

FileBox

...

Isabell Deak, Will Abrams, Jessie Ryce, Artem Nekrasov, Ford Mulligan

About

PURPOSE:

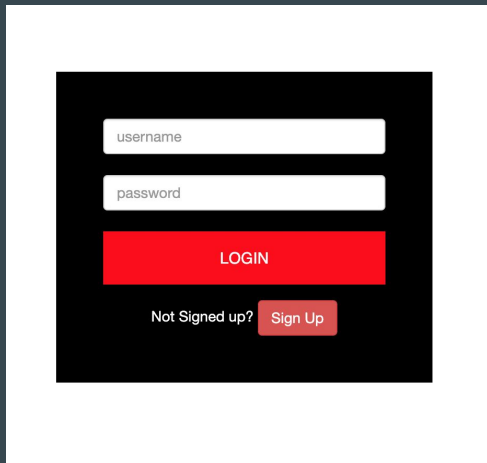
To offer users a quick, lightweight file hosting service that is straightforward to use.



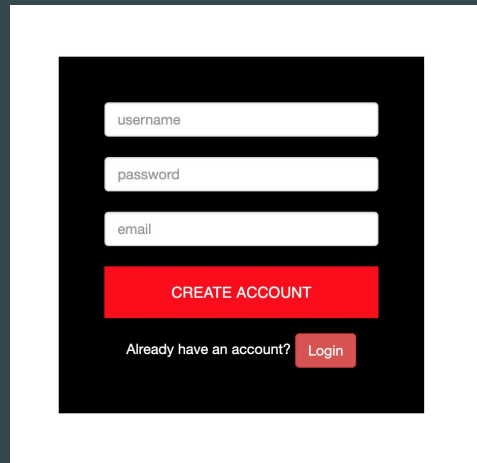
1. Features

Log In / Sign up

Whether you are a new user or existing user, you can login to your account or create a new account. This is the start to a great FileBox.



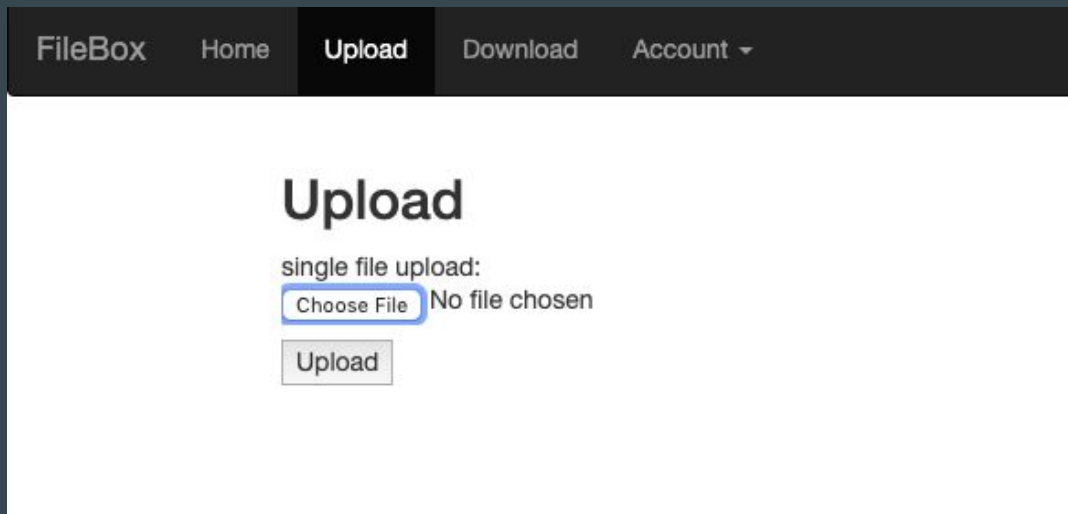
A login form mockup with a black background. It features two white input fields labeled 'username' and 'password'. Below these is a large red button with the text 'LOGIN'. At the bottom, there is a link 'Not Signed up?' followed by a red button labeled 'Sign Up'.



A sign up form mockup with a black background. It features three white input fields labeled 'username', 'password', and 'email'. Below these is a large red button with the text 'CREATE ACCOUNT'. At the bottom, there is a link 'Already have an account?' followed by a red button labeled 'Login'.

Upload Files

The upload feature allows users to choose a file off their computer and upload it to FileBox



The screenshot shows the 'Upload' page of the FileBox application. At the top is a dark navigation bar with links for 'FileBox', 'Home', 'Upload' (which is highlighted), 'Download', and 'Account' with a dropdown arrow. The main content area is white and features the heading 'Upload'. Below this, it says 'single file upload:'. There is a blue-outlined button labeled 'Choose File' and the text 'No file chosen' next to it. Below these elements is a grey 'Upload' button.

Download Files

The download feature allows users to download files from FileBox onto their computer.

[FileBox](#) [Home](#) [Upload](#) [Download](#) [Account ▾](#)

Download

File Visibility:

☒ Public
☐ Private

PUBLIC FILE

Public ID

2. Design

Methodologies / Tools

- Methodology: Continuous Integration
 - Rank: 5
- Project Tracker: Trello
 - Rank: 5
- VCS: Github
 - Rank: 4
- Multiple IDE: Atom, Sublime, VS CODE
 - Rank: 5
- Database: MySQL
- Deployment Environment: AWS, tested in LocalHost
- Framework: Node.js

Upload Function

- Purpose:
 - allow users to upload files to the server - these are kept track of in a files table in our database
- Assumptions:
 - small file sizes (files over ~2G don't upload well with POST)
 - small file names (MySQL varchar has a max size of 255)
 - single files (currently, only single file uploads are allowed, but this is temporary)
- Challenges:
 - avoiding duplicate named files
 - ensuring both the DB and FS were updated concurrently

Download Function

- Purpose:
 - allow users to download files from the server to their computers
- Assumptions:
 - file must exist in the database (files from the table come from the database)
 - single files (currently, only single file downloads are allowed, but this is temporary)
- Challenges:
 - making sure files come from a request, not just letting the user navigate through the website

3. Process

Challenges

- Large File Uploads [Gave Up (for now)]
- Large File Downloads [Gave Up (for now)]
- Transitioning to AWS
- LocalHost Debugging for Everyone
 - disabling MySQL connection for Front-End development
- Communication and Scheduling
 - Sitting down and intentionally making time/setting expectations on coordination