YouTube Search Premium: A more flexible YouTube search engine

Summary

We are planning to use the YouTube Trending Video Dataset from Kaggle.com. This dataset provides various information such as views, likes, tags, and more about daily trending videos on YouTube. We plan to utilize this dataset to create a video querying service and host it on a website. Our inspiration for proposing this website came from one limitation of YouTube's search engine. Only video titles are queryable; any other metadata of a video such as view count, like count, comment count are unsearchable in the traditional YouTube searching process. Therefore, our project idea will provide extra flexibility to users when searching for videos, without them having to explicitly memorize the title.

Functionalities

Implementing a video querying system requires a couple of vital functionalities. First and foremost, users should be able to search for trending videos by certain tags, view count or number of likes. These tags are a list of predefined keywords stored in the dataset's "Tag" column, and we need to extract all distinct tags from the dataset in order to let the user to "select", instead of "search" in a traditional search engine sense.

Another feature is allowing users to create a favorites playlist as a table that can be inserted, queried, updated and deleted, thus supporting the CRUD functionality of the database. Both at the time a user created the account, or when the user had completed his/her/their first query, the system will prompt the user to enter a custom list of tag preferences he/she/they may endorse. This is not mandatory but to enhance the user experience in future queries. After the user completes this action, the system will create a separate table entry storing the user's preference of videos, reusing them in later querying processes. Alternatively, the user can choose to save the queried videos, and our system will parse the saved video(s) as a list of tags again into the preference table.

Once the list of queried videos is returned, there will also be data visualization that allows users to better understand the trends and popularity of the YouTube videos they are interested in. Additionally, the user can sort the returned query by chronological order or one or more of the video statistics (likes, comments, views, etc.) A rough sketch of the UI is drawn below.

We are also going creative on extending a data analysis component from the basic querying functionality. This is to examine and model the relationship between trending status and the genre or tagging of the video. For each trending video returned by the query, we will analyze its potential popularity given the statistics from the database such as view count, number of comments and likes, and most crucially, how quickly did the video get on the trending page. This cannot be achieved by the basic data from the dataset, so some degree of statistical learning

algorithms such as multilinear regression models and k-means algorithms is needed to achieve this task.

Usefulness

The usefulness of our project lies in its convenience and flexibility when searching. The number of videos one has watched each day is massive. If the user had found one video particularly memorable and wanted to re-watch it again after a couple of days, it might take a while to find in his/her watch history. Also, it might be difficult to remember what the video title/description is, hence searching using other metadata might prove to be a more flexible alternative. We believe using tags is especially useful. This can be very helpful for those users who are not very good at memorizing large chunks of information, like elderly people, young-kids, or users having memory-impaired diseases. We believed that our tag-based querying application rarely has similar examples in release, as the search engine does not return anything close to our application-in-proposal.

Realness

This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the IN, US, GB, DE, CA, FR, RU, BR, MX, KR, and JP regions (India, USA, Great Britain, Germany, Canada, France, Russia, Brazil, Mexico, South Korea, and Japan respectively), with up to 200 listed trending videos per day. Each region's data is in a separate file. Data includes the video title, channel title, publish time and date, tags, trending date, views, likes and dislikes, description, and comment count. The data also includes a category_id field, which varies between regions. To retrieve the categories for a specific video, find it in the associated JSON. One such file is included for each of the 11 regions in the dataset.

This dataset is updated every day and collected using the YouTube API.

https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset?select=US_youtube_trending_data.csv

Since the dataset is obtained from a Kaggle user who directly extracted such information from YouTube with exact video name and publishing time, we are assured that the authenticity of the content will reflect real-life YouTube video genres. The user-entered data tables utilized as preference enhancement would also be updated directly from the prospective users of the website at deployment, thus making the data trustworthy. (To ensure this, we may need to prompt users to login).

Work Distribution

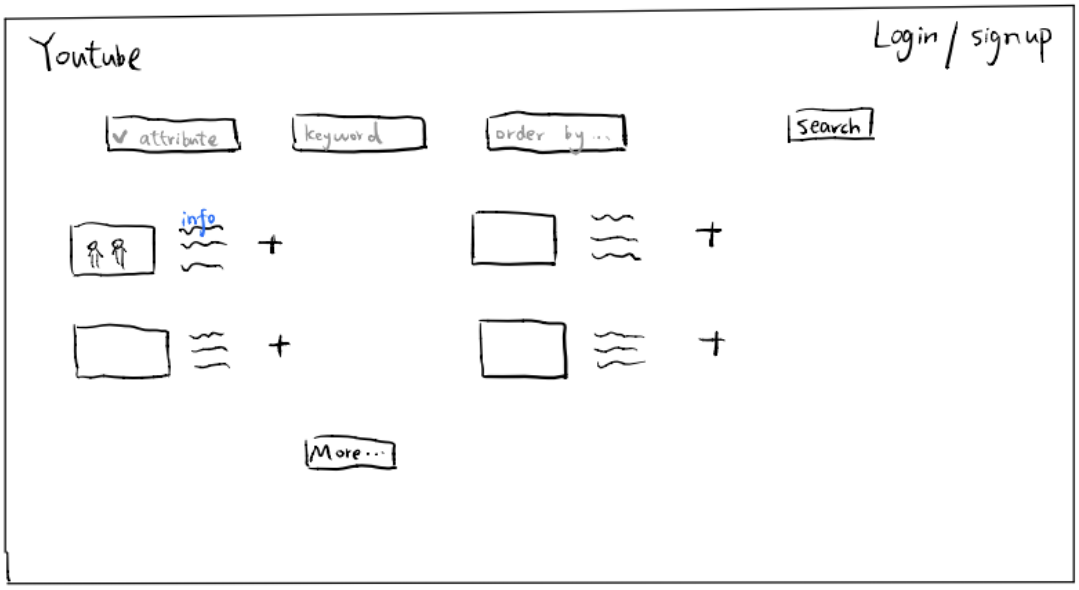
Huiqian will be mainly responsible for frontend development, while Kevin, Davis and Tinrey will be responsible for backend.

