Final Project Report

1. **Please list out changes in direction of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

   We did not accomplish some functions described in the initial proposal, which are:
   - Users can like/dislike other people's comments.
   - Users can leave replies to people's comments.
   - Users can edit their comments.

   These functions do not directly relate to our main application, which are providing students the tools to look up a UIUC course's information and to rate and comment on the course.

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

   Our application is now functioning in terms of displaying a course's basic information, avg GPA, and many other information given the subject and course number. This is the most important aspect of usefulness. In addition, users can comment and rate the course.

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

   No. We have not changed the database schema nor the source of data.

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?**

   Since we have not changed the schema, this question is irrelevant.

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

   We did not add any functionalities beyond the scope of our initial proposal. We have removed the three functionalities described in part 1.

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

   Our advanced database program includes a stored procedure and a trigger.

   The stored procedure calculates the average rating based on all ratings for a course. To be specific, for one specific course, users can visualize the average quality, rating of the professor, workload, rubric reasonability, difficulty level, and the GPA received by the raters with the six corresponding progress bars. Such a stored procedure helps users to see important information about a course in a straightforward way.

The trigger is on the comments of the course. It automatically adds additional information about the newly-added comments under the circumstance that there are already a few comments for a course. Such a trigger helps users to read displayed comments with higher dimensional information and get better opinions of the course.

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.**

   Zhuofan: We had a difficult time trying to figure out how to connect our Flask app to the existing MySQL database. A lot of online tutorials were based on SQLite which is much simpler than MySQL. We ended up creating an engine using the “create_engine” function from sqlalchemy and a database connection string with pymysql plus our MySQL database credentials.

   Tim: We need to be super careful when writing redirect function names since when it gets wrong a totally different webpage will come up or there will be a connection error.

   FrankWe need to pay attention to the data structures in the Flask app, in our back-end python, and the result from MySQL. As we used python code such as for loop and if statements in the flask, we need to figure out what kind of data structures were sent by the backend. Without knowing it, the flask had difficulty showing our data at first.

   Ruipeng: As I was retrieving the initial data from the raw dataset, I noticed that the source data has compatibility issues (different types, formats), so I had much issues pre-processing or cleaning these data so we can fill them in our data tables. To resolve this issue, I wrote a python parser that specifically handles this issue, which took me a lot of time. So maybe next time we should be carefully selecting the source data and watch for compatibility with the database schema.

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

   Comparing the original proposal and final application, we did not include professors’ related information, which we may need to improve in the future and the interface could be better like the user’s interface should be more organized.

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

   In the future, we may need to add some authentication for certain operations like deleting a course or adding a course to prevent people’s mis-operation causing the total database changes. In addition, we would like to enable users to sign up their own account with user name and setting passwords so they can login and logout. Thus, we can add more applications such as users can add classes to bookmarks.

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

Functionality:

Functionality	Ruipeng HanHan	Houze Yang	ZhuofanJia	Kailiang Chen
Home Page		✔	✔	✔
Search UI		✔	✔	✔
Course Display		✔	✔	✔
Update/Delete		✔	✔	✔
Rating/View Rating of Course	✔	✔		
Comment/View comment of course	✔	✔		
Average Rating	✔	✔	✔	✔
Trigger			✔	

Data Base:

Functionality	Ruipeng HanHan	Houze Yang	ZhuofanJia	Kailiang Chen
Data Base Construct	✔	✔	✔	✔
Course Search Result SQL			✔	✔
Advence Query	✔			✔
Data Base Server Construct	✔			
Data collection		✔		
Data base design	✔	✔	✔	✔

Data base Implement	✔			
Indexing			✔	✔

In the final division of labor, beside original distribution table, we basically distritubed work based on experience of our teammate to do what they are good at. Also, we considered the time flexibility of each member. For example, if one member has an exam this week but his/her part is due at the same time, we can switch works in our team.