IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

Crypt-Pay (WEB-3.0)

Sunit Salunkhe[1], Uday Bharadwa[2], Rudra Mondal[3], Shalmal Sakpal[4], Mrs. Poonam Thakre[5]
Department of Artificial Intelligence & Machine Learning[1-5]
Universal College of Engineering, Mumbai, Vasai, India

Abstract: *Embark on an extensive and immersive journey into the fascinating realm of Web 3.0 technologies and methodologies, where the decentralized internet, driven by blockchain and smart contracts, undergoes a paradigm shift, fundamentally reshaping the landscape of modern web development. Navigate through the intricate terrain of Solidity, the robust and versatile programming language empowering secure smart contracts and facilitating seamless automation and trustless interactions within decentralized applications. Delve deep into the pivotal role of MetaMask as an indispensable bridge between conventional browsers and decentralized ecosystems, facilitating secure transactions and interactions with blockchain networks. With a relentless focus on practicality, embark on a comprehensive journey through detailed step-by-step instructions for setting up, coding, integrating, testing, and deploying fully functional Web 3.0 applications. Explore the decentralized nature of blockchain, where immutable blocks linked by cryptographic hashes form a distributed ledger, enabling secure digital asset exchange across global networks. Embrace the innovative potential of Web 3 Wallets, revolutionizing digital wallets with advanced features within the decentralized web and offering users worldwide a plethora of benefits, both digitally and cryptographically, in their interactions with decentralized applications and blockchain networks. Through hands-on exploration, gain insights into the democratizing potential of Web 3.0, fostering a more inclusive and transparent digital economy while also mitigating issues like censorship and data privacy concerns. Discover how Web 3.0 technologies empower individuals to take control of their digital identities and assets, transcending traditional centralized models and paving the way for a more decentralized and resilient internet ecosystem.*

Keywords: Web 3.0 technologies (ReactJS), MetaMask, Solidity, smart contracts

I. INTRODUCTION

Web 3 technology, rooted in the decentralized framework of blockchain, fundamentally transforms cryptocurrency websites by imbuing them with enhanced security, privacy, and user autonomy. This innovative infrastructure decentralizes website operations through distributed networks, mitigating risks associated with central servers and fostering continuous uptime and data integrity. Smart contracts, a cornerstone of Web 3, automate transactions with trustless efficiency, streamlining payment process

An e-wallet, or electronic wallet, is a digital tool that allows individuals to store, manage, and transact with their digital assets, such as cryptocurrencies or digital tokens. To implement an e-wallet using smart contracts and Solidity, we leverage the capabilities of blockchain technology to ensure security, transparency, and decentralization.

Embarking on the guided journey through Web 3.0 technologies unveils a transformative landscape for modern web development. The decentralized internet, powered by blockchain and smart contracts, forms the foundation of this journey. Solidity, a robust programming language, becomes the key to creating secure smart contracts that automate actions within the Web.

While the core functionalities of Web 3.0 applications often lie in smart contract interactions and data management, user experience can be significantly enhanced by incorporating various modules. In my case, I've taken the initiative to implement a scanner module within my React.js application. This scanner module could serve diverse purposes, such as integrating QR code scanning for secure wallet authentication or barcode scanning for product verification on a decentralized marketplace. By leveraging the capabilities of React.js libraries like react-qr-scanner, the scanner module can be seamlessly integrated, providing a smooth and interactive experience for the users.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

3.0 applications. As we dive into Solidity, step-by-step instructions guide us through the process of setting up, coding, integrating, testing, and deploying fully functional applications. MetaMask emerges as a pivotal player, acting as a bridge between traditional browsers and the decentralized environment. Its role is critical in enabling secure interactions with both the application and the blockchain. The emphasis throughout this journey remains on practicality, ensuring a hands-on experience.

Within the decentralized platform of blockchain, multiple peers globally exchange digital assets securely. The interconnected blocks, facilitated by cryptographic hashes, maintain the order of transactions, ensuring a tamper-proof and transparent ledger. This intrinsic feature of blockchain establishes a secure and reliable foundation for the development of Web 3.0 applications.

