



NLC INDIA LIMITED

“NAVRATNA” – A GOVERNMENT OF INDIA ENTERPRISE

INTERNSHIP REPORT ON

WEB DEVELOPMENT

SUBMITTED BY

NIRAINJAN C (412522104097)

HARINARAYANAN A (412522104046)

SANTHOSH S (412522104132)

B.E COMPUTER SCIENCE AND ENGINEERING



SRI SAIRAM ENGINEERING COLLEGE

(AUTONOMOUS)

Sai Leo Nagar, West Tambaram, Chennai – 600 044.

UNDER THE GUIDANCE OF

Mr. STANLEY SAHAYARAJ A

CHIEF MANAGER/COMPUTER SECTION

CORPORATE OFFICE/NLC INDIA LIMITED



NLC INDIA LIMITED

**“NAVRATNA” – A GOVERNMENT OF INDIA ENTERPRISE
NEYVELI – 607 801, TAMIL NADU**

BONAFIDE CERTIFICATE

This is to certificate that the project entitled in “Web Development” is the Bonafide record of the work done by the following students

NIRAINJAN C (412522104097)

HARINARAYANAN A (412522104046)

SANTHOSH S (412522104132)

The Internship work done during the period from 07-01-2025 to 21-01-2025 at
Corporate Office, NLC India Limited, Neyveli, TamilNadu

**Mr. STANLEY SAHAYARAJ A,
CM/CS/CO/CPF:36845/NLCIL,
NLCIL, NEYVELI.**

Permitted to submit the report to university authorities.

PLACE: Neyveli

DATE:

**CHIEF MANAGER / L&DC,
LEARNING AND DEVELOPMENT CENTRE,
NLCIL, NEYVELI.**

DECLARATION

We hereby declare that the internship at Corporate Office, NLC India Limited, Neyveli submitted to department of computer science and engineering at Sri Sairam Engineering College Chennai, is a record of original work done ourself under the guidance MR. STANLEY SAHAYARAJA/CM/CS/CO/CPF:36845.

This internship work is only for reference only and no part of the report will be published copied anywhere without the written permission from NLCIL officials, NLCIL, Neyveli.

SIGNATURE OF THE STUDENTS:

NIRAINJAN C (412522104097)

HARINARAYANAN A (412522104046)

SANTHOSH S (412522104132)

ACKNOWLEDGEMENT

I express my sincere thanks to **NLC INDIA LIMITED** for providing me the opportunity to carry out the Internship Training.

We sincerely thank **Shri. SARAVANABHAVAN A. Kr, GM**, Unit Head / Learning & Development Centre, **Shri. K. PRABAKARAN, DGM**, Learning & Development Centre, for their kind patronage and all the facilities offered to do this internship successfully.

I sincerely thank **Mr. STANLEY SAHAYARAJ A/CM/CS/CO/CPF:36845**, for providing guidance for the internship training program during the 28 days in NLCIL NEYVELL

I sincerely thank all the respectful dignitaries in the corporate office for providing great support for successful completing the internship training in the NLCIL, NEYVELI.

I sincerely thank **Dr. B Latha Head of Department of Computer Science Engineering in SRI SAIRAM ENGINEERING COLLEGE, CHENNAI** for providing this opportunity.

I would have never succeeded in completing my task without the cooperation and help of executives, supervisors and others personalities of Corporate Office.

My acknowledgements are also to many others, who have rendered their help in many occasions during my training period.

INDEX

S.NO	TITLE	PAGE.NO
1	INTRODUCTION <ul style="list-style-type: none">• Lignite and Coal Mines• Thermal Power Generation• Neyveli New Thermal Power Station• Operational and Safety Management (OSMG)• NLC India Ltd.	7
2	WEBSITE 2.1 About Website 2.2 Building of Website	10
3	LANGUAGES 3.1 HTML 3.2 CSS 3.3 JAVASCRIPT	12
4	TOOLS AND FRAMEWORK	15
5	DATABASE	18
6	WEBSITE DEVELOPMENT	19
7	SOURCE CODE	20
8	CONCLUSION	28
9	LEARNINGS	30

ABSTRACT

This web-based project aims to create a File Management System for efficient document handling. The application allows users to upload PDF files along with their respective names and descriptions. The uploaded data is securely stored in an Excel sheet for organized record-keeping and is dynamically displayed in a table format on the website. Users can also view individual files using the integrated "View" button for better accessibility.

The project is developed using HTML, CSS, and Node.js, employing modern web development practices. The frontend, built with HTML and CSS, ensures a responsive and user-friendly interface, while the backend leverages Node.js with the Multer library for file uploads and the xlsx library for Excel file manipulation.

Key features of the project include a file upload system that allows users to upload PDF files with names and descriptions, Excel integration for storing file details in an organized manner, a dynamic table for seamless display of uploaded files, and a "View" button for accessing files directly. These features ensure efficient document handling, making the system suitable for various applications like business workflows, academic portals, and personal file management. This modular and scalable approach makes it a practical solution for managing documents effectively.

1. INTRODUCTION

NLC India Limited (NLCIL), formerly known as Neyveli Lignite Corporation Limited, is a prominent Indian government-owned enterprise primarily engaged in mining and power generation. Established in 1956, NLCIL operates under the administrative control of the Ministry of Coal. Over the years, NLCIL has expanded its operations across various domains, including lignite mining, coal mining, thermal power generation, renewable energy projects, and more.

1.1 Lignite and Coal Mining

NLCIL is one of the largest lignite mining companies in India. Its mining operations are primarily located in Neyveli, Tamil Nadu, where it operates several open-cast lignite mines. In addition to lignite, NLCIL has also ventured into coal mining, diversifying its resource base and enhancing its capacity to meet the energy demands of the country.

1.2 Thermal Power Generation

NLCIL has made significant strides in thermal power generation, with several thermal power plants contributing to India's energy grid. These plants use lignite and coal as primary fuels, harnessing the energy contained within these resources to generate electricity. Key thermal power plants include:

- **Thermal Power Station I (TPS-I):** One of the oldest plants, NTPS-I has been a cornerstone of NLCIL's power generation capabilities.
- **Thermal Power Station II (TPS-II):** A more advanced facility that continues to be a major power supplier.
- **Neyveli New Thermal Power Station (NNTPS):** This plant represents NLCIL's commitment to adopting newer technologies and increasing efficiency in power generation.

1.3 Neyveli New Thermal Power Station (NNTPS)

The Neyveli New Thermal Power Station (NNTPS) is a modern facility that underscores NLCIL's focus on upgrading its infrastructure to meet contemporary standards of efficiency and environmental sustainability. Equipped with state-of-the-art technology, NNTPS is designed to produce higher output with lower emissions, contributing to the company's goal of sustainable energy production.

1.4 OPERATING SERVICES MONITORING GROUP (OSMG)

Operation Services Monitoring group-Thermal (OSMG Thermal) was established in NLCIL in the year 2017 with the objective of continuous monitoring and improving the performance, efficiency and availability of all thermal power station in NLC India Limited and Joint Venture companies. The group assists Power Plant Engineers in detecting performance and operational, anomalies, enabling them to take corrective actions and move to a predictive-maintenance system by continuously monitoring system.

OSMG performs many plant improvement activities such as:

1. Daily plant performance report to Director Secretariat
2. Conducting Quarterly review performance meeting for all stations.
3. Instrumental in conducting knowledge sharing workshops in all stations.
4. Analyzing trip reports and giving performance improvement suggestions to all stations.
5. Performing pre-ort and ort meetings for all stations.
6. Conducting Technical Audits for all stations.
7. Visiting all plants during Annual Overhaul.
8. Deputing Knowledge teams to Stations for Analysis purpose.
9. Maintaining a database of all the critical parameters of all the stations.
10. Pivotal in releasing technical documents for all Plants.

1.5 NLC INDIA

NLC India Limited (NLCIL), formerly Neyveli Lignite Corporation Limited, operates as a profit making Navratna Government of India (GoI) Enterprise, functioning under the administrative control of Ministry of Coal. NLCIL organises its businesses under Mining (Lignite and Coal), Power Generation (including thermal, solar and wind) and Power Trading. NLCIL serves as an important source of power generation to the states of Tamil Nadu, Andhra Pradesh, Karnataka, Kerala, Telangana, Rajasthan, and Union Territory of Puducherry.

NLCIL currently operates four opencast lignite mines with a combined capacity of 28.1 MTPA and one coal mine of 20.0 MTPA. During FY 2021-22, NLCIL has achieved an aggregate lignite production of 25.11 MT and coal production of 6.36 MT. The details of the existing operational mines are as shown in the table below:

Mines	Location	Total Reserves (MT)	Excavated till 31 st mar 2022	Capacity (MTPA)	Balance Reserve (MT)	Commissioning Year
Mine I	Neyveli, TamilNadu	463.92	380.09	8	83.83	1962
Mine II	Neyveli, TamilNadu	621.00	338.5	15	282.45	1985
Mine IA	Neyveli, TamilNadu	218.74	57.08	3	161.66	2003
Barsingsar	Rajasthan	53.00	15.27	2.1	37.73	2010
Talibara II & III OCP	Odisha	553.98	7.41	20	546.57	2021

NLCIL is currently operating five thermal power stations, four at Neyveli, Tamil Nadu and one at Barsingsar, Rajasthan with total capacity of 3,640 MW. NLCIL has 1,421 MW of renewable energy installed capacity, including solar power plants aggregating 1,370 MW and wind power plant of 51 MW. The Installed power generation capacity of NLCIL have been presented in the below table.

Projects	Location	Capacity (MW)	Commissioning Year
TPS-I Expansion	Neyveli, TamilNadu	420	2003
TPS-II	Neyveli, TamilNadu	1,470	1986-1993
TPS-II Expansion	Neyveli, TamilNadu	500	2015
Barsingsar TPS	Barsingsar, Rajasthan	250	2011-2012
Neyveli New Thermal Power Station (NNTPS)	Neyveli, TamilNadu	1,000	2021
Total Thermal		3,640	
Wind	Tirunelveli, TamilNadu	51	2017
Solar	Tamil, Nadu	10	2015
Solar	TamilNadu	130	2017
Solar	TamilNadu	1.06	2018

2. WEBSITE

The website's web pages are linked with hyperlinks and hypertext and share a common interface and design. The website might also contain some additional documents and files such as images, videos, or other digital assets.

