Krish Patwari
Jimmy Xiao
Rhea Tiwari
Group 25

Final Project Proposal

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 was supposed to help users create football teams based on their preferences and player statistics collected from over the last 10 years. They would also be able to create, update, and manage their teams. We realized that this idea would be a little complicated to implement and would had been heavily focused on a coding language like python other than the usage of sql and would take longer than the time we had, so we decided to simplify our application a little. Instead, users are allowed to choose a favorite player and team and enter the fan board system.This allows them to get an interactive experience with fans all around the globe and find similar interests. They can also search for matches, players, and teams and filter out their targets to help fine tune their understanding of the game. They can also select trivia questions and interact with the website.

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

We think in terms of what we wanted to achieve with our project, our application was useful. We changed a few things regarding functionality, but we still were able to use the data and have users interact with the application. The users can find different players, teams, and matches, and learn more about them with the data.Overall our goal was to better the experience of football for fans and help them with fantasy teams.We are somewhat on the same lines allowing fans to filter out data and at the same time stay engaged with the game with features like trivia questions to arouse excitement about crucial football details like aggressive/defensive teams etc and make informed choices.

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

We did not change the schema or source of the data because the data that we had was useful to the main all of the functionalities we were implementing. We had already done a lot of research work in selecting our source of data.We could fetch csv files that we needed and because football is such a common sport data was readily available.

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?

We ended up changing some attributes to our table entries. We reduced the number of attributes we had to make the code more efficient and to get rid of attributes we didn't need. We also changed one of our tables from Player Game History to Match Game History. This change helped with what we changed our goal to and helped with the queries. Some of our foreign key assumptions were made assuming that our data sources were clean. This turned out to not be the case, and some foreign key relations that ideally would be in a perfectly maintained database had to go.

Discuss what functionalities you added or removed. Why?

One functionality that was removed was having a search that would create a team with 11 players for the users based on their preferences. We removed this functionality because we realized the formula was getting too complicated and we didn't have enough time. We simplified this by adding some other functionality including a User Data Form where users can put in their name, favorite player, and team. This data will be added to the fan board. They can also search for different players, matches, or teams after inputting some data. They can also select trivia questions and interact with the application.

Some constraints were designed under the assumption that we had perfect data. Unfortunately, our sources often had missing data such as some players not having full stats or games referencing teams that did not exist in the teams database. Going through each game and figuring out what happened so we could clean the data did not prove feasible within the time period, so we had to cut some foreign key constraints out.

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

Our triggers make sure that users cannot enter multiple entries into the fan board and that their favorite player has to play for the team they enter. This maintains consistency and does not clog the fan board with repeated entries.

Our stored procedures enhance the readability of our functions in server.js.This helped with debugging and making further developments easier.Moreover it housed all the complex queries for the trivia section.

Our constraints are primarily in foreign key constraints that enforce users to use players and teams that exist in our database, as well as only allowing users we currently trust into the database.

Our transactions were used on the fan board page. Since we are on the read committed isolation level, we may experience unrepeatable reads. However, this is acceptable as our fan board is not extremely serious and sensitive data. The transactions we used are aimed to ensure that writes and edits to the database are isolated and should anything happen to the database in the middle of a write/update, the database's integrity is not compromised.

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.

Krish:One of the significant technical challenges I encountered during the development of the football database application was efficiently retrieving and displaying large datasets, such as player statistics, match details, and team information. The challenge involved ensuring that the application could handle user queries promptly without compromising performance, especially when dealing with large volumes of data.The large dataset caused problems with the heavy usage of javascript and ajax and especially connecting it with the database on GCP.The environment was a bit unfamiliar to work with that caused a bit of issues so I would definitely recommend a web-development course prior to this course to help with the course.

Jimmy: I had difficulties with connecting the front-end to the back-end. When we were re-building a page after adding transactions, I had forgotten to send a success signal after a query succeeded. This would've taken me ages to find out had I not saved previous states of the project. If someone were to take this project over, I would recommend immediately using version control and getting everyone on the same page with how to use Git and GitHub.

Rhea: One challenge that I encountered was with the triggers. I wrote down the code for what I wanted it to do, but I kept running into issues with the implementation. I tried different variations, but was not able to get it to run. I think there might be an error with how I was implementing it. I used the online lectures to help me.

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

The User Interface was redesigned based on our new requirements and we came up with a design that most suited our needs.This was a bit different from the UI we proposed which we didn’t realize would be too difficult to implement .We ended up changing the goal of the application and some functionalities as described above, but there was nothing more we changed.

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

We could maybe add some more functionality and add the formula that we originally were going to implement. We can add some more user interaction with the application, which we could do with the formula. Right now we run on the honor system for editing/deleting fan board entries. Proper user authentication would be a great addition.Adding features to increase fan engagement, such as polls, forums, and social media integration is an option as well.

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

Krish worked on the frontend, while Jimmy and Rhea worked on the backend. Krish implemented the search algorithm. Jimmy implemented the storage procedures and transactions. Rhea implemented the triggers. The other functionalities were a collective team effort.

A version control like Git would have made collaboration much easier and more efficient. Peer feedback helped improve the overall quality of the project.The team was flexible in adjusting roles and tasks based on project needs. For instance, if one area was falling behind, team members would pitch in to help meet deadlines.

References

There was use of bootstrap in developing the frontend portion of the application.For connecting the backend to the frontend we took inspiration from the node js workshop offered to us in class and worked out a similar approach as this concept was relatively new to us.