Problem Statement and Market Considerations

Nowadays, with the growing emphasis on security and privacy, decentralization is becoming increasingly popular. Our project aims to provide a file storage system, but in a decentralized manner. This means that the actual files are stored on a decentralized, node-to-node network, rather than in centralized storage. We plan to utilize some existing Web3 APIs in this project.

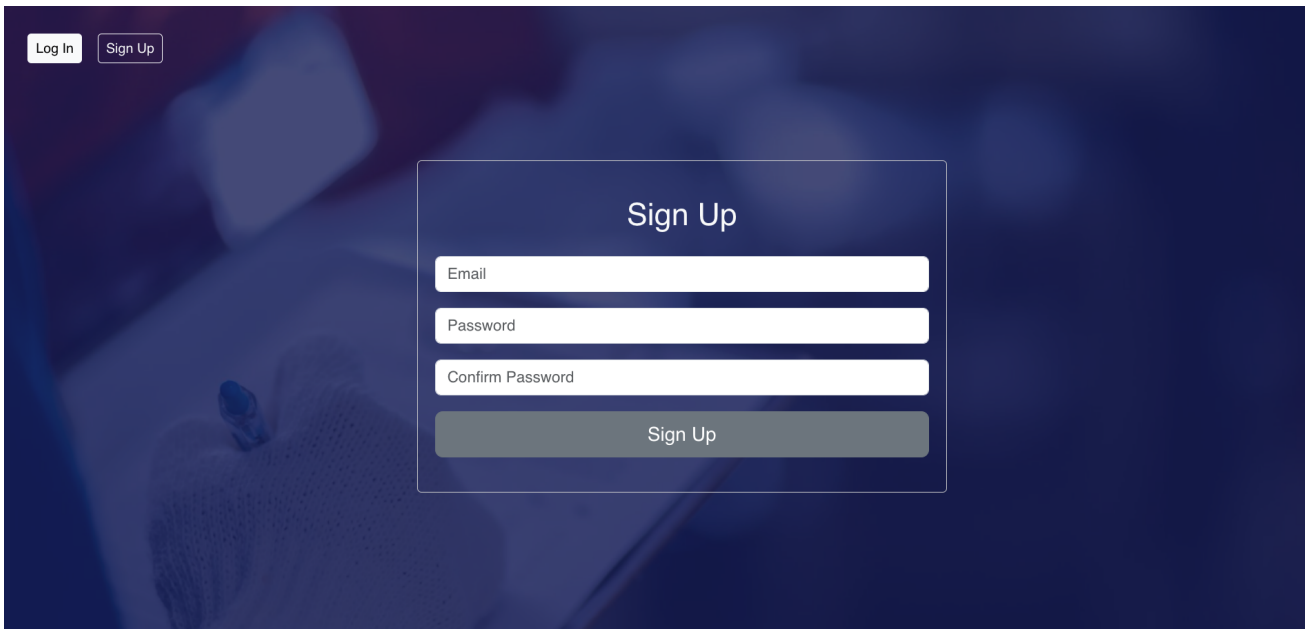
Another motivation for this project stems from the current landscape of decentralized file sharing. While IPFS, a decentralized protocol and network, is available for sharing data and files, and there are APIs that can communicate with IPFS, Web3 remains a new and controversial topic. Most of these services require coding skills, creating a scenario reminiscent of the early days of computers, where most users were developers. We believe that decentralized data storage is the future, as it offers greater security and better protection of user privacy. Therefore, making these APIs and the complex knowledge about Web3 more user-friendly is another key motivation.

This approach also constitutes our key competitive advantage when compared with other products in the market. Currently, most products offer only APIs, which, while flexible, are difficult for non-technical users to understand and apply. We are committed to the principle of decentralization and aim to make it accessible to users across both technical and non-technical domains.

