PROJECT- DECENTRALISED FILE STORING AND FILE SHARING PLATFORM USING BLOCKCHAIN

ARHAM AKHTAR,
ROLLNO. 19223017,
MCA 4TH SEM

Introduction- The internet we know today is all centralized, and companies are investing heavily in huge server farms that hold all our data and information. Hence, in spite of offering some unique benefits, centralised systems come at the cost of severe drawbacks like data hacks and security breaches, censorship, and lack of control over your data. We'll be talking about a decentralised platform for file sharing and storing in this project.

Decentralization, means that the system doesn't rely on a central authority where the decision making is performed independently by all the participating nodes, instead of relying on a single node. Decentralized file storage means that you can store your data, not on one single server or location, but many different nodes spread across multiple locations. We'll be using blockchain technology to serve the purpose of this project. Are you also wondering how blockchain will help in storing large files?

 A block can store only 20 kb of data and the average size of a picture is more than 1MB at least, implying that 50 blocks for a picture of 1 MB. This is terribly costly.Thus, we will be using blockchain technology and IPFS(Inter Planetary File System) protocol in this project. IPFS comes to the rescue as this protocol returns a hash of the data that was just added. This hash is cryptographically guaranteed to be unique to the content. If

the same content is added to IPFS again, the same unique hash will be returned. To retrieve content that has been stored on IPFS, the same cryptographic hash returned from the storage process is provided back to the IPFS network. IPFS allows us to store data and then later retrieve it, with the knowledge that our data wasn't tampered, which is what one wants. And by combining IPFS with blockchain we can see this turning into a very powerful platform.

Literature- Too much centralization also means that the governments can ban your access to any application, leaving you with no other options whatsoever. One recent example came from Turkey where the government banned Wikipedia in 2017, claiming it was a 'threat to their national security'. China has blocked access to popular social media, search engine platforms, and replaced them with applications that come with lots of government surveillance, content blockage, and censorship. So now the question arises, how can we make the internet decentralized again with no censorship and more public control? The answer to this is decentralised cloud storage space.Having discussed the drawbacks of centralised systems, the need for a decentralised atmosphere is evident. Here, an application based on blockchain and IPFS will serve the purpose. Instead of a single server, IPFS works on a huge swarm of nodes that store different blocks of data and users accessing the network can retrieve this data from the nearest node instead of contacting the server every time. Hence, removing the chances of single point failure.

