

Metube Project Technical Report

Christopher Crawford and Michael Mendes

Group 7

April 18, 2019

## Overview

The Metube Project 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 that helps student learn database systems.

The project can be found at: <http://webapp.cs.clemson.edu/~mamende/metube/index.php>

## System Design

Metube is implemented on Clemson's webapp server using HTML, PHP, Javascript, and MySQL. Users who first visit Metube can only browse and download media that is public. To upload and have access to other functions users must register and sign in. When a user uploads a file, they input all the metadata relevant to the media file. Metadata includes title, description, keywords, sharing settings, and ability to turn on/off rating and commenting. When a user views a media file they have the ability to rate or comment (if it's turned on) and add to their favorites list or their playlist. Users can see and edit their favorites list and playlist on the home page. All media files a user has uploaded can be seen on their channel, and a user can also subscribe to their channel on this page. A user's subscriptions can be managed on the home page. A user can also navigate to their profile page from the home page. A user can then see their uploaded media, see their bio, see their contact list, add a contact, remove a contact, edit a contact, edit their bio, and change their password on their profile page. A user can add a blocked person from their contact list to stop them from seeing or downloading their media. On the home page a user can go to their own channel. On the home page user can send a message to any user who does not

have them blocked, and they can see their messages in their inbox. A user can also join a discussion from the home page and submit posts to the ongoing conversation. Back on the browse page a user can go to a channel listing seen in the dropdown, and the user can search for media based on a keyword, file type, file size, category, most recently uploaded, and most viewed. An the coolest feature is the word cloud, which displays the most popular keywords and displays the videos associated with that keyword.

### ER Diagram

See the attached pdf - metube7ERDiagram.pdf

### Database Schema

See the attached pdf - metube7schema.pdf

### Function Design

User\_pass\_check

Input:

\$username = the row of the account table to be checked

\$password = the given password

Output:

Return = 2 if the password does not match that found in the account table

Return = 0 if the password was correct.

Die if the query was unsuccessful.

## register\_ID

### Input:

Input for this function are the required fields in register.php, including username, password, email, and type, which is always set to 0. User functions are delineated in the profile and channel pages. Media changes, edits to the profile or contact lists are all completed from private profile pages rather than including different user type values to give access to different functions.

### Output:

register\_ID created the account row as well as the php file for the new user's page, as '\$username'.php within the channels folder. It also sets a default biography value in the biographies table.

## updateViewCount

### Input: \$mediaid

Mediaid = the unique key for the media table who's viewCount will be incremented

### Output:

Incremented value in the viewCount column of the media table for the given media file.

Return = 0 if successful, die otherwise.

## updateMediaTime

Input: \$mediaid

Mediaid = the row of the media table to update

Output:

Updates the last access time of the file in the media table. Return = 0 if the update was successful, die otherwise.

## Upload\_error

Input:

\$result = the return value from media\_upload\_process.php inserts new values into the media table for a new file upload

Output:

An appropriate message for the attempted media upload process. See the function itself for individual message values.

## Account\_exists

Input:

\$username = the name to be checked in the account table

Output:

Returns 1 if the account does not exist

Return 0 if the account exists

## Contact\_exists

Input:

\$username = username in the contacts table to be checked

\$contactName = contactName to be checked in the contacts table

Output:

Return = 0 if the username + contactName combination was not in the list

Return = 1 if the contact can be added to the list

register\_contact

Input:

\$username = the user who's contact list will be updated

\$contactName = contact to be added to the list

\$contactType = the type of contact this will be

Output:

Die if unsuccessful query, else return = 0 to display appropriate message in

new\_contact\_process.php.

Remove\_contact

Input:

\$username = the user who's contact list will be updated

\$contactName = the contact to be removed from the list

Output:

Removes the row from the contacts table where the username + contactName information matches. Die if the query was unsuccessful.

