**[Theo - Instructor](https://app.slack.com/team/U02H1CZ1LLT" \t "_blank)**  [9:47 PM](https://ricevirtdatap-nc12469.slack.com/archives/C02HKNKUG48/p1647370069498219)

**@channel** please read and re-read Justin's post before tomorrow as it will set the aim for group work and best-using class time (we've seen a few projects at this point and know what makes groups sink or swim, and we want everyone to swim, swimmingly!!)

* A note about flowcharts, prototypes, and digital collaboration:
  + <https://www.lucidchart.com/> - A favorite **collaborative flowchart tool**. Spending Wednesday as a group architecting your app
    - **SPECIFICALLY,** ex: How does a user get to and interact with the frontend?
    - Where does frontend live?
    - Where does the backend live and what routes are exposed for the frontend to call?
    - What parameters are required by these calls, and which are optional?
    - What is the data structure of a successful response?
    - What is the database schema?
    - What pre-processing is needed to get data in that form?

**Literal** answers to these should be represented in that diagram and putting the work in upfront will take away most 'he said, she said...' arguments about architecture and roles and plugging code together in the weeks to follow:

* + Creating a **great artifact to include in README.md**).
  + Also, a fair interview question is "architect this app" and be given a flowcharting tool to represent user input <-> software response.
  + Start with the user!
    - A favorite **interactive prototyping tool**.
    - **This is extra credit** as the tool is a whole UI/UX career by itself but can really help in quickly constructing prototypes that can be clicked on and getting a feel for how app components will look on different screens without continually recoding .css.
    - Laptop only view is fine for these projects,
      * But be intentional about how that screen space is used.
  + Also exports snippets of frontend code which can make the actual coding faster for people that don't yet think in frontend elements <https://miro.com/> - A favorite **interactive whiteboard**.
    - Fantastic for brainstorming as a team with stickynotes and has scheduling and flowchart capabilities built-in as well.
    - <https://trello.com/> - A favorite **task card manager** for assigning and owning development tasks.
      * Pro development teams work from cards
        + hold tasks
        + assigned to owners
        + drag the cards across swim lanes (ex: to-do -> in progress -> ready to test -> done)
        + teams can comment on cards and can integrate with GitHub branches as well.
* None of these are specifically required
* We don't care if your architecture diagram is crayon as long as it exists and describes how your app is wired together
* Nor is this a comprehensive list of all tools available to help you collaborate and build great software quickly.

Feel free to add your favorites as well and we look forward to seeing some great work!!

**Presentation:**

We will **present** this **Wednesday**, one week from today.

* **Aim for ~10 mins per group** (2-3 mins talking per member).
* A few things to note:
  + **Every group member must present** novel content (not just introducing other members) **on camera**
  + **The focus should be on question(s) that are being explored as well as the wiring diagram of how data will flow to a user of your visualization app.**
  + The **most important deliverable for this segment is the README.md** in your repository which should answer all high-level questions about your project aim in an intuitive and readable way.
    - Anywhere that implies 'connected' does not need to be live wired, as long as there is a clear explanation of the data schema and how it will flow to the model and frontend.
    - Machine learning requirement for this segment is specifying what is **[X] => [Y]** for your topic.
    - You should spend the 2nd half of Monday practicing your presentations
      * Even one or two dry runs of presenting will help the fluency of your live presentation and keep you on time (naturally, you'll talk longer than you think you'll need for the material without practicing delivering it.
      * **I'll start making animal noises to rush you at the 11–12-minute mark, so be concise!)**

Looking forward to awesome work **@channel** :) (edited)