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 using 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 deployed and tested in the environment provided by the School of Computing at Clemson University, which is an Apache web server under Linux operating system.

Testing Environment:

The project must be deployed on a Linux system with Apache web server provided by the School of Computing and tested with IE, Firefox and Chrome 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 a Web browser. If a user wants to upload multimedia files and manage (annotate, update, remove, etc.) their uploaded files in the MeTube system, he/she has to register an account in the system. When a user uploads a multimedia file, Meta information about the multimedia file should be entered through the Web interface. The Meta information includes 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.). The registered users can also annotate, update, remove their uploaded multimedia files, and update their account profiles.

When a registered user logs into his/her account, all multimedia files that they uploaded, downloaded, and viewed should be listed in the Web browser. The user can also organize media files he/she viewed into playlists. All multimedia files uploaded by one user form a channel of this user. Other registered users can subscribe to this channel if this user has allowed other users to view his/her uploaded multimedia files. A registered user can also create a favorite list of multimedia files. A registered user can also maintain a contact list and 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 blocklist 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 create or join a discussion group to share and discuss interests and multimedia files. Once a user creates or joins a group, it can start a discussion topic, post a multimedia files, or post comments on discussion topics 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 based on 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 and make comments on this file if the owner who uploaded the multimedia file enabled the rating 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, students are encouraged to identify the MeTube system requirements by exploring YouTube system and implement as many functions as possible, they must implement a set of basic functions in their MeTube project. Graduate students must also implement advanced functions or features suggested in this document. Undergraduate students may elect to implement advanced functions or features only after they have finished the implementation of basic functions. Extra credits may be awarded to the undergraduate students who elect to implement the advanced functions only if they have completed the 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 blocklist). (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 multimedia file's meta-information, such as title, description, keywords, category, etc., 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 publically in *MeTube* system or privately shared by a friend. Viewing a multimedia file must be through 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. The signed-in users should be able to organize their uploaded media files and their interested media files in different ways, including (not limited to) playlists (a playlist is a list of media files that are put together by a user so that he/she can play/view easily) and favorite lists (a list of media files that a user likes the most). A signed in user must be able to subscribe to the channels of any other users. 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:*** The 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*:** 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 the related multimedia files to a user who is currently viewing one particular multimedia file; (iii) searching media files based on low-level file features, such as file size, type, and image properties.

In addition to the requirements discussed in this document, students are encouraged to identify all functions provided by YouTube system and implement them in their projects.

Special Note:

CpSc 4620 students are required to implement all basic functions specified in this document. CpSc 6620 students must implement all (basic and advanced) functions. CpSc 4620 students may earn extra credits by implementing advanced functions only if they have finished all basic functions.

Project Policy:

- The project should be done by a team of three students (two-student team may be allowed with the approval of the instructor). Each team should elect a team leader who will be responsible for organizing team meetings and communicating with the instructor on project issues.
- 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.

Deliverables:

- **Implementation and Deployment (75%):** Team project must be deployed in the environment provided by the School of Computing. Source code must be submitted to Canvas along with final project report.
- **Project Report (25%):** Your project report should include system design, ER diagram, database schema, function design, implementation details, test cases, testing results, and user manual or instruction.

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.YYY", team U5 should name the file as "U5.YYY", etc. Please submit your files through Canvas. Don't send your submission by email because email submissions will not be graded. Only the team leader needs to submit the zipped project file. All team members must evaluate the contributions of other team members to the project. Please download the evaluation form from Canvas and finish one evaluation for each teammate in your team. Please name your evaluation form to include the team number, your name, and the name of the teammate whom you evaluate, and submit the evaluations through Canvas.

Questions and Concerns:

If you have any questions or concerns regarding this project, or if you feel any part of the project description is not clear, please talk to the instructor or the TA. Making false assumptions about the project may result in a low grade.