CPSC - 6620, DATABASE MANAGEMENT SYSTEMS

METUBE MULTIMEDIA VIEWING AND SHARING
PLATFORM

-Abhinav Singh

-Eshwar Bychanahalli

Table of Contents

System Design:

The multimedia sharing platform, compromises with many advanced user interface features, which makes users to handle their account, uploading videos, commenting and chatting over it will be easier. To develop this advanced system, we used MySQL, PHP, HTML, CSS, JS. This multimedia system has been in the local system using Apache server.
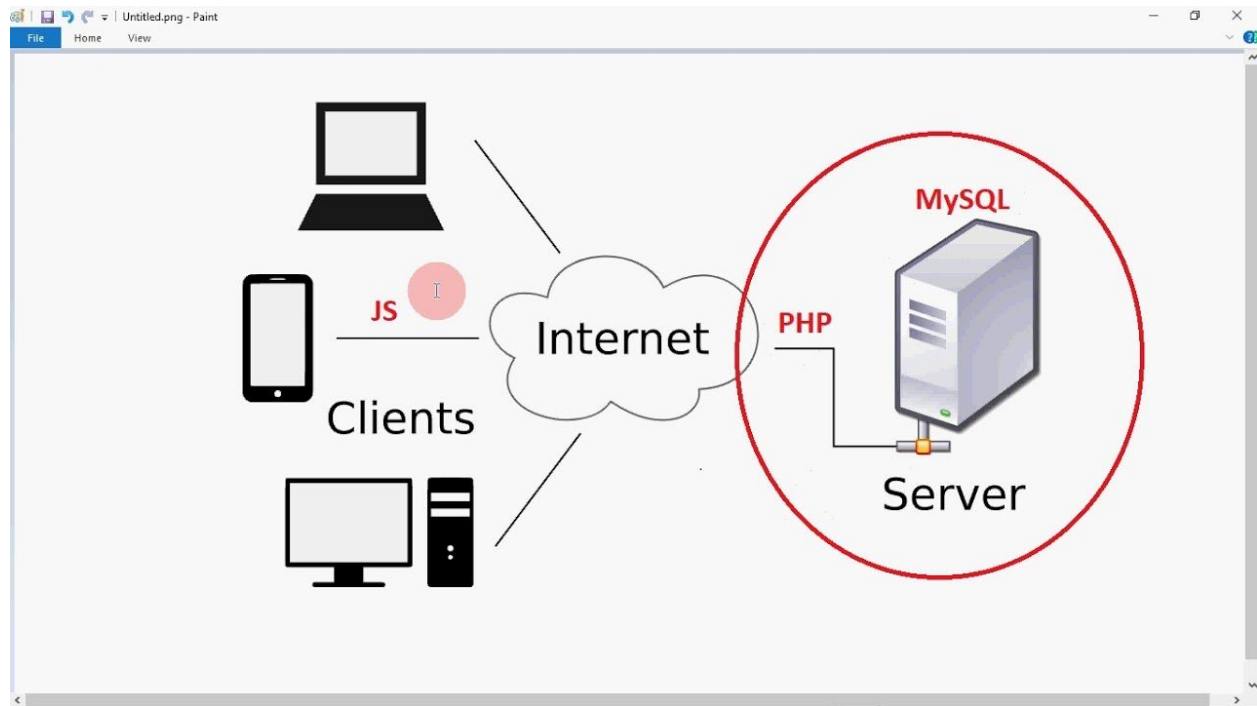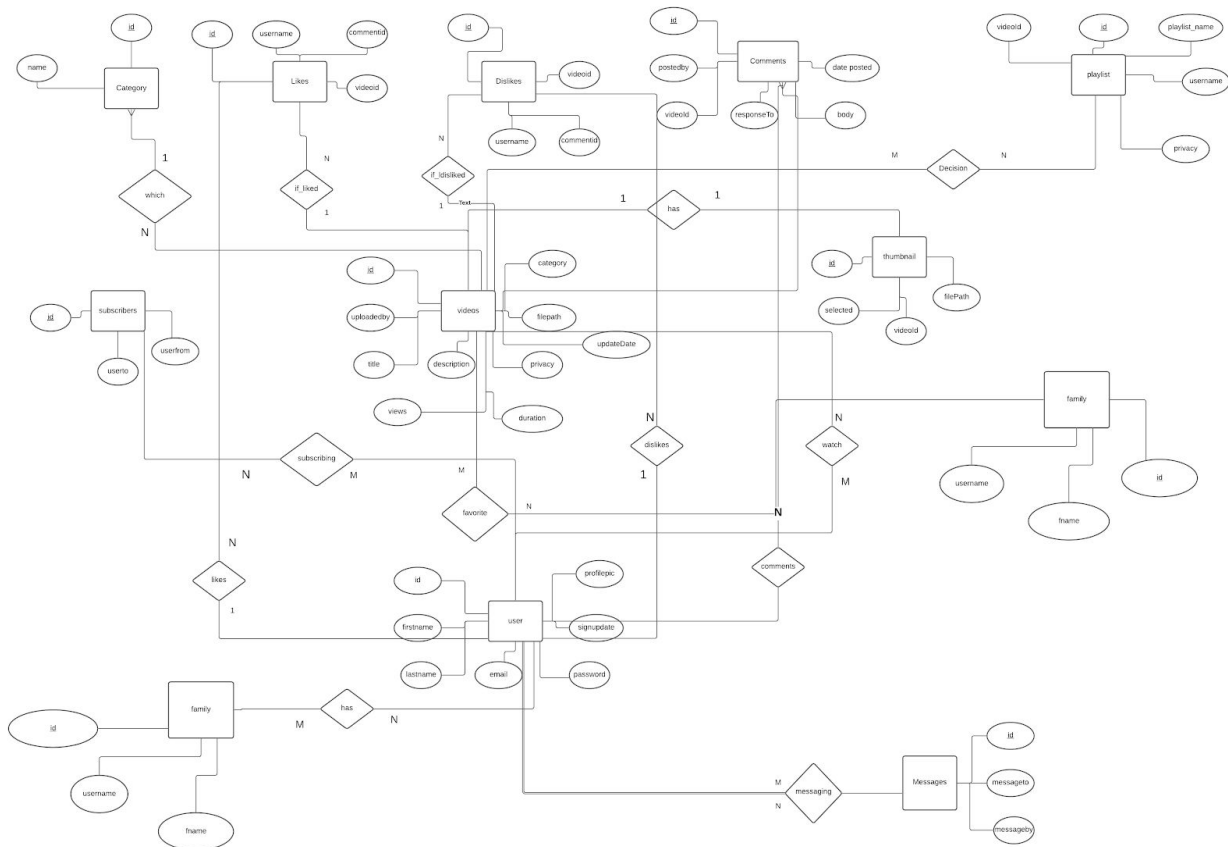

In this project, we have mainly used these following tools.
- PHP Programming Language
- JavaScript
- HTML4/HTML5
- SQL
- MySQL
- Bootstrap
- Jquery
- phpMyAdmin
- XAMPP Server

Major Learning Outcomes:

- Full understanding of how databases are used, represented, interacted, and returned.
- How webpages are made.
- How significant functions like, chat, comment, like, block are made.
- How to combine multiple, session pages and DB connections.
- Writing Complex queries

ER Diagram:

