Cole Earl

Joshua Lacy

Wesley Fulford

CPSC 4910

MeTube Project

**System Design**

Our metube website is built off the preexisting framework provided to us. It uses apache, php, JavaScript, and MySQL. Most interactions between the front end and the server are facilitated using MySQL queries to the databases using php. A restriction would be MySQL’s inability to store array values and create unique structures on demand. This would be better for comments on videos or direct messages between users as they are structurally simple and would be easier to manage the content of in an admin program.

**Function Design**

Logout: Php sessions is cleared via a quick redirect

Registration: Registration checks that users have filled out every box and the uses a query to check if the username is unique. If the username is unique and the 2nd password matches the first then the user information is added.

Profile: The form checks if the user has filled any of the fields. If they 1 or more is then it updates the field. Once again the two passwords are checked to see if they are identical and if they aren’t the password won’t be updated. Username is also checked again to see if it’s unique.

Dms: The page first will fill a list of all users and the user must select one to receive the comment and fill out the comment. Upon doing so the comment will be added to the file and the receiving and sending user will be able to see it and any other pervious comments they made or received.

Media (Comments): Comments allows a user to make a comment and see any previously made comments and who made them.

Media (viewing): Viewing a video allows a user to see the video or an image if it’s an image. There is also a message if the media type they attempt to view isn’t supported.

Upload: A user who uploads a file to the database can add a title, description, and keywords for the video. Keywords is hidden from viewing in the media as it is intended to be used for searching.

**Implementation details**

Logout: If user is logged in, clear session information and move them to index.html

Registration: If the user filled out all forums, the 2 passwords match, and the username is unique add them to the database and return them to the main page.

Profile: If a user fills out a field push it to the database unless it’s username or password. If username was changed make sure it’s unique, and if it’s the password both passwords match. Reload the page on completion.

Dms: Query the database to pull all users, loop thru and display in dropdown. User must have set the user to be sent to for a message to be sent. (Since both are filled automatically no changes need to be made to send a message). The page will then display the message. The page also displays all DMs sent and received by the user in the order which they were added FIFO via querying the db and looping thru any that have the currently logged in user as the sender or receiver.

Media (Comments): Comments checks if the user has added a comment and then pushes it to the comment table with their username. Any comments made on the media will be queried by the media’s mediaid and any found will be displayed with the user who added the comment to the video.

Media (Viewing): The database will pull the requested media and mediainfo displaying the media, title, and description. The function will attempt to allow the user to view a video if its mp4, ogg, or webm. Pictures are supported, but if it finds a file type that isn’t the function will output that the media isn’t viewable and should be downloaded.

Upload: The database only checks that the file is unique nothing else as the videoid assigned in upload is used to match the video information to the video. A user doesn’t need to fill out any of the information for the video for it to be uploaded. It will be inserted into the DB as long as the file is unique to the user.

**Test Cases & Results**