

Know Your TX – Dissecting a Transaction

School: ............................................................................................................. Campus: ....................................................... Academic Year: ...................... Subject Name: ........................................................... Subject Code: ..........................

Semester: ............... Program: ........................................ Branch: ......................... Specialization: .......................... Date: .....................................

(Learning by Doing and Discovery)

**\* Coding Phase: Pseudo Code / Flow Chart / Algorithm**

ALGORITHM:

1. Start.
2. Create a .env file and add API credentials (Pinata keys & token).
3. Import the required modules (fs, axios, dotenv, form-data).
4. Define an asynchronous function uploadToIPFS().
5. Inside the function:

* Read the file using fs.createReadStream().
* Append the file to FormData.
* Post the FormData to Pinata API endpoint.

1. Handle the API response:

* Display the IPFS hash and gateway link.

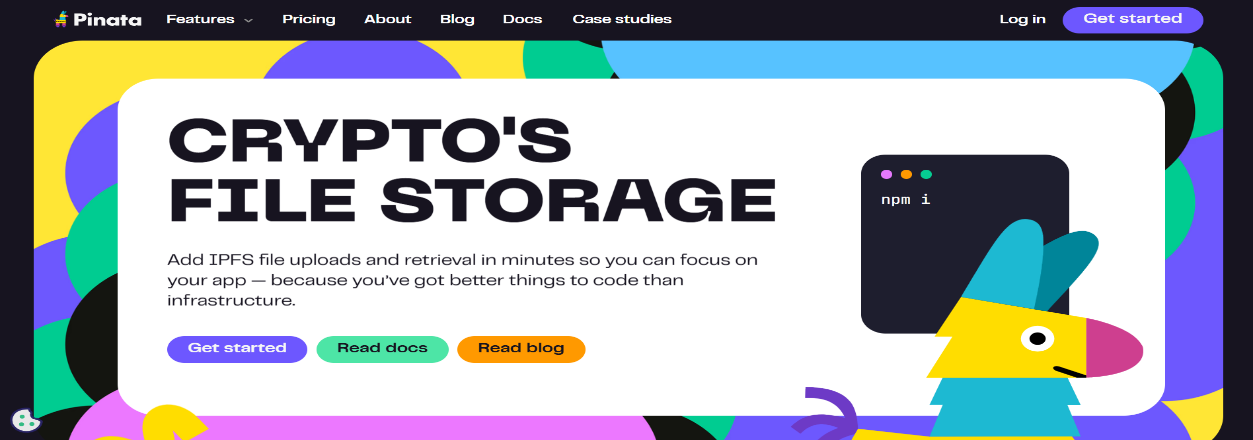
1. Handle any errors (invalid keys, connection issues).
2. End

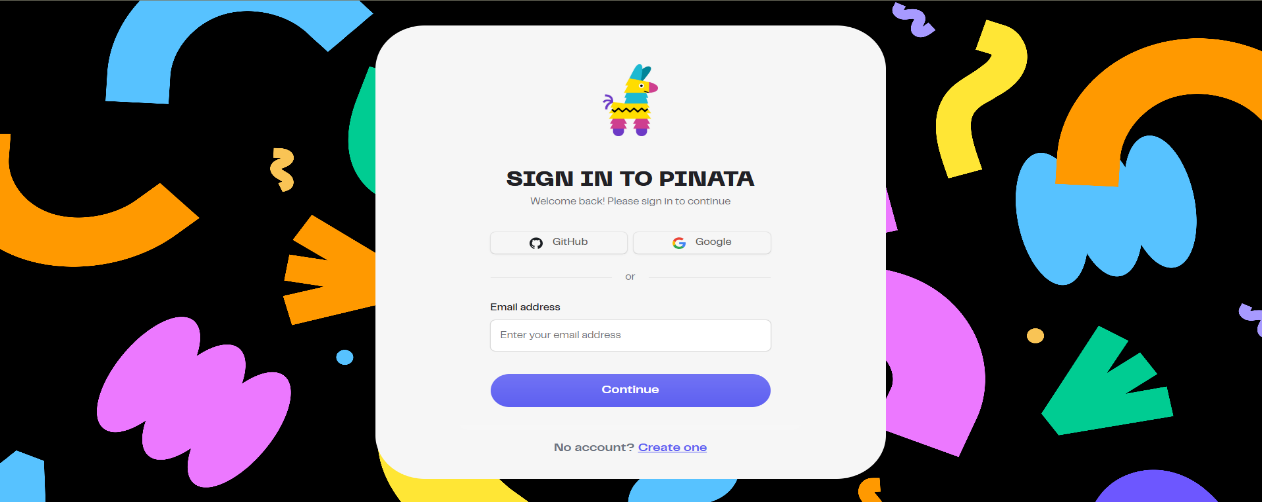
# \* Software Used:

1. **Node.js**
2. **Axios**
3. **Dotenv**
4. **Form-data**
5. **Pinata (IPFS storage service)**

