1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

Our execution of the project is very similar to what we initially planned. There are very few changes we made to our initial proposal. Our checklist and group features are all fully functional, each implemented well and in conjunction with our SQL queries.

2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.

In terms of usefulness, our application achieved exactly what we hoped to achieve. We have a functional checklist along with a group mechanism where each user can join many groups each with their own selection of users. Through this social checklisting and alerting system, we are able to provide a more productive environment centered around finishing tasks instead of abandoning them.

3. Discuss if you changed the schema or source of the data for your application

The Schema of our application is very similar if not the same as the schema that we initially had planned in our earlier steps. The source of the data was initially to be all user generated in a fully operational context, but we added our custom generated template data for proof of concept and demonstrations.

4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

In the original design, we had the Alert entity connected to the User table. This was to simply represent who was pinging whom. However, we changed that to a User creates an Alert for a certain Task. This was because instead of generally pinging a User in your friend group to be productive, it would be much more productive to ping which task the User should focus on. This provides more purpose and clarity to the Alert system, thus is a more suitable design.

5. Discuss what functionalities you added or removed. Why?

Most of the functionalities that we removed were peripheral components. For example, the profile page changed from initially being a comprehensive report of the user (gender, age, etc) to a page representing the important information only (e.g. completion rate). This was because in order to implement the core functionality, we were unable to add in these details.

6. Explain how you think your advanced database programs complement your application.

Our advanced database program abstracts the database from the middleware. Every time we complete/abandon a task, we need to calculate the completion rate of the task. This requires many SQL operations: get the count of each task, calculate and update new completion rate, and delete alerts associated with the task. On top of that, if we were to delete a checklist, the associated challenge of dealing with the tasks inside

became hard. As further explained in 7, multiple queries to the database is not possible, so Stored Procedures helped make this task into one query. This abstracts the middleware as it only does simple read and write tasks instead.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

Riki: One technical challenge that I faced was not being able to do multiple queries in one SQL statement. For example, “INSERT INTO Alert VALUES (0, “Message”, 0); SELECT message FROM Alert WHERE taskId=0;” has to be broken up and queried twice. This is apparently a security measure to avoid SQL injections. When we query, we essentially make an asynchronous operation, so two parallel queries are impossible. Then, the only solution becomes layering one call into the other call to guarantee order, which is not the best practice. The best solutions to this are to either create Stored Procedures or to create multiple routes which each do a separate SQL task.
Jessica: One technical challenge was creating indices that actually helped enhance the advanced query performance as we did not have too much data populated at the time and since many of our tables already had primary keys that was sufficient enough for our queries while adding more indices took more time. Another one was embedded javascript templates as that was my first time using that to inject data into the html template.
Shlok: One technical challenge I had was working with GCP. Creating and working along with hosting our project on GCP was a challenge. In our id.json file we stored the latest indices of our row which we auto increment every time we added to a table. However, this approach was difficult to use as if multiple “users” used our site it didn’t work. One way we went around this problem was by changing this object on every instance of our users and making sure that object was shared across all users.

8. Are there other things that changed comparing the final application with the original proposal?

One important change is the UI compared to our original low fidelity UI mockup in the proposal. If we look at the project now, each aspect of the project has a fully functioning, robust interface, and looks much cleaner than the low-fidelity mockup. The mockup centered around the profile and checklist, but we have of course taken that style and expanded it to neatly cover all the aspects described in the proposal. For example, the profile now focuses on the Alerts you received instead of the user information as described in the mockup.

9. Describe future work that you think, other than the interface, that the application can improve on

Along with the UI, there are other features we wanted to be able to implement. In our application, we wanted the ability for users to be able to view and comment on their group peer's checklist. The problems we had with this is that this would allow the user to also have editing permissions so figuring out the permissions would be a challenge.

10. Describe the final division of labor and how well you managed teamwork.

Our group worked on the project together over Discord for every checkpoint. One of our group members shared their screen while we discussed and worked on the project collaboratively.