Database Schema:

meTube categories
- id : int(11)
- name : varchar(50)

meTube subscribers
- id : int(11)
- userTo : varchar(50)
- userFrom : varchar(50)

meTube dislikes
- id : int(11)
- username : varchar(50)
- commentid : int(11)
- videoId : int(11)

meTube comments
- id : int(11)
- postedBy : varchar(50)
- videoId : int(11)
- responseTo : int(11)
- body : text
- datePosted : datetime

meTube users
- id : int(11)
- firstName : varchar(25)
- lastName : varchar(25)
- username : varchar(25)
- email : varchar(100)
- password : varchar(255)
- signUpDate : datetime
- profilePic : varchar(255)

meTube videos
- id : int(11)
- uploadedBy : varchar(50)
- title : varchar(100)
- description : varchar(1000)
- privacy : int(11)
- filePath : varchar(500)
- category : int(11)
- uploadDate : datetime
- views : int(11)
- duration : varchar(20)

meTube playlists
- id : int(11)
- Playlist_name : varchar(10)
- username : int(11)
- privacy : int(10)
- videoId : int(11)

meTube friend
- id : int(11)
- username : varchar(50)
- fname : varchar(50)

meTube family
- id : int(11)
- username : varchar(50)
- fname : varchar(50)

meTube favourites
- id : int(11)
- videoid : int(11)
- username : varchar(50)

meTube thumbnails
- id : int(11)
- videoid : int(11)
- filePath : varchar(250)
- selected : int(11)

meTube messages
- id : int(10)
- messageTo : varchar(50)
- messageBy : varchar(50)
- body : text
- datePosted : datetime

meTube block
- id : int(10)
- blockedBy : varchar(50)
- blockedTo : varchar(50)

meTube likes
- id : int(11)
- username : varchar(50)
- commentid : int(11)
- videoId : int(11)

Functional Design:

1. editvideo.php:
   - This page confirms, if the video uploaded successfully or not.
   - If video uploading succeeded, then prompts a success message, else prompts an error message.

2. index.php:
   - Loads the header.
   - Displays the video grid and the side navigation bar after the header.

3. logout.php

- Logging out from youtube.
- Session ends.

4. phpfile.php
   - Prompts messaging.
   - Establishes a connection between database and frontend, with profile username and userligedIn values as the parameters.
   - Prompts error, if there is a trouble with the connection.

5. search.php
   - Enables search prompt
   - Returns a message saying that "You must enter a search term" , if the user clicked on the empty search option.
   - When the term matches, it will return according to the highest number of views, or by the most recent ones.
   - Users have the option to change as they prefer.
   - If no results found, return the message saying "No results found".

6. settings.php
   - Prompts to change the first name, last name, email.
   - If change is not successful, prompt an error message, saying "Something went wrong".
   - Prompts to change the password.
   - For changing the password, user need to enter the previous password, then only he can change the password.
   - Need to confirm the password, by entering the password twice.
   - By default first name, last name and email is provided.

7. signin.php
   - Asks the user for username and password.
   - If the credentials match with the database, then the user will be logged in.
   - Else, Error message will pop up.

8. signup.php
   - A form will drop off, to requesting firstname, lastname, username, email, password and confirmation of password.
   - If all the credentials match the required criteria, then the account will be created.
   - Else, the respective error message will pop up.

9. subscriptions.php

- Provides a subscription button and it's a toggle button. After subscribing, when you click on the button it will change to unsubscribe. Before that it shows for subscription.

10. trending.php
  - The Trending page will show the most watched videos from the last week.
  - If there are no trending videos, it will popup a message saying "No trending videos to show".

11. upload.php
  - Calls the video upload drop down box. This will also assist the user, while uploading the video.
  - If a video is uploaded successfully, a suitable successful message will be prompted, else respective errors will be displayed.

12. watch.php
  - This is the page, where the video will be watched by the user.
  - In this page, video will be played and below that there will be a comment section, to post comments on that video.

13. watchfavorites.php
  - If no url passed here, then a message will be displayed saying "No url passed into page".
  - On this page, saved favorites will be displayed and in sequence favorite videos will start playing. Even here, video players will be there, the comment section and suggested videos are lined up.

14. block.php
  - using Jquery we have implemented this, we have added a toggle button of block.
  - Once blocked a particular user is blocked, the button will turn to unblock. And he won't be seen ever on the page.

15. getCommentreplies.php
  - In this page, posting comments and replying for that particular comment can be done.
  - Like and disliking teh comment is also programmed.

We have more php pages than what we are mentioning in the report. Since, the pages are nested, and for handling the small functions, we have made a separate function. Functions are the same, but PHP pages are more.

IMPLEMENTATIONS:

As you login into the page, this looks most similar to youtube as you can see below.
First Page
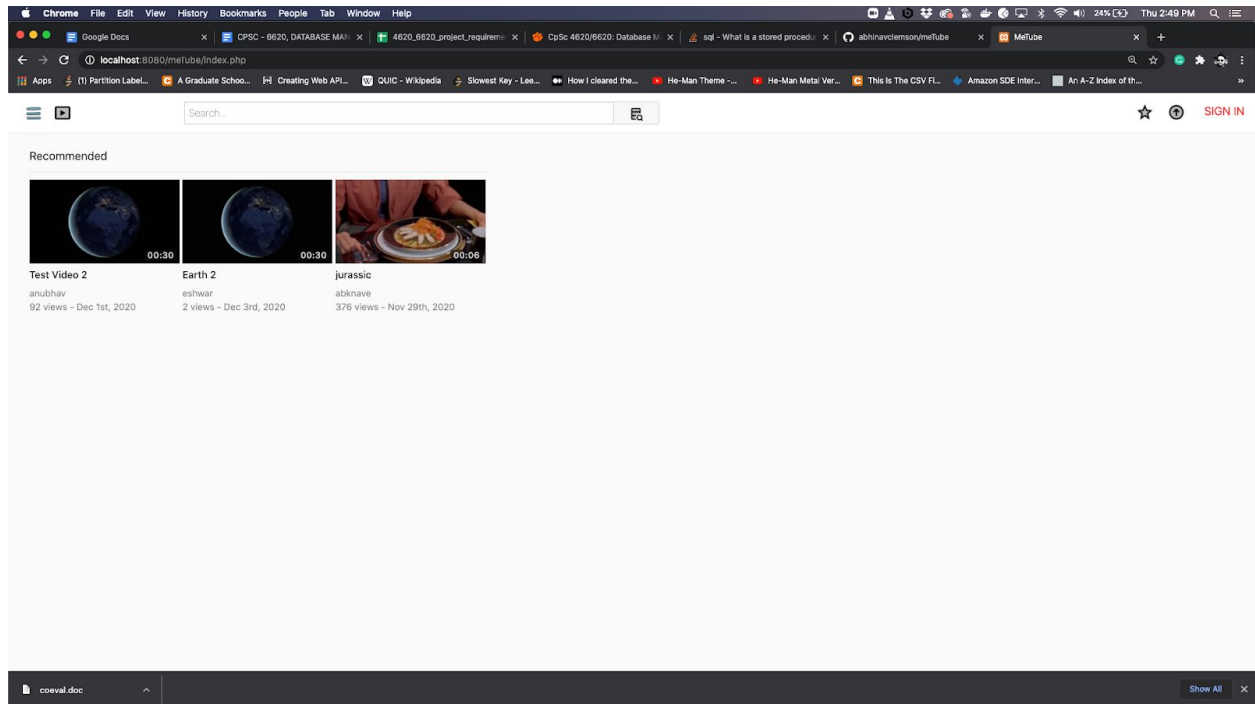
The user icon in the top right corner needs to be clicked to open, for user login page. Here, the account details form is present as shown below. Basic information of the user is added to the database.