With the Internet invading every sphere, we see websites for all kinds of causes and purposes. So, we can also say that a website can also be thought of as a digital environment capable of delivering information and solutions and promoting interaction between people, places, and things to support the goals of the organization it was created for.

Components of a Website: We know that a website is a collection of a webpages hosted on a web-server. These are the components for making a website.

Webhost: Hosting is the location where the website is physically located. Group of webpages (linked webpages) licensed to be called a website only when the webpage is hosted on the webserver. The webserver is a set of files transmitted to user computers when they specify the website's address.

Address: Address of a website also known as the URL of a website. When a user wants to open a website then they need to put the address or URL of the website into the web browser, and the asked website is delivered by the webserver.

Homepage: Home page is a very common and important part of a webpage. It is the first webpage that appears when a visitor visits the website. The home page of a website is very important as it sets the look and feel of the website and directs viewers to the rest of the pages on the website.

Design: It is the final and overall look and feel of the website that has a result of proper use and integration elements like navigation menus, graphics, layout, navigation menus etc.

Content: Every web pages contained on the website together make up the content of the website. Good content on the webpages makes the website more effective and attractive.

The Navigation Structure: The navigation structure of a website is the order of the pages, the collection of what links to what. Usually, it is held together by at least one navigation menu.

2.1 Building of a website

Phase 1: Research and Planning:

Before diving headfirst into the design process, conduct thorough research to understand your target audience, competitors and set clear objectives and goals for your website.

Phase 2: Sketch Out Your Website Map:

After brainstorming on the basic aims of your website, you have to understand the overall organization of the website. A website map is a desperately needed tool for planning and designing web sites.

Phase 3: Wireframing and Prototyping:

Wireframes are fundamental line drawings that allow you and your client to focus on the layout of the website. They are prototype models built for testing purposes. Wireframes show you how your website will look and feel so you can identify the functionality of the website even before engaging in development

Phase 4: Visual Design:

Visual design is where creativity flourishes. This phase involves crafting an aesthetically pleasing website by selecting color schemes, typography, imagery, and other visual elements that align with your brand identity. Striking a balance between aesthetics and usability is crucial to engage and retain visitors.

Phase 5: Content Creation:

Compelling content is the lifeblood of any website. In this phase, you develop and optimize content that aligns with your brand messaging, target audience, and SEO strategies. It includes crafting persuasive copy, creating engaging visuals, and optimizing the website for search engines.

3. LANGUAGES

3.1.HTML

HTML stands for Hyper Text Markup Language and it is used to create webpages. It uses HTML tags and attributes to describe the structure and formatting of a web page.

HTML consists of various elements, that are responsible for telling search engines how to display page content. For example, headings, lists, images, links, and more.

Features of HTML

- It is easy to learn and easy to use.
- It is platform-independent
- Images, videos, and audio can be added to a web page
- Hypertext can be added to the text.
- It is a markup language.

HTML Elements and Tags

HTML uses predefined tags and elements that instruct the browser on how to display the content. HTML elements include an opening tag, some content, and a closing tag. It's important to remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

This section will dive into the basic structure of an HTML. page, which includes essential building-block elements like doctype declaration, HTML, head, title, and body elements.

3.2.CSS

CSS (Cascading Style Sheets) is a language designed to simplify the process of making web pages presentable. It allows you to apply styles to HTML documents, describing how a webpage should look by prescribing colours, fonts, spacing, and positioning. CSS provides developers and designers with powerful control over the presentation of HTML. elements.

HTML uses tags and CSS uses rulesets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document

Why CSS?

- **Saves Time:** Write CSS once and reuse it across multiple HTML pages.
- **Easy Maintenance:** Change the style globally with a single modification.
- **Search Engine Friendly:** Clean coding technique that improves readability for search engines
- **Superior Styles:** Offers a wider array of attributes compared to HTML
- **Offline Browsing:** CSS can store web applications locally using offline cache, allowing offline viewing

SYNTAX:

`<h2>hello world </h2>`

CSS Style

`h1 { colour blue, f-size: 12px, }`

Where

Selector-h1

Declaration-{ colour: blue; font-size: 12px; }

3.3 JAVASCRIPT

JavaScript is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it.

JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both an imperative and declarative type of language

Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client side are AngularJS, ReactJS, VueJS and so many others.

Server-side: It supplies objects relevant to running JavaScript on a server. For if the server-side extensions allow an application to communicate with a database, and provide continuity of

information from one invocation to another of the application, or perform file matupulations on a server. The useful framework which is the most famous these days is node.js.

Imperative language: In this type of language we are mostly concerned about how it is to be done. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async await we are thinking about what is to be done further after the async call

Declarative programming: In this type of language we are concerned about how it is to be done basically here logical computation requires. Her main goal is to describe the desired result without direct dictation on how to get it as the arrow function does.

4. TOOLS AND FRAMEWORKS

BOOTSTRAP:

Bootstrap is the most popular open-source front-end framework for simplified web development. It provides a collection of HTML, CSS, and JavaScript components and tools that enable developers to easily build responsive, mobile-first websites.

Bootstrap is a free and open-source tool collection used to create responsive web pages and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Websites created using Bootstrap are compatible with all browsers (IE, Firefox, and Chrome) and with all screen sizes (Desktops, Tablets, Phablets, and Phones). Bootstrap was initially developed by Mark Otto and Jacob Thornton of Twitter, but it was later declared an open-source project.

Internet Information Services (IIS):

Internet Information Services (IIS) is a flexible, general-purpose web server from Microsoft that runs on Windows systems to serve requested HTML pages or files.

An IIS web server accepts requests from remote client computers and returns the appropriate response. This basic functionality allows web servers to share and deliver information across local area networks (LAN), such as corporate intranets, and wide area networks (WAN), such as the Internet.

REACT:

ReactJS, also known as React, is a popular JavaScript library for building user interfaces. It is also referred to as a front-end JavaScript library. It was developed by Facebook and is widely used for creating dynamic and interactive web applications. In this article, we'll explore the key concepts of React.

React operates by creating an in-memory virtual DOM rather than directly manipulating the browser's DOM. It performs necessary manipulations within this virtual representation before applying changes to the actual browser DOM. React is efficient, altering only what requires modification.

- A Front-End Web Development Tool is actually the software that allows the front-end developers to build the website layout and UI more efficiently and without any hassle. With the help of such tools the work of front-end web developers, especially the repetitive or monotonous tasks, gets reduced that subsequently fastens the web development process.

- There are numerous front-end web development tools out there for various specific requirements such as HTML, CSS, and JavaScript tools, Code-Editing tools, Deployment Tools, Prototyping & Wireframing tools, Security tools, and many more.

- **Chrome Dev Tools**
- **HTML5 Boilerplate**
- **Sass**
- **Visual Studio Code**

NODE:

Node.js is a powerful JavaScript runtime built on Chrome's V8 engine, allowing developers to execute JavaScript code outside the browser. It is primarily used for server-side programming and excels in building scalable, high-performance web applications. Node.js follows a non-blocking, event-driven architecture, enabling it to handle thousands of concurrent connections efficiently. This makes it particularly suitable for real-time applications, such as messaging platforms and gaming servers. With built-in modules like `http` and `fs`, Node.js simplifies tasks like handling requests, managing file systems, and creating servers. Its package manager, `npm` (Node Package Manager), offers access to thousands of libraries and tools, speeding up development. Node.js is known for its lightweight and fast execution, making it a preferred choice for modern web development.

EXPRESS:

Express.js is a minimalist web application framework for Node.js, designed to streamline the process of building servers and APIs. It provides a robust set of features, including routing, middleware support, and HTTP utilities, making it easier to handle requests, manage sessions, and serve static files. Express.js simplifies the development of RESTful APIs, enabling developers to create endpoints for data retrieval and manipulation. Its middleware architecture allows developers to add custom logic for authentication, error handling, and data validation. With its flexibility and compatibility with other Node.js libraries, Express.js is ideal for creating scalable backend systems. Whether you're building a simple web server or a complex API, Express.js provides the tools needed for rapid development.

MULTER:

Multer is a middleware library for handling multipart/form-data, a format commonly used for file uploads. Built for Node.js, it simplifies the process of handling file uploads by automatically parsing incoming requests and storing files in a specified directory. Developers can configure Multer to set file storage destinations, rename uploaded files, and enforce limits on file

size and type. This ensures secure and efficient file management in web applications. Multer integrates seamlessly with Express.js, making it a popular choice for projects involving user-uploaded files, such as profile pictures, documents, or media content. Its ability to handle large files efficiently is a key feature for modern web applications.

xlsx:

The xlsx library is a JavaScript package used to work with Excel files in .xlsx and .xls formats. It allows developers to create, read, and edit spreadsheets programmatically, making it useful for applications that involve data storage, reporting, or analysis. With xlsx, developers can export structured data into Excel files or import data from spreadsheets for processing. The library supports advanced Excel features, such as formulas, formatting, and data validation, providing robust functionality for managing tabular data. In the context of web development, xlsx is ideal for projects requiring dynamic report generation or bulk data uploads, enhancing productivity and user convenience.

FILE SYSTEM (fs):

The File System module in Node.js provides a set of APIs for interacting with the operating system's file system. It allows developers to perform essential operations like reading, writing, deleting, and updating files or directories. The fs module supports both synchronous and asynchronous methods, giving developers flexibility in handling file-related tasks. It is commonly used for managing static assets, logging application activity, and storing user-uploaded files. In web applications, the fs module enables developers to persist data on the server or retrieve resources dynamically, making it an essential tool for backend development.

