



School: Campus:

Academic Year: Subject Name: Subject Code:

Semester: Program: Branch: Specialization:

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiment : Store with IPFS – Decentralized file upload.

Objective/Aim:

- To build a web-based application that uploads files to the **InterPlanetary File System (IPFS)** using **Pinata's API** through a secure, frontend-only method using **JWT (JSON Web Token)**.

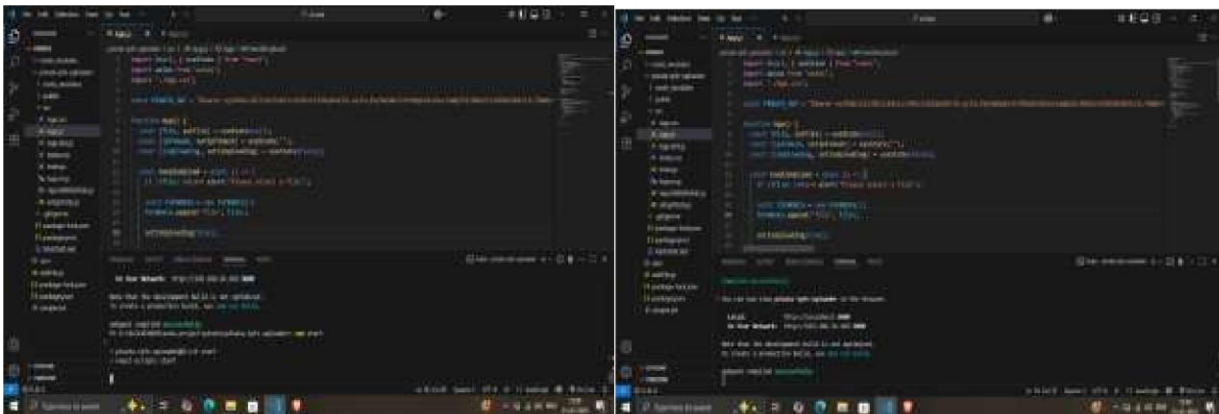
Apparatus/Software Used:

- ☐ Laptop/PC
- ☐ PowerPoint/Word for documentation
- ☐ Internet for research
- ☐ <https://pinata.cloud/>

Theory/Concept:

- ☐ **IPFS (InterPlanetary File System)** is a decentralized protocol for storing and sharing files using a peer-to-peer network.
- ☐ **Pinata** provides an IPFS API gateway that allows users to pin files so they remain permanently accessible.
- ☐ **JWT (JSON Web Token)** is used to securely authenticate frontend users without exposing secret keys.

1. **Create a React app** using `npx create-react-app` and navigate into the project folder.
2. **Install Axios** using `npm install axios` to handle HTTP requests.
3. Inside the `App.js` file, **set up the file upload interface** using a file input and a button.
4. Use the **`useState`** hook to manage file selection, upload state, and the IPFS hash result.
5. When the user selects a file and clicks upload, **create a `FormData` object** and append the file to it.
6. Make a **POST request to Pinata's IPFS API endpoint** (`https://api.pinata.cloud/pinning/pinFileToIPFS`) using `Axios`.
7. In the request, include the **Authorization header with your Pinata JWT token** (must begin with "Bearer ").
8. If the upload is successful, **receive the `IpfsHash`** from the response and display the gateway URL to access the file.
9. **Test the uploaded file** using `https://gateway.pinata.cloud/ipfs/<your-ipfs-hash>` to verify successful storage on IPFS.



```
import { useState } from 'react';
import axios from 'axios';

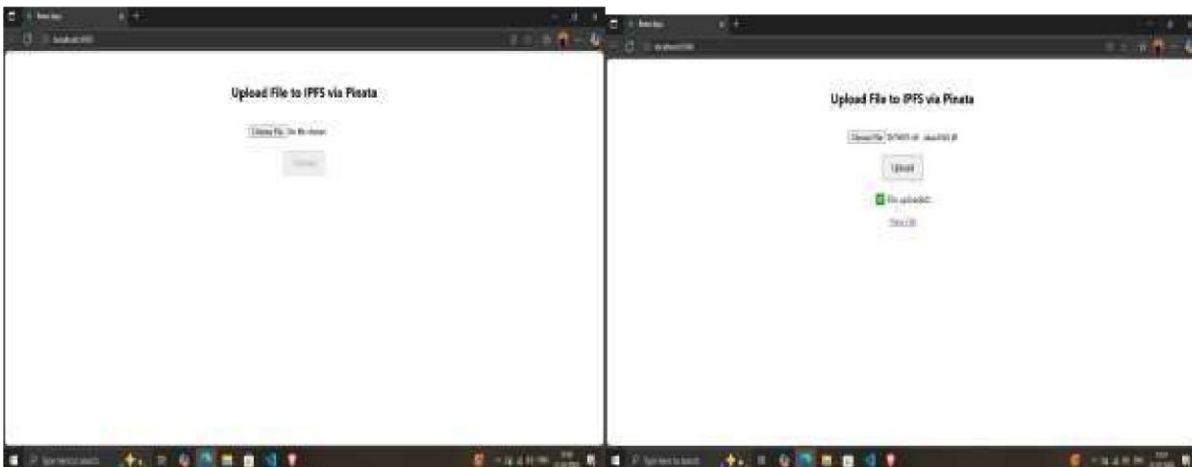
function App() {
  const [file, setFile] = useState(null);
  const [uploadState, setUploadState] = useState('idle');
  const [ipfsHash, setIpfsHash] = useState('');
  const [gatewayUrl, setGatewayUrl] = useState('');

  const handleFileChange = (e) => {
    setFile(e.target.files[0]);
  };

  const handleUpload = async () => {
    setUploadState('uploading');
    const formData = new FormData();
    formData.append('file', file);

    try {
      const response = await axios.post('https://api.pinata.cloud/pinning/pinFileToIPFS', formData, {
        headers: {
          'Authorization': `Bearer ${process.env.REACT_APP_PINATA_JWT_TOKEN}`,
        },
      });
      setIpfsHash(response.data.IpfsHash);
      setGatewayUrl(`https://gateway.pinata.cloud/ipfs/${response.data.IpfsHash}`);
    } catch (error) {
      console.error('Error uploading file to IPFS:', error);
    }
    setUploadState('idle');
  };

  return (
    <div>
      <input type="file" onChange={handleFileChange}/>
      <button onClick={handleUpload}>Upload</button>
    </div>
    <div>
      <p>IPFS Hash: {ipfsHash}</p>
      <p>Gateway URL: {gatewayUrl}</p>
    </div>
  );
}
```



Observation Table:

Test File Name	File Type	IPFS Hash (CID)	Status
test.png	Image	QmXyz...	Uploaded
project.pdf	PDF	Qm123...	Uploaded
network.docx	DOCX	Qm456...	Uploaded

ASSESSMENT

	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Interpretation	10		
Record of Applied and Action Learning	10		
	10		

Signature of the Student:

Signature of the Faculty:

Name :

Regn.No.