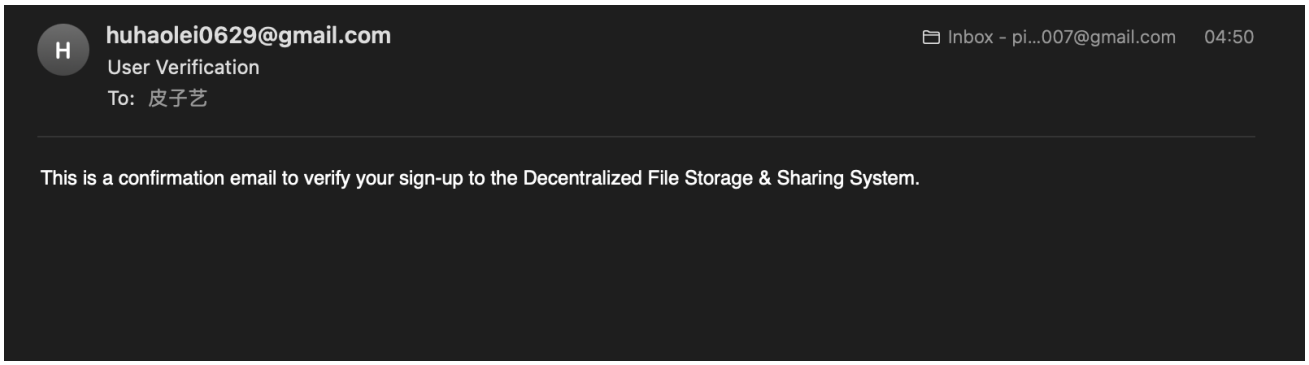
User Flow

- Landing page to log in or sign up

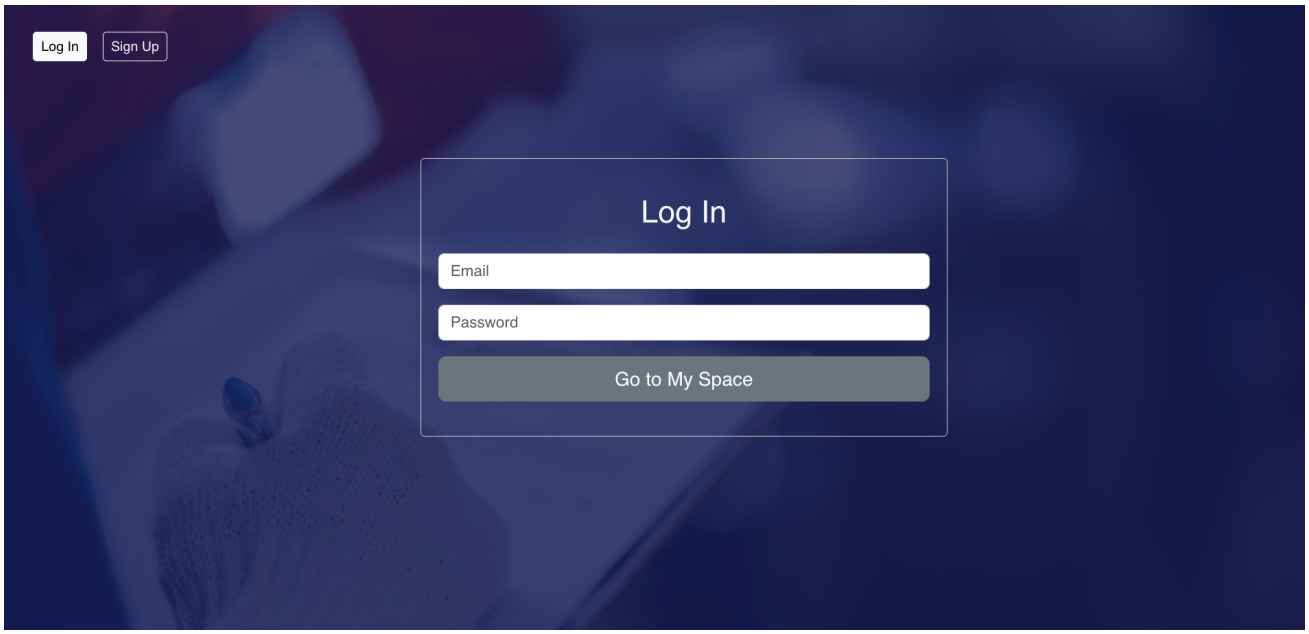
- After clicking “Sign Up”:



