1) **Benefits:** As I begin prototyping Clarity, a platform hat can be accessed through both web and iOS platforms, I think that I have truly come to understand the importance of tokenization. Initially, while making my first iteration of blank wireframes, I did not tokenize the components, meaning when I went back to add colour and specific typography, I had to add it to each component individually. Did I use OpenSans on the copy or the headers? What radius did I use for the “next” button? How much shading did I include to the welcome message? Tokenization becomes even more important when developing a platform over different delivery modes because what a 12pt copy looks like on an iPhone is drastically different than what it looks like on a desktop, but you still have to recognize that the different sizes are following the same design rules or it will look like two completely different platforms, which may confuse users and take away from their experience.

2) **Drawbacks and challenges:** As someone extremely new to the idea of tokenization, the main drawback that I see is time. If the app is complex and you believe that it will require many iterations to become approved, I believe that the time needed to set up the initial tokenization will pay off in dividends in the long run. However, if the platform is a small, basing one run by just a few people, I believe that the time spent tokenizing each individual component may take away too much time from other aspects of the platform that may make it succeed more that having the exact grid or shadow effect tokenized. That being said, the main aspects of an app or website such as colors and typography should always be tokenized, as those are very noticeable when changed and do not take as much time to make as tokenizing each individual component. Th greatest challenge that zi faced was getting started. I had the help of the figma plugin, but after I ran that, some of the colors and fonts were off, and I had to manually tweak them, including different names for each individual component. Another challenge that I had was visualizing what I wanted the app to look like in it’s final draft while I was still working on the first iterations. Should this colour be solely used for back ground, or can I include in in type somewhere else? Where can I use the secondary and tertiary colors? What fonts should be used where? Those are the questions that I asked myself when making the initial token set, and I continue asking them as the prototype continues coming together.

3) **Version Control and Governance:** When managing tokens on an individual team, I found it quite easy to make sure that they remained controlled as I was the only one inputing and tweaking them. However, I think that if I were a team made up of more than just myself, it would be extremely easy for a token system to get lost in translation, which would result in the final product looking un-cohesive. I believe that a code review including the designer’s approval for visual changes is the best practice to follow for a cohesive brand as the designer is the one who understands the aesthetic of the platform the best. While it is the engineers who bring the platform to life, without the approval of the designers, the app or webpage runs the risk of not following the existing brand design which may, in turn, take away from the brand experience.

4) **Tokens in the Wild:** After looking into tokenization case studies, I quickly learned that “tokenization” is a rather broad term that also applies to art and finance. If I were to try selling a company on using tokenization, I would describe tokens to be like building the foundation for a house. It takes a long time and a lot of effort, but when it’s done right, it makes your life a lot easier in the long run. If you have a house on the side of a cliff or soft ground and do not build a strong foundation, you will have to continue fixing the house to keep it from falling off the cliff or sinking into the ground, altering the overall design of the house and making it look more like patchwork than a properly designed house. Similarly, building an extensive token set takes a long time and gives very little o show at first. However, in the long run for your brand, it will allow you to scale and change and grow without looking like a patchwork platform, as you can just edit one token and have it affect

every instance that token is used throughout the platform . The style guide is how the design looks on paper, while code is how it looks on the screen. Tokenization allows for the style guide to keep the names and appearance, while having the code to use on the platform, acting as a “google translate” between the designers and engineers. Airbnb uses tokenization to have their design appear seamlessly between many platforms such as web, iOS, and Android to name a few, and allows for further growth within the company without needing to redo every component. Airbnb is a great example of how tokenization can improve scalability. In the beginning, as the company grew and new employees joined the design team, Airbnb faced an in-cohesive design due to the different fonts, colors, and other design components used by the groups working on different platforms. After tokenizing their design system, they were able to show their brand style throughout all platforms even though they had many different groups working on them, because they had presets for everyone to use. It is important to understand when it is too early vs when it is too late to tokenize your design system based on time, money, and expertise needed. When working on a group of three, it is probably more effort than it’s worth to tokenize the design system. However, as the team grows, tokenization is key before it gets too big and everyone is off working in their own groups with less communication than the initial three person team.