Progress Report

- Deployed MVP
- Submitted Reflections

Agenda

1. Review Progress report.
2. Go over Feedback for MVP:
    - Great UI
    - Great attention to detail (even down to the package.json!)
    - I notice you have some deeply nested callbacks. If these become annoying to write/read, I'd recommend looking into the bluebird promise library or async.js (less powerful, but easier to learn). You don't need to use them for this project, but they may be useful to know about for future projects.
    - Why is your server listening on port 80? This is a privileged port. Perhaps you should listen on an unprivileged port (e.g. 3000) so that it's not necessary to run your app as superuser. The reason I recommend this is because running as superuser is a security risk - if an attacker compromises the superuser process running the app, they have full privileges to wreak havoc on your server. Running as an unprivileged user prevents them from doing as much damage.
    - In the interest of making your code as testable as possible, try to move as much functionality as you can to the model. You've already done a good job with this by keeping your controllers (routes/api/*.js) short, but see if you can make them even shorter and make sure that for future features, the bulk of the implementation is in the model modules.
    - I notice that you use functionals on the server side, but just know that you can also use them on the client side (e.g. map, filter, reduce)
    - Why did you write your own ajax() function instead of using jQuery's ajax() function
3. Address Division of Labor
4. Discuss Demonstration of Almost Final Version

Minutes

- Things to change
    - Server defaults to 80
    - routes are too long for now. Will be penalised in the future? perhaps move some logic into the model
    - ajax() function - get rid of it. don't duplicate code.
    - Clean up code, reduce duplication, refactor
    - use more functionals (referring to nested callbacks)
        - foreach
        - map, filter, reduce
- security
- tests (move to models!)
- specs
- Suggestions on splitting work more evenly
    - using intermittent deadlines
    - code review
    - dividing backend, algo, ui etc. as opposed to features

- functionality required going forward
    - **deadline, draft pages (priorities)**
    - gradebook improvements -- not enough time to build that for now
        - jks don’t do this at all
- Final presentation rubric (about 5 minutes)
    1. demonstrate major functionality (from both student and teacher)
    2. make it clear to audience what the app does and what the point is
    3. be prepared, smooth presentation and make the sample data ahead of time!
    4. avoid bugs in the middle of the presentation