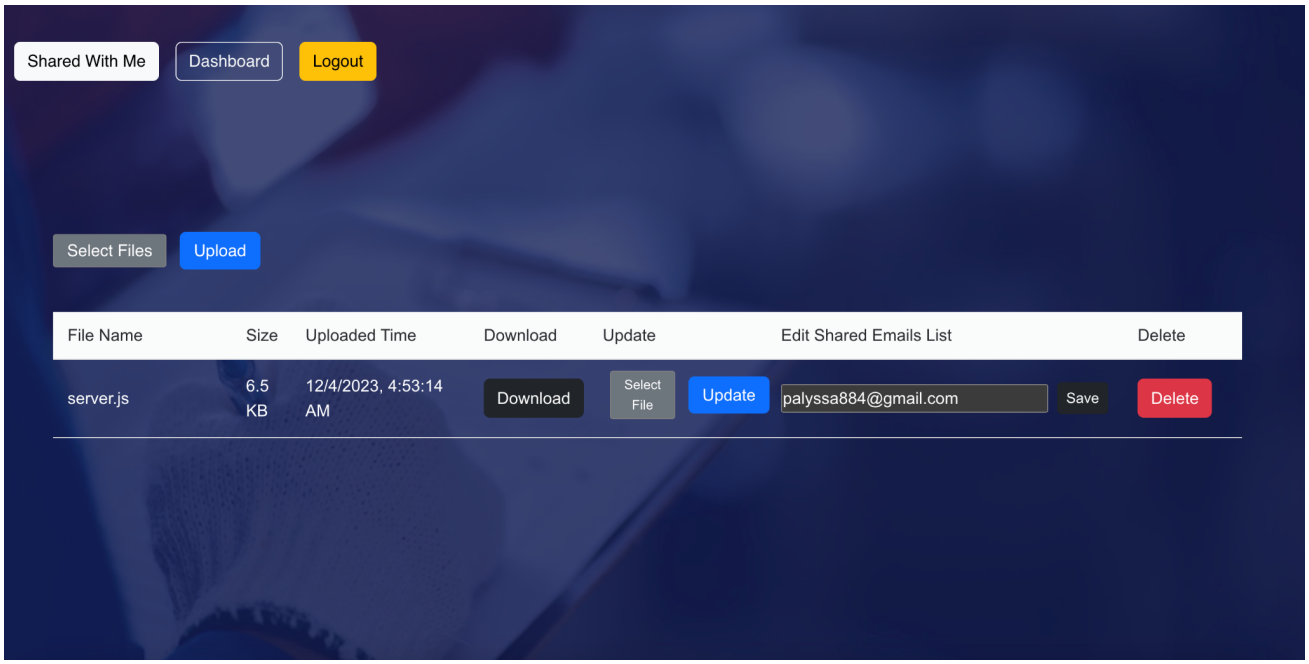
- Able to receive an email indicating the sign up is successful



- Or log in directly:

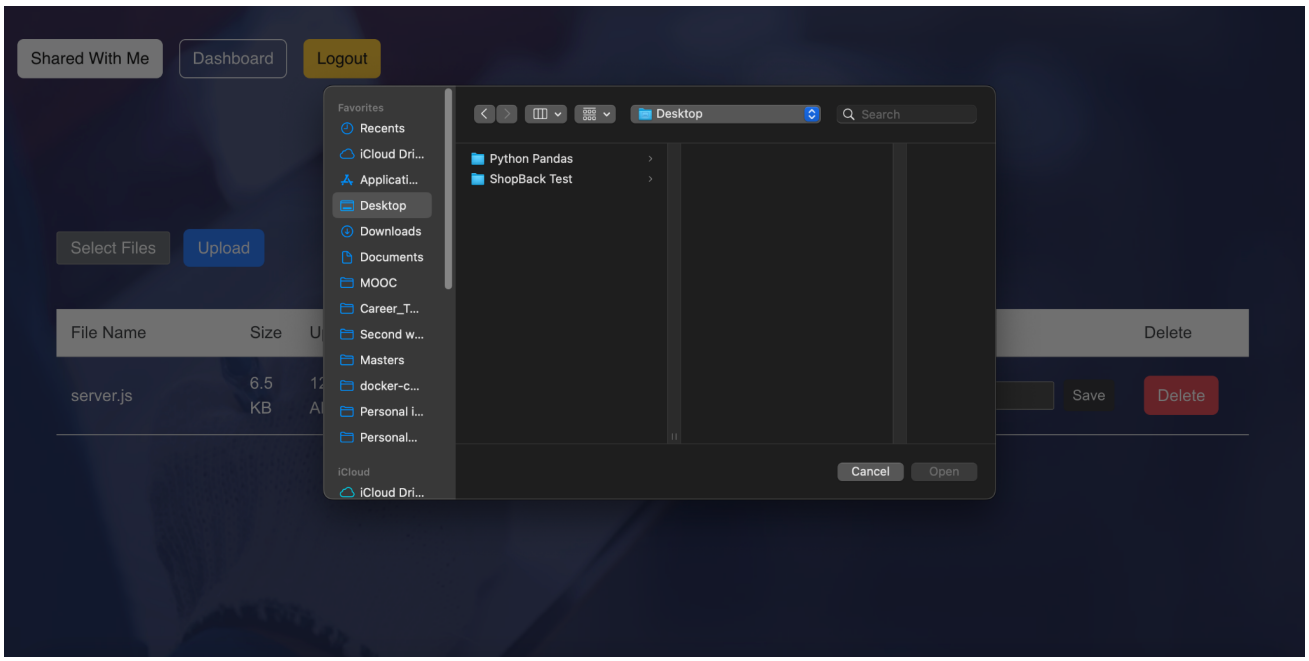


- After logging in:

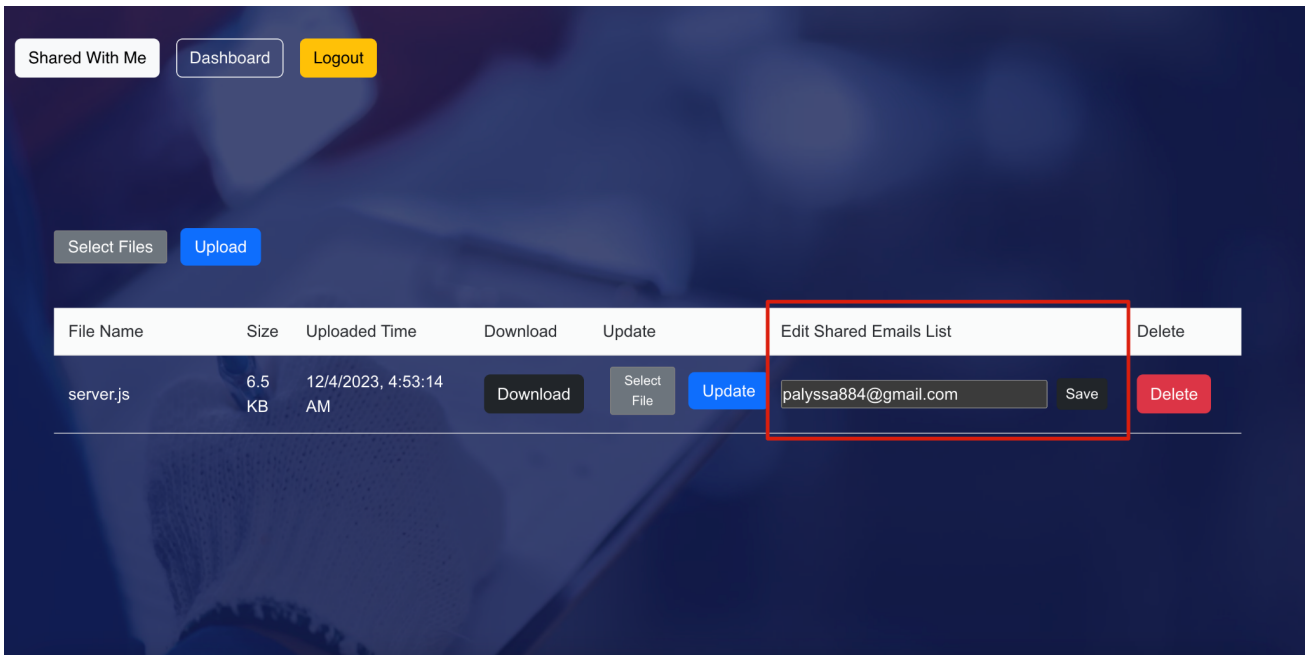


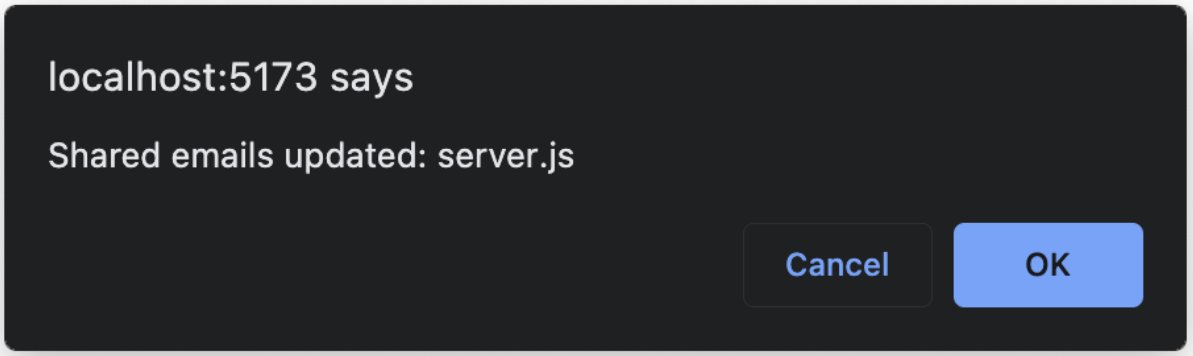
  - User is able to maintain his/her logged in session with sessionStorage, unless she/her logged out

- Clicking on “Select Files”