Methodology- To proceed with this project, several dependencies were installed. Node js(for client side application) is installed which is followed by installation of truffle framework dependency(a framework for creating Ethereum smart contracts where one can write and test smart contracts written in solidity). Then, the Ganache personal blockchain is installed(a blockchain that runs on our computer to run transactions on the blockchain without actually paying money). Ganache has 10 different accounts with 100 fake ethers to perform transactions for developers. Lastly, MetaMask extension is added to the browser(which is later connected with Ganache and Web3). Having set the atmosphere, we shall discuss the steps now. Before that, a glance of what all is to be achieved is shown through this flowchart.
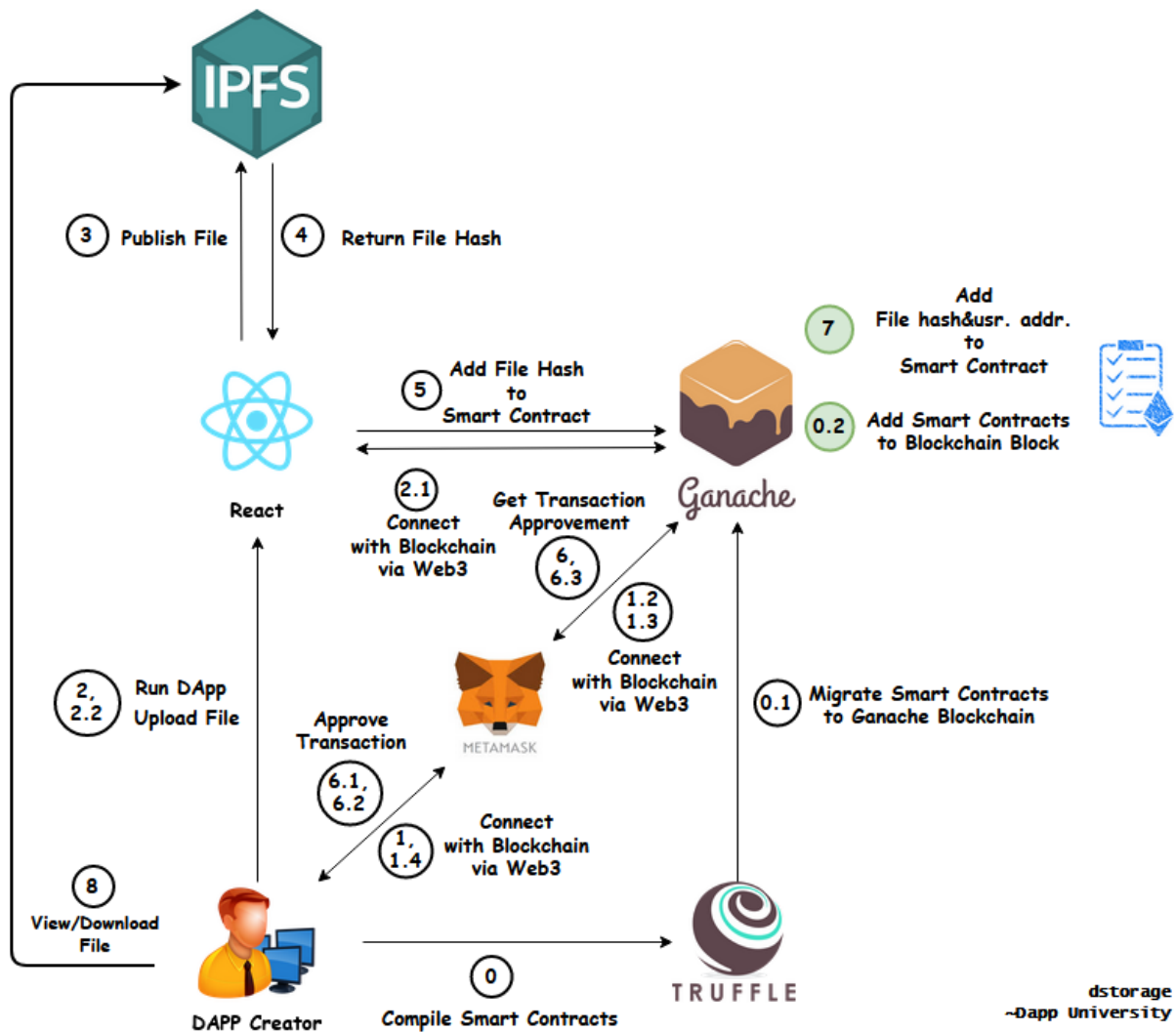
With the help of a react application, a user will upload files to IPFS. IPFS will return a hash which is then stored into a smart contract which will be written in the solidity programming language on Ganache. This smart contract(containing hashes) is then added to the blockchain. Here, the smart contract can be seen as a database for storing hashes.

I have used Visual Studio Code to write the code.
truffle-config.js file is created for the configuration of blockchain where the network development configuration is made.
A contract directory is made to have a migration contract(migration.sol) and a file for creating our own smart contract(say DStorage.sol) Code for smart contract is:

```solidity
pragma solidity >=0.5.0 <0.8.0;

contract DStorage {
```

```solidity
  string public name = 'DStorage';
  uint public fileCount = 0;
  mapping(uint => File) public files;

  struct File {
    uint fileId;
    string fileHash;
    uint fileSize;
    string fileType;
    string fileName;
    string fileDescription;
    uint uploadTime;
    address payable uploader;
  }

  event FileUploaded(
    uint fileId,
    string fileHash,
    uint fileSize,
    string fileType,
    string fileName,
    string fileDescription,
    uint uploadTime,
    address payable uploader
  );

  constructor() public {
  }

  function uploadFile(string memory _fileHash, uint _fileSize, string memory _fileType, string memory _fileName,
    string memory _fileDescription) public {
    // Make sure the file hash exists
    require(bytes(_fileHash).length > 0);
    // Make sure file type exists
    require(bytes(_fileType).length > 0);
    // Make sure file description exists
    require(bytes(_fileDescription).length > 0);
    // Make sure file fileName exists
    require(bytes(_fileName).length > 0);
```

```
    // Make sure uploader address exists
    require(msg.sender!=address(0));
    // Make sure file size is more than 0
    require(_fileSize>0);

    // Increment file id
    fileCount ++;

    // Add File to the contract
    files[fileCount] = File(fileCount, _fileHash, _fileSize, _fileType,
_fileName, _fileDescription, now, msg.sender);
    // Trigger an event
    emit FileUploaded(fileCount, _fileHash, _fileSize, _fileType,
_fileName, _fileDescription, now, msg.sender);
   }
}
```