User creation page
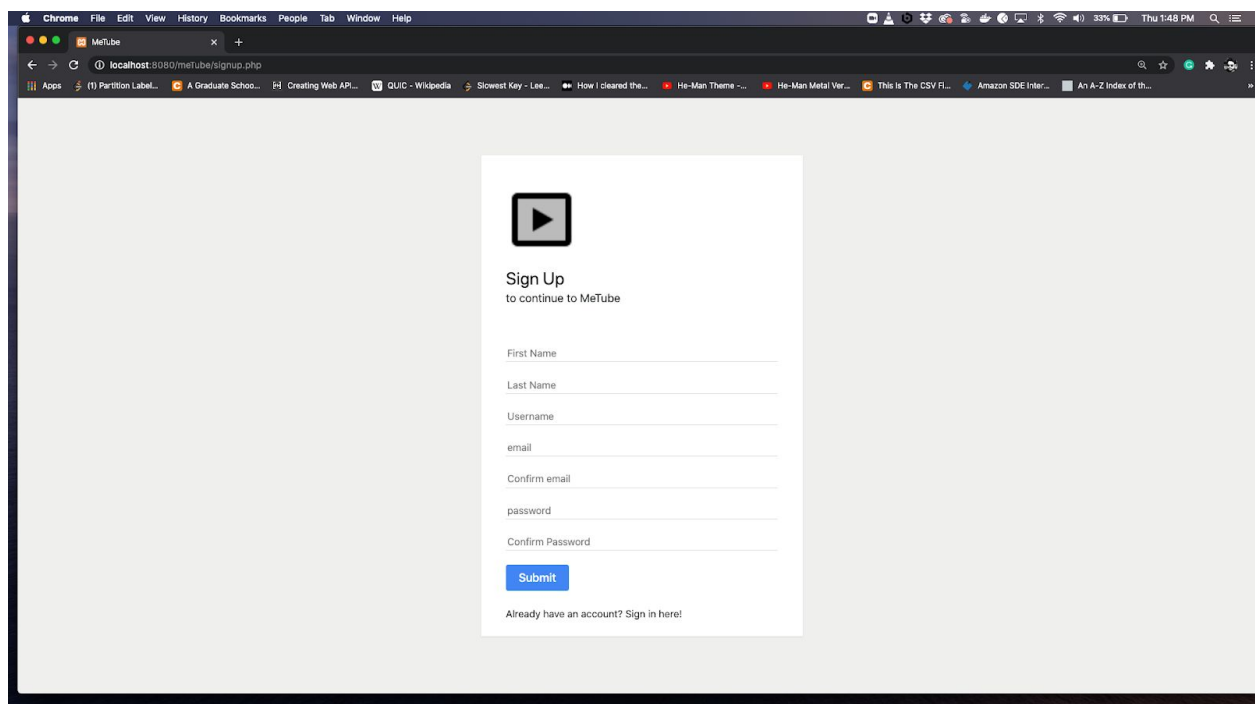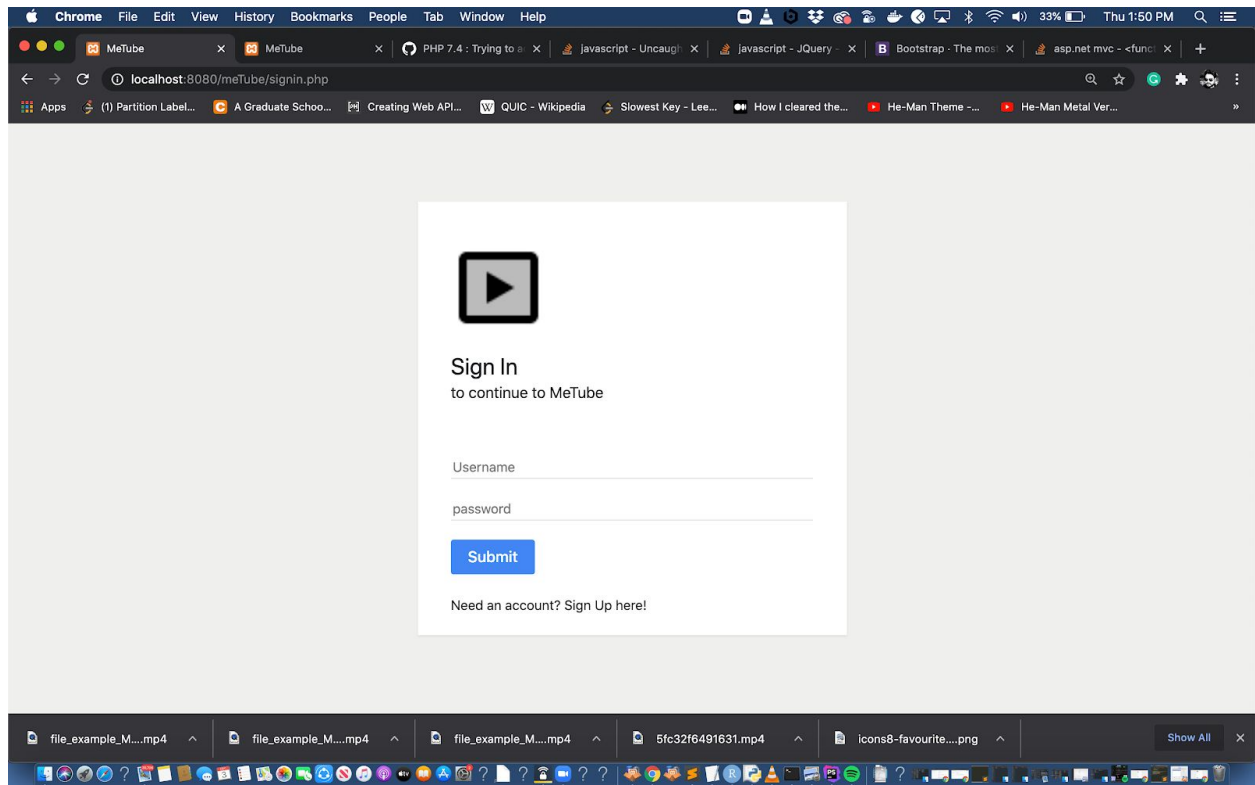
Profile page is used to show the details about the user profile. Messages, subscription and block buttons are present on this page. In the profile page, you can view details about the user. Even the messages for the particular user comes in his profile page.

