

ClimaCloset - Weather Based Outfit Suggestion App
Jack Moran, Ben Goldman, Livia Bezati, and Reda Mehdaoui
Stage 4 - Team 085 - April 28, 2025

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 main addition we made was to add a restaurants tab and table within the database. At the time of the submission of stage 1, we had yet to make a diagram with the five tables that we would include in the project. We came up with the restaurants table while working on stage 2 in order to add additional functionality to the app and satisfy the requirement for 5 tables.
- Instead of using current weather data and forecasts we decided to use a dataset with the average temperature and weather for every single day of the year, for every single location on the User table. This allowed us to simplify the approach in order to create a useful and working version which could be later expanded to include truly live weather data.
- We decided that the clean/dirty/wash tag on the clothes was not necessary since the user may not like to come back later and update the tag or use the app every time they would like to pick a new outfit for the current weather. In such a case the tag would create a false idea of cleaned clothes when they were not, or dirty clothes when they had already been cleaned and thus constricting the options for our program to choose from.

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

We believe that our application met all requirements regarding its usefulness. We believe that the application we constructed for this checkpoint is fully functional and could be used by users as we originally intended. However, we also recognize that if given more time, we could further expand on this implementation to be even more useful in terms of recommending outfits.

We believe that this application accomplished its original goals. This included allowing many different users to login using unique login information which can also be modified by the user. They can then view all clothing items which they own, and add or delete these items if necessary. Users can also view their outfits which the application has recommended in the past or that the application might use for recommendations in the future. From there, the user can go to the home page where they can have ClimaCloset recommend an outfit for them based on their location, the current date, and past weather data. Finally, we added an additional restaurant feature. This allows users to find restaurants, within their city, which matches the style of their outfit for them to visit. In the end, we believe that we met all requirements and completed the application as we intended to from the beginning.

Although we built a useful application that works as intended, we also believe that this working model could be expanded to make it even more useful and intuitive for users. Some of these potential changes and expansions will be further explained below.

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

- We found average weather data for every single day of the year per a single location on a government website. Since we had a .csv file for every location, we took the time to combine all of the location weather data into one big weather dataset. Additionally, we got rid of extra columns of data that we did not mean to use, to save space and optimize our program.
- For this checkpoint, we added the restaurants table within the database and its associated feature. In order to make sure this was as useful as possible, we had to make a few modifications in order to make sure it matched well with our other data. Ratings from the Restaurants table were converted from a string (ex. "4.1/5") to a decimal (ex. 4.1). This was done for easier comparison between restaurants. In addition, this dataset included restaurants from cities throughout India. As this was a complete and good dataset, we made modifications to say that they were within the cities offered by ClimaCloset. For this reason, the websites for each restaurant will show a different location than through our app as this dataset is for a proof of concept and would need to be modified to be more accurate to the actual cities.

Discuss what you changed 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?

From our original ER diagram, we removed some of the attributes from our tables that we thought were not necessary and that we weren't going to use. For example, we removed some attributes relating to the restaurants that we thought were not useful for the users of our app. In addition, we simplified the attributes of the weather table to allow for easier outfit selection. Otherwise we didn't make any major changes. We spent a lot of time devising the ER diagram, and thus we didn't feel the need for any dramatic changes in stage 3 or 4.

Discuss what functionalities you added or removed. Why?

We did not add or remove any major functionalities from the ones that we specified and explained in stage 2 of the project. Again, there were some modifications made to specific attributes but this still allowed for the completion of the application as specified. The main function we added from stage 1 would be the addition of the restaurant feature. In addition, the only major feature we left out was the use of dirty and clean tags for the user as we thought that it would simply complicate the use of the app and make our application not as intuitive for our desired users

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

We believe that the use of advanced database programs helps our application manage data more efficiently. The ability to use complex queries helps us search, analyze and return the data however we want, while maximizing performance and speed. Additionally, we have a number of constraints, triggers and stored procedures, that allow us to make rules and maintain valid relationships between tables. They allow us to make sure data is not lost or duplicated or modified without permission or intent. Ensuring that our database was properly set up prior to building out the application allowed us to construct the app very fast while maintaining proper data requirements.

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.

Reda: One challenge we encountered was modifying a table after we uploaded it, i.e. choosing to adapt a column by doing it manually using excel and the csv format, or by directly modifying it in the SQL console through everything we learned in the class. I would recommend doing the latter, although depending on comfort with excel, there's a good argument for using it. Once we became more comfortable with the SQL console, we were able to build out the database much faster.

Livia: Another challenge we faced was that the weather data we found was split into many csv tables with one per each location. We were having a hard time adding the other csv tables after we had inserted one. So we decided to combine all of them into one csv file with the help of excel and then upload only one csv file onto the database. This helped to simplify our database and allowed us to add this large dataset without having to modify it within the SQL console.

Ben: We have limited experience with connecting frontend and backend together for a working application. Although there was a video to help, we would suggest starting early because figuring out this part took time. We took advantage of office hours for technical assistance getting set up and then learned much of this through trial and error.

Jack: We also really struggled to initially set up the VM for our application and connect it with our website. There were no provided videos on how to get this connection setup. Therefore, we spent time in office hours to get help throughout this process. Similar to Ben, I would recommend starting early and using online resources to figure out how to properly set up the app.

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

As noted in prior sections, the main addition since stage 1 was the restaurants feature. This ensured that we had 5 complete tables within our database and allowed our application to

recommend restaurants to the user based on their location and current outfit. The other main change we had was the removal of the dirty and clean tags for clothing. We removed this aspect as we believe it would have complicated the application for the user and would have led to a more negative user experience.

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

It would be helpful if the application can obtain more data so that it is more accessible to more people. For example, if we can add data from other countries other than the US, then we can gain users from those countries as well. This could also be expanded to use an API to gather real time weather data based on a user's location. This would allow for more accurate outfit selection and could really improve user experience. Another thing that could be worked on in the future is to add more attributes and relationships within the database, so that the decision making for outfits and the suggestions made by the application are more complex. Our current outfit selection process is done through querying based on the current weather temperature. However, if we could add attributes like rain or wind into this selection process, we could really improve user experience.

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

Overall, we believe our group did a good job of managing labor. Over the course of the project, we primarily worked as a group, meaning that we met up in person to work on the different stages and the final code. The final division is detailed as follows:

Jack - Assisted with all documentation for each stage. Set up the VM for the application. Worked to add and set up database tables including users and clothing items. Assisted with building out the application when working in a group including creating some of the advanced queries. Assisted with group project report and video.

Ben - Assisted with all documentation for each stage. Worked to add and set up database tables including outfits and clothing items. Primary coder of the application. With assistance from group members, built our front end, back end, and added database functionality to the application. Assisted with group project report and video.

Reda - Assisted with all documentation for each stage. Set up the SQL cloud database for application. Worked to add and set up database tables including restaurants. Helped to code the application. Expert with the front end coding and helped to add database functionality. Assisted with group project report.

Livia - Assisted with all documentation for each stage. Helped to come up with the database schema and find good sources for data. Worked to add and set up database tables

including weather. Assisted with building out the application when working in a group including creating some of the advanced queries. Assisted with group project report and video.