

miguelf4, rgyanm2, mcroos2, ak85; Team Captain: miguelf4

Team WXYZ

## Project Progress Report

In the time since the project proposal, our team has spent the majority of our time with research and planning for our chrome extension. While we were all familiar with chrome extensions from a user standpoint, we had no idea how they actually functioned. We each spent roughly 2 hours individually going through the official chrome extensions tutorials at <https://developer.chrome.com/docs/extensions/mv3/getstarted/>. This proved to be extremely useful as we now had a foundation of knowledge to build off of. It turns out that chrome extensions are very similar to web-apps and are composed of html, css, and javascript.

Now that we knew the basics of how chrome extensions worked, we also learned of a limitation, there's no easy way to run other programming languages. The reason this is a problem is because there are many existing libraries (specifically in python) that will make the computation for our retrieval functions much more simple. To overcome this problem we found that the best way would likely be to run a cloud-based server completely separate from our chrome extension. This server can do all of the "heavy lifting" (python code).

Finally, now that we had a much better understanding of the lower level details of our project we began to work on basic implementation. This breaks down into the chrome extension (client) and the server. We split up the work based on who had more experience with frontend or backend. For the client, we first tried to follow a tutorial that would allow us to use REACT and typescript to implement our UI. While this would be nice, it seemed that it was more trouble than it was worth as we had to use custom compilation techniques and things that went a bit over our heads. So we went back to basics and decided to not do anything too fancy (still just html, javascript, css).

Next we started designing and implementing the REST endpoints on the Python backend server (using Django) to compute the BM25 ranking function and return the result to the front end.