Profile Page
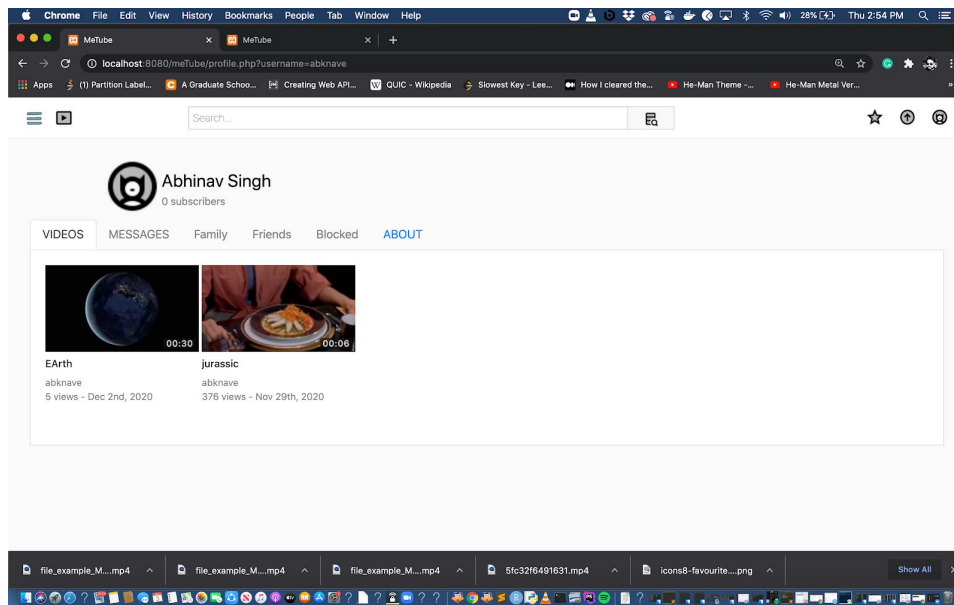
Self profile view.
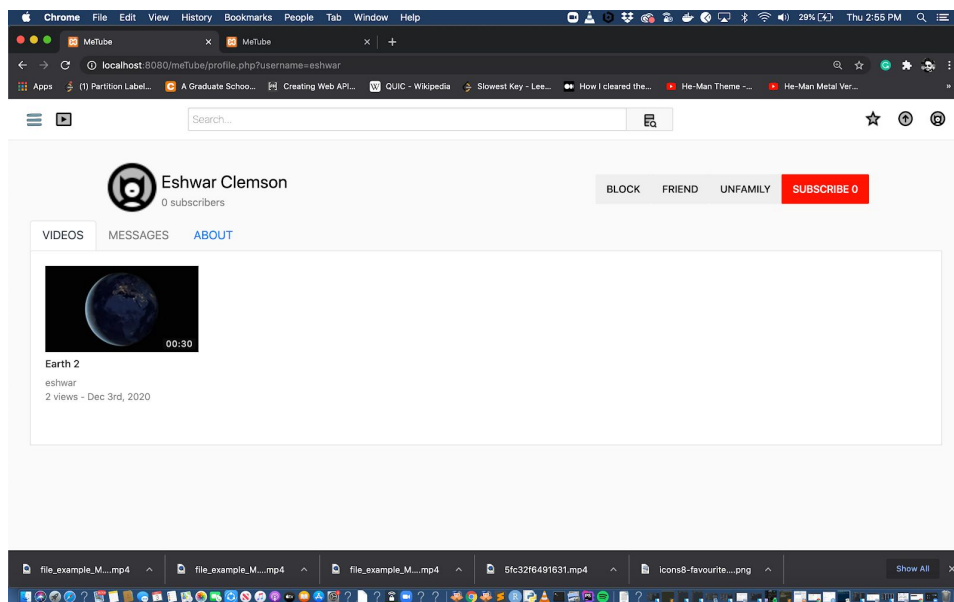
Profile view of another user, from a user account.
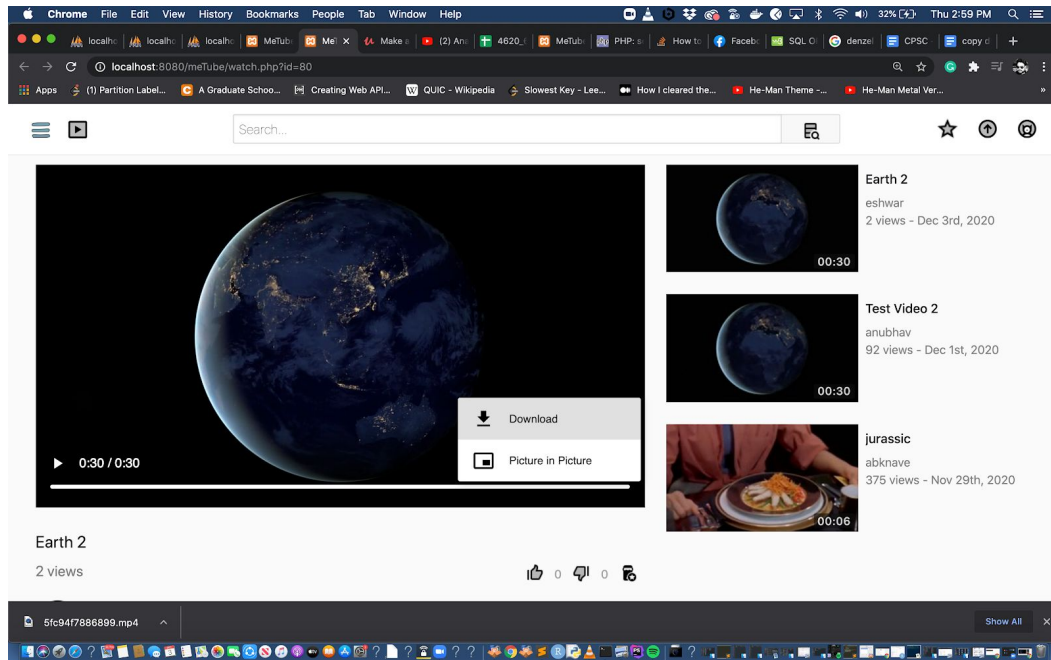
Video player page, in which number of views to the video is also visible.
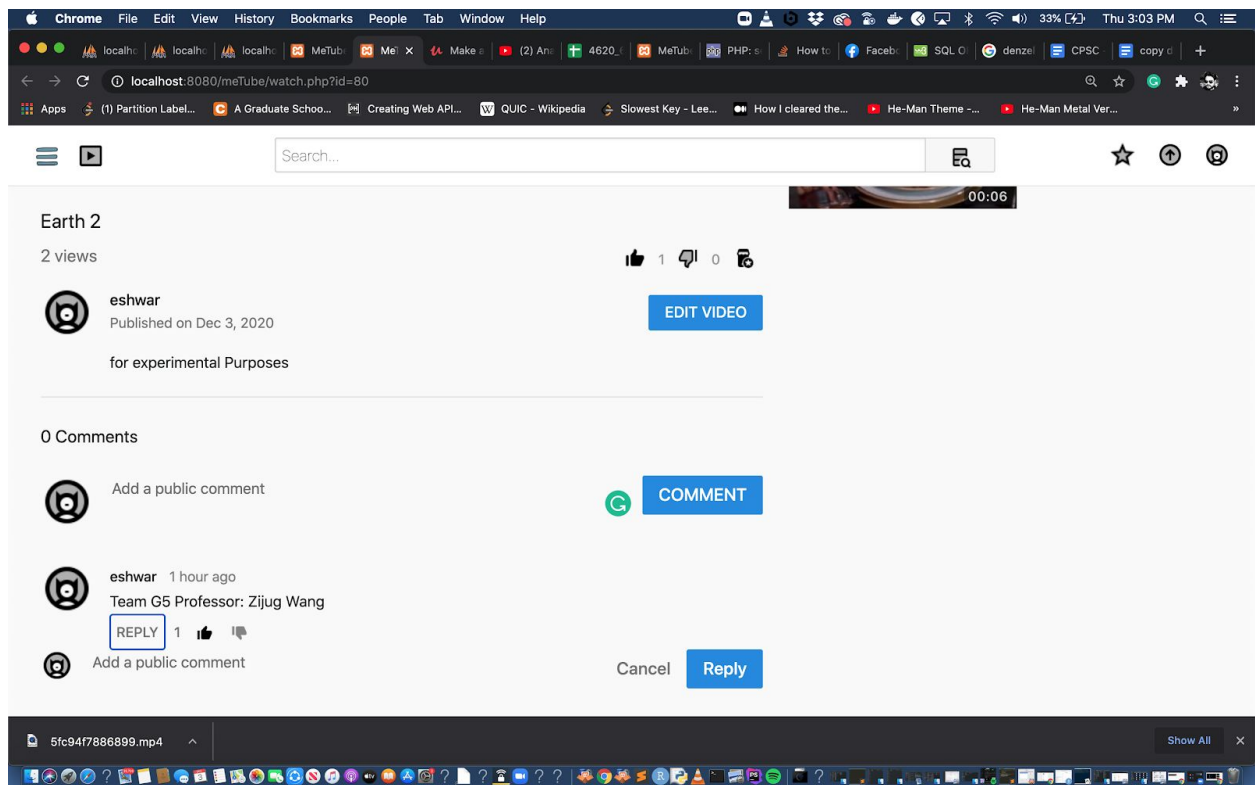
Like and dislike functionality, commenting and adding the video to favorites.
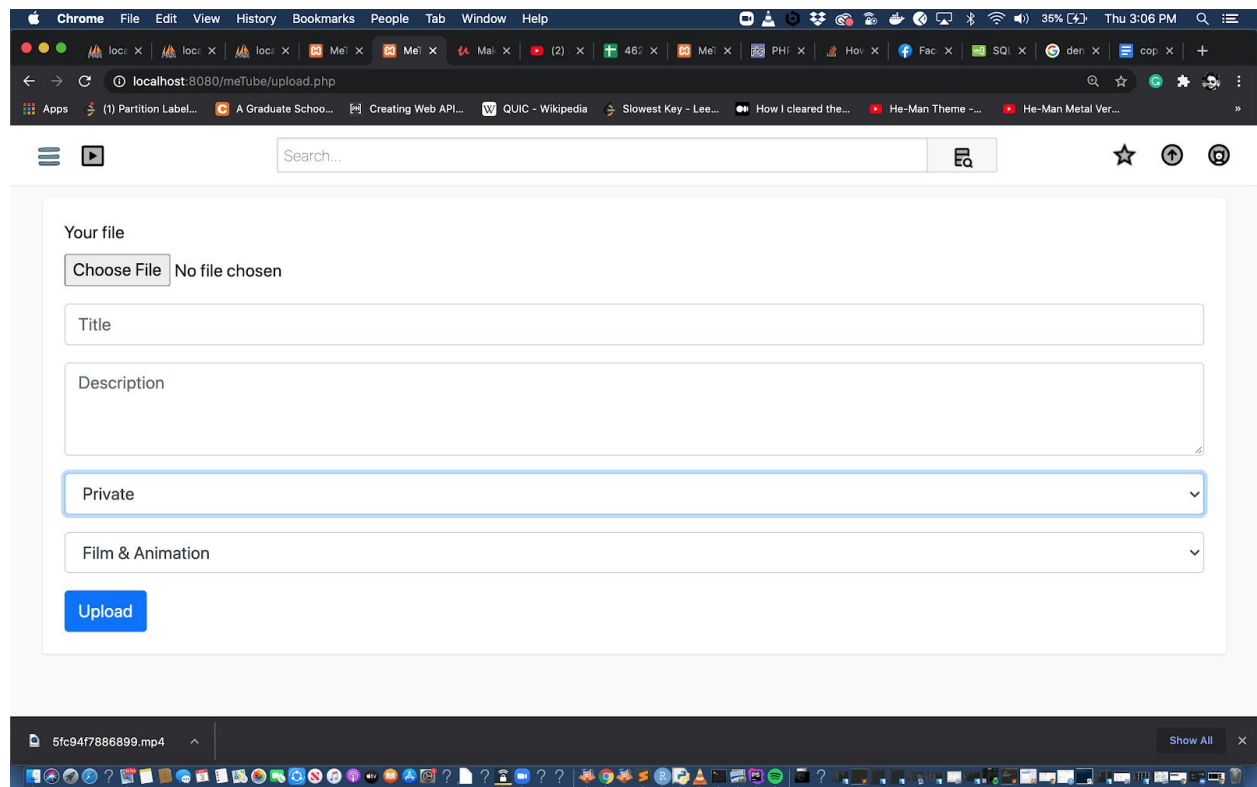
Video uploading page.
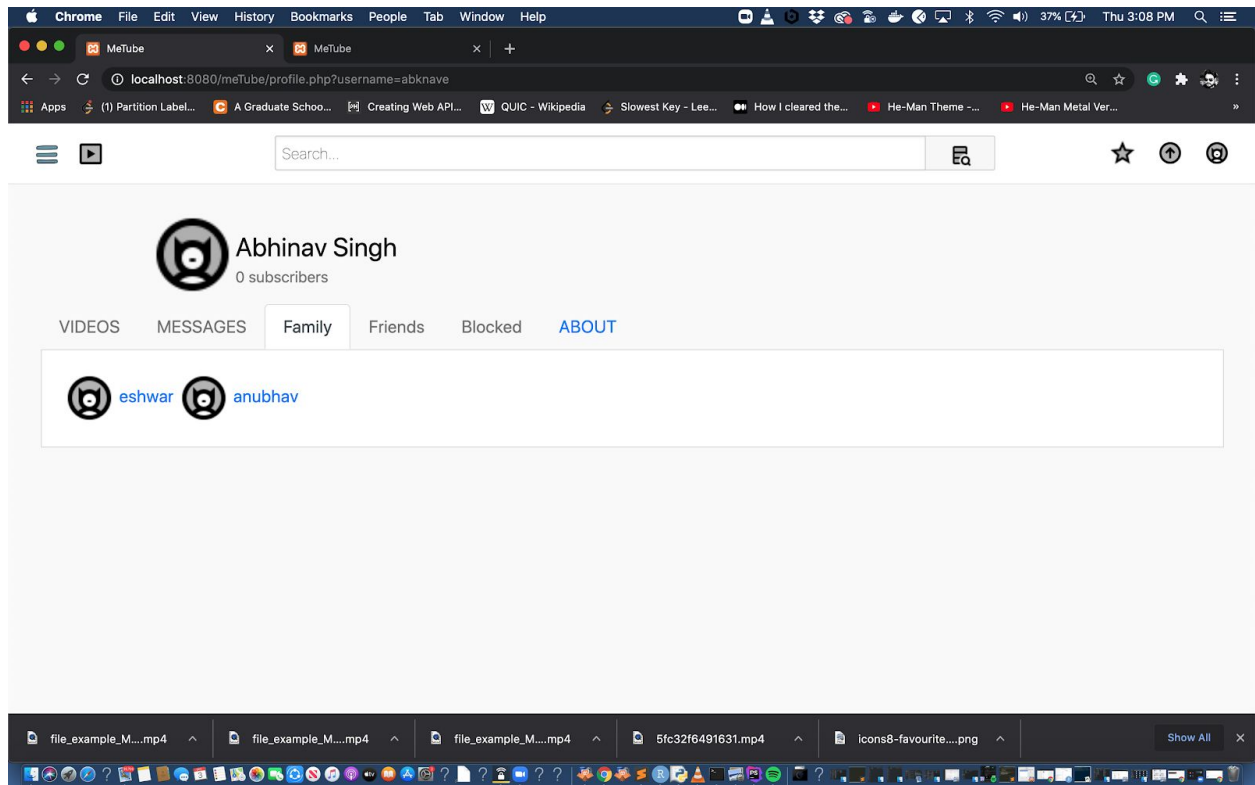
Family, friends, contact list. Block also can be seen here.

