Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission). Discuss what you think your application achieved or failed to achieve regarding its usefulness.

Generally, the core of the project is the same. We were able to create an app in which users could create workout plans as well as user other people's workout plans. For our first proposal, out purpose was well defined but our UI and specific implementations weren't. Throughout the project we went through many frontend iterations to make the design more streamlined and intuitive to the user while still staying consistent with our original mission.

Overall, we are happy that our app was able to meet our original goal of allowing a user to make and share plans. We're glad that we were able to make the creating plan process intuitive, and the display of how your plan is being built creates for a better user experience. In terms of the sharing usefulness, we're glad that users can use other people's plans, but it would have been better if they were able to search by attributes such as the creator's name, or popularity of the plan, or to reuse components as originally envisioned.

Discuss if you changed the schema or source of the data for your application. Discuss what you change to your ER diagram and/or your table implementations.

As discussed in prior stages and their revisions, some changes to the schema and data sources were made then – in particular to better fulfill the requirements of Stage 3.

We didn't make any big changes to our ER diagram during Stage 4. We did have to adjust a few of the FK constraints, and also added auto-incrementing to the IDs on most tables in order to avoid having to manually manage making unique IDs. In previous stages we majorly changed our ER diagram according to our TA's guidance so that we may better model the real world. In Stage 4, we added a BMI attribute to the user table as it needed to be used in our Trigger, other than that, we didn't make major changes.

What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

The original design was not deeply thought out as we did not know what data we would need to complete our app. We ended up adding Login and Register pages that allowed the creation of users, and the storing of the logged in user's information throughout the app's pages. We also decided to split certain pages like the Find Plan and Create Plan because together they created a messy UI and a difficult user interface to navigate. We decided that splitting them would make it easier to understand while still maintaining the same functionality.

Discuss what functionalities you added or removed. Why? Explain how you think your advanced database programs complement your application. 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. Are there other things that changed comparing the final application with the original proposal?

On the dashboard, we initially wanted to give the user the opportunity to edit all their information, but we later realized that some data cannot change so that it stays consistent with our constraints and reality. For example, the user cannot change BMI, because it is not dependent on subjective opinion, but rather a calculation from the user's height and weight. User_id also cannot be changed by the user because it needs to be unique from other User_ids. Our advanced database programs such as our triggers allow us to keep our data consistent and up to date. We used constraints to maintain relationships between tables as well as ensuring that all our data would always be valid, for example, the primary keys had to be unique, and certain values such as weight couldn't be negative.

Technical challenges-
Kaz: The trouble of modifying just one table was really hard since it required me to make changes on other tables that have dependencies on it. My advice would be to test code in as small chunks as possible and add debug statements before starting.
Sona: Integrating the SQL queries into the backend code to return the data for user and plans data. My advice would be to double check your data types and make sure that they are being parsed correctly.
Sana: Storing the user id so that it can be accessed from any part of the app, as well as unique to the user logged in proved to be difficult. My suggestion would be to always double check the names of your variables and make sure they are consistent throughout your app because it could save you from many unnecessary bugs.
Nathan: Kept receiving 404 errors when the code seemed correct and the console log statements seemed otherwise OK. The issue was a crashed server, so my advice would always to double check your connections before getting stuck on the same lines of code for hours.

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

Future work could involve creating places where users can rate and suggest plans. This would allow us to expand our functionality including algorithmized suggestions based on trending plans that align with the user's goals. Other functionality could include being able to copy and edit other people's plan to make it your own, or completing plans with a buddy. It would also be nice to track your effort on each day of completing your plan. As you can see, the future functionality is as endless as the weight loss industry.

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

Teamwork was overall good. As busy CS students, it was often hard to find times where we could all meetup. This slowed our progress at the beginning, but we eventually figured it out by meeting two or three at a time and staying consistently available on discord.

Division of labor is hard to quantify because the team members of our group had different strengths. We all helped each other review and complete our respective portions. Sona and Kaz first worked on a skeleton frontend, where we all took a page that had CRUD functionality and an advanced database feature. Sona started on the Login and Register page than Sana later took over to add more attributes, constraints, and revise the authentication. Sana started on the dashboard which Kaz and Nathan helped guide her on React and SQL components they were more familiar with. Nathan started on the Find Plan which Kaz later completed. Nathan then went to work on more advanced database features. Kaz started on the Create Plan page which he and Sona then worked together to complete.