Project Report:

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 final project is nearly identical to the original proposal, maintaining all the functionalities including keyword search, paper recommendation, and leaderboard. Additionally, we implemented another section that presents the top 15 most-liked papers among the papers that the current user liked in the past 30 days. We implemented a gmail checker not specified. We did not implement the trending research section.

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

Our application successfully provides users most relevant economic papers based on their search query. It also suggests users with some top-liked papers from other users so they get to know what are some trending topics and papers. One restriction of our application is that it's restricted to only economic papers due to our original data. We would like to improve it so that it contains more topics.

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

We slightly modified our original data by applying the BERT and clustering algorithm to find group-relevant papers for our recommendation feature. We assign each paper a group_id to identify which papers are in the same group. This helps the recommendation feature so that after one user likes a paper, other relevant papers with the same group_id are likely to be suggested to the user.

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 have made several changes to our UML diagram based on the instructor's feedback. Firstly, we implemented the "appears in" relationship to map the relationship between the Papers entity and the Leaderboards entity, which is much more suitable compared to our previous design, where we had 10 paper_id attributes in the Leaderboards entity. Similarly, we added the “Likes” relationship to map the relationship between User and Papers. These changes adhere more to the database design practices and ensure easier maintenance.

Discuss what functionalities you added or removed. Why?

As mentioned above in the first question, we added a new section detailing the top 15 most-liked papers among the papers that the current user liked in the past 30 days. While the user will be able to examine the papers they liked under the “Profile” section, the information is very messy. We added this feature because we assume the users might want to retrieve the papers they liked before easily.

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

Our advanced database programs constitute the main functionalities of the application. For instance, our advanced SQL queries correspond to the functionality of keyword searching, paper recommendation, and popularity leaderboard.

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.

The first technical challenge we encountered was designing the UML diagram. It was hard to brainstorm and come up with five main entities, but it was good practice as we all gained a deeper understanding of the entity relationships. In order to efficiently build a functional UML diagram, we made sure to plan everything out in terms of ideas first, and know exactly how each component would interact with one another - once we had a clear idea of this, it was much easier to build the diagram itself.

The second technical challenge was implementing the four advanced queries. As these queries were supposed to match the key functionalities of our application, these queries were quite complicated and took us some time to implement. We experienced some issues getting the queries to work and to run on GCP - however, to resolve these issues, we made sure that we weren't flooding the GCP console with queries and made sure the queries were correct and performant in order to achieve success in GCP.

The third challenge was creating a workflow that allowed all of us to modify the same repository without any issues. We wanted to find a way that would allow all of us to work simultaneously in a manner such that if one person made an error, it wouldn't affect the runtime of the other teammates' applications. To solve this problem, we decided to each create a branch for different parts of the application, allowing us to work simultaneously and smoothly merge our code together into one final product.

The last challenge was the actual implementation of our application. There are many parts to it, including a user register/login-in page, a user information page, the search functionality, the recommendation system, etc. This part of the project really tested our overall understanding of database management and web development.

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

No, except for the additional functionality over the user's liked papers, our final application fulfilled all the requirements from the original proposal.

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

Future work for the FindMyPaper application could focus on several key areas for improvement. Advanced search features such as full-text search indexing, semantic search using embeddings, and support for boolean operators would greatly enhance usability. Enhanced paper recommendations could be achieved through collaborative filtering, content-based suggestions, and incorporating user interests and reading history. Improving data quality by validating papers, adding metadata like DOIs and publication dates, enabling user corrections, and verifying citations would build user trust. Introducing analytics features like citation analysis, trend tracking, research impact metrics, and recommendations based on citation networks could provide valuable insights.

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

The final division of labor for the project is as follows: Jason built the main framework for the website and got the initial setup running. This included the base application, connecting to the SQL and a basic UI. He also made the stored procedure for the recommended papers based on the clusters. Jack built the search engine using SQL query. Alex then built the login and register, and profile pages. He also built the liking and unliking system. He also turned the search engine into a stored transaction as well as handling merge conflicts from others' branches. Prajwal created the leaderboard functionality for the website, and created a stored procedure that allows users to query the papers that made the leaderboard. Jason and Jack were also the primary doers of the final report with Alex and Prajwal helping with less of the questions.