Cheetah Sign

Iteration 1

Levi Shelley, Riley Jamison, Hunter Johns, Tia Self

Project Summary

Project

Cheetah Sign
An in-house e-signature application for Accutech administrators to send and track legal documents to clients.

Client

Accutech Systems
Project Owner: Josh Rittenhouse
Mentor: Gabe Chandler

Mentor Feedback

Be Careful with API

We made our endpoints direct functions with a singular purpose - ensuring we don't clog the API.

Separation in Front End

We made sure that different features of the front end were put into their own Vue components, making them easier to split up and handle.

Tidy Up Code

We ensured our messy-spaghetti code made from functionality was tidied up for easier coding in the future.

Client Feedback

Separate pages for each table

Prioritize separating the admin page into different ‘tabs’ for iteration 2.

PDF rendering through vue

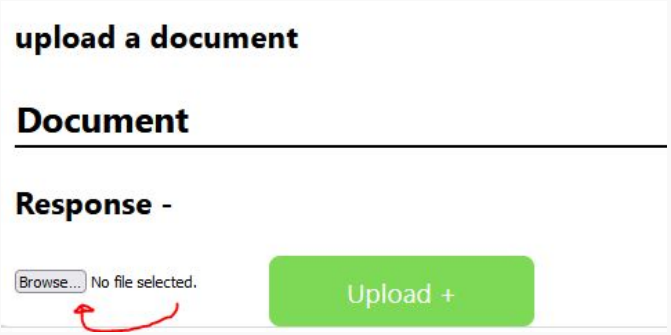
It’s vital that clients feel like they are directly signing a document, not just checking a box on a website.

Good proof of concept

yipee!

Iteration 1 features

Upload Documents



- Created an HTML Form that takes type ‘file’ as the input
- Clicking upload calls a function that creates a form data object that captures the file upload
- The form data object becomes the body of our post request and is sent to the server using our Minimal API
- The form data is sent to our /upload endpoint and is handled as the C# type ‘IFormFile’
- The IFormFile is then converted into a byte array form
- Lastly, it’s added to the database as a byte array type, along with the file name

Store Documents

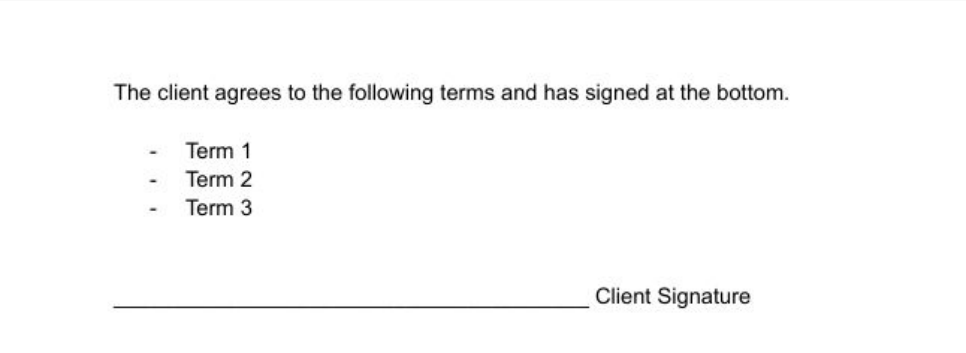
- To connect to our PostgreSQL database we used Microsoft's Entity Framework
- Entity Framework maps database tables to C# classes (known as entities)
- The most important class in EF is the DbContext, it acts as a bridge between your code and the database
- In this class you define your sets and entities which are your database tables
- Each time you update your database it creates a migration file. Think of this as a version control for your database schema

Send Documents

- All Documents are listed after being grabbed from the API
- User can input a name and the document and name get sent to the DB as a job
- Job contains filename, clientname, and status (“Sent” or “Signed”)

