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

Coding Phase : Pseudo Code/Flow Chart/Algorithm

```
• Start
• Create a .env file with API credentials.
• Import required modules.
• Define an asynchronous uploadToIPFS() function.
• Inside the function:
    • Read the file using fs.createReadStream.
    • Append to FormData.
    • Post to Pinata endpoint.
• Handle response:
    • Display IPFS hash and gateway link.
• Handle errors.
• End
```

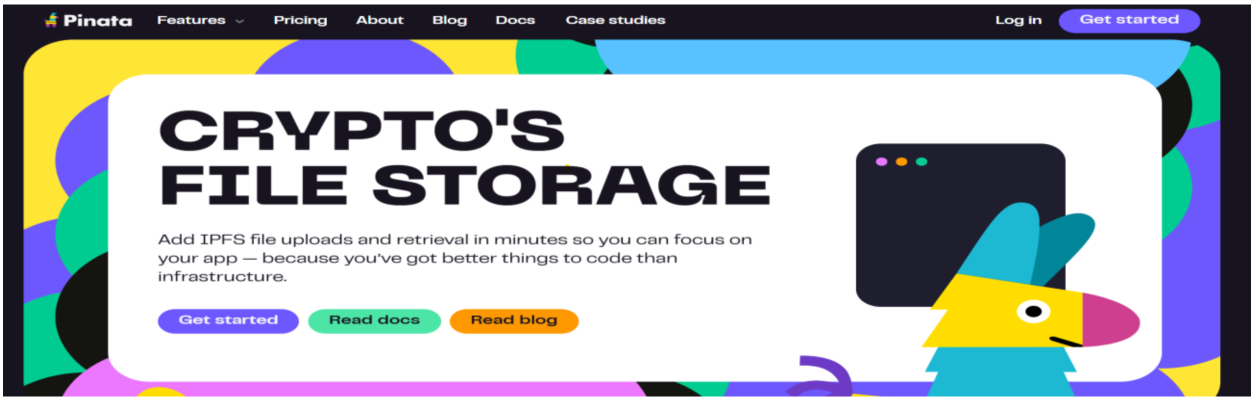
Apparatus/Software Used:

- Node.js
- Axios
- Dotenv
- Form-data
- Pinata

Testing Phase:

