PROJECT REPORT

On

“Decentralized File Storage Application using IPFS and Ethereum”

Submitted by :-

Sr. No.	Name	Roll Number	Branch, Sem
1	Prachi Nagdeve	20118068	IT, 8^{th} Sem
2	Rohit Kumar	20118078	IT, 8^{th} Sem
3	Aryan Sabat	20118016	IT, 8^{th} Sem
4	Sanidhya Patel	20118085	IT, 8^{th} Sem

Under the supervision of :-

Dr. Govind Prasad Gupta

National Institute of Technology, Raipur

Acknowledgement

We are grateful to Dr. Govind Prasad Gupta, Assistant Professor, Department of Information Technology, NIT Raipur, for his proficient supervision on the project work on "**Decentralized File Storage Application using IPFS and Ethereum**". We are very thankful to you Sir for your guidance and support.

20118068 Prachi Nagdeve

20118078 Rohit Kumar

20118016 Aryan Sabat

20118085 Sanidhya Patel

INDEX

1. Abstract

In response to the growing demand for secure, decentralized file storage solutions, this project introduces a novel application leveraging the Interplanetary File System (IPFS) and Ethereum blockchain technology. The objective is to create a robust, censorship-resistant platform for storing and accessing files, ensuring data integrity and user privacy. Through the integration of React, CSS, and JavaScript for frontend development, coupled with Ethereum smart contracts written in Solidity and IPFS for decentralized storage, the application provides users with a seamless and secure experience. The project methodology encompasses frontend design, smart contract development, IPFS integration, testing, and deployment, culminating in a fully functional decentralized file storage application. The findings demonstrate the viability and potential of blockchain-based decentralized storage systems in addressing contemporary challenges of data security and sovereignty.

2. Introduction

In today's digital age, where data has become the lifeblood of modern society, the centralized nature of traditional file storage systems presents significant vulnerabilities and limitations. Centralized servers, owned and controlled by single entities, are susceptible to a myriad of risks, including cyberattacks, data breaches, and unauthorized access. Moreover, these systems often involve intermediaries that exert control over user data, leading to concerns regarding privacy, censorship, and data ownership. In response to these challenges, decentralized technologies have emerged as a promising alternative, offering a paradigm shift towards a more secure, transparent, and democratic model of data storage and management. Among these technologies, blockchain and the Interplanetary File System (IPFS) have garnered considerable attention for their potential to revolutionize the way data is stored, accessed, and shared.

Blockchain, best known as the underlying technology behind cryptocurrencies like Bitcoin and Ethereum, provides a decentralized and immutable ledger that records transactions securely across a network of nodes. Smart contracts, self-executing contracts with predefined conditions written in code, enable programmable and transparent interactions, facilitating a wide range of applications beyond financial transactions. IPFS, on the other hand, is a distributed protocol designed to create a peer-to-peer network for storing and sharing hypermedia content in a decentralized manner. Unlike traditional HTTP-based protocols, which rely on centralized servers to host content, IPFS uses content-addressing to retrieve data from any node in the network, ensuring redundancy, availability, and censorship resistance. This project seeks to harness the combined power of blockchain and IPFS to develop a decentralized file storage application that addresses the shortcomings of centralized storage systems. By leveraging React, CSS, and JavaScript for frontend development, the application aims to provide users with a seamless and intuitive interface for uploading, accessing, and managing their files securely.

Through the implementation of Ethereum smart contracts written in Solidity, the application establishes transparent and auditable access control mechanisms, ensuring that only authorized users can modify or access specific files. Integration with IPFS enables decentralized storage and retrieval of files, while also enhancing data integrity and availability.

In the subsequent sections of this report, we will delve into the methodologies, technologies used, implementation details, challenges encountered, and insights gained throughout the development of the decentralized file storage application. By exploring the potential and implications of decentralized technologies in the context of file storage, this project aims to contribute to the ongoing discourse surrounding data sovereignty, privacy, and decentralization in the digital age.

