ESTATE VAULT

First Iteration

FROM THE TEAM

ESTATE VAULT DEVELOPERS

"With such a quick turnaround and heavy flow of assignments, I am unbelievably proud of the team, and I'm excited to see where Estate Vault goes and where it takes us!"

Peyton Schaefer
Frontend

"I think the iteration went about as well as a junior developer's first big project iteration could go. I learned so much in such a short amount of time, and I can't wait to learn a lot more over the next few months."

Landon Hammond
All Rounder

"This iteration, with the help of my team, I learned so in what feels such a short time. Having to step my feet into things I was not comfortable really paid off. I am excited to see what we can do moving forward, and I excited to learn more. "

Cameron Allan
All Rounder

"Our first iteration was a great learning experience. We had some huge successes and some big failures along the way. Our whole team is incredibly excited to grow our software development skills over the course of this school year."

Nate Slagter
Backend/API

- Accutech fulfills its purpose to make great things happen for other people by delivering innovative trust and wealth management technology solutions and exceptional, personalized service to over 200 banks and wealth management companies nationwide.
- Located in downtown Muncie.

MENTOR MEETING (MID ITERATION)

- Michael suggested using AWS for storing all information related to project.
  - Accutech uses AWS for their systems.
  - Integrated the document storage system with S3 Buckets
  - Plans to migrate the Postgres database to AWS RDS.
- Michael gave feedback for switching to Postgres to fix errors running on mac, unfortunately to no avail

CLIENT MEETING

- Dan and Accutech eventually want everything hosted on AWS.
- Dan was mostly interested in what is to come
- Unfortunately the document storage was not set up officially
- This iteration and especially this last week was a catapult

FIRST ITERATION FEATURES

- Frontend File Upload with Axios and Tags
- Frontend Navigation
- User Registration and Login
- File upload system
- ASP.NET Web Api with two controllers
  - Each controller contains multiple different routes.
- Postgres database to store user information.

FRONTEND NAVIGATION

- Simple and Easy to Navigate UI
  - Older user base
  - UI that is able to jump to mobile with minimal changes
  - Typical user interface for a website, home page with navigation routes to pages such as the document vault

FRONTEND NAVIGATION

FRONTEND FILE UPLOAD

- Our good friend, Axios
  - Axios makes it really easy to send files
  - Utilizes formData with a content type of multipart/form-data
  - Backend is able to accept this easily and alleyoop it to AWS
- Tagging Files!
  - The connection with UI and S3 allows users to tag files when they are uploaded and it will automatically place that file in the tagged S3 directory
  - This also allows for a cleaner user flow and separation of information

FRONTEND FILE UPLOAD

JWT AUTHENTICATION

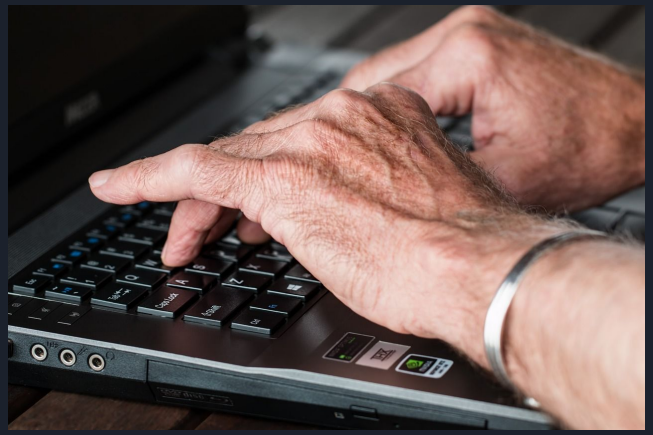
- API uses JWT Tokens to authenticate a user and provide access to the API.
  - JWT Tokens are bearer tokens.
  - The frontend passes them to the backend when making calls to the Document endpoint. If you do not have a valid bearer token, the endpoint will refuse your connection.

DOCUMENT ENDPOINT

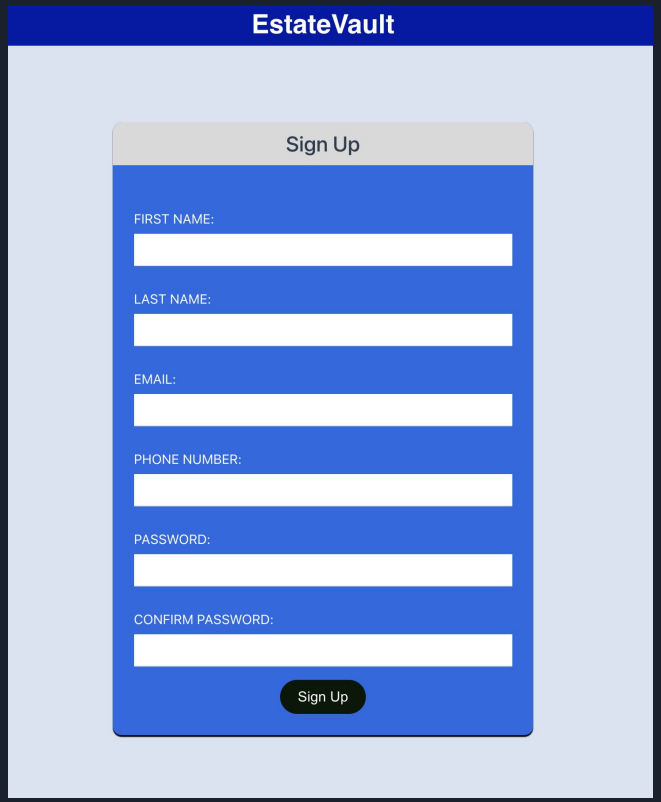
- API exposes two endpoints related to client's documents.
  - POST request sends a file to the backend.
    - AWS handles the rest.
    - Files have a tag associated with them to state the type of legal document that they are.
  - GET request retrieves list of all files that a user has.
    - These are sorted by their tag.

