iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 Issue: IV Month of publication: April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59674

Blockchain based Decentralized Storage System

Prof. Shanti Guru[1], Prof. Yasmin Khan[2], Yash Ashok Gokakkar[3], Shriniket Kulkarni[4], Ashutosh Raj Gupta[5], Suyash Dighe[6], Viraj Sonagra[7]
Department of Computer Engineering, D Y Patil College of Engineering, Akurdi, Pune, India

Abstract: This research introduces a novel decentralized storage system implemented through the integration of blockchain technology. The system is manifested in the form of a website developed using React.js, tailwindcss, and MySQL as the backend database, complemented by the Inter Planetary File System (IPFS) for the secure storage of diverse document formats such as .pdf, .jpg, and .png. Users can securely manage their documents within their respective accounts on the platform. One of the primary features of the system is the ability for users to securely request access to specific documents from other users via the blockchain. This mechanism ensures a transparent and tamper-resistant transaction process, bolstering the security and integrity of document sharing. Additionally, users can seamlessly store their own documents within their personalized accounts, enhancing the platform's user-centric functionality. The utilization of React.js and tailwindcss ensures an interactive and visually appealing user interface, contributing to an enhanced user experience. The backend infrastructure, supported by MySQL, provides a robust and scalable foundation for efficient data management. This research contributes to the field by presenting a comprehensive solution that addresses the challenges associated with centralized storage systems. The decentralized architecture, coupled with blockchain-based access control, not only enhances security but also promotes transparency and accountability in document management.

Keywords: Decentralized Storage, Blockchain, React.js, tailwindcss, MySQL, IPFS, Document Management, Security

I. INTRODUCTION

In an era characterized by an escalating reliance on digital documents and a growing need for secure and transparent data management, the development of decentralized storage systems has emerged as a pivotal area of research and innovation. This paper introduces a pioneering solution, integrating blockchain technology to establish a decentralized storage system that leverages the capabilities of React.js, tailwindcss, MySQL, and the Inter Planetary File System (IPFS)[1]. Traditional centralized storage systems, while prevalent, are not without their shortcomings, particularly in terms of security vulnerabilities and dependence on centralized authorities. This research seeks to address these challenges by proposing a novel decentralized storage framework that not only mitigates security concerns but also introduces an intricate layer of transparency and accountability through blockchain technology. The core architecture of the system is implemented using React.js and tailwindcss, ensuring a dynamic and visually intuitive user interface. The backend infrastructure, powered by MySQL, provides a robust foundation for efficient data management. The incorporation of IPFS for document storage further enhances the system's resilience and reliability[2]. One of the distinguishing features of the proposed system is its utilization of blockchain for access control and document sharing among users. By securely facilitating access requests through a transparent and tamper-resistant blockchain, the system ensures the integrity of document transactions. Users can securely share documents and manage their own files within personalized accounts, thereby fostering a user-centric environment. This research not only contributes to the evolving landscape of decentralized storage systems but also aligns with the broader objective of promoting secure and transparent data management practices. Through a synthesis of blockchain technology and contemporary web development tools, our solution endeavors to redefine the paradigm of document management systems in the digital age. The increasing prevalence of digital documents across various domains has underscored the critical need for secure, transparent, and efficient storage systems. Conventional centralized storage solutions, while widely adopted, exhibit inherent vulnerabilities such as single points of failure and susceptibility to unauthorized access. In response to these challenges, our research pioneers a decentralized storage system that harnesses the potential of cutting-edge technologies, including React.js, tailwindcss, MySQL, and the Inter Planetary File System (IPFS)[2]. The architectural foundation of our system centers on React.js and tailwindcss, delivering an interactive and visually compelling user interface. The backend infrastructure, supported by MySQL, ensures a scalable and resilient platform for robust data management. Augmenting this framework is the integration of IPFS, providing a decentralized and tamper-resistant storage solution for various document formats, ranging from .pdf to .jpg and .png.

A pivotal aspect of our research lies in the incorporation of blockchain technology to enhance the security and transparency of document management.

The blockchain facilitates secure access requests between users, fortifying the integrity of document transactions. Users can securely share documents and maintain their personal files within dedicated accounts, thereby fostering a user-centric ecosystem. By addressing the limitations of centralized storage systems, our research not only contributes to the evolving landscape of decentralized storage but also aligns with broader efforts to instill security and transparency in contemporary data management practices.