**Step-1: Open Pinata Website:**

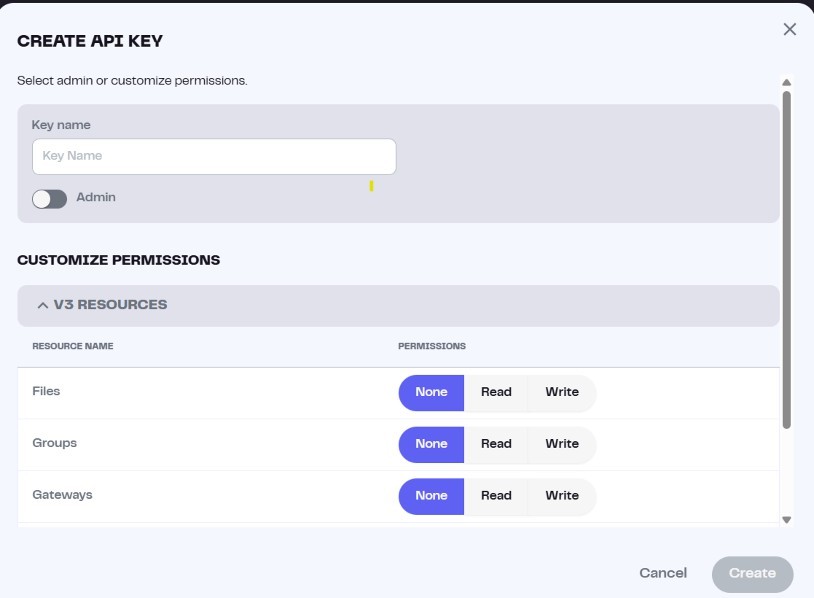
* Visit pinata. cloud.
* Login or sign up with your Gmail Account.



****

**Step-2:** **Create an API Key in Pinata:**

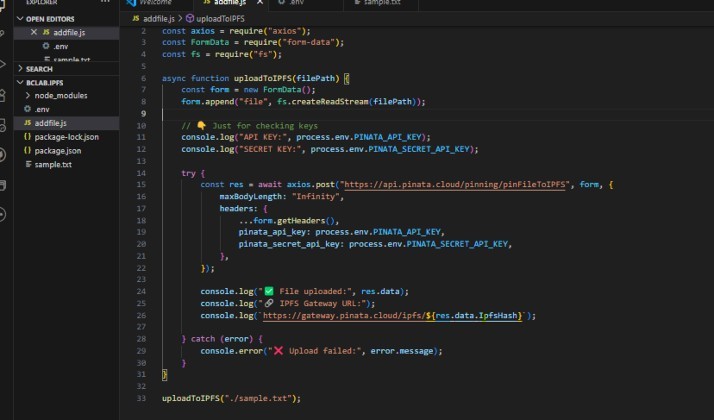
* Go to API Keys then Click New Key.
* Provide the name and Select Admin.
* Generate API Key.



Step-3: Open **VS Code:**

* First we will Open VS Code and create a new project folder.
* Inside the folder, create:

1. **addfile.js** → Write the code for uploading file to IPFS.
2. **.env** → Store your API credentials:
   * + - * **PINATA\_API\_KEY=xxxx**
         * **PINATA\_SECRET\_API\_KEY=xxxx**
         * **PINATA\_JWT\_TOKEN=xxxx**



Step-4: Open terminal and Initialize Node project:

* npm init -y
* npm install axios dotenv form-data

Step-5: Run the script:

* node addfile.js

Step-6: Output after running the script:

Json document code:

{

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',

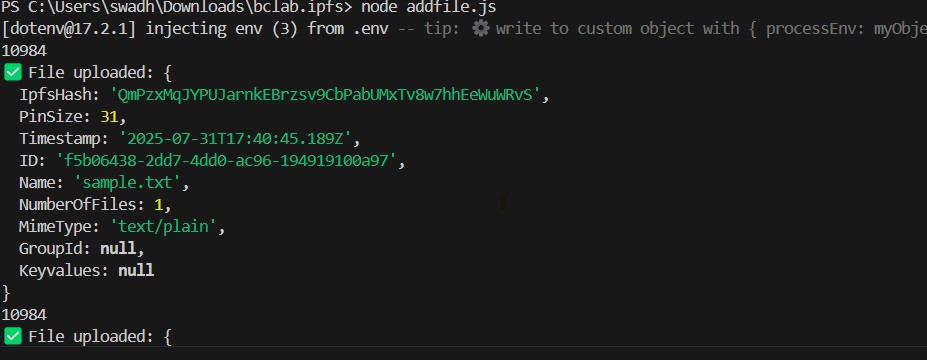
GroupId: null,

Keyvalues: null

}

Step-7: Observe the output response with IPFS Hash, Timestamp, and File Details.

## \* Implementation Phase: Final Output (no error) Applied and Action Learning



# Observation:

1. File was successfully uploaded to IPFS using Pinata’s API.
2. A unique CID (Content Identifier) was generated for the file.
3. The uploaded file can be accessed by anyone using the gateway link.
4. No local IPFS node setup was required; only browser + internet + API keys were sufficient.
5. This method is efficient for decentralized file storage and sharing.



|  |  |  |  |
| --- | --- | --- | --- |
| **Rubrics** |  |  |  |
| 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:***



***Signature of the Faculty:***