## Edit\_contact

### Input:

\$username = the user who's contact information we are editing

\$contactName = the name of the contact who's type we are changing

\$contactType = the new type for the contact

### Output:

Updates the contacts table if successful, or die if the query could not be performed

## Edit\_bio

### Input:

\$username = the user who's biography is being changed

\$biotext = the text of the new biography

### Output:

Updates the biographies table with the new biography, or die if the query was unsuccessful.

## Check\_media\_permission

### Input:

\$mediaID = the mediaid of the media file the user is attempting to view

\$currentUser = the current user

### Output:

Checks to make sure that the user is allowed to view the file based on contact status type.

Return value is sent to media.php to either view the file or display a denial of access page.

## editConMess

### Input:

\$result = the return from edit\_contact

### Output:

Return = appropriate message for the result of page edit\_contact\_process.

Edit\_contact\_process invokes account\_exists, contact\_exists, and edit\_contact to check for appropriate values before making the mysql query. See the function itself for individual messages.

## editUpload

### Input:

All metadata from the media table where mediaid = \$mediaid.

### Output:

Updates all of the fields in metadata from the new form provided at edit\_upload.php.

First removes the previous keywords associated with the file before entering the new ones into the keyword table. Also updates sharing in case the share setting was updated.

## editUploadMess

### Input:

\$result = the result of editUpload

### Output:

Returns a success or failure message to edit\_upload\_process.php



## deleteUpload

### Input:

\$mediaid = the mediaid of the media to be deleted

### Output:

Effectively cascades deletion from all tables involving the media, including the media, keyword, and sharing tables. Returns 1 in the case that the download or media table's information could not be deleted.

## deleteUploadMess

### Input:

\$result = the result of deleteUpload

### Output:

Checks the result to see if the media file was successfully removed, or if there was an error.

Return = Message to delete\_upload.php

## Send\_message

### Input:

to : the user the message is sent to

From : the writer of the message

Message : the message contents

### Output:

Checks to make sure the sending user (\$from) is not blocked by the sent- to user (\$to) before sending the message. Also checks to make sure the account \$to exists.

Return = 1 : the message was successfully sent

Return = 3 : the sending user is blocked

Return = 2 : \$to user does not exist

## Send\_comment

### Input:

mediaId: the media file to post a comment to

userId: the comment poster

Comment: the comment content

### Output:

Return 1 if the query was successful, die otherwise

## Send\_discussion

Input: \$discussionid, userId, \$post

Post is the message to be sent to the discussion. discussionid is the discussion to post the message to, and the userId is the poster.

Output: return = 1 if the mysql query was successful, die if unsuccessful

## Implementation Details

1. Index
  - a. Post information for what media you want to view and display it
2. Login
  - a. Post login information and check to see if user exists, and then if password is correct
3. Register
  - a. Post information and check to see if user already exists, if not then the registration is successful
4. Browse
  - a. Post information for what media you want to view and display it
5. Media
  - a. Gets id and displays the media associated with the id
6. Favorites and Playlist
  - a. Posts information for what the user has added to their list
  - b. Post to delete a item from a user's list.

## 7. Profile

- a. Post information you want to update for user profile

## 8. Contacts

- a. Post contacts and their type (you block a user with this feature)

## 9. Upload

- a. Post meta data and information in the files

## 10. Message

- a. Post all the message information

## 11. Inbox

- a. Query to get all the messages sent to user

## 12. Subscriptions

- a. Query to get all current subscriptions
- b. Post to delete a subscription

## 13. Discussion

- a. Query to get all topics and discussions
- b. Post to submit a post on the topic

The folder /channels/ contains each of the created php files for the channels created when a user registers their account. The /uploads/ folder contains each of the individual folders for users who have uploaded a file, as per the template provided by the course. The docs/dist/ folder contains the css stylesheets we used as provided by the Spectre.css CSS Framework as provided:

<https://picturepan2.github.io/spectre/>

The majority of pages produced by the webserver are contained in the metube folder itself.