II. LITERATURE REVIEWS AND PAST RESEARCHES

The paper proposes a blockchain-based solution for issuing and verifying educational and professional certificates. The methodology involves utilizing a permissionless blockchain, particularly Ethereum, for the storage of certificate hash values. Additionally, a distributed peer-to-peer storage system (IPFS) is employed for storing the actual certificates. Certificates originate from universities/colleges, with their hash values securely stored on the blockchain[3]. This system allows employers to verify certificates using either the certificate ID or transaction hash value. The paper extensively discusses implementation details and highlights the advantages of leveraging blockchain technology for certificate verification.

The paper introduces Meta-Block, a novel cross-layer storage management strategy designed specifically for decentralized blockchain storage applications. The focus lies on enhancing the utilization of storage resources within decentralized storage systems by capitalizing on the unique features of blockchain storage applications. The methodology includes a redesign of the organization of Log Structured Merge Tree (LSM-Tree) and the implementation of a data prefetching strategy aimed at enabling direct storage access.

III. METHODOLOGY

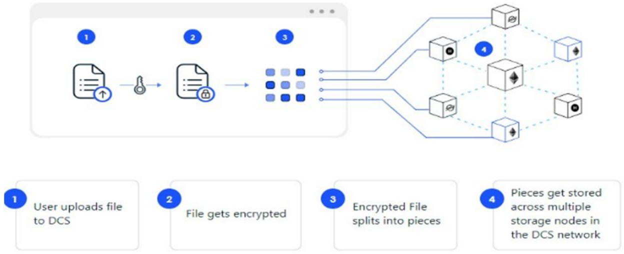

1
2
3
4
User uploads file to DCS
File gets encrypted
Encrypted File splits into pieces
Pieces get stored across multiple storage nodes in the DCS network

Fig. 1. System Architecture

A. User Interface and User Experience

The initial phase of the methodology revolves around crafting a user interface that integrates sleek aesthetics and an intuitive design, ensuring an optimal user experience. This process begins with a comprehensive analysis of user expectations and industry standards to establish a solid foundation for UI development.

1) *User-Centric Approach:* Emphasizing a user-centric approach, user surveys and interviews are conducted to understand preferences, pain points, and expectations. This insight guides the design process to create an interface that resonates with users.
2) *Wireframing and Prototyping:* Leveraging industry-standard tools, wireframing and prototyping are employed to visualize the layout and flow of the UI. This iterative process allows for quick adjustments based on user feedback and ensures the final design aligns seamlessly with user expectations.
3) *Sleek Aesthetics:* The design philosophy prioritizes a clean and modern aesthetic. Using React.js and tailwindcss, a minimalist design is implemented that enhances visual appeal while minimizing clutter, contributing to a sleek and sophisticated UI.
4) *Intuitive Navigation:* To enhance user experience, an intuitive navigation structure is implemented. This involves strategically placing elements, employing clear and concise labels, and optimizing the user journey to enable users to effortlessly navigate through the system.
5) *Responsive Design:* Recognizing the diverse array of devices users may utilize, a responsive design is implemented that ensures a consistent and engaging experience across various screen sizes and resolutions.
6) *User Testing:* Continuous user testing is integrated into the design process. Feedback from real users is invaluable for refining the UI, ensuring that it not only meets aesthetic standards but also aligns with user expectations, making it easy to comprehend and navigate in a single glance.

B. Hashing Algorithms

1) SHA-256

SHA-256, a member of the SHA-2 family of hash functions, stands as a cornerstone in modern cryptography. This algorithm generates a 256-bit (32-byte) hash value, which serves as a digital fingerprint for data integrity verification. Widely regarded for its robustness, SHA-256 has endured rigorous cryptanalysis, emerging as a stalwart in cryptographic applications[4]. Its security stems from its resistance to various cryptographic attacks, including collision and preimage attacks, owing to its design principles and the sheer complexity of the algorithm. Mathematically, SHA-256 operates through a series of bitwise logical operations, including AND, XOR, and NOT, combined with modular addition and rotation functions. The algorithm processes data in fixed-size blocks, iteratively transforming input data into a fixed-size output hash. Its utilization spans across diverse fields, from ensuring the integrity of digital signatures to underpinning the infrastructure of certificate authorities. In essence, SHA-256 stands as a testament to the resilience of cryptographic primitives, safeguarding sensitive data and digital communications in an ever-evolving digital landscape.

