Enhancing Electoral Integrity through Blockchain-Enabled Decentralized Voting System

Dikshant Rajput [1], Ranu Rajput [2], Durgesh Nandini [3]
[1,2] UG student, Computer science and engineering, Galgotias University
[3] Assistant professor, Computer science and engineering, Galgotias University

Abstract-* The Centralized approach to E-voting uses decentralized technology to form transparent election systems where voters can rely on secure ballot protection from tampering. Elected votes are accurately stored within the Blockchain network because its decentralized setup creates protection from unauthorized modifications. Every vote within an E-voting system adopts an unchangeable position across multiple servers to allow peer-to-peer verification of transactions. The system maintains an unalterable status for each vote so it functions as a separate entry without any modification capability. The release of voting outcome reports starts immediately when the voting period finishes. Democratic procedures in all voting nations utilize secret voting documentation along with proprietary methods similar to voting papers. Traditional elections typically encounter three significant problems that involve tampering of votes and passive voter turnout in addition to technical complications during ballot administration. The proposed system consists of decentralized voting mechanisms that enhance security protocols alongside operational effectiveness as well as election trust mechanisms.

I. INTRODUCTION

Election is the means by which a population chooses a government, ultimately contributing to the formation of a government through voter participation. It is known that elections is one of the basic foundations of any democracy where citizens elect an experienced leaders that build a good democracy. In a conventional vote, People register with the election commission voting list before voting day. Ballots can be cast either in person or by mail, and must be submitted by a certain time. However, this process can be interrupted by natural calamities or long lines [1]. Elections in India extend back to the early Vedic era, when the assembly of people elected the ruler. Around the end of the 16th century, modern elections began to take shape in both North

America and Europe. Modern electronic systems like EVM (Electronic Voting Machines) gradually displaced difficult and prone to errors traditional processes when new techniques were introduced. The concept of computerized voting systems was a significant advancement since it made voting easier, required less physical labor, and increased accuracy and dependability. To improve security, effectiveness, and usability, the suggested system incorporates technologies like smart contracts, blockchain, and electronic voting [2]. With this combination, the weaknesses of earlier voting procedures will be addressed, and enhance the democratic process.

E-Voting:

Electronic voting refers to using electronic technology both for vote casting and vote counting. The voting procedure stores ballot information across different media forms including smart cards and tape cartridges before the central site compiles the votes. The implementation of electronic voting systems relies on optical scanners and direct electronic recording (DER) technology among others.

There are two main types of digital voting: On-s electronic voting: Using this technique, voters cast their ballots at polling places manned by electronic voting machines while being watched over by government representatives. Voters may have to wait in line to cast their ballots and must be physically present at the polling station. Remote computing voting: This enables people to cast their ballots via computers, cell phones, or voting kiosks from any location. Voters now exercise their right to vote online without needing to go to polling stations because electronic options such as the internet and SMS messaging and alternative systems are available to them.

Blockchain:

The reliable alternative to traditional system infrastructure emerges through blockchain technology which offers unalterable files while maintaining complete transparency. Data arrives in an organized fashion by utilizing block structures which maintain a linked relationship between different blocks throughout the chain. The first segment of this sequencing is designated as the "genesis block." The system generates new blocks by placing them directly above previously established blocks in sequence [3]. The stored data, its distinct hash, and the hash of the block before it are the three main parts of a block. Any modification to a block's data modifies its hash, but the next block keeps the modified block's original hash, making the modified block and all subsequent blocks invalid.

Smart Contract:

A smart contract is a self-implementing contract that runs on a blockchain and is incorporated in computer code. Within the code both contractual terms and procedures exist to establish interaction regulations between parties executing contracts. When all contractual conditions fulfil their established requirements the agreement automatically starts its execution phase. Digital assets and access authorization mechanisms are structured using smart contracts as a robust operational system between several parties. [4].

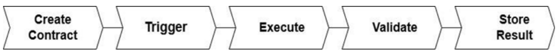

Fig. 1. Working Principle of Smart Contract

II. LITERATURE REVIEW

The electronic voting system was widely adopted in many nations and represented a major advancement over the conventional pen-and-paper approach. E-voting has been included into electoral procedures in numerous countries. Improved voter turnout, auditability, cost effectiveness, accessibility, and convenience were among its many advantages, but it also raised a number of serious issues and problems.

Cruz J. P. et al. [5] proposed an electronic voting system based on blind signatures and Bitcoin. However, voting systems' proof-of-work mechanism is linked to high energy usage. Furthermore, it can be difficult to identify and stop voters from using illegal Bitcoin addresses or abusing several Bitcoin addresses to cast votes repeatedly. Additionally, these systems frequently ignore some security flaws in the Bitcoin protocol, like selfish mining attacks, 51% assaults, and double-spending attacks. McCorry et al. (2017) [6] proposed a blockchain-based voting administration solution to increase election process transparency. This framework eliminates the need for physical polling places by enabling digital voting via blockchain. The suggested electronic voting system is subject to Denial-of-Service (DoS) assaults, even if it places a strong emphasis on voter privacy.

Jonathan Alexander et al. [7] used NetVote as the user interface for their decentralized application (dApp) program. The dApp administrator provides electoral administrators with the ability to manage electoral policies and voting procedures and registration guidelines as well as election start dates and end dates. External systems together with biometric reading devices operate as voting identification methods. The Tally Dapp application performs election result evaluation tasks. The electoral system within Net vote contains three distinct formats which include private elections and open ones together with token-holder elections.

Khan et al. (2020) [8] demonstrated how blockchain technology can build a decentralized electronic voting platform which solves the existing issues with traditional voting processes. The system implements a voting process that stays accessible in all times to deliver secure data transfers while operating validation systems and maintaining voter identities. The system implements MetaMask and Truffle and Ganache technologies for its operation. The voting process has one major disadvantage in that ballots can be observed during voting operations. Voters must possess a generated random token for ballot generation that remains untouchable. The voting token remains susceptible to three safety risks including theft, physical damage and loss that may block voters from participating in elections.

Boshri et al. (2019) [9] designed an Ethereum-based system for blockchain voting according to their research. The election committee establishes an Ethereum account within its framework to protect and handle voter information. Voters who do not own smartphones can verify their identity with biometrics and then proceed with voting at polling stations. The system defends voting anonymity through the implementation of chain-based ring signatures. The voting method reduces decentralization because it needs third party involvement for uploading ballots to the blockchain instead of utilizing smart contracts for these procedures.

AMVchain represents a blockchain system created by Li et al. (2021) as a solution that executes decentralized voting across a secure and decentralized framework with optimized scalability and performance delivery. Smart contracts from the system control a three-layered authorization framework which handles access authentication procedures along each stage. AMVchain implements chain-based ring signatures as its major means to ensure balloting anonymity throughout voting. Additional proxy nodes allow the system to operate with enhanced scalability and higher voting efficiency by increasing its concurrent activities.

The authors Clarke et al. (2023) [11] established an e-voting system based on Hyperledger Fabric to allow voters secure ballot casting and encryption functions. The approach protects voter

The system preserves voter privacy but provides decentralized complete transparency to voting outcomes. The design achieves high performance capacity along with scalability according to experimental results that prove support for thousands of participants. The system incurs particular costs while operating with a personal blockchain network which could affect equality but creates potential opportunities for manipulation.

Dimitriou et al. (2020) [12] created a blockchain voting system to protect elections by requiring minimal participation from voters. A randomization token enables the system to prevent receipt generation and reduce external pressure placed on voters. The blockchain adds information only at the end which requires minimal trustworthy election authority while ensuring everyone can verify results. When electronic votes lose voter engagement it damages the principles for which electronic voting systems were developed.

Multiple problems within current electronic voting software create obstacles to achieving valid voting results which include:

a. Their susceptibility to hacking efforts.
a. Inefficiency in executing extensive audits.
b. Voter intentions are misunderstood.
d. System creators' potential political bias.
e. Modifying software applications.
f. Lack of proper security systems put cast ballots at risk.
g. technical problems and hardware malfunctions.

This study creates a blockchain-enabled secure electronic voting system to tackle the possible glitches which traditional voting methods encounter. The solution can be implemented for various uses such as businesses and educational institutions as well as during national elections. Electronic voting systems suffer from frequent incidents such as machine tampering and vote manipulation as well as disinformation operation planning. This proposed model tackles the operational flaws through a proficient method.

III. PROPOSED SYSTEM

A combination of four essential tools forms the proposed system utilizing Truffle framework with npm and Ganache and MetaMask. Truffle serves to develop blockchain-based smart contracts which can be accessed using the Ganache emulation environment through MetaMask. Users must have a minimum of Ether currency to generate their wallet address for a new account. Blockchain transactions require the payment of gas fees for successful recording on the blockchain. After vote submissions the network processing occurs through several nodes who operate as miners. User payments to miners result in Ether distributions to miners who successfully process the votes. In this method, the mining operation is carried out using Ganache software rather than real network nodes.

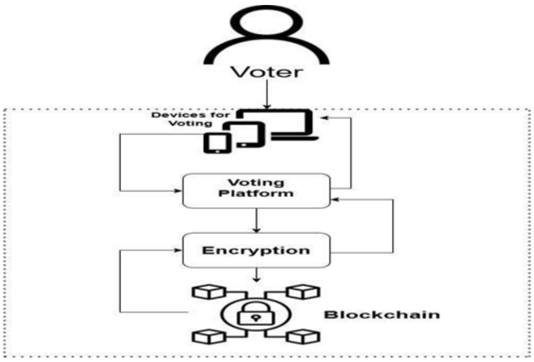

Fig. 2. Proposed Decentralized Voting System Based on Blockchain

Preliminaries:

The recommended framework can be used with 64-bit hardware, a computer running Windows 7 or later, and a number of necessary tools and frameworks, such as Ganache, the Solidity toolkit, MetaMask, the Truffle framework, and NPM dependencies. The following are essential elements needed for implementation:

i. "Node Package Manager (NPM)"
ii. "Truffle Framework"
iii. "Ganache"
iv. "MetaMask"
v. "Programming Languages: Solidity, HTML, JavaScript, and CSS "

NPM (Node Package Manager):

Node.js packages are installed, updated, and removed from an application using NPM, a package manager. It functions as a command-line interface. The NPM can operate through two distinct modes which are local and global operation. While local mode only affects the application's unique directory, global mode affects all Node.js applications [13].

Truffle Framework:

The Truffle Framework is a well-liked asset pipeline, testing framework, and development environment for Ethereum blockchain apps. It is a powerful framework made specifically for Ethereum smart contract development. Among its many capabilities, it manages packages and networks, provides a platform for automated contract testing, and makes it easier to perform operations like assembling, installing, and connecting smart contracts [14].

Ganache:

Ganache, is a software development for testing Blockchain application on localhost. Ten default Ethereum addresses are generated by the virtual blockchain it builds, each of which is preloaded with a private key and a fictitious balance of 100

Ether. In contrast to conventional mining, Ganache verifies transactions automatically. Operating systems including Windows, Linux, and macOS are compatible with it [15].

MetaMask:

MetaMask functions through its connections with the Ethereum blockchain network together with other blockchain networks. This cryptocurrency wallet serves as an entry point to blockchain applications. MetaMask serves as an easy-to-use open-source tool which enables Ethereum transactions with its graphical user interface. Without requiring a complete Ethereum node users can execute Ethereum DApps through their browser using MetaMask. MetaMask connects the browser platform to the Ethereum blockchain operation. [16].

Solidity:

For the creation of smart contracts developers use Solidity which serves as a specific high-level programming language made for this purpose.. Its syntax is comparable to that of JavaScript. It creates machine-level code for Ethereum Virtual Machines (EVMs), which is subsequently translated into simple instructions. Boolean, integer, address, and string are the four main value types that Solidity offers. It also employs operators that are the same as those found in JavaScript [17].

Working

The voter uses the MetaMask Chrome plugin to log into the voting website before linking to the local blockchain. After making a connection the web page shows information regarding both the candidates and the current vote count during page refresh. The voter can proceed by selecting the "Vote" button after completing candidate selection. Users understand their voting Ethereum transaction through an interface that MetaMask displays through a pop-up window. After transaction confirmation the system will cast a vote for the selected candidate unless the voter has already performed a vote. Voters who try to vote more than once will experience transaction failure since their extra votes remain uncounted.

A blockchain local network starts based on Ganache specifications as MetaMask executes connection procedures to this network. With Truffle framework users gain an easier way to transfer Solidity-written smart contracts onto their local blockchain. User votes within MetaMask enable the inter-account transfer of Ether. Each voter account gets a uniform Ether distribution and obtains personalized Ethereum address and private key combinations. The voter sees their Ether transferred to the candidate's account after making their vote. The blockchain system divides every transaction into blocks until the system launches publicly thereby exposing all data.

By ensuring total transparency, this technique enables voters to confirm their ballots. A lower Ether balance will appear at the voter's address after they have cast their ballot. The transaction will fail if the voter tries to cast another ballot, and the extra vote won't be counted. Although network nodes normally carry out mining, under this configuration, Ganache is set up to manage auto-mining on behalf of other nodes.

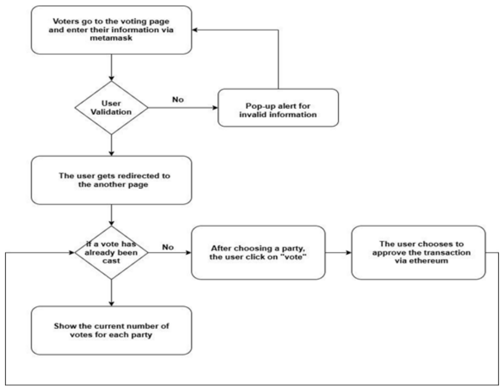

Fig. 3. Flowchart of the Decentralized Voting System based on Blockchain

IV. METHODOLOGY

An overview of the different entities included in the system is given in Fig. 4. It explains each stage of the planned voting process, from the administrator presenting a new ballot for the election to the large number of voters casting their ballots. The Fig. also shows how these entities interact, as well as the steps and processes involved in decision- making. It also emphasizes the system's functional correspondences. The functional relationships between the various modules are also depicted in this image. There are several important modules involved:

i. Login as a user
ii. Making a fresh ballot
iii. The voting process
iv. Instantaneous outcomes

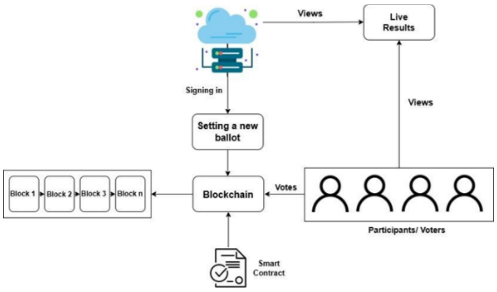

Fig. 4. Decentralized Voting System architecture

Initializing the Ganache

Initializing Ganache is the first step in starting a local blockchain.

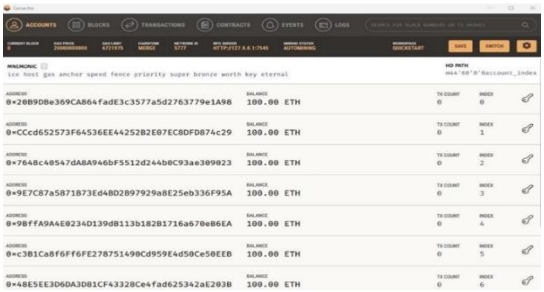

Fig. 5. Initializing the Ganache

Since no actions have been taken yet, there won't be any transactions at first after Ganache is conFig.d. The Fig. 6 given below attests to the fact that no transactions have occurred.

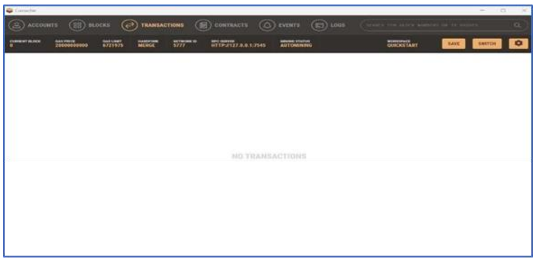

Fig. 6. No transactions in Ganache

By executing commands in the command line, the smart contract's data is put onto the blockchain via the Truffle framework. For this procedure, the command prompt is also used to reach the NPM directory.

To do this, the following commands given in Fig. 7 (a) & (b) are utilized

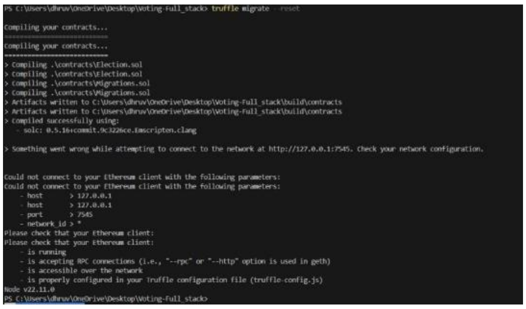

Fig. 7(a). Command for Truffle Framework

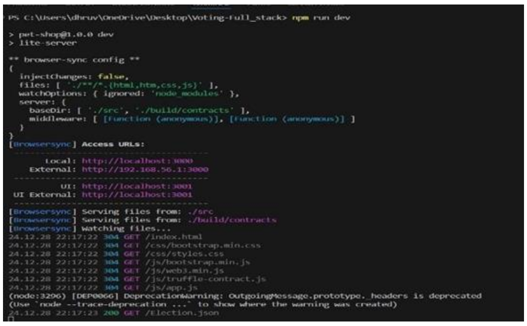

Fig. 7(b). Command Lines for Truffle Framewor

Visual Interface

Through its graphical interface users can use the electronic voting platform. Fig. 8 displays the UI which users will experience when viewing the system. The interface loads only when the voter completes their login with MetaMask. The sequence of screens appears to the voter when they start MetaMask login.

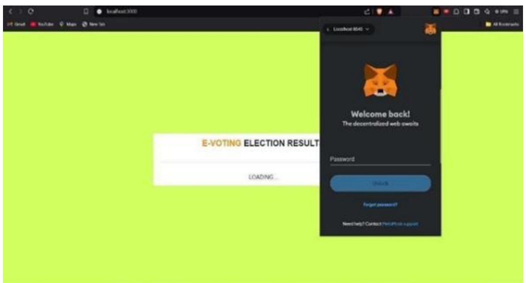

Fig. 8. Voters Accessing the Platform Through MetaMask

The user successfully registers, and the main screen initially displays with no votes. The user must import their account by entering their private key, as illustrated in Fig. 9, before they can cast a vote.

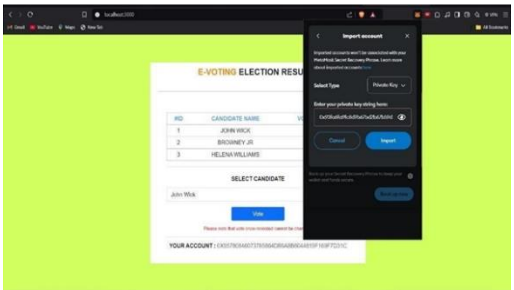

Fig. 9. Preview of Importing Account

The voter enters the private key in the appropriate field to import their account. When they click the vote button to finalize the transaction after choosing their favorite candidate, a MetaMask pop-up window displays.

Once the transaction has been verified, the homepage is displayed to the voter, where voting is no longer permitted and just the results are shown in given Fig. 10. In a similar manner, other users can import their accounts and vote.

After successful transaction verification, the homepage is displayed to the voter, where additional voting is prohibited and just the results are shown. In a similar manner, other users can import their accounts and vote.

Fig. 10. Preview of the result on the main page

Monitoring the transaction

Users will find it easier to check their votes through public access to the transaction list in the system. Through this system voters can verify their voting participation by accessing the public transaction list. Users can verify correct vote recording and track their voting history through the transaction list that makes the verification process straightforward. Voters are given the ability to independently verify their electoral donations, which also increases systemic trust. A thorough representation of the transaction list is shown in Fig. 11, which also illustrates how the data is arranged for user clarity and verification.

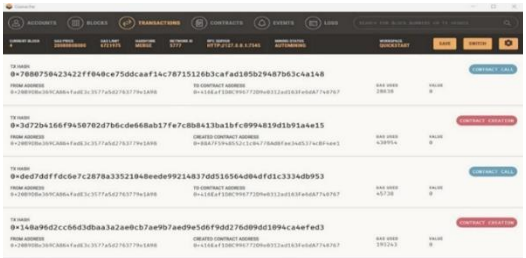
Fig. 11. Transaction List

V. RESULTS

Fig. 10 demonstrates how elections can be held remotely using blockchain technology. This study investigates the role of blockchain in a transparent smart contract-based voting system. The three main steps in the procedure are voting, counting, and authentication. Multiple people can vote at once, and counting begins after the voting is complete.

The voting and outcome stages occur simultaneously in the suggested system, and the authentication step is predefined. Voters can view real-time updates thanks to this

method, which ensures transparent, real-time results. Voters can now instantaneously watch the live results while choosing their favorite candidate. The subsequent figures, which are displayed in Figs. 12 and 13, compare the suggested system with the current one.

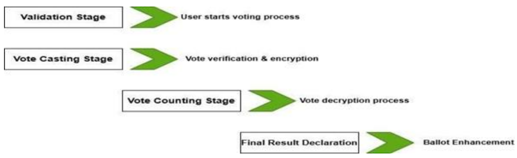

Fig. 12. Performance of existing system based on time frame

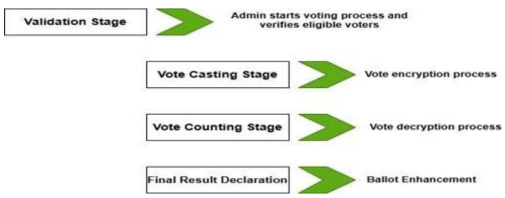

Fig. 13. Performance of proposed system based on time frame

VI. CONCLUSION

The adoption of blockchain technology in modern voting system development creates systems which offer security alongside affordability and speed. The system serves better than typical practices as a dependable and precise replacement by delivering enhanced accuracy. The article develops blockchain-based electronic voting through smart contract mechanisms that define communication and decision-making processes between participating parties.

The development utilized Ganache together with Truffle framework and NPM along with MetaMask tools as implementation components. Blockchain operates outside central control so it protects information through minimized unauthorized modifications and alterations. Our proposed method offers users a convenient interface to improve voter convenience. Users gain access to voting through the system where they can participate and check their outcomes immediately after account import. The project's future development plans aim for using machine learning, particularly via incorporating a facial recognition system. Because it would stop fake and duplicate votes, this feature would greatly increase security. Additionally, the system may incorporate two-factor authentication, which would prohibit unauthorized proxy voting even in the event that voter credentials were compromised.

REFERENCES

[1] Gibson, J. P., Krimmer, R., Teague, V., & Pomares, J. (2016) "A review of e-voting: the past, present and future." Annals of Telecommunications, 71, 279-286.

[2] Çabuk, U. C., Adiguzel, E., & Karaarslan, E. (2020) "A survey on feasibility and suitability of blockchain techniques for the e-voting systems." arXiv preprint arXiv:2002.07175.

[3] Alrebdi, N., Alabdulatif, A., Iwendi, C., & Lian, Z. (2022) "SVBE: Searchable and verifiable blockchain-based electronic medical records system." Scientific Reports 12 (1), 1–11.

[4] Zhang, S., Wang, L. & Xiong, H. Int. J. Inf. Secure. (2019) Chaintegrity: blockchain enabled large-scale e-voting system with robustness and universal verifiability. International Journal Information Security.

[5] Cruz, J. P., & Kaji, Y. (2017) "E-voting system based on the bitcoin protocol and blind signatures." IPSJ Transactions on Mathematical Modeling and Its Applications, 10(1), 14-22.

[6] McCorry, P., Shahandashti, S. F., & Hao, F. (2017) "A smart contract for boardroom voting with maximum voter privacy." In Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21 (pp. 357- n 375). Springer International Publishing.

[7] Jonath Alexander, Steven Lander and Ben Howerton (2018). Netvote: A Decentralized Voting Network Available at: https://netvote. io/wp- content/uploads/2018/02/Netvote White-Paper-v7.pdf

[8] Khan, S., Arshad, A., Mushtaq, G., Khalique, A., & Husein, T. (2020) "Implementation of decentralized blockchain e-voting." EAI Endorsed Transactions on Smart Cities, 4(10).

[9] Bosri, R., Uzzal, A. R., Al Omar, A., Hasan, A. T., & Bhuiyan, M. Z. A. (2019, August) "Towards a privacy-preserving voting system through blockchain technologies." In 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) (pp. 602-608). IEEE.

[10] Li, C., Xiao, J., Dai, X., & Jin, H. (2021) "AMVchain: Authority management mechanism on blockchain-based voting systems." Peer-to-peer Networking and Applications, 14, 2801-2812

[11] Clarke, R., McGuire, L., Baza, M., Rasheed, A., & Alsabaan, M. (2023) "Online voting scheme using IBM Cloud-Based Hyperledger Fabric with privacy- reservation." Applied Sciences, 13(13), 7905.

[12] Dimitriou, T. (2020) "Efficient, coercion-free and universally verifiable blockchain- based voting." Computer Networks, 174, 107234.

[13] https://www.tutorialsteacher.com/nodejs/what-is-node-package-manager

[14] https://www.edureka.co/blog/developing-ethereum-dapps-with-truffle

[15] https://www.codementor.io/@swader/developing-for-ethereum-getting-started- with-ganache l6abwh62j

[16] S.K. Vivek, et al., E-voting systems using blockchain: an exploratory literature survey, in: Second International conference on Inventive Research in Computing Applications, ICIRCA, 2020, pp. 890–895, https://doi.org/10.1109/ ICIRCA48905.2020.9183185.

[17] https://www.edureka.co/blog/solidity-tutorial/