I’m now effectively around day 55 of the 100 days of code program. I should be past the halfway point and really getting through the advanced python topics, but my training has not gone as expected. It is safe to say that my Python skills do not match those expected of someone who has completed 55 days of code. Truthfully speaking, part of the lag is explained by my sluggishness. I have not felt the best physically recently and that has affected my concentration and energy levels. Part of it is also explained by the social nature of the last couple of weeks. Saturdays, which are my go to days for work, have been consumed by other things recently. It has meant I’ve needed to push some of my work to Sunday and that usually has negative consequences for my Mondays. But there are other more laudable reasons why I have fallen behind schedule - I have been learning a bunch of other really cool things. One of those things is system design. I recently fund a useful blog by Semi Koen that listed a number of extremely useful GitHub repos. With a hint of exaggeration, I would say that the list is good enough to change someone’s life. The links are incredible, ranging from resources on how to ace an interview to practical examples of how to plan out the design of something like Twitter (with some Python code provided!). The other thing that has consumed my time is the ideation and documentation of Foundr. There is considerably more to think about than I anticipated. Even with a project this small, there is so much I am learning and having to think through. The overwhelming feeling that I would get seems far more manageable. I am becoming more adept at slicing of a piece of the puzzle and focusing on that. I have found that the more shape the program forms, the easier it is to move to different elements. For example, I recently spent a whole Friday reading up on security. I could have spent more time on it, but the rabbit hole was deep enough and emerging from it was necessary at the end of the day. However, it was still a day particularly well spent. If you’re wondering why I could spend a whole day working on security when it does not form part of my official work designation, keep wondering. I’m joking. It is because I did have work on that Friday and so I decided to wonder. For reference purposes, it worth nothing some of the things I covered. These included passwords and concepts around protecting them such as hashing, salting, and encryption. I read up on history and found myself in the middle of some long running debates. In the end, I put together a one-pager describing the principles that I would like to implement for Foundr. At this point, I find myself going down another rabbit hole of sorts. I thought I would start developing the user interface. I initially found one of the lessons in the 100 days of code that uses the Tkinter module. It is very similar in spirit to the turtle module, in fact I think the turtle module is built on top of Tkinter, so it is fairly simplistic. So I decided to find something more sophisticated which led me to the PyQt module. And then further to the Qt designer application which removes the need to add components using code. It was certainly an improvement of Tkinter but I felt it still lacked the level of sophistication I required. My final and current solution was to find the Django course that I had neglected a while back. Undoubtedly a steeper learning curve, but it should offer the necessary level of sophistication and transferrable skills that I want. It has been going well and it feels like the right decision. The only concern, and it is indeed a minor one, is that I might be moving away from providing a minimum viable product. It is usually advised that you get the program running as quickly as possible before adding frills. Tkinter and PyQt possibly represented the no-frills approach. I don’t know. I might only find out later.