Evaluation of project from team planning perspective

- We didn't have many in person meetings, especially while coding. We made it, but how did it go in people's perspectives?
    - Shinjini: I liked coding by myself and working on my own time. I think the communication with fb chat was fine, especially because we were very responsive in general, although that would probably not be the case with other teams.
    - Kim: I prefer working in groups, so for me the times when I got the most done the most quickly was when I was working directly with others nearby. Specifically, I find it especially helpful to be able to ask questions without having to switch windows to enter an online chat.
    - Casie: I flipped back and forth between finding working with people stressful and guilt-inducing, and finding it motivational and helpful This made things complicated in terms of planning to work with people, but I also think the guilt was mostly due to the feeling of being restricted by how much work I was physically able/medically allowed to do.
- Our actual meetings got derailed very often - should we have tried to be more on topic?
    - Casie: I think it's good to get derailed sometimes and in moderation - I think we often got derailed with redundancy, which was less useful.
    - Kim: In spite of our occasional lack of focus, I believe we consistently left our meetings having finished what we needed to finish and with a clear understanding of what needed to be done next. I never left a meeting feeling that we had not settled what we needed to settle.
- Shinjini: I think our expectations of how long things will take was not good (pretty consistently). Especially the design docs and revision took a whole lot of time.
- Kim: Even though we did not have a full four-person team, we still managed to create a final product just as good as many others created by the rest of the class.
- Casie: I think in general, we did a good job of supporting each other through an obviously difficult project, but our on-topic technical communication could have been better.
- Kim: Since two of us were sick and one dropped, Shinjini ended up doing most of the work for the earlier part of the project, and in order for me and Casie to get any credit for the project we practically had to stop her from doing more

Summary of key lessons learned

- Shinjini: Life sucks when people get sick and leave :(
- Shinjini: I liked that we aimed high but were able to figure out key features while time was getting short
- Casie: Every problem in a group can be adapted to work for its members or some combination of its members.
- Kim: It's important to establish what people's work patterns are (What time of day/night do you work? How close to the deadline? For how long at a time?) more specifically than the ideal you strive for

- Kim: Each member of the team needs to know their contribution is valuable as something that the other members could not necessarily have done. This is important to keep up general team morale and thus to improve efficiency.
- Kim: When one team member wants to blaze ahead of everybody else, it can feel stifling to the other members of the team who aren't able to go as fast. This needs to be managed somehow.

What to do differently in the future

- Casie: I think doing more pair programming would have helped us a lot. Most pair programming was done over chat, which adds a lot of redundancy to debugging and communicating and general loss of time to typing.
- Casie: I think one of the problems we
- Kim: I also think pair programming would be a good idea and that even working in the same room would have been beneficial.
- Kim: We should set up a better distribution of work in terms of who contributed to which parts of the project (e.g. tests, routes, frontend, etc.).
- Kim: We should try to practice the presentation once before attempting to give it.
- Kim: We should have more thorough code reviews and write tests first instead of last.
- Shinjini: I disagree with writing tests first, because the functions you are testing inevitably change their inputs and outputs throughout development and you have to keep updating tests, and that's a time drain. I think they should be written only after we are reasonably confident that we are done with a function.
- Kim: We should not get as sick next time, obviously.