User Interface Software Project 5 Writeup - Divit Singh & Sanchit Chadha

This web application is for anyone who is looking to view and share their music with a community of music artists. Users can create an account to interact with this web application. As a teaser, when a user is not logged in, they can see the data in the community through the 'Community' page, however, they can't access any of the data. Only when a user is logged in, can they view and play albums and tracks from different users. CollabTunes also offers a plethora of other features such as providing different access levels for different user account roles, the ability to upload and download music files and the ability to comment on albums. We also implemented bidirectional collaboration between users of this application, and an activity or event feed that keeps track of what the user and their collaborators are uploading or changing on their account. We used the Twitter REST API to provide a twitter feed for users to see their own and their collaborators' tweets. We also implemented a Chrome browser extension that will be discussed further.

The browser extension is designed to be used by a user whose primary use case is to access a YouTube video, download it in MP3 format, create a new album on their account, and then be able to add, edit or delete track(s) through the interface. The extension is designed as a shortcut for a CollabTunes user to quickly access their tracks and albums and be able to add a song that they really enjoyed. We used an API from youtubeinmp3.com that allowed us to pass in a YouTube URL and convert the video at a click of a button. The only caveat that we found out about this API is that if the video has never been converted before through the website's API, then the extension will direct the user to youtubeinmp3.com to go through the conversion and download process. However, if the API has previously converted the video, then a one click download is good to go. The extension also features a notification system that pops up pending collaboration requests from other users using CollabTunes. CRUD operations are provided for tracks since we wanted the primary focus of this extension to be the user's interaction with their tracks.

The comments that we received from our previous submission of the project recommended we split up our monolithic JavaScript file into multiple smaller files. In addition, the comments also recommended that we not have a separate file for each controller action. We strongly agreed with these recommendations since these changes would not only provide better organization and more intuitive structure of our code base, but also make maintenance much easier. It is only natural if actions are related to a certain object, then it should be its own stand-alone file. Thus, for this submission, we condensed all of the controller actions to files which contained all the actions pertaining to that object. For example, we have an Album Controller, User Controller etc. We also broke up our JavaScript file into smaller files which contained functions necessary for a specific page.

We decided to implement the Twitter API for this project. A user need not have a Twitter account when they make an account with us. However, if they do wish to add their username, it will allow them to see their personal timeline on their My Music page under the Twitter Feed section. In addition, if the user has any collaborators that have Twitter usernames attached to their accounts, their timeline will also appear in the Collaborator Twitter Feed. The

Collaborative Twitter Feed contains the latest 3 tweets from all your collaborators that have a Twitter name attached to their account. The Collaborative Twitter Feed can be accessed from the Community Page as well as the Collaborators Page. This feed strictly contains the Twitter timelines of your collaborators, whereas the Twitter Feed in My Music strictly contains timeline of that specific user. We made this distinction because we wanted to make a user's profile page personal. The Twitter feeds that are displayed are dynamic. In fact, every 10 seconds, we ping Twitter's API to update the tweets. (We later found out that in a 15 minute window, we could ping the API a maximum of 300 times. Therefore, we increased the ping rate to 60 seconds.) Thus, if you are on the page, you can see new Tweets coming in without having to refresh the page. A user does not need to log on to Twitter to attach a Twitter handle to an account. Since all information related to Tweets is public knowledge and we are only providing functionality to display the tweets, we didn't see the point in enforcing a password to a Twitter handle. If a user wants to use another person's Twitter handle, the only thing that will happen is they will have an easy way to read the Tweets.

URL: http://ec2-54-173-61-131.compute-1.amazonaws.com/CS5774/CollabTunes/

User Info
Username: schadha   Password: test   Admin
Username: divit52   Password: test   Admin
Username: test   Password: test   Moderator
Username: kluther   Password: test   Regular User

SET UP (If running locally)
Assuming Xampp is installed correctly on your machine:
1) Unzip CollabTunes.zip to Xampp/htdocs folder
2) Inside, it contains a collabtunes.sql file
    a. Import this file into a new database called collabtunes
3) Edit the config.php file to set the correct SERVER_PATH, DB_PASS, and DB_DATBASE
4) If file uploads do not work, then:
    a. For Mac OS X
        i. Edit /Applications/XAMPP/xamppfiles/etc/php.ini
        ii. Set upload_max_filesize=50M
        iii. Set post_max_size=50M
        iv. Restart your apache server
    b. Also, make sure the uploads folder is chmod 777
5) Navigate to the localhost directory for this application in Chrome/Firefox and you should be good to go

Extension Set Up
1) In the \CollabTunes\extension\popup.js file, edit line 2 to provide the same localhost URL as the one you entered in config.php for SERVER_PATH.
2) Drag/drop the extension folder from \CollabTunes into your extensions page on Google Chrome