5. DATABASE

A database is an organized collection of data stored in a computer system and usually controlled by a database management system (DBMS). The data in common databases is modeled tables, making querying and processing efficient Structured query language (SQL) is commonly used for data querying and writing.

The Database is an essential part of our life. We encounter several activities that involve our interaction with databases, for example in the bank, in the railway station, in school, in a grocery store, etc. These are the instances where we need to store a large amount of data in one place and fetch these data easily

What is Data?

Data is statically raw and unprocessed information. For example-age, sex, marks, etc. In computer language, a piece of information that can be translated into a form for efficient movement and processing is called data. Data is interchangeable information

What is a Database?

A database is a collection of data that is organized, which is also called structured data. It can be accessed or stored in a computer system. It can be managed through a Database Management System (DBMS), a software used to manage data. Database refers to related data in a structured form.

Most Popular Database Management Systems:

Oracle: Oracle DBMS, an effective object-relational DBMS, a multi-model database management system, is mostly used for processing online transactions, and data warehousing. Its latest version is 12c where c stands for cloud computing. The supporting systems are Windows, UNIX, and Linux.

MySQL: MySQL (My-Structured Query Language), is a free, open-source relational and again a multi-model database management system. A reliable, cost-effective tool that is implemented to enhance the security and scalability of the database. Its functionality includes high-speed data processing and data recovery increases its popularity. It supports Windows, Linux, Mac, and Ubuntu. Large organizations such as Google, Adobe, Facebook, and WordPress use this tool for the database management system.

PostgreSQL: PostgreSQL is a free, advanced, and open-source relational database management system. Exporting and importing data is a key functionality this tool possesses. Combining features like storing and scaling complex data workloads. To enhance functionality, it has a wide range of plug-ins. It supports programming languages like JSON and Python.

6. WEBSITE DEVELOPMENT

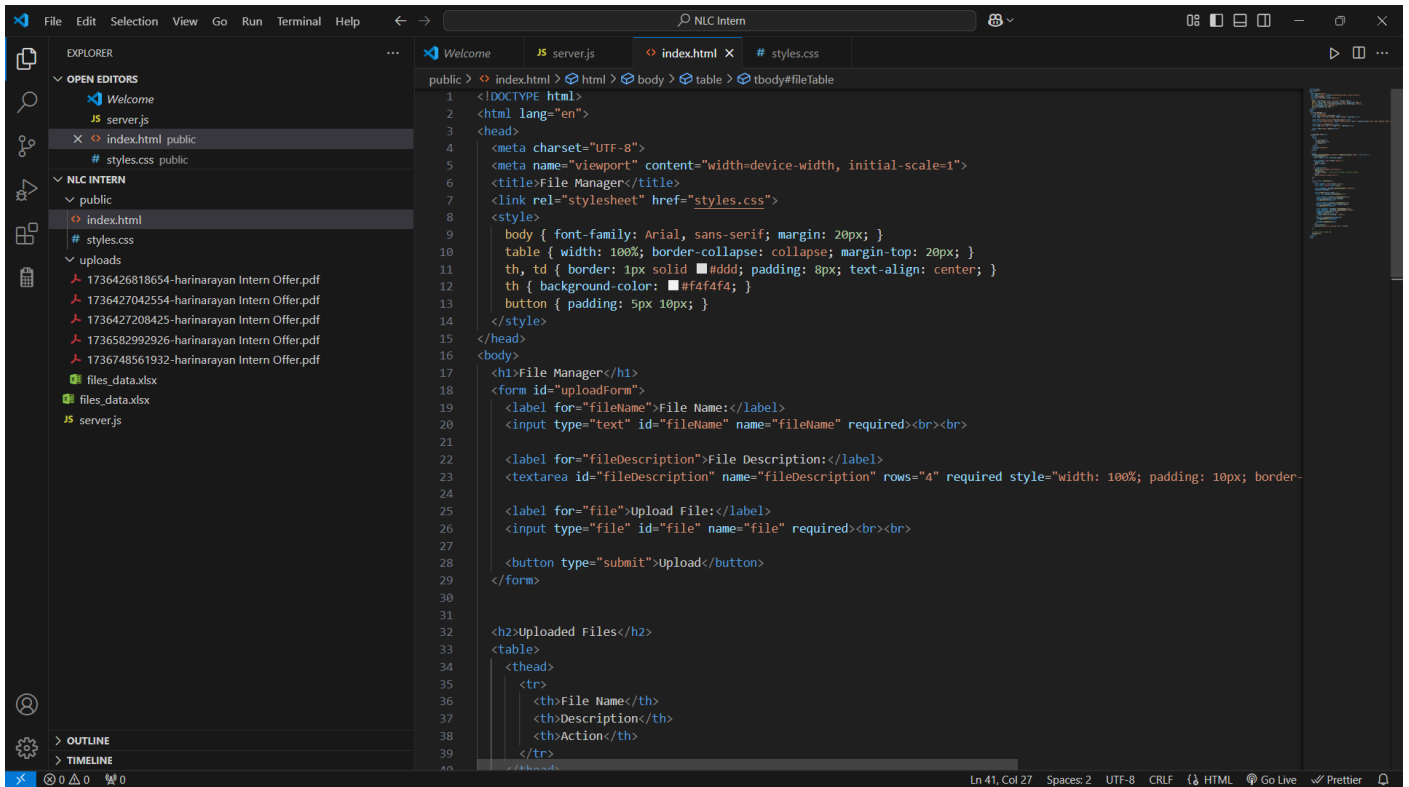
This project aims to develop a File Management System that allows users to upload, view, and manage files through a web-based interface. The system is designed to simplify file organization and retrieval by enabling users to input file details, upload files, and store metadata (like file names and descriptions) in an Excel file. Users can also view a list of uploaded files and access them conveniently through the application.

The project incorporates dynamic functionalities using modern web development technologies like HTML, CSS, JavaScript, and Node.js, along with file handling and storage capabilities. It is designed to provide an intuitive user experience with responsive layouts and efficient backend management.

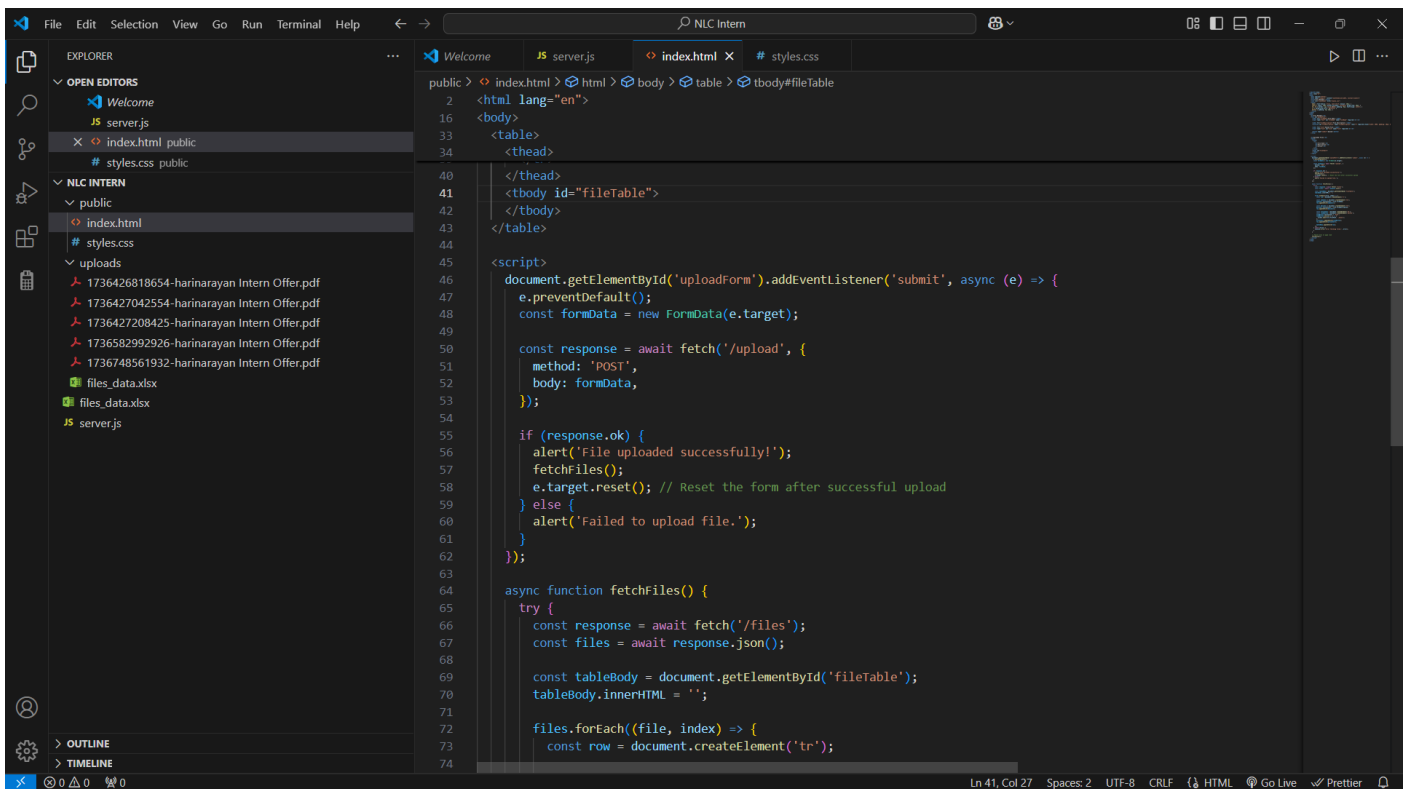
Components Needed:

- Web Browser
- Local Server
- Operating System
- Text Editor or IDE
- Express.js Framework
- Multer Middleware
- xlsx Library

