Blockchain Mini-project

Title: Decentralized Cloud Storage

Madhusudan M : PES1UG22CS319

Mohith M Jogi : PES1UG22CS359

Tarun T R : PES1UG22CS649

Monish S : PES1UG22CS361

Introduction:

DStore is a decentralized application (dApp) that attempts to change the way digital files are stored and shared. It is completely governed by blockchain technology to ensure the demand for security, transparency. The integration of Ethereum smart contracts powered by decentralized storage networks such as IPFS (InterPlanetary File System) allows DStore to provide a non-reputable, immutable file handling environment.

The core focus of DStore is all about getting rid of weaknesses any centralized storage system may possess, such as single points of failure, privacy concerns, or lack of transparency. DStore holds complete user control over their data, with integrity, privacy, and reliability guaranteed by the immutable blockchain ledger, cooperating with distributed file storage provided by IPFS.

On the backend, the logic for file ownership, access permissions, and sharing mechanisms is managed by smart contracts. These Ethereum smart contracts automatically enforce access control and data-sharing policies with transparency, minimizing the need to involve any third-party trusted entities.

The front end of DStore combines a modern-looking, user-friendly interface built with a React framework. This gives an intuitive experience for putting files on and retrieving them, storing them, sharing them-all this occurs behind the scenes while blockchain interaction takes place, requiring no technical capabilities on behalf of the user.

Problem Statement

Traditional file storage paradigms expose users to threats such as data breaches, lack of transparency, and a centralized system of control. All of these systems have a single point of failure and thus exist in an environment prone to hacking activities and data loss. Furthermore, since data is stored on centralized servers owned by a third party, the users themselves have almost no control over their own data. The Dstore project solves these issues by providing a decentralized, secure, and transparent framework for storing and sharing files.

Blockchain Implementation Details

The Dstore project utilizes the Ethereum blockchain to manage file access and permissions through smart contracts. Key implementation details include:

Smart Contracts:

The Upload.sol contract, written in Solidity, takes care of uploading files, access control, and sharing functions.

Decentralized Storage:

Files are stored on the InterPlanetary File System (IPFS), ensuring data integrity and availability at all times.

Frontend Integration:

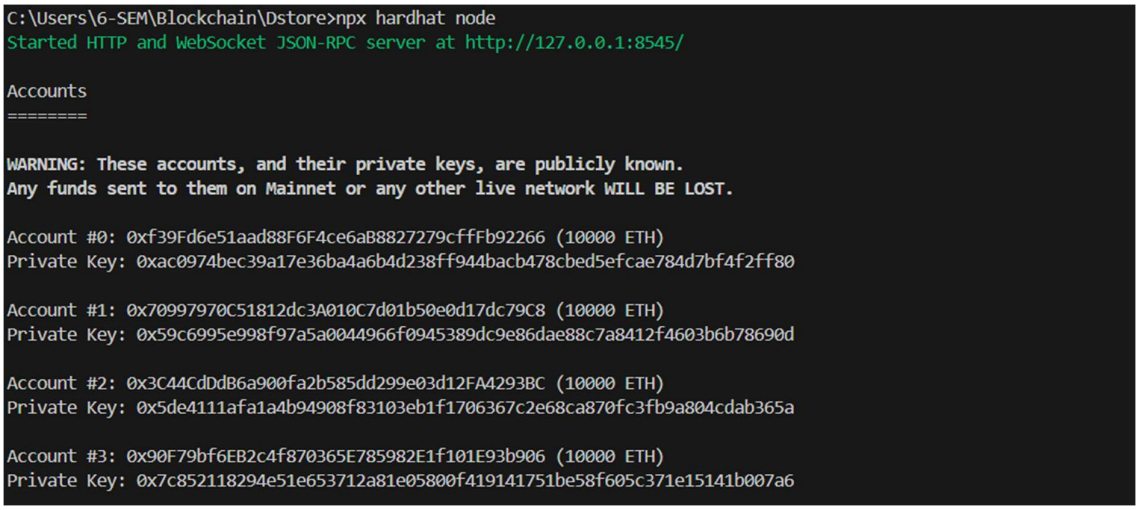
The React-based frontend interacts with the blockchain through libraries, mainly ethers.js, allowing users to upload files, change permissions, and view access logs.

MetaMask Integration:

Users connect to their Ethereum wallets to sign transactions safely and interact with the smart contracts.