3. Related Works

Previous research and projects in decentralized file storage systems, blockchain-based applications, and integration of IPFS with Ethereum provide valuable insights into the design choices, challenges, and potential improvements for the current project. Hao et al. [1] investigated a storage scheme integrating blockchain and IPFS for tracking agricultural products. Real-time data on product quality, including video and picture data, are collected by sensors throughout manufacturing, processing, and logistics. Data is first parsed and encapsulated by a private server before being written to IPFS, with the hash address stored on the blockchain. However, the absence of direct data storage on IPFS poses risks, such as data loss in case of server failure, and lacks a keyword search function for quick information retrieval. Rajalakshmi et al. [2] proposed a framework for access control in research records, combining blockchain, IPFS, and traditional encryption methods. Verification metadata from IPFS is stored on the blockchain using Ethereum smart contracts, ensuring tamper-proof record-keeping. However, the scheme lacks keyword search functionality and only stores PDF files, limiting information accessibility. Vimal et al. [3] introduced a method to enhance the efficiency of P2P file-sharing using IPFS and blockchain, incorporating trustworthiness and proximity awareness during file transfer. Miners are incentivized for successful resource transfer. While this study emphasizes security and IPFS-based incentives, it primarily focuses on file transfer services. Chen et al. [4] proposed a more efficient P2P file system by addressing the high-throughput issue for individual IPFS users through a novel zigzag-based storage model. This model enhances IPFS block storage by considering data reliability, availability, and storage overhead for service providers. Wang et al. [5] presented a video surveillance system leveraging permissioned blockchains, edge computing, and IPFS technology. Edge computing processes wireless sensor data, while IPFS enables large-scale video data storage. Although this system utilizes advanced technologies, it primarily focuses on video surveillance applications. Sun et al. [6] proposed a blockchain-based secure storage and access scheme for electronic medical records in IPFS, ensuring necessary access while preserving retrieval efficiency. Medical data is encrypted before storage on IPFS, and hash values are broadcasted to the blockchain. However, the scheme requires encryption of IPFS values before blockchain storage, potentially impacting efficiency.

4. Key Technologies Used:

4.1. React, CSS, and JavaScript:

These technologies were employed for frontend development, enabling the creation of an intuitive and responsive user interface. React provides a robust framework for building modular and interactive components, while CSS and JavaScript facilitate styling and functionality implementation, respectively.

4.2. IPFS (Interplanetary File System):

IPFS serves as the decentralized storage protocol for storing and retrieving files in the application. It operates as a peer-to-peer hypermedia protocol, distributing content across a network of nodes rather than relying on a central server. IPFS addresses files using content-based addressing, where each file is assigned, a unique cryptographic hash based on its content. This ensures data integrity and enables efficient file retrieval through the network.

4.3. Ganache:

Ganache is utilized as a local Ethereum blockchain for development and testing purposes. It provides a simulated blockchain environment where smart contracts can be deployed and tested without incurring actual transaction costs. Ganache offers features such as instant mining, configurable accounts, and simulated gas costs, facilitating rapid iteration and testing during development.

4.4. Truffle:

Truffle is an Ethereum development framework used for writing, testing, and deploying smart contracts. It streamlines the process of smart contract development by providing tools for contract compilation, migration, and testing. Truffle's built-in development server simplifies interaction with the Ethereum blockchain, enabling seamless integration with frontend applications.

4.5. Ethereum:

Ethereum serves as the underlying blockchain platform for executing smart contracts and managing transactions. It is a decentralized platform that enables developers to build and deploy decentralized applications (DApps) utilizing smart contracts. Ethereum's programmable blockchain architecture supports the execution of arbitrary code, enabling the implementation of complex business logic within smart contracts.

4.6. Solidity:

Solidity is the smart contract language used to define the logic and behaviour of the decentralized application (DApp). It is specifically designed for writing Ethereum smart contracts, enabling developers to implement complex business logic securely and efficiently. Solidity's syntax is similar to that of JavaScript, making it accessible to developers familiar with web development.

4.7. Smart Contracts:

Smart contracts are self-executing contracts with the terms of the agreement directly written into code. In the context of this project, smart contracts are used to define the rules and logic governing the decentralized file storage application. They handle functions such as file storage, access control, and authentication, ensuring transparent and transparent interactions between users.