Inside the smart contract, mapping is done for the hash and the file(as smart contracts will store hashes of files uploaded on IPFS). Each file is uniquely identified by its Id, hash, size, type, name, description, upload time and address from where it is uploaded. uploadFile function does the task of uploading files. The uploaded file is accessible by anyone who knows the hash.

Connection is established with the IPFS through :

const ipfsClient = require('ipfs-http-client')

const ipfs = ipfsClient({ host: 'ipfs.infura.io', port: 5001, protocol: 'https' }

1. Prepare file for upload(by creating a buffer object)
2. Upload file on IPFS(through a form)
3. Store the hash returned from IPFS in blockchain

The the files are uploaded on IPFS (through a form)

Browser is connected to blockchain browser by importing an account from Ganache blockchain and metamask is also connected to web3.js(which connects our application to blockchain)

Below is the brief explanation of what happens to the files on the IPFS network:

● The file is divided into chunks of data called blocks. Each block is given a unique hash.
● IPFS works on deduplication, which means that all the redundant files are removed from the network.
● Every node participating in the IPFS network stores the content with its hash and some indexing information.

- When a user wants to retrieve the file, he is telling the network to find a list of nodes that have the content behind a particular hash.
- With IPNS, a decentralized naming system, each file can be easily found by human-readable names.

Commands to run smart contracts -
'truffle migrate --reset' compiles the contracts. It also deploys a new copy of the smart contract to the blockchain as blockchain is immutable so we can use this command to make changes in the smart contract.
'truffle test' is used to test the smart contracts.
'truffle console' takes to the truffle console where we can run our smart contracts using javascript commands.
'npm run start' command is used to start the server for seeing the client side application that we have built to access this platform.