FRONTEND USER LOGIN

- Axios was very helpful here
  - A form in the frontend that calls the backend with the users data like name, phone number, email, password
  - Receives a user data object and stores it in localStorage for displaying the name on the homepage and userID for calls to the backend (S3)
  - Stores JWT, groundwork for later



FRONTEND USER LOGIN



USER ENDPOINT

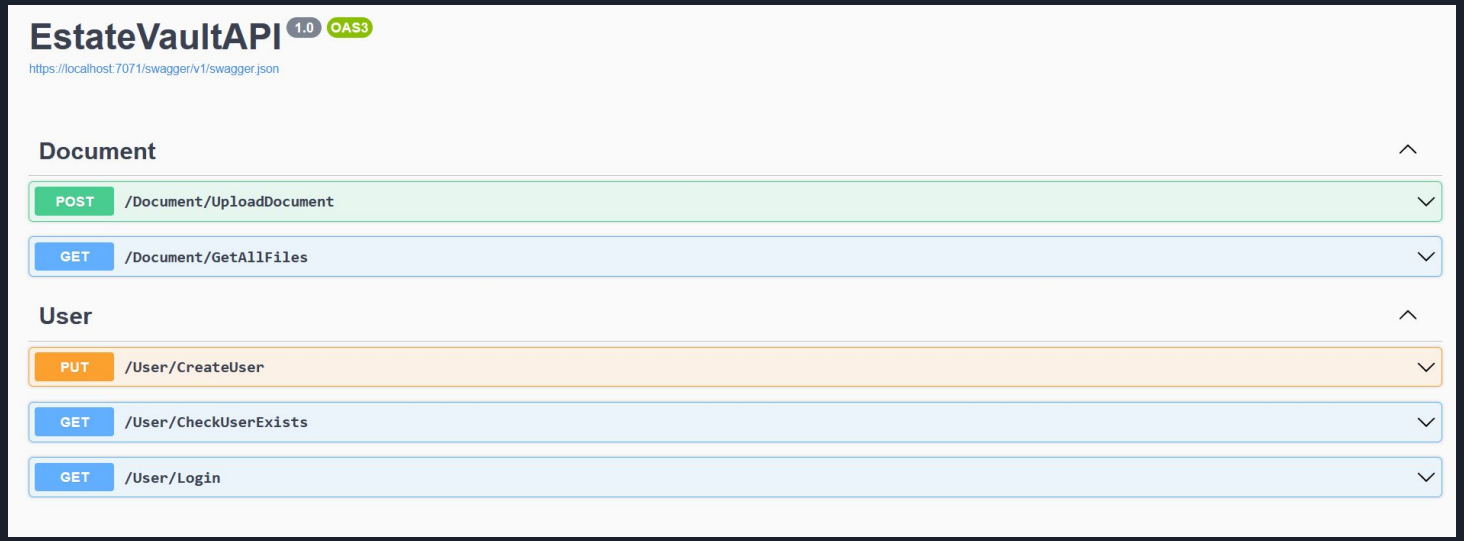
- User endpoint allows for multiple different routes to be called related to users.
  - PUT request for creating a new user.
  - GET request for getting an existing user's information.
- User endpoint is also in charge of returning the JWT Authentication Token back to the browser.

API AWS S3 INTEGRATION

- API uses AWS services to store files received from clients.
  - Documents are hosted in the cloud in an S3 bucket.
  - The API can retrieve files pertaining to a certain user and send them back to the user.
  - Much more secure than using a local database.

SWAGGERHUB

- SwaggerHub automatically generates documentation for the API.
- Displays schemas needed for request bodies.



Schemas

User {

firstName	string *nullable: true*
lastName	string *nullable: true*
phone	string *nullable: true*
email	string *nullable: true*
password	string *nullable: true*

}

PLANNED ITERATION 2 FEATURES

- Migration of local Postgres database to AWS relational database.
- Rehaul of Frontend UI.
- Contact system to allow a client to add people to their network within the system.
- Permissions system to allow a client to limit which of their contacts can view certain files.

SILLY/GOOFY SLIDE

NOT PHOTOGRAPHED
LANDON AND CAMERON
AT THE OTHER DAIRY QUEEN