7. SOURCE CODE



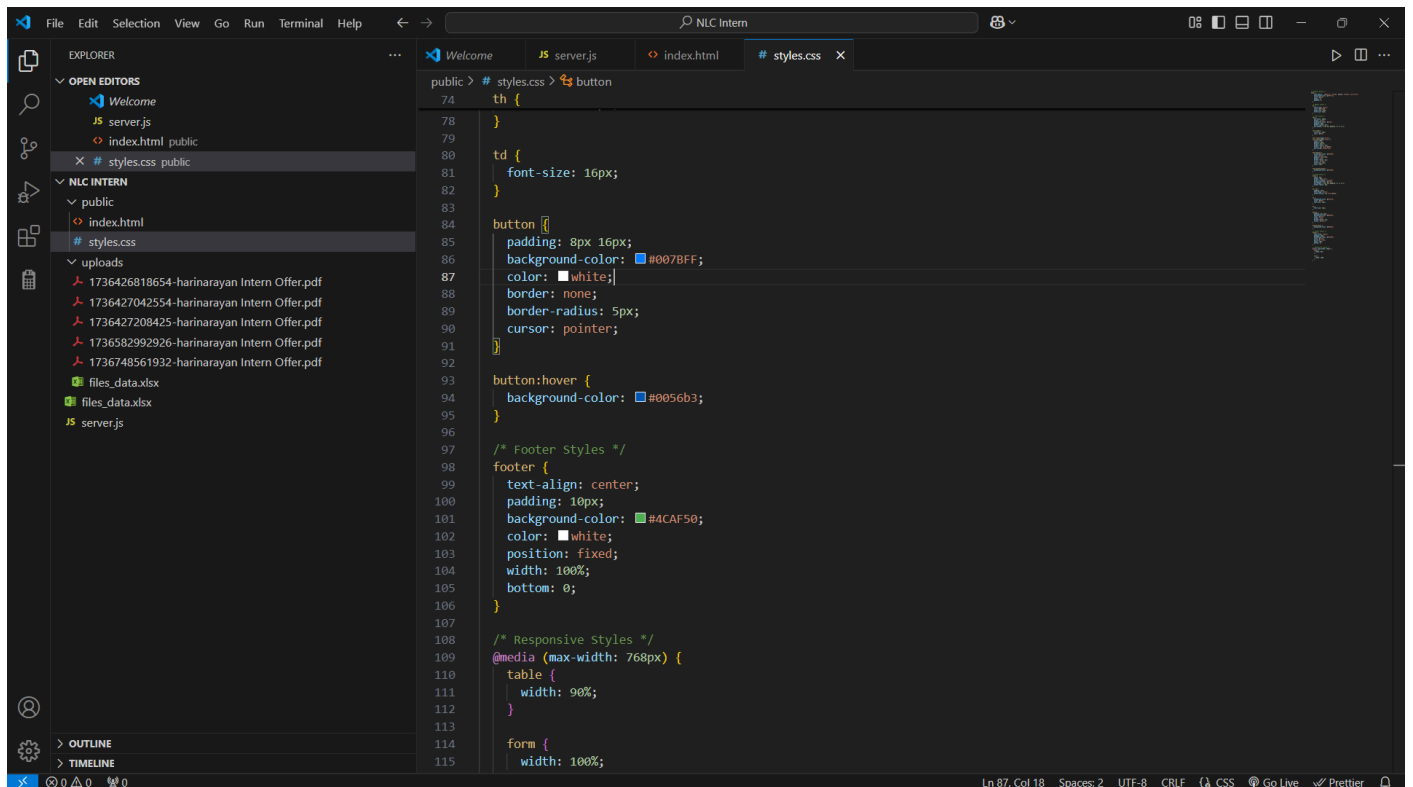
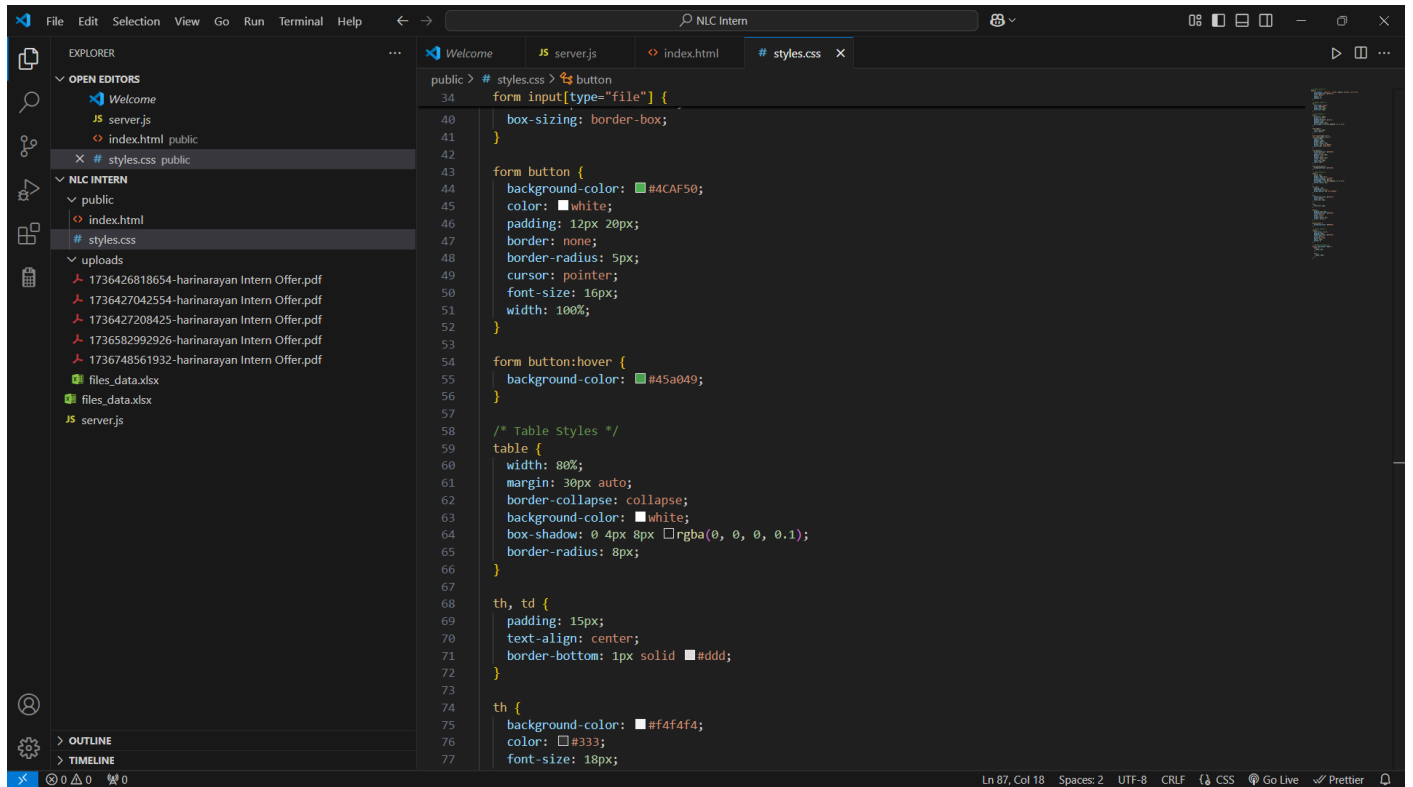
```
public > index.html > html > body > table > tbody#fileTable
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>File Manager</title>
7    <link rel="stylesheet" href="styles.css">
8    <style>
9      body { font-family: Arial, sans-serif; margin: 20px; }
10     table { width: 100%; border-collapse: collapse; margin-top: 20px; }
11     th, td { border: 1px solid #ddd; padding: 8px; text-align: center; }
12     th { background-color: #f4f4f4; }
13     button { padding: 5px 10px; }
14   </style>
15 </head>
16 <body>
17   <h1>File Manager</h1>
18   <form id="uploadForm">
19     <label for="fileName">File Name:</label>
20     <input type="text" id="fileName" name="fileName" required><br><br>
21
22     <label for="fileDescription">File Description:</label>
23     <textarea id="fileDescription" name="fileDescription" rows="4" required style="width: 100%; padding: 10px; border-
24
25     <label for="file">Upload File:</label>
26     <input type="file" id="file" name="file" required><br><br>
27
28     <button type="submit">Upload</button>
29   </form>
30
31
32   <h2>Uploaded Files</h2>
33   <table>
34     <thead>
35       <tr>
36         <th>File Name</th>
37         <th>Description</th>
38         <th>Action</th>
39       </tr>
40     </thead>
```



```
public > index.html > html > body > table > tbody#fileTable
2  <html lang="en">
16 <body>
33   <table>
34     <thead>
40       </thead>
41     <tbody id="fileTable">
42     </tbody>
43   </table>
44
45   <script>
46     document.getElementById('uploadForm').addEventListener('submit', async (e) => {
47       e.preventDefault();
48       const formData = new FormData(e.target);
49
50       const response = await fetch('/upload', {
51         method: 'POST',
52         body: formData,
53       });
54
55       if (response.ok) {
56         alert('File uploaded successfully!');
57         fetchFiles();
58         e.target.reset(); // Reset the form after successful upload
59       } else {
60         alert('Failed to upload file.');
```