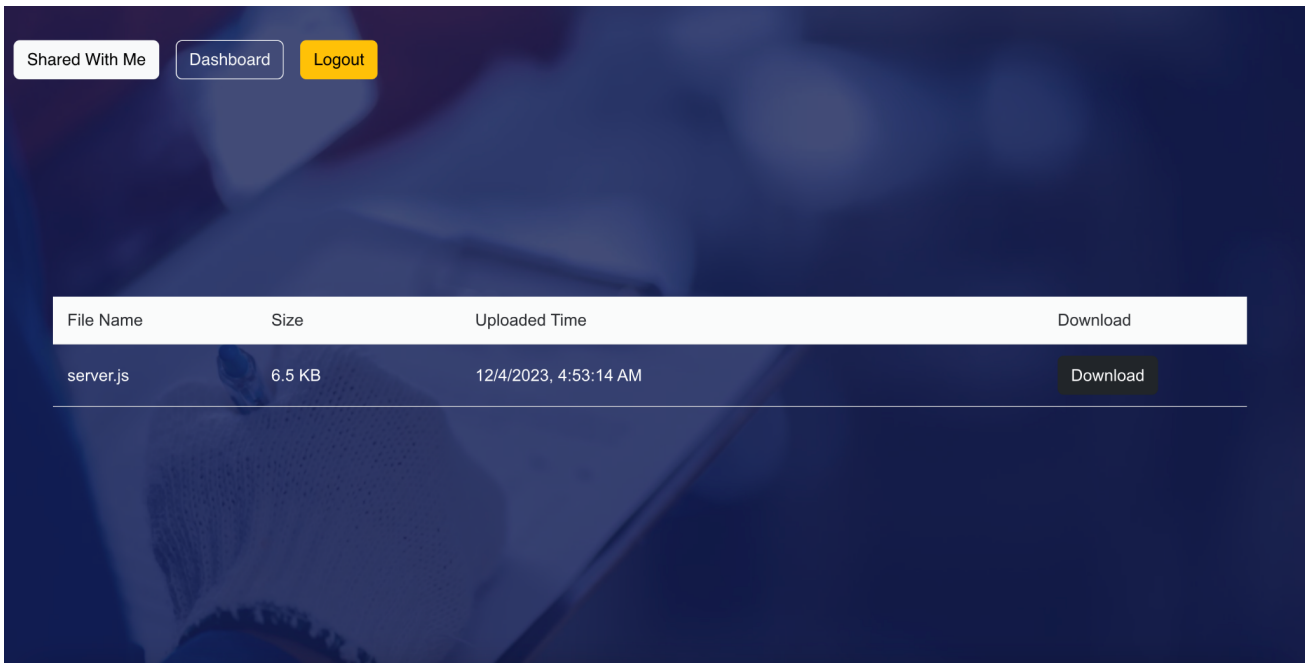


- Clicking on “Edit” and “Save” to edit shared access of a file:





- View / download files shared with me:



Technical Challenges and Optimizations

The key challenge is that, due to the dynamic nature of Web3, the HTTP service is always volatile. Previously, we had a mature service, but now it's deprecated, and we need to use the new Helia, which was just released this year. Therefore, there are fewer use cases to refer to. After reviewing several GitHub repositories, we identified two essential providers for uploading and retrieving data. In storage, the actual files are only on IPFS; we store only hashes in the centralized MongoDB storage.

Another challenge is maintaining user sessions in a stateless HTTP environment. Initially, we considered using cookies and sessions, but due to security concerns and scalability issues, we opted for token-based authentication, which is encrypted and therefore more secure. The process involves setting the token as one of the HTTP request headers, which the server then decrypts upon receipt.

For optimization, we implemented a bi-directional database to efficiently retrieve users from files and vice versa. We also enabled updating files even when they are shared, similar to how Google Drive folders work: when you update the folder content, previous users with access can see the changes. We are using two servers, Apollo and Vite, for backend and frontend respectively. MongoDB is used to store user information and file hashes. In terms of API, we are referencing HeliaProvider.

References and Open Source

> GitHub Repository for Helia APIs:
> GitHub - ipfs-examples/helia-examples: Main

We could make the codebase open source by sharing it on GitHub. To safeguard copyright, selecting

an appropriate license, such as the MIT License, would be advisable. This license allows others to utilize the code for almost any purpose, including commercial use. Alternatively, the Apache License is also a suitable choice. Compared to the MIT License, it offers the added benefit of granting patent rights.

In addition to selecting a license, if the project transitions to a commercial phase or undergoes expansion, incorporating a contributor license agreement or trademark might be prudent. These measures can provide excellent ways for contributors to protect their rights.