2) Bcrypt

Bcrypt, a cryptographic hash function, is renowned for its robustness and resilience against brute-force attacks. It employs a modified version of the Blowfish encryption algorithm, incorporating salt and key stretching techniques to enhance security. Bcrypt produces a fixed-length hash value, typically 192 bits, making it suitable for securely hashing passwords and sensitive data[3]. Its strength lies in its computationally intensive nature, making it resistant to high-speed attacks and parallel processing. Mathematically, bcrypt operates by repeatedly applying the Blowfish encryption algorithm in a loop, with the input data and randomly generated salt serving as inputs. This iterative process effectively slows down the hashing process, thwarting attempts to crack hashed passwords quickly. Due to its effectiveness in thwarting password-based attacks, bcrypt is widely adopted in various security-critical applications, including password hashing in web authentication systems. Its robustness and proven track record make it a favored choice among security professionals seeking to fortify their systems against unauthorized access and data breaches.

C. IPFS

The development of a blockchain-based decentralized storage system represents a paradigm shift in data management, leveraging the combined power of blockchain technology and the Inter Planetary File System (IPFS). At its core, this system operates on the principles of decentralization and immutability, ensuring data integrity and accessibility without reliance on centralized authorities. The process begins with users uploading files onto the IPFS network, where they are broken down into smaller chunks and distributed across a network of nodes. Each chunk is assigned a unique cryptographic hash, which serves as its identifier on the IPFS network[2]. Simultaneously, the metadata associated with the file, including its hash and additional attributes, is recorded on the blockchain. This linkage between the blockchain and IPFS ensures transparency and traceability, as the blockchain records the transaction details and references to the IPFS file addresses.

The algorithm underlying this system involves several key steps. First, the file is divided into smaller chunks using a content-addressable hashing mechanism, such as SHA-256. These chunks are then distributed across the IPFS network using a distributed hash table (DHT), ensuring redundancy and fault tolerance. Meanwhile, the blockchain records the metadata associated with each file upload transaction, including the hash of the file and its IPFS address. Smart contracts govern the interactions between users and the system, ensuring trustless execution of file storage and retrieval operations[6]. Through this innovative approach, the blockchain-based decentralized storage system offers a robust and censorship-resistant solution for securely storing and accessing digital assets, ushering in a new era of decentralized data management.

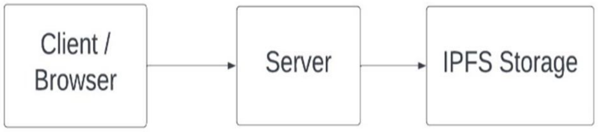


Fig. 2. Client and IPFS interaction

Algorithmic steps for utilizing IPFS:

1) *File Chunking*
   - Break files into smaller chunks.
   - Generate a unique hash (CID) for each chunk.

2) *Distributed Storage*
   - Store chunks across multiple nodes using a DHT.
   - Ensure redundancy and fault tolerance.

3) *Peer-to-Peer Retrieval*
   - Retrieve chunks using a P2P protocol.
   - Directly transfer chunks between nodes.

4) *Merkle DAG*
   - Construct a Merkle DAG to represent file structure.
   - Facilitate efficient retrieval and verification.

5) *Immutable Data*
   - Uploaded files are immutable.
   - Any changes result in new CIDs.

6) *Data Linking and Versioning*
   - Link related files using their CIDs.
   - Enable versioning with distinct CIDs.

D. Blockchain

In the pursuit of creating a blockchain-based decentralized storage system, the integration of blockchain technology stands as a pivotal aspect of the project's architecture. At its core, blockchain serves as the underlying framework that orchestrates the decentralized storage network's operations. Through the utilization of smart contracts, a fundamental component of blockchain technology, seamless interaction with the Inter Planetary File System (IPFS) is facilitated. These smart contracts automate and enforce the rules governing the storage and retrieval of data within the decentralized network[5]. By leveraging the transparency, immutability, and cryptographic security features inherent in blockchain technology, the decentralized storage system ensures trustless and tamper-proof interactions among network participants. Each transaction and data access request is cryptographically secured and recorded on the blockchain, providing a verifiable and auditable trail of data custody and ownership.

