

DATABASE MANAGEMENT SYSTEMS – METUBE MULTIMEDIA VIEWING AND SHARING PLATFORM

Nirali Bandaru (nbandar)

Reek Majumdar (rmajumd)

Clemson University – Spring 2020 – Dr. Zijung Wang

Submitted On: April 23, 2020

TEAM 2

Table of Contents

System Design

ER Diagram

Database Schema

Function Design

Implementation Details

Test Cases

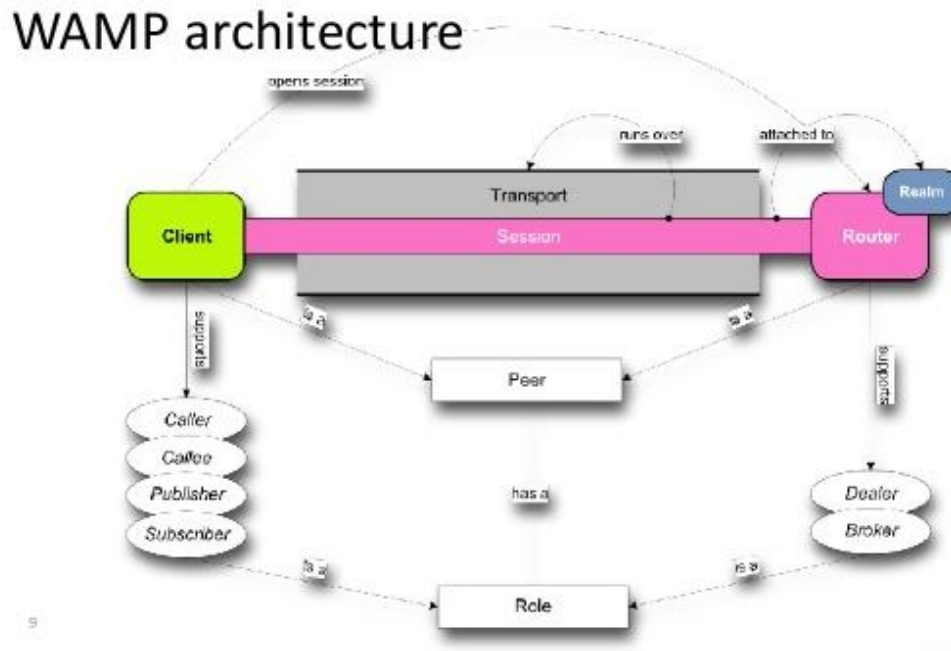
Testing Results

User Manual

System Design

The multimedia sharing platform, MeTube system, was set up using WAMP Apache Server on the local system.

MeTube is a platform where, like YouTube, users are able to register, login and upload and view various types of media to their liking. Media types include video, audio, gifs, and images. The following figure shows the WAMP server architecture.



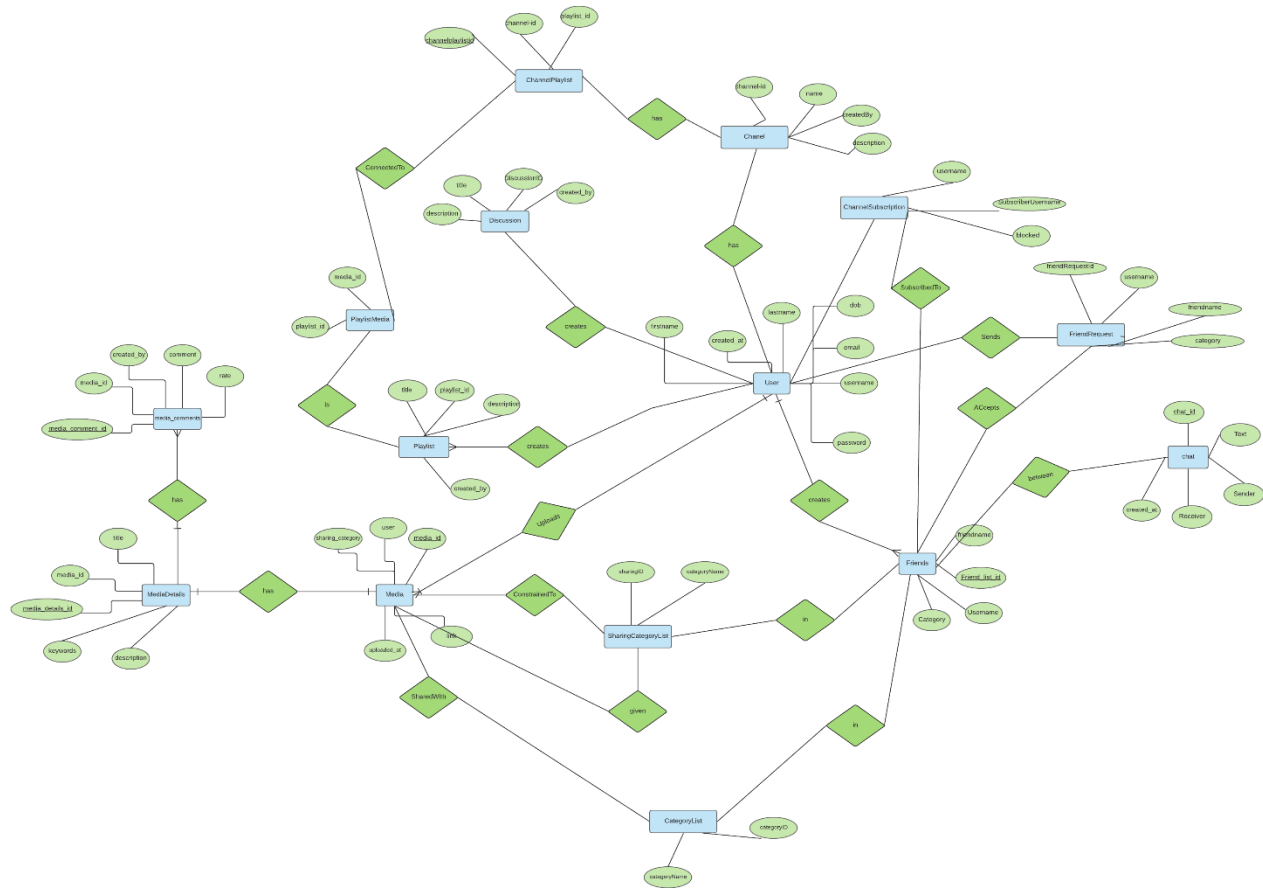
In this project, we have developed a system using the following tools:

- PHP Programming Language
- JavaScript
- HTML4/HTML5
- SQL
- MySQL
- Bootstrap
- phpMyAdmin
- WAMP Server
- CISCO Connect to connect to Clemson VPN

