

- 1) Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

Originally we planned for a more complex user interface where items could be favorited, ranked, etc but in our final application we used a much simpler scheme where users can search, add, and delete as the main functions. We also planned a creative component that would implement a hurdle game of sorts but in our final application we didn't end up implementing it.

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

I think our application achieved its usefulness. It allows the users to look up, add, and remove information about the Tokyo 2021 Olympics in a single robust application. Without our application, users would have to search the internet and possibly not find what they were looking for or scrap through the data themselves.

- 3) Discuss if you change the schema or source of the data for your application.

We did not change the schema or the source of the data for our application.

- 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?

Our final design of the table implementation used more foreign and primary keys than the original design. This is because we wanted to more strictly enforce relationships between tables as well as reduce the amount of values that were in the table when they were joined together.

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

We decided to add insert and delete functionality to the application because we thought that if the user wanted to add custom data to the database for testing purposes, they should be able to do so. We removed the favorite functionality because we didn't believe it would be very useful to the application.

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

I think the advanced database programs are helpful because they save the user time and energy. The trigger when someone adds to one database and it gets applied to all databases will help with reducing the amount of bugs when custom data is added. The stored procedure can be refactored to do all sorts of different comparisons between specific data points vs the averages.

- 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.

Bobby: A technical challenge I encountered was the connection between vscode and the gcp vm. The main thing is how to properly install the gcp package. I think a video guide lesson detailing that would be very helpful for future students.

Akhil: A difficulty that I had was adding the trigger and stored procedure to our GCP SQL database. Throughout the project we had use the GCP MyCloud Console to access our database. When adding our trigger and stored procedure we ran into issues related to syntax and copy-pasting the code in. To fix this we used the MySQL Workbench connected to the database. This allowed for a easy way to view what was wrong in our code, along with pushing directly into the database.

Paul: While designing the front end of the application, I found it challenging to visualize results and test function calls. Multiple programs were required to test the application, including the mySQL workbench, the Google Cloud VM, as well as my local test environment in Visual Studio Code. In order to test the connection from the front end to the back end, I needed to copy code from the VM, create & debug the front end on my local environment, re-upload the code to the VM, and I needed to open the SQL workbench to see if anything worked. The constant back-and-forth nature of testing made the process tedious, and it could also become impossible if one or more programs failed to start. Future teams should try to consolidate code as much as possible and have more local testing environments, such as the provided Github, before committing to the VM.

Stephanie: A challenging aspect I encountered was connecting the front end code with the POST calls in the back end. We ran into an issue with the syntax of our ejs code and it made it so our forms weren't sending the data to the server. For future students, make sure multiple people look over the code so that small errors you might have missed get caught.

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

The final user interface is more of a single page application that goes straight down the page rather than a flex box design with more dynamic javascript utilities. We also did not include a user database to keep track of users as well as a visual graph component to display data query results

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

I think that the setup of relational tables in the database would really help the speed of the database program. Also I think that if we were to filter down our queries before the joins/where clause that would also help with runtime for certain functions. Another thing that could be interesting if implemented would be using a NoSQL database to store our data. Using a database like Neo4j to

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

We met over zoom multiple times and used group chats to divide up work and help each other when we were stuck.

- 11) Create a release with the correct tag for your submission **and submit it on canvas**