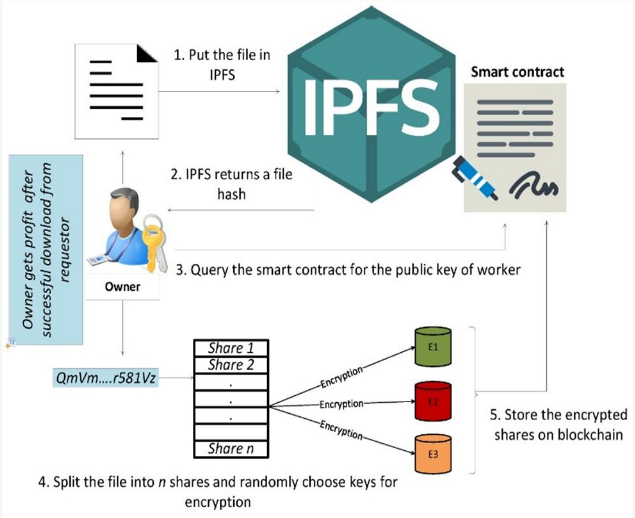

1. Put the file in IPFS
IPFS
Smart contract
2. IPFS returns a file hash
Owner gets profit after successful download from requestor
Owner
3. Query the smart contract for the public key of worker
QmVm....r581Vz
Share 1
Share 2
Share n
Encryption
Encryption
Encryption
E1
E2
E3
5. Store the encrypted shares on blockchain
4. Split the file into n shares and randomly choose keys for encryption

Fig. 3. Workflow for file sharing

Furthermore, the integration of blockchain technology enhances the resilience and reliability of the decentralized storage system. Through the use of consensus mechanisms such as proof of work (PoW) or proof of stake (PoS), the network achieves consensus on the validity of transactions and data updates. This consensus mechanism ensures the integrity and consistency of the stored data, mitigating the risk of data manipulation or unauthorized access. Additionally, the decentralized nature of blockchain technology eliminates single points of failure and central authorities, promoting censorship resistance and data sovereignty. Ultimately, by harnessing the power of blockchain technology, the project endeavors to realize a robust, secure, and censorship-resistant decentralized storage solution, empowering users with greater control over their digital assets and data privacy.

E. Backend

In the development journey of a blockchain-based decentralized storage system, the creation of a robust and scalable backend infrastructure is fundamental. Leveraging Node.js as the backend framework provides flexibility and efficiency in handling asynchronous operations and managing HTTP requests. Through the implementation of RESTful APIs, communication between the frontend and backend layers is facilitated, enabling seamless interaction with the decentralized storage network. The backend system orchestrates various functionalities, including user authentication, file upload and retrieval, and data management operations. By incorporating MySQL as the database management system, the backend ensures data persistence and integrity. Utilizing features such as joins and keys in MySQL enables efficient data querying and retrieval, optimizing the performance of the storage system. The integration of advanced database concepts enhances data organization and accessibility, laying the foundation for a scalable and reliable decentralized storage solution.

Moreover, the implementation of HTTP requests and APIs in the backend layer facilitates seamless integration with external services and applications. Through standardized HTTP protocols, communication between different components of the storage system is streamlined, fostering interoperability and ease of integration. The backend system acts as the bridge between the frontend user interface and the decentralized storage network, translating user actions and requests into blockchain transactions and data interactions. By abstracting the complexities of blockchain technology and decentralized storage operations, the backend layer provides a user-friendly and intuitive interface for interacting with the storage system. Overall, the combination of Node.js, MySQL, HTTP requests, and APIs forms the backbone of the backend infrastructure, empowering the project to realize its vision of a blockchain-based decentralized storage system with seamless user experience and robust functionality.