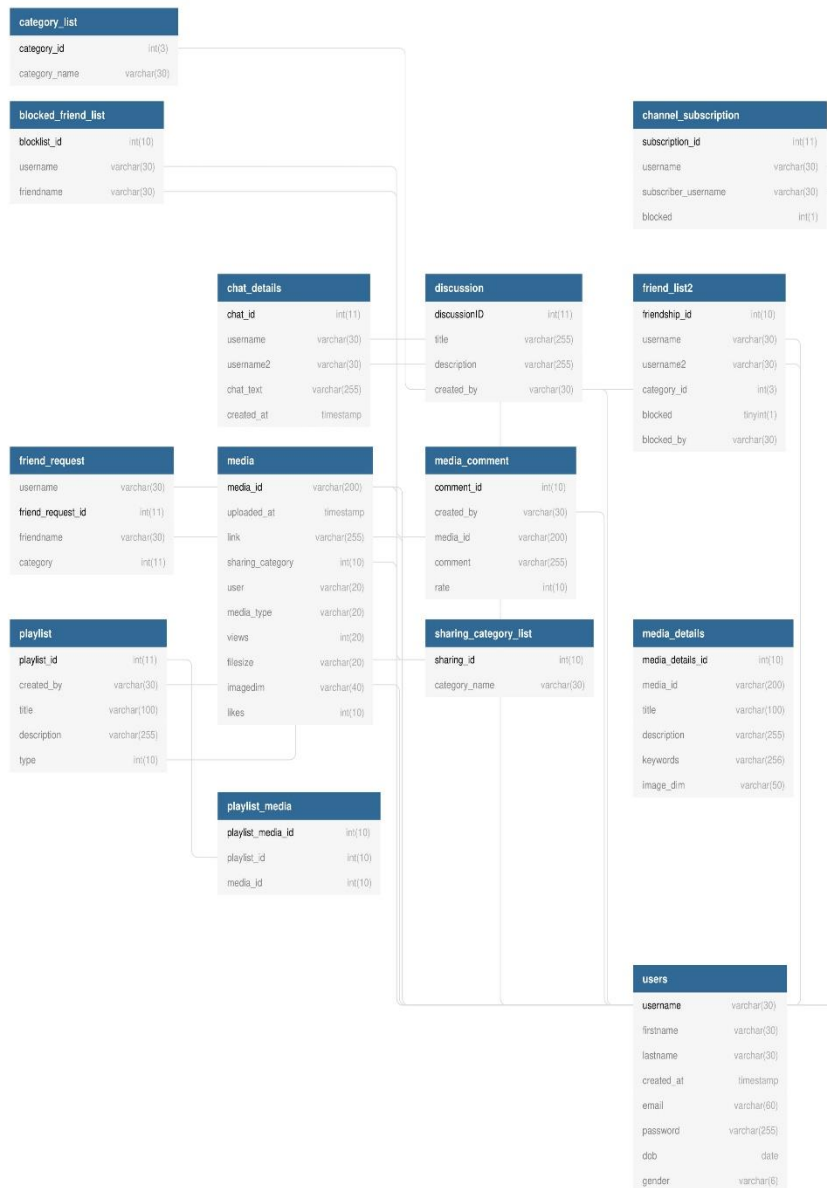
Major Learning Outcomes:

- Full understanding of how database is used, represented, interacts, and returns.
- How webpages are made.
- Sessions.

ER Diagram



Database Schema



Functional Design

- *Config.php*
 - Connecting to the database using `mysqli_connect`.
 - Session variables defined.
- *Config1.php*
 - Includes information about the database and host.
- *queryImplementation.php*
 - Constructs the query and runs it using customized function, `run_query`.
 - `Get_error` function returns error.
- *queryFunctions.php*
 - Contains all queries used to implement the system by defining them as functions.
 - Included in every code file.
- *welcome.php*
 - Basic homepage with only a description of the webpage, and the option to sign in/sign out.
 - Registration form, login form included.
 - Permissions for folders and file access granted.
 - Folders for different media types are created upon completion of registration.
- *profile.php*
 - This page consists of basic iframe that changes based on the inner tab of user's navigation bar.
 - Allows user to traverse through his discussion, playlists, and update information through settings.
- *channel1.php*
 - Channel is the first page a visitor views in the profile, first page user sees in his profile. Contains most recent videos, and the user's profile picture based on the user's gender.
- *uploadmedia.php*
 - Upload media was done by checking if the user is in session.
 - Each media type is checked for constraint satisfaction. If the media being attempted to get uploaded does not satisfy the criterion, error is returned.
 - Upon upload, a sharing category as given by user, is assigned to the user and updated in the database.
- *contacts.php*
 - User contact cards are used to display all contacts in the user's list.
 - Blocking, and receiving and sending friend requests.
 - Updating contact sharing-category(friends, family etc.)
- *discussion.php*
 - User and user's contacts are able to create a discussion on a certain topic.

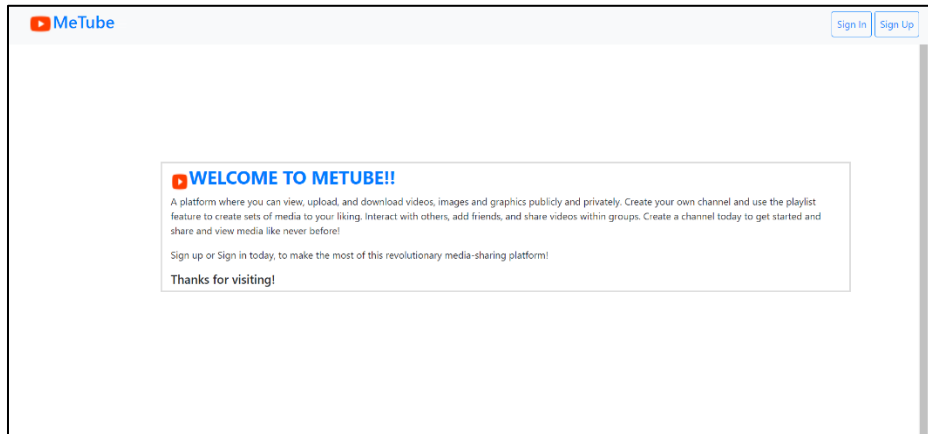
- Any user can participate in discussion.
- *settings.php*
 - User can update information.
 - Update query is used to connect to backend and replace with new details.
 - Page must be refreshed before the updated details are displayed.
- *playlists.php*
 - User adds videos from existing playlists or creates new playlists.
- *home_videos.php*
 - Displays the most popular videos available to this user.
 - Customized to user's accessibility to media.
 - Ordered by number of views.
 - User is only able to preview the media.
 - All controls for videos on this page are disabled.
 - User is redirected to a different page for viewing this media.
 - This was done to allow for views to be tracked.
- *home_audio.php*
 - Displays most popular audio available to this user.
 - Customized to the user's accessibility to media.
 - Ordered by number of views.
 - User is only able to preview the media.
 - All controls for videos on this page are disabled.
 - User is redirected to a different page for viewing this media.
 - This was done to allow for views to be tracked.
- *home_images.php*
 - Displays most popular images available to this user.
 - Customized to the user's accessibility to media.
 - Ordered by number of views.
 - User is only able to preview the media.
 - All controls for videos on this page are disabled.
 - User is redirected to a different page for viewing this media.
 - This was done to allow for views to be tracked.
- *home_graphics.php*
 - Displays most popular graphics available to this user.
 - Customized to the user's accessibility to media.
 - Ordered by number of views.
 - User is only able to preview the media.
 - All controls for videos on this page are disabled.
 - User is redirected to a different page for viewing this media.
 - This was done to allow for views to be tracked