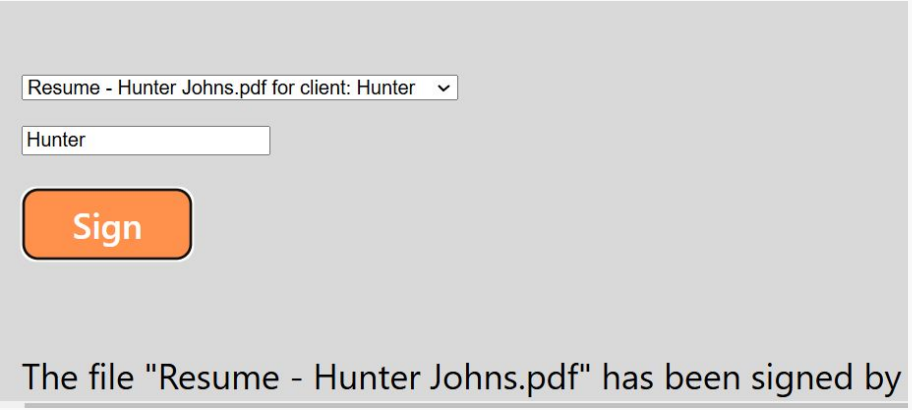
View Documents

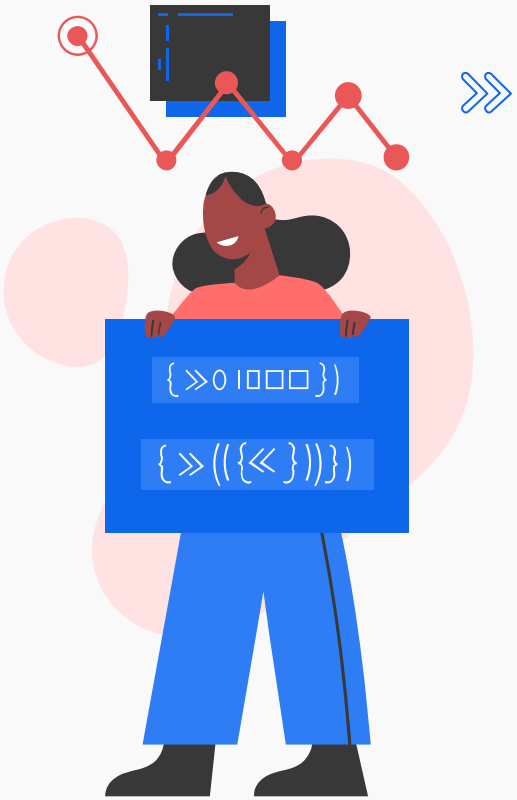
- Any documents, uploaded or sent, are stored in the server
- The documents within the server can be located by name
- Documents are exposed by an endpoint in the API
- Can be accessed directly through the endpoint and be downloaded.



“Sign” Documents

- From the list of documents found within the DB, populate a dropdown menu
- Only when a document is selected, provide a text box for your name and a sign button
- If document is selected and name is filled, change status of document to signed and provide a confirmation





Software Demonstration

Iteration 2 Features

01

PDF Rendering

The browser will pull up a rendered version of the document, including where to sign

02

Altering Documents

Fill in the blanks with the corresponding information, and return the new, signed, copy

03

Email Documents

When a URL can be generated for a signable document, it can be sent directly to clients

04

Conclusions

This will be the main iteration that will make it feel like a dedicated document signer

Retrospection

What we learned from this iteration:

- What does each team member think about this iteration? (i.e., lessons learnt)
- Riley - I learned a lot of about setting up a more complex development environment, using Docker, HTTP requests, and our tech stack in general.
- Hunter - Capstone projects are an entirely different beast from in-class assignments
- Levi - I feel significantly more confident with post/get requests and front end design.
- Tia - Working with tables in Vue was tough at first, but it allows for more than basic HTML.

What properties of quality software did we sacrifice for the sake of functional software?:

- We kept everything on a single page, making it a little cluttered
- There isn't an engaging signing process yet, just a sign button and a name field
- There is not a login system, meaning the information is not yet as secure as we want it

Retrospection

Our Approach for iteration 2:

- We want to emphasize PDF rendering. Accutech made it very clear that this is a high priority that they want to see early on, so it needs to happen now.
- We want to make the front end less clunky to use. More pages for documents and delete buttons are a massive part of this.
- We want to start implementing network features that are expected of a web application. Being able to email the document access to clients is a very important part of this.


hallloweeeeeeeeeeeeeeeeeeeen