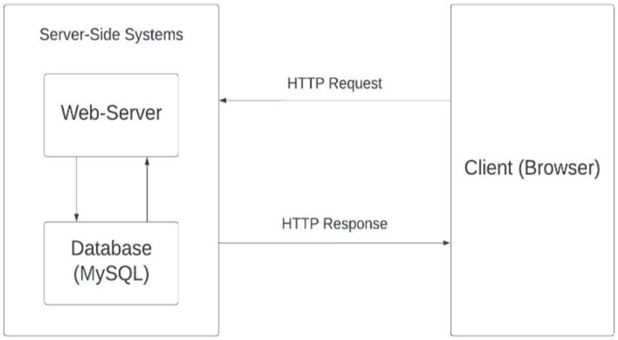


Fig. 4. Http request and response

F. Smart Contract

In the endeavor to develop a blockchain-based decentralized storage system, the integration of smart contracts serves as a cornerstone in ensuring the system's functionality and integrity. Smart contracts, autonomous and self-executing contracts encoded onto the blockchain, play a pivotal role in securely storing the hash values of documents within the decentralized network. These contracts are programmed to execute predefined logic and enforce the rules governing data storage and retrieval operations[3]. In the context of the project, smart contracts act as custodians of document integrity, facilitating transparent and tamper-proof interactions between users and the storage system. By leveraging the immutability and transparency of blockchain technology, smart contracts provide verifiable proof of document ownership and authenticity, enhancing trust and reliability in the decentralized storage ecosystem.

Furthermore, the utilization of smart contracts enhances the security and trustworthiness of the decentralized storage system. Through the deployment of cryptographic hashing algorithms within smart contracts, the integrity of document hash values is safeguarded against tampering or unauthorized modifications. Each document upload transaction triggers the execution of a smart contract, which securely stores the document's hash value on the blockchain. This immutable record ensures the traceability and auditability of document storage activities, fostering accountability and transparency within the decentralized network. Overall, smart contracts serve as the backbone of the blockchain-based decentralized storage system, enabling secure and reliable data storage and retrieval while upholding the principles of decentralization and cryptographic security.

G. Document sharing

In the quest to create a blockchain-based decentralized storage system, the inclusion of document sharing features amplifies the platform's utility and versatility. By incorporating a secure document sharing mechanism, users can seamlessly exchange documents within the decentralized network while ensuring data privacy and integrity. The process begins when a user requests access to a specific document from another registered user. Upon receiving the request, the system initiates a secure communication protocol, wherein the document owner is notified and prompted to approve or deny the access request. If the owner consents to the request, only the document's hash key is provided to the requesting user, ensuring that sensitive document content remains securely stored within the decentralized network. This approach not only streamlines document sharing but also enhances data security by limiting access to document contents solely to authorized users, bolstering trust and confidentiality within the decentralized storage ecosystem.

Furthermore, the implementation of secure document sharing features underscores the system's commitment to user-centricity and privacy protection. By leveraging blockchain technology's transparency and immutability, the document sharing process remains auditable and tamper-proof, mitigating the risk of unauthorized access or data manipulation. Each document sharing transaction is recorded on the blockchain, providing an immutable record of user interactions and access permissions. Through this transparent and accountable approach, the decentralized storage system fosters a culture of trust and collaboration among users, empowering them to securely exchange documents while maintaining control over their data. Ultimately, the integration of document sharing capabilities enhances the platform's functionality and user experience, positioning it as a reliable and secure solution for decentralized document management and collaboration.

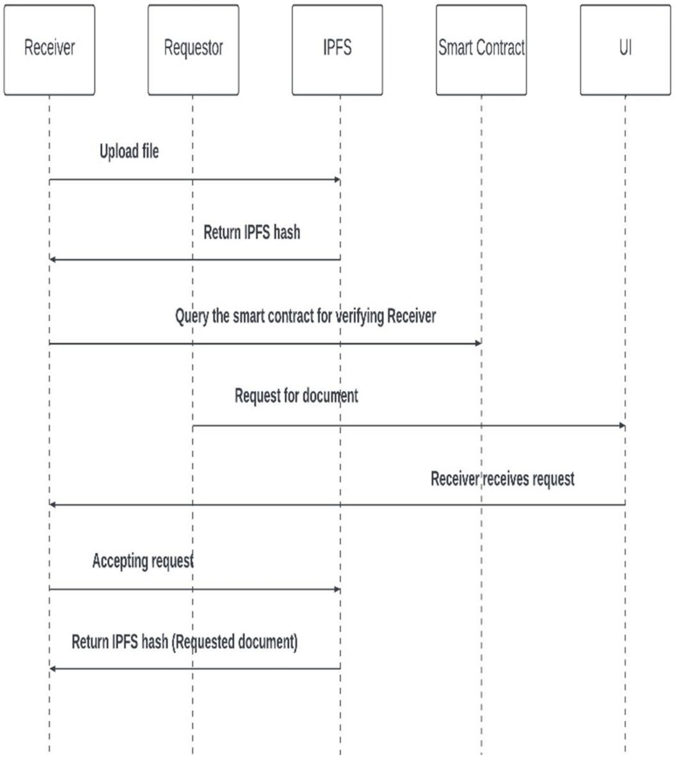

Receiver
Requestor
IPFS
Smart Contract
UI
Upload file
Return IPFS hash
Query the smart contract for verifying Receiver
Request for document
Receiver receives request
Accepting request
Return IPFS hash (Requested document)