A significant component of this evolution is the Web 3 Wallet, a digital wallet tailored for the decentralized web. With innovative features and functionalities, it caters to the demands of Web 3.0 environments. The benefits for digital or crypto wallets within the Web 3.0 landscape are profound, ranging from enhanced security measures to increased user control an interoperability. As we navigate this comprehensive journey, the synthesis of Web 3.0 methodologies, Solidity programming, MetaMask integration, and blockchain fundamentals sets the stage for the creation of a new era in modern web development.

The project idea stemmed from recognizing the glaring limitations of traditional web applications in terms of decentralization, security, and handling of cryptocurrencies and smart contracts. Understanding the vulnerabilities of centralized systems and the growing importance of blockchain technology, the concept evolved to harness the potential of Web 3.0. Aiming to bridge the knowledge gap and empower users and developers, the idea focuses on integrating blockchain frameworks, employing Solidity for robust smart contracts, and facilitating seamless cryptocurrency integration. The objective is to offer an innovative solution that not only addresses existing shortcomings but also provides a practical and educational pathway for embracing the transformative capabilities of blockchain in modern web applications. The potential for enhancing security, automating processes, and ensuring trustless transactions became the driving force behind this promising project idea

II. RELATED WORK

Modern web development has seen a surge in innovation since the advent of Web 3.0 technologies, especially around decentralised internet architectures powered by blockchain and smart contracts. This section examines earlier studies and projects that expand our knowledge of Web 3.0 technology and principles.

In [1] There has been a lot of research done about using blockchain technology in web development. Scholars have investigated the application of blockchain technology and smart contracts to provide transparent and safe online environments for a range of activities. Solidity is a powerful programming language designed specifically for creating smart contracts, and it is a key instrument in this field.

In [2]The use of modules, especially in React.js settings, has drawn interest in the field of improving user experience in Web 3.0 applications. Most notably, adding a scanner module to React.js apps provide an adaptable way to improve user experiences. Modules such as react-qr-scanner provide the smooth integration of features like barcode scanning for product verification on decentralised marketplaces or QR code scanning for secure wallet authentication.

Within [3] The purpose of the Practical lessons and guidelines is to make it easier to construct Web 3.0 apps using Solidity and MetaMask. These materials offer detailed instructions on how to install, configure, test, integrate, and code smart contracts in decentralised systems. By acting as a link between conventional browsers and blockchain networks, MetaMask is essential to facilitating safe communication and exchange of information.

In [4] The fundamental features of blockchain technology, such as its ability to enable safe and transparent transactions, have been thoroughly examined in cryptographic hashes preserve the integrity and immutability of transaction records through interconnected blocks, creating a tamper-proof ledger in decentralized systems. This focus on openness and security creates the foundation for the creation of reliable Web 3.0 apps.

Within [5] Web 3 Wallets, a development of digital wallets designed for the decentralised web, offer cutting-edge capabilities and functionality. These wallets provide greater security features, more user control, and compatibility with different blockchain networks to meet the specific requirements of Web 3.0 environments. The use of blockchain technology and the development of decentralised ecosystems are facilitated by the inclusion of such wallets.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

III. PROPOSED MODEL

A smart contract is a program that contains functions and states. Each smart contract runs on a specific address in the Ethereum blockchain. Because smart contracts are an autonomous type of account on Ethereum, they can send transactions and have balance. Ethereum's smart contracts are written in Solidity language. Ethereum has a permissionless ledger type and is open for public use. Its consensus mechanism is proof of work, which is known to be quite slow. Ethereum has a cryptocurrency called Ether. Ether is used to pay for creating and initializing a transaction in the Ethereum blockchain.