Videos uploaded by subscribed channels
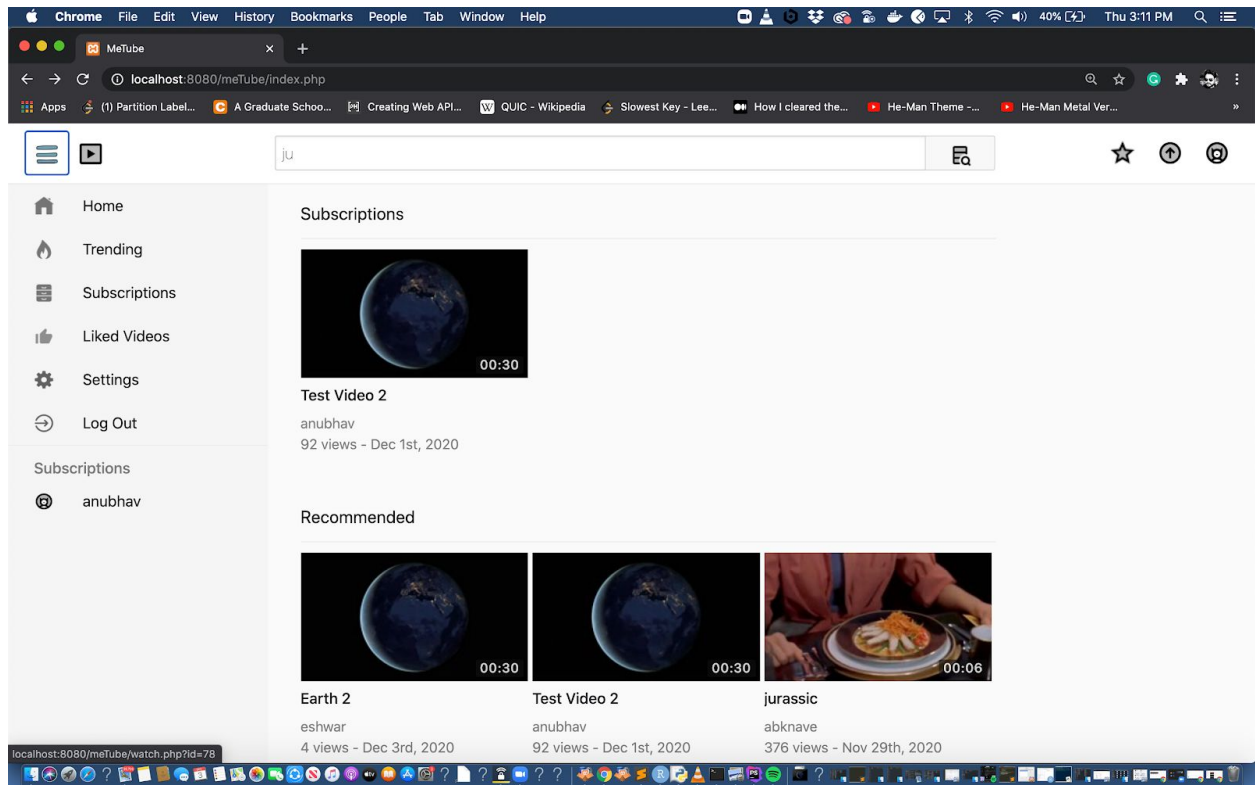
Trending feature, shows the most of watched videos from the last 7 days.

