Stage 4: Checkpoint 2 & 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).

- There are no significant changes in the directions of our project from our original proposal.
- There were two significant differences from the previous direction:
  - We added in additional rating fields like food rating and healthcare rating separate from climate rating to allow users to add in additional information about each country
  - We added in an enhancement for giving extra weight to the climate ratings of frequent users. If a user provides many feedback responses on trips to different countries, then their climate rating is given double the weight.

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

- We believe that our application is successful in providing climate related information to the user. We provided a lot of functionality for users to provide their feedback about different countries in different durations. In the rankings, we implemented queries to calculate different metrics to let users know about the country's climate impact. We provided an easy to use map visualization to allow users to find different countries to click on. We also provided a search bar for users to search by country name or city name. After clicking, we provided a great dashboard for users to learn about each country's climate and economic data as well as the average ratings of each country in different categories. Our tool could be valuable for people to learn more about each country's climate impact as well as make informed discussion when visiting them. Students and researchers can use our tool to research each country's climate impact.
- Our app also has a few limitations. The country data is from a Kaggle dataset, and it is not updated automatically. One way to fix this is to add a web scraper to scrape new data and update the country data automatically. The country visualization can be improved too by potentially adding red circles to locations with high climate impact to allow users to visualize climate impact by location. However, we couldn't implement this due to time constraints.

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

- We didn't make any changes to the source of the data for our application. We used the same Kaggle datasets from stage 2.

- One main change we made to the schema was to add the isFrequent attribute to UserInfo. When the user has added 5 UserInputs, the isFrequent field is set to true using a trigger, and the user's climate rating will be given double the weight as outlined before.

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?

- Some changes we made were:
  - Added AUTO_INCREMENT to primary key fields to generate the primary key automatically.
  - Widened the password field to be able to store the password hash and salt
  - Added constraints like enforcing DateVisitedForm comes before DateVisitedTo
  - Added the isFrequent attribute
- We think that the final design is more suitable. Initially, we had given a short VARCHAR width to the password field as we thought we would store them in plain text, but in the final design, we widened the password to store the hash of the password, which is more secure. We also added additional constraints to enforce data integrity. We think that giving more weight to climate ratings of frequent users will give more accurate results.

Discuss what functionalities you added or removed. Why?

- Added functionalities:
  - Added password hashing to prevent storing passwords in plain text and make the application more secure
  - Added a transaction to check for overlapping feedback time durations and automatically fix the time duration if possible
  - Added a ranking stored procedure to give a text climate ranking to each country
- Removed functionalities:
  - We initially planned to display more data in the map, but it proved too challenging to implement within the time frame.

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

- 4 advanced queries:
  - In stage 3, we designed 4 advanced queries as well as indices to speed them up. We used those queries to display the climate related data like energy deficit and those countries with higher than average co2 emissions and lower than average forested area.
- Transaction, stored procedure and trigger
  - InsertUserInputWithOverlapValidation transaction (inside a stored procedure): Users shouldn't be allowed to insert feedback with overlapping time durations, and initially we had made a trigger to reject a violating feedback. Then we

decided to make this stored procedure to attempt to find a way to change the time duration so that it can be inserted into the table. This is more user-friendly than rejecting the request.
    - CalculateCountriesWithWorstClimateImpact stored procedure : takes in start and end dates and outputs the text ranking. It takes the rankings from many queries and combines it into a single text ranking to be displayed to the user.
    - UpdateIsFrequentInUserInfo trigger: updates the user's isFrequent attribute when the user has 5 feedback at any time. When this is true, the user's climate rating is given twice the weight, which we think will give more accurate results.
- Full text search:
    - We added the full text index to the country name, abbreviation, capital city and largest city columns. When the user typed in a location into the search box, the full text search gave relevant search results to display to the users.

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.

- Anupam - setting up the MYSQL database was quite challenging. The Kaggle datasets had many formatting issues like malformed characters and extra commas within column data like the city names. The Kaggle dataset was able to be read correctly by Pandas but when importing using the MYSQL import tool, there were many issues. The way to solve the issue is to read the data using pandas and then connect to MYSQL and use the to_sql function in Pandas as the MYSQL import tool isn't as good as Pandas.
- Ayush - this was my first time learning full stack development. I found React and Redux toolkits quite challenging as there were many concepts like react hooks, functional components, reducers and async-await. I found that the best way to learn React is through YouTube tutorials and websites like w3schools.
- Sayuj - I did have some frontend and react experience from before but as this was a large project, we were also using redux toolkit, which is new to me. Redux toolkit is a more advanced way to manage data in the application, and I learned a lot when using it here.

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

The primary thing that changed for the final application was the addition of the user input database compared to just a comprehensive dashboard of countries' climate impact. Additionally, one of the original proposed features we were thinking about implementing was a comparison mode where users can select two or more countries to compare queried economic and climate data between countries. However, not only did this present a challenge in the frontend, we realized that this would limit our ability to add user input and also not allow for more complex queries or triggers.

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

The country data is from a Kaggle dataset, and it is not updated automatically. One way to fix this is to add a web scraper to scrape new data and update the country data automatically. This way the user will be able to access updated climate and economic data about each country.

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

Anupam - I designed the user input page, and developed the backend controllers to handle user interactions as well as the trigger for the isFrequent tag. I also set up the Cloud SQL database.

Ayush - I helped design the database schema and collaborated with Anupam on the Data Definition Language (DDL) scripts, stored procedures, and advanced queries. Additionally, I also designed the main dashboard frontend and connected the API endpoints to the backend country geo information.

Sayuj - I focused on improving the usability and overall design of frontend components, ensuring an intuitive and engaging user experience. I also worked with Ayush to assist Anupam in connecting the login and registration pages to the backend, facilitating more robust user authentication and authorization.

Despite the setback of a team member's departure, the rest of us were able to effectively manage our responsibilities and schedule meetings with each other to discuss questions, concerns and struggles - helping us ensure that project checkpoints were met and quality standards were maintained.