Fig. 5. Sequence diagram for document sharing

IV. RESULTS

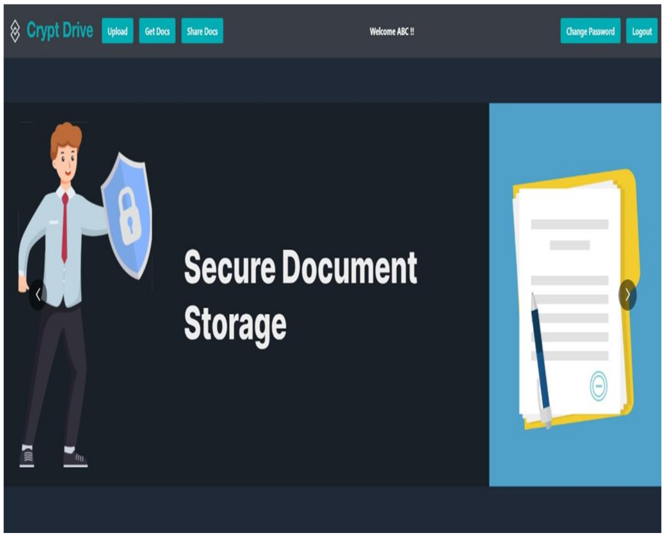
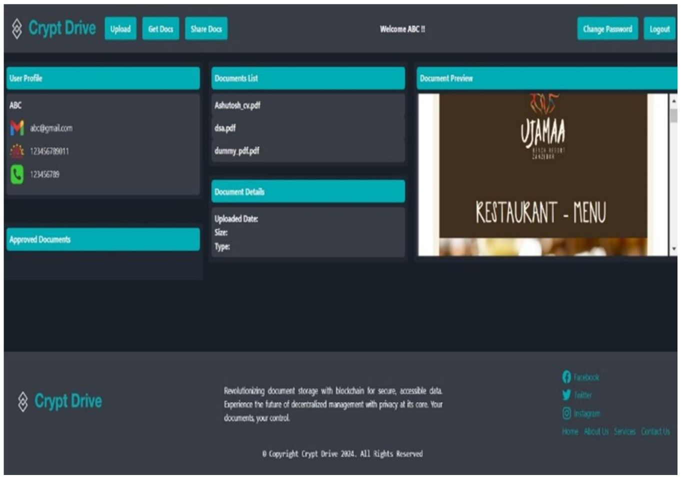


Fig 6. User login home interface



ABC
abc@gmail.com
123456789011
123456789

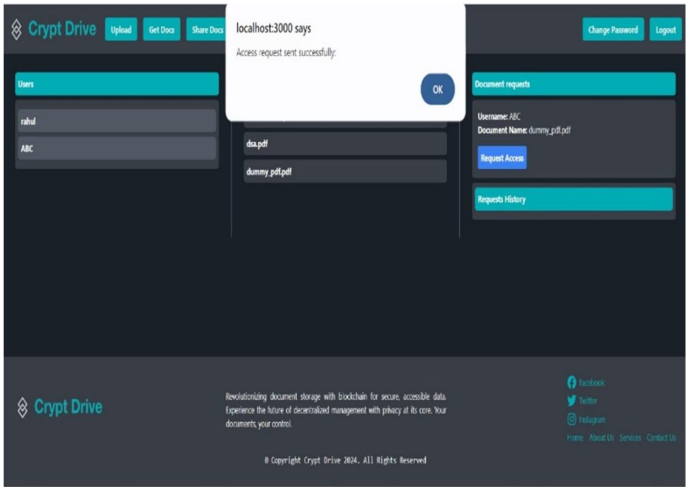
Ashutosh_cv.pdf
dsa.pdf
dummy_pdf.pdf

Uploaded Date:
Size:
Type:

Revolutionizing document storage with blockchain for secure, accessible data. Experience the future of decentralized management with privacy at its core. Your documents, your control.

