Getting Started with Documnet Management

This project was bootstrapped with Create React App.

Instructions how to run & test the application

- In the directory of `./seal` run:
  1. Run `npm install` to install all dependencies. (Note: Node.js version 14 or higher is required)
  2. Run `npm start` to start the front-end.
  3. Run `node server.js` to start the server.
  4. The uploaded files will be saved in the `./uploads` directory.
  5. To see the list of uploaded documents, click on the "Upload" button.
  6. To download a document, first copy its ID from the document list, and paste it into the input text-box of the download section.
  7. To set an expiration time for share-link generation, enter the desired number of hours in the "Expiration Duration" box. If you choose 0 hours, it will expire immediately.

Tech Stack & Used Packages

- Tech Stack:
  - Node.js
  - Express.js
  - React.js
  - SQL
- Used packages:
  - `npm install`
  - `npm install axios`
  - `npm install multer`
  - `npm install cors`
  - `npm install uuid`
  - `npm install express sqlite3`
  - `npm install react-router-dom`

Ideas and proposals to improve the application from a user or technical perspective

- Deploying the Express server to a Heroku platform to make it publicly accessible.
- Split the database table into two separate tables, one containing the general information of the uploaded files and the other one keeping track of the share links' information.
- Implement multiple file upload/download functionality.
- Set a preview image of the first page. Currently, there are technical problems preventing the implementation of this feature. Future attempts can be made using server-side or front-end approaches.

  - For the server-side attempt, "pdf2pic" was used to save the first image in a specific directory and later utilize it in the front-end for preview demonstration.
  - For the front-end attempt, { Document, Page, pdfjs } from 'react-pdf' were used to directly demonstrate the preview images in the front-end.