Christian Farris
cfarris@wisc.edu

Journal Entry 1 - 2/10
I'm a front-end focused engineer, so the main focus on mine for this project is going to be on the front end. My main focus for this project would be the front-end, specifically the layout of the components, utilizing vuetify to its potential, and making my components reusable.

I laid out the pages to be:

- About
- FunFact
- Index
- Login
- Other
- Profile
- SignUp
- UpdatePassword

As far as components for all the pages:

- NavBar
- Footer

I went ahead and set up the NavBar and Footer components, in addition to adding them to all the pages.

Journal Entry 2 - 2/11
I started laying out how I wanted the simple static pages to lay out.

- Index
    - Link to login page
- Profile
    - Link to static cards on sub pages
- About
    - About Section card component
- FunFact
    - Fun Fact Section card component
- OtherStuff
    - Other Stuff Section card component
- Links
    - Links Card component with accordion component

The three pages (About, FunFact, OtherStuff) can all use a shared component with props to handle the differences between them. Layout wise, they can all layout the same. These components can use the v-card, v-card-text, v-card-heading, and v-card-subtext components from vuetify. Index page can feature the same card, linking to the login page. The links page can use the same components as the other three sub pages in addition to the v-expansion-panel components. The profile page can use the components rendered by the pages all on the same page. This required messing with the template, script, and style portions of vue files. The script parts were importing components and setting up props.

Journal Entry 3 - 2/12
Starting laying out form pages. I created the forms for Login, Signup, ChangePassword, UpdateInformation.

- Login
    - Uses email and password to fetch user data
- Sign Up
    - Creates a new user, must have the same password. If any fields are empty, it fills them as empty strings.
- ChangePassword
    - Change the password of user
- Update Information
    - Load information from users into form, on update submit any changes to form.

Implement forms into components using vuetify. Then plug components into formed layouts on the pages. This required me to make vue components with the template, script, and style sections. I specifically used components, methods, data functionalities in the script tag.

Journal Entry 4 - 2/13
I set up state management on the front end today. This required me to add the computed property to pages that needed state management. This would be used for user data, logged in status, page edit status.

I noticed some page delays when changing the logged in status, so I set up a welcome component to render in the between state.

Journal Entry 5 - 2/14
Did not work on the project today

Journal Entry 6 - 2/15
Next up I began setting up the table for my project. I wanted to restrict the project down to one table, so I did not have to worry about foreign keys, one to many relationships, or many to many relationships.

I set up a simple users table with an id then all the necessary data for the users.

Journal Entry 7 - 2/16
I initialized a new flask project, got it to print hello world on the local host. After this I started reading sqlalchemy docs to learn how to set up the database with the flask server.

Journal Entry 8 - 2/17
Did not work on the project due to a midterm the following day.

Journal Entry 9 - 2/18
I figured out how to connect the flask server to the database using it’s uri. After this I set up the user model and attempted to get the user endpoint to function. I struggled with receiving a proper response.

Journal 10 - 2/19
Did not work on the project due to medical issues.

Journal 11 - 2/20
I got the user endpoint to return json. I wrote a serializer function to serialize the information retrieved from the database into a json object. I also finalized how I wanted the user model to be created.

After getting the get endpoint to work, I set up the post endpoint so that users could sign up.

Journal 12 - 2/21
I wrote put endpoints to change the users password and to change the entirety of user data. This relied on me writing an extra method to set the fields of the desired db row to the desired values.

PROJECT REFLECTIONS

Frontend Notes:
My main goal of this project was create a sleak, reusable nuxt project that used state management to handle user authentication on the front-end. Ultimately this project achieves that. There’s a large amount of viewed space that is sharing a single component. I also leveraged the vuetify library rather well. In addition to this the css is clean and easily readable.

Backend Notes:
I’ve never written a traditional REST API and am definitely a giant fan of GraphQL. Coming from a GraphQL background, this experience only made me value the improvements GraphQL has made for apis. In addition to this, writing python for anything other than simple scripts is something I’ve always dreaded doing. This project was no different. It pushed me towards wanting to write Javascript even more. It really showed me the strengths of JavaScript having to complete these tasks in python. In addition to this, having to alternate between two languages for the back and front ends of the project was frustrating.

DataBase Notes:
Working with PostgreSQL and relational databases is always straightforward. The complications with this were more so with python and flasks documentation of their main database module, sqlalchemy that had subpar documentation in my opinion. Nonetheless, PostgreSQL didn't cause any unforeseen issues.

Resources:

- **Front-End**
  - https://vuejs.org/
  - https://vuejs.org/v2/guide/instance.html
  - https://nuxtjs.org/
  - https://vuetifyjs.com/en/
  - https://css-tricks.com/snippets/css/a-guide-to-flexbox/
  - https://www.npmjs.com/package/vuex-persist
  - https://github.com/championswimmer/vuex-persist/issues/70
  - https://github.com/axios/axios
  - https://kapeli.com/cheat_sheets/Axios.docset/Contents/Resources/Documents/index
- **Back-End**
  - https://flask.palletsprojects.com/en/1.1.x/
  - https://flask.palletsprojects.com/en/1.1.x/api/
  - https://flask-restful.readthedocs.io/en/latest/
  - https://docs.sqlalchemy.org/en/13/
  - https://www.youtube.com/watch?v=MwZwr5Tvyxo&list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH
- **Database**
  - https://www.youtube.com/watch?v=xaWlS9HtWYw&list=PL-osiE80TeTsKOdPrKeSOp4rN3mza8VHN