



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 :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Setting Up IPFS with Pinata, Creating API Keys, and Automating File Uploads with JavaScript

1. Create a Pinata Account and Get API Keys

- First, head over to Pinata.cloud and sign up for a free account.
- Confirm your email address to activate it.
- Once inside your dashboard, navigate to API Keys and click on New Key.
- While generating the key, make sure to enable:
 - pinFileToIPFS → for uploading files
 - pinJSONToIPFS → (optional) for adding JSON/metadata
- After creation, copy down the following values safely:
 - API Key
 - API Secret
 - JWT (Bearer Token) → this will be used for authentication

2. Automating Uploads with Node.js (JavaScript)

- Project Setup
 - Install Node.js on your computer if not already installed.
 - Open PowerShell and create a new project folder:

```
mkdir pinata-upload
cd pinata-upload
npm init -y
```
 - Install the required dependencies:

```
npm install @pinata/sdk dotenv axios form-data
```
 - Configure Environment Variables
 - Inside the root directory, create a file named .env and place your keys inside it:
 - PINATA_API_KEY=your_api_key_here
 - PINATA_API_SECRET=your_api_secret_here
 - PINATA_JWT=your_jwt_token_here
 - Write the Upload Script
 - Create a file called upload.js in your project folder.
 - In this script, you will use Pinata's SDK along with your .env values to send files to IPFS automatically, and the script will return the IPFS hash (CID) once the upload is complete.

Coding Phase: Pseudo Code / Flow Chart / Algorithm

addFile.js:

```
require('dotenv').config();
const express = require('express');
const multer = require('multer');
const fs = require('fs');
const axios = require('axios');
const FormData = require('form-data');
const cors = require('cors');

const app = express();
const PORT = 4000;
const upload = multer({ dest: 'uploads/' });

app.use(cors());
app.use(express.json());

// POST /upload - Upload file to Pinata
app.post('/upload', upload.single('file'), async (req, res) => {
  try {
    const filePath = req.file.path;

    const form = new FormData();
    form.append('file', fs.createReadStream(filePath));

    const response = 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,
        },
      }
    );

    const ipfsHash = response.data.ipfsHash;
    const fileName = req.file.originalname;

    // Store uploaded files in a local JSON file
    let files = [];
  }
});
```

```

if (fs.existsSync('uploadedFiles.json')) {
  files = JSON.parse(fs.readFileSync('uploadedFiles.json', 'utf8'));
}

files.push({ name: fileName, hash: ipfsHash });
fs.writeFileSync('uploadedFiles.json', JSON.stringify(files, null, 2));

res.json({ success: true, hash: ipfsHash });
} catch (error) {
  console.error('Upload error:', error.message);
  res.status(500).json({ success: false, error: error.message });
}
});

// GET /files - Return uploaded files
app.get('/files', (req, res) => {
  if (fs.existsSync('uploadedFiles.json')) {
    const files = JSON.parse(fs.readFileSync('uploadedFiles.json', 'utf8'));
    res.json(files);
  } else {
    res.json([]);
  }
});

app.listen(PORT, () => console.log(` ✅ Backend server running at
http://localhost:\${PORT}`));

```

- **Run the Script In the powershell:**
node addFile.js

Output:

You'll get the IPFS Hash (CID) and a link to view the file.

* Softwares used

Development Environment / Tools

Frontend (React):

- VS Code
- React.js
- HTML/CSS
- Browser
- Terminal

Backend (Node.js):

- Node.js, Express.js
- Axios, Multer, Form-data
- Dotenv, CORS

* **Testing Phase: Compilation of Code (error detection)**

No Error

* Implementation Phase: Final Output (no error)

The screenshots illustrate the final output of the implementation phase, showing no errors.

- Screenshot 1:** The main dashboard page for "CRYPTO'S FILE STORAGE". It features a large yellow banner with the text "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." Below the banner are three buttons: "Get started", "Read docs", and "Read blog". To the right is a sidebar with sections for IPFS (Files, Groups, Gateways, Analytics), DEVELOPER (API Keys, Webhooks, Access Controls), and EXTENSIONS (Marketplace, Integrations). A central "FILES" section shows a "PUBLIC" tab and a "PRIVATE" tab. A message box says "NO PUBLIC FILES YET". On the far right is an "Onboarding Checklist" with four items: 1. Upload your first file, 2. Create your API keys, 3. Organize files into a group, and 4. Explore the documentation.
- Screenshot 2:** A modal window titled "CREATE API KEY". It asks to "Select admin or customize permissions". A key name "ExPinataKey" is entered, and "Admin" is selected. Below is a "CUSTOMIZE PERMISSIONS" section with "V3 RESOURCES" and "PERMISSIONS" tables for Files, Groups, and Gateways. Buttons for "Cancel" and "Create" are at the bottom.
- Screenshot 3:** A modal window titled "FILE UPLOAD". It asks to "Confirm the name" with "Name: gitignore" and "Privacy Settings: Public". A "FILE UPLOAD" button is visible.
- Screenshot 4:** The "API KEYS" page. It shows a table with columns: NAME, API Key, API Secret, and JWT (secret access token). One row is shown with the name "1stPinataKey", API Key "f336b3cefb9ddade9ac2", and API Secret and JWT fields containing redacted text. A "Copy All" button is present. To the right is a table showing API key usage: TIMES USED, STATUS, and REVOKE.
- Screenshot 5:** The "FILES" page. It shows a table with columns: NAME, CID, SIZE, CREATION DATE, and FILE ID. Two files are listed: "sample.txt" (CID: Qm...AwQH, Size: 6 B, Creation Date: 7/31/2025) and ".gitignore" (CID: bafk...hjrca, Size: 148 B, Creation Date: 7/30/2025).

Page No.....

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

Implementation Phase: Final Output (no error)

CLI at Frontend

```
Windows PowerShell x + v
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhis\OneDrive\Desktop\PINATA>npx create-react-app pinata-frontend
Creating a new React app in C:\Users\abhis\OneDrive\Desktop\PINATA\pinata-frontend.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

You can now view pinata-frontend in the browser.

Local: http://localhost:3000
On Your Network: http://10.13.5.43:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully
Compiling...
Compiled successfully!

CLI at Backend

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhis\OneDrive\Desktop\PINATA\pinata-project>npm install express multer axios dotenv form-data cors
added 78 packages, and audited 103 packages in 3s

18 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

C:\Users\abhis\OneDrive\Desktop\PINATA\pinata-project>node addFile.js
[dotenv@17.2.1] injecting env (3) from .env -- tip: ⚙️ enable debug logging with { debug: true }
✔ Backend server running at http://localhost:4000
```

* Implementation Phase: Final Output (no error)

Applied and Action Learning

* Observations

Files can be uploaded through a web interface to IPFS, a decentralized storage network, and later accessed using a unique content identifier (CID), eliminating the need for traditional centralized cloud services.

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. :

Page No.....