

Self-Assessment

In the process of developing ULang, I learned a lot about the inner workings of a web application and how everything interacts with each other. One of the things that I spent a lot of my time on and learned the most about was global state management within the front end of the application using a library called Vuex. Vuex is a library that provides a state management pattern that integrates with VueJS as a user interface framework to provide a way to access certain pieces of data from anywhere in the application easily. I used it to manage things like which language the user was currently studying to be attached to many of the API requests that were language specific, managing the user's logged in status and displaying different UI elements based on that, and some other settings and pieces of the user's information. This really helped me to understand how to make a component-based application work on a fairly large scale rather than trying to do what is called "prop drilling" – meaning emitting data up the chain of nested components and then sending it back down a different chain if needed – which is something that I was originally trying to rely on but didn't work very well and caused a lot of issues early on in the project. Another thing that I learned a lot about was security. I implemented a system with JSON web tokens to provide the user with a token when they sign into the application and the server checks and validates that token for every request before fulfilling any HTTP requests. I also added a hook in the database querying service to hash the password of the user before creating a new user as well as hashing the submitted password to check if it matches when an existing user attempts to log in. This is something that will be very valuable any time I need to make an account-based application, which I expect will happen when I start working full-time after graduation.

I also spent a lot of time building different pages of the front-end of the application and learned a lot about different styling tricks to make the components look exactly as I wanted. I am currently taking a web design course as an elective class, which has helped me learn about some new CSS properties and better understand some more specific features of ones I already use, such as Flexbox. All of this will also greatly help me with building any user interfaces in my future career as a web developer. There were some aspects of developing these interfaces that I think I could have done better though. For some of the views I created, I should have broken it up more into sub-components to abstract out some of the work to other, possibly re-usable, components to cut down on the complexity and size of the Vue files I was making. I also learned a bit about unit testing Vue components, which is another very valuable skill, particularly when working in an Agile team where proper testing is imperative for rapidly releasing new software to keep defects to a minimum. I also did get to spend a lot of time designing many of the pages seen throughout the interface of ULang. I used a lot of skills I learned from researching UI

patterns and common good practices to make a clean, balanced, and modern UI. Additionally, I also accomplished something that I had not originally set out to do but ended up being a nice way to wrap up the project, which was to containerize each piece of the application, the client, server, and database, with Docker. This should help to make releasing to the cloud in the future a more streamlined process because all that will need to be done is execute the docker compose file with each release to fill the new containers and push the containers to the cloud service.