

Lessons learned

1. It isn't always feasible to envision exactly how everything is going to function before you build something
 - a. You can get better at this with experience
2. When developing features incrementally it is nice to develop specifications for those features, develop them in the back end first and then be able to format everything from the back end in the front end for ease and satisfying development
3. Working on this project in spurts/ sprints heightened the quality of delivery. I developed something that exceeded my expectations for how well it would work
4. Developing a design beforehand doesn't mean you have to stick with that design it just gives you a more macro vision and understanding of the project that helps you head in the right direction that you can pivot off of or towards
5. Don't ignore non-functional requirements. This app could have been cool but taken a long time to provide each list of recommendations which would have made it less useable and a less desired project
6. Team development is much better than solo development. However if you are going to do team development everyone needs to spend a similar amount of time with the code/ project and pair programming can really help level your teams understanding of the project