

## Notes Sharing App

A lightweight, **frontend-only** notes sharing platform designed for students to crowdsource and access study materials seamlessly. By leveraging the Google Workspace ecosystem (Forms, Sheets, and Drive), this application functions as a full-stack experience without the need for a dedicated backend or database server.

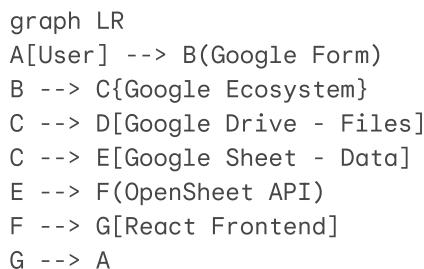
### Features

-  **Serverless Uploads:** Users contribute notes via a customized Google Form, keeping the frontend simple and secure.
-  **Dynamic Gallery:** Notes are instantly fetched and displayed in a clean, searchable grid interface built with React.
-  **Direct Downloads:** Optimized links allow users to download PDF/Image notes directly to their devices without leaving the app.
-  **Cloud Storage:** All files are stored securely in Google Drive with metadata managed in Google Sheets.
-  **Modern UI:** Built with React and Tailwind CSS for a responsive, mobile-first experience.

### Architecture

The app follows a "No-Backend" architecture:

1. **Submission:** User uploads a file through a **Google Form**.
2. **Storage:** The file is automatically saved to a **Google Drive** folder, and a response entry is created in a linked **Google Sheet**.
3. **Bridge:** The **OpenSheet API** converts the public Google Sheet into a JSON endpoint.
4. **Frontend:** The **React App** fetches the JSON, parses the data, and renders the UI.



### Tech Stack

- **Frontend:** React.js (Vite)
- **Styling:** Tailwind CSS / Lucide React (Icons)
- **Database:** Google Sheets

- **File Storage:** Google Drive
- **Data Ingestion:** Google Forms
- **API:** OpenSheet API

## Setup & Installation

### 1. Google Workspace Configuration

To act as your database, the Google environment must be configured correctly:

#### 1. Google Form:

- Create a form with fields: **Subject**, **Topic**, **Author**, and **File Upload**.
- In the **Responses** tab, click "Link to Sheets".

#### 2. Google Sheet:

- Open the linked sheet. Click **Share** -> Change access to "**Anyone with the link can view**".
- Note the **Sheet ID** from the URL:  
`docs.google.com/spreadsheets/d/[SHEET_ID]/edit`.

#### 3. Google Drive Folder:

- Find the folder where the form saves uploads.
- Set folder sharing to "**Anyone with the link can view**". *Crucial: If the folder isn't public, the download links will fail for other users.*

### 2. Frontend Setup

```
# Clone the repository
git clone [https://github.com/yourusername/notes-sharing-app.git](https://github.com/yourusername/notes-sharing-app.git)

# Install dependencies
npm install

# Run the development server
npm run dev
```

### 3. Connecting the Data

In your `App.jsx` or data-fetching hook, update the API endpoint:

```
// Format: [https://opensheet.elk.sh/](https://opensheet.elk.sh/)[SHEET_ID]/[TAB_]
const API_URL = "[https://opensheet.elk.sh/YOUR_SHEET_ID_HERE/FormResponses1](htt
```

## Implementation Notes

## Handling Direct Downloads

Google Drive provides "view" links by default. To enable direct downloads, the app replaces the URL string:

- **Original:** `drive.google.com/file/d/FILE_ID/view?usp=drivesdk`
- **Modified:** `drive.google.com/uc?export=download&id=FILE_ID`