- **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).**
  Our final deliverables maintain the same direction as our initial proposal, which is to provide users with school application advice regarding their background.

- **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**
  In our final application, we have implemented features for account creation, login, and personal information modification. Additionally, we have developed a functionality to display the most relevant cases based on an individual's background. However, we did not implement the functionality to calculate competitive scores and admission probabilities. And we didn't provide the best matching school list.

- **Discuss if you changed the schema or source of the data for your application**
  Yes. In terms of data source, we transitioned to a new dataset that offers a more comprehensive range of data, including students' academic backgrounds, statuses, and application decisions. Most importantly, in contrast to the original dataset which contains only static data, the new dataset is authentic, it is directly sourced from gradcafe. The authenticity matters because we want to develop more accurate models for displaying results in the application. Consequently, our schema changed accordingly with the changes of the dataset.

- **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 did not modify the ER diagram or table implementations. Our tables were completed according to the structure designed at the beginning.

- **Discuss what functionalities you added or removed. Why?**
  We have added account login, creation, and deletion functionalities because this allows us to conveniently record user data to more accurately push matching cases for their reference. It also facilitates users in posting their own cases, which helps increase the number of cases in our database and improves the accuracy of future recommendations. When users no longer plan to use the website, we provide a deletion feature to clean up their personal information, thereby protecting their data privacy.

We removed the feature that displayed a list of best matching schools. This is because different areas may match with different schools, and since users prioritize different parameters, it is difficult to provide accurate feedback. Therefore, we chose to remove this feature and instead provide real cases for users to refer to.

- **Explain how you think your advanced database programs complement your application.**
  Our advanced database programs enhance the application by delivering accurate and relevant data for university recommendations. By using stored procedures and complex queries, we efficiently filter and aggregate applicant data, providing personalized recommendations and competitive insights for users based on their academic profiles. The data-driven logic ensures that the recommendation engine is accurate, scalable, and adaptable, allowing users to receive tailored university suggestions while understanding their standing among other applicants.

- **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.**
  Qi: A significant challenge I encountered was setting up the environment of our Node.js server. This is the first time I interact with a platform like GCP. There were many nuances that were not covered in the instruction video, particularly the differences in managing package dependencies between my local machine and the GCP virtual machine. My advice was to .gitignore Node.js modules and install dependencies separately on each platform to ensure compatibility. This should avoid potential conflicts that arise from package version discrepancies across different environments. I spent hours figuring out how to configure Redis in CommonJS. Check out our Github for the correct command, don’t listen to Stackoverflow or ChatGPT, many answers are outdated there, surprisingly.

  Xiaotian: The technical challenge I encountered was selecting the most suitable algorithm for our project. Initially, I hadn't considered Random Forest. I explored its advantages over alternatives like KNN, particularly its robustness to overfitting and handling of missing data. After evaluating our dataset's characteristics and performance requirements, I chose Random Forest for its ability to handle categorical data effectively and produce reliable predictions in our context. This decision was crucial for optimizing our model's accuracy and scalability.

Jiadong: I encountered a significant technical challenge while trying to run Python code within our Node.js application. I initially relied on PythonShell.run, which didn't work in our setup due to no output. I spent considerable time troubleshooting before ultimately switching to using the exec function to call Python directly. This approach required careful command formatting and output parsing but ensured consistent data exchange between Node.js and Python.

Yuhui: While developing the user profile page, I faced several challenges. One key issue was achieving seamless integration between the Node.js backend and the React frontend, which required careful API design for effective data exchange and state management. I think it’s a good idea to ensure that API endpoints are consistently structured and adhere to RESTful principles. This consistency will make it easier to manage data exchange and reduce the complexity of the integration. And also establish regular testing phases where both frontend and backend teams can test and provide feedback on the integration. Automated end-to-end tests can also be very helpful in catching integration issues early.

- **Are there other things that changed comparing the final application with the original proposal**？
  Firstly, we did not display a competitiveness score based on personal backgrounds. Secondly, instead of displaying the most closely matched schools, we showcased matching cases. This was due to the incompleteness of our database, insufficient data volume, and the lack of suitable formulas/models for calculation. Therefore, we chose to present cases to provide users with the most authentic data for their reference.

- **Describe future work that you think, other than the interface, that the application can improve on**
  Firstly, we can use more machine learning models to predict admission rates and provide users with cases that better match their backgrounds. Secondly, we can offer a feature to bookmark cases, making it easier for users to find their preferred cases. In addition, we can allow users to select their area and schools in text form, and they don’t need to refer to a guidebook to fill these out, enhancing the user experience.

- **Describe the final division of labor and how well you managed teamwork.**
  Qi: My role involved managing the server, which includes overseeing the overall structure, behaviors, and versions. Specifically, I was responsible for

implementing the login and profile creation logic. In addition to developing various APIs and templates, I also handled a transaction as outlined in the project rubric. Apart from that, I also maintained the session logic with Redis.

Xiaotian: In the final division of labor, I focused on implementing the Random Forest algorithm for school selection based on database information. My team members handled data preprocessing, feature engineering, and model evaluation.

Jiadong: My primary role was to build the API for the recommendation system, connect it to the machine learning model, and then return the output to the user. In addition, I complete the stored procedures and triggers portion of the rubric. My work ensures reliable data exchange between components and a consistent user experience.

Yuhui: In this project, I am tasked with developing the user profile page, which involves both the backend and frontend aspects using Node.js and React, respectively. My responsibilities include designing a user interface that effectively displays personal information. I also implemented features that allow users to update their data, such as contact details and preferences. Additionally, I created a functionality for users to post their personal cases, which may involve complex form handling and data validation. Lastly, I developed a secure and efficient process for users to delete their accounts if they choose to do so.