BLOCKCHAIN BASED DECENTRALIZED FILE STORING AND SHARING SYSTEM USING IPFS

Submitted in partial fulfillment of the requirements for the degree of

Integrated-Master of Technology

in

CSE SPECIALIZATION IN DATA SCIENCE

by

SANJAY KUMAR K-19MID0073

PRAVEEN SAGAR K V J-19MID0081

CSI3013-BLOCKCHAIN TECHNOLOGIES

Under the guidance of

Prof. / Dr. JOTHI K R

SCOPE VIT,VELLORE.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

APRIL/2022

DECLARATION

I hereby declare that the thesis entitled "**BLOCKCHAIN BASED DECENTRALIZED FILE STORING AND SHARING SYSTEM USING IPFS"** submitted by me, for the award of the degree of *Bachelor of Technology in* CSE SPECIALIZATION IN DATA SCIENCE to VIT is a record of bonafide work carried out by me under the supervision of JOTHI K R.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date :

SANJAY KUMAR K-19MID0073

PRAVEEN SAGAR K V J-19MID0081

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Thesis title" submitted by SANJAY KUMAR K [19MID0073] & PRAVEEN SAGAR K V J [19MID0084], **INTEGRATED M.TECH CSE WITH DATA SCIENCE,SCOPE SCHOOL**, VIT University, for the award of the degree of *Bachelor of Technology in Programme*, is a record of bonafide work carried out by him under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : **Signature of the Guide**

Internal Examiner **External Examiner**

ACKNOWLEGMENT

Innovation inspiration and motivation have always played a key role in the success of any venture that man takes upon himself.

We would like to express our deepest appreciation to all those who have provided us with the resources and possibility to complete this project. As it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere gratitude towards all of them.

We feel the need to acknowledge our indebtedness and deep sense of obligation to Prof.JOTHI K R who kindled our interest in this topic. We thank him for the valuable guidance and constant supervision throughout the implementation of project and also for the support in helping us complete the project. We are immensely appreciative to our classmates for helping us by constantly encouraging us throughout, in the process of implementing and completing our project.

SANJAY KUMAR K-19MID0073

PRAVEEN SAGAR K V J-19MID0081

CONTENTS **Page No.**

.

1. INTRODUCTION

1.1. OBJECTIVE

To implement an Decentralized application(Dapp) for the File storing and sharing system using Interplanetary File System (IPFS) and Ethereum Blockchain.

1.2. Motivation

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system .A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralised database managed by multiple participants is known as Distributed Ledger Technology (DLT).Blockchain is a type of DLT in which transactions are recorded with an immutable cryptographic signature called a hash.The features of Blockchain are Independent and Decentralized, Transaction Transparent, Data Security.But Storing huge files in blockchain is really expensive.To Overcome the weakness of Blockchain – Instead of storing the whole file on the blockchain, we only store the file's hash generated by IPFS to make the blockchain light and more efficient.. IPFS is a **versioned file system** which can store files and track file versions over time.It is a protocol and **peer-to-peer network** for storing and sharing data in a distributed file system.It uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices.It uses **Distributed Hash Tables** (DHT). In DHT the data is spread across a network of computers, and efficiently coordinated to enable efficient access and lookup between nodes.

1.3. Background

It is prohibitively expensive to store a lot of data on the Ethereum blockchain.According to Ethereum's yellow paper it is approximately 20,0000 gas for 256bit/8 bytes (1word). Based on 02/28/2018 gas prices of 4 gwei/gas. 20,000 gas per Transaction of 8 bytes x 4 gwei/gas = 80,000 gwei for 8 bytes. 80,000 gwei for 8 bytes. x 1000bytes/8 = 10,000,000 gwei/kB = .01 Ether. 01 Ether/kB x 1000kB = 10 Ether to store a 1Mb at $860/ether = $8600.00! It would cost $8,600,000.00 to store a 1 GB file on the Ethereum blockchain!To store Ethereum's 38 page PDF yellow paper (520Kb) = $4472 USD. If we can only afford to store a couple of Kb of data on the blockchain, then we would still have to rely on a centralized server to store data. There exists, a solution to store data on decentralized network called the InterPlanetary File System ("IPFS") .

