The Picobytes Core Team learned a lot throughout the last semester. We all gained significantly more experience in Flask, we utilized an external database in a Flask project for the first time, and we had a significantly bigger project than ever before. In terms of the frontend, we had a pretty interesting experience overall. When we started this project, the five of us quickly identified what we were good and bad at, and what we would like to work on during our sprints. We essentially decided that two would be almost all frontend, two would be almost all backend, and one would be somewhat neutral. This was a good idea, especially for those working on the backend, as we got our tasks done quite quickly; however, we quickly discovered that the frontend team was not on the same pace as the backend team. This, and some drastic communication issues early on, was a huge setback in terms of the development of the app itself; eventually, one person on the backend team (who was originally supposed to only do backend work) decided to take matters into their own hands, and clean up what they could. At this point, the homepage was looking desolate, with the same problems recurring (and nobody really attempting to fix them). The individual that decided to move over a bit to the frontend side heavily reformatted the entire theme and style of the website itself, utilizing the newest AI models to take inspiration somewhat directly from the popular language learning app, Duolingo. The result was an almost immediate improvement in terms of the style of the website, which eased a lot of tensions for the team. Perhaps if anything else, this taught us all that we cannot really be confined to a particular section of development itself, and also, we learned the valuable lesson that just because someone says they are going to do something, does not mean they are going to do it. On the back end, we learned many valuable lessons about how not to organize a project of this size, along with how AI can work for you and against you with something of this scale. We nailed it when in the development process we wanted to switch from a local database to an external database; however, the rollout of that database was less than smooth. We learned a lot when it came to teamwork and we never quite ended up with a system that worked perfectly. The Kanban board in Trello was very underutilized and we ended up communicating on what needed to be done, who was going to do it, and their progress more in slack than in the board. We found that pull requests were an excellent way to review code but never had a universally good way of handling prs that had merge conflicts. Robust testing of all code before pushing to GitHub would have been extremely nice to have and saved us from a lot of time spent working through bugs found after the fact. However, with the scale and time for this project, we're not convinced that the time spent writing the tests would have been less than the time spent troubleshooting. Overall, we managed to successfully complete our Minimum Viable Product and go beyond in some of the features we planned to have for admins. Our UI design looks very good and is usable by people of all skill levels. However, we did not end up with the robust lesson plans or auto continuing from where we left off that we wanted to. This would be great for a v2 of Picobytes. We also never managed to take our app beyond being hosted locally and while we managed to handle some errors gracefully, we weren't able to fix others to a point where an error was recoverable without having to re-run the program.