

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

We changed from a emphasis on listing car prices to listing popular cars based on user likes and overall rating. The final application had a larger emphasis on car ratings and popularity.

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

Our application achieved the initial goal of creating a car searching app, by allowing the user to successfully search for cars by filtering for features of their choice. We were also successfully able to implement the custom rating system where we assign each car our own custom rating based on an algorithm that takes multiple features into account. One aspect we failed to achieve is creating the History table, making it slightly less useful for the user since they couldn't see what cars they picked or were previously interested in.

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

We did not change the source of the data for the application. The original from Kaggle was sufficient. We also modified our schema by adding two new tables to our functionality dealing with Reviews and car ratings. More details about these changes are found in #4.

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?

One thing we changed was not including a history table. Instead, we made a custom review table that held our website's car rating for each car. We thought this would be more useful information for the user. We also added a Reviews table, which allowed users to leave reviews for cars and give upvotes or downvotes on the cars. We think our final result is more suitable for our application because we used this new Reviews table to generate lots of useful information for the user such as average ratings of certain cars.

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

We added the upvote and downvote features because we realized we had no other real effective way to "Update" any of our databases, so we settled on a "upvote" system allowing the user to upvote and downvote cars. We also added the "star" feature which indicates whether a model is on average rated highly or not which will be diffused more below. We removed the history table because we decided to change it to a Reviews table for simplicity's and times sake.

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

The stored procedure improves the usefulness of our application because it provides users with aggregated information about cars that is specific to our application. We used the stored procedure to generate a “CarEZ score” for each score, which used our own unique formula to rate cars. This is something users will not find on other similar websites. Our trigger classifies car models as either “starred” or “not starred” based on aggregate data on upvotes of that specific model. This is useful to our application because it gives users an idea of which models are most liked and least liked by other users on the platform. In addition, both of these features update as more information is entered into the database, so they will always have up to date information.

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.

Om: When we were trying to join two different tables, they would try joining on the Primary key however, since they had a column that was named the same and not joined upon, it outputted nothing. Therefore we had to delete one of the Columns from one table and use the value from the other table.

Andrew: we had a problem initially figuring out how to link the buttons. We were all unfamiliar with flask and ended up spending a lot of time trying to figure this out, however this could have been avoided with gaining some slight flask background knowledge prior to the start of the project

Connor: When creating the trigger, we had problems because our trigger was “on update” of the Reviews table, but inside the trigger, we called updates on the Reviews table. This caused us confusion and bugs because we set off an infinite cycle of triggers being called, since it was calling itself. We fixed this problem by only updating the new, updated row, and not touching the rest of the table inside the trigger, and updated the rest of the table elsewhere.

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

One of the more significant differences between the original proposal and the final application was the visual layout of the user interface. In our proposal, we intended on having images of each of the cars that were searched up. However, assigning an image to each of the thousands of cars in our database was something we decided was unnecessary, and we added various other visual features such as the bar graph instead. Another change we made was focusing more on reviews and feedback about cars

instead of user account histories. The upvote and downvote functions were not in the original proposal, we came up with them as we developed the application because we thought they made good additions to the existing functionality.

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

We can definitely improve our stored procedure time since it does take a few seconds to complete. Users waiting for a query is the last thing we'd want.

Another thing we would want to make more useful is the account management system. We'd want to incorporate user accounts more into our website to be able to keep track of who wrote reviews and keep track of user search history.

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

The labor was split evenly since we would get together everytime we needed to work on it. Majority of the work was done in person with all three group members present.

Demo video:

<https://drive.google.com/file/d/1MYGf2mgGGnbpCajATSZQDI26jZm-oT/view?usp=sharing>