- *myaudio.php, myvideos.php, mygraphics.php, myimages.php*
 - Consists only of user's uploaded media.
 - Includes media in private sharing category.
 - HTML code in these files is only for viewing the media.
- *media_view.php*
 - Most important of all, this file allows for the user to watch videos, listen to audio and view images and gifs.
 - Includes a form for liking the media.
 - Tracks views that visit this page.
 - This implementation was easier for tracking views.
- *recentuploads.php*
 - Consists of media ordered by time, most recently uploaded media, of all media types.
- *searchmedia.php*
 - Navigation bar includes the search button, which allows user to search for media and organize it according to their will and select among the given options.
 - In all pages that the navigation bar is included, the submitted search form redirects to this php file.
 - Takes the string sent by other php files and passes queries according to the combination chosen in the navigation bar, i.e., if the search word is "cute" and the user wishes to search by "name" and organize by "filesize," the user would choose the respective select options and then submit the form.
 - The values are then compared with the keywords and titles available in the database. Appropriate results are then returned.
- *Chat.php*
 - User details and user's friends details are retrieved from database.
 - Iframes are used in the frontend to facilitate chatting experience.
 - Datetime is displayed each time a user sends a message in the chat.
- *comments.php*
 - Comments page implements the php code for retrieving comments from media and displaying them.
- *recommendations.php*
 - Recommendations are made through the similarity between keywords of the currently viewing media file and other available media that have the same keywords.
- *logout.php*
 - Simply allows user to be redirected back to login page, where welcome to metube is displayed.

Implementation

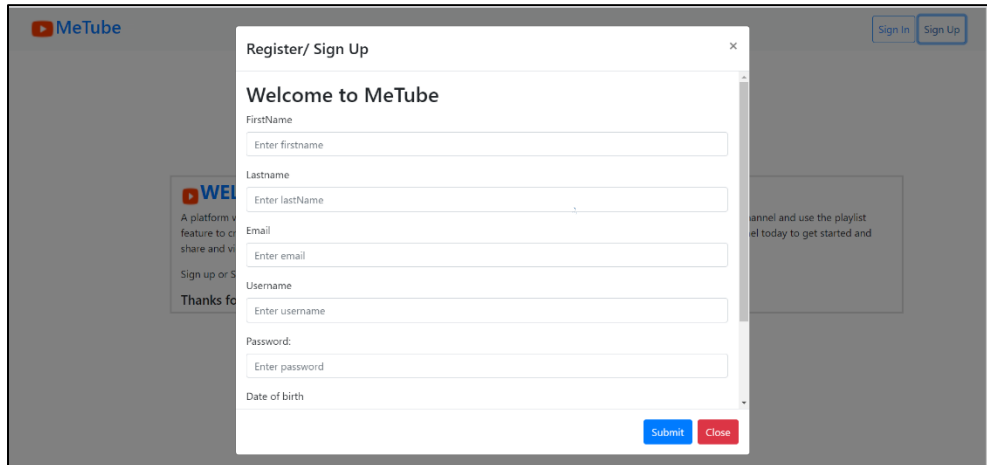
The following steps are steps that a visitor would take to participate in the MeTube system:

1. Welcome Page



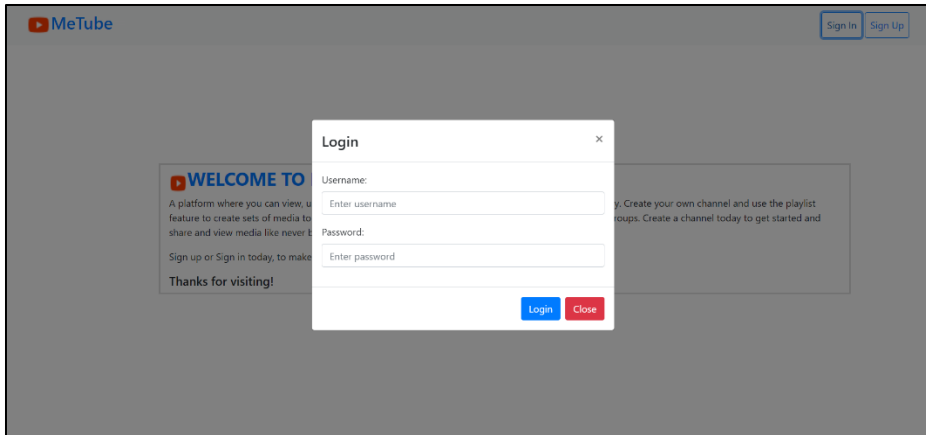
Upon visiting the website URL, visitor would only be given two options: **Sign in** or **Sign up**.

2. Registration Page



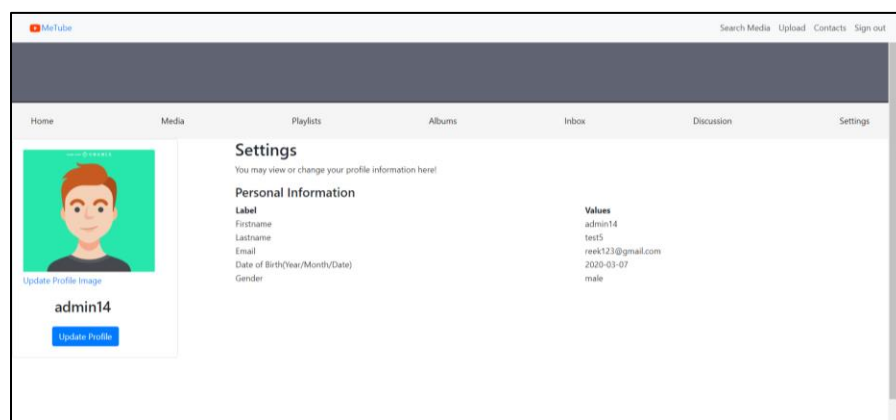
Registration modal was used to take in basic information of user and register details into the database.

