Centurion University
Shaping Lives... Empowering Communities...

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 Experiement : Store with IPFS – Decentralized File Upload

Objective/Aim:

To understand how decentralized file storage works using IPFS by uploading a file, generating its content identifier (CID), and retrieving it via IPFS gateways.

Apparatus/Software Used:

- Web Browser
- Pinata Cloud or Web UI IPFS Gateway
- File (image, text, etc.) to upload
- IPFS Public Gateway: https://ipfs.io/ipfs/
- IPFS Upload Demo: https://www.pinata.cloud/
- Vs code

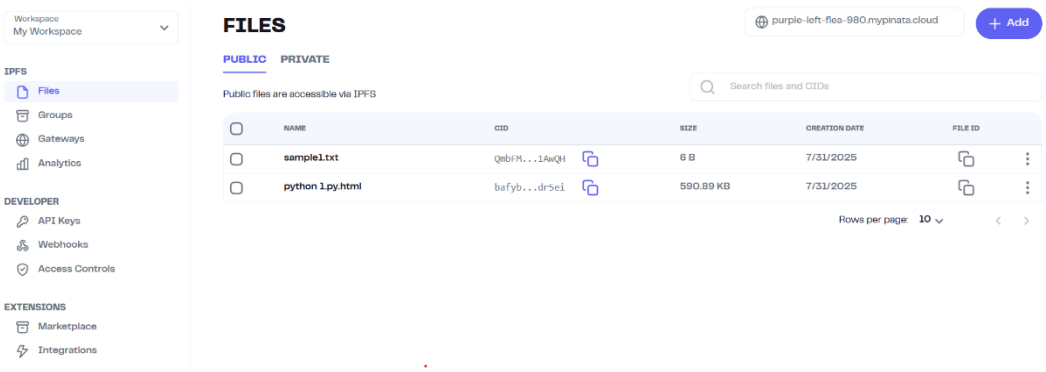
Theory/Concept:

IPFS (InterPlanetary File System) is a peer-to-peer hypermedia protocol for storing and sharing data in a distributed file system.

- Every file uploaded to IPFS is broken into chunks and stored across a distributed network.
- Each file is given a unique **CID (Content Identifier)** which can be used to retrieve the file.
- Files are content-addressed, not location-addressed (unlike HTTP).

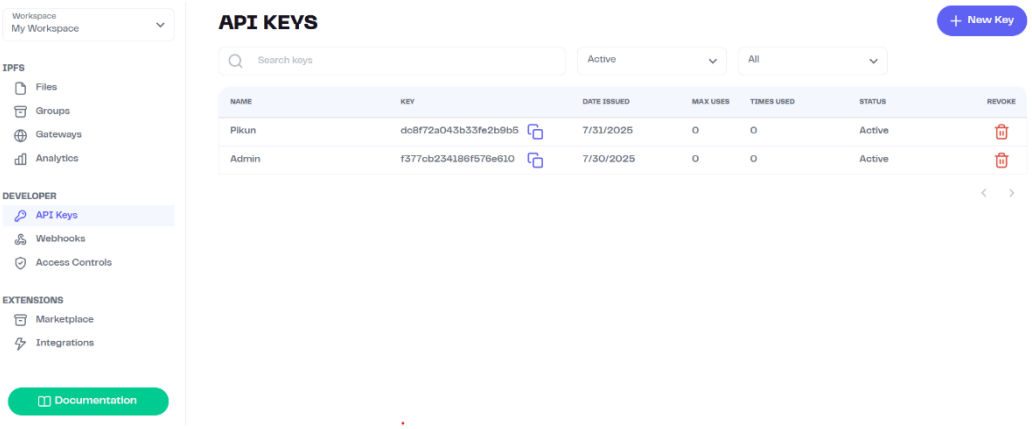
Procedure:

1. **Go to Pinata**: https://app.pinata.cloud/
2. **Sign in** with your GitHub or email account.
3. Go to **"Files"** tab
4. Click on **"Upload" > "File"**