System Architecture:

In designing and deploying a modern Web 3.0 blockchain application, meticulous attention to key elements ensures a seamless and secure user experience. The architecture begins with the development of smart contracts in Solidity, adopting a modular and scalable approach to

accommodate diverse functionalities. Decentralized user authentication, leveraging blockchain identities, establishes a robust verification system. Cryptocurrency integration is implemented with a focus on supporting multiple tokens for secure and efficient transactions. The user interface, designed adhering to Web 3.0 principles, prioritizes decentralization, reducing reliance on central servers. A real-time dashboard enhances user transparency by displaying live transaction data through web sockets or similar technologies.

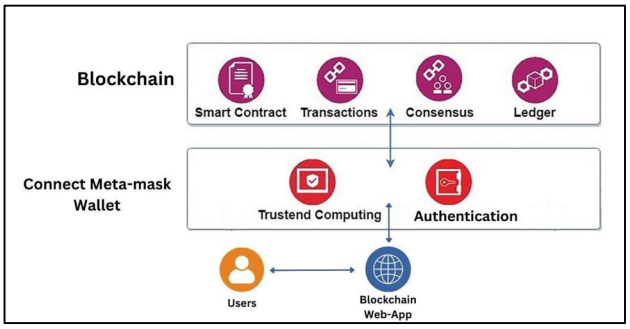


Figure: System Architecture

Entity-Relationship (ER) Diagram:

Users can have multiple MetaMask Accounts, and each MetaMask Account belongs to one user (One-to-Many relationship). Users can initiate multiple transactions, and each transaction is associated with one user (One-to-Many relationship). MetaMask Accounts can be associated with multiple transactions (One-to-Many relationship). The Web App can have multiple smart contracts, and each smart contract belongs to one web app (One-to-Many relationship). Transactions can involve the transfer of tokens. Each transaction involves one or more tokens, creating a Many-to-Many relationship between Transaction and Tokenentities.

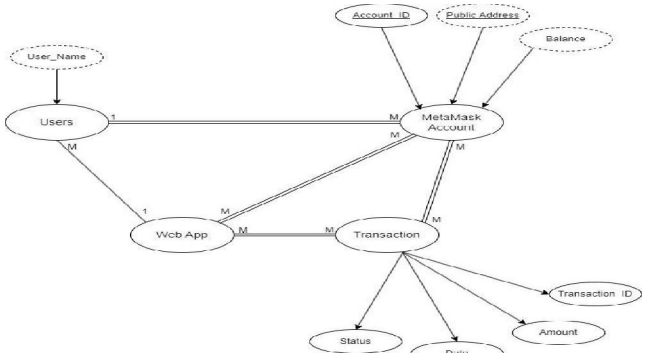


Figure: ER Diagram

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

Data Flow Diagrams (DFD) -Level 0 and 1 :

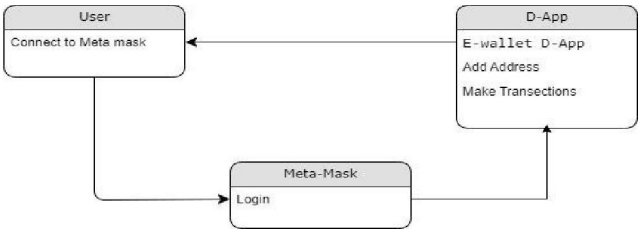


Figure: DFD Level 0

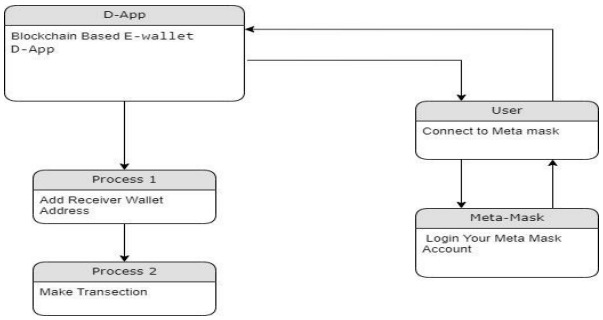


Figure: DFD Level1

UseCase Diagram:

The Use Case Diagram (Figure 1.6) provides a holistic view of the system's functionalities from the stoner's perspective.

