CS 411

Fall 2022

Final Project - Stage 6

Roosters' Fantasy Football Tracker

Team #041: Shouri Addepally, Chirag Kikkeri, Raj Pulugurtha, Dhruv Vishwanath

Teaching Assistant: Kerui Zhu

Changes from Original Project Proposal

For the majority of our project, we were able to adhere to the functionality we wanted to implement as part of our original project proposal. For our data collection we ended up using pro-football-reference.com and we kept our overall tables structure the same as we wanted to as indicated in our proposal. As part of our proposal we also wanted to be able to execute basic CRUD operations on our database and we were able to accomplish this as well. However, the way we were performing our recommendations was slightly different than what we proposed. If a user has no existing players on their watchlist, then we return a list of ten players from the team with the best record in the league. Otherwise, we would return the top players from teams that the user already has as part of their watchlist. We also were not able to add the creative visualizations in addition to our watchlist, but this is something we can definitely add in the future.

Achievements of the Application

The achievements of our application are that we are able to maintain our comprehensive database while also being able to generate a valid recommendations list for each user. In addition, we were able to generate advanced queries which were relevant and in-line with how our watchlist was set up. We were also able to maintain an efficient UI and all of our CRUD operations worked in our database. From a holistic perspective, our project was able to create a beginner-friendly system for fantasy football, so that those who do not have a lot of experience can be recommended players that would benefit them. Apart from all of these benefits of our application, we ran into challenges when we were trying to set up our initial advanced queries as our database got wiped in the process. We also could modify our existing watchlist to take into account other attributes such as player position. This could better address scenarios where the user may only want to see the best players at a specific position and further improve the recommendations that we offer as part of our watchlist. We consider this to be a failure because of how this situation of adding functionality put us in a time crunch, thus taking away the opportunity for us to add more creative elements to our project.

Changes to Schema and Data Sources

The source of the data stayed the same throughout the entire project. We used the same pro-football-reference.com to gather all our data at the beginning and then used that same data till the end of the project. We also didn’t change the schema from the schema that we submitted in stage two. All the data that we said would be in the tables in stage two are in the database. We did change from stage one to stage two by eliminating the Years of experience table and instead adding that as an attribute to the players table, but that is the only change that we have had with the schema.

Changes to ER Diagram and Implementations

We largely follow the ER diagram that we set out in stage two. We ended up not using all of the tables in the database that is laid out in the ER diagram, as we pivoted to focusing mostly on the watchlist, but the required relations are all present. The only variation in our implementation is that the watch_list name table name in the diagram is now just watchlist, which is a minor difference in the grand scheme of the project.

Changes in Proposed Functionalities
In our proposal we said that we would have a “fantasy view” which would be some creative visualizations to allow our users to understand the stats and trends that we present them. However, we didn’t end up doing that, as we underestimated the difficulty of setting up our user interface, and over time we decided to focus on nailing down all of the basic functionality first before adding any sort of creative aspect that really wasn’t necessary to the project. We also added the advanced queries into our website that we hadn’t thought of in the proposal. Comparing the best players across two divisions and displaying the average points per team were two new additions that came about as a result of needing to have some advanced queries.

Utilization of Advanced Database Programs
Our advanced database programs are integral to the functionality of our application. Our trigger was key in making the user interface more intuitive. Before, if a user left a blank note, the table value looked incomplete and could have signaled to the user that the data wasn’t retrieved correctly. However, our trigger replaced any blank notes with “N/A” which makes it more clear that the display table was completely correct. In addition to our trigger, our stored procedure was integral in recommending new players. The procedure had two aspects depending on the status of the user’s watchlist. If the user’s watchlist was empty, then we would recommend players from the best teams. However, if the user’s watchlist was populated, we would recommend players on teams that were already represented on the user’s watchlist. Both of these options gave the users valuable recommendations that would fit their use case.

Technical Challenges
Chirag: One of the main technical challenges I faced was scraping our player data from pro-football-reference.com. Initially I was using a value from their tables as our primary key because I thought they wouldn’t repeat that value. However, I found out that the value had duplicates so then I needed a new strategy for our primary key. I ended up creating our own player ID as we added it to our database instead of the website’s.

Raj: One major challenge I faced was writing our advanced query for the stored procedure. For some reason, when I used the EXISTS set operation to not recommend players already in the watchlist, I was unable to exclude those values. What I found out was that there was something else wrong in how I created the temporary table, so the query wasn’t incorrect. I discovered the issue by running the query by itself instead of in the stored procedure.

Shouri: I encountered a major challenge with our SQL database. My laptop must have restarted between stages four and five, and our database got wiped off my laptop. It was very frustrating because we had to create our tables from scratch again, but luckily we had all of our data still saved. We ended up using a friend’s laptop to host our database and connected through wifi to it since that laptop was not being used for anything else. Next time, we could try using GCP instead because I think it would be more reliable.

Dhruv:
One problem I had with the User Interface was having a dynamic table to report our results for Search and the watchlist. For some reason, the HTML table didn't update properly even though I sent the data from the query correctly. After doing some more research, I found a method that worked better for our use case. Although it took some more time to implement, the overall utility and design of the new table was far better than the old table.

Additional Changes from Proposal
For the most part, we do not have major additional changes from our proposal. Our changes come primarily from our watchlist and how we select players to suggest in our watchlist based on whether or not the user already has players on their watchlist. The way our watchlist is built is by selecting the top players from teams where current watchlist players play on. If players from a specific team already exist on the watchlist, we will try to gather more players from this team. Our team can improve this existing functionality by grouping by positions instead of teams, and perhaps looking for divisions where a specific position has the highest point average. Apart from the watchlist, we have a basic user interface implemented and we can make this more elegant in the future to enhance our overall webpage.

Potential Future Developments
Based on what we accomplished with our project so far, there are many potential avenues for improvement, and several different directions in which we could expand our work. One key improvement we could make would be to work on our application outside of the watchlist functionality so that information could be broken down to the users in multiple ways. An example of this would be to be able to categorize players by position, which is a feature we did not add due to time constraints. We could also improve the recommendations section of our application (which is directly connected to our stored procedure) so that it recommends players more dynamically. In order to do this, we would need to add some artificial intelligence aspect to our project so that the recommendation list would change over time based on different factors that are specific to the user, such as the players on the user's fantasy football team, and what the weaknesses are of a respective user's team. Based on this, we could have a system that recommends specific players so that we can tailor results to whoever is using our application. Lastly, we can also improve the appearance of our application. Two examples of this would be to enhance the interface so that it looks more professional, and adding a login page so that we can have multiple users on the website at the same time.

Division of Labor
Generally speaking, the division of labor amongst the group was as discussed in the original project proposal, with more flexibility among groupmates than initially anticipated. A unique advantage of our group is that all of the members live together, so it was easy to keep ourselves motivated and on top of the project since a group member would see another member doing their part, which would incline them to work on their own part. Additionally, it was easy for us to help each other out, as if one person got stuck on something, there was likely someone nearby who had the ability to understand and help fix whatever bug came up. For these reasons, our group worked together very coherently, and we were able to bounce back from every setback

gracefully as a unit. Although each group member was involved in every aspect of the project and most of the work was done together, towards the end of the project it ended up working out to where Chirag and Dhruv took the lead on the frontend and user interface portion of the project, while Shouri and Raj were in charge of working on getting the backend portion of the databases set up and figuring out how the group wanted to implement our various SQL queries.