© Copyright Crypt Drive 2024. All Rights Reserved

Fig 7. User uploaded the documents



localhost:3000 says
Access request sent successfully:

rahul
ABC

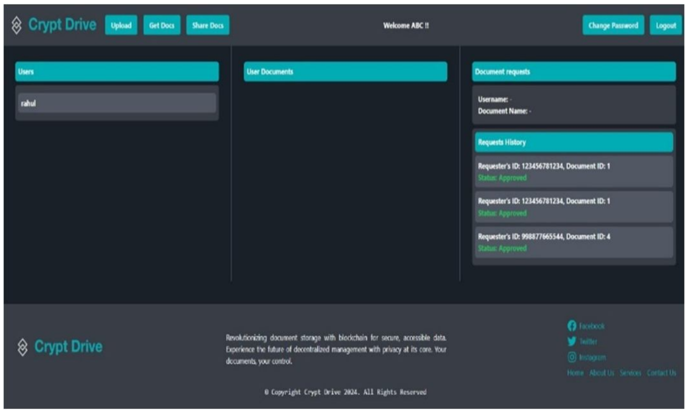
dsa.pdf
dummy_pdf.pdf

Username: ABC
Document Name: dummy_pdf.pdf

Revolutionizing document storage with blockchain for secure, accessible data. Experience the future of decentralized management with privacy at its core. Your documents, your control.

© Copyright Crypt Drive 2024. All Rights Reserved

Fig 8. User “XYZ” requested for user “ABC’s” document



Welcome ABC !!

rahul

Username: -
Document Name: -

Requester's ID: 123456781234, Document ID: 1
Status: Approved

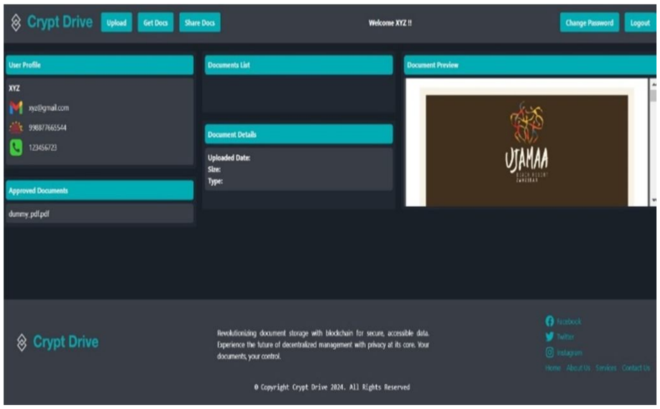
Requester's ID: 123456781234, Document ID: 1
Status: Approved

Requester's ID: 998877665544, Document ID: 4
Status: Approved

Revolutionizing document storage with blockchain for secure, accessible data. Experience the future of decentralized management with privacy at its core. Your documents, your control.

© Copyright Crypt Drive 2024. All Rights Reserved

Fig 9. User "ABC" approved the request



Welcome XYZ !!

XYZ
xyz@gmail.com
998877665544
123456723

dummy.pdf.pdf

Uploaded Date:
Size:
Type:

Revolutionizing document storage with blockchain for secure, accessible data. Experience the future of decentralized management with privacy at its core. Your documents, your control.

© Copyright Crypt Drive 2024. All Rights Reserved

Fig 10. User "XYZ" can now access the document

V. CONCLUSION

This research has successfully introduced a novel decentralized storage system that leverages blockchain technology to address the shortcomings of centralized storage systems. The implementation, in the form of a user-friendly website developed with React.js and tailwindcss, coupled with a robust backend using MySQL and IPFS, provides a secure and efficient platform for document management. The key features of the system, such as secure document storage on IPFS, user-centric document management, and blockchain-based access control, contribute to enhanced security, transparency, and accountability in document sharing. The ability for users to securely request access to specific documents via the blockchain ensures a tamper-resistant transaction process, further bolstering the integrity of document management. The utilization of React.js and tailwindcss not only ensures a visually appealing and interactive user interface but also contributes to an overall improved user experience. The backend infrastructure, supported by MySQL, provides a scalable foundation for efficient data management, ensuring the platform's robustness.

This research makes a significant contribution to the field by presenting a comprehensive solution to the challenges associated with centralized storage systems. The decentralized architecture, coupled with blockchain-based access control, not only addresses security concerns but also promotes transparency and accountability in document management. The proposed system stands as a promising alternative for secure and efficient document storage in a decentralized environment. Future work may involve further optimizations, scalability testing, and real-world implementation to validate the system's effectiveness in diverse scenarios.