2.PROJECT DESCRIPTION AND GOALS

Currently, Many software systems rely on cloud data storage for data management.With enormous growth of digital world, cloud storage has turned out into most reliable and convenient way of storing data .Generally, Data stored using cloud storage is stored in centralized manner.A major advantage of traditional cloud storage is, it is not only easy to handle but also easy to access the data. The data stored on cloud storage can be easily accessed by number of devices at a time. This way of storing data can cause single point failure, denial of service attack which may further lead to unavailability of data. So,our project aims to develop a system that can store data in an **Decentralized manner** where files can be stored and shared securely.This system can definitely overcome problems like single point failure, data unavailability etc…By integrating **blockchain** technology, we make sure that the files shared on the network **cannot** be corrupted and the complete history of all the files shared will be stored giving users a sense of security.

3.LITERATURE REVIEW

• [1] provided by Satoshi Nakamato,solves the problem of double spending using a peer-to-peer(P2P) network

• Rajalakshmi et al [2] proposed an academic research record keeping model using blockchain and IPFS with access control mechanisms.

• Nishara Nizamuddin et al [3] proposed a Blockchain-based solution for the Authenticity of Online Publications i.e Posted freely on the Internet using IPFS and smart contracts

• Shreya Khatal et al [4] presents an idea on how blockchain based file sharing application can be built with access control mechanisms.

• Anuradha Jayakody et al[5] proposed a model named as Air-Talk, a private communication platform developed mainly by using IPFS and Blockchain

• Jatin Pahuja et al[6] proposed a comparison of Interplanetary File System with Block chain Based File System and evaluated its factors and concluded that IPFS is the preferred choice over a purely blockchain based system

4.TECHNICAL SPECIFICATION

We have used the following tech stack to implement this project.

1)**Ethereum Blockchain**-To store the File's hash.

2)**Solidity** -To code smart contracts .

3)**infura:** Used to access an IPFS network.

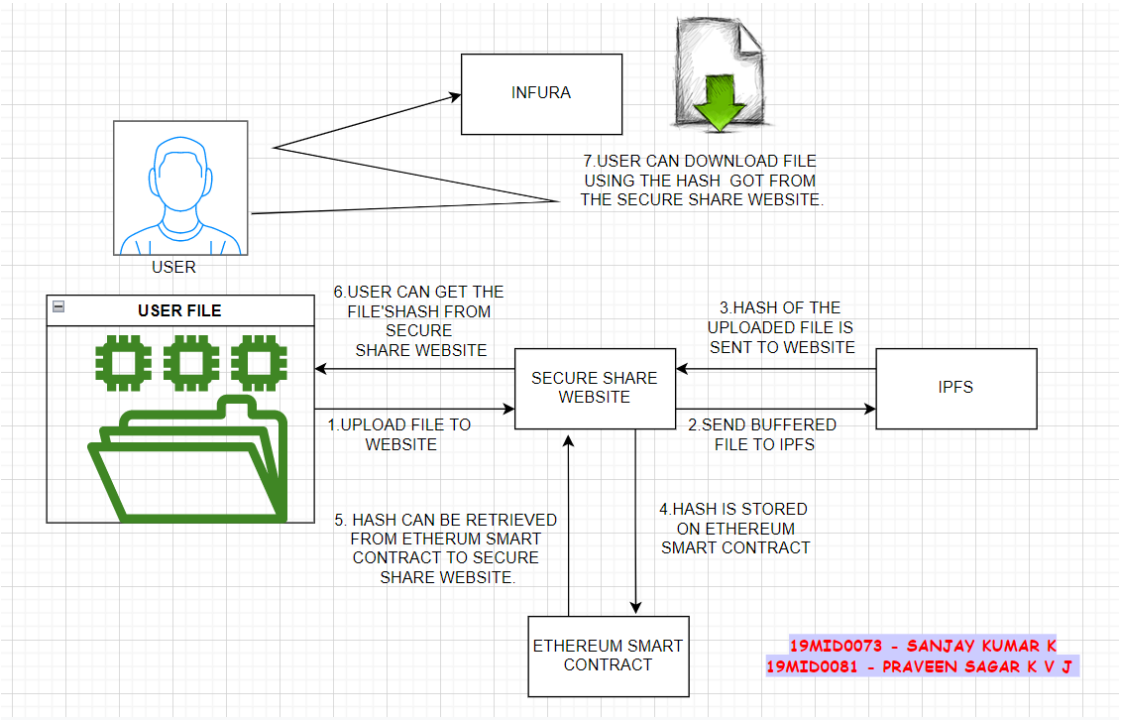
4)**IPFS**-to store the Files.

5)**Metamask**-Web3 Provider.

6)**HTML/CSS/JavaScript/Reactjs**-For the Frontend of our application.

7)Visual studio Code-To run Reactjs

5.DESIGN APPROACH AND DETAILS :



.