Step 1 : Open Pinata Website

- Visit piñata.cloud
- Login or sign up with your e-mail

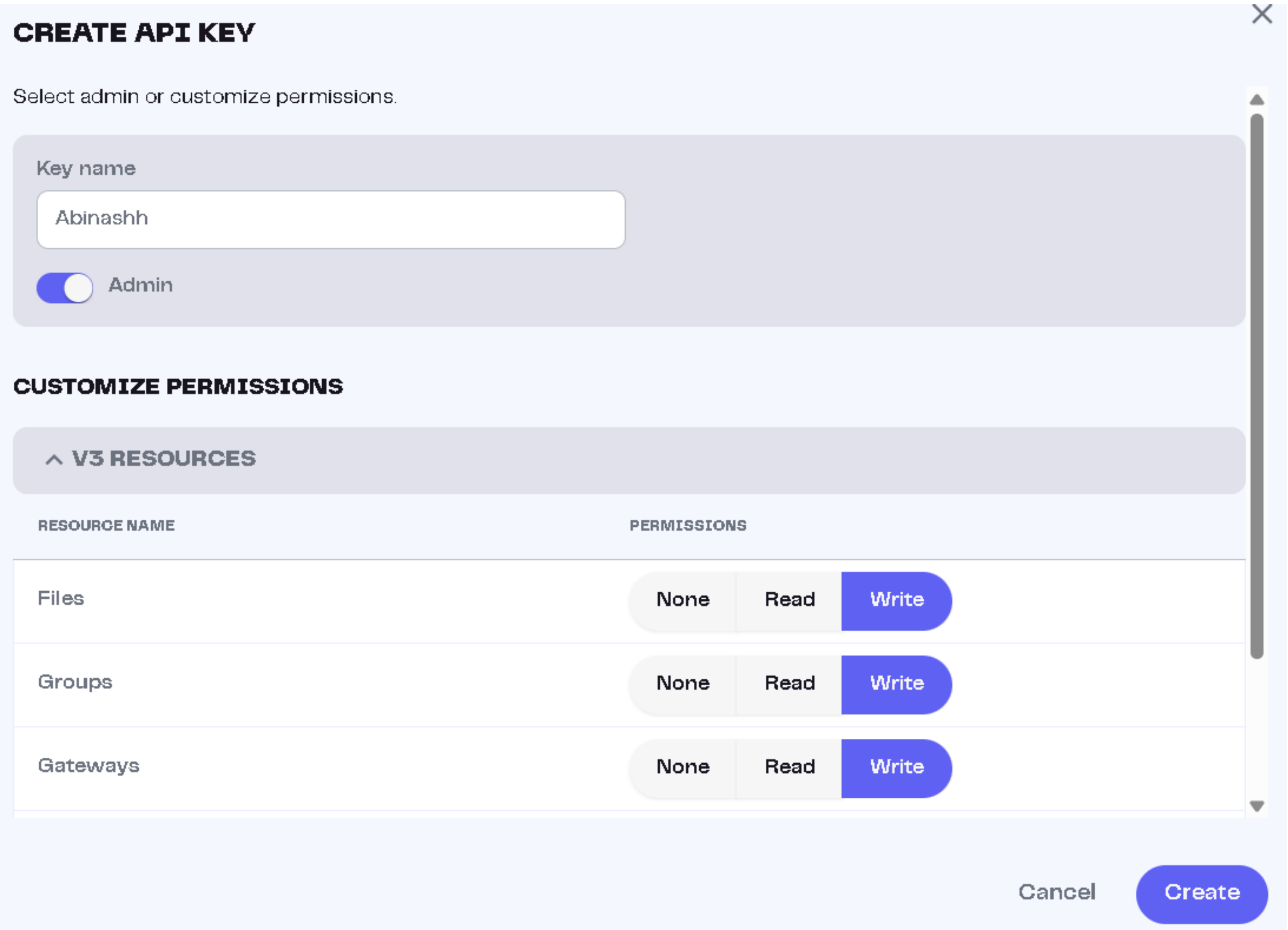


CRYPTO'S FILE STORAGE

Add IPFS file uploads and retrieval in minutes so you can focus on your app — because you've got better things to code than infrastructure.

Step2: Create api key

- Click api keys options
- Click new key
- Enter key name and click on admin option and click create



CREATE API KEY

Select admin or customize permissions.

Key name: Abinashh

● Admin

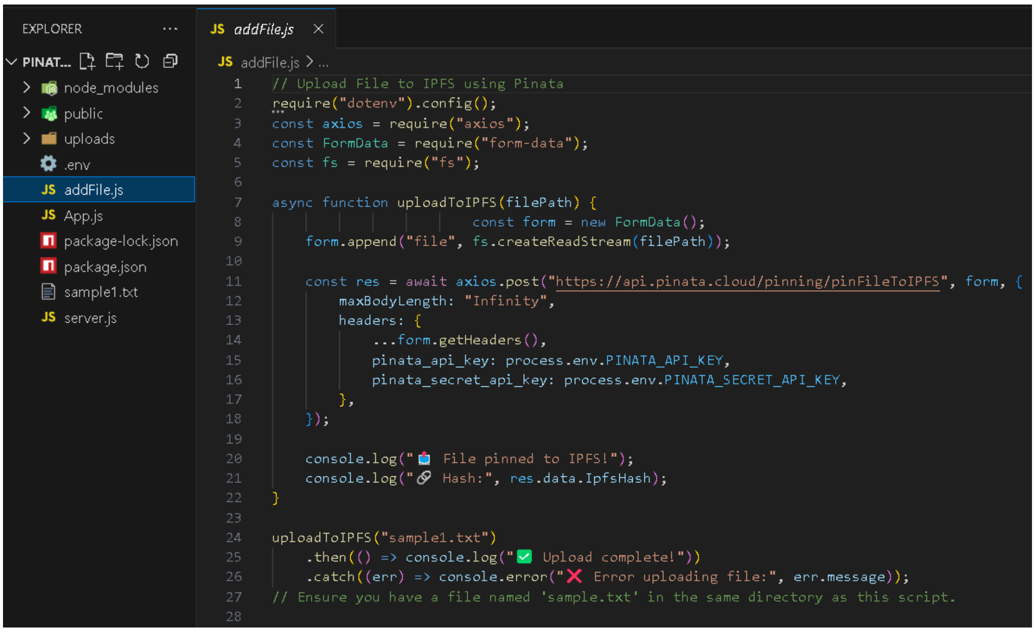
CUSTOMIZE PERMISSIONS

V3 RESOURCES

RESOURCE NAME	PERMISSIONS
Files	None Read [✓ Write]
Groups	None Read [✓ Write]
Gateways	None Read [✓ Write]

