Please list out changes in the directions of your project if the final project is different from your original proposal.

The orientation remains unchanged, prioritizing the places themselves over the user's contribution. We haven't fully developed our interface as anticipated; perhaps we'll need additional time to work that out.

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

We have successfully completed the CRUD process and developed a web frontend for our page. However, we are unable to add new comments to our database from the front end, despite spending several days debugging the issue. However, we can insert or add comments in our backend, and we have the functionality of constraining that only an existing username can add or edit comments. We failed to create an interaction map for our user, and we might need more time if we want to figure that out.

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

We gathered the data on recreation sites from various sources such as Wikipedia, Google Datasets, and Kaggle. Since we didn't have any real users in our user and comment databases, we had to auto-generate this information to ensure it aligned with our table format. Discounts were a mix of the two; it was difficult to find long lists of discounts that were in the correct format required for our table, so we combined data we found online with generated data to meet the 1000 entries requirement.

Discuss what you change to your ER diagram and/or your table implementations.

We found that our table implementations easily followed our ER diagram, and so we did not need to make any major changes. However, we did rename some attributes, such as 'comments' to'messages', for clarity.

What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

We focused on simplifying and streamlining the final design. Our initial plan called for an interactive map that could show every piece of knowledge on recreation in a specific state; however, time constraints prevented us from completing this. Instead, we list all the recreation names and addresses on our front page. If users are aware of the specific states or recreational facilities that they wish to visit, they may use the search box to locate the comments that have been associated with each institution.

Discuss what functionalities you added or removed. Why?

Searching for comments and managing them using CRUD (Create, Read, Update, Delete) worked well. People can look at recreations and read, add, change, or remove notes with this app. Through the use of the app's search engine, users are able to easily locate leisure sites or comments, consequently boosting ease of use and accessibility. There was a

slowdown or deletion of the interactive map and suggestion system. Time and technology constraints deprioritized these, despite their intended boost in user engagement and customization. Through focusing on important functions, the program met its objectives and left room for future growth.

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

Constraints and Triggers: Using constraints (e.g., foreign keys, unique, not null) and database triggers to maintain data integrity automatically. For instance, ensuring that no duplicate comments are posted or automatically updating metadata when a comment is edited. Indexing: By creating indexes on frequently queried fields such as Username, CommentId, or DatePosted, database queries can be expedited significantly, which is crucial for fetching comments efficiently.Query Optimization: Using optimized queries or stored procedures can reduce the load on the server, decrease response times, and increase throughput.

Each team member should describe one technical challenge that the team encountered. These should be detailed enough for a future team to use as advice on starting a similar project or maintaining yours.

Linking Frontend and Backend
A major challenge was maintaining a stable connection between the frontend and backend, as it frequently broke due to mismatched APIs or data format inconsistencies. Future teams should ensure thorough testing of endpoints and consistent data structures to avoid similar issues.
Using GCP
Setting up and managing the application on Google Cloud Platform was challenging, especially with configuring database connections and permissions. Familiarity with GCP's authentication and quota systems can save time.
Frontend Design
We struggled with frontend design due to limited access to relevant tutorials. Future teams might benefit from exploring frameworks with better documentation or seeking out community-driven resources.
Database Triggers
Implementing triggers caused conflicts with other operations, leading to errors. Careful planning and testing are essential when using triggers to avoid disrupting database workflows.

Are there any other changes between the final application and the original proposal?

The final application and proposal had similar broad direction and basic objectives. However, we were unable to implement certain intended supplementary functionalities, such as a recommendation system. These features could have enhanced the user experience by providing personalized recommendations based on the user's preferences or past activities. Despite these omissions, the project continued on a strong path and effectively built the basic features. Prioritizing these features in future iterations of the application could further align with the original vision and increase the overall value of the application.

Other than the interface, describe future application improvements.

Adding a feature that lets users add their own places or recreation spots to the database instead of basing it on the privilege of a determined list is a major enhancement. This capability would make the app more dynamic and user-driven and allow the database to develop naturally as users share beloved or novel leisure areas. Additionally, this feature would promote user engagement by giving them ownership and participation in app content improvement. We need strong validation procedures to keep data accurate and stop people from adding irrelevant or duplicate information if we want to do this.

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

A challenge was finding a time to meet that worked for everyone in the group, especially on a regular basis. To overcome this challenge, we would take advantage of weeks when there were no class meetings, meet during the regular class times on those days, and also meet a few hours before office hours so we could attend them together. Each of us is responsible for different problems in our project. Colin is responsible for backend design and implementation for our project. Karena writes for all the frontend web design and handles web route issues. Yanran tackles all the issues regarding GCP and databases, like connecting GCP to our backend. Momo helps the whole team to connect the frontend and backend together.