**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).**

Originally, our project had these planned features:

- Game recommendations based on price range, reviews, played games
- Searches based on price range across different regions and stores
- like/dislike feature for games
- Filter games by price, region, and store

Our final project has these features:

- Using our database, our website gives game recommendations based on overall developer reviews and game ratings (plus a user-based recommendation feature that shows a list of games based on the user's played games).
- A register/log in feature that lets users have a saved username/email and password in the database
- Users inputting games played and rating games

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

Our application was able to pull lots of data about video games and deliver useful recommendations to users. It Also was able to store user login data, and then use that user data to deliver specific recommendations.

In this way, it achieves the core goal of the project, which was to take data from different sources, and make it useful in a specific way.

**Discuss if you change the schema or source of the data for your application**

We changed our schema and data sources multiple times. Our vision of the site kept changing in small ways, and our understanding of our schema kept changing with it. We started off with a very broad schema, but then made it more specific to our application. For example, we had a "reviews" table, but then used it to give recommendations for specific users.

**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?**

The changes we made in our ER diagram include making 'Tag' a separate table instead of an attribute for 'Games'. We changed our original attribute 'Genre' into a separate 'Tag' table because there are many genres and categories that define what a game is. Therefore, it is important to address the many labels that games can have.

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

The original function we wanted to implement was to search games and view real-time price comparisons across stores from different regions. However, due to time concentrates and how complicated/specific it is to not only find dataset for every single game with multiple game stores but also from every single region which is too complicated to move forward with.

Another functionality that had to be dropped is to make a game recommendation based on a user's Steam profile activity. Due to Steam not giving the public access to their private dataset of its users' gaming activities, implementing the function would not be possible unless given private access.

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

Our advanced database programs complement our application in many ways. First off, Indexing mechanisms were used to speed up data retrieval and make our program faster. Especially when searching for games indexing allows our application to quickly find and give relevant information regarding a user's game taste . Secondly, Our API helps us greatly with integration and exchanging between processes such as logging in, searching, and finding specific game titles. Finally, the ACID properties also help maintain our data within our application when registering, logging in, and viewing user data.

**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.**

1. One challenge that I faced that would be helpful advice to a future team is to freshen up on your git knowledge. When working in a big group where everyone is consistently making changes it is important you have the up to date code at all times. In certain times it can get more complicated than just pulling and pushing and it would be helpful to have a good understanding of how git works.

2. One challenge that we faced was front to back-end integration. Front to backend integration is a crucial part of these projects and it is important to define clear API endpoints and to make sure everything is connected properly. It would be a good idea to understand how API's work and how to connect endpoints.

3. One challenge that we faced was running multiple queries. It is important to keep track of all of the queries and to make sure outputs are sent correctly. Using operations such as Promise.all was beneficial for running queries concurrently.

4. One challenge that we faced was editing through VIM on the GCP terminal was challenging. Using a third-party application to edit code i.e VSCode was very helpful for debugging, testing, and pulling/pushing.

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

The original proposal planned to focus on comparing the prices of games from different regions. The final application does not focus on the prices but rather focuses on recommending games based on similarity to other games the user has played.

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

Other than the interface the application can improve in many areas. The application could have better security, include more games in the database, and look at more factors than just genre, reviews, and developers. Such as difficulty, art work and gameplay mechanics.

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

Mike Lee: implemented queries for the backend, set up the application on Google Cloud Run, fleshed out the user file for the application,

Uriel Arellano: also implemented backend database queries, integrated connections between the frontend and backend, help cleaned out the frontend

Mohammad Fattash: implemented the frontends of the applications, style of choice was with Bootstrap, implemented the templates for the integrated interfaces, assisted in implementing queries for the backend

Jonathan Biel: implemented frontend for logging in and registering. Implemented home page UI. assisted in the implementation of queries, databases, and this report.

Overall, everything was managed well and efficiently, with frequent communication between team members.