6.PROJECT DEMONSTRATION .

SMART CONTRACT :

```
pragma solidity ^0.8.7;
contract SMART_CONTRACT
{
 string ipfsHash;

 function sendHash(string x) public
 {
   ipfsHash = x;
 }

 function getHash()

 public view returns (string x)
 {
   return ipfsHash;
 }
}
```

App.css :

```
.App {

  text-align: center;

}



.App-logo {

  height: 40vmin;

  pointer-events: none;
```

```
10
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
```

```
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

App.js :

```
import React, { Component } from 'react';


//import logo from ‘./logo.svg’;
import './App.css';
import web3 from './web3';
import ipfs from './ipfs';
import storehash from './storehash';
class App extends Component {

   state = {
     ipfsHash:null,
     buffer:'',
     ethAddress:'',
     blockNumber:'',
     transactionHash:'',
     gasUsed:'',
     txReceipt: ''
   };
captureFile =(event) => {
       event.stopPropagation()
       event.preventDefault()
       const file = event.target.files[0]
       let reader = new window.FileReader()
       reader.readAsArrayBuffer(file)
       reader.onloadend = () => this.convertToBuffer(reader)
     };
 convertToBuffer = async(reader) => {
     //file is converted to a buffer for upload to IPFS
       const buffer = await Buffer.from(reader.result);
     //set this buffer -using es6 syntax
       this.setState({buffer});
   };
onClick = async () => {
try{
       this.setState({blockNumber:"waiting.."});
       this.setState({gasUsed:"waiting..."});
//get Transaction Receipt in console on click
//See: https://web3js.readthedocs.io/en/1.0/web3-eth.html#gettransactionreceipt
await web3.eth.getTransactionReceipt(this.state.transactionHash, (err, txReceipt)=>{
         console.log(err,txReceipt);
         this.setState({txReceipt});
       }); //await for getTransactionReceipt
await this.setState({blockNumber: this.state.txReceipt.blockNumber});
       await this.setState({gasUsed: this.state.txReceipt.gasUsed});
     } //try
   catch(error){
       console.log(error);
```

```
    } //catch
  } //onClick
onSubmit = async (event) => {
    event.preventDefault();
   //bring in user's metamask account address
    const accounts = await web3.eth.getAccounts();

    console.log('Sending from Metamask account: ' + accounts[0]);
  //obtain contract address from storehash.js
    const ethAddress= await storehash.options.address;
    this.setState({ethAddress});
  //save document to IPFS,return its hash#, and set hash# to state
  //https://github.com/ipfs/interface-ipfs-core/blob/master/SPEC/FILES.md#add
    await ipfs.add(this.state.buffer, (err, ipfsHash) => {
      console.log(err,ipfsHash);
      //setState by setting ipfsHash to ipfsHash[0].hash
      this.setState({ ipfsHash:ipfsHash[0].hash });
  // call Ethereum contract method "sendHash" and .send IPFS hash to etheruem
contract
  //return the transaction hash from the ethereum contract
 //see, this https://web3js.readthedocs.io/en/1.0/web3-eth-contract.html#methods-
mymethod-send

      storehash.methods.sendHash(this.state.ipfsHash).send({
        from: accounts[0]
      }, (error, transactionHash) => {
        console.log(transactionHash);
        this.setState({transactionHash});
      }); //storehash
    }) //await ipfs.add
  }; //onSubmit
render() {

    return (
      <div className="App">
        <header className="App-header">
          <h1> Ethereum and IPFS with Create React App</h1>
        </header>

        <hr />
<Grid>
        <h3> Choose file to send to IPFS </h3>
        <Form onSubmit={this.onSubmit}>
          <input
            type = "file"
            onChange = {this.captureFile}
          />
           <Button
```

```
          bsStyle="primary"
          type="submit">
          Send it
          </Button>
        </Form>
<hr/>
 <Button onClick = {this.onClick}> Get Transaction Receipt </Button>
  <Table bordered responsive>
            <thead>
              <tr>
                <th>Tx Receipt Category</th>
                <th>Values</th>
              </tr>
            </thead>

            <tbody>
              <tr>
                <td>IPFS Hash # stored on Eth Contract</td>
                <td>{this.state.ipfsHash}</td>
              </tr>
              <tr>
                <td>Ethereum Contract Address</td>
                <td>{this.state.ethAddress}</td>
              </tr>
              <tr>
                <td>Tx Hash # </td>
                <td>{this.state.transactionHash}</td>
              </tr>
              <tr>
                <td>Block Number # </td>
                <td>{this.state.blockNumber}</td>
              </tr>
              <tr>
                <td>Gas Used</td>
                <td>{this.state.gasUsed}</td>
              </tr>