3. Once registered, user is prompted to log in. **Log in** modal shows up:



Login is also done through modal view. Once submitted, user is redirected to the user's profile:

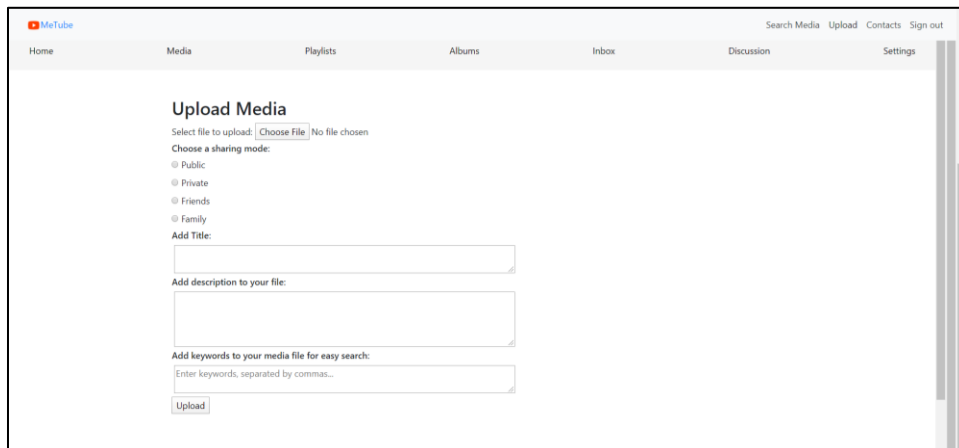
4. User's **profile** is the main means through which media can be searched or uploaded. All features of the MeTube system are only accessible after user has logged in. Profile contains an iframe that allows user to navigate through different tabs on a personal navigation bar.
 - a. **Channel:** This page is accessible to other users of MeTube. Channel page displays user's media in different categories such as media type, and it also allows for easy navigation through the user's public media.
 - b. **Playlists:** This is a personal page that allows for the user to group videos together under a common name. All playlists for videos and audio files are saved in this page.
 - c. **Discussion:** Users of MeTube can make discussion threads to discuss a specific topic of their interest.
 - d. **Settings:** This page allows user to update profile information



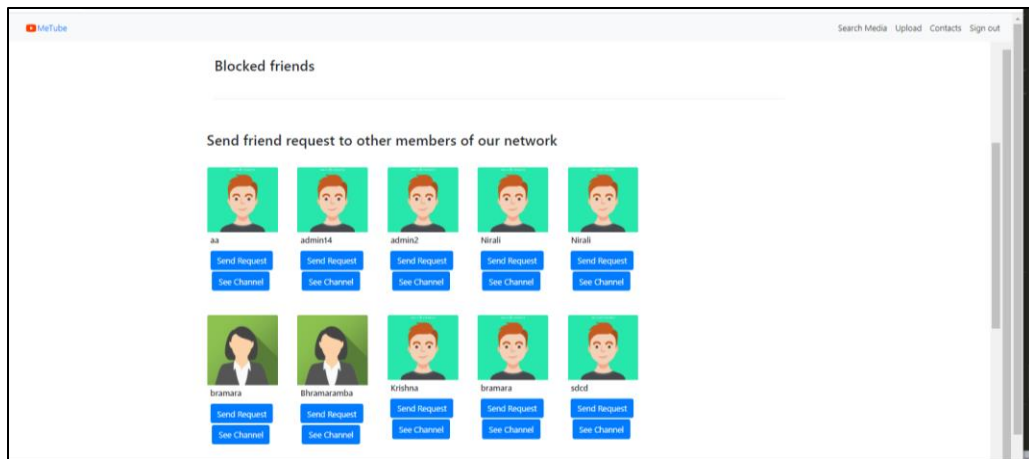
- e. In addition, a sidebar navigation is included.

There are certain changes in the navigation bar that occur after user is logged into the website:

- a. Search Bar: Appears after login.
- b. Search by: Allows user to choose between “Name” and “Keyword” to search media.
- c. Organize By: Allows user to choose between “Name”, “Media Type”, “File size”, “Upload Time” and “Image Resolution” to organize the search results.
- d. On the right-hand corner, the following are included:
 - i. Search Media: This is the primary button that redirects to media page.
 - ii. Upload: This page allows user to upload media files in the following formats:
 - 1. Images – .jpg, .jpeg, .png
 - 2. Audio – .mp3
 - 3. Videos – .mp4
 - 4. Graphics – .gif
 - a. Other formats aren’t allowed. Error is displayed if a format other than allowed formats is selected.



- iii. Contacts:
 - 1. This is where accept friend request and send friend request are handled.
 - 2. Friend suggestions are made.
 - 3. This page contains lists of contacts:
 - a. Friends
 - b. Family
 - c. Acquaintances
 - d. Blocked contacts



