# Stage 6: Project Reflection Report

1. 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).

Comparing the final application to the original proposal, there were some slight differences that occurred. Originally, we planned on including data such as likes and comments in our original post, but later shifted our focus to other items such as categories, and tags. Moreover, the UI that we originally planned out looked different from the final application. Lastly, we were also unable to incorporate the creative component we had in our original proposal in our final application.

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

We believe that we were able to construct an application that would be useful to people wanting to collect more information and identify trends regarding the videos that are posted onto YouTube. Our database collected a number of various items including channels, videos, tags, and categories. The database also was unique in the fact that it allowed the user of the application to modify the database themselves by either adding, updating, or deleting videos. Moreover, we were able to construct a couple of advanced queries that allowed users to identify which categories the trending YouTube videos reside in. This could help content creators identify which categories of videos are trending more than others. For what we could have improved on, many of the features of our application were quantitative in nature, and we were unable to add any visuals to our application such as graphs that could have explained our data in another format. This is something we could further implement in the future.

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

For both schema and source, we did not change anything. We used schema that we originally planned to use and also used the YouTube trending video dataset that was provided by the list of datasets provided by the class.

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 our ER diagram, there originally was a HasTag table that checked if a video had a tag or not. However, In our final design, we removed the HasTag table as we figured out that every video in our dataset has at least one tag, meaning HasTag table has no use. I think the final design is more suitable for our project because there is no need for the server to spend extra time on whether the video has a tag or not. We are able to directly access the tags that each video has.

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

- For CRUD operation, we first had DELETE operation for user log-in information. However, we received a feedback that we lacked interactions on other tables, so we deleted it and added DELETE operation for a particular video in the video table.
- We added separate pages for each operations, advanced queries, and stored procedures so that the website is more useful and looks organized when the user interacts with the website.
- Since we were trying to analyze trending videos, it was very important for us to find out the top videos or top channels that have impact on trending. Therefore, we added advanced queries and stored procedures that finds the following: top videos by average views, top channels by category, and number of videos in each channel.

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

- Trigger: UpdateChannelViews
  - This trigger complement our application by automatically maintaining an accurate count of total views for each channel, ensuring the application's data remains consistent and up-to-date without requiring manual updates or additional queries.
- Stored Procedure: GetPopularVideosByCategory
  - This stored procedure complement our application as it allows users of the database to identify the popularity and engagement of certain categories of videos. It supports our purpose of analylzing trending videos, and so this could be useful to people interested in knowing which categories are trending and which may consistently generate the most user engagement.
- Stored Procedure: GetTopChannelsByAverageVideoViews

○ This stored procedure complement our application since it provides valuable insights into the most popular channels by identifying those with the highest average video views, helping users discover content that resonates with a larger audience. Additionally, it enables content creators and marketers to analyze the top-performing channels to gain inspiration and better understand the factors driving success on the platform.

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.

- A technical challenge that we encountered at the beginning of the project was our group's limited experience with both front-end and back-end work. More specifically, most of our group members had limited experience working with NodeJS, Google Cloud, and HTML. Understanding how to build an entire application without having a background in these areas was one of the difficulties.
- In the beginning of the project, we had to insert data in csv format into the table. However, we had some foreign language characters that are not in ASCII, and so it often crashed and not all the data were inserted. Therefore, we went to the office hours and received help on how to convert the foreign language characters into a convertible, acceptable version. We eventually figured out a way to change the type of csv file so that the foreign language characters are accepted when importing to the database. This took us several days, so it was one of the most time-consuming technical challenges we had.
- Another challenge we faced was designing and implementing the database schema design. Originally we had planned to have various foreign keys and primary keys, and also implement weak entities but we realized that inserting data into these tables required more work and we faced a bunch of errors when trying to properly implement the weak entity table, etc. But with help from Stackoverflow and TA's office hour, we eventually figured out how to handle weak entities and foreign keys.
- We also faced a challenge where we don't know how to integrate the advanced queries and advanced database programs into our website. Most of the queries we wrote at first had several errors, and we spent a great amount of time solving such errors. Eventually, we found out a way to implement advanced queries and advanced database programs by

focusing on writing advanced queries that are related to youtube channels, number of views for videos, etc.

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

Comparing the application to the original design, there were several similarities to what we envisioned and some aspects of the design that we were unable to figure out how to implement. For similarities, we were able to store and use the data given from the dataset pretty similarly to how we initially pictured. The tables we created and the relationships we had between them seemed to be close to what we had initially envisioned. However, there were some aspects to our initial design that we had a hard time implementing. For instance, the creative component we initially had involving graphs was something that was beyond our scope at the end.

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

We could implement a NoSQL database into this application which would be useful. If we were to include a NoSQL database, we would likely incorporate it into our application using data that was less structured. User-generated content such as comments, likes, and subscriptions would be examples of this type of data that could be stored in a NoSQL database. Additionally, if we wanted to measure the behavior of user engagement, a NoSQL database could be useful in storing this data, which could in turn provide valuable insights into user behavior and preferences. This would enable us to provide more insightful analytics which would be useful to understand factors important to make a good youtube video.

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

In our project, we recognized the importance of ensuring that every team member had an opportunity to learn and contribute in each stage of the project. In every stage involving GCP, the database implementation, the website, we made sure to figure out the setup together and then divide the work into smaller tasks and rotated responsibilities throughout each stage.
During the planning stage, we held brainstorming sessions to identify the project's requirements and discuss potential ideas. During the design stage, we rotated roles to make sure everyone had a chance to contribute to the database

design and had a chance to learn. During the development stages such as implementing the CRUD operations, we figured out how to get one operation working together and decided to divide the other CRUD operations, one per person.

Finally, during the testing and deployment, we worked together to ensure that the project was thoroughly tested and deployed correctly. We assigned team members to manually test and fix any issues that arose.
By rotating tasks we made sure that every team member learned something and were able to grow their computer science skills.