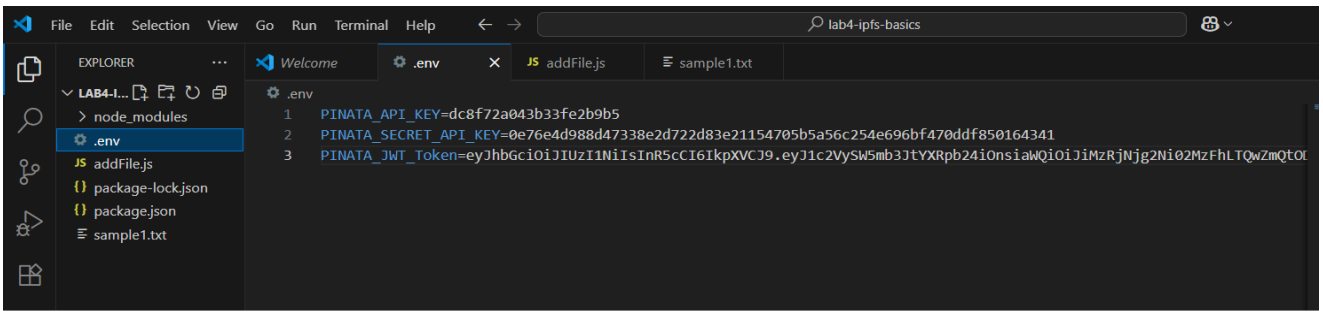


5. Navigate to **"API Keys"** under your account dropdown.
6. Click on **"New Key"** and generate a new API key:

- Give it a name (e.g., Pikun)
- Enable permissions (at least pinFileToIPFS and pinList)
- Copy the **API Key**, **API Secret**, and **JWT Token** (save them securely)



7. Open your project folder in **VS Code**.
8. Create a .env file in the root directory (if not already present).
9. Add the following environment variables:



10. In your JavaScript/Node.js code, use a package like axios or the official Pinata SDK:

```
PS C:\Users\pikun\OneDrive\Desktop\lab4-ipfs-basics>  npm install axios dotenv form-data
```

11. Sample upload script using axios:

```
Welcome    .env    addFile.js    sample1.txt
addFile.js > ...
1   require("dotenv").config();
2   const axios = require("axios");
3   const FormData = require("form-data");
4   const fs = require("fs");
5
6   async function uploadToIPFS(filePath){
7       const form = new FormData();
8       form.append("file", fs.createReadStream(filePath));
9
10      const res = await axios.post("https://api.pinata.cloud/pinning/pinFileToIPFS", form, {
11          maxBodyLength: "Infinity",
12          headers: {
13              ...form.getHeaders(),
14              pinata_api_key: process.env.PINATA_API_KEY,
15              pinata_secret_api_key: process.env.PINATA_SECRET_API_KEY,
16          },
17      });
18
19      console.log("📌 File pinned to IPFS!");
20      console.log("🔗 Hash:", res.data.IpfsHash);
21
22  }
23  💡
24  uploadToIPFS("sample1.txt");
25  // Ensure you have a file named 'sample.txt' in the same directory as this script.
```

12. Running my script:

```
PS C:\Users\pikun\OneDrive\Desktop\lab4-ipfs-basics> node addFile.js
[dotenv@17.2.1] injecting env (3) from .env -- tip: 🔭 observe env with Radar: https://dotenvx.com/radar
📌 File pinned to IPFS!
🔗 Hash: QmbFMke1KXqnYyBBWxB74N4c5SBnJMVAiMNRcGu6x1AwQH
```

13. Frontend Of The Script

Observation Table:

- After generating a new API key and configuring it in the .env file, the script was able to authenticate and connect to Pinata's API successfully.
- The selected file (e.g., image, text, PDF) was uploaded using the API without needing manual web interface interaction.
- Upon successful upload, the API responded with a CID (Content Identifier)
- Accessing https://ipfs.io/ipfs/QmXk123abc...xyz in a browser successfully loaded the uploaded file.
- Any change in the file content resulted in a new CID, proving that IPFS is content-addressed.
- The file remains available as long as it's pinned to IPFS (by Pinata or another IPFS node).

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:
Name :
Regn. No. :

Signature of the Faculty: