When implementing our final product, we stayed true to our original proposal, except for some minor changes in functionality and user experience. We changed the look of our application due to our time constraint, and we do not track how many calories that a workout burns. We also did not implement much regarding workout progress due to the database that we used.  None of our data sources changed from our proposal. Our primary changes from our proposal were to the UI and the number of/content on our pages, though we maintained the integrity of the proposed application.

Ultimately, we achieved the primary purpose of this application, which is to track a user's meals and workouts while providing them insights into their progress and recommendations to help them succeed. We display how many calories they've eaten, how their weight has progressed over time, and how close they are to their goals. We recommend foods based on these goals. However, as previously mentioned, we failed to provide much information/insights into their workouts. We planned to track their workout progress, show them how many calories they've burned, and personalize recommendations. However, the dataset we used didn't have information about how many calories each workout burns.

While implementing our application, we had to add a table called "WeightLogs" in order to provide a history of the user's weight. This slightly changed our ER diagram. Everything else was consistent with our proposal. Our more simple advanced queries are essential to the basic functionality of our application, such as calculating a user's daily calories. Our advanced database programs–e.g. recommending foods to the user's goal according to other users who have succeeded in the same goal, being able to search for food, and recommending top workouts–make our application more user-friendly and personalized. The keyword search for foods was a functionality we added onto our original proposal since scrolling through the list was too tedious. We also added the functionality of recommending foods, as we thought our application was lacking in personalization without it.

Although we achieved the core goals of our proposal, we definitely have a vision for future work that could improve our product. We want to customize meal plans more according to the user's goals, preferences, and dietary restrictions. We also want to recommend customized workout plans and provide information about the calories burned in their sessions. We would add a page for the history of their calories consumed so they can see how consistent they've been with their goals. Most importantly, we would utilize an API for login verification and connect their accounts to their emails so that they can receive motivational quotes, friendly reminders, and promotions.

The process of implementing this application was well-managed in terms of teamwork. Snigdha and Utkarsh worked on the backend while Viven and Adi worked on the frontend–a bit different from our proposal, but this is what fit our strengths and weaknesses best. Snigdha focused on queries while Utkarsh focused on the table creation and indexing analysis. Ultimately, there was collaboration on all sides when debugging and problem solving.

For example, Adi encountered port occupation issues when doing “node server.js” which was resolved by getting the PID of the current process running on that port and killing it. Viven faced a challenge where the “Delete Meal” functionality didn’t seem to be working. After further investigation, the issue was the naming convention; the same attribute had different names on the server side and the page-rendering side, which caused empty deletes. From this, we learned to maintain appropriate, detailed, and consistent variable names between the two sides.

Snigdha had an error fetching the results of the query that retrieves top workouts, but only upon the second loading of the page. This could only be remedied by logging out, then logging back in. However, looking at the corresponding stored procedure revealed that a temporary table was created but never deleted. Dropping this temporary table fixed the error, and we learned to always delete temporary tables when they are no longer necessary.

When first creating the database, Utkarsh was having trouble importing the external csv of workout data, and all of the data seemed to be shifted left; i.e. the information for column i was being displayed in column i-1. For this issue, if using GCP, we advise importing the data into a separate table that is formatted the same way as the original data, and then modifying this table into what is most useful for the application. To elaborate, import the csv with all of the columns, and *then* drop any unnecessary columns or rows, such as the column titles or, in our case, the workout description, workout rating description, etc.