- Google Login
    - Admin login works
- FAAPlatform name should be changed by deployment
    - Not priority
- Tuong got deleted
- Implement Drop down menu for roles addition
    - Done
- Recheck edit route path
- Table for admin
    - Alternating color between rows of table
    - Sorting and searching
    - Filtering by diff role types
- Focus on analytics (main idea) or quality of life smaller details
    - Client: don't care about what you do, give me product
    - I2-4 = analytics
    - i5 = quality of life, polish everything up
- Create a bunch of synthetic data
- What's up next:
    - Finish login, redirect, and role checking
    - Add feature of video storage
    - Add ability for metadata
    - Make lofi user mockup for all these story point ups\
- Practice management system
    - Potential name?
- Returned video
    - Sends time stamps for each play
    - Can choose player and see the routes they ran
        - Click on that play/route and see that clip
- Non relational database?
    - Prof says its possible, use nosql
    - Still need a relational database
        - For users
    - Database for practice data can be nosql
    - As client, dont care
    - As prof, be SURE how you want to progress with this
    - Code should work no matter what kind of database it is
    - Controller and models shouldn't care or know what kind of database you're using
    - Backend adapter that adapts Ruby database calls to whatever database you're using

- MVC
    - Nothing about MVC knows anything about database
    - Model vs Database
        - Model is not the database
        - Model is the object / in data representation
    - Using rails, you can make a bunch of models via auto gen
        - However models are empty
        - No logic, view, models, or controllers,
        - Rails will take care of mapping, don't worry about how the database connects
    - **At what point does the model interact with the database?**
        - **Rails handles marshall/serializing and unmarshal/deserializing of model to database and back**
        - **Object memory is rewritten by ruby to and from database**
        - **Gem of database IS the adapter**
        - In that case
            - Use relational database
            - Worse for optimization, better for programmers
- With rails, if you use a non-relational database, you won't use activerecord, but it'll work the same?
    - Go try it with a few example code and show ritchey if we want to use a non-relational database
- Final Notes
    - Handoff
        - Make sure we can transfer ownership easily and securely
    - Make sure there's always at least one admin
    - Make sure two accounts can't have the same email?
        - So that there's no conflict of a coach user and admin user having the same email