1. Our final project closely adhered to the original proposal. Initially, we aimed to implement a creative feature involving game recommendations based on user preferences even if there weren't any games that fit the exact searches; however, we decided not to pursue this due to the fact that there would need to be a machine learning aspect with this and we were not versed enough for that. We also made changes to the user interface that we had originally designed. Initially, we had a welcome page from which you could go into different pages. In the actual application, we made the game search page as the opening page, from where you can go into different tabs such as the Game Recommender, Account Info, Friends, and Your Games.

2. Our application achieved its usefulness by helping users filter their games based on various criteria like multiplayer compatibility, price, genre, and platforms. We also allowed users to connect and add friends with other users that play the same game. Moreover, our application allows users to create their own personalized lists of their favorite games. Our application successfully implemented our initial intent, which was a user friendly game recommendation website.

3. One thing we changed about the schema was that originally, the values for whether the game was supported on a particular platform was reported as TRUE or FALSE. This resulted in an error when we inserted the tables into gcp as there was no output when selecting the columns. We figured out the problem was with the TRUE/FALSE columns; therefore, we wrote a script to convert those values from TRUE to 1 and FALSE to 0, which fixed this issue.

4. We had a few differences between the original ER diagram and the final design.
In the original ER diagram, we had the GameName, Rating, and TimePlayed in the User Recommendation table. We introduced an additional attribute to the User Recommendation table, enabling users to view others who recommend games when their own ratings surpass the average rating. In our original schema's computer information table, we had columns to keep track of the computer's specs such as GPU, CPU, RAM. In our final application, we ended up only filtering from the user's OS. During our implementation, we recognized the need to incorporate essential attributes that were not apparent during the original ER implementation. Our final design is more suitable than our original ER implementation.

5. We implemented a keyword-based game search functionality into the database. Originally, our intention was to offer game recommendations solely through filters; however, we realized that a truly useful website would have a search feature where any game can be manually searched and added. We did not remove any features that we originally planned to implement. As intended, we successfully incorporated the friends' adding feature, allowing users to identify recommended games from other users.

6. Our advanced database program compliments our application as it shows the highest rated games from a user’s friends list. This was one of the core features we wanted to add from the beginning as it gives incentive to the user to add friends and also review games. Through this advanced database, several advanced queries were used to get the friends, and we used another query to get the average score of the game. This helps decide what games are shown to the user as we only want to show the highest rated games a user’s friend played.

7.
Junu: With my limited experience with GCP, assessing and finding any significant change in cost variation tied to different attributes and choosing which columns to index on was particularly challenging since the column we chose to index on made no cost changes most of the time. Also devising a useful trigger that was relevant and impactful for our project was very challenging. Unlike in homework scenarios, where the trigger questions are already provided, we had to conceive and come up with a valid question that could be solved by using a trigger while also making it work properly.

Manu: Deploying the vm proved to be more challenging than I initially expected. Since there was a very detailed video going step by step on how to set up the vm, I expected it to go smoothly. But while setting it up, there were some unexpected struggles with installing all of the libraries we required to host the website. We tried many different commands but I believe I just failed to consider the fact that I was working on a vm which resulted in a lot of wasted time. An hour before the presentation, when trying to test our website one last time, we realized that the vm was unresponsive in our region, so our only option was to create a duplicate vm and set everything up again. But as the time to present approached, we realized we might have to forego the extra credit. However, just in time, the original vm started running again.

Afif: A technical challenge I faced was connecting our GCP sql instance effectively with the rest of our python code for running the web server. There are libraries that handle executing queries and google provides a specific library to connect to the instance however I had an issue with essentially validating/ giving permission to our python code to modify & query our GCP database. In order to do this I had to use GCP CLI and authenticate my local computer using that however I ran into issues downloading the CLI initially as I was working on a linux VM. The GCP CLI linux installation was broken and did not want to install so I switched to developing on windows where the GCP CLI has a much simpler installation process to authenticate my local computer. After this the code was able to run smoothly and access the database without any issues.

Rishi: A technical issue that occurred for me was when I was developing the advanced database program, more specifically the stored procedure for it. We wanted this procedure to get the rating of a game from one of the user's friends and compare it to the average rating of the game. We thought that if a friend rates a game higher than average, it is in their best interest to recommend that game. The issue came into play when comparing ratings as every game a friend rated higher than 0 was recommended. When looking at our application, the users can review a game from 0-100 but the average ratings end up being 0 or 1 for the game. This is because in the reviews dataset, the ratings were integers and ended up rounding up when aggregated. The way to fix this issue was to cast the average rating to a decimal using the CAST() function. We also did the same for our friend's rating by dividing the user rating by 100 after casting their rating to a decimal. This allowed for both ratings to be in the same range of 0-1 of each other which allows for a more accurate comparison.

8. We stuck to our original proposal as much as we could. However, as the project progressed, we made a few changes to the schemas to better suit the project and enhance the user experience. Overall, we still kept true to our original proposal.

9. Currently, the application can only be used by those with a link to our server. However, to actually make this a real life application, it could be beneficial to connect the website to the cloud, where anyone can access it. Furthermore, it could be a good idea to add a function to allow the user to vote on the game, and then have the option to sort based on the number of votes.

10. As a group, we worked very well together. There was constant communication with everyone as we had several meetings a week to go over what everyone was doing and discuss future plans. Everyone did what they were supposed to do and expectations were fulfilled. While we didn't set strict roles for everyone in the group, we all ended up focusing on things we were more experienced in.

Afif was responsible for the frontend design & web-server infrastructure implementation through python. He connected the backend work developed by the rest of the group to the user interface.

Rishi was mostly responsible for developing most stored procedures. Through discussions with the group, he was able to convert the ideas of what we were doing into stored procedures and was able to implement it into GCP.

Manu was responsible for setting up the vm we used to host the website and connecting it to github. He developed one of two advanced queries that was implemented to filter the game search.

Junu was responsible for database management by developing one of the advanced queries and performing index analysis through GCP. He also came up with a valid trigger and implemented it into GCP.