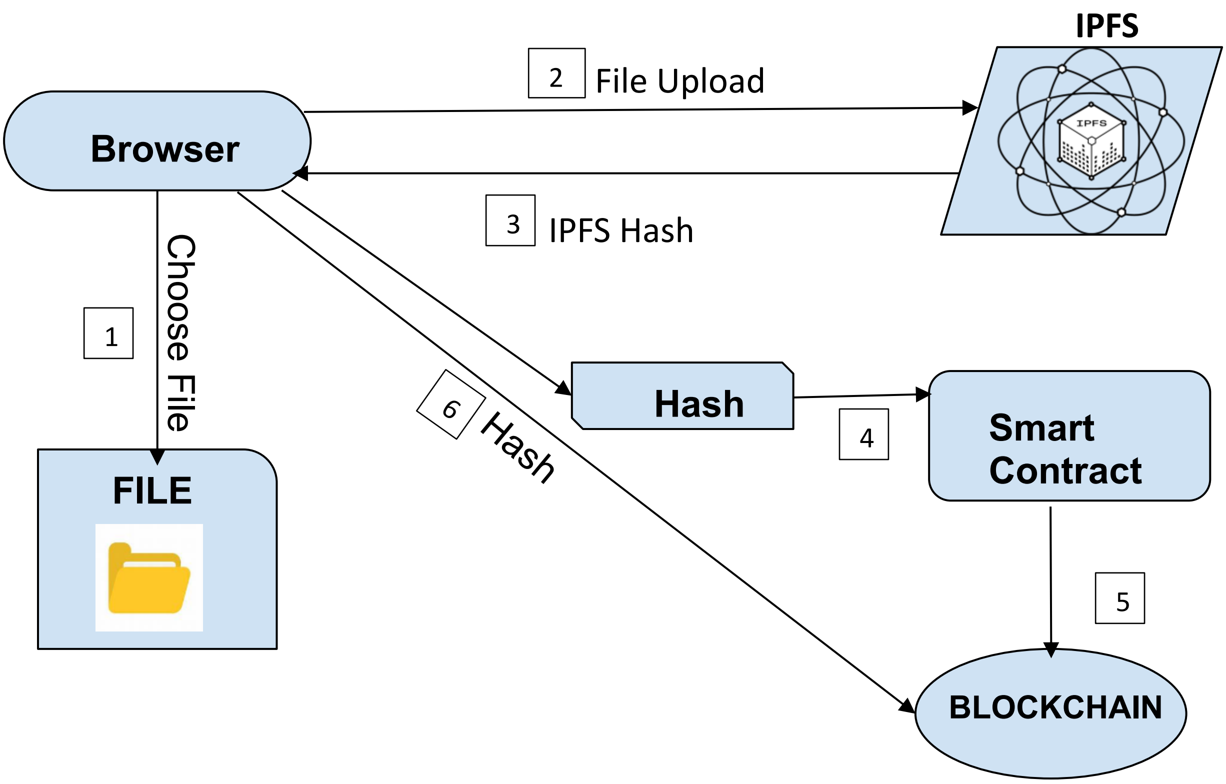
4.8. Node.js:

Node.js is utilized for server-side development and deployment. It provides a runtime environment for executing JavaScript code outside of a web browser, enabling backend functionality and interactions with the Ethereum network. Node.js facilitates communication with IPFS nodes, Ethereum clients, and smart contracts, enabling seamless integration of decentralized technologies into the application architecture.

These key technologies collectively enable the development of a decentralized file storage application with features such as secure file storage, authentication, access control, and decentralized execution, leveraging the capabilities of IPFS and Ethereum blockchain.

5. Methodology

By following this methodology, a decentralized file storage application was successfully developed, combining the capabilities of IPFS for decentralized storage and Ethereum smart contracts for secure and transparent transaction execution. The integration of React provided a user-friendly interface, enhancing the overall usability and accessibility of the application.



5.1. Installation and Configuration of IPFS:

The methodology began with the installation and configuration of the IPFS desktop version, where the IPFS configuration file was customized to include essential access control methods like "PUT", "GET", and "POST". Following configuration adjustments, the IPFS node was initialized using the "ipfs init" command, and subsequently started locally via "ipfs daemon". This setup provided the foundational infrastructure for decentralized file storage, ensuring that the IPFS protocol was operational on the local system and configured to accommodate the required access control methods for seamless interaction within the decentralized application environment.

5.2. Smart Contract Development:

A pivotal stage involved crafting a smart contract using Solidity, Ethereum's smart contract language. This contract delineated the operational framework governing the decentralized file storage application, encapsulating essential functions for file storage, retrieval, and access control. Through meticulous coding, the smart contract was tailored to ensure seamless interaction between users and the decentralized network, guaranteeing integrity and security throughout file transactions.

5.3. Installation and Compilation with Truffle:

Truffle, an Ethereum development framework, was installed to streamline smart contract development tasks. With Truffle, the smart contract underwent compilation using the command "truffle compile", ensuring syntactical correctness and generating bytecode for deployment. This step facilitated efficient management and organization of the smart contract codebase, enabling seamless integration with other components of the decentralized file storage application.