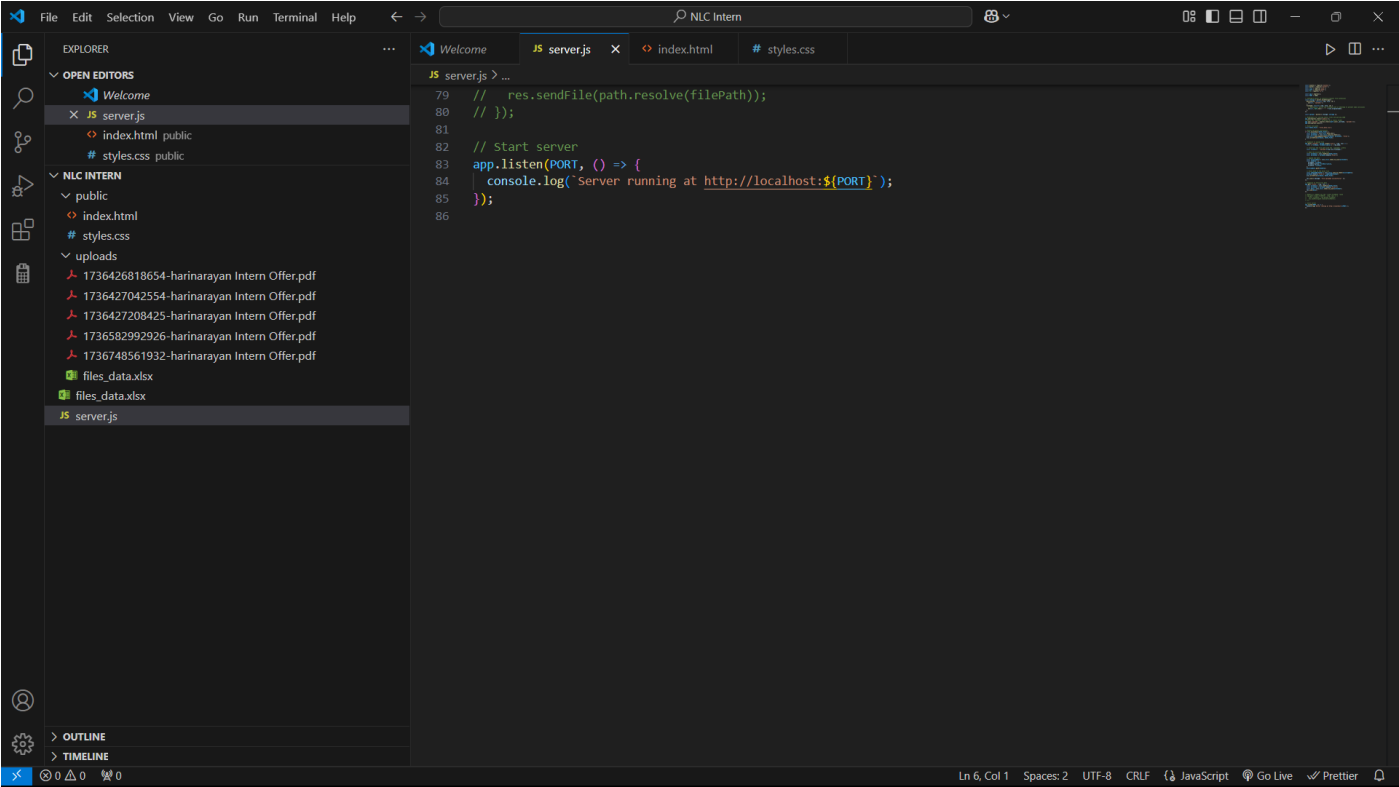
```
public > index.html > html > body > script > fetchFiles
2 <html lang="en">
16 <body>
45 <script>
64   async function fetchFiles() {
72     files.forEach((file, index) => {
74
75       const nameCell = document.createElement('td');
76       nameCell.textContent = file.FileName;
77       row.appendChild(nameCell);
78
79       const descCell = document.createElement('td');
80       descCell.textContent = file.FileDescription;
81       row.appendChild(descCell);
82
83       const actionCell = document.createElement('td');
84       const viewButton = document.createElement('button');
85       viewButton.textContent = 'View';
86       viewButton.onclick = () => {
87         window.open(file.FilePath, '_blank');
88       };
89       actionCell.appendChild(viewButton);
90       row.appendChild(actionCell);
91
92       tableBody.appendChild(row);
93     });
94   } catch (error) {
95     console.error('Error fetching files:', error);
96   }
97 }
98
99 // Fetch files on page load
100 fetchFiles();
101 </script>
102 </body>
103 </html>
104
```

```
public > # styles.css > button
1 /* General Styles */
2 body {
3   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
4   background-color: #f4f7fa;
5   color: #333;
6   margin: 0;
7   padding: 0;
8 }
9
10 /* Header Styles */
11 h1 {
12   text-align: center;
13   color: #4CAF50;
14   margin-top: 30px;
15   font-size: 36px;
16 }
17
18 /* Form Styles */
19 form {
20   max-width: 600px;
21   margin: 0 auto;
22   background-color: white;
23   padding: 20px;
24   border-radius: 8px;
25   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
26 }
27
28 form label {
29   font-size: 16px;
30   color: #555;
31 }
32
33 form input[type="text"],
34 form input[type="file"] {
35   width: 100%;
36   padding: 10px;
37   margin: 10px 0;
38   border-radius: 5px;
39   border: 1px solid #ddd;
40
```

