Questions for Julio/Dr. Cook:

* [Request access to existing resources (code base, WFs Julio spoke about, presentation document - had more info than I was able to copy down and might be a good resource)]
  + **See document he shared with us via e-mail.**
* You mentioned building instructor data pages (w/ infographics, diagrams), support for different data types of reactions, and running user tests on the finished data pages during your presentation.
  + Can we discuss these features further to get a better idea of your needs?
  + How would you prioritize these features?
  + What are your expectations for our group, and how will you measure success in terms of their completion?
  + Are there additional features that weren't covered during your presentation?
* Schedule weekly meeting
  + **~~Tuesday after Capstone class, before 7:00pm potentially?~~**
  + **Created Discord Server for general communication with Julio and to hold weekly meetings on Tuesday afternoons between 4:05-5:30; they’ll be ~30 min each.**
* Coordinate preferred means of communication, and how available/willing he'll be to provide support in between meeting
  + Would you prefer to field any questions/clarifications as they arise, or does it work better for you if we consolidate them to ask periodically?
  + **Discord is ideal for convenient, fastest response times; e-mail secondary option.**
* What browsers will need to be supported? All? Chromium? For example, I’ve noticed that some interactive apps like wileyplus doesn’t work on chromium browsers when accessed through Linux but does on the same browsers (all most recent builds) from Windows?
  + OS/browser combinations we’ll need to test for?
    - **Preferably all of the browsers:**
    - **Julio’s experience - edge styling is worst; but don’t worry about it too much**
    - **“Just focus on the data tab”**
    - **Current bug is minimizing window logs you out; if we finish everything else, we might be able to address some of these existing bugs but they’re not a priority**
    - **Also doesn’t look great on mobile, but that’s not something we need to focus on unless there’s plenty of extra time - that’s more of a distant goal**
* Can you provide any more insight into the testing methodologies/tools that we should apply to this project
  + **Last semester: testing with other people, made click-through demo first and give it to another student in class (someone fresh to site) to see if they struggled navigating/with ui elements - this was a requirement for them for each sprint last semester**

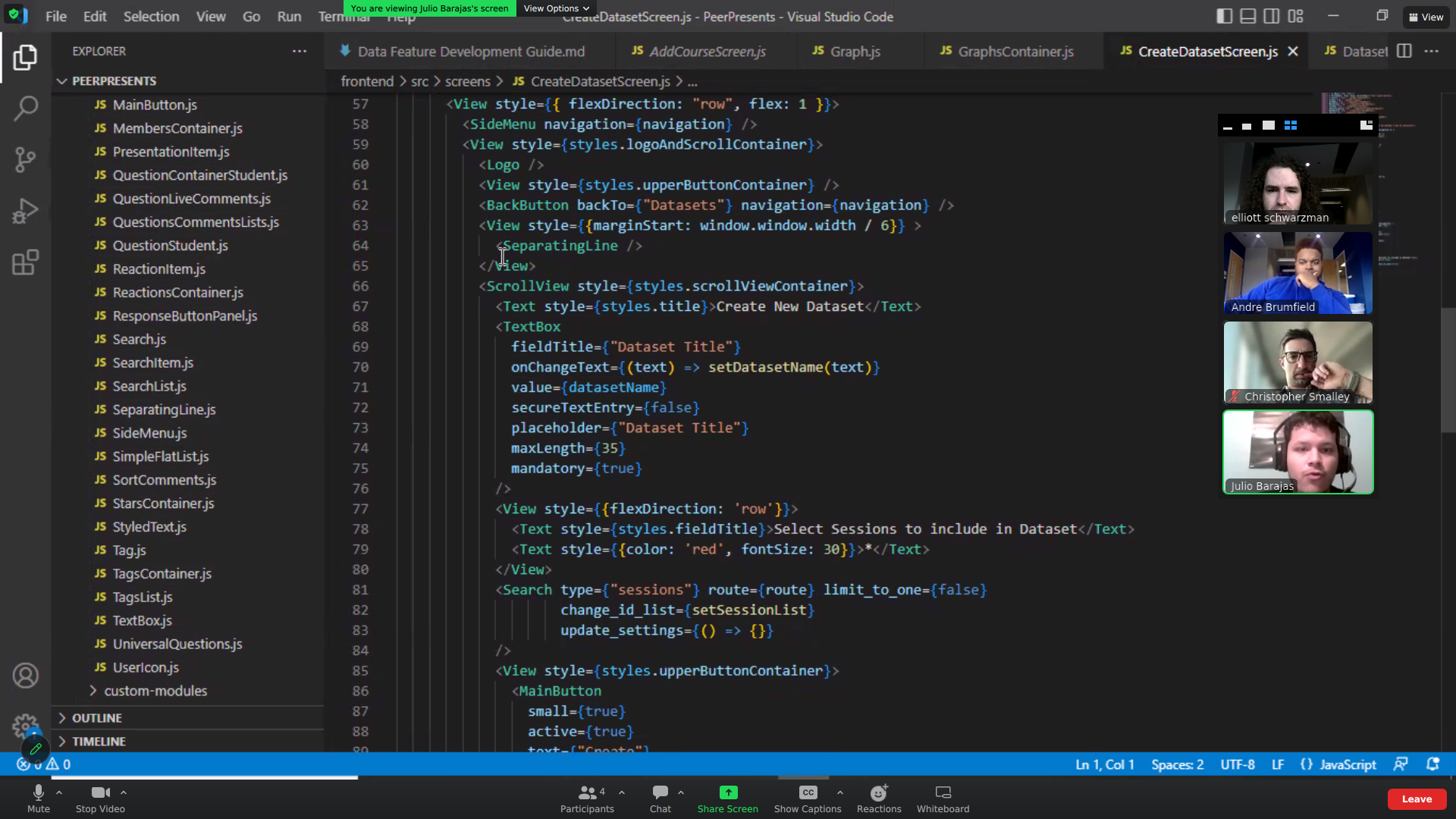
Try to be set up locally by next week

Notes from the meeting:

* Discord is the best way for fast turnaround time in communications
* For Tasks: PP already has GH page; “we might want to use GH Projects for Tasks”
* GH workflow:
  + Follow guide on naming branches
  + See local development guide on GH main page/readme for
  + Clone dev branch from main repo - it’s the most up-to-date version
    - Has last semester’s documentation
    - Check out the /Peerpresents/docs/Capstone Fall 2022/ directory first before setting up/when setting up local dev env.
  + **When making pull requests make sure we have it set to push to dev branch, not the master branch**
* Weekly code review meetings:
  + Julio checks our code!
  + No particular expectations on amount of code/progress; he’s flexible as long as we’re moving forward
    - Check what branches everyone’s working on
    - Provide general help/answer problems
    - Opportunity to make suggestions or changes to existing code - would be something to bring up
* See Recommended Next Tasks for immediate attention
  + Setup: install these things
    - Required:
      * Node
      * Expo
      * Repo (git clone)
      * 2 files, place in specified dirs (e-mailed directly to us, keep private)
        + Api\_conf
        + Ppdb\_conf
  + Running it 3x terminals: and run >npm start for each
    - cd to PeerPresents/storage/config
    - cd to PeerPresents/backend\_worker/config
    - cd to PeerPresents/frontend/src
* Expect invites to:
  + ~~Github~~
  + Figma
  + ~~2x config files~~
  + Setup admin accounts
    - Need a code to make instructor accounts (live website?) probably last step; need local env working first; update: not live, locally

Wireframes/Designs:

* Existing Figma: basically looks like actual website
* Expect invite to figma; ours will be a separate copy, so we can make changes as necessary
* We’ll be working on data page/data wireframes (see the section with graphs on figma)
* Model diagram in Lucid Spark (we’ll need accounts to view it)
  + UserSchema attributes in db for user accounts
  + Arrows point to show how backend connections DatasetSchema
  + “For every reaction there’s an upvote schema”
  + Won’t need to make anything in db
  + Call attributes whenever you need to use them
  + Star schema is in an array - response schema - basically these are comments. Unique id for star schema will be in response schema array
    - Same for upvote/react/star/tag\_list
    - “Just reference the schema”
* Basics are Figma WFs and Lucid Spark for model diagrams
* Code:
  + Layout:
    - Front end
    - Back end layout(?)
    - Back end master
  + Can see attributes under /storage/app/schema/data\_models.js (MongoDB)
    - Shouldn’t have to add anything to db
  + Re: frontend file
    - Contains all of the components which makes up everything you’ll see on the website
    - Ex: CourseCards, Logo, \*.js
    - Mainly working in frontend/components and /screens
    - There’s already a CreateDatasetScreen used to call the components
    - See screenshot: ScrollView which is a default react component which works the same way as if we made our own component in this directory
      * … extends Component {}



* + - Shouldn’t need to add any new components or screens
    - Note: already a graph screen
    - Workflow: visiting existing screens and adding components to them; generally ad them to the code and then visit the website (“play with the code”)
    - Note: existing guide on the github for our resources
  + Key files for us Graphs component, Graphs Container, DatasetScreen, DataTab; he set up a basic doc that covers most of this
* Some calls to MongoDB to call data and display it on the graph