Standup 10/30

Next Steps

- **First ADR**: Maitree to create first ADR for Greenfield vs Brownfield (pros and cons of each), should be in our github ADR folder
- **Project Pitch**
    - Everyone please work on your assigned portion of the project pitch document. We should aim to have this completed **by Monday Nov. 4th**.
    - We've already decided on most of these areas as a team and they are documented in the notes below, so the pitch document should simply expand on what we've already discussed
- **User Personas/Stories**
    - Everyone please complete your user personas and stories based on the prompts Maitree will send. Please have these completed **by Friday Nov 1st.**
- **Next meeting**
    - We'll meet via Zoom on Friday, Nov 1st.
    - Basic agenda
        - Decide which features are primary and secondary
        - Come up with rough timeline
        - Check pitch progress

Team Meeting

- **Project roles**
    - Backend engineers
        - Ishan
        - Siddhant
        - Ronit
    - Frontend engineers
        - Sarah
        - Maitree (if needed, can sub for Ishan in backend while he's out of town)
        - Hanzhi
        - Shelby (can also lead UX research)
    - Everyone will be responsible for their own documentation – we can define what this should look like in the code architecture phase
- **Problem statement**
    - Between interview prep, personal projects, and classes, CS students have a lot of priorities to manage. We want to create a dashboard to help students juggle their busy schedules and priorities.
- **User personas/User stories**

    - Since we are creating a product that we would use, we should be the user personas
    - We will all write our user persona and story based on the format provided by Maitree
- **Risks/rabbit holes**
    - We may not finish the project
    - Too many features
    - APIs we're adding
    - We probably need to research and add more to this section for our pitch
- **Features**
    - These should be divided into primary and secondary so that we can create a timeline – we can discuss this more on Friday
    - Potential primary features
        - Overview
            - Overall to-do list
                - Being able to organize by class, project, internship prep, etc.
                    - Auto-classify projects and tasks based on keywords (nice to have/if time)
            - Quicklinks (github, etc)
            - Calendar/schedule that displays message when close to due dates
        - Classes
            - Schedule, assignment calendar (sends to google calendar), discussion hours/syllabi, quicklinks (like mattermost and course sites)
            - Specific to-do list associated with classes
        - Internship tracker
            - Application tracker – status, where they've applied, OA etc
            - Leetcode problem suggester/suggests new leetcode problem
            - Tag leetcode problems you can't solve on first try and it will display reminder to retry after a few days
            - Specific to-do list associated with internships
        - Personal projects
            - Project tracker and to-do list
            - Project quicklinks
        - Sign on flow
            - Dashboard building when someone signs on/sign on flow
                - Helps build dashboard – ie if they're not taking classes it will remove class dashboard
    - Secondary features (if time permits)
        - Drag and drop widgets so people can customize their dashboard, organize based on importance, or hide widgets they won't use
        - Adding people to assignments, making teams and tagging people

        - Mobile app (siddhant)
- **Requirements**
    - Mobile optimized
    - Dark mode option
    - Secondary requirements (if time)
        - Meeting more robust accessibility requirements
- **Wireframe**
    - Homepage dashboard the important things for each feature
    - Clicking on nav bar view to narrow in on class, project, or internship pages
- **Timeline**
    - First step will be code architecture
        - function definitions, comments, naming conventions, when we go to a new file, how documentation should look – anything we can do to make it look like the code all came from one person
    - We'll decide on the rest at our Friday meeting

TA Meeting

Here are the things that were clarified during our weekly meeting with Kashish:

- **Tech**
    - We are allowed to use Python and Flask for the backend, along with JS, CSS, HTML for frontend
- **Sprints**
    - When we begin, we'll have to work in sprints (we can do weekly or biweekly, but Kashish recommends weekly)
    - We need to have a sprint retrospective at the end of each call to go in our repo & submit to Kashish – should include what each person did for the week, how sprint went, and what next week will look like
- **Project logistics**
    - Be careful to minimize scope
    - Project is 40% of grade and process is 60%, we need to prioritize following process correctly
    - For design process, think about perspective of user
    - There should be at least six ADRs, but more are better – do them whenever we make a large decision