## Test Cases and Test Results

### Register Test - PASSED

1. Click Register on the home page
2. Fill out forms and click "Submit"
  - a. If username exists already, the page should display an error message
  - b. If any of the forms are not filled in, the page should display an error message
  - c. If the passwords do not match, the page should display an error message
  - d. If everything is correct then you should be redirected back to the login screen

### Log in - PASSED

1. Click on the home page
2. Fill out forms and click "Login"
  - a. If username does not exist, the page should display an error message
  - b. If any of the forms are not filled in, the page should display an error message
  - c. If the password does not match the database, then the page should display an error message
  - d. If everything is correct you should get directed to the logged in browse page
3. Confirm the user has logged in by going to the login.php page, and you should be redirected to the browse home page.

### Profile Test - PASSED

1. Log in and then navigate to the profile page
2. Click the add contact button

3. Fill out the form and click “submit”
4. Click “My profile”
5. The contact you added should appear in the contact list
6. Click “Remove Contact”
  - a. Type a name that is not in your contact list, and click “Submit”. A page should display saying this user is not in your list.
  - b. Type a name that is in the contact list, and click “Submit”. A page should display saying this user has been removed from your list.
7. Navigate back to the profile page
8. Click “Edit Contact”
  - a. Type a name that is not in your contact list, and click “Submit”. A page should display saying this user is not in your list.
  - b. Type a name that is in the contact list, and click “Submit”. A page should display saying the contact has been edited
9. Navigate back to the profile page
10. Click “Edit Bio”
  - a. Click “submit”, page should display saying your bio was edited
11. Click “Edit Upload”
12. Fill out the metadata for an upload
  - a. Click “submit” and a page should appear saying the upload was edited

Browse - PASSED

1. Click through all the search options and make sure the right title and media is being displayed
2. Click on the dropdown and choose a user. Make sure the channel page displays media for the correct person
3. Navigate back to the home page, and click the word cloud button. Make sure the word cloud appears correctly

#### Upload - PASSED

1. Navigate to the home screen and click “Upload”
2. Fill out all the metadata and click “Submit”
3. If the file is uploaded correctly, no error message should be displayed
4. Navigate to your profile and make sure the media file is uploaded

#### Media - PASSED

1. Click on a media file
2. If rating allowed, rate the file, and make sure the rate button disappears
3. If commenting is allowed, comment on the file, and make sure the comment appears in the table
4. Click “Add to Playlist” and make sure the button disappears
  - a. Navigate to the playlists and make sure the media you added is in the playlist
5. Click “Add to Favorites” and make sure the button disappears
  - a. Navigate to the favorites list and make sure the media you added is in the list
6. Make sure some recommendations appear in the side table
  - a. Click on a link and make sure it takes you to that media file page

### Discussion Test - PASSED

1. Navigate to the home page and click “Discussion”
2. Choose a category in the dropdown and click “Submit”
  - a. Make sure the right category is shown
3. Write a post and click “submit”
  - a. Make sure your post appears
4. Change the category and make sure the posts change to the right category
  - a. Try to post and make sure the right post appears in the right category

### Word Cloud Test - PASSED

1. Navigate to the home page and click “Word Cloud”
2. Click a word
  - a. Make sure the media associated with that word displays below
  - b. Click many words and make sure the right media is appearing

### User Instruction

1. What a guest can do:
  - a. Register for a new account
  - b. Login to an account
  - c. View any media shared publicly
  - d. Browse public media files in different categories
  - e. Download public media
2. What a Metube account can do:
  - a. Everything a guest can do



- b. Upload media
- c. Update profile information
- d. Sign out
- e. Manage media in playlists and favorite lists
- f. Message other users
- g. Post to discussion board
- h. Add contacts and block other users
- i. Rate, Comment, Add to Favorites, Add to Playlist on media where applicable
- j. Subscribe to a user's channel and manage subscriptions