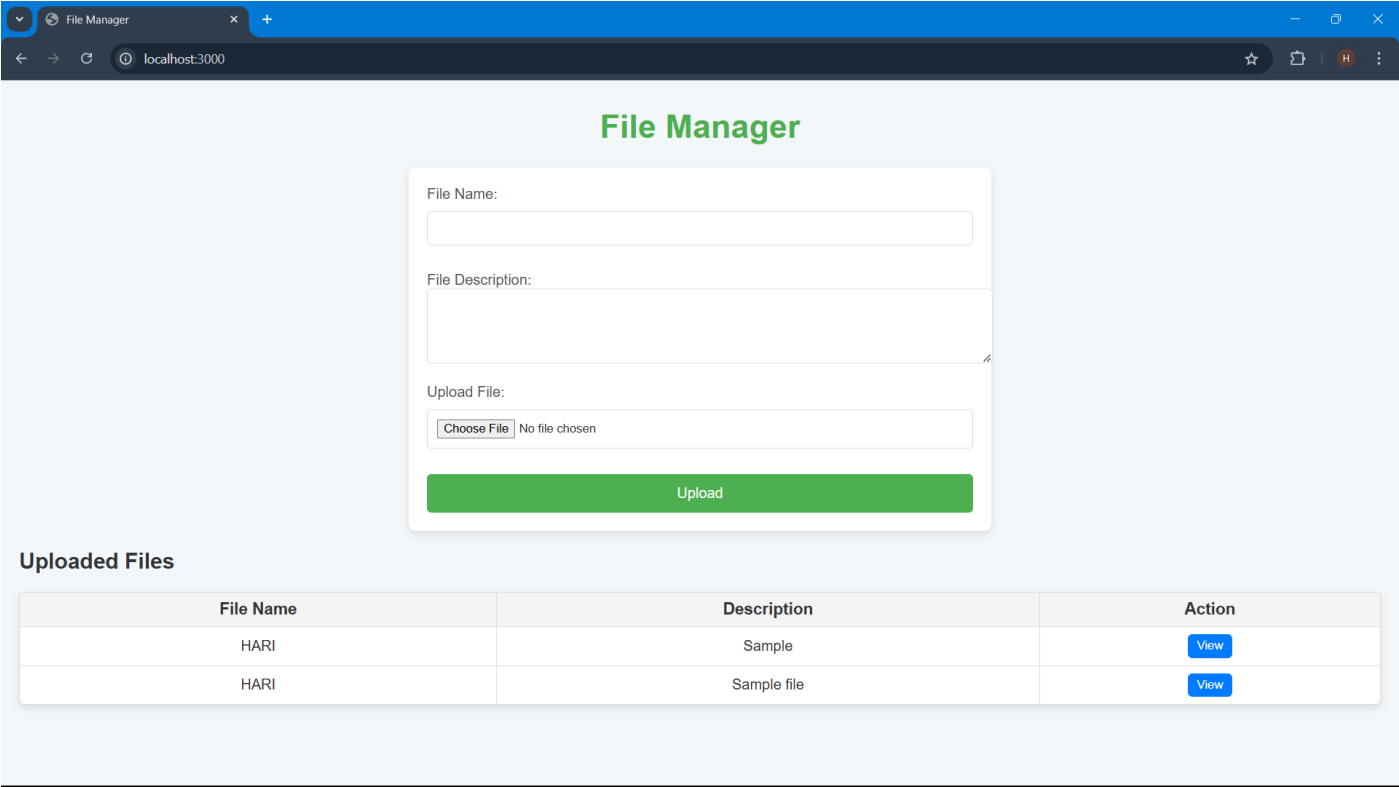


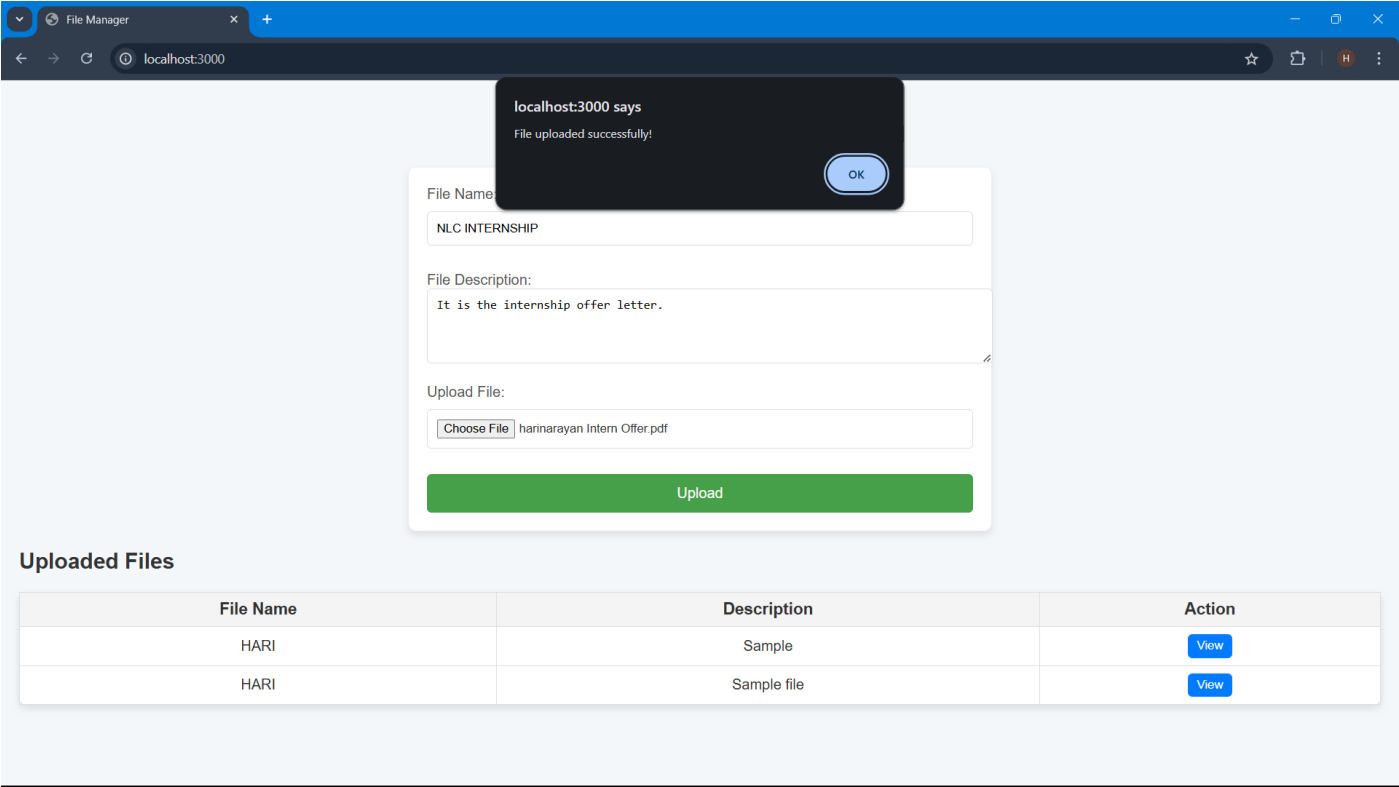
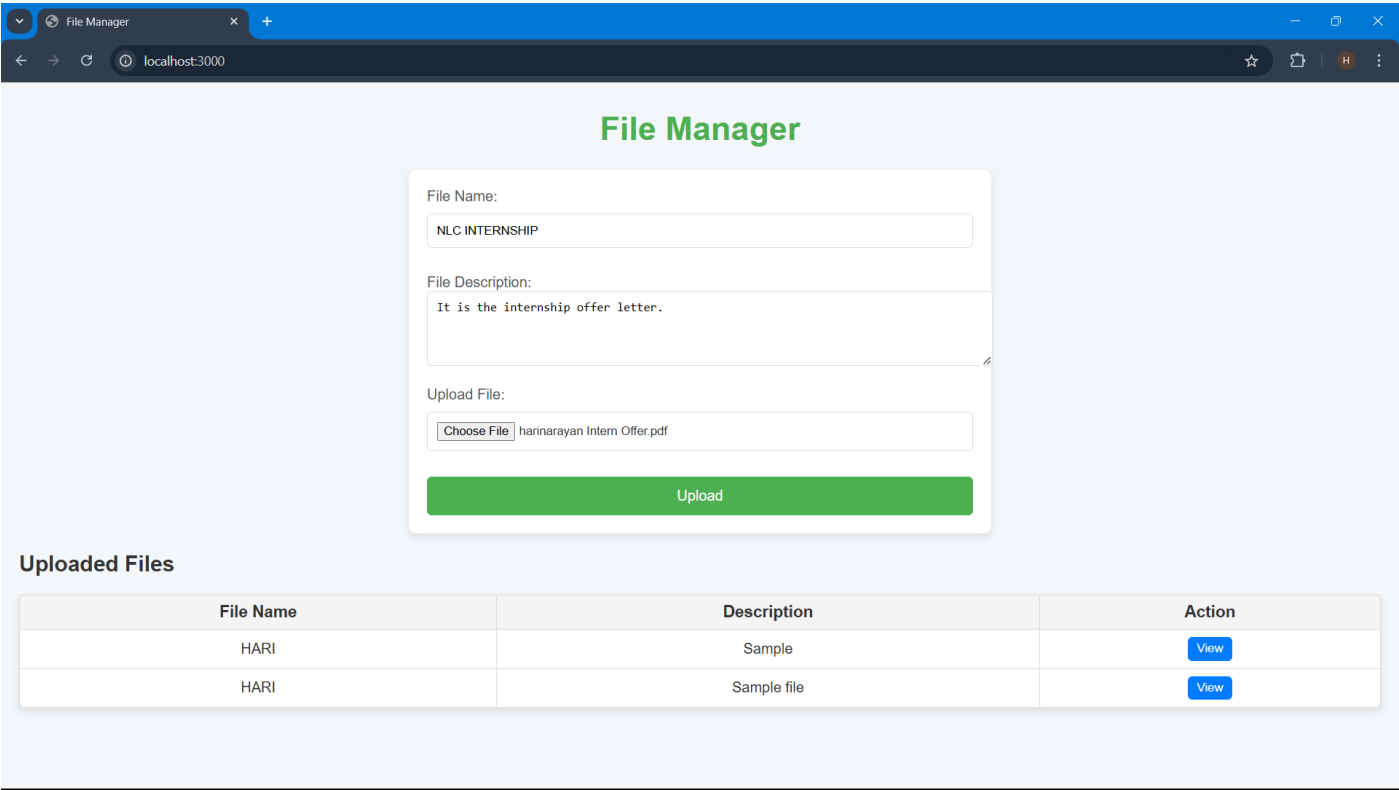
```
1  const express = require('express');
2  const multer = require('multer');
3  const xlsx = require('xlsx');
4  const path = require('path');
5  const fs = require('fs');
6
7  const app = express();
8  const PORT = 3000;
9
10 // Configure multer to preserve original file extensions
11 const storage = multer.diskStorage({
12   destination: function (req, file, cb) {
13     cb(null, 'uploads/');
14   },
15   filename: function (req, file, cb) {
16     // Optionally, prefix the filename with a timestamp to prevent name collisions
17     cb(null, Date.now() + '-' + file.originalname);
18   },
19 });
20
21 const upload = multer({ storage: storage });
22
23 // Middleware for serving static files and parsing JSON
24 app.use(express.static('public'));
25 // Serve the 'uploads' directory as a static folder
26 app.use('/uploads', express.static(path.join(__dirname, 'uploads')));
27 app.use(express.json());
28
29 // Excel file path
30 const EXCEL_FILE = 'files_data.xlsx';
31
32 // Ensure the Excel file exists
33 if (fs.existsSync(EXCEL_FILE)) {
34   const workbook = xlsx.utils.book_new();
35   const worksheet = xlsx.utils.json_to_sheet([]);
36   xlsx.utils.book_append_sheet(workbook, worksheet, 'Files');
37   xlsx.writeFile(workbook, EXCEL_FILE);
38 }
39
40 // Endpoint to upload file
```

