- 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).

We added a function that finds out the food that was consumed by a user that has the most calories.
The overall direction of our final project is the same compared with our original proposal. We created a product that tracks the user's daily calorie intake and exercise levels depending on user input, while also generating a weekly report based on the user's performance from the previous week.

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

Achievement:

We developed core features to monitor users' daily calorie consumption and activity levels based on their input, while also producing a weekly summary reflecting their progress from the prior week.

We offered details on various exercises, as well as food and beverage options, retrieved from specified datasets. Each food and drink entry includes calorie information, enabling users to track their calorie intake effectively.

Failed:
We failed to implement the AI API to our program. Instead, we calculated the average calories consumed by the user to provide some insights into the user's habits.

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

Most of the schema is the same as our proposal. However, we added a column TotalCalories inside Drink and Food table to track the calories consumed based on the drink/food's calories per gram and quantity consumed by the user. We also removed our Prompt Table since we failed to implement the AI feature for our database.

- 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?

We fixed some previous errors in our ER diagram about the relationship between tables. We added a TotalCalories field in Drink and Food table, because we find having an extra field that calculates the calories for the specific category of food useful when inserting

the meal information. We removed the Prompt table since we abandoned the AI feature in our app. If we have extra time to improve our database, we could add foodId and drinkId under the Meal table, where both of them FK to their original table Food or Drink. An additional column of total calories consumed in one day could also be added to the Meal table.

- Discuss what functionalities you added or removed. Why?

We added a function that shows a “daily quote” on our app to enable our user to learn a quote each day. This inspires and motivates users, encouraging them to learn something new and start each day with a positive thought.

We removed the function that suggests a workout plan by AI due to its limited customization and alignment with user-specific fitness goals, ensuring the app remains user-focused and avoids providing generic or potentially unsuitable recommendations.

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

Our advanced database features complement our application by achieving efficient storage, retrieval, and management of user data.
We have implemented CRUD operations to our database. So users could manage their data, such as adding, viewing, editing, and deleting meal logs, exercises, and progress records. This flexibility ensures personalized tracking and updates, empowering users to maintain accurate and up-to-date information to achieve their health and fitness goals.

We have implemented transactions that Identify Underactive Users Who Haven't Logged Meals or Exercises in the Last Week and remove the users. This could help us maintain the server by deleting inactive users. We also calculate total Weekly Calorie Intake for a User to let the user know their status of calories consumption.

We also implemented stored procedures that help provide ready-to-use functions. We provide a detailed summary of total time spent on each type of exercise for a specific user. It also handles multiple exercises for a user dynamically, regardless of the number of exercise types. We Identify the food item contributing the most calories to a user's diet in the past week and help users or nutritionists monitor and manage calorie intake effectively. I also implemented the SP that calculates the total calories a user consumed in one week, which let the user have a sense of their eating habits.

For constraints part, all of our database table are linked together by FKs, such as MealId which links to both Food table and Drink table, or UserId that links to Report table and Meal table. We have also set correct primary keys or keysets to prevent error in our database.

- 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.

Ying Chen: I found implementing the trigger very challenging. The problem I encountered is that when there is a trigger that monitors whether a row is changed, it is hard to follow the constraints and try to edit the table since the trigger is always functioning. Therefore, when applying triggers to the database, make sure you understand what the trigger is doing and what influence it will have if you try to update the table. Otherwise, an error will pop out indicating that your query is in conflict with the trigger you made.

Yanze Lu: One technical challenge I encountered was optimizing SQL queries to improve performance, particularly for complex joins and filtering large datasets. Initially, some of our queries were taking a long time to execute due to redundant joins and missing indexes on frequently queried columns. This caused delays in our application's responsiveness. To address this, I had to analyze the database schema and query execution plans using tools like EXPLAIN. I learned that adding appropriate indexes and simplifying nested queries significantly improved performance. My advice for future teams is to always monitor query performance during development, use indexing wisely, and avoid overcomplicating joins or subqueries. This can save a lot of debugging time later and make your application more efficient.

Cobbi Liu: For me, since I was primarily working on backend, it was quite challenging to manage all the API routes in conjunction with the correct SQL queries. For example, DELETE and GET routes work in quite different ways, especially since they sometimes involve multiple joined SQL queries or different procedures, triggers, and transactions. Since different members worked on different parts, for example one member worked on procedures, sometimes there were conflicts in name. For example, one time we had a field in a procedure named Total_Calories while I was trying to do a post route for inserts on a field called CaloriesTotal, which took a while to debug as procedure fields are not explicitly shown in a database.

Denny Li: My main job is implementing the frontend. My main challenge was about ripping apart the parts of the homepage and make them into custumized components so that it is easier for the backend to connect and use the frontend work. There is a lot of components needed for out home page since the home page is the only page we have for the app. It needs to hold 8 different components in order to meet our need and design for the application. Every component will be connected to the backend in order to get the information that the user needed. They took me a lot of time to research and finish. In the end, I also need to work with the member that worked on backend to fix certain issues or additional UI that they needed.

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

The differences are basically covered in the previous sections. We removed one table, added two columns total calories, removed the AI suggestion section, added a daily quote section, and provided some calories-related database features that complements our product.

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

We could still try to integrate AI into our application, as it could provide suggestions on the food/exercise. It is also interesting to add a group system where users could create a group and see the calories intake by their family members or friends. It is encouraging for people who want to follow a healthier diet when there are other people who are working together.

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

Runzhong Li: Front-End design and workflow (CRUD basics).
Yichen Liu: Connection between front-end and back-end (server.js) and api routes from SQL.
Ying Chen, Yanze Lu: Database management, including procedures, triggers, transactions, constraints, as well as adding data and linking different tables

It has been a great semester working together. We collaborated well and even though we've distributed our works, we reach out to other members for help.