Team Sanguine Thirst / reelFeels / https://github.com/mmacquarrie/reelfeels

Members: Zack Bebbington, Megan MacQuarrie, Matt Rubin, Humad Syed

Team Write Up

Overview

The idea is to have a platform where content creators, similar to those on YouTube or Twitch, can either upload or share their content for users to view. Apart from the usual metrics such as user views and likes/dislikes, the content creators will be able to see in graph form how the users felt while viewing content. This is extremely valuable information for content creators as it allows them to determine weaknesses in their content and equips them with the data necessary to produce better content. We have not deviated from our proposal.

Design Overview

Guests viewing our website see a link to the login page in the header. If they go to this page and click the link to the sign up page, they can fill out the requested information (email, username, photo, password) and create an account. Once they're logged in (which a successful sign-up does automatically), they will see a link to their profile and a link to log out in the header. If they go to their own profile, they will be able to edit the information there. If they click on a video that they uploaded, they will be able to edit the video's information or delete the video. These options are only available to the owner, and others do not see the edit/delete buttons. If they happen to go to the /edit or /delete urls, they see an error page. Logged in users can upload new videos and comment on videos. Guests see a link to the sign up page in place of both.

Problems/Successes

One of the main problems we encountered was linking our profile model to django's built in user model. Since we already had records in the database, we ran into a lot of errors during migration. It took about an hour to untangle and we ultimately had to delete all of our records and edit the database file directly. Another obstacle we faced was the shorter time period allotted for this project. This, in addition to scheduling conflicts, made it so that we couldn't meet outside of class like we normally do. As for successes, we successfully implemented all of the forms we wanted to add and were able to connect to Affectiva (an important first step for our team choice feature).

Team Choice

For the team choice portion of the project, we will be integrating an external API, Affectiva, into our web application. The API will record a user's emotions while they watch a video and we will store this information in our database. We will also link this data to google charts so that it can be displayed in real time to the user, and so that we can display overall stats for a particular video.

Individual Writeups

Zachary Bebbington: 25%

I worked on creating a functioning comment form. This involved the assurance that only those that are logged in can post a comment and those that are not are prompted to sign in to make a comment. Additionally, I created a form that allows users to edit their current profile information. This form is only available to the users themselves, such that a user can only update their own profile information.

Megan MacQuarrie: 25%

I worked on creating a functioning sign up page. This involved extending the built in User Creation Form to allow both the user and their respective profile to be created simultaneously. Additionally, I created a form to edit a video's information and a form to delete a video. Both are restricted to the owner, and if a non-owner tries to access the /edit or /delete urls they get a 404 error. I also cleaned up some of the front end scripting and styling and completed our team write up. Finally, I looked over the project specifications, created specific tasks for the major project requirements and outlined the various changes we would need to make to our existing views.

Matt Rubin: 25%

I worked on setting up the video upload form that allows a signed-in user to paste in a YouTube URL, and it will tell the user if they didn't enter a valid YouTube URL. If they did enter a proper YouTube URL, they can then give the submission a Title and Description, and click the Share button to complete the upload and save it to the database, allowing all other users to view it.
I also worked on the initial implementation of the Affectiva API on the video view page. I got the page to activate the webcam feed and tell you if it couldn't connect to the webcam. It it could, then it will start to display five of your emotion readings by showing the numbers (out of 100%) as well as the bar graph.

Humad Syed: 25%

I worked on implementing the login and logout functionality in our application. I made sure that users could log in using their credentials or be told that their credentials were incorrect. I also updated some of the views to be tailored to whether or not the user is logged in. For example, if a user is logged in, they will be able to access the upload page, and will also see a link to their profile in the header next to a logout button. Users who are not logged in will not see the profile and logout buttons, and will be redirected to the login page when they try to upload something.