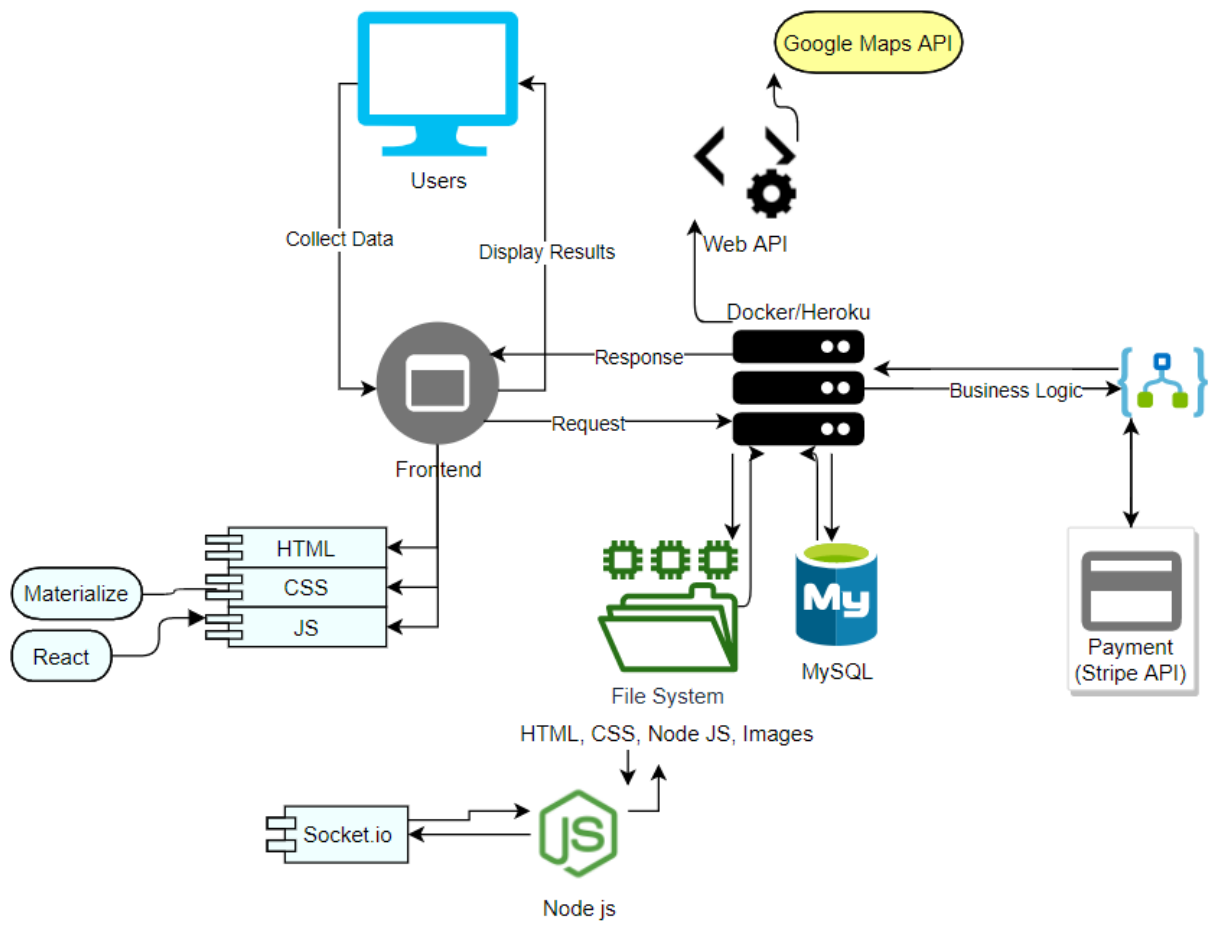


Questions:

- What features were completed?
 - The login page
 - Registration page
 - The feed where tasks appear
 - The company profile
 - Create Tasks
 - Rate Worker
 - Display review component
 - Display rating
- What worked during the demo?
 - The drop down menu
 - All buttons that are linked to pages
 - Feed
 - Login Page
 - Registration page
 - Company Profile
- What issues were faced either during the development or during the demo?
 - Most of the team members had a hard time learning to use react js
 - Getting our database setup
 - Did not understand how to use node js with react js
- What were the suggestions offered by the TA?
 -
- Individual contributions by each team member
 - Registration Page,LoginV2 ,Drop down menu (Manoj)
 - Created a reusable task component created a feed where the tasks are displayed (Tim)
 - Company profile, create tasks, rating workers, user review, view rating, workspace, display tasks in detail, high-level architectural diagram (Josh)
 - Logo, profile page and edit profile (Jules)
 - nodeJS API for database access (Gregory)
 - Landing page, nav to login and register (Ryan)



Features:

1. *Figure A.* We have a working MySQL database with some initial seed data. Scripts for the set up are here:
https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/tree/main/data
2. *Figure B.* A nodeJS REST API has several routes implemented. This facilitates database access, and returns JSON data. We have working routes for **create user**, **update user**, **get a user by id**, and for **create task**, **update task**, **get a task by id**, or **get a list of all tasks**. This last one can use a query string to filter type and or status. Code for the api, along with a readme documenting routes, is here:
https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/tree/main/code/api
3. *Figure C.* There is a special login route which returns no body, but instead a jwt cookie.