REFERENCES

[1] Singh, A., Chauhan, S. P. S., & Goel, A. K. (2023). Blockchain Based Verification of Educational and Professional Certificates. In 2nd International Conference on Computational Systems and Communication (ICCSC).

[2] Habeeb, A., Shukla, V. K., Dubey, S., & Anwar, S. (2022). Blockchain Technology in Digital Certificate Authentication. In 10th International Conference on Reliability, Infocom Technologies, and Optimization (ICRITO).

[3] Wang, Y., Liao, J., Yang, J., Li, Z., Ma, C., & Mao, R. (2023). Meta-Block: Exploiting Cross-Layer and Direct Storage Access for Decentralized Blockchain Storage Systems. IEEE Transactions on Computers, July 2023.

[4] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain mutability: Challenges and proposed solutions," IEEE Trans. Emerg. Topics Comput., vol. 9, no. 4, pp. 1972–1986, Oct./Dec. 2021.

[5] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things:A survey," IEEE Internet Things J., vol. 6, no. 5, pp. 8076–8094,Oct. 2019.

[6] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," Future Gener. Comput. Syst., vol. 88, pp. 173–190, Nov. 2018.

[7] A. Dorri, S. Kanhere, and R. Jurdak, Blockchain for Cyberphysical Systems. Norwood, MA, USA: Artech House, 2020.

[8] V. Dedeoglu et al., "Blockchain technologies for IoT," in Advanced Applications of Blockchain Technology. Singapore: Springer, 2020, pp. 55–89.

[9] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: Alightweight scalable blockchain for IoT security and anonymity," J. Parallel Distrib. Comput., vol. 134, pp. 180–197, Dec. 2019.

[10] H. El-Sayed et al., "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," IEEE Access, vol. 6, pp. 1706–1717, 2017.

[11] W. Yu et al., "A survey on the edge computing for the Internet of Things," IEEE Access, vol. 6, pp. 6900–6919, 2017.

[12] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," IEEE Internet Things J., vol. 6, no. 3, pp. 4719–4732, Jun. 2019.

[13] F.Chang et al., "Bigtable:Adistributed storage systemfor structured data," ACMTrans. Comput. Syst., vol. 26, no. 2, pp. 4:1–4:26, 2008.

[14] LevelDB,2022.[Online].Available: http://code.google.com/p/leveldb

[15] RocksDB, 2022. [Online]. Available: http://rocksdb.org/

[16] Cassandra, 2022. [Online]. Available: http://cassandra.apache.org/

[17] B. F. Cooper et al., "PNUTS: Yahoo!'s hosted data serving platform," Proc. VLDB Endowment, vol. 1, no. 2, pp. 1277–1288, 2008.

[18] Storj Labs, Inc., "Storj: A decentralized cloud storage network framework," 2018. [Online]. Available: https://storj.io/storjv3. Pdf

[19] H. Li, M. Hao, M. H. Tong, S. Sundararaman, M. Bjørling, and H. S. Gunawi, "The case of FEMU: Cheap, accurate, scalable andextensible flash emulator," in Proc. 16th USENIX Conf. File Storage Technol., 2018, pp. 83–90.

[20] M. Bjørling, J. Gonzalez, and P. Bonnet, "LightNVM: The Linux open-channel SSD subsystem," in Proc. 15th USENIX Conf. File Storage Technol., 2017, pp. 359–374.

[21] L. Lu, T. Pillai, A. Arpaci-Dusseau, and R. Arpaci-Dusseau, "WiscKey: Separating keys from values in SSD-conscious storage," in Proc. 14th USENIX Conf. File Storage Technol., 2016, pp. 133–148.

[22] F. Mei, Q. Cao, H. Jiang, and L. Tian, "LSM-tree managed storage for large-scale key-value store," IEEE Trans. Parallel Distrib. Syst., vol. 30, no. 2, pp. 400–414, Feb. 2019.

[23] S.-H. Chen, Y. Liang, and M.-C. Yang, "KVSTL: An application support to lsm-tree based key-value store via shingled translation layer data management," IEEE Trans. Comput., vol. 71, no. 7, pp. 1598–1611, Jul. 2022.

[24] M. Ali, J. Nelson, R. Shea, and M. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in Proc. USENIX Annu. Tech. Conf., 2016, pp. 181–194.