test Friends



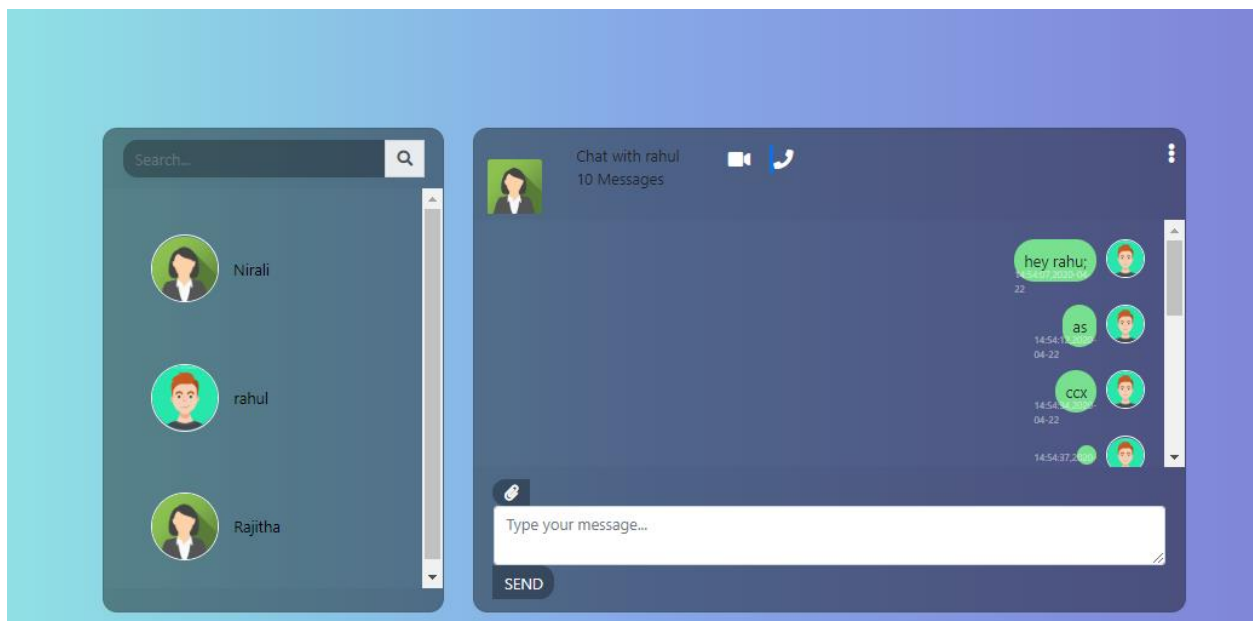
test2

Update
Relationship

chat

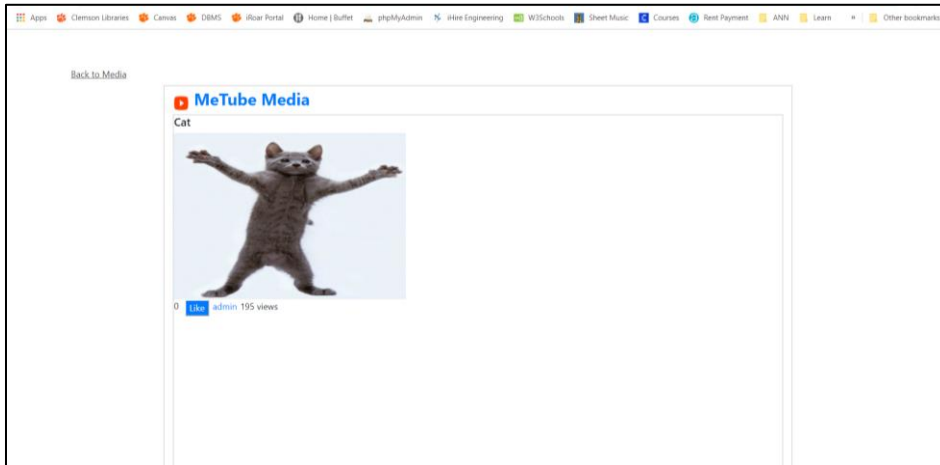
Block

See Channel

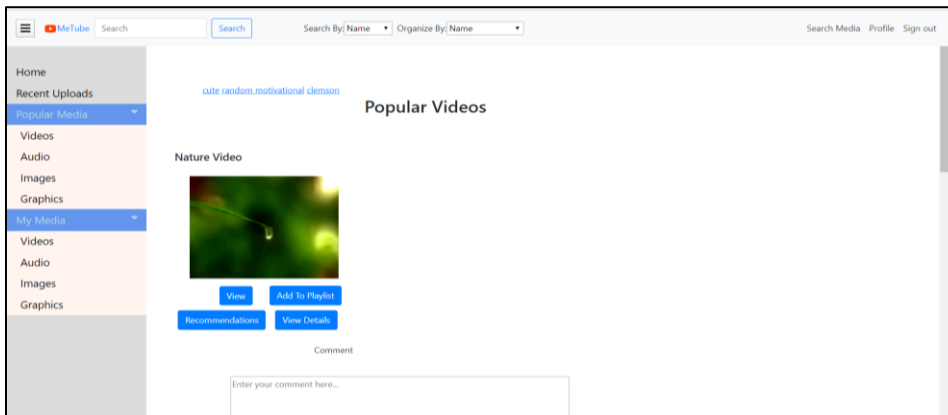


4. Also allows a friend's category to be updated to either one of the other categories.
5. Allows user to chat with a friend
6. Allows to view the friends' channels if user is subscribed to them.

5. Media Page (Search Media)



- a. A sidebar navigation is included to allow for easy navigation:

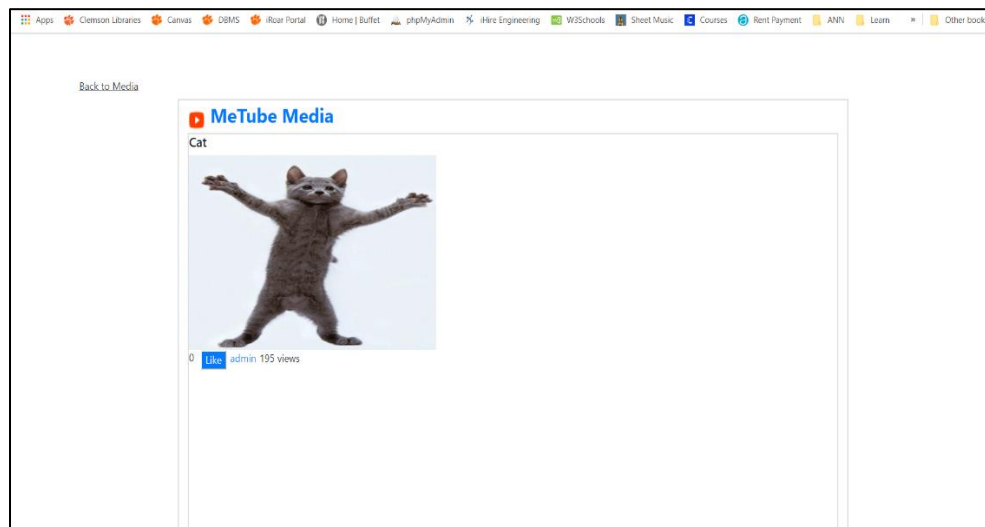


- i. Home redirects to Popular Videos page
- ii. Popular Media contains:
 1. Most viewed videos
 2. Most viewed audio
 3. Most viewed images
 4. Most viewed graphics
- iii. My Media contains:
 1. All the media in each category that this particular user in session is allowed to view- including a friend's or family member's media that the user has access to.

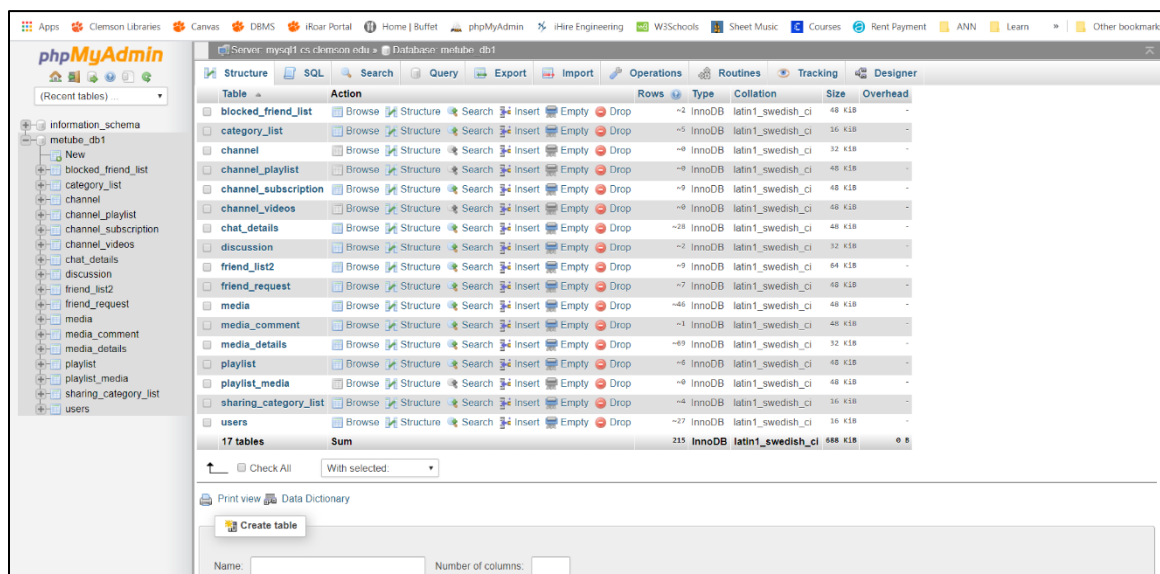
- iv. Recent Media lists the most recently uploaded media, limiting to 10 files per category.
- v. The links on top right-hand corner change for this page:
 1. Search Media
 2. Profile
 - a. Redirects user back to their home channel page.
 3. Sign Out
 - a. Allows user to sign out and redirects to homepage.