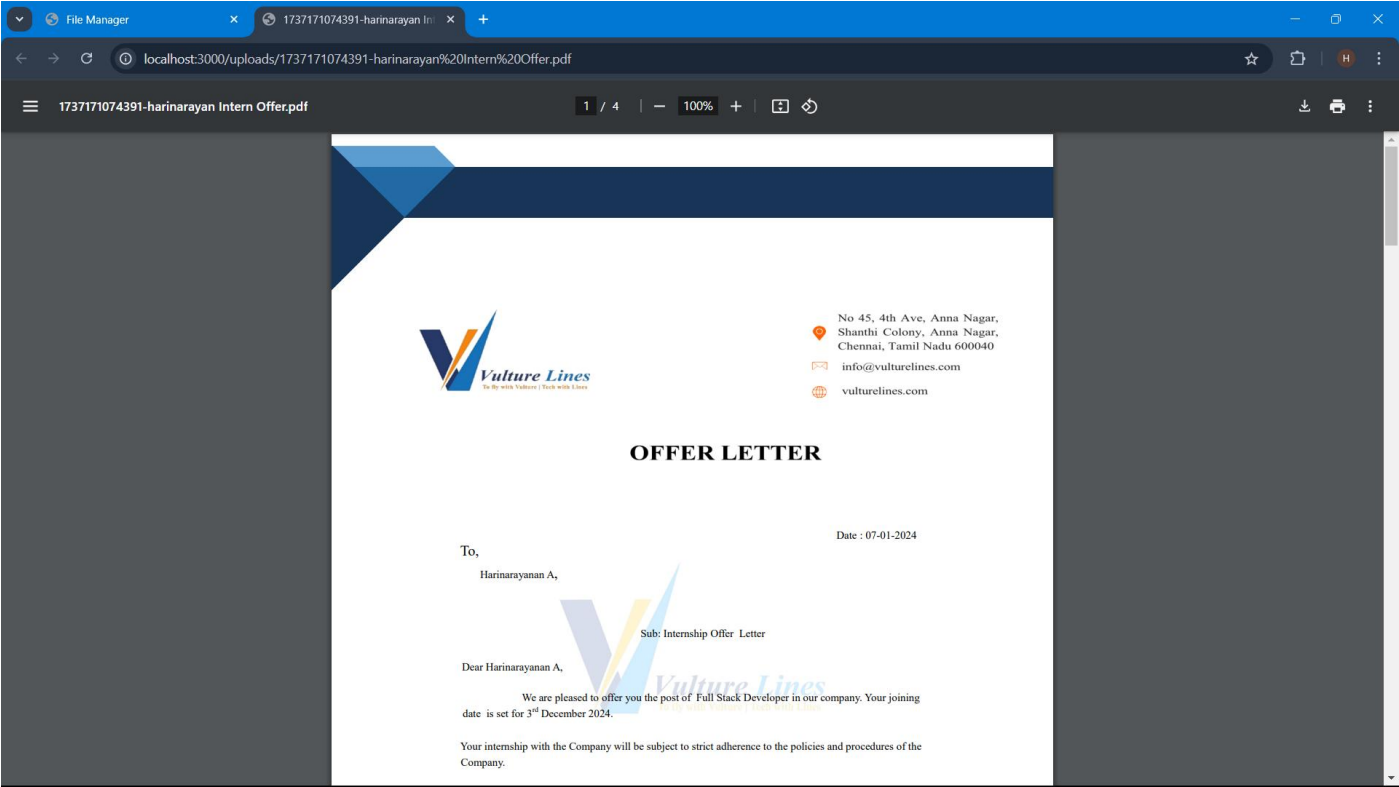
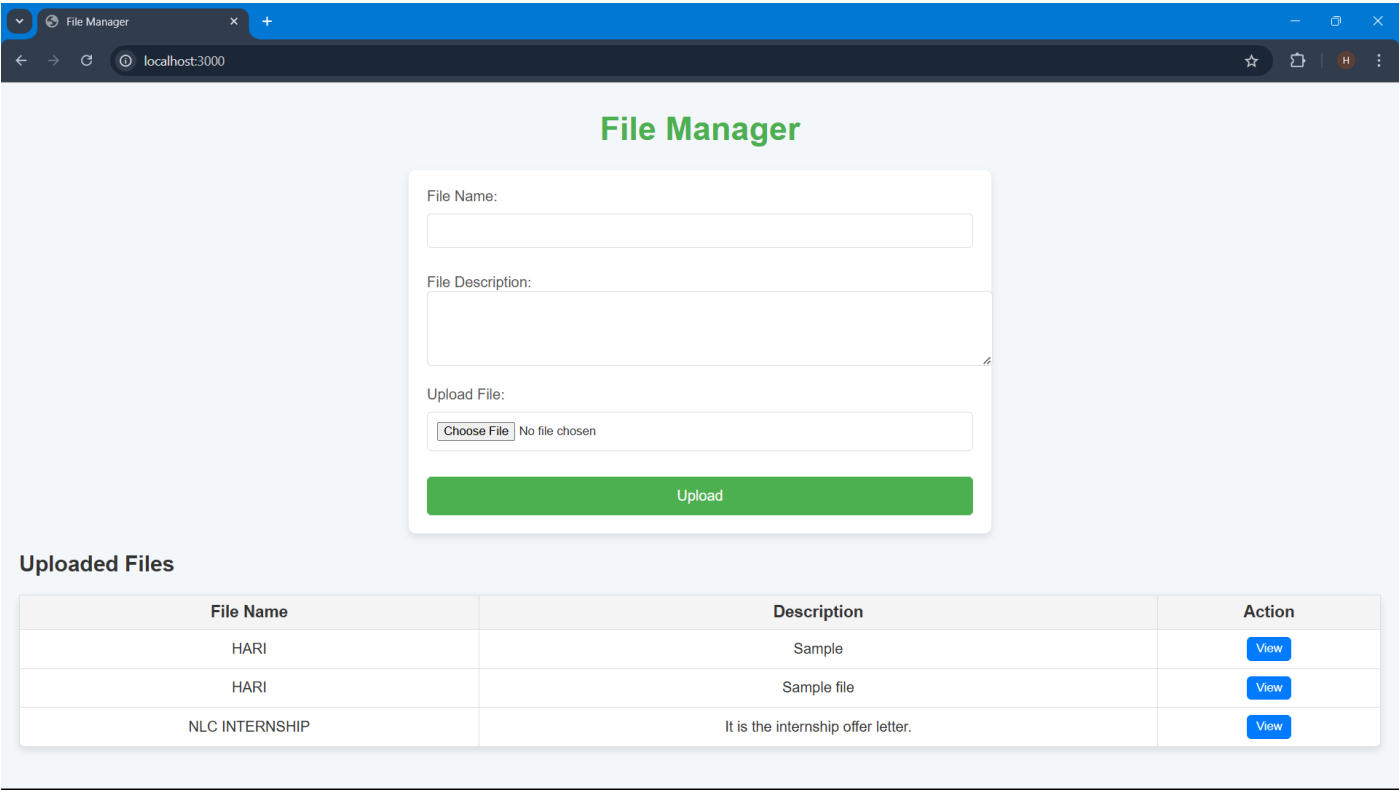
```
40 // Endpoint to upload file
41 app.post('/upload', upload.single('file'), (req, res) => {
42   const { fileName, fileDescription } = req.body;
43
44   // Construct the file path with the '/uploads/' prefix
45   const filePath = '/uploads/${req.file.filename}';
46
47   // Read the existing Excel file
48   const workbook = xlsx.readFile(EXCEL_FILE);
49   const worksheet = workbook.Sheets['Files'];
50
51   // Append new data
52   const existingData = xlsx.utils.sheet_to_json(worksheet);
53   const newData = {
54     FileName: fileName,
55     FileDescription: fileDescription,
56     FilePath: filePath,
57   };
58   existingData.push(newData);
59
60   // Write back to the Excel file
61   const updatedWorksheet = xlsx.utils.json_to_sheet(existingData);
62   workbook.Sheets['Files'] = updatedWorksheet;
63   xlsx.writeFile(workbook, EXCEL_FILE);
64
65   res.json({ message: 'File uploaded successfully!' });
66 });
67
68 // Endpoint to fetch file data
69 app.get('/files', (req, res) => {
70   const workbook = xlsx.readFile(EXCEL_FILE);
71   const worksheet = workbook.Sheets['Files'];
72   const data = xlsx.utils.sheet_to_json(worksheet);
73   res.json(data);
74 });
75
76 // Remove or comment out the '/view/:filePath' route
77 // app.get('/view/:filePath', (req, res) => {
78 //   const filePath = req.params.filePath;
```

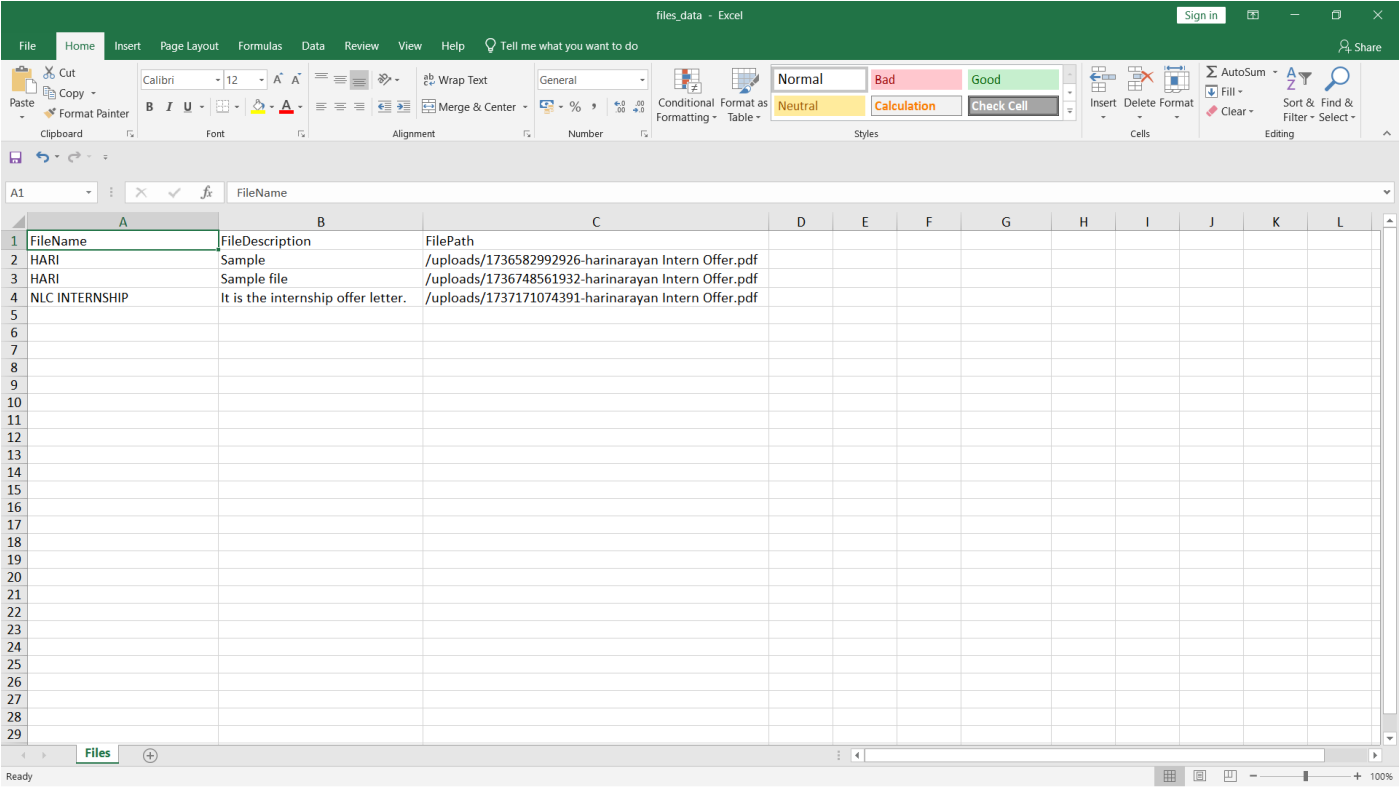
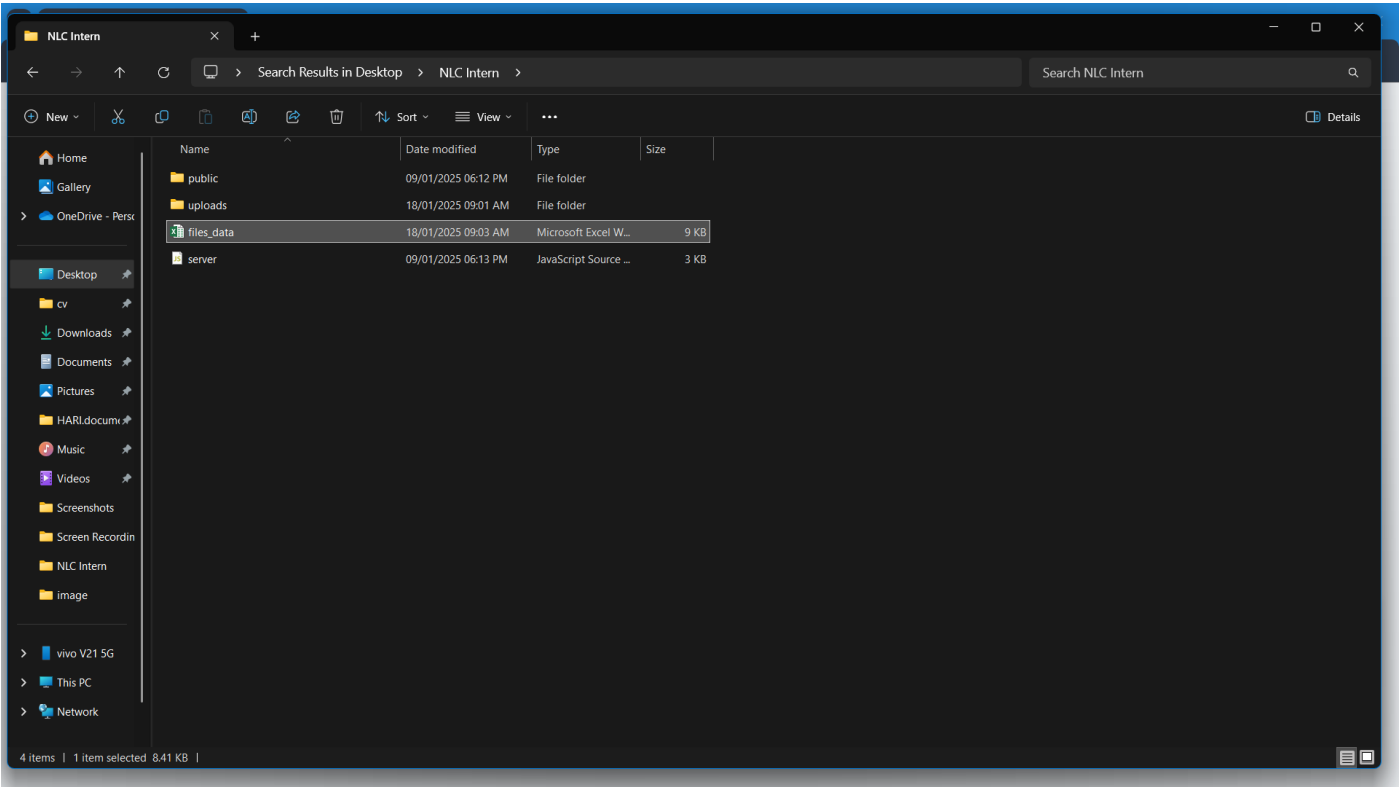


