Link to the repo: <https://github.com/cs411-alawini/sp23-cs411-team015-WeLoveDB>

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).

During the development of stage 5, the team decided to switch the objective of the creative component from providing a comprehensive report on the video’s statistics and predicting its trending potential using machine learning algorithms to a scatterplot data visualization on each queried video’s view count and like count, due to the lack of training exclusive data we acquired. (The entire Youtube Video Dataset is utilized in our application, therefore, no data can be used for training machine learning algorithms.)

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

Our application achieved the usefulness of aiding users to recall memorable video content. Our functionality of tag saving and video name saving are the two ways for users to memorize their invaluable experience while watching their favorite videos. However, our application does not achieve the functionality of querying tags and returning all videos containing the queried tag. This limitation posts a significant reduction of searching power, making the usefulness of tagging not as good as that of videos in our website.

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

We did not change the schema or dataset for our application.

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?

The only difference between the original and final design is making Videos a weak entity set of Channels. Even though every Youtube video has an unique video id, once a channel is deleted, all of its videos will also be deleted. Hence the final design is more suitable.

Discuss what functionalities you added or removed. Why?

We added a lot more data analysis to our frontend after our initial design. We ended up adding information about videos that have been favorited, and we added data analysis to whenever a user searched for a video. We did this because we felt like users would enjoy seeing information about the videos that they and others are watching or favoriting.

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

We have two advanced queries. The first finds videos which either have more likes than the average number of likes or more views than the average number of views. The second finds the most popular tags of trending YouTube videos.

Our database and website are designed around the constantly changing status of trending videos. Our stored procedure provides valuable and interesting information for the users of our website. It provides which videos are being favorited and information about each video. Users are able to see what videos that other people are watching. We have a column that shows if a video has more likes or views than the average trending video. We are also showing users a “uniqueness value” which is a percentage of the top 15 most popular tags that the video contains. Users could see a video with a low percentage (meaning it does not contain many popular tags) that is above average and recognize that that video might be special, as it has been viewed many times while also being very different from most videos.

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.

**Tinrey Du**

During the creation of the advanced database programs, we encountered a fair amount of bugs. This was due to trying to create the entire stored procedure before testing. We also discovered a few bugs in our front end when trying to utilize the stored procedure. It is important to not only start early, but to work in parts. You should not try to implement the full functionality of something and then test it.

**Huiqian Jing**

When implementing the API for returning the video and its tags by video id, the results returned by the SQL query contain multiple results with the same id, but only differ in tag id and tag description. I didn’t notice that multiple results were returned and I only sent the first matching result to the front end. In the front end page, I found that each video only had one tag. And then I ran the sql query and found multiple results were returned. I wrote a helper function to combine the tag id and tag descriptions and saved them in an array. And then returned one result with combined tag id and tag descriptions to the front end.

**Davis**

Initially, we had difficulty connecting the backend to the database. Nobody in our group had much experience in backend development. My advice would be to start earlier and get help from TAs in office hours if you are stuck.

**Kevin**

In the proposal stage of our team project, we have neglected the difficulties on creating a ‘tags’ data table that is fully integrable to the ‘videos’ table, which is the key feature in the later stage of development on implementing the tag-saving or tag-searching functionalities. We encountered a couple failures in the SQL database all regarding defining the schema of ‘tags’, and handling the null values within the original dataset when the ‘tags’ table is created. We even created a couple auxiliary data tables in our database just to accommodate this issue, but this then added more complications in developing tag-related functionalities in the final web application. Therefore, if we were aware of this issue in the very beginning and had performed data cleaning in the original dataset, it can alleviate so much in the data integration stage to make more clear and coherent tables.

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

In the original proposal, we wanted to implement searching on the tags, view counts, number of likes and also on users’ favorite video lists, but in our final application, we only implemented the search functionality on the video title. And we wanted to exact the tags associated with the user's favorite videos and save that as the user's favorite tags. In the final application, we used another way to let users add favorite tags.

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

The original dataset is updated everyday, but our database is not updated when the original database changes.

We could include the number of times the video went on the trending page in the video database. And we can improve our user authentication process both in the backend and frontend. In the final application, we just stored the user id in the localstorage, which is not safe enough. And we could encrypt the user id in the backend and save in the frontend instead.

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

After we all initially worked on coming up with the basic design and structure of the website and backend database, we split our team into a backend and frontend group. Throughout the semester, we would all set goals for the group to reach so both teams would be at about the same level. Eventually, when we were ready, we were able to connect the frontend and backend. For the most part, we were able to manage our teamwork well. The only issue was initially connecting the frontend to the backend. After this small obstacle, we were able to get right back on track. We all kept each other accountable and would check in on the progress of each team.