

Qui Nguyen  
6.170 Final Project Reflection

## I. Peer Review

### Casey

- Good at finding bugs and identifying important design decisions
- No issues with code, able to do everything that was assigned to her
- Her sleep schedule was sometimes a little different from ours, but she was able to finish her work and it wasn't a problem

### Annie:

- Did a really good job helping us with the styling (CSS/Javascript), and improving our styling skills
- Awesome job implementing the calendar, and other smaller features
- She often made big changes, which were always a good surprise, but would have been more helpful/prevented some extra work on our part if she had told us what she was working on in the beginning

### Katie:

- Always willing to do the less fun work (like testing and debugging the API, fixing the database), which was really helpful
- Could have checked code more while writing it

## II. Evaluation

We planned really well for the MVP and delivered everything that we wanted to. However, I think that made us a little complacent for the final, and we wasted some time. We should have spent more time deciding on features, and a timeline for those features, at the beginning, and stuck to it.

However, in the end, everything worked out well. We still met all of our deadlines, and the communication was good between us. We met a few times to work together, which helped us coordinate and was very motivating and effective.

We worked very well together, and I'm glad that I was able to trust any portion of the code to any one of my teammates. I'm happy that we implemented something that could be used by teachers to improve the classroom experience, and I might even want to keep improving it in the future.

## III. Lessons learned

Good design beforehand makes your code much cleaner and easier to maintain, especially in a group

- For views, knowing what classes, etc should exist makes CSS much more streamlined

and less prone to conflicts/errors

- If features/functions are planned out, specific pieces of code are easier to find

Better to write tests while writing the features (if not before, at least several times throughout)

- Then you can remember exactly what the methods should do
- Find bugs early on, before deadline

After implementing the fun parts of an app, there are a lot of features, security, etc that need to be implemented and well-tested for it to be actually usable. This time shouldn't be discounted while planning.

- We had to do a lot of this at the end