"User" represents the external user interacting with your web app.

"System" represents your decentralized web app, which includes the MetaMask integration. The lines connecting the actors to the use cases indicate the interactions or actions they can perform. Here are some use cases related to your web app:

Authenticate: The user can log in or authenticate themselves to the web app using MetaMask
for secure access.

View Content: Users can access and view the content available on your web app.

Make Transactions: Users can initiate transactions using MetaMask, such as sending and receiving cryptocurrencies or interacting with smart contracts.

Interact with DApps: Users can interact with decentralized applications (DApps) through your web app, which may involve various actions like gaming, trading, or voting.

Manage Wallet: Users can manage their MetaMask wallets, which includes checking balances, transferring assets, and managing settings.

Explore Blockchain: Users can explore and access blockchain data or information through.

your web app. This use case diagram provides a high-level overview of the interactions between users and your decentralized web app, highlighting the key functionalities and actions that can be. performed.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

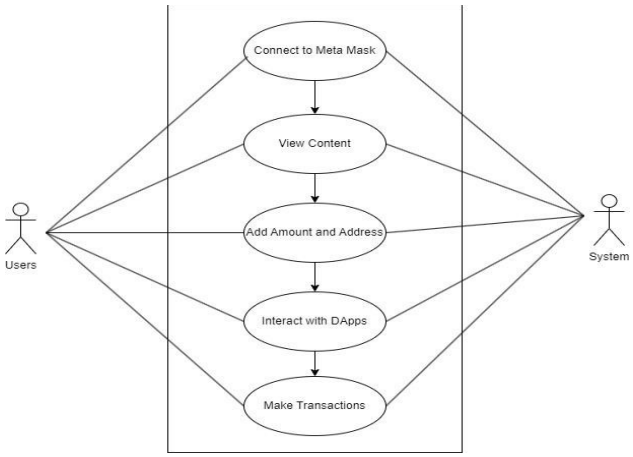


Figure: UseCase Diagram

Activity Diagram:

The Activity Diagram (Figure 1.7)

Start: This is the initial point in your diagram.

User Opens the Web App: The user accesses your decentralized web app in their web browser.

Web App Requests MetaMask Interaction: The web app prompts the user to connect their MetaMask wallet.

User Connects MetaMask: The user clicks on a "Connect with MetaMask" button in the web app, and MetaMask requests permission to connect.

User Grants Permission: The user approves the connection request in their MetaMask extension.

Web App Fetches User Account: The web app retrieves the user's Ethereum address from MetaMask.

User Interacts with the Web App:

User Activity 1: The user initiates some interaction within the app, such as creating a transaction, sending cryptocurrency, or interacting with a decentralized application (DApp) feature.

User Activity 2: Similarly, the user can perform other actions within the app.

Web App Requests Signature: When a user initiates a transaction or action that requires a blockchain confirmation, the web app requests a signature from MetaMask.

User Confirms Transaction in MetaMask: MetaMask prompts the user to review the transaction details and confirm.

End: The process concludes here.

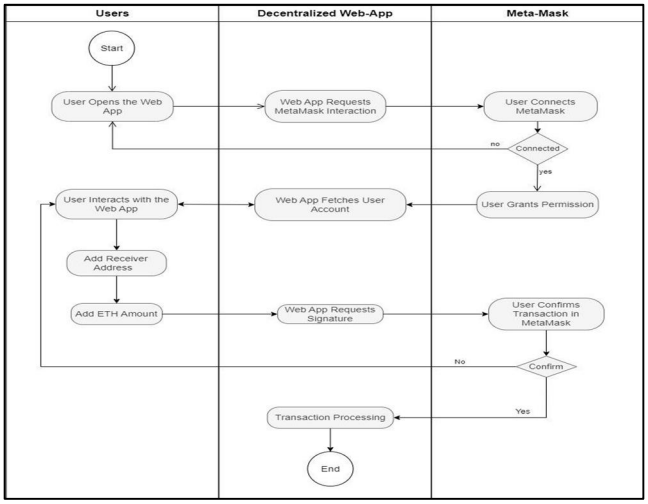


Figure1.7:ActivityDiagram

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

Sequence Diagram

The Sequence Diagram (Figure 1.8) we can observe the sequence diagram. The user opens the web app.