Huiqian will be working on data visualization and user login/authentication. Kevin will be working on data analysis. Davis will be working on database querying based on the user's inputs, while Tinrey will be working on creating the playlist of videos for each user.

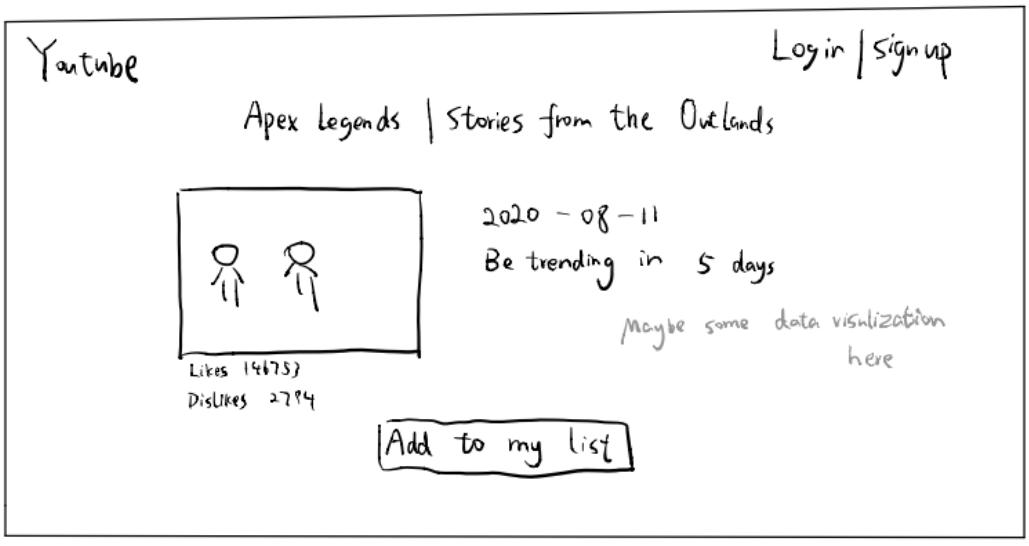
This doesn't mean that there will not be collaboration amongst group members. We will all be helping one another since there is overlap between our tasks, but the previous paragraph highlights which functionalities each member is taking the lead in.

UI Mockup

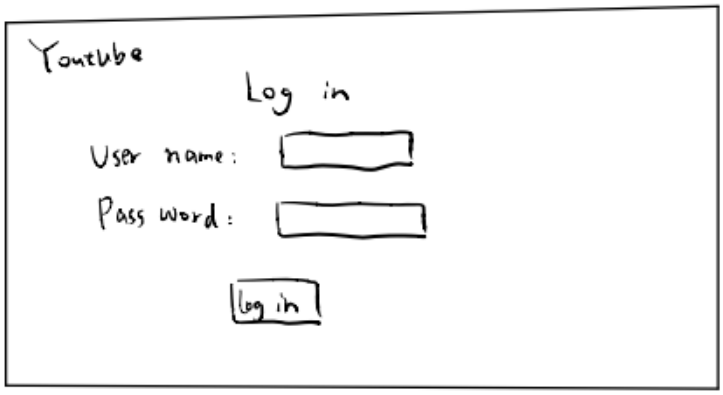
Homepage "/"



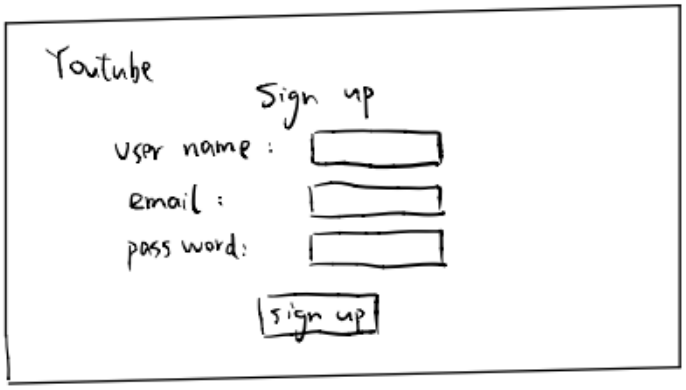
Video Detail page "/ <videoid>"



User login page "/login"



User sign up page "/sign up"



User detail page "/<user id>"