Result Analysis- Decentralized file storage puts another threat on the table: privacy, security, and integrity of data. Blockchains are immutable. Hence, by combining with blockchain and time stamping the data, helps us achieve security and integrity and it also provides non repudiation.
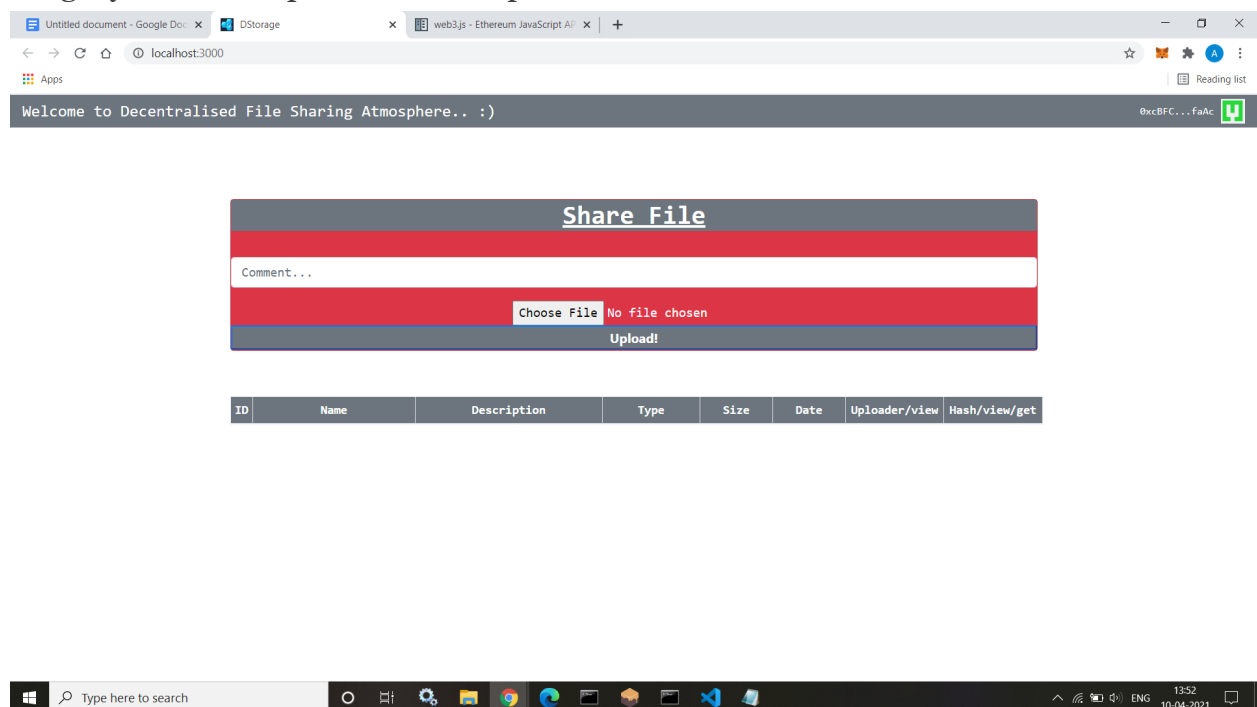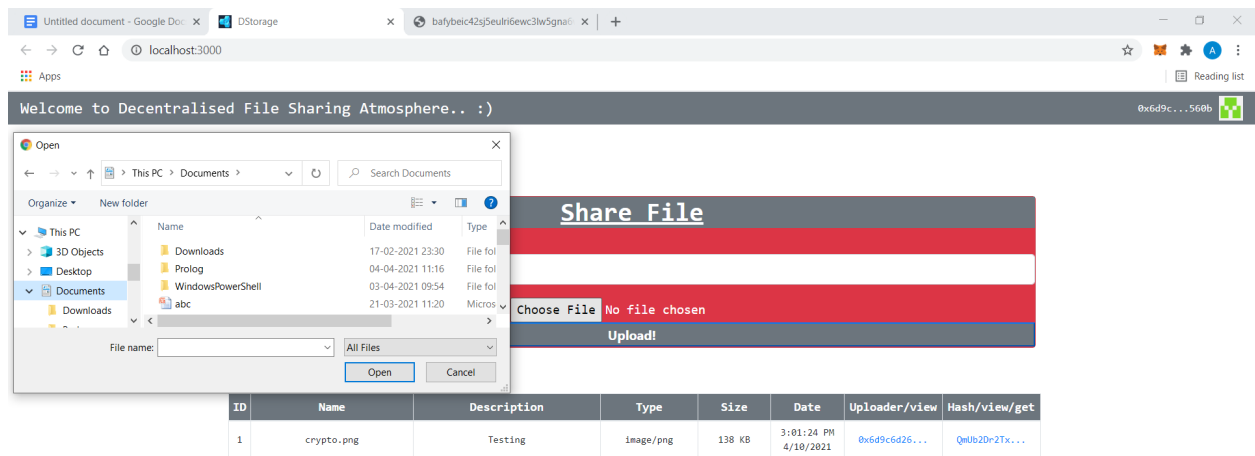