The user clicks the "Make Transaction" button. The web app prompts the user to connect to MetaMask. The user approves the connection. The web app requests transaction data from the user's MetaMask. MetaMask provides the transaction data to the web app. The web app requests MetaMask to sign the transaction. MetaMask shows a UI for the user to confirm the transaction. The user confirms the transaction in MetaMask. MetaMask informs the web app that the transaction is confirmed. The web app broadcasts the transaction to the blockchain network. The blockchain network processes the transaction. The blockchain network acknowledges that the transaction was successful. The web app notifies the user that the transaction is complete. The user closes the web app

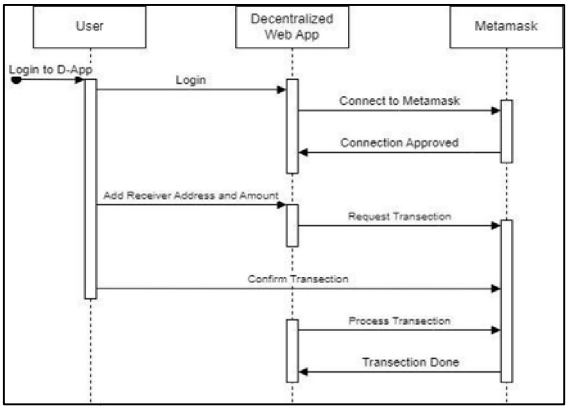


Figure: Sequence Diagram

IV. RESULT AND OUTPUT

WEB VIEW

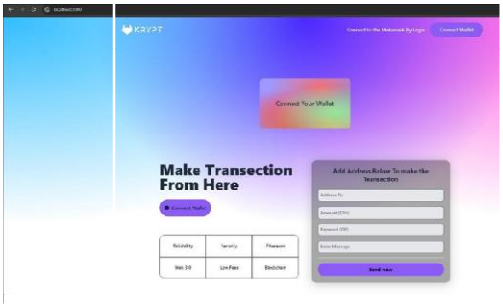


Fig 1. Home Page

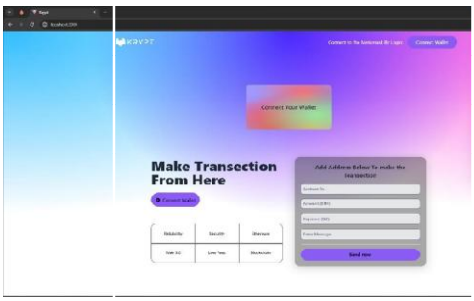


Fig 2. GUI of Connect to MetaMask

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

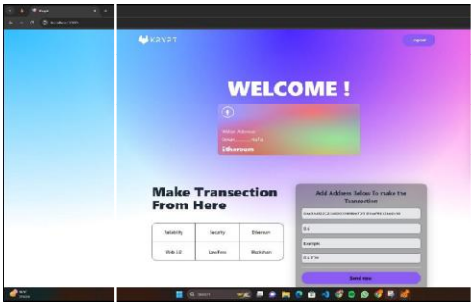


Fig 3. Transfer Ether

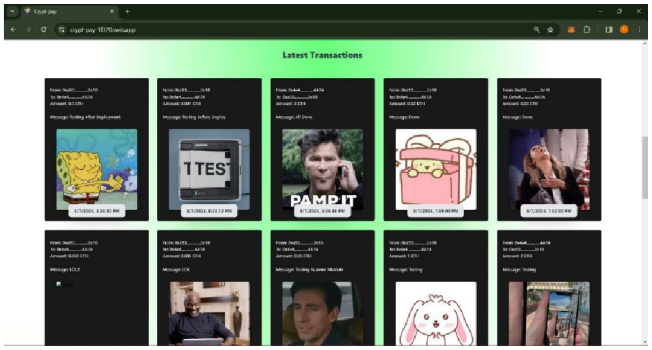
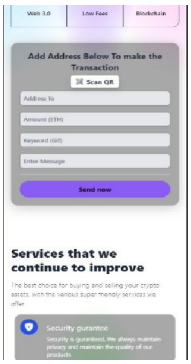
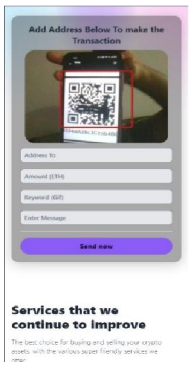
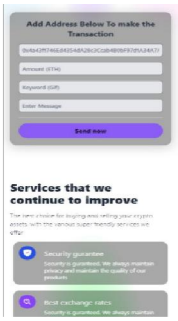


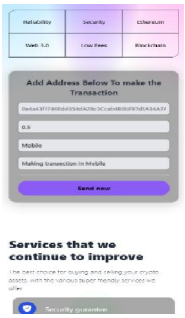
Fig 4. Transaction using GIPHY API

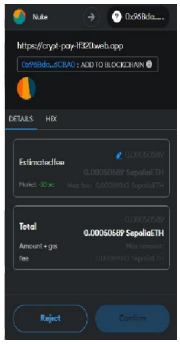
Mobile View











IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, April 2024

V. CONCLUSION

In conclusion, integrating blockchain technology into a wallet system offers a myriad of advantages. The decentralized and secure nature of blockchain ensures that financial transactions are transparent, tamper-resistant, and resistant to fraud. This not only enhances the overall security of the wallet but also fosters trust among users. In the long run, the adoption of blockchain in wallets has the potential to revolutionize traditional financial systems, offering users a