Query used to retrieve the most watched video in last 7 days,

```
SELECT * FROM videos WHERE uploadDate >= now() - INTERVAL 7 DAY
                                ORDER BY views DESC LIMIT 15
```

By clicking on the favorite, it will take you to the favorites page.
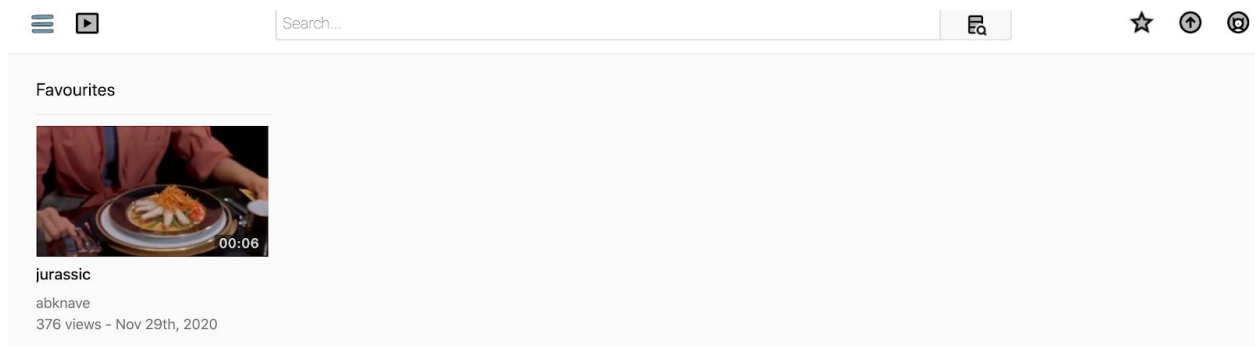
Video category is shown in the below image.
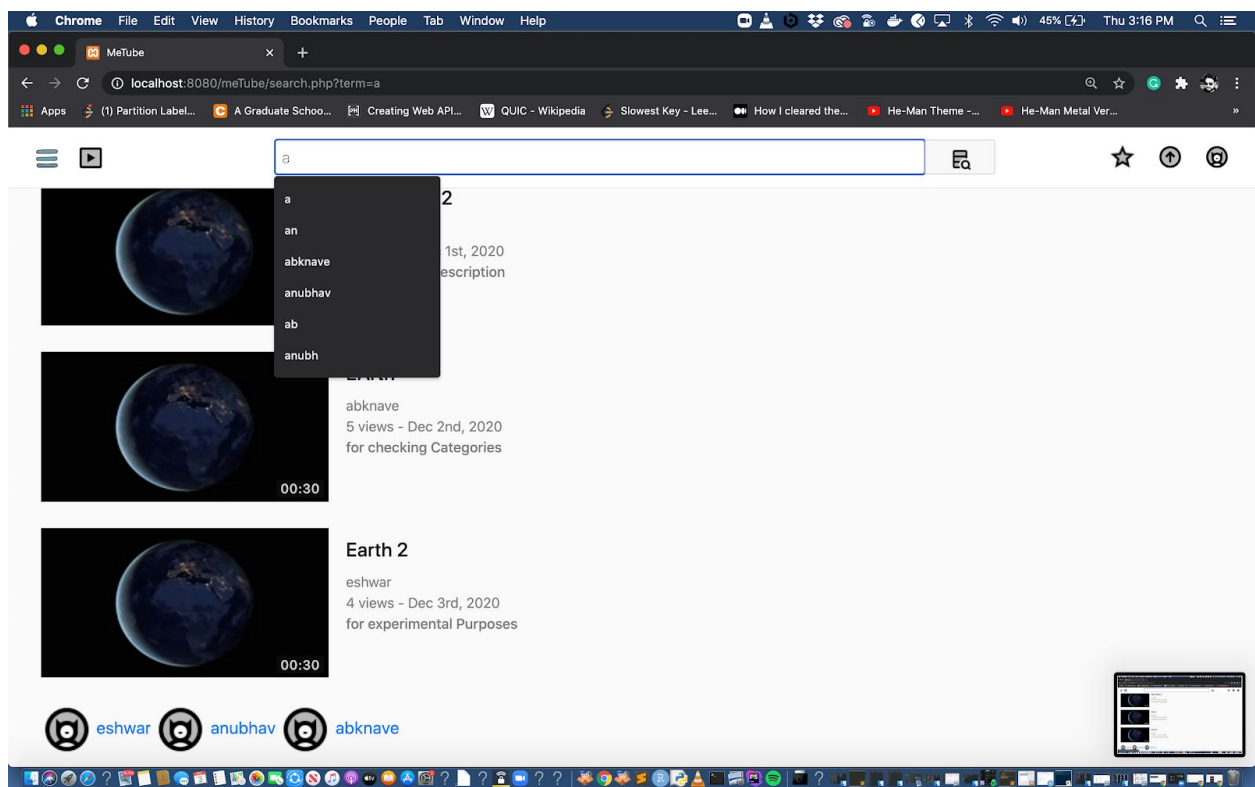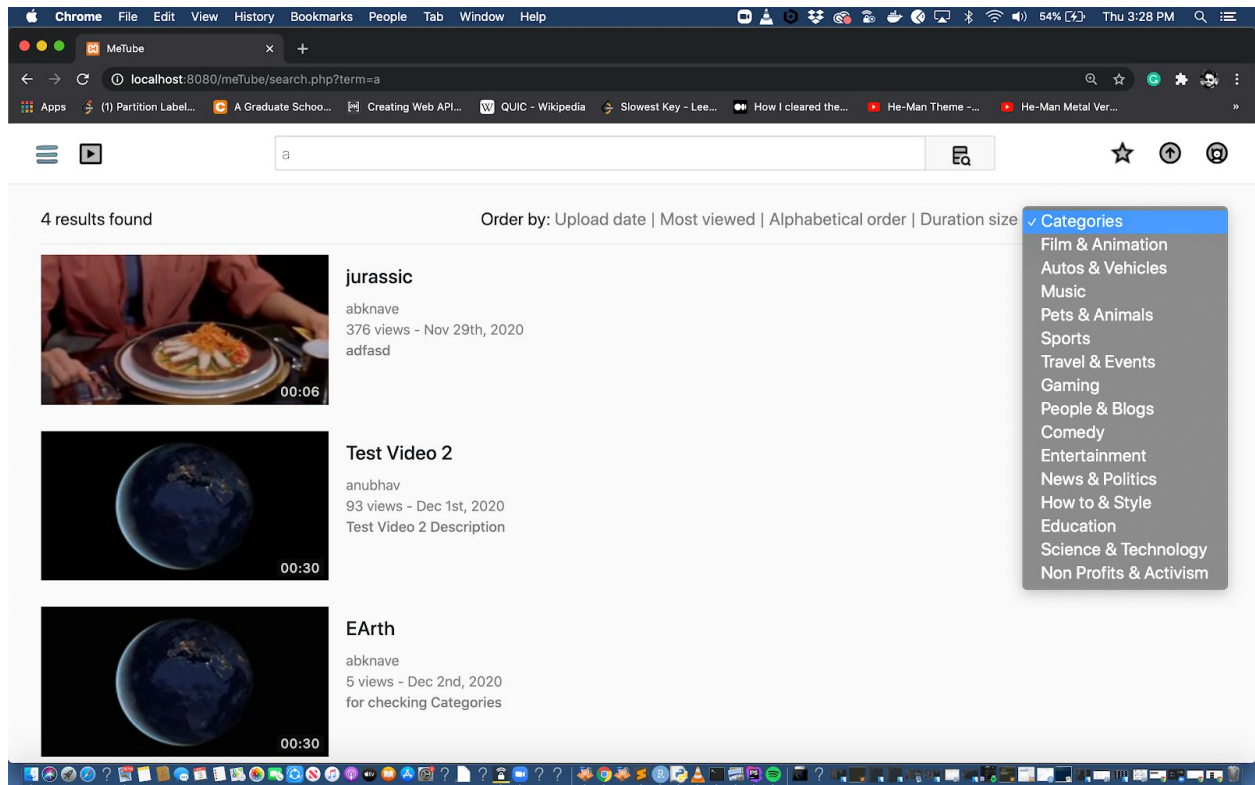
While typing in the search box, a drop down suggestion of relative keywords will appear as shown above.