Step3 : Open vs code

- Open vs code with a folder .
- Create a file name is addfile.js
- Write the code
- Create a other file .env and write the your
  -PINATA_API_KEY
  -PINATA_SECRT_API_KEY
  -PINATA_JWT_TOKEN



```javascript
// Upload File to IPFS using Pinata
require("dotenv").config();
const axios = require("axios");
const FormData = require("form-data");
const fs = require("fs");

async function uploadToIPFS(filePath) {
                        const form = new FormData();
    form.append("file", fs.createReadStream(filePath));

    const res = await axios.post("https://api.pinata.cloud/pinning/pinFileToIPFS", form, {
        maxBodyLength: "Infinity",
        headers: {
            ...form.getHeaders(),
            pinata_api_key: process.env.PINATA_API_KEY,
            pinata_secret_api_key: process.env.PINATA_SECRET_API_KEY,
        },
    });

    console.log("📤 File pinned to IPFS!");
    console.log("🔗 Hash:", res.data.IpfsHash);
}

uploadToIPFS("sample1.txt")
    .then(() => console.log("✅ Upload complete!"))
    .catch((err) => console.error("❌ Error uploading file:", err.message));
// Ensure you have a file named 'sample.txt' in the same directory as this script.

```

Implementation Phase: Final Output (no error)

1.Open terminal

•Write code

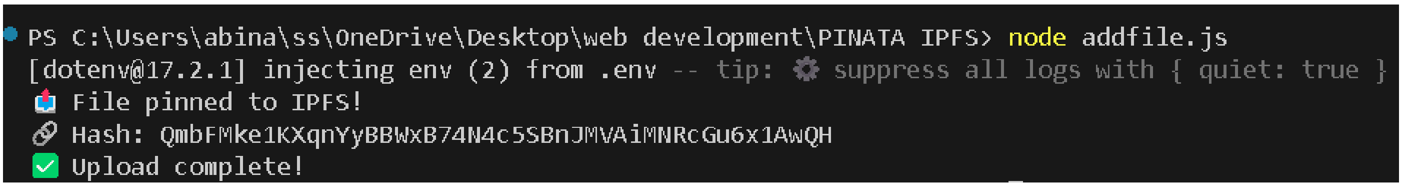
- Npm init –y

- npm install axios dotenv form-data

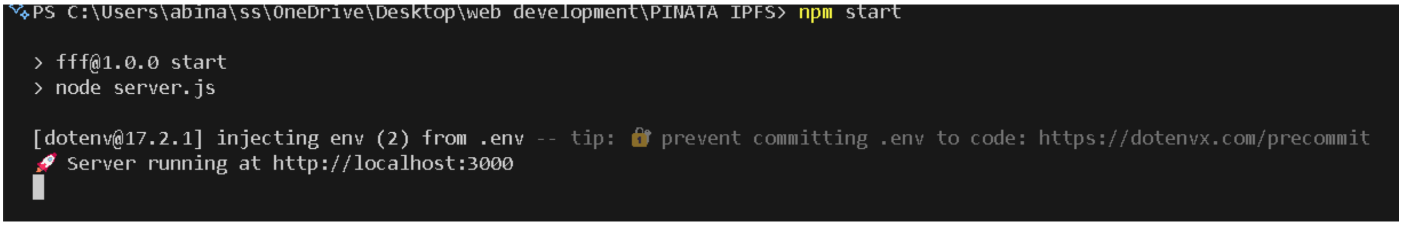
- node addfile.js

2.Out put

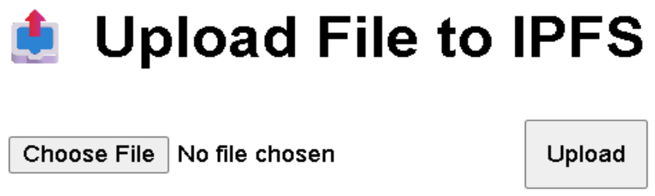
```
{

  IpfsHash: 'QmPzxMqJYPUJarnkEBrzsv9CbPabUMxTv8w7hhEeWuWRvS',

  PinSize: 31,

  Timestamp: '2025-07-31T17:40:45.189Z',

  ID: 'f5b06438-2dd7-4dd0-ac96-194919100a97',
  Name: 'sample.txt',

  NumberOfFiles: 1,

  MimeType: 'text/plain',

  GroupId: null,

  Keyvalues: null

}
```



```
PS C:\Users\abina\ss\OneDrive\Desktop\web development\PINATA IPFS> node addfile.js
[dotenv@17.2.1] injecting env (2) from .env -- tip: ⚙ suppress all logs with { quiet: true }
📤 File pinned to IPFS!
🔗 Hash: QmbFMke1KXqnYyBBWxB74N4c5SBnJMVAiMNRcGu6x1AwQH
✅ Upload complete!
```

