Project Description:

- Using Datajoint to create tables and a schema.
- Using Flask as a RESTful API server.
- Using React to display the data.
- I was able to easily access the database using Datajoint and I was able to easily access tables and queries using Flask because both Datajoint and Flask use Python.
- The Flask API worked well with React.

How to run flaskAPI_Datajoint.py (Datajoint data definition and Flask API)

- I used the command line on macosx and within the same folder as the file, i was able to run the script by entering “sudo python3 flaskAPI_Datajoint.py” and entering my password to my computer.

How to run React web app

- From within react-subject-datajoint folder run npm start
    - If necessary run “npm install” to install missing dependencies

How to run admin.py (Used to clear data for demo)

- From within the same folder as the file, run “python3 admin.py”

How to clear tables for demo

- Run the admin.py script and make sure that the drop() commands are uncommented and in proper order.

How to load tables

- Run the flaskAPI_Datajoint.py script and this will connect to database, create schema, load table and populate them with given data.

How to load custom data into tables

- Use fields with appropriate description. I wasn’t able to get dynamic loading to work so for you to see the data you entered you would have to manually refresh the page.

How to insert data into Session table

- Session date could be whatever you would like but both name and experiment session (set up) have to already exist in Subject and Experiment Setup tables respectively.