Just in case anyone has a hard time finding the repo:
https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_g8y9a_h8r1b_r0a3b

Please feel free to add to the timeline/tasks below

Timeline/Tasks

Nov 6 - 10: Building database (SQL DDL and INSERT statements)
- Create necessary tables (Colton)
- Populate tables (Kevin & Amy)

Nov 11 - 18: Writing queries (10 operations) - implement queries for the following
- Projection (Colton)
- Join (Colton)
- Aggregation with Group By (Kevin)
- Aggregation with Having (Amy)
- Nested Aggregation with Group By (Amy)
- Division (Colton)
- Insertion: (Kevin)
    - Add users to user table
    - Add recruiters to recruiter table
    - Add messages to message table
    - Add work experiences to workExperience table
    - Add companies to company table
    - Add cities (and their associated postal code) to municipality table
    - Add provinces (and their associated postal code) to district table
    - Add hashtags to hashtag table
    - Add job postings to jobPost table
    - Add job descriptions to jobDesc table
    - Add inbox ID and associated message ID to contains table
    - Add hashtag and their associated posting ID to hasHashtag table
    - Add job posting id and their associated company name to hiringCompany table
    - Add company name and associated ID to workedAt table
    - Add job posting id and user id to apply table
    - Add message id and recipient user id to receivedBy table

- Delete: (Amy)
    - Remove users from user table
    - Remove recruiters from recruiter table
    - Remove messages from message table
    - Remove work experiences from workExperience table
    - Remove companies from company table
    - Remove cities (and their associated postal code) from municipality table
    - Remove provinces (and their associated postal code) from district table

- Remove hashtags from hashtag table
- Remove job postings from jobPost table
- Remove job descriptions from jobDesc table
- Remove inbox ID and associated message ID from contains table on update and on delete
- Remove hashtag and their associated posting ID from hasHashtag table on update and on delete
- Remove job posting id and their associated company name from hiringCompany table on update and on delete
- Remove company name and associated ID from workedAt table on update and on delete
- Remove job posting id and user id from apply table on update and on delete
- Remove message id and recipient user id from receivedBy table on update and on delete
- Update: (Kevin)
    - Update a user's age
    - Update a recruiter's pass end date (in case they decide to renew)
    - Update a work experience's start and end date
    - Update a company's name and postal code
    - Update a job posting's title and body
    - Update base salary in job description
    - Update start date in worked at table
- Select: (Amy)
    - Companies can select users who've worked at certain companies before in the past
    - Users can select job postings from a certain company
    - Users can select job postings by position title
    - Users can select job postings by companies' location (province and city)
    - Companies/users/recruiters can select users based on name
    - Users can select job postings by hashtags
    - Companies can select users by age (have to be older than 19 to work at a bar for example)

Nov 18 - 25: GUI
- Create a GUI (PHP)
    - You can create an account (as a recruiter or a regular user) and fill out associated information (name, create new work experience, age, etc) (Colton)
    - Create an account as a company (Amy)
    - Can log in to an account (Colton)
    - Can see other accounts (Colton)
    - Can see an inbox with messages (Kevin)
    - Can send messages to other users (Kevin)
    - Explore job postings page (filter jobs by different category) (Amy)

- Make job postings as company/recruiters with associated attributes (company, base pay, position, description, etc) (Amy)

Nov 25th: Milestone 4 - Code Implementation Due Date