CpSc 4620/6620 Project

Objective:

This project requires students to develop an online multimedia database system, MeTube, which enables users share multimedia files online. The goal of this semester-long multimedia database project is to allow students to gain hand-on experiences in applying the database theories and techniques they will learn in the course to solve a real-world database application problem.

Development Environment:

The project will be implemented on MySQL and PHP. Although Apache web server on Linux is preferred, students can use MySQL and PHP with any other web servers (such as IIS) under their respective operating systems to develop their project initially. However, the project must be tested under Linux operating system with Apache web server.

Testing Environment:

The project must be deployed on a Linux system with Apache web server provided by the School of Computing and tested with both IE and Firefox as the web browser respectively.

MeTube System:

MeTube system is a modified version of the popular YouTube system (http://www.youtube.com). The content of MeTube system includes graphics objects, video, audio, images, and animation clips.

Using MeTube system, users are able to upload and download multimedia files through a web interface. Users can also view multimedia files online through proper media players embedded in the web browser. Any user with a Web browser can view multimedia files in MeTube system. A user must register an account to upload multimedia files and manage (annotate, update, remove, etc.) their uploaded files. When a user uploads a multimedia file, Meta information about the multimedia file should also be entered through the Web interface. The Meta information includes the title, description, and keywords used for searching the multimedia file. The user can also specify how to share the media file with others (for instance, share with everybody or just friends, allow discussion or not, allow rating or not, etc.). A Web interface must be designed to allow the registered users to annotate, update, and remove their uploaded multimedia files.

Registered users can display all multimedia files they uploaded, downloaded, and viewed with an appropriate Web interface when they log into their accounts. They can also organize media files they have viewed into playlists. A registered user also owns a channel which contains all multimedia files uploaded by this user. A registered user can subscribe to another user's channel if that user has allowed other users to view their uploaded multimedia files. A registered user can also create a favorite list of multimedia files. A registered user can maintain a contact list. A registered user can only send messages to users in his contact list. A user can organize their contacts into different categories, such as friends, family, etc. and he/she can also put a list of users into a blacklist so that these users cannot communicate with him/her. A user can also block other users from viewing/downloading the media files he/she uploaded. A user can invite users in his/her contact list to view/download media files through a simple messaging system. This messaging system works as a web-based email system with which users can send, receive, reply, and delete messages. A user can also create or join a discussion group in which users share interests and multimedia files, and discuss them. Once a user creates or joins a group, it can start a discussion topic, post a multimedia files, or post comments on the discussion topic or multimedia files.

A search interface must be implemented in MeTube system. A user can use this interface to search multimedia files based on keywords or media file properties (such as dates uploaded, file size, data format, etc.). Users can also browse multimedia files by category, time, popularity, etc. After a user finishes viewing a multimedia file, he/she can rate the file based on his/her viewing experience and make comments on this file if the owner who uploaded the multimedia file enabled the discussion option for the file. When a user selects a multimedia file to view, links to other related media files should be provided (This is called recommendation).

Project Requirement:

Although, as described in the syllabus, students should identify the MeTube system requirement by exploring YouTube system, a minimum set of basic functions that students must implement in their MeTube project is presented here. Graduate students must also implement advanced functions or features suggested in this document. Undergraduate students may elect to implement such advanced functions or features only after they have finished the implementation of basic functions.

(1) ***User account*:** A user needs to register an account to use all *MeTube* system functions. Basic account functions include user registration, sign-in, sign-out, profile update, and contact list. The advanced features include: (i) contact list organization (friend, family, favorite, etc.); (ii) blocking a user in the contact list (a blacklist). (2) ***Data sharing*:** A web interface must be implemented to allow users to upload multimedia files into *MeTube* system. This web interface should allow users to input the meta-information, such as title, description, keywords, category, etc., of a multimedia file into *MeTube* system. A user must sign into the system to be able to upload multimedia files. Any user with a Web browser should be able to download and view multimedia files available in *MeTube* system. Viewing a multimedia file must be through a media player embedded in the implemented web interface. Advanced data sharing features include: (i) setting the sharing methods (public, private, friends, rating permission, etc.) for multimedia files that a user uploaded; (ii) blocking certain users from downloading or viewing media files that a user uploaded. (3) ***Media organization*:** All users should be able to browse the multimedia files by categories. Signed-in users should be able to organize their uploaded media files and their interested media files in different ways, including playlists and favorite lists. Advanced features include: (i) showing the most-viewed media files and the most-recently uploaded files; (ii) ordering media files in different ways (such as ordering based on name, size, uploading time, etc.). (4) ***User interaction:*** signed-in users should be able to interact with each other by exchanging messages and commenting on media files. Advanced features include: (i) allowing users to rate multimedia files; (ii) allowing users to form discussion groups and discuss on certain topics related to multimedia files. (5) ***Search*:** The students must implement a *YouTube*-like search interface to allow users to search multimedia files based on keywords. Advanced features include: (i) displaying a word cloud linking to the popular multimedia files based on the popularity of the keywords; (ii) recommending related multimedia files to the user who currently views the multimedia files; (iii) searching media files based on low level media features, such as file size, type, and other image properties.

In addition to the requirements discussed in this document, students must create an account in YouTube and try all functions available in the YouTube system. Students are encouraged to identify all functions provided by YouTube system and implement them in their own projects.

Special Note:

CpSc 6620 students must implement all functions required in this project requirement document and CpSc 4620 students must implement all basic functions. CpSc 4620 students may earn extra credits only if they have finished all basic functions.

Project Policy:

- The project should be done by a team of three undergraduate students (two-student team may be allowed with the approval of the instructor) or two graduate students. Each team should elect a team leader who will be responsible for organizing team meetings and communicating with instructor on important issues related to the project.
- Each team must finish their project independently. Any form of cheating (including copying or reusing code from any source without the instructor's approval) will result in **0 (zero)** point for the project.

How to submit:

Please zip all your files (source code and documents) into a single ZIP file. Your submission must include a cover page listing all team members and indicating the team number. You should name the file as "XXX.YYY" in which "XXX" is your team number and "YYY" is the zip file extension. That is, team G4 should name the file as "G4.zip", team U3 should name the file as "U5.zip", etc. Please submit your files

through Canvas. Don't send your submission by email. Email submission will not be graded. Only the team leader needs to submit the zipped project file. However, all team members must submit their evaluations of all other team members. Please download the evaluation form from the course website and finish one evaluation for each teammate. Please name your evaluation form to include the team number, your name, and the name of the teammate you evaluate.

Questions and Concerns:

If you have any questions or concerns regarding this project, or if you feel any part of the project description is confusing, please talk to the instructor. Making false assumptions about the project may result in a low grade.