

Write-up:

I have very mixed feelings about my accomplishments.

I feel that overall, I did okay – it is my first time creating an unguided web application, as well as my first time using Golang and Postgresql for a backend.

However, I underestimated the time and complexity needed even for a “simple” web-app with bare functionalities, due to the added difficulty of Redux, Typescript, Golang etc. This led to me not planning well and running out of time, only managing to implement a very bare-bones, unpolished product. Hence, I feel disappointed as the web-app that I deployed lacked a lot of features that I sought out to implement.

One example is the karma system – the naïve implementation of storing karma as a simple counter would result in many problems like a user being able to upvote/downvote an unlimited number of times. Hence, it requires a lot of extra work like creating additional SQL tables, storing each upvote instance from each user and additional React components, which I did not have enough time to do.

Another example is the thread system from Reddit – it is a feature that I learnt how to appreciate now. While trying to implement it, I realised that it requires a lot of technical skill that I do not have right now – the implementation would require having to deal with deeply nested structures, not to mention the frontend design would be very challenging to implement. To keep things manageable, I only allowed comments to have one direct parent.

Short user manual:

Users can create accounts and login using their username and password. (I am aware that plaintext passwords are bad, and a proper implementation would be to use an encryption library)

Logging in would get a JWT token from the backend, which is used to authenticate requests and show forms which are not visible to non-logged in viewers.

Threads can be created, and comments can be created with respect to threads and at most one parent comment.

There is a basic filter present – which filters for a single tag and shows corresponding threads.

However, no delete and update functionalities are available as I did not have enough time to implement it properly.