5.4. Deployment with Ganache:

Ganache, a local Ethereum blockchain, was installed to serve as the deployment environment for the smart contract. The Truffle configuration file was adjusted to connect with the Ganache server running locally. Subsequently, the smart contract was deployed to the Ganache blockchain using the command "truffle migrate --reset", ensuring that the contract's functionality was successfully deployed and accessible within the local blockchain environment provided by Ganache.

5.5. Frontend Development with React:

The frontend interface of the decentralized file storage application was crafted using the React framework, enabling efficient and modular development of user-facing components. Through seamless integration with the deployed smart contract and IPFS, users were provided with a user-friendly interface where they could effortlessly interact with the application. By incorporating the addresses of the smart contract and IPFS, users could securely upload documents, triggering the generation of confirmation messages displaying the IPFS hash upon successful uploads. Additionally, a straightforward retrieval process was implemented, allowing users to effortlessly access documents stored on IPFS by simply clicking a designated button, which seamlessly reflected the retrieved document on the UI.

5.6. Integration of Smart Contract and IPFS within React Application:

In the React application, seamless integration of the smart contract and IPFS addresses was achieved to enable file storage and retrieval functionalities. Upon user interaction, such as uploading a document, the application dynamically interacted with the deployed smart contract and IPFS network. This integration allowed for the generation of real-time confirmation messages upon successful document uploads, accompanied by the corresponding IPFS hash. Similarly, users could trigger the retrieval of documents by interacting with the UI, prompting the application to fetch the document from IPFS and display it within the interface. This cohesive integration ensured a smooth and intuitive user experience, enhancing the overall functionality and usability of the decentralized file storage application.

6. Conclusion

The development of a decentralized file storage application utilizing IPFS and Ethereum blockchain technology represents a significant advancement in decentralized computing. Through the integration of key technologies such as React, CSS, JavaScript, IPFS, Ganache, Truffle, Ethereum, Solidity, smart contracts, and Node.js, we have successfully created a robust platform that addresses the limitations of centralized file storage systems.

The installation and configuration of IPFS provided a decentralized storage protocol, ensuring data availability and redundancy across a distributed network of nodes. Smart contract development using Solidity facilitated the implementation of file storage logic and access control mechanisms, ensuring secure and transparent interactions within the decentralized application.

The utilization of Truffle for smart contract compilation and deployment with Ganache enabled seamless integration with the Ethereum blockchain, allowing for efficient testing and deployment of smart contracts in a local development environment. Frontend development with React resulted in an intuitive user interface, enabling users to upload and retrieve files stored on IPFS with ease.

Integration of smart contracts and IPFS within the React application facilitated seamless communication and interaction between the frontend and backend components of the decentralized file storage application. Users could upload files to IPFS, trigger transactions on the Ethereum blockchain, and retrieve files stored on IPFS through a user-friendly interface.

In conclusion, the developed decentralized file storage application demonstrates the potential of blockchain and IPFS integration in addressing the challenges of centralized file storage systems. Moving forward, further enhancements and optimizations can be explored to enhance scalability, interoperability, and usability, paving the way for a more decentralized and resilient internet infrastructure.

7. References

[1] Hao, J.; Sun, Y.; Luo, H. A Safe and Efficient Storage Scheme Based on BlockChain and IPFS for Agricultural Products Tracking.J. Comput. 2018, 29, 158–167.
[2] Rajalakshmi, A.; Lakshmy, K.V.; Sindhu, M.; Amritha, P. A blockchain and IPFS based framework for secure Research record keeping. Int. J. Pure Appl. Math. 2018, 119, 1437–1442.
[3] Vimal, S.; Srivatsa, S.K. A new cluster P2P file sharing system based on IPFS and blockchain technology. J. Ambient. Intell Hum. Comput. 2019, 1–8.
[4] Chen, Y.; Li, H.; Li, K.; Zhang, J. An improved P2P file system scheme based on IPFS and Blockchain. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2652–2657.
[5] Wang, R.; Tsai, W.-T.; He, J.; Liu, C.; Li, Q.; Deng, E. A Video Surveillance System Based on Permissioned Blockchains and Edge Computing. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 27 February–2 March 2019; pp. 1–6.
[6] Sun, J.; Yao, X.; Wang, S.; Wu, Y. Blockchain-Based Secure Storage and Access Scheme For Electronic Medical Records in IPFS. IEEE Access 2020, 8, 59389–59401.