

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).

For the most part, our final project was similar to our original proposal during stage 1. The biggest change is that in the original proposal, we intended to have the focus of the app to be on the rankings themselves, while in the end the focus was more on providing the user with a wider array of information, such as information about the university, the location, and the salary potential of different majors. Some of the other changes include that we had multiple pages on the application, including the University details, the Country details, and the User Favorites page. These pages allowed our project to convey more information to the user that would be helpful in making decisions about college.

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

Originally, our goal was to allow users to create personalized rankings, so that they can identify universities that are a good match for them based on what they value the most. Our final application was able to achieve this on a high level as users can select several universities as favorites and compare their scores for different categories, such as research, teaching, and employment. Additionally, on the favorites page, users can compare the salary potential of subjects that are offered at each school. This also helps achieve the usefulness that we intended. If we were able to implement filters on other columns on the main page, we would have fully achieved our original goal, so the lack of filtering is where we failed to achieve our usefulness goal.

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

During the development of our application, we did not change the source of our data. We slightly altered the schema by adding a column to the Users table that tracks the number of favorite colleges. This allowed us to implement the trigger that limits the number of favorite colleges to 10 per user.

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?

While working on our project, our only change to the table implementations was adding a column that tracks the number of favorite colleges to the Users table. This was done in order to implement a limit of 10 favorite colleges per user. Between these two designs, there is not too much that is significantly different, but if we had to choose one, the latter design would provide slightly more functionality, so it is more suitable.

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

Throughout the development process, we found that certain functionalities could be simplified while functionalities were more complicated than we initially thought. The main functionality that was left out was the filtering of other columns on the index page of our application. After a few different attempts at implementing this functionality, we were unable to find a solution, so we decided to leave it out. We decided to replace this functionality with a few new elements, such as the new pages that provided information about the country and the university. This helped us transform our application from being solely used for university comparison to an application that can be used for more widespread college planning.

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

Our advanced database programs allowed us to add a few extra details to our application that would help out a potential user. For example, through our stored procedure, we allowed the user to identify elite colleges in their favorites section. This is particularly helpful when comparing schools that are already liked by the user. Likewise, our trigger also added a small detail that limited the user to having 10 favorite colleges. This prevents the user from having an excess amount of favorite colleges, which would bog down the information presented when they are comparing universities.

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.

Maanas: During the initial phase of inserting data into the database, we ran into a few issues when we had to insert data into a table that pulled from multiple datasets. For instance, when we were creating the Universities table, we used data from two different datasets, which had slightly different naming conventions for the University (some had “The” in front of the name and some were in the native language). This caused the final dataset to lose a decent amount of university data for those that were not in both datasets. Additionally, there are some universities with 0 or NULL values because of this. In the future, I would recommend trying to find datasets from the same source or performing more extensive data cleaning and preparation.

Navtej: One technical challenge that I faced was working with .ejs files for frontend development. I have used the traditional HTML/CSS file folder structure and it took me a bit of time to fully understand how it worked. I also would highly recommend the teams learn how to incorporate javascript into the frontend pages earlier than later. When we were building, I found myself confused with where the separation occurred between the

backend .js files and the frontend .ejs files. I think that there was some inefficiency in my code and perhaps some duplicates in work done that would have been nice to have reduced. I also think from a JS perspective, it might actually be easier to do some of the things in Python and Flask rather than JS. I'm not sure how much of the DB stuff would have translated effectively, but from an app dev standpoint, it is much easier to use Flask or Django.

Ashmit: A significant technical challenge our team faced early in the development process of our application was understanding UML diagrams for database design. None of us had prior experience with UML, and we needed to not only learn what these diagrams are but also understand how to use these diagrams effectively to plan out our database tables and their relationships. We were really lost in the beginning when trying to figure it out since we didn't have set steps that we would take to approach the initial setup, but after going through that experience I would recommend that when you're planning out your initial idea, you should already know what tables and data you'll require so while learning UML or ER diagram production, you're already formulating in your head what you want the connections to look like rather than using the UML diagram to figure out how you want the database to be connected.

Khushi: One technical challenge I faced during the indexing process of our advanced queries. It was our first time properly working with the GCP terminal and so learning how to utilize that and the commands that work with manipulating our tables was all new and something I had to get used to. On top of that, I would run into issues with our advanced queries not outputting a large enough set of data so we would make adjustments there, then translate those changes into our indexing. One main issue I would run into is figuring out how to handle the foreign key constraint errors. If I were to accidentally index a foreign key I would run into this error, so we had to figure out how to drop the whole foreign key and alter the table to go back to its original state. A few times the table would be permanently altered and we would have to redo that index. Once we were able to solidify a way to combat the error the indexing went smoothly, and it would have been easier to check our primary and foreign keys for each table first before attempting to index on them.

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

Other than the changes that were mentioned in the previous responses, we were able to achieve most of what we sought out to include in the original proposal. The key change was switching the focus of the application from the rankings to providing the user with a wide array of information.

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

For future development of our application, there are several areas where improvements can be made, particularly in enhancing database interaction through a better use of stored procedures. Our current implementation uses stored procedures for filtering within the "favorites" section of user accounts, but we think we could expand this approach to the main search and ranking interface when users initially open the application, which would allow more usability. In depth, this means allowing users to sort and group universities based on various parameters through the interface. For example, enabling users to group universities by country or region, or to sort them according to specific performance metrics like graduate employability rates, would make the user interface more interactive and provide quicker, personalized insights that enhance the process for users. Also, along with this would come a heftier emphasis on UI/UX design for smoother user experience.

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

Our final division of labor consisted of group meetings for each project stage. We would initially meet over Zoom at the beginning of each project stage to figure out what work needs to be done for the respective project stage and sort of plan it out. Then based on everyone's availability, we would meet at the CIF periodically throughout the project stage to collectively complete the requirements. For example, for project stage 2 with planning our UML diagram and tables, we'd first meet and decide how exactly we wanted our application to look and once everyone was on the same page and knew what needed to be done/how to do the work, we'd split it accordingly based on how free everyone was. This means that we would make a document and list out things that needed to be completed and would either meet together at the CIF to work on them or do them virtually and meet back together to confirm. This is how the division of labor worked the whole semester and because of this all of us could learn every aspect of building our final project.