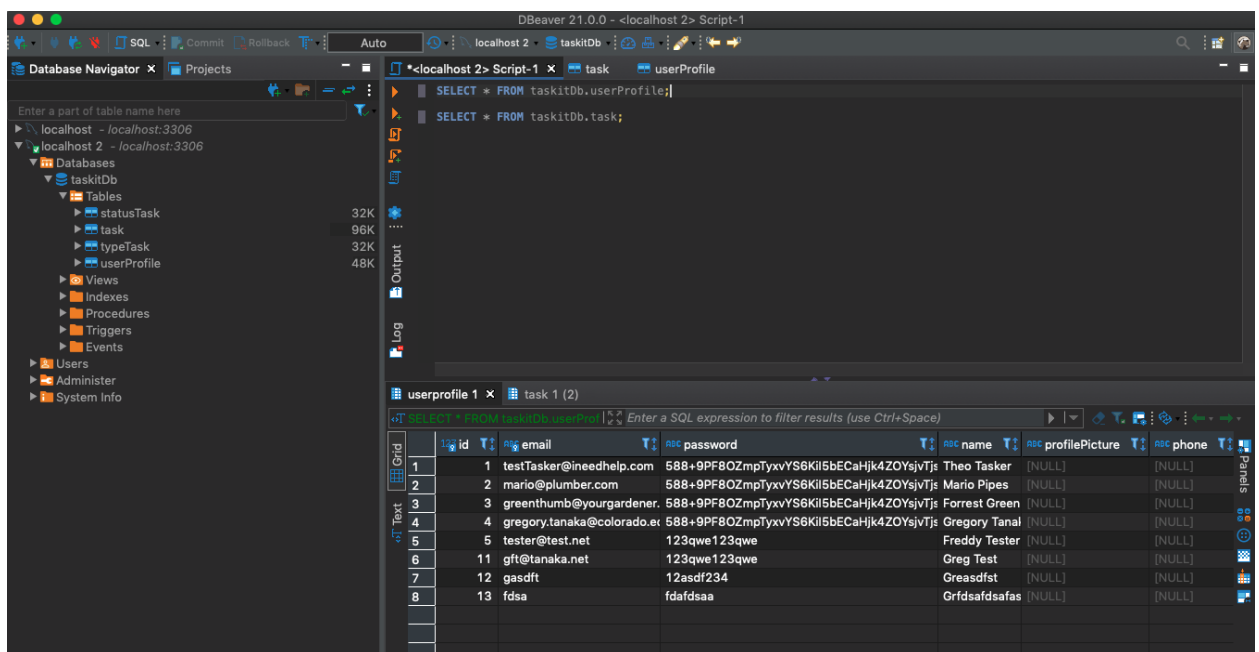


Figure A. Database example

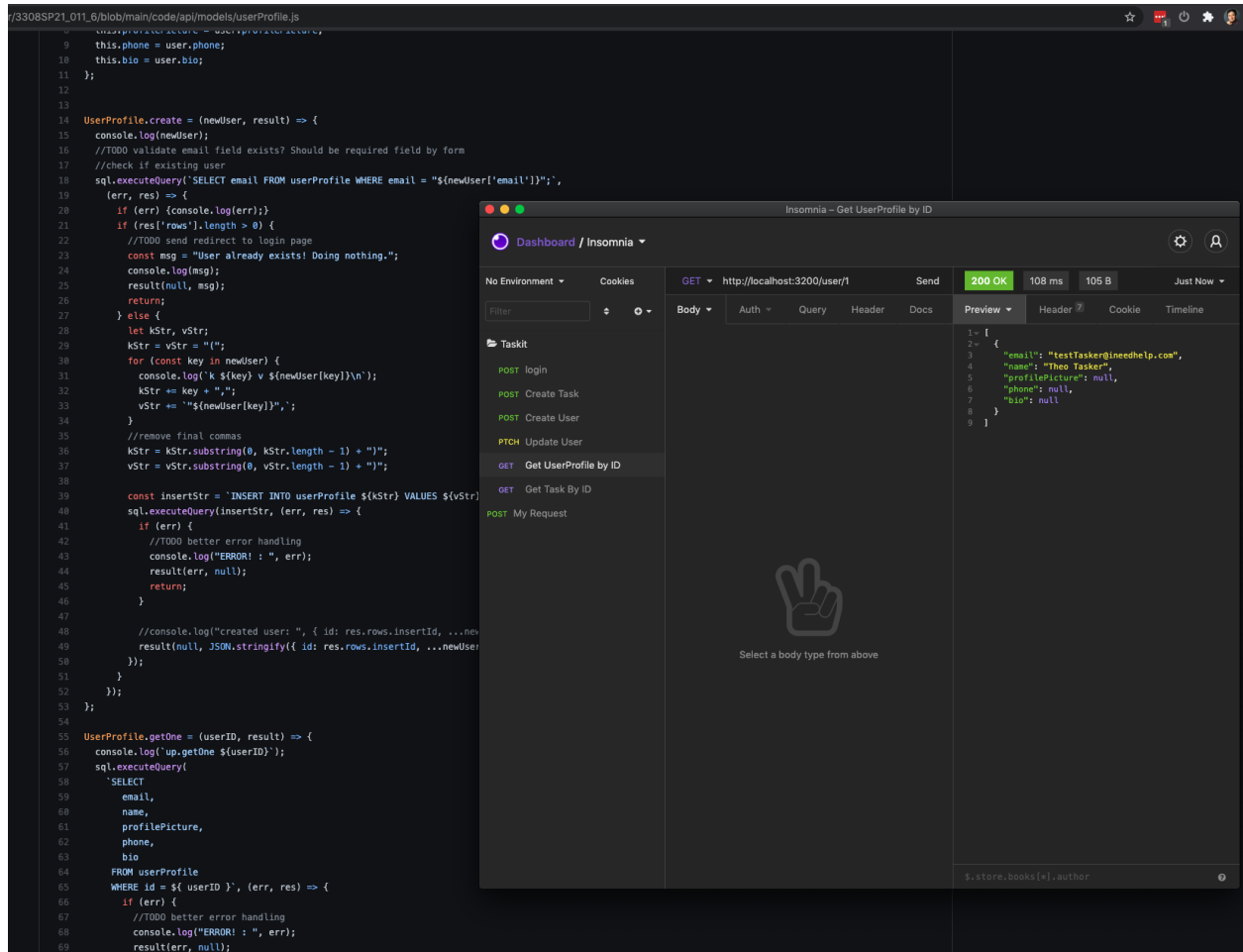


Figure B. Example of route in use through Insomnia rest client, code in background. This demonstrates the `UserProfile.getOne` route.

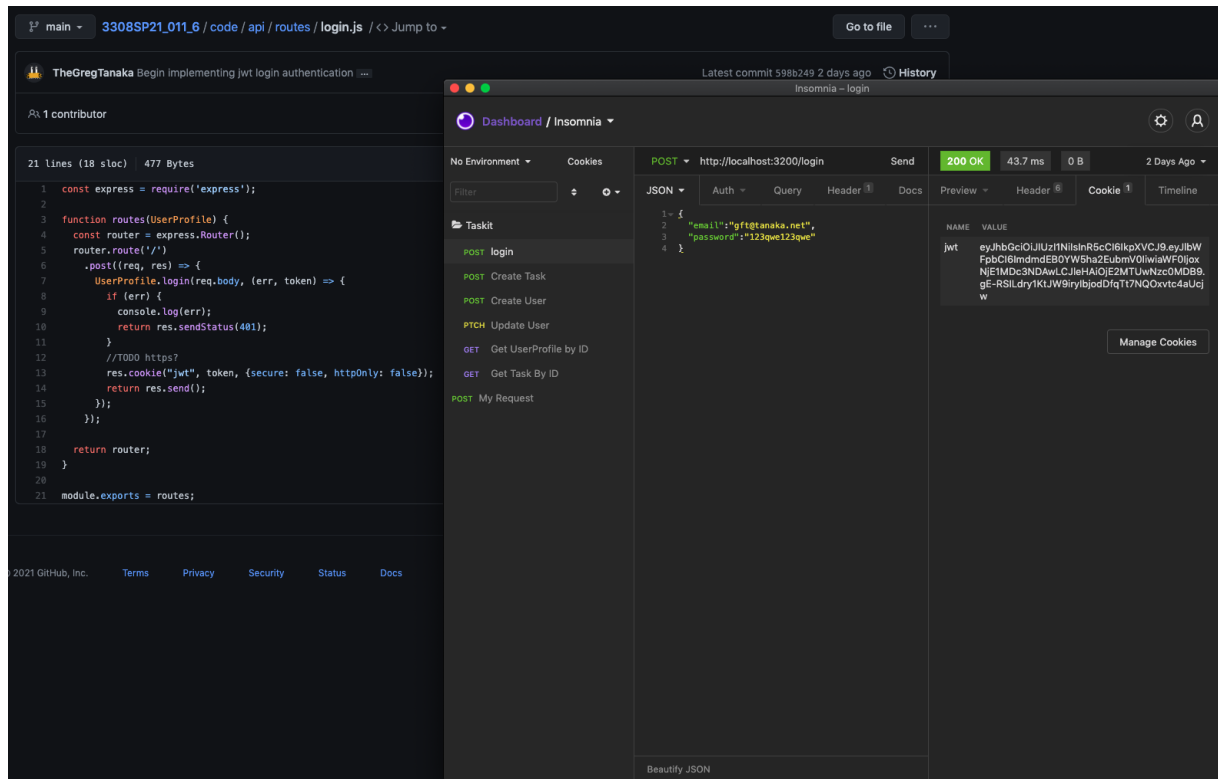


Figure C. Example of login route returning jwt cookie. Code in background.

Greg's note for 1-3, I've been working on getting functionality first, and will clean up the code base as well as adding validation and improved error handling later. I wanted to start with the basics and then refine later on.