6. Media View

- a. When elements on the media page are clicked, they are redirected to a single element view page called the media view page. This page displays the media item selected. It also gives an option to like and has a view count.



7. Database



8. Major In-built Functions Used

- a. Session_start()
- b. Session_destroy()
- c. Mysqli_fetch_array()
- d. Mysqli_query()
- e. Mysqli_error()
- f. Mysqli_close()
- g. Mysqli_commit()
- h. Mysqli_num_rows()
- i. Window.location.href
- j. Document.getElementById functions
- k. Document.querySelector
- l. .addEventListener()
- m. .toggle
- n. Uniqid()
- o. Explode()
- p. Move_uploaded_file()
- q. Getimagesize()
- r. Chmod()
- s. Console.log
- t. Alert()
- u. Json_encode()

Test Cases

User registration

Sign-in

Sign-out

Profile update

Contact List

Contact list organization
(friend, family, favorite, etc.)

Blocking a user in the contact list.

Upload multimedia files
Web interface for Meta data of
uploaded files such as title,
description, keywords, category, etc.

Download:

View media files:

Sharing Mode:

Blocking:

Browse by category

Channel

Playlists

Favorite Lists

Most viewed

Most recently uploaded

Ordering media files in different
ways (such as name, size, uploading
time)

Messaging Exchange between user

Commenting on media file

Media rating

**Group discussion related to
multimedia files:**

Keywords-based search
Word cloud

Media recommendation

Feature-based media search
(such as file size, type, and other
image properties)

Result: Passed

APPENDIX

Query Functions

```
function upload_file($media_id,$uploaded_at,$link,$sharing_category,$user,
$media_type,$filesize,$imagedim) {

    $query = "INSERT INTO media (media_id, uploaded_at, link, sharing_category, user,
media_type,filesize,imagedim) VALUES
('".$media_id."','".$uploaded_at."','".$link."','".$sharing_category."','".$_SESSION['username']."'
, '".$media_type."','".$filesize."','".$imagedim."')";

    return $query;

}
```

```
function upload_file_details($media_id, $title, $description, $keywords="", $image_dim) {

    $query = "INSERT INTO media_details (media_id, title, description,
keywords,image_dim) VALUES
('".$media_id."','".$title."','".$description."','".$keywords."','".$image_dim."')";

    return $query;

}
```

//PUBLIC VIEWING

//FOR USER CHANNEL

```
function get_latest_videos_user($user) {

    $sql = "SELECT * from media WHERE sharing_category=1 AND user='$user' AND
media_type='video' order by uploaded_at LIMIT 10";

    return $sql;

}
```

```

function get_latest_images_user($user) {
    $query = "SELECT * from media WHERE sharing_category=1 AND user='$user'
media_type='image' order by uploaded_at LIMIT 10";
    return $query;
}

function get_latest_audio_user($user) {
    $query = "SELECT * from media WHERE sharing_category=1 AND user='$user' AND
media_type='audio' order by uploaded_at LIMIT 10";
    return $query;
}

function get_latest_graphics_user($user) {
    $query = "SELECT * from media WHERE sharing_category=1 AND user='$user' AND
media_type='graphics' order by uploaded_at LIMIT 10";
    return $query;
}

//FOR SHOWMEDIA PAGE

//Query for user's videos in showmedia page: in categories friends, family and self.
function get_all_videos_user($user) {
    $query = "SELECT * from media m
INNER JOIN media_details md ON m.media_id = md.media_id
WHERE
media_type='video'
AND (user='$user'
OR (user in
(SELECT username2 from friend_list2

```

```

WHERE username = '$user'
AND category_id in (1,2,3,4)
AND (sharing_category in (1,2,4))))";

    return $query;
}

```

//audio

```

function get_all_audio_user($user) {
    $query = "SELECT * from media m
INNER JOIN media_details md ON m.media_id = md.media_id
WHERE
media_type='audio'
AND (user='$user'
OR (user in
(SELECT username2 from friend_list2
WHERE username = '$user'
AND category_id in (1,2,3,4)
AND (sharing_category in (1,2,4))))));";

    return $query;
}

```

//images

```

function get_all_images_user($user) {
    $query = "SELECT * from media m
INNER JOIN media_details md ON m.media_id = md.media_id
WHERE

```

```

media_type='image'
AND (user='$user'
OR (user in
(SELECT username2 from friend_list2
WHERE username = '$user'
AND category_id in (1,2,3,4))
AND (sharing_category in (1,2,4))))");
    return $query;
}

```

//graphics

```

function get_all_graphics_user($user) {
    $query = "SELECT * from media m
INNER JOIN media_details md ON m.media_id = md.media_id
WHERE
media_type='graphics'
AND (user='$user'
OR (user in
(SELECT username2 from friend_list2
WHERE username = '$user'
AND category_id in (1,2,3,4))
AND (sharing_category in (1,2,4))))");
    return $query;
}

```

//VIDEO ELEMENTS

```
function update_likes($media_id,$likes) {
    $query = "UPDATE media SET likes= '$likes'+1 WHERE media_id = '$media_id'";
    return $query;
}
```

```
function update_views($media_id, $views) {
    $query = "UPDATE media SET views = '$views'+1 WHERE media_id = '$media_id'";
    return $query;
}
```

```
function get_most_viewed_videos() {
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE sharing_category in (1,2,4) AND media_type='video' ORDER BY views
DESC";
    return $query;
}
```

```
function get_most_viewed_images() {
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE sharing_category in (1,2,4) AND media_type='image' ORDER BY views
DESC";
    return $query;
}
```

```
function get_most_viewed_audio() {
```

```
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =  
md.media_id WHERE sharing_category in (1,2,4) AND media_type='audio' ORDER BY views  
DESC";
```

```
    return $query;
```

```
}
```

```
function get_most_viewed_graphics() {
```

```
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =  
md.media_id WHERE sharing_category in (1,2,4) AND media_type='graphics' ORDER BY views  
DESC";
```

```
    return $query;
```

```
}
```

```
function get_media_query($media_id) {
```

```
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =  
md.media_id WHERE m.media_id = '$media_id'";
```

```
    return $query;
```

```
}
```

```
function get_latest_videos() {
```

```
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =  
md.media_id WHERE media_type='video' AND sharing_category in (1,2,4) ORDER BY  
uploaded_at DESC LIMIT 10";
```

```
    return $query;
```

```
}
```

```
function get_latest_images() {
```

```
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =  
md.media_id WHERE media_type='image' AND sharing_category in (1,2,4) ORDER BY  
uploaded_at DESC LIMIT 10";
```

```

        return $query;
    }

    function get_latest_audio() {

        $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='audio' AND sharing_category in (1,2,4) ORDER BY
uploaded_at DESC LIMIT 10";

        return $query;
    }


    function get_latest_graphics() {

        $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='graphics' AND sharing_category in (1,2,4) ORDER BY
uploaded_at DESC LIMIT 10";

        return $query;
    }


    // MEDIA ORGANIZATION IN SEARCH RESULTS


    function get_video_by_size() {

        $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='video' AND sharing_category in (1,2,4) ORDER BY filesize
DESC";

        return $query;
    }


    function get_audio_by_size() {

        $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='audio' AND sharing_category in (1,2,4) ORDER BY filesize
DESC";