Query related to search drop down

```
SELECT * FROM videos WHERE (title LIKE CONCAT('%', :term, '%')
                           OR uploadedBy LIKE CONCAT('%', :term, '%') ) AND
category=$category ORDER BY $orderBy $alpha_order
```

Query to search matching user

```
SELECT username FROM users WHERE (firstName LIKE CONCAT('%', :term, '%')
                        OR lastName LIKE CONCAT('%', :term, '%')
                        OR username LIKE CONCAT('%', :term, '%') )
                        ORDER BY username $alpha_order
```

The below screenshot shows the chatting with the another user.
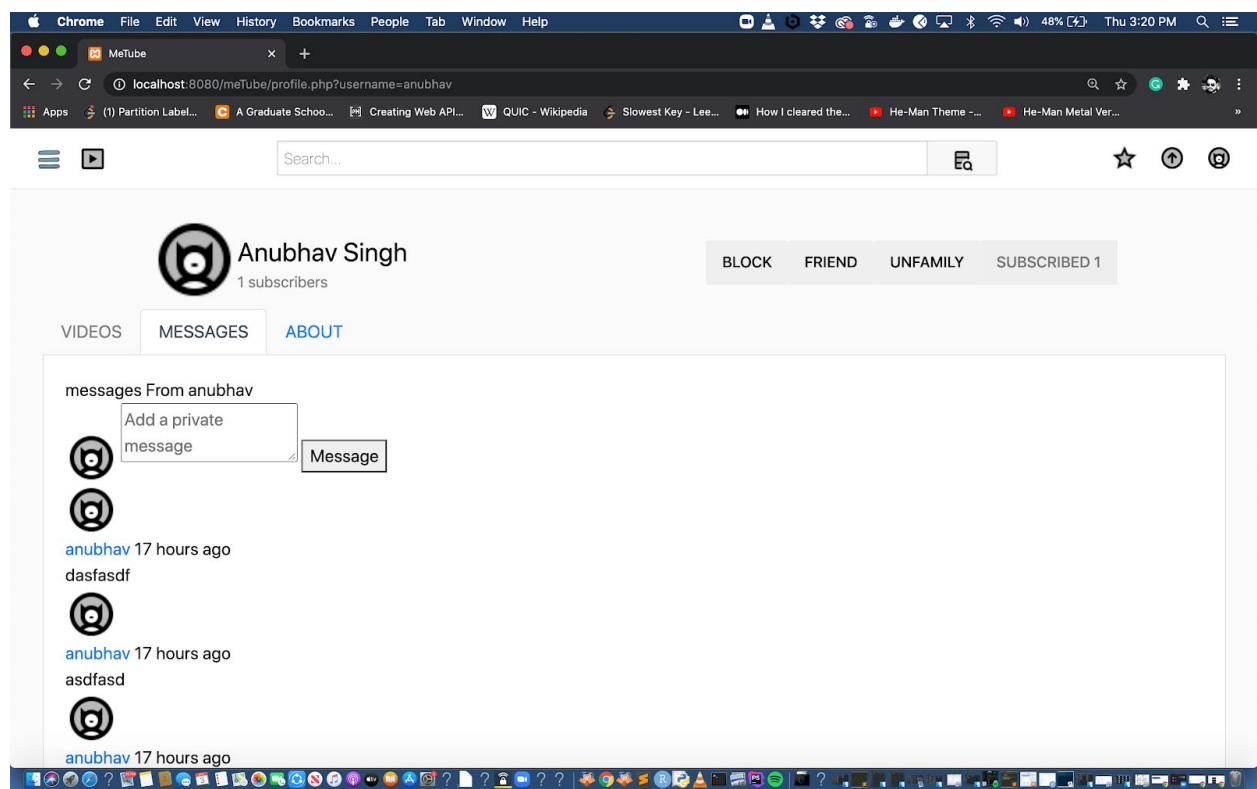
 Query for insertion of new messages.

messagesBy is the current logged in user, and **messageTo** is the user of designated profile ,to which Logged in user wants to send a message to.

```
INSERT INTO messages(messageBy, messageTo, body) VALUES(:messageBy, :messageTo,
:body)
```

Query to fetch messages.

```
SELECT * FROM messages where (messageBy=:messageBy and messageTo=:messageTo) OR
(messageBy=:messageBy and messageTo=:messageTo)
```

