PDF Watermarking System

Version 22-mar-2022

Problem definition

Consider a system which stores candidates' CVs in PDF format: a user uploads their CV and we store it for later querying/retrieval. During this process, we need to watermark it: we need to superimpose an image on top of the original PDF resume.

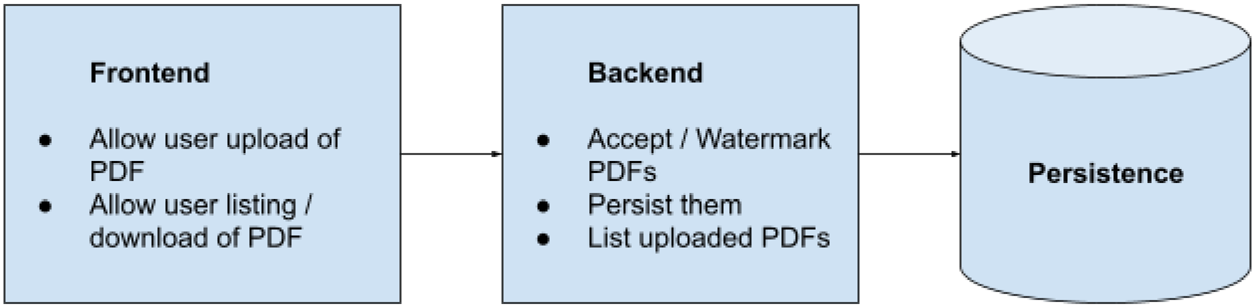
> **Note**: A classic example is to watermark an image with a company logo to prevent copyright infringement; iStockPhotos does that when searching and previewing their images, see below:

Upon uploading a PDF, the system will apply the watermark (specified as an image) and will store both the watermarked version and the original. The watermark image will be the same for each and every PDF file uploaded and as such does NOT need to be specified with each PDF upload request.

The system offers an endpoint that allows uploading (and watermarking in one go) a PDF and another one for listing and downloading previously uploaded CVs, both the watermarked version and the original.

Deliverables

We would like you to build a basic application to support these requirements. Along with this, build out a server using a node-js web framework of your choosing that is capable of accepting PDF uploads and listing those previously uploaded, along with the appropriate mechanism to serve the raw files. It is up to you how you handle server-side state, whether it is in memory, sqlite, a json document, etc. just note we expect it to be persisted between requests (not necessarily between restarts!) and the code to be of sufficient quality. The architecture is up to you as long as the frontend is capable of doing the above.


Frontend
- Allow user upload of PDF
- Allow user listing / download of PDF
Backend
- Accept / Watermark PDFs
- Persist them
- List uploaded PDFs
Persistence

An Example Architecture

A template has been provided for you here: https://github.com/mdisrupt/watermark

2 ways to send us your coded solution:

- Via a public github repo (send us the link)
- Zip up the directory with your code and send the zip via email (to engineering@mdisrupt.com)

In both cases, to make it easier for us to mark it, add a plain text file (readme.txt) in the root of your project containing details on how to compile and run the project locally.

While not necessary, it is a good idea to document your code, especially in places where you made certain decisions on implementation / architecture etc which might not be obvious. You can achieve this either via comments in the source code or by adding sections in the readme file detailing these decisions and how you arrived at them.

Also, in the readme file, make sure you detail what are some shortcomings of the solution? Every project has to make sacrifices / compromises, some known, some unknown. Are there any that you made (particularly in the interest of time) that you think could cause issues down the line?

Final Question

In a few words, if you were to extend this to be backed by S3 rather than storing files locally, what design changes would you need to make? How easily could you accommodate this?

Bonus Points

The following are not necessary, but would help impress:

- Using Typescript with detailed type declarations
- Nicely documented functions / endpoints
- Functional React using hooks
- Tailwind css
- Proper error handling and a nice api

Guidelines

This technical challenge has been designed to reflect some of the work we do. There are multiple ways this can be implemented, however please note that we are interested in seeing how you tackle problem solving and reason with the engineering tradeoffs. This is not an invitation to write suboptimal code, we still value good clean code; we are not focused though in this exercise in finding the perfect solution. You are free to use whatever tools / libraries you like to complete the assignment, however we are early-stage and currently need engineers who can contribute to the source code quickly, so we have created templates for a typescript / react frontend and a java / spring backend, to match our stack. You are allowed to change the templates (or ignore them completely); if you want to use other languages / frameworks you are welcome to as well.

UI / UX

We have no specific constraints as we'd like to see how you approach design and styling. Flashy stuff is fun, and feel free to show off what you can do, however the functionality is most important and we want to be respectful of your time. Simple and functional is better than pretty but buggy. You are more than welcome to use a UI component toolkit if you want to. If there is additional relevant data you'd like to surface on the UI, or you have any fun ideas, go for it!

Timelines

Typically we allow 1 week for this assignment to be completed at home at your own pace; if however you finish earlier feel free to send it ahead of time. Or if you need more time to work through it please reach out to us (see below) and we can adjust the completion timeline as needed.

Getting in touch

Finally, some aspects of this assignment might not be clear, in such cases and if you have any questions and/or need assistance throughout this challenge (or you are trying to provide feedback!), please reach out to engineering@mdisrupt.com and someone will get back to you.

Good Luck!