```



```

        return $query;
    }

function get_image_by_size() {
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='image' AND sharing_category in (1,2,4) ORDER BY filesize
DESC";

    return $query;
}

function get_graphics_by_size() {
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='graphics' AND sharing_category in (1,2,4) ORDER BY filesize
DESC";

    return $query;
}

//ORDER BY IMAGE PROPERTY
function get_image_by_dimensions() {
    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE media_type='image' AND sharing_category in (1,2,4) ORDER BY
dimension DESC";

    return $query;
}

//GET ALL MEDIA FROM USER
function get_all_media($user) {
    $query = "SELECT * from media WHERE username = '$user' AND (sharing_category=1
OR (user in (SELECT username2 from friend_list2 WHERE username = '$user' AND category_id
in (1,2,3,4)) AND (sharing_category in (1,2,4))))";

```

```

        return $query;
    }

//SEARCH BY KEYWORD

function search_key_order_name($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE keywords LIKE '%$search%' ORDER BY title";
    return $query;
}

function search_key_order_type($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE keywords LIKE '%$search%' ORDER BY '$media_type'";
    return $query;
}

function search_key_order_uploadtime($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE keywords LIKE '%$search%' ORDER BY '$uploaded_at'";
    return $query;
}

function search_key_order_filesize($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE keywords LIKE '%$search%' ORDER BY '$filesize'";
    return $query;
}

```

```

// function search_key_order($search, $keywords,$order) {
//     if ($order=='name') {
//         $orderq = 'title';
//     }
//     else if($order == 'type') {
//         $orderq = "
//     }
// }

//SEARCH BY NAME

function search_name_order_name($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE title LIKE '%$search%' ORDER BY title";
    return $query;
}

function search_name_order_type($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE title LIKE '%$search%' ORDER BY media_type";
    return $query;
}

function search_name_order_uploadtime($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE title LIKE '%$search%' ORDER BY uploaded_at";
    return $query;
}

```

```
function search_name_order_filesize($search) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id=
md.media_id WHERE title LIKE '%$search%' ORDER BY filesize";
    return $query;
}
```

//RECOMMENDATIONS

```
function recommendation($key, $media_type) {
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id =
md.media_id WHERE media_type='$media_type' AND (keywords LIKE '%$key%' OR description
LIKE '%$key%'";
    return $query;
}
```

```
function viewDetails($media_id){
    $query = "SELECT * from media as m INNER JOIN media_details as md on m.media_id =
md.media_id WHERE m.media_id = '$media_id'";
    return $query;
}
```

```
function insertComments($created_by, $media_id, $comment) {
    $query = "INSERT INTO media_comment(created_by, media_id, comment) VALUES
('".$created_by."','".$media_id."','".$comment."')";
```

```

        return $query;
    }

function displayComments( $media_id) {

    $query = "SELECT * from media_comment as mc INNER JOIN media_details as m ON
mc.media_id = m.media_id WHERE m.media_id = '$media_id' ORDER BY uploaded_at";

}

//PERSONAL VIEW

//RESULT BY MEDIA TYPE - ALL MEDIA

function get_my_videos($user) {

    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE user='$user' AND media_type = 'video'";

    return $query;

}

function get_my_audio($user) {

    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE user='$user' AND media_type = 'audio'";

    return $query;

}

function get_my_images($user) {

    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE user='$user' AND media_type = 'image'";

    return $query;

}

```

```

function get_my_graphics($user) {

    $query = "SELECT * from media m INNER JOIN media_details md ON m.media_id =
md.media_id WHERE user='$user' AND media_type = 'graphics'";

    return $query;

}

//

//
*****
*****

```

```

function login_query($username,$password){

    $query= "
    select * from users where
    username= ''.$username.'"
    AND password= ''.$password.'"
    ";
    return $query;

}

```

```

function
register_query($username,$password,$firstname,$gender,$dateofbirth,$email,$lastname=""){

    $query = "INSERT INTO users (username, password,
firstname,gender,dob,email,lastname)
VALUES ('$username','$password','$firstname','$gender','$dateofbirth','$email','$lastname')";

    return $query;
}

```

```
}
```

```
function
```

```
update_user_details($username,$firstname,$gender,$dateofbirth,$email,$lastname=""){
```

```
    $query = "UPDATE users
```

```
        SET
```

```
        firstname = '$firstname', lastname= '$lastname',gender='$gender',
```

```
        dob = '$dateofbirth',email='$email'
```

```
        WHERE username = '$username';";
```

```
//    $query = "INSERT INTO users (username, password,  
firstname,gender,dob,email,lastname)
```

```
// VALUES
```

```
(' $username',' $password',' $firstname',' $gender',' $dateofbirth',' $email',' $lastname');";
```

```
    return $query;
```

```
}
```

```
function remove_registered_user($username){
```

```
    $query = "DELETE FROM users WHERE username='$username';
```

```
    return $query;
```

```
}
```

```
function get_user_information($username){
```

```
    $query= "
```

```
    select firstname,lastname,email,dob,gender from users where
```

```
    username= "".$username."
```

```
    ";
```

```

        echo $query;
        return $query;
    }

function show_users($username){
    // $query = "SELECT *from users where username <> '$username'";

    $query = "Select * from users \n"
        . "where \n"
        . "username <> '". $username. "' AND \n"
        . "username NOT IN (Select friendname from friend_request \n"
        . "where \n"
        . "username = '". $username. "')";

    return $query;
}

function get_category(){
    $query = "SELECT * FROM `category_list`";
    return $query;
}

function get_friendship_details($username,$frdname){

```



```

        $query= "Select *from friend_list2 where username= '$username' AND username2=
'$frdname'";

        return $query;

