DBMS Project Report

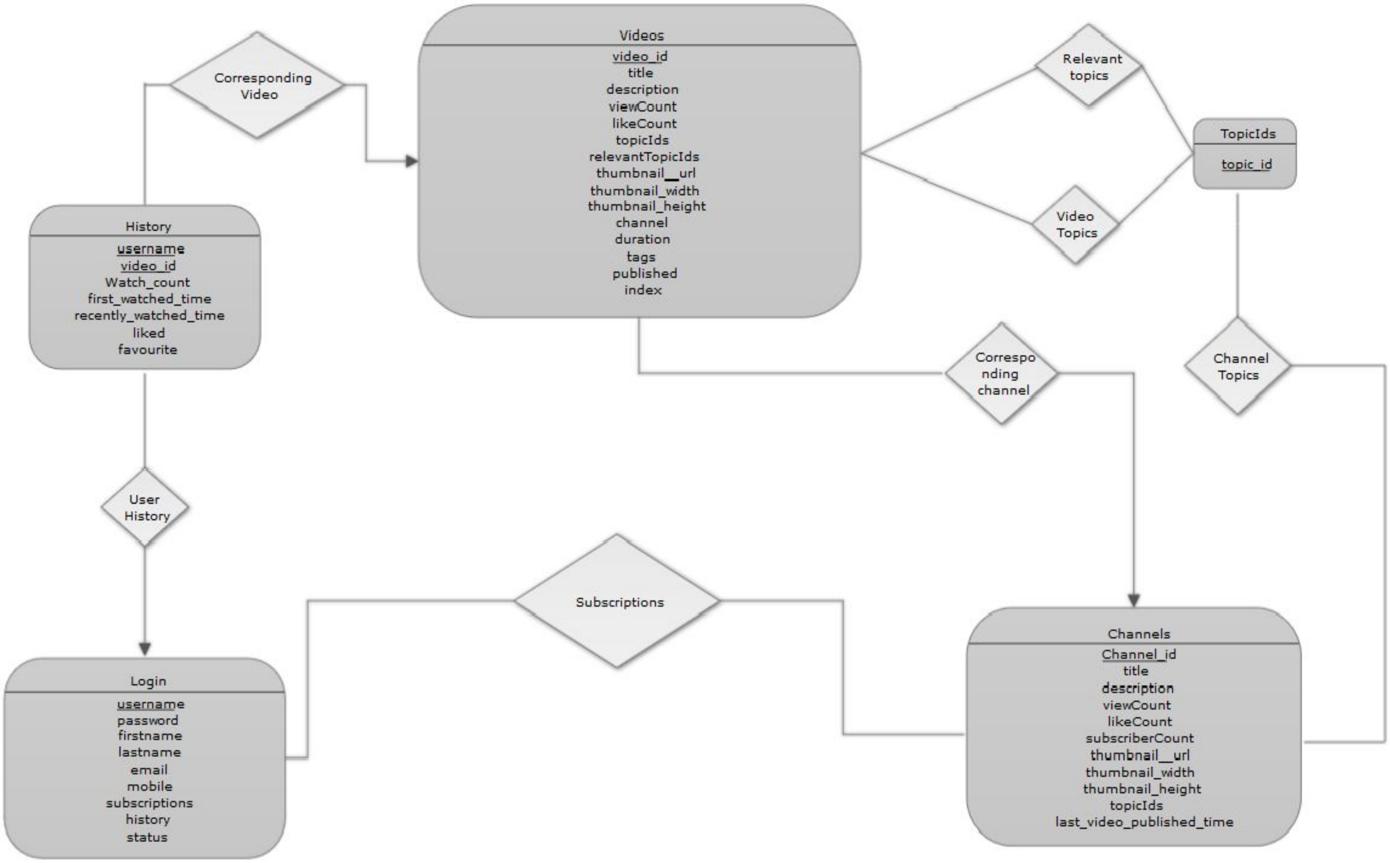
MUSIC DISCOVERY AND RECOMMENDATION SYSTEM

Akilesh B(cs13b1042)
Phaneendra Babu(cs13b1037)

General Idea :

This project creates a feed of new songs published on popular music channels on YouTube which displays, chronologically, the songs uploaded on YouTube. To recommend songs to a user based on his view history, currently popular music, currently trending music and other parameters. Our project is a simple, easy to use music discovery and recommendation system.

E-R Diagram :



Data Flow :

Our project briefly consists of the following pages, username is passed to all the pages:
-> Landing page : Home page. The first page seen by a user.
-> Login page : To login.
-> Register page: To register a new user.
-> User page : List of videos (recommended, most viewed, popular now)
-> Watch page : To watch the clicked video.
-> Search page : Search based on certain query terms.
-> Account page : Contains the details of the account information and provides an option to update his credentials.
-> Favourite page : Contains list of favourite videos marked by a user.

How do we recommend?

We store the watch count, view count and like count for every video. We also maintain the channels subscribed by a user.
We keep track of most frequently watched videos, most recently watched videos of every user. In addition to it, we consider videos which are trending or currently popular during that time. We recommend videos to the user based on his history and trending videos.
Every video also has tags describing that video. We also recommend videos which have common tags to those videos watched by the user. Every channel also has tags describing what that channel is about. If a user watches video of a particular channel, we recommend videos of other channels which have some common channel tags.

Technologies Used :

Our project is based on Django which is a high-level Python web framework that encourages rapid development and clean, pragmatic design.

Database: SQLite (Default database for Django). We faced some issues integrating PostgreSQL with Django.

Youtube Data API is used to retrieve required data. Our database consists of approximately 10, 000 videos and 200 channels. Whenever a user clicks on a particular channel, the database is updated with videos uploaded by that channel in a given time frame (which can be specified by us).

Front-end: jQuery which is a fast and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation and Ajax much simpler.

CSS for describing the presentation of a document written in a markup language.

Used Ajax our application can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page.

USP of project

-> Light weight.
-> Scalable.
-> Fast.
-> Compatible with many browsers.
-> Advertisements free.
-> Uses the best video player(YouTube player).
-> No unwanted recommendations depending on the peers’ search in a particular network