

Project created on 09.09.2016 00:08.

Report for project Senior Design


Task created on 10.02.2017 00:22.

Website Development

No due date

We will be coding the website in html, css, and javascript according to our finalized mockups.


Task tags: *No tags*

 Updated Manage Songboard code for teachers [*teacher_manage_songboard.html*]
Uploaded by Yanlin Ho on 31.03.2017 15:33.

 Comments for result Updated Manage Songboard code for teachers

Yanlin Ho on 31.03.2017 at 15:36: This is the updated code for the teacher side of the page that will allow them to save the music that they like from youtube onto our website and add it to different playlists/songboards. Tabs were created to compartmentalise and streamline the flow of logic according to their use. The first tab allows teachers to add music that they already have uploaded to the website to different playlists, and choose whether or not to show or hide that playlist to the students. The second tab allows teachers to upload their preferred songs from YouTube onto the website so that they can attach it to a playlist in the future.

Yanlin Ho on 31.03.2017 at 15:41: Completion of this page allows us to prepare to replicate the format for each of the other sections of the website on the teachers side, with modifications to allow the page to call from and update different classes on the database.

 Updated Server Code [*server.js*]
Uploaded by Carlie Abraham on 31.03.2017 05:02.

 Comments for result Updated Server Code

Carlie Abraham on 31.03.2017 at 05:12: I added a few more functions to the server code to allow more access to the database. First, I added a function that allows for updating data. This function takes in a collection, a selection (or what attribute needs to be updated) and the replacement for that attribute. This will be very useful when we need to update passwords or any other data stored in the database.

Carlie Abraham on 31.03.2017 at 05:12: Additionally, I added a function to remove a data element by its ID. This function takes in a collection and the id that corresponds to the data element that needs to be removed. This comes

in handy when users have to be removed from the database. The client will make a call to the database, for example, to the User collection with an ID that corresponds to the user to be deleted. I am not going to add any other function that removes data because I only want data to be removable with its unique identifier, the ID.


Carlie Abraham on 31.03.2017 at 05:13: Additionally, a function has been added to attempt a login from the client side. The function takes in a username and password. The User database is queried by the username, and the returned password (if there is a match) is compared to the inputted password. If the passwords match, then the server returns true. If it is not a match, the server returns false. More steps will be added to this procedure to take care of encryption of data.

 Login JS Code [*login.js*] Uploaded by Carlie Abraham on 31.03.2017 05:03.

 Comments for result Login JS Code