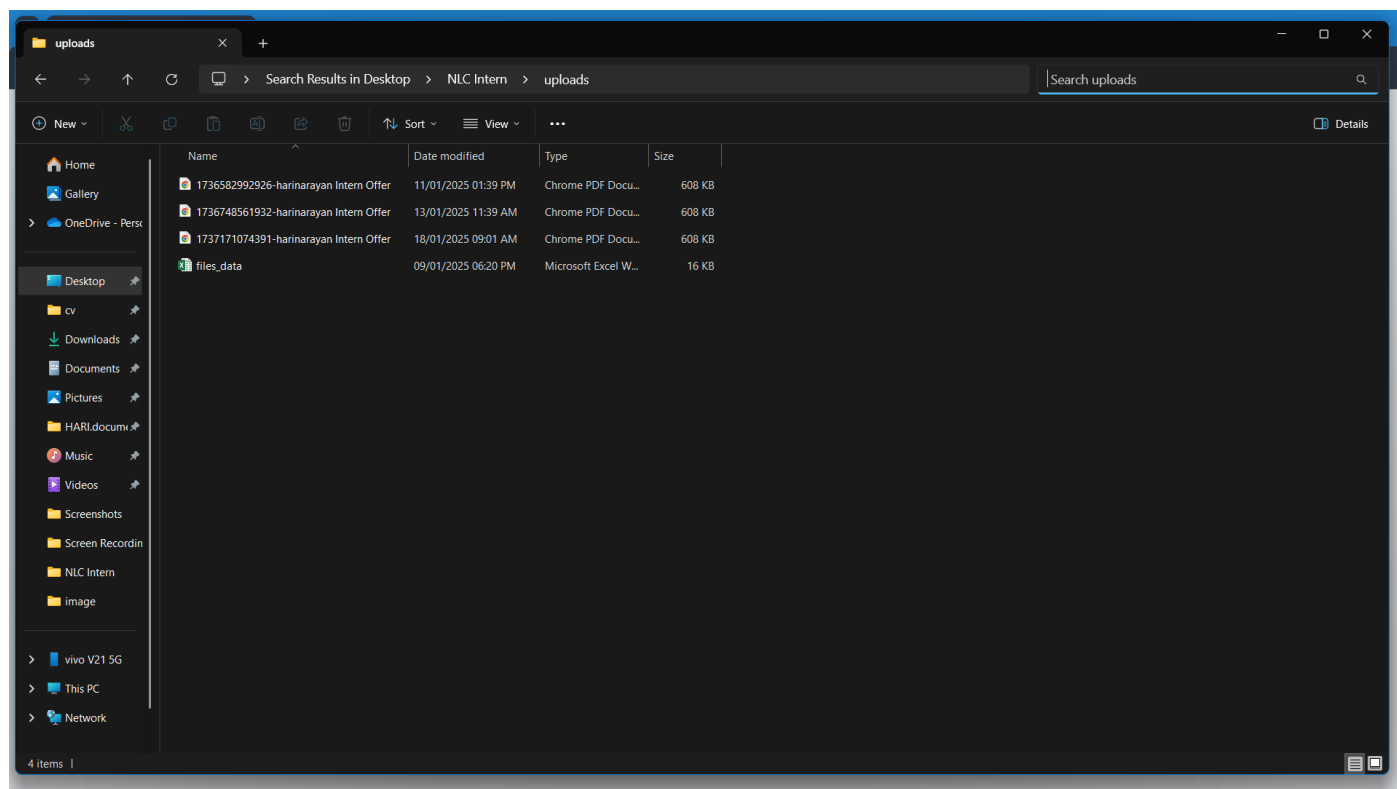
OUTPUT











The output of this project is a fully functional **File Management System** that provides an efficient and user-friendly platform for uploading, organizing, and accessing files. Users can input a file name, provide a detailed description, and upload files through a simple web interface. The uploaded files, along with their metadata, are securely stored in an Excel file for easy reference and future retrieval. Once the files are uploaded, users can view a list of all uploaded files in a tabular format displayed on the web application. The table includes the file name, description, and a "View" button for each entry, which allows users to open and access the file directly in a new tab.

The system is designed to handle various file types and ensures that all uploaded files are stored in an organized manner within a dedicated folder. Additionally, the backend keeps track of all file paths and metadata, ensuring seamless integration between storage and display. This project simplifies file management by allowing users to interact with files in a structured way without directly dealing with complex file systems. It provides a responsive and interactive interface, making it suitable for real-world applications like document management systems in small-scale organizations or personal use cases. Overall, the project demonstrates the effective use of modern web technologies for developing practical and intuitive solutions.

8. CONCLUSION

This project marks a significant achievement in developing a robust and practical **File Management System** that seamlessly integrates user-friendly design with efficient backend functionality. By leveraging technologies like HTML, CSS, JavaScript, Node.js, and Express.js, the project provides a comprehensive solution for file uploads, metadata management, and easy access to files. The responsive design ensures that users can interact with the application across different devices, enhancing accessibility and usability. The system effectively manages uploaded files, stores metadata securely in an Excel file, and displays the data dynamically in a tabular format with convenient options for accessing files.

The implementation of middleware like Multer demonstrates the ability to handle file uploads securely, while the use of the `xlsx` library highlights skills in data manipulation and storage. This project reflects a deep understanding of full-stack development concepts, including server-client architecture and the HTTP request-response cycle. It also showcases the ability to design and implement practical solutions to real-world problems in file management.

Key Takeaways from this Internship

Technical Proficiency: Enhanced understanding of web development concepts and tools, including proficiency in both frontend technologies (HTML, CSS, JavaScript) and backend frameworks (Node.js, Express.js). Improved skills in data handling using Excel and libraries like `xlsx`.

Problem-Solving Skills: the ability to design and develop a fully functional web application tailored to specific needs, overcoming challenges such as file handling, secure data storage, and responsive design.

Project Management: Gained experience in planning, organizing, and executing a complete project, including the integration of multiple components like servers, APIs, and frontend interfaces to deliver a seamless user experience.

Independent and Collaborative Work: Developed the ability to work independently on technical implementations while collaborating effectively with peers and mentors for feedback and guidance throughout the project lifecycle.

9. LEARNINGS

Fundamentals of Web Technologies:

HTML/CSS: Mastery of HTML and CSS is crucial for creating well-structured and styled web pages. Understanding semantic HTML enhances accessibility and SEO, while CSS frameworks like Bootstrap or Tailwind can streamline responsive design.

JavaScript: Gaining proficiency in JavaScript enables you to add interactivity and dynamic functionality to websites. Learning about modern frameworks and libraries, such as React, Angular, or Vue.js, can further enhance your development capabilities.

Responsive Design Principles:

Mobile-First Approach: Designing with a mobile-first mindset ensures that your site performs well on smaller screens and scales up effectively. This approach often leads to better overall design and user experience.

Media Queries: Implementing media queries allows you to create adaptable layouts that respond to different screen sizes and orientations, ensuring a consistent experience across devices.

Version Control Systems:

Git: Using Git for version control helps manage code changes, collaborate with others, and maintain a history of project development. Platforms like GitHub or GitLab provide additional tools for code review and collaboration.

Security Best Practices:

Data Protection: Implementing HTTPS, data encryption, and secure authentication methods (e.g., OAuth) helps protect user information and maintain site integrity.

Continuous Learning and Adaptability:

Keeping Up-to-Date: The web development field is dynamic, with frequent updates and new technologies emerging. Staying informed about the latest trends, tools, and best practices is essential for ongoing success.

Experimentation: Trying out new frameworks, libraries, and techniques can lead to valuable insights and improvements in your development practices.