            </tbody>
          </Table>
      </Grid>
    </div>
     );
   } //render
} //App
export default App;
```

App.test.js :

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Index.css :

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

Index.js :

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App.css';

import 'bootstrap/dist/css/bootstrap.min.css';
ReactDOM.render(<App />, document.getElementById('root'));
```

ipfs.js :

```
const IPFS = require('ipfs-api');
const ipfs = new IPFS({ host: 'ipfs.infura.io', port: 5001, protocol: 'https' });
export default ipfs;
```

reportWebVitals.js :

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

setupTest.js :

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

storehash.js :

```
import web3 from './web3';
//Your contract address
const address = '0x610dd75057738b73e3f17a9d607db16a44f962f1';
//Your contract ABI
const abi = [ {  "constant": false,  "inputs": [   {    "name": "x",    "type":
"string"   }  ],  "name": "sendHash",  "outputs": [],  "payable": false,  "stateMutability":
"nonpayable",  "type": "function" }, {  "constant": true,  "inputs": [],  "name":
"getHash",  "outputs": [   {    "name": "x",    "type": "string"   }  ],  "payable":
false,  "stateMutability": "view",  "type": "function" }]
export default new web3.eth.Contract(abi, address);
```

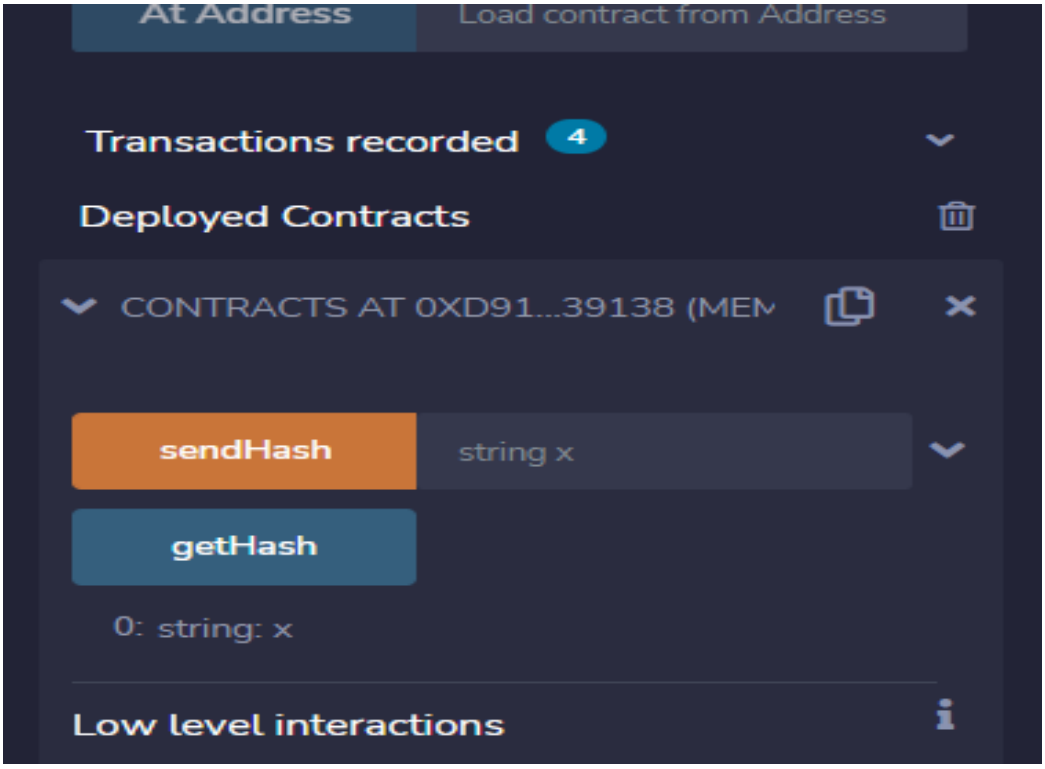
package.json :

```
{
  "name": "ipfs",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
```

```
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.1.1",
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^5.1.3",
    "react": "^18.0.0",
    "react-dom": "^18.0.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

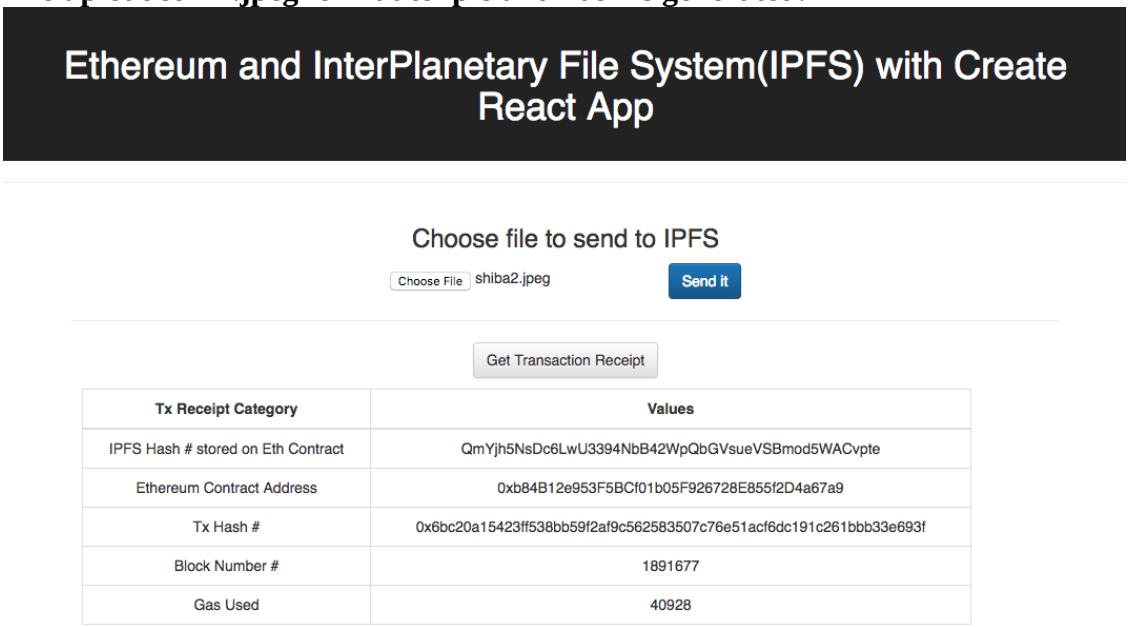

RESULT:

CONTRACT DEPLOYMENT:



DECENTRALIZED APPLICATION:

File uploaded in .jpeg format to ipfs and hash is generated:



