CS523: Term Project Proposal for party.ryanstreur.com

Ryan Streur: streur@pdx.edu

January 23, 2025

I'll be calling this project rs_party, mainly because I'm planning to host it at party.ryanstreur.com, a subdomain of my (currently defunct) personal website. Broadly, the topic area is application development. The specific vision is for an event planning application composed of three components:

1. A postgres database
2. A Rust web api
3. A Vue.js front-end

Link to the repo

The part of the project which applies to the project requirements for CS523 is the middle tier, the Rust web API. I'll be using the Rocket API framework, with sqlx for database access.

1 The Big Picture

I'm planning to keep the data model for this app pretty simple. There will, of course, be Users, and those users will have Sessions. Application clients will be issued a session UUID which they will include in an Authorization header to implement a bearer token authentication scheme. Authenticated clients will be able to create and manage Events, and they will be able to create invitations to those events which they can share with other users. When a user visits the invitation link they can create an RSVP for that event. The owner of each event can see the invitations they've sent out and the list of responses to each. I'm also planning to add role-based authorization so that event creators can add other users as organizers who will also be able to create invitations.

2 What I've Done So Far

So far I've got the database and API elements of the application connected together, if a little messily. The different layers of the API — data access and controller, basically — are pretty much working together. I've implemented a registration endpoint, which allows a user to enter an email address and a password (no validation or password requirements yet) and stores that user in the database. I've also scaffolded the client application, but done little development work on it.

3 The Weeds

I spent a good chunk of time struggling with getting database access in different parts of the API layer. Specifically, Rocket has a few macros which make writing API endpoints more convenient, and getting a reference to the connection pool within an API route is no problem. However, I was trying to figure out a way to put database access in some middleware so I could log requests and the results (status code, at least) to the database. I was also trying to put data access in middleware which runs before the main route processing which would access the database and fetch the user to provide it as a struct for the main route processing function. These two lines of development have proven so challenging that I'm thinking the only winning move may be not to play.

4 What's Next

My plan for the logging is to put a log statement at the end of each route function till I can figure out a cleaner way to do it. With the middleware, I can get the session ID out of the auth header and provide that to the route. The problem with that is that it could get a little repetitive if I'm always querying for the session for authentication and authorization, but I'll cross that bridge when I get to it. Really I just want to make sure I get started writing the CRUD endpoints for the core event / invitation / RSVP models so I can get a working version up.

5 Concerns

Now that I've figured out the session-in-the-header thing, I'm feeling much more confident about this project. I think I should be able to get it all done in time, but I feel like my experience with this language is such that I may not yet fully know what I should be concerned about.