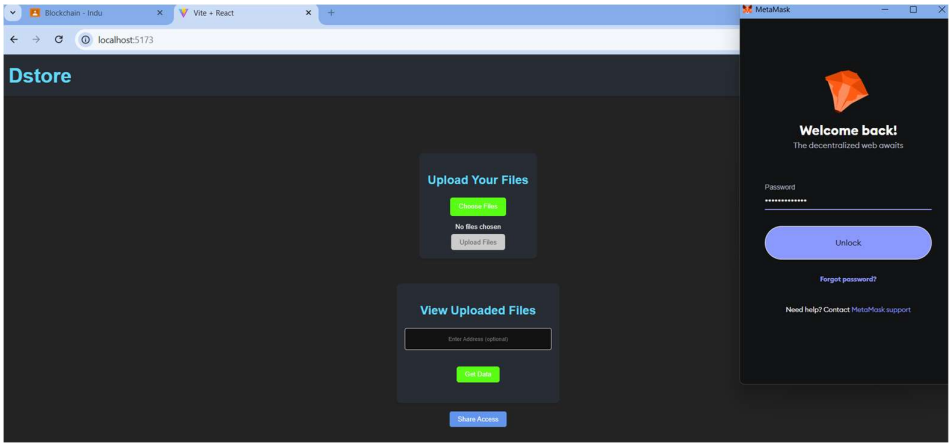
Screenshots

Running local blockchain:

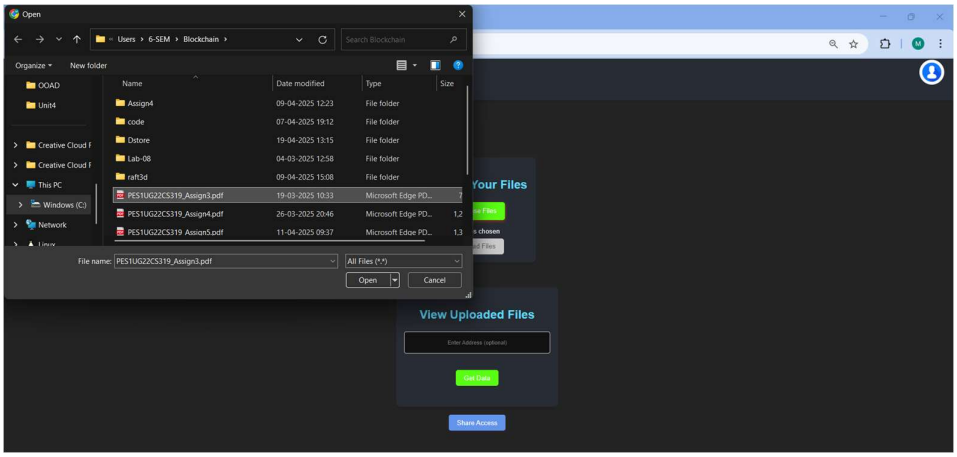


Deploying smart contract to local blockchain:

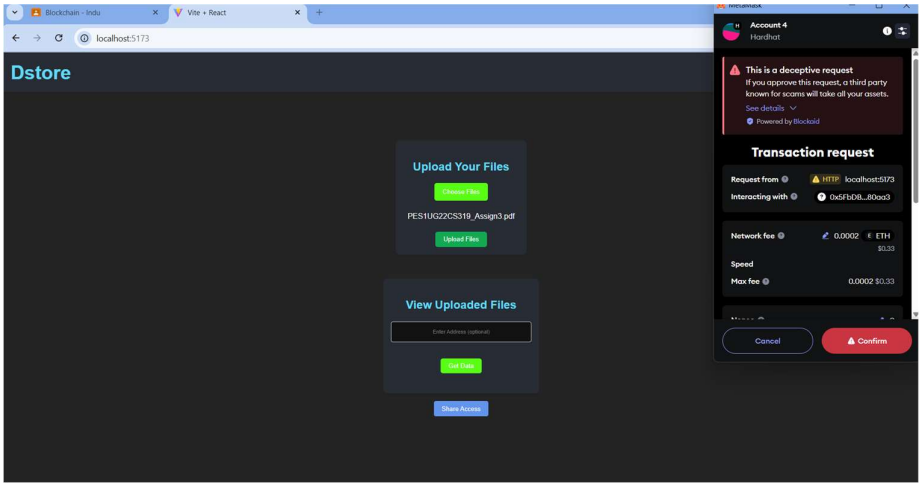
Connecting Metamask wallet:



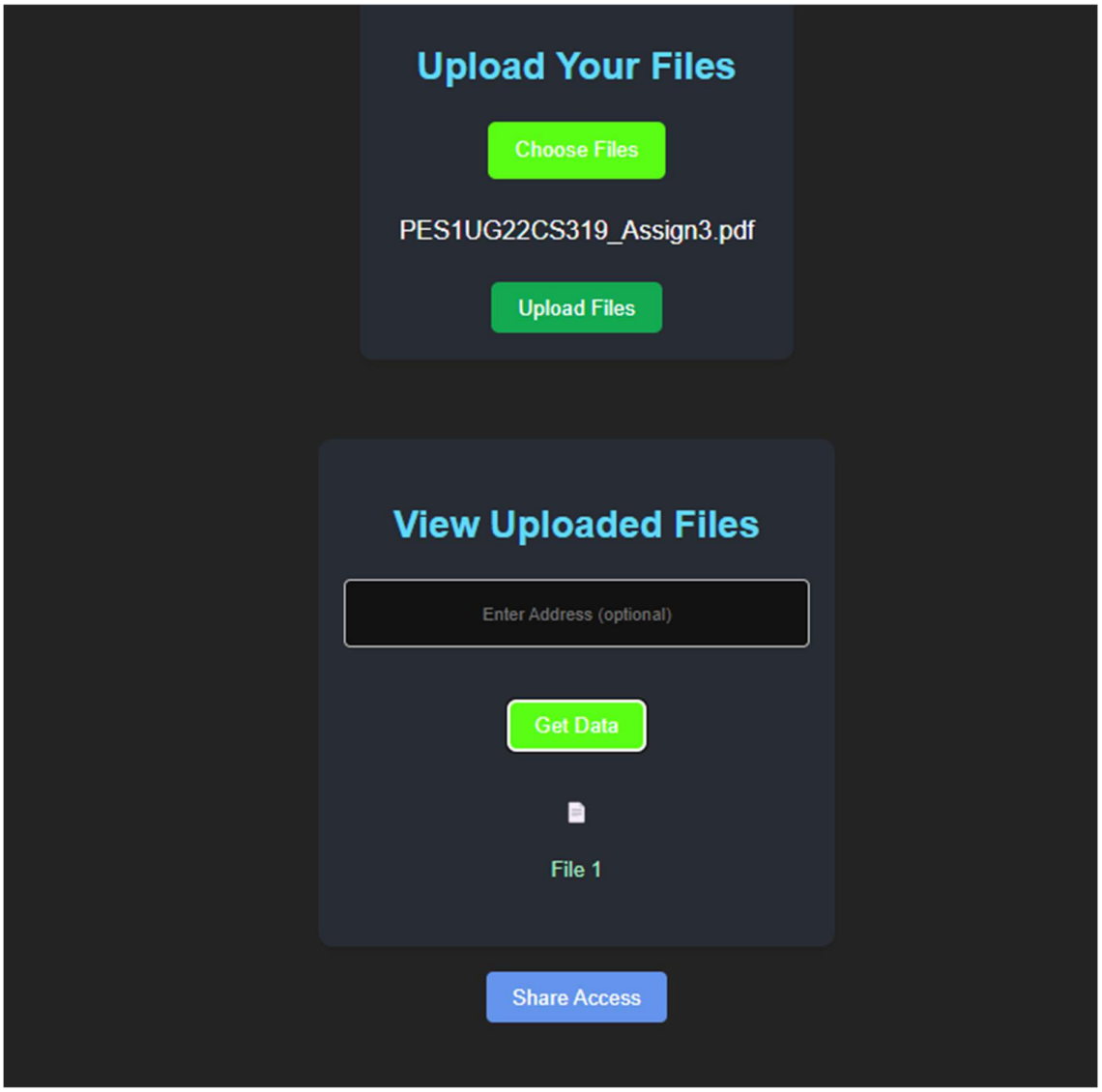
Choosing file to store:



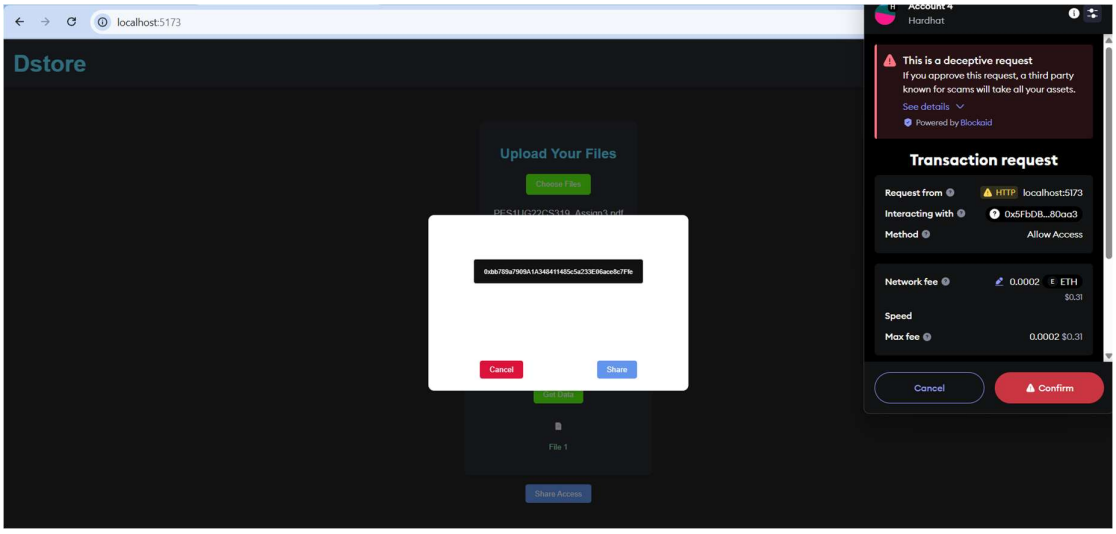
Uploading file:

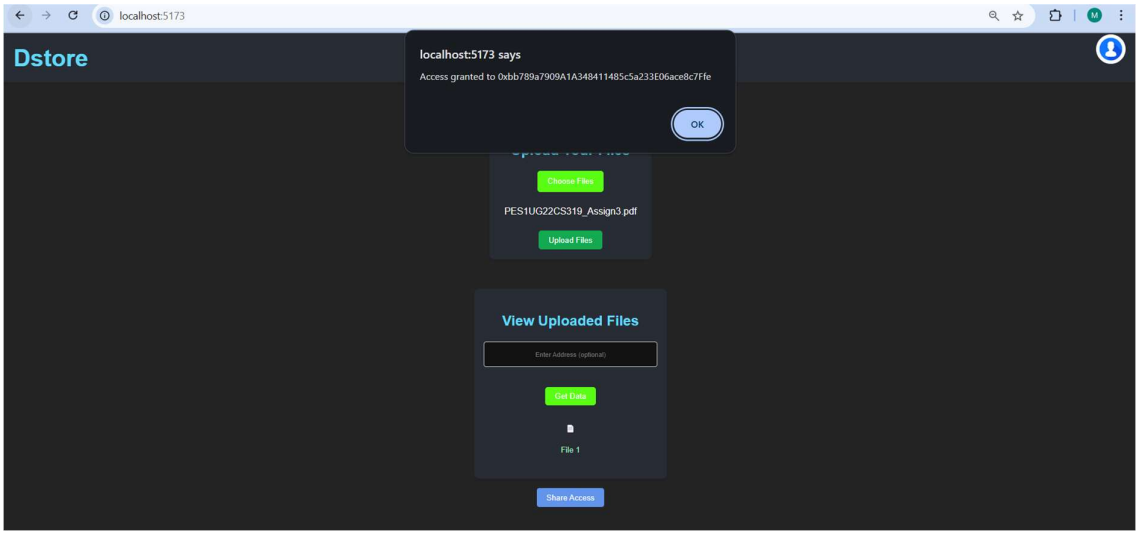


Retrieving files:



Sharing file access:





Future Enhancement

Improved User Interfaces:

Modifying front-end designs for better usability and accessibility.

Advanced Access Control:

Introducing role-based access permissions and time-bound file sharing.

Mobile Application:

Building a mobile app to let users access features of the platform on-the-go.

Tokenization:

Implementing a token-based incentive structure for rewarding users who contribute to the network.