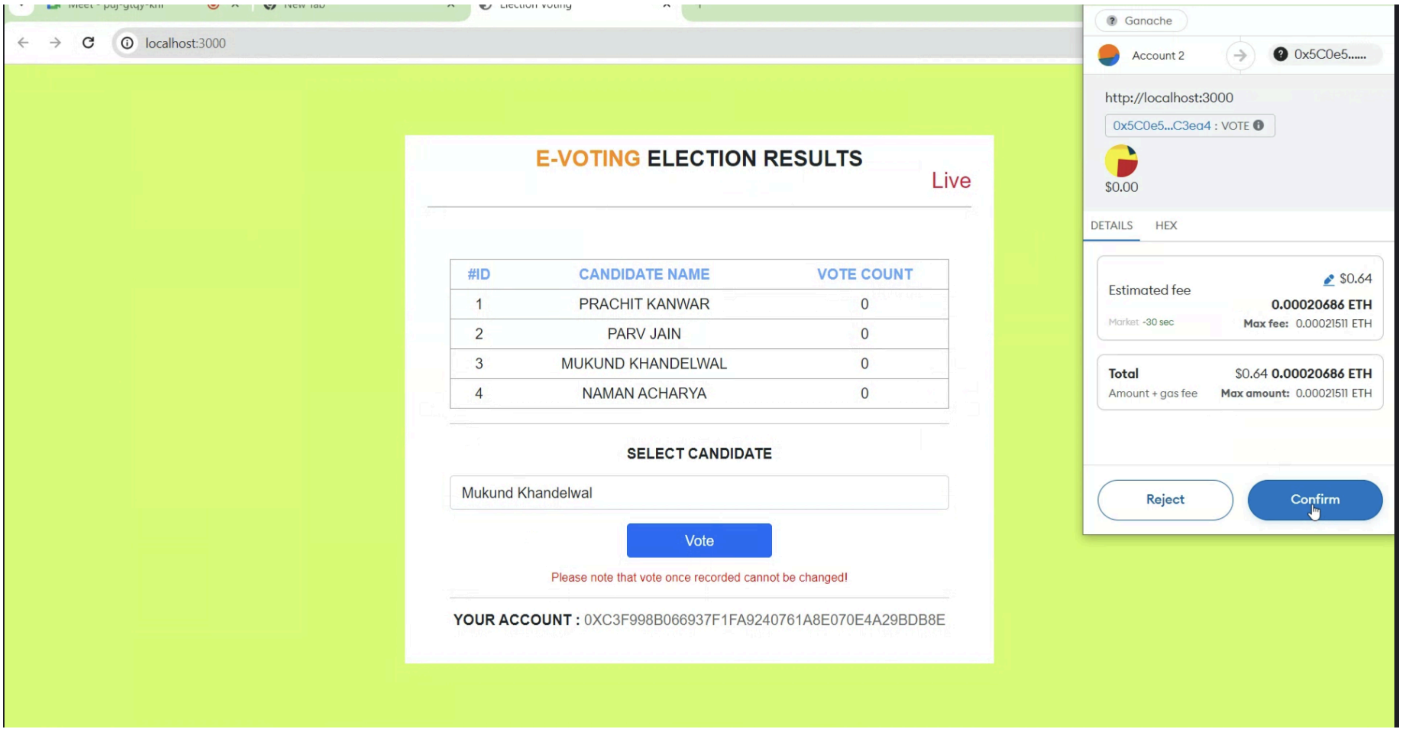

FIGURE 2: CONNECTING META MASK

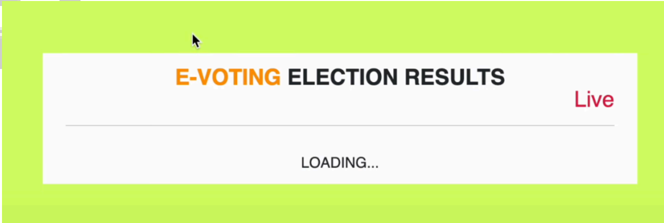

FIGURE 3: VOTING INITIALIZED

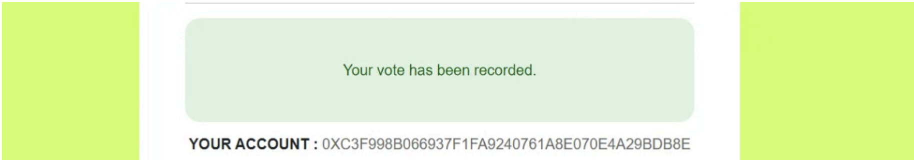

FIGURE 4: VOTE RECORDED

V. RESULT AND DISCUSSION

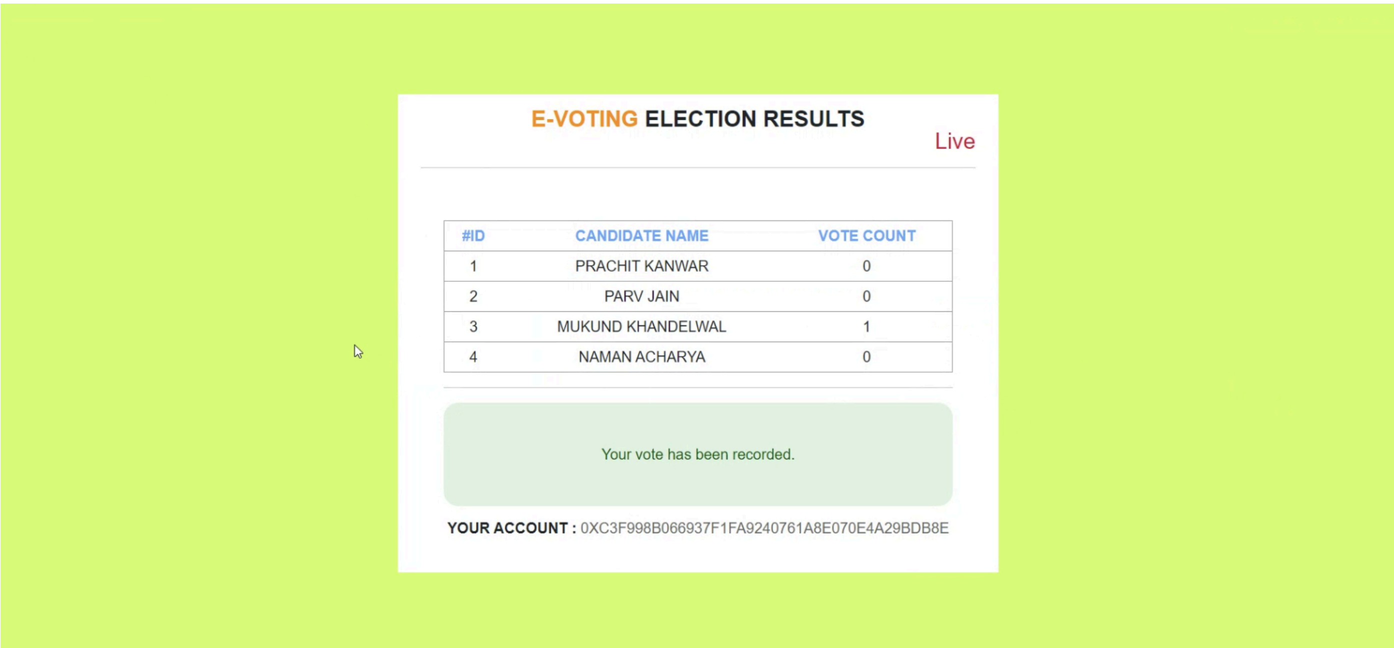

FIGURE 5: Election Result

1. Transparency and Immutability: The blockchain-based e-voting system demonstrated high levels of transparency and immutability. Each vote was securely recorded on the blockchain, ensuring that once a vote was cast, it could not be altered or deleted. This immutable record enhances trust in the electoral process.

2. Security through MetaMask Integration: By utilizing MetaMask wallets for voter authentication, the system provided a secure and user-friendly interface. MetaMask's encryption and security features ensured that voter identities were protected and only eligible voters could cast their ballots.

3. Real-Time Vote Counting: The blockchain's decentralized nature enabled real-time vote counting. As each vote was cast, the vote count on the blockchain ledger increased by one instantaneously. This real-time update feature eliminates delays associated with traditional vote counting methods.

4. Decentralization and Reduced Fraud: The decentralized architecture of the blockchain reduced the risk of centralized manipulation and fraud. Since the voting data was distributed across multiple

nodes, tampering with the election results would require compromising a majority of these nodes, which is highly impractical.

5. Scalability and Performance: The system demonstrated good scalability, handling a large number of votes without significant performance degradation. However, further stress testing is required to evaluate performance under maximum load conditions, which is critical for large-scale elections.

6. User Experience and Adoption: Feedback from test users indicated that the integration with MetaMask made the voting process straightforward and accessible. However, the requirement for a MetaMask wallet might pose a barrier for less tech-savvy voters, suggesting a need for user education and possibly simplified wallet solutions.

VI. Conclusion

In this project, we introduced a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient elections while guaranteeing voters privacy. It offers a new possibility to overcome the limitations and adoption barriers of electronic voting systems. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain. To this end, we believe an effective model to establish trustworthy provenance for e-voting systems will be crucial to achieve an end-to-end verifiable e-voting scheme.

VII. References

- Efanov, D. and Roschin, P. (2018) 'The all-pervasiveness of the blockchain technology', Procedia Computer Science, Vol. 123, pp.116–121, DOI: 10.1016/j.procs.2018.01.019.

- EOS.IO Software (2018) EOS.IO [online] https://eos.io/ (accessed 20 August 2018).

- Grech, A. and Camilleri, A.F. (2017) Blockchain in Education, JRC Science for Policy Report. Hyperledger (2015) Hyperledger Project [online] https://www.hyperledger.org/.

- Jesus, E.F. and Chicarino, R.L. (2018) 'A survey of how to use blockchain to secure internet of things and the stalker attack', Security and Communication Networks, Vol. 2018, Article ID 9675050, pp.1–27, DOI: 10.1155/2018/9675050.

- Johnson, D., Menezes, A. and Vanstone, S. (2001) 'The elliptic curve digital signature algorithm (ECDSA)', International Journal of Information Security, Vol. 1, No. 1, pp.36–63.

- Kikwai, B.K. (2017) 'Elliptic curve digital signatures and their application in the bitcoin crypto-currency transactions', International Journal of Scientific and Research Publications, Vol. 7, No. 11, pp.135–138, ISSN: 2250-3153.

- King, S. (2013) Primecoin: Cryptocurrency with Prime Number Proof-of-Work [online] http://primecoin.io/bin/primecoin-paper.pdf (21 August 2018).

- Kosba, A., Miller, A., Shi, E., Wen, Z. and Papamanthou, C. (2016) 'Hawk: the blockchain model of cryptography and privacy-preserving smart contracts', IEEE Symposium on Security and Privacy (SP), pp.839–858.

- Lamport, L., Shostak, R. and Pease, M. (1982) 'The byzantine generals problem', ACM Trans. Program. Lang. Syst., Vol. 4, No. 3, pp.382–401.