STAGE6: PROJECT REPORT AND DEMO VIDEO

For this stage, your team is expected to submit the final project deliverables, including a project reflection report and a video demo.

A.)

Your project reflection report should contain the following topics. The report would be graded on completeness and correctness:

1. 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).

Generally our project proposal was similar to our end product with a few exceptions we will outline below. Firstly, our proposal creative component was a favorites button which the user can add a selected video to their favorites list. We included this in our final project but not as the creative component. Our final project creative component however was a visualization bar graph displaying the top categories that were added to favorites by different users. Additionally our UI mockup was somewhat similar layout wise but as stated previously we added the visualization creative component as well adding separate pages for the users favorite videos the main search homepage and the visualization to help cater the user to find the video they truly want to watch

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

We think our application provides a very interactive YouTube like search tool in which a user can look for YouTube videos that are popular. We achieved this by adding filters to the amount of likes, views etc while making the experience more catered towards the user rather than YouTube's non-transparently generated recommendations. In our application the user has more control to find what videos they like by simply searching and our only filters we apply are like and view count to give the user what they want. This is different from YouTube since their algorithm depends on factors like videos the user has seen and other unknown factors the user is not even aware of.

What our application failed to do is develop a complex algorithm much like YouTube's algorithm where the user's past videos watched are taken into account and put into a recommended page. While I am sure YouTube's algorithm relies on unknown data trends and factors that probably involve psychological motives I think it would still be cool to add some sort of complex algorithm into our application. This would be a cool feature to implement if we had more time but our recommendations for now just depend on a simple search tool with filters and a visualization showing the top categories that have been selected to favorites by the user.

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

The data source did not change; we still ended up using the YouTube Trending Videos dataset from Kaggle. Our Schema also somewhat changed as we implemented a favorites table which contained the userId, videoId, playlistName and addedDate. The favorites table was the main relation updated accordingly when a user added a video to favorites or removed the video from favorites. We also added the HotVideos table which stored the videos that were most favorited by users, given that they met certain criteria.

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?

With our ER Diagram, we originally made the Category a separate entity but we realized we could put the CategoryName as an attribute of Video since each Video belongs in a Category. We had to update this change in the ER Diagram and our table implementation since we originally had a table for Category but had to remove it and put the CategoryName attribute in the Video table. Having the category name as an attribute of the Video entity is a more suitable design since it reduces the amount of tables needed to be accessed and it adds more functionality for the Video entity since we can easily see the category the Video belongs to.

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

We removed the functionality of filtering the results by the time period in which they were trending or by when they were uploaded so the users can see recent videos that match what they’re looking for or look for videos irrespective of that. We did this because we only have the published date of the video and not the date it was trending and we believed that it wouldn't have improved our application. We also added a HotVideos table which displays videos that are trending that fit a certain criteria so users are able to see videos that other users also like and that are well liked and have a large amount of views.

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

We used the stored procedure and triggers for our advanced database programs. This has complemented our application because we were able to avoid massive SQL queries and simply call a function. We were able to call the procedures to either create a table that displays popular videos which were videos that match certain criteria such as having a like to dislike ratio above 0.95 or to count the number of favorite videos for each video category and display the top results that the user can specify the amount. We also chose a trigger so that our application would update the table immediately according to an action which in our case was favoriting or deleting a video from favorites which would immediately update other attributes or tables. This allowed our application to update other information if necessary. The playlists would be updated immediately which removes any errors.

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.

Itay:

When Yu and I worked on backend I was not too familiar with Flask library and the syntax but after Yu explained an example of code in detail I got the hang of it and we were able to develop python APIs for the 4 CRUD sql operations and it was cool to see it work.

Yu:

During the development of this application, we met the Cross-Origin Resource Sharing(CORS) problem. To be more specific, when a web page attempts to make a cross-domain request without proper CORS headers, the web browser will block the request, and the server will not receive the request. To solve this problem, we used http-proxy-middleware, a Node.js module that provides a proxy middleware for use with the Express or Connect web frameworks. This middleware is used to forward HTTP requests to a remote server, typically used as a way to proxy requests from a client to a backend API. This resource[1] is very useful to help us solve this problem.

Shreya:

Working on the SQL queries was something that I partially struggled with since I did struggled with writing them to produce the output I needed. However, after looking at the notes on SQL queries from class and looking at SQL keywords and what they do, I was able to understand how to properly write them based on our implemented tables.

Shams:

One technical challenge that I encountered was when working on the frontend, because our website had a lot of styling and that was confusing for me when I made components within other components and the styling would extend to the inner components in ways that I did not expect. I would recommend not worrying about the styling too much until all of the foundational parts of the project are done, and simplifying the code as much as possible.

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

We added a bar graph that is a visualization of all the video categories and the number of videos that all users have liked in each category. We wanted a level of personalization since the user can be able to see what other users have favorited and it may influence them to look at the top categories or discover categories they haven't looked at previously.

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

- We would set up the database to update with more recent YouTube videos as they come out so it’s not a fixed database

- We could implement a way to watch and interact with the videos on the website instead of just linking them (link it to their YT account somehow)
- We would store more information relating the that specific user (their likes, what types of videos they spend time watching) and change our procedures to recommend more relevant videos to them
- Modify one row at a time for changing playlistName rather than across the whole database.
- Deploy the application to a cloud service provider, such as GCP, AWS, or Azure.

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

The final division of labor was overall equal as we decided to set up calls and work on either code diagrams or report typing depending on the stage we worked on. We would also meet in person at times especially for coding as it was an easier option to collaborate and help debug the code if problems arise which plenty did especially during the backend development phase. While the proposal stated front end would be done by Yu and Shams while back end would be done by Shreya and Itay, Yu and Shreya swapped roles so our role assignment was flexible but still equal. Yu also did the majority of the github pushing and stage release submissions since he was very familiar with Github control, so the rest of the group tried to compensate by working more on the reports and the written parts. Overall if one person did a lot of work in one area, another would make up any slack in another so the group dynamic was very good.

Reference

1. CodeofRelevancy. (2021, February 24). How to Set Up a Proxy Server in React - Dealing with CORS. Dev. https://dev.to/codeofrelevancy/how-to-set-up-a-proxy-server-in-react-dealing-with-cors-87e