CS4530 Final Project: “Document Sharing” “Group 704”: Daniel, Joseph, Aminah, Ise

Our Feature : Covey Docs

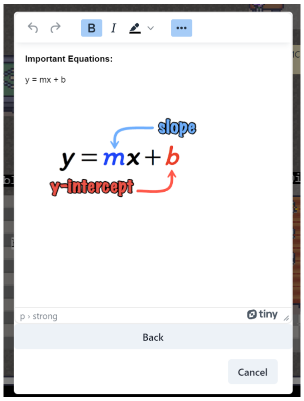
The problem that we have chosen to address for our project is that users in Covey.Town can’t cooperatively work on documents together. For this reason, we plan to implement a built-in collaboration platform inCovey.Town that allows users to work on documents together in real time.

Our implementation allows users to create personalized accounts tracking their documents, create new documents, invite people as collaborators under various permission levels, and edit documents together with document changes reflected for all users.

The documents can contain images and text than can be moved around and resized.

Demo and Source:

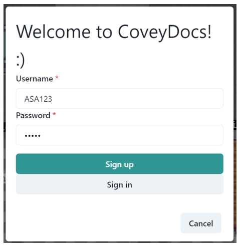
Our demo site can be found at https://coveydocsui.onrender.com/, and our code can be found at https://github.com/neu-cs4530/fall23-team-project-group-704.



Important Equations:

y = mx + b

Documents can contain text and images. The images can be moved around, and such changes are reflected for other users every 5 seconds.



Welcome to CoveyDocs! :)

Username *

ASA123

Password *

••••••

A login page for our app.

Our Technology Stack & Design

We utilized the existing covey.town code base to implement our document sharing functionality on. We used React and Chakra UI to visualize our application. We also used the library TinyMCE to create the document component and allow the picture uploading functionality on the document.

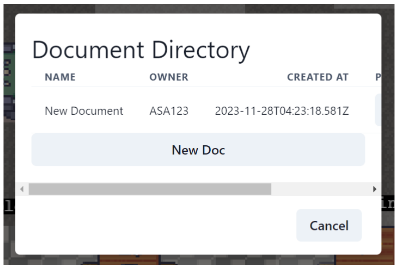
This library enables us to store all document content as raw HTML. When a user goes to our document area in covey.town they can access any documents they have permissions to either edit or view as well as create new documents and give other users permission to edit or view their documents. Users are also able to login into our application and sign up if they don't have an account yet or want to create a new account.
We used socket-io to sync our controller with the backend, to query data directly from the backend, or to receive events from the backend when documents get edited, etc.

For our backend, we utilized PostgreSQL database to store information regarding users, documents, and permissions. We had three tables, one for each of the objects we are tracking: permissions, document, and users. Our continuous integration pipeline automatically executes our tests and style checks. Our site is deployed using Heroku.

Future Work

Some features we did not include in our user stories were the ability to rename documents, transfer ownership, or delete documents. These quality of life features could be future work.

We could also use a more sophisticated way to sync edits in real-time for multiple users. We synced documents through pushing updates every few seconds from each user’s frontend. Perhaps a differential approach would create a more streamlined experience instead of overwriting the entire document every few seconds.



Document Directory

NAME	OWNER	CREATED AT	P
New Document	ASA123	2023-11-28T04:23:18.581Z	

A display for documents owned and shared with a user.