CS 411 - Database Systems Spring 2023 PT1 Final Project Report

Submitted by -
Syed Ammar Raza (saraza2)
Payal Mantri (payaljm2)
Stephen Moy (moy26)
Youhan Li (youhanl2)

PlyRank - Fantasy Soccer

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).**

   The project we developed was quite similar to our original proposal of a web portal for virtual fantasy soccer. However, there was a significant change in the backend implementation. Instead of using Java and Spring Boot, we decided to use Express APIs. This change was primarily because we initially planned to use ORMs but later found out that they were discouraged as per the project guidelines. Therefore, we thought it would be easier to implement queries using simple Express APIs. This approach also facilitated easy deployment on GCP.

   The final portal we built was a fantasy soccer league where users could create teams, search players, and perform various other actions. Additionally, the admin could view the user details.

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

   Our team has successfully developed a virtual fantasy soccer portal for soccer enthusiasts that includes core functionalities such as user login, team creation, player search, and viewing statistics related to clubs, players, and tournaments. The application enables existing users to login using their account details such as email and password, while new users can register to participate in the contests. The admin panel provides useful information about the users of the portal.

   Although we could not implement a few of the functionalities from the initial proposal, such as managing player purchases based on available funds, the final application achieved most of the proposed functionality. Currently, all users can add any player to their team without checking whether they have sufficient virtual money in their wallets. However, this feature will be useful if we want to commercialize the app, where users

can add virtual money using actual money and win real prizes. Overall, the developed application can provide an enjoyable and informative experience for soccer fans, and the missing features can be developed in future iterations to enhance the usefulness of the portal.

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

   In our application, we did not make any changes to the schema or the data source. We stuck to the schema we had initially designed during stage 2 of the project, and the data was sourced from Kaggle. However, we did perform some cleaning processes on the data, which resulted in slight differences from the original data. Despite this, we were able to utilize the data effectively to implement the functionalities of our virtual fantasy soccer portal.

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?**

   There were no modifications made to the final table implementation based on the ER diagram and relational schema created in stage 2. Therefore, there are no differences between the original design and the final design. We believe that the initial design was suitable for our project requirements, and it did not require any major alterations.

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

   During the course of developing our application, we made some changes to the original proposal. In terms of functionality, we added several features to enhance the user experience. These additions include a Tournament List, displaying top performing players, a Club List with top performing players, as well as validations to ensure that each user team cannot have more than 11 players and that users cannot register with an email address previously used.

   In contrast, we removed two features from our original proposal. These were the ability to manage player positions within a user team, and the ability for an admin to archive or deactivate players.
   One of the reasons for not implementing additional functionalities, such as managing player positions and archiving/deactivating players by admin, was the time constraint. We wanted to focus on developing the core functionalities first, and due to the limited time available, we were unable to implement all the desired features. Therefore, we decided to prioritize the essential functionalities first .

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

   Advanced database programs have been an integral part of our application's development, and they complement our application in several ways. We used advanced queries, such as stored procedures and triggers, to enhance the functionality of our application.

   One example of how we used advanced queries is in the use cases of querying the top performing players for a tournament and querying the top performing players of a club. To achieve this functionality, we used stored procedures. Stored procedures are powerful and efficient, as they allow us to write complex queries that can be executed with a single command. This enabled us to retrieve the required data quickly and efficiently.

   Another example of how we used advanced queries is in the use case of validating that each user can add only 11 players to their team. We implemented this validation using triggers. A trigger is a special type of stored procedure that is automatically executed when a specific event occurs in the database. In our case, we created a trigger that would fire whenever a user tries to add more than 11 players to their team, resulting in an error.

   In summary, the advanced database programs we used, such as stored procedures and triggers, added an extra layer of functionality and efficiency to our application. They allowed us to implement complex queries and validations with ease and made our application more robust and reliable.

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.**

   Payal: As a team member, I encountered few technical challenges during this project. Being experienced in working with databases as a full stack developer, I initially underestimated the complexity of setting up a project from scratch and deploying it on GCP. It was a major challenge for me. Although there are a lot of tutorials available to help in such things, finding the right one that worked for our project was difficult. We tried many approaches and finally put everything together. Additionally, teaching the team some basic concepts of a web app and managing the team were great learning experiences for me. I believe that future teams could benefit from our experience by taking the time to thoroughly research the best methods for project setup and deployment, and by being patient and flexible when teaching new concepts to team members.

   Ammar: When doing the frontend aspect of our project, one major issue I encountered

was the learning curve of React and JavaScript, especially as a first time user. While YouTube tutorials were helpful, majority of them explained how to implement certain features by creating a project from scratch rather than as a part of an already well established project. For situations like these, AI models, such as ChatGPT, were incredibly helpful in explaining how to implement a certain feature given a set of preexisting conditions.

Stephen: When testing our APIs in the backend, we were unable to visualize the results without first implementing them in the frontend. As a result, we could not determine if our queries were working. For this issue, we were able to log our queries and the results of the queries in the console.

Youhan: When testing our backend program, we sometimes want it to run continuously in the background, without interfering with other jobs in the foreground. In this case, we found out that the npm module “forever” and its debugging variant “forever-monitor” is very useful. The module can run multiple processes at the same time, and its interface is very convenient.

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

   No, there were no additional changes made to the final application beyond those discussed in Q1 to Q5.

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

   There are several areas that can be improved for the application beyond the interface, such as:

   Implementing transactional operations: As mentioned earlier, the current application allows users to add any player to their team without checking whether they have sufficient virtual money in their wallet. Implementing transactional operations to manage player purchases based on user funds can make the application more realistic and enable monetization of the application.

   Implementing machine learning models for prediction: The application currently provides statistics on players and clubs, but it does not provide any predictions. Implementing machine learning models for predicting player performance or match outcomes can enhance the user experience and increase engagement.

Adding social features: Adding social features such as chat or forums can make the application more engaging and foster a community among users.

Enhancing UI/UX design: While the current interface is functional, enhancing the design can make the application more visually appealing and user-friendly, leading to increased user engagement and retention.

Improving the current player dataset: The current FIFA23 dataset which we utilized does not have many data entries with regards to goals, assists, etc. when it comes to lesser known players. We could utilize a dataset which contains more detailed data for all players in order to make the application more realistic and interactive.

Overall, there are many areas for improvement beyond the interface, and focusing on these areas can enhance the functionality, user experience, and monetization potential of the application.

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

Our team worked collaboratively to deliver the final project. During the brainstorming phase, all teammates contributed ideas, and we worked together to create the ER diagram, database design, and implementation in stages one through three. For stages four and five, we divided the workload based on our initial proposal, with Ammar and Payal focusing on frontend components and Stephen and Youhan working on backend API and deployment.

Ammar was responsible for designing SQL queries in MySQL and creating frontend components using JavaScript and React.

Stephen was in charge of designing SQL queries, developing backend API using Node.js, loading data to GCP, and creating stored procedures in MySQL.

Youhan worked on the backend API, loading data to GCP, and implementing stored procedures in MySQL.

Payal designed the frontend API using JavaScript and React, developed frontend components, designed the database, created the application structure and interface design, and implemented trigger operations in MySQL.

Throughout the project, our team worked well together, with good work distribution and a positive team atmosphere. We communicated directly and clearly, and each member was assigned a role they were comfortable with. Our TA, Kevin Pie, was supportive and provided valuable advice that guided us in the right direction