This is how my application looks. One can choose a file to upload. After the file is uploaded, its ID. name,description(what we enter in the above textbox before uploading), file type, size, date and hash will be displayed in the table.
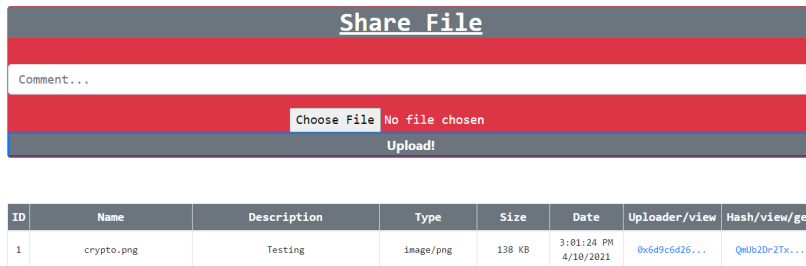
File uploaded successfully. One can access the file with the link mentioned in the hash column. It is
https://bafybeic42sj5eulri6ewc3lw5gna6voes24dparioyxtwyks4gqjooqjhe.ipfs.infura-ipfs.io/ Hence the file is accessible to anyone with the hash.
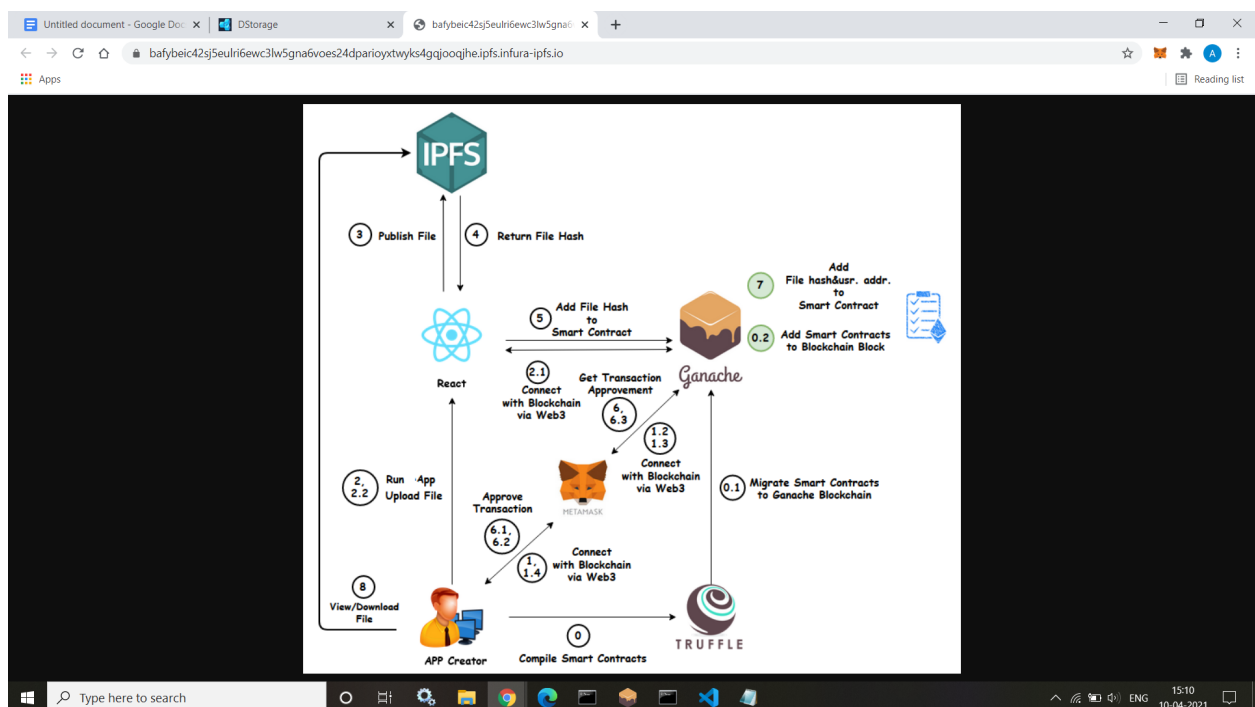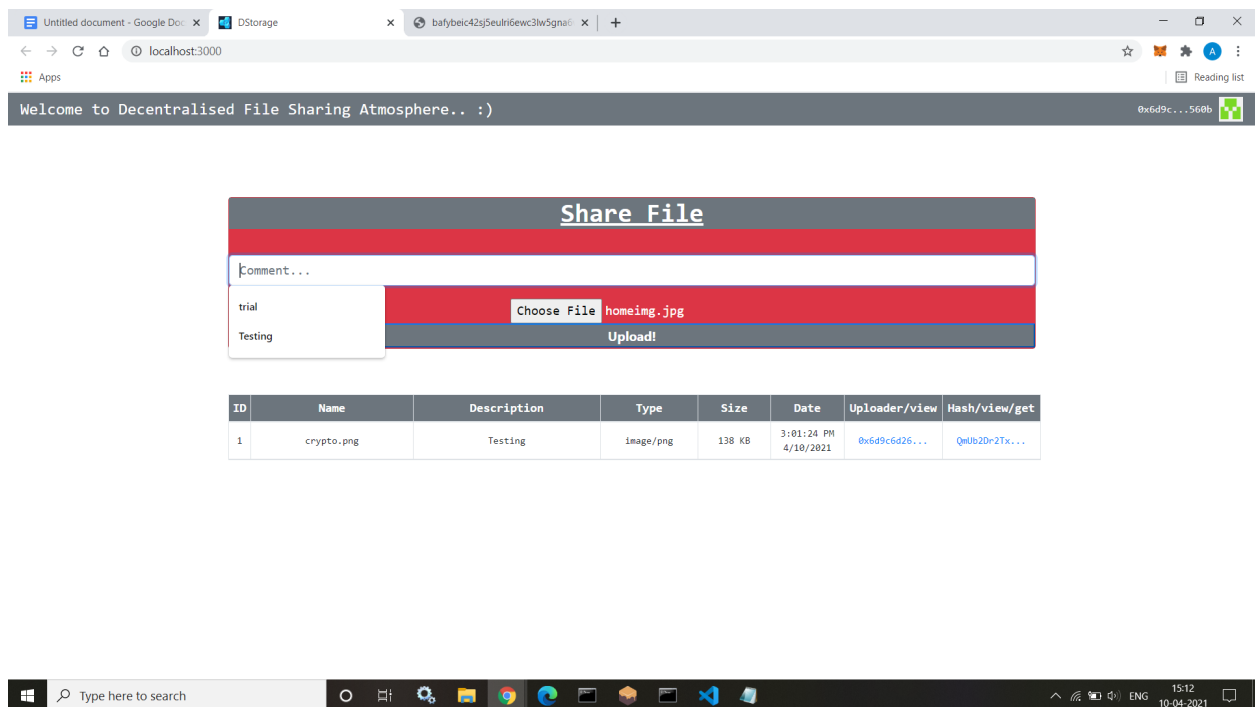
This is the uploaded image. Hence one can access the file with its hash not its location(unlike HTTP) and this hash is dependent on content and unique for a given file.

The Choose file button offers you to pick a file for uploading.

One can write a comment/description for the file to be uploaded.
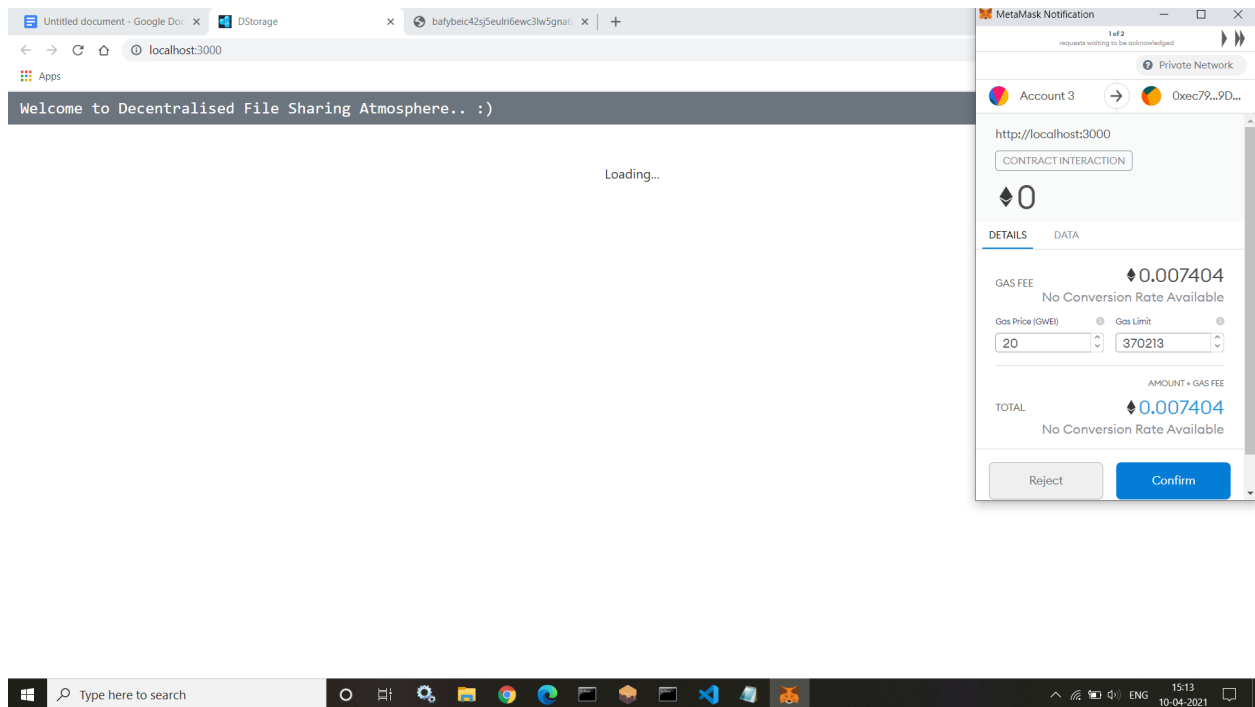
For each transaction(i.e, for every upload) MetaMask asks for permission to reject or confirm the transaction. Hence converting our browser to blockchain browser.
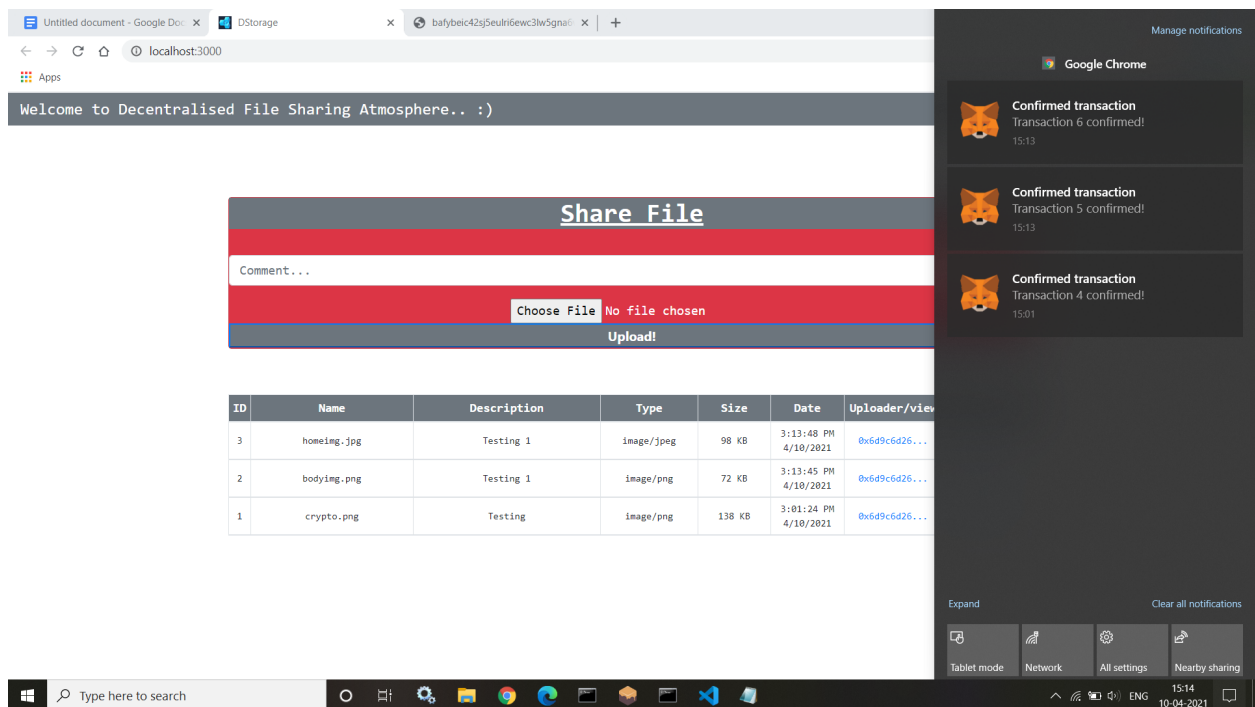
On confirming the transaction, one gets notified.

Pdf file is uploaded. On trying to access it using the hash one gets,

Which means that it is accessible. Thus, we successfully uploaded files(images and documents on our blockchain and IPFS based file sharing platform)

This is a broader perspective of how secure decentralized cloud storage is. With no central location of your files and with encryption built into the system, decentralized cloud storage might be more secure than the centralized solutions available today. One can encrypt the blocks before uploading them on IPFS and the hash has to be decrypted with the private key for accessing the file.

Conclusion- Hence, a blockchain and IPFS based file storing and file sharing platform is made in this project which has the benefits of both, blockchain and IPFS One can conclude that this serves as a suitable decentralised platform to store and share files.