CpSc 462/662 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 version 5.0 or later and PHP 5 or later version. 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.

Testing Environment:

The final project must be deployed on a 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). But unlike YouTube system in which video is the only media type hosted, 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 interface. Although an Internet user does not need an account to view multimedia files, it is necessary for him/her to register an account to upload and manage (annotate, update, remove, etc.) the media files. When a user uploads a media file, Meta information about the media file should also be uploaded. The Meta information includes the title and description of the media file, and keywords used for searching the media 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.). The user can also change the Meta information of an uploaded media file or remove a media file if he/she does not want to share it anymore.

Users with a registered account can view a list of media files they uploaded, downloaded, and viewed when they log into their respective accounts. Users can also organize media files they viewed into playlists. A user can create many playlists. All media files uploaded by a user are organized into a broadcasting channel that other users can subscribe to. A user can also create a favorite list of media files. A user can subscribe to any channel created by another user. Besides the password, email options, and personal information, a user can also create contact lists which contain the information of users you want to contact. A user can also block another user from viewing/downloading the media files he/she uploaded. A user can invite friends 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 manage messages. A user can also create or join a group in which users share interests, exchange media files and discuss them. Once a user joins a group, it can start a discussion topic or post comments on a discussion topic.

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

Project Requirement:

Although, as described in the syllabus, students should identify the MeTube system requirement by exploring YouTube system, a minimum set of functions that students must implement in their MeTube project is presented here. Students may also implement the advanced functions or features suggested in this document or identified by them through studying the YouTube system.

The basic and advanced functions for the project implementation include: (1) ***User account*:** A user needs to register for an account to use all *MeTube* system functions. Students need to implement the basic account functions, including registration, sign-in, and profile update. The advanced features include contact list management, friend/foe list management, user blocking, etc. (2) ***Data sharing*:** A signed-in user should be able to use a web interface to upload multimedia files into *MeTube* system. This web interface should allow users to input meta-information about the multimedia file to be uploaded. Any Internet user should be able to browse, download, and view media files available in *MeTube* system through a media player embedded in the web interface. Besides implementing the basic upload/download functions, students may elect to implement more advanced features. For instance, a signed-in user can set the sharing methods for media files he/she uploaded; block certain users from downloading or viewing media files he/she uploaded. (3) ***Media organization*:** All users should be able to browse the media files by categories. Signed-in users should be able to organize their uploaded media files and their interested media files in different ways, including channel, playlists, favorite lists, etc. Students may also implement advanced features such as showing the most-viewed media files, the most-recently uploaded files, etc. (4) ***User interaction:*** signed-in users should be able to interact with each other by exchanging messages and commenting on media files. A signed-in user can also rate media files he/she has viewed. Students may also implement advanced features such as group discussion, etc. (5) ***Search*:** The students are required to implement a *YouTube*-like search interface to allow users to search media files based on keywords. Students may elect to implement advanced features such as word cloud, media recommendation, feature-based media search, etc.

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 for CpSc 462 Students:

CpSc 662 students must implement all functions required and suggested in the project requirement and CpSc 462 students must implement all basic functions.

Project Policy:

- The project should be done by a team of three undergraduate students (two-student team may be allowed) or two graduate students. Each team should nominate a team leader who will be responsible for organizing team meetings and communicating with the 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) will result in **0 (zero)** point for the project.
- Students must submit their project deliverables at the specified due dates. Late submission will not be accepted unless being approved by instructor.
- Intermediate results of the project will be evaluated and feedbacks will be given to improve the project. The instructor may not grant any extension of the project deliverables to ensure feedbacks will be given in a timely manner.

Project Deliverables (Due dates will be posted online):

1. ER Diagram (10%)
2. Relational Schema and Database Tables (10%)
3. Design necessary database queries for MeTube System (10%)
4. System implementation, evaluation and demonstration (50%)
5. Final report (10%)
6. Presentation (10%)

How to submit:

Please zip your source code or documents into a single file. You must include your team number in the file name and upload the file into the digital drop box on http://bb.clemson.edu. For instance, **team 3** should name their **phase 2** submission as "**T3P2.ZIP**". Please remember to click on the send button after you finished uploading. It will ensure your submission being delivered. Please send an email to the instructor immediately after your submission. You submission may be treated as late if you fail to send a notification email. Students must also email peer evaluation forms to the instructor after they submit their deliverables.

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.