```

```

}

```

```

function update_rel($username,$frdname,$category_id){

    //$query= "UPDATE `friend_list2` SET `category_id` = '$category_id' WHERE
username='$username' AND username2='$friendname'";

    $sql = "UPDATE `friend_list2` SET `category_id` = '$category_id' WHERE
username='$username' AND username2='$frdname'";

    return $sql;

}

```

```

function show_usr_frds($username){

    $sql = "SELECT username,firstname,gender from users WHERE username IN (SELECT
username2 from friend_list2 where username='$username')";

    // $sql = "Select username2 from friend_list2 where username='$username'";

    return $sql;

}

```

```

function friend_request($username,$friendname,$category=1){

    $query = "INSERT INTO friend_request (`username`, `friendname`, `category`) VALUES
('$username', '$friendname', '$category')";

    return $query;

}

```

```

function show_friend_request($username){
    $sql = "Select u.username,u.firstname, u.gender ,f.friend_request_id,f.friendname\n"
    . "from \n"
    . "users as u\n"
    . "Inner join\n"
    . "friend_request as f\n"
    . "ON\n"
    . "u.username = f.username\n"
    . "where\n"
    . "f.friendname='$username'\n"
    . "\n";
    return $sql;
}

```

```

function accept_request($username,$username2,$category=1){
    $query = "INSERT INTO friend_list2 (username,username2,category_id)
    VALUES
    ('$username','$username2','$category'),(' $username2','$username','$category)";

    // $sql = "INSERT INTO friend_list2 ( `username`, `username2`, `category_id`, `blocked`)
    VALUES ( \ 'Nirali\ ', \ 'reek\ ', \ '1\ ', \ '0\ '), (NULL, \ 'reek\ ', \ 'Nirali\ ', \ '1\ ', \ '0\ ');";

    return $query;
}

```

```

function accepted_ignored_request($username,$friendname){
    //first run accept_friend_request from front end,when successful then run this function

```

```

        $query = "DELETE FROM friend_request WHERE username='$username' AND
friendname='$friendname' ";
        return $query;
    }

```

```

function accept_friend_request($username,$friendname,$category=1){
    $query = "INSERT INTO friend_list2 (username,username2,category_id)
VALUES ('$username','$friendname','$category')";
    return $query;
}

```

```

function show_friends($username){

    $query = "SELECT username,firstname,lastname,gender from
users
WHERE
username IN (
SELECT username2
from friend_list2
WHERE
username = '$username' AND blocked = 0
)
";
    return $query;
}

```

```

function show_blocked_friends($username){
    $query= "select username,firstname,lastname,gender from users
    where username IN (
        Select username2 from friend_list2
        where username = '$username' AND blocked = 1
    )";
    return $query;
}

function find_frd_list_id($username,$friendname){
    $query="Select friendship_id from friend_list2
    where
    (username = '$username' AND username2 = '$friendname') or
    (username = '$friendname' AND username2= '$username') ";
    return $query;
}

function block_friend($username,$friendname){
    $query= "UPDATE `friend_list2` SET `blocked` = '1' WHERE username='$username' AND
    username2='$friendname'";

    // IMPL1

    //$sql = "INSERT INTO blocked_friend_list ( username, friendname) VALUES
    ('$username', '$friendname');";

    // impl2

```

```
        // $query= "UPDATE `friend_list2` SET `blocked` = '1' WHERE `friendship_id` =  
$friendship_id";
```

```
        return $query;
```

```
    }
```

```
function unblock_friend($username,$friendname){
```

```
    $query= "UPDATE `friend_list2` SET `blocked` = '0' WHERE username='$username' AND  
username2='$friendname'";
```

```
    return $query;
```

```
}
```

```
// chat details
```

```
function get_chat_details($username,$frdname){
```

```
$sql = "select
```

```
username, username2, chat_text, DATE( created_at ) AS date, TIME( created_at ) AS time
```

```
from chat_details\n"
```

```
. "where\n"
```

```
. "(username='$username' AND `username2`='$frdname') \n"
```

```
. "OR\n"
```

```
. "(username2='$username' AND `username`='$frdname') \n"
```

```
. "order by\n"
```

```
. "`created_at`\n"
```

```
. "";
```

```

    return $sql;
}

function insert_chat_details($username,$frdname,$chat){

    $sql = "INSERT INTO chat_details ( `username`, `username2`, `chat_text`) VALUES
('$username', '$frdname', '$chat');";

    return $sql;
}

function channel_Subscription($userId,$subscriberID){

    $sql = "INSERT INTO `channel_subscription` ( `username`, `subscriber_username`) VALUES
('$userId', '$subscriberID');";

    return $sql;
}

function checkSubscribed($userId,$subscriberID){

    $sql = "SELECT * FROM `channel_subscription` WHERE `username` = '$userId' AND
`subscriber_username` ='$subscriberID'";

    return $sql;
}

function get_all_media_by_user_by_category($username,$media_type){

    // "SELECT * from media m INNER JOIN media_details md ON m.media_id = md.media_id
    WHERE user = '$username' AND sharing_category in (1,2,4) AND media_type='$media_type'
    ORDER BY views DESC";

    $sql = "SELECT * FROM `media` m INNER JOIN media_details md on m.media_id =
md.media_id WHERE `user` = '$username' AND media_type='$media_type' \n"

```

```

        . "AND `sharing_category` = 1 ORDER BY views DESC";
    return $sql;
}

function insert_comment2($user,$media_id,$comment){

    $sql = "INSERT INTO media_comments ( created_by, media_id, comment) VALUES
('$user', '$media_id', '$comment')";

    return $sql;
}

function createPlaylist($user,$title,$description){

    $sql = "INSERT INTO playlist ( created_by,title, description) VALUES ('$user', '$title',
'$description')";

    return $sql;
}

function show_play_query($user){

    $sql = "SELECT * FROM `playlist` WHERE `created_by` ='$user'";

    return $sql;
}

function createDiscussion($user,$title,$description){

    $sql = "INSERT INTO discussion ( created_by,title, description) VALUES ('$user', '$title',
'$description')";

    return $sql;
}

```

```

function show_discussion(){
    $sql ="Select *from discussion";
    return $sql;
}

function show_discussion_by_id($id){

}

function find_full_media_details_query($media_id){

    $sql = "select *from media m\n"
    . "Inner join \n"
    . "media_details md\n"
    . "on \n"
    . "m.media_id=md.media_id\n"
    . "inner join\n"
    . "media_comment mc\n"
    . "on \n"
    . "mc.media_id = m.media_id LIMIT 0, 30 ";

    $sql = "select *from media m\n"
    . "Inner join \n"
    . "media_details md\n"
    . "on \n"
    . "m.media_id=md.media_id\n"
    . "inner join\n"
    . "media_comment mc\n"

```



```
. "on \n"  
. "mc.media_id = m.media_id  
where m.media_id='$media_id' "  
return $sql;  
}  
?>
```