Tx Receipt Category	Values
IPFS Hash # stored on Eth Contract	QmYjh5NsDc6LwU3394NbB42WpQbGVsueVSBmod5WACvpte
Ethereum Contract Address	0xb84B12e953F5BCf01b05F926728E855f2D4a67a9
Tx Hash #	0x6bc20a15423ff538bb59f2af9c562583507c76e51acf6dc191c261bbb33e693f
Block Number #	1891677
Gas Used	40928

We can access our uploaded file at https://gateway.ipfs.io/ipfs/hash

SUMMARY:

In this Project , We implemented a File storage and Sharing System using Interplanetary File sharing SyStem (IPFS) and Ethereum blockchain which is a decentralized application and it is very secured because We store the hash value generated by IPFS in Ethereum Smart Contract.

References

[1]. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, Retrieved on 05 June 2021.

[2].Rajalakshmi, A. & Lakshmy, K.V. & Amritha, P.P.. (2018). A blockchain and IPFS based framework for secure Research record keeping. International Journal of Pure and Applied Mathematics. 119. 1437-1442.

[3].Nizamuddin, Nishara & Hasan, Haya & Salah, Khaled. (2018). IPFS-Blockchain-Based Authenticity of Online Publications. 10.1007/978-3-319-94478-4_14.

[4].Shreya Khatal, Jayant Rane, Dhiren Patel, Pearl Patel, Yann Busnel. FileShare: A Blockchain and IPFS framework for Secure File Sharing and Data Provenance. ICMLCI 2019 : International Conference on Machine Learning and Computational Intelligence, Dec 2019, Bhubaneswar, Kantabada, India.10.1007/978-981-15-5243-4_79. hal-02451022.

[5].AnuradhaJayakody,Rajitha De Silva,..."Air-Talk": An IPFS and Blockchain-based Private Decentralized File Sharing and Chat Application.InternationalResearchJournalOfEngineeringandTechnology(IRJET),NOV 2020,VOLUME: 07 ISSUE: 11.

[6].Jatin Pahuja, Dr. Neha Agrawal, Dr. Meenu Garg:Comparison of Interplanetary File System with Block chain Based File System.International journal of creative research thoughts(IJCRT),Volume 9, Issue 6 June 2021 | ISSN: 2320-2882.