## Writeup and Reflection

We were able to accomplish our goal of building a calendar web-app with a focus on Canvas integration. The user can add, delete, and edit assignments from a sidebar interface that prioritizes user experience and utility, or they can choose to import an .ics file from other calendar apps, namely Canvas, to store their assignments in one place. Users can also mark assignments complete when they are finished. Within assignments, users can add the assignment's due date and time, the subject, and the details. Assignments are listed on the sidebar to give users a heads-up on what they still need to finish.

We were not able to accomplish one of the stretch goals, which was our implementation of AI syllabus parsing. This was an ambitious goal given that none of our team members had any experience implementing OpenAI's APIs. We also did not get to "game-ify" Planvas with the leaderboard functionality. However, neither of these affect the core functionality of the app, so we prioritized getting the web-app functional without any bugs in its fundamental goals. If we had more time, future goals would be to implement this syllabus parsing as well as leaderboard functionality. We would also like to try to implement a way to scrape assignments from canvas automatically, so that users do not have to update the calendar manually when new assignments are given.

We believe that the project is an impressive accomplishment given the lack of previous experience in our team. The first few weeks were spent learning unfamiliar developer tools, and this struggle continued throughout the course of the project as we kept running into issues with version incompatibility, dependencies causing errors, and programs refusing to work together. Along with these issues was a general lack of knowledge in the workflow involved with using these tools, with our development progress being slowed when we prioritized the wrong parts of the web-app first and had to essentially scrap code that would no longer work with the direction the project was headed. Our decision to "specialize" and "divide and conquer" only served to exacerbate this problem. This lack of "being on the same page" so to speak forced us to rewrite certain backend functions and sacrifice some of our security features. For example, we needed to quickly rewrite User creation backend files and needed to make the decision to sacrifice our password encryption to save time.

If we had more time, figuring out a password encryption based outside of Spring Boot would be something we would also like to implement. We think this could have been better handled had we gone with a more full-stack approach, where all of our team members had at least an idea of how things worked on all aspects of the project. In reflection, everyone on our team thinks that if we had prioritized weekly in-person meetings as opposed to frequent text exchanges, our work progress could have been expedited.