To edit the video in between.
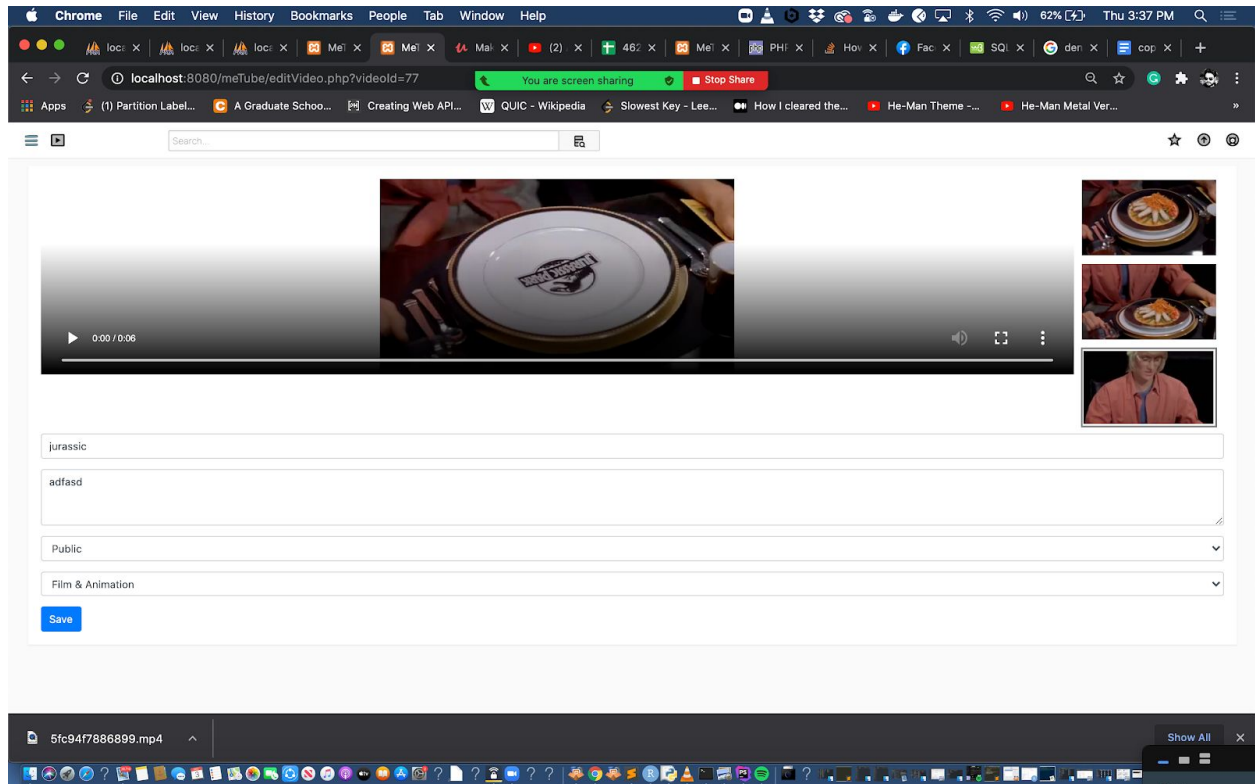

```
SELECT * FROM thumbnails WHERE videoid=:videoId
```

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.getElementBy 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
1. User registration : Verified
2. Sign-in Sign-out :Verified
3. Profile update :Verified
4. Contact List : Verified
5. Contact list organization (friend, family, favorite, etc.) : Verified
6. Blocking a user in the contact list. :Verified
7. Upload multimedia files Web interface for Meta data of uploaded files such as title, description, keywords, category, etc. : Verified
8. Download : Verified
9. View media files : Verified
10. Sharing Mode : Verified

11. Blocking : Verified
12. Browse by category : Verified
13. Channel : Verified
14. Playlists : Verified
15. Favorite Lists : verified
16. Most viewed Most recently uploaded Ordering media files in different ways (such as name, size, uploading time) : Verified
17. Messaging Exchange between user Commenting on media file Media rating : Verified
18. Group discussion related to multimedia files: Verified
19. Keywords-based search Word cloud Media recommendation : Verified
20. Feature-based media search (such as file size, type, and other image properties) : Verified

Result: Passed

APPENDIX

Search page code

```php
<?php
require_once("includes/config.php");
require_once("includes/classes/FormSanitizer.php");
require_once("includes/classes/Account.php");
require_once("includes/classes/Constants.php");

$account = new Account($con);


if(isset($_POST["submitButton"]) ){
    $firstName = FormSanitizer::sanitizeFormString($_POST["firstName"]);
    $lastName = FormSanitizer::sanitizeFormString($_POST["lastName"]);

    $username = FormSanitizer::sanitizeFormUsername($_POST["username"]);

    $email = FormSanitizer::sanitizeFormEmail($_POST["email"]);
    $email2 = FormSanitizer::sanitizeFormEmail($_POST["email2"]);

    $password = FormSanitizer::sanitizeFormPassword($_POST["password"]);
    $password2 = FormSanitizer::sanitizeFormPassword($_POST["password2"]);

    $wasSuccessful = $account->register($firstName,$lastName,$username,$email,$email2,$password,$password2);

    if($wasSuccessful){
        //Success
        //Redirect user to index page

        //Creating the session variable
        $_SESSION["userLoggedIn"] = $username;

        //to redirect to index.php
        header("Location: index.php");

        echo "Success!";
    }
    else{
        echo "Fail!";
    }

}


function getInputValue($name){
    if(isset($_POST[$name])){
        echo $_POST[$name];
    }
}
```

Settings.php

```php
<?php
require_once("includes/header.php");
require_once("includes/classes/Account.php");
require_once("includes/classes/FormSanitizer.php");
require_once("includes/classes/SettingsFormProvider.php");
require_once("includes/classes/Constants.php");

if(!User::isLoggedIn()) {
    header("Location: signIn.php");
}

$detailsMessage = "";
$passwordMessage = "";
$formProvider = new SettingsFormProvider();

if(isset($_POST["saveDetailsButton"])) {
    $account = new Account($con);

    $firstName = FormSanitizer::sanitizeFormString($_POST["firstName"]);
    $lastName = FormSanitizer::sanitizeFormString($_POST["lastName"]);
    $email = FormSanitizer::sanitizeFormString($_POST["email"]);

    if($account->updateDetails($firstName, $lastName, $email, $userLoggedInObj->getusername())) {
        $detailsMessage = "<div class='alert alert-success'>
                            <strong>SUCCESS!</strong> Details updated successfully!
                        </div>";
    }
    else {
        $errorMessage = $account->getFirstError();

        if($errorMessage == "") $errorMessage = "Something went wrong";

        $detailsMessage = "<div class='alert alert-danger'>
                            <strong>ERROR!</strong> $errorMessage
                        </div>";
    }
}

if(isset($_POST["savePasswordButton"])) {
    $account = new Account($con);

    $oldPassword = FormSanitizer::sanitizeFormPassword($_POST["oldPassword"]);
    $newPassword = FormSanitizer::sanitizeFormPassword($_POST["newPassword"]);
    $newPassword2 = FormSanitizer::sanitizeFormPassword($_POST["newPassword2"]);

    if($account->updatePassword($oldPassword, $newPassword, $newPassword2, $userLoggedInObj->getusername())) {
        $passwordMessage = "<div class='alert alert-success'>
                            <strong>SUCCESS!</strong> Password updated successfully!
                        </div>";
    }
    else {
        $errorMessage = $account->getFirstError();

        if($errorMessage == "") $errorMessage = "Something went wrong";

        $passwordMessage = "<div class='alert alert-danger'>
                            <strong>ERROR!</strong> $errorMessage
                        </div>";
    }
}
```

Header.php

```php
<!DOCTYPE html>
<html>
<head>
    <title>MeTube</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E2
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakI
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
    <script src="assets/js/commonActions.js"></script>
    <script src="assets/js/messageActions.js"></script>


</head>
<body>

<div id="pageContainer">

    <div id="mastHeadContainer">
        <button class="navShowHide">
            <img src="https://img.icons8.com/doodle/48/000000/menu.png"/>
        </button>

        <a class="logoContainer" href="index.php">

            <img src="https://img.icons8.com/material-two-tone/48/000000/video.png" title="me
    tube" alt="meTube"/>
        </a>

        <div class="searchBarContainer">

            <form action="search.php" method="GET">
                <input type="text" class="searchBar" name="term" placeholder="Search...">
                <button class="searchButton">
                    <img src="https://img.icons8.com/material-two-tone/24/000000/search-database.png" title="Search button" alt="Search"/>
                </button>
            </form>

        </div>





        <div class="rightIcons">
            <a href="favouritePage.php">
                <img src="assets/images/icons/star.png"/>
            </a>
            <a href="upload.php">
                <img src="https://img.icons8.com/material-two-tone/48/000000/circled-up.png"/>
            </a>


            <?php
            //if user is logged in it would return html code with link to user's profile picture
            echo ButtonProvider::createUserProfileNavigationButton($con, $userLoggedInObj->getUsername());
            ?>


        </div>

    </div>

    <div id="sideNavContainer" style="display:none;">
        <?php
        $navigationProvider = new NavigationMenuProvider($con, $userLoggedInObj);
        echo $navigationProvider->create();
        ?>

    </div>

    <div id="mainSectionContainer">

        <div id="mainContentContainer">
```