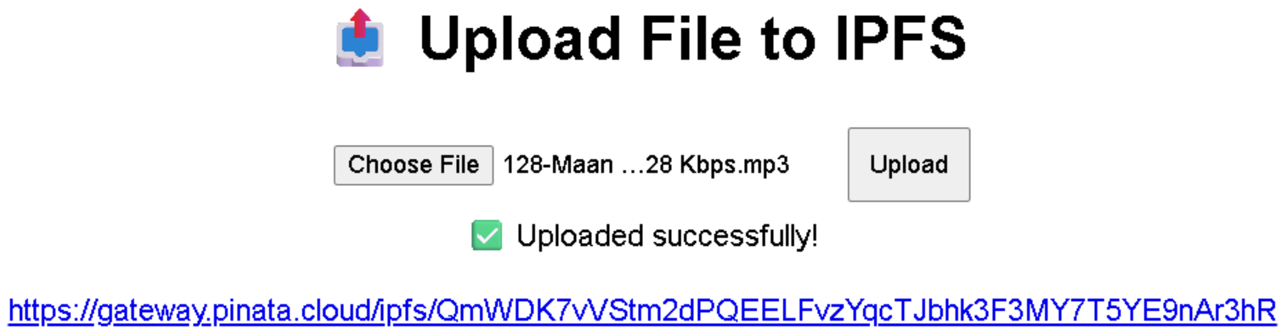


```
PS C:\Users\abina\ss\OneDrive\Desktop\web development\PINATA IPFS> npm start

> fff@1.0.0 start
> node server.js

[dotenv@17.2.1] injecting env (2) from .env -- tip: 🔐 prevent committing .env to code: https://dotenvx.com/precommit
🚀 Server running at http://localhost:3000
```



📤 Upload File to IPFS

No file chosen



📤 Upload File to IPFS

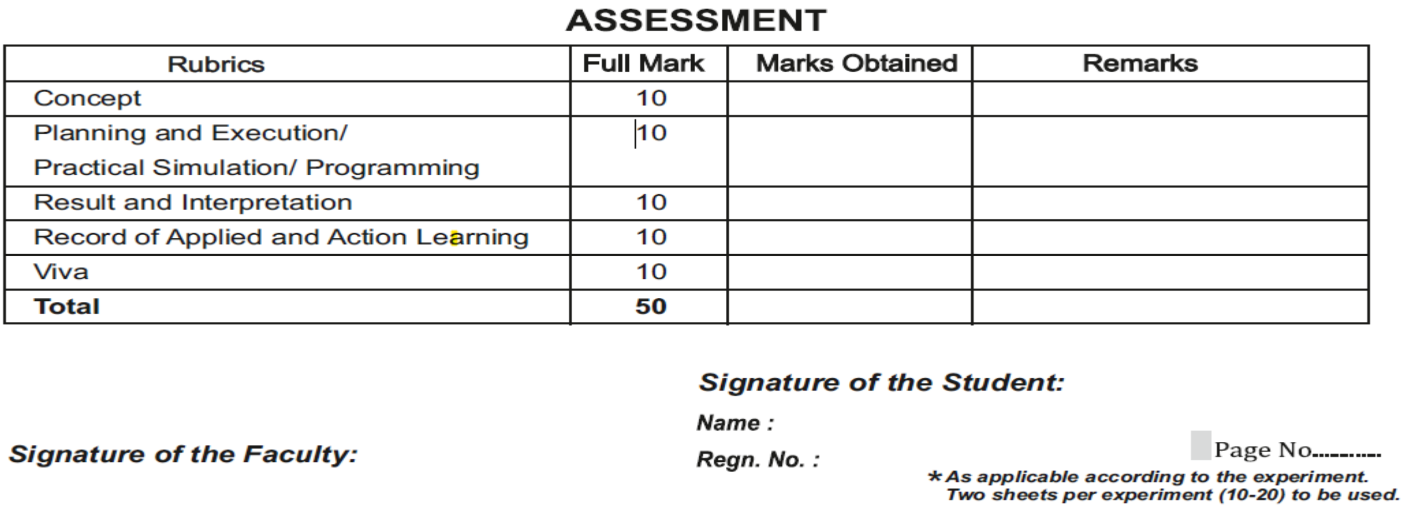
128-Maan …28 Kbps.mp3

✅ Uploaded successfully!

https://gateway.pinata.cloud/ipfs/QmWDK7vVStm2dPQEELFvzYqcTJbhk3F3MY7T5YE9nAr3hR

Observations

- File was uploaded successfully to IPFS using **Pinata’s Web UI**
- A unique **CID** was assigned to the file
- Anyone with the **gateway link** can access the file
- No coding or local node was required — just browser + internet
- Very useful for storing and sharing decentralized files



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 :

Signature of the Faculty:

Regn. No. :

****As applicable according to the experiment.***
Two sheets per experiment (10-20) to be used.