more secure, transparent, and efficient means of managing their finances. As technology continues to evolve, it's exciting to anticipate the further innovations and improvements that will shape the future of blockchain-based wallets. Furthermore, the use of smart contracts within the blockchain wallet can automate and enforce contractual agreements, streamlining processes and reducing the likelihood of errors. This automation not only improves efficiency but also enhances user experience.

VI. FUTURE WORK

As technology continues to evolve, it's exciting to anticipate the further innovations and improvements that will shape the future of blockchain-based wallets. Furthermore, the use of smart contracts within the blockchain wallet can automate and enforce contractual agreements, streamlining processes and reducing the likelihood of errors. This automation not only improves efficiency but also enhances user experience.

REFERENCES

[1] “The Integration of Blockchain and AI for Web 3.0: A Security Perspective.” Ieeexplore.ieee.org,ieeexplore .ieee.org/document/10068672. Accessed 31 Oct. 2023.
[2] Petcu, Adrian, et al. “A Secure and Decentralized Authentication Mechanism Based on Web 3.0 and Ethereum Blockchain Technology.” Applied Sciences, vol. 13, no. 4, 9 Feb. 2023,
[3] Lin, Yijing, et al. “A Unified Blockchain-Semantic Framework for Wireless Edge Intelligence Enabled Web 3.0.” ArXiv.org, 26 Oct. 2022, arxiv.org/abs/2210.15130. Accessed 31 Oct. 2023.
[4] Agbozo, Ebenezer, et al. “Applying Multi-Criteria Decision Making to Prioritization of Web 3.0 Development Factors.” E-Business Technologies Conference Proceedings, vol. 3, no. 1, 10 June 2023, pp. 229–232, www.ebt.rs/journals/index.php/conf-proc/article/view/154. Accessed 31 Oct. 2023.
[5] Shreya Sudhakaran, et al. “Blockchain-Based Transparent and Secure Decentralized Algorithm.” Algorithms for Intelligent Systems, 20 Dec. 2019, pp. 327–336,https://doi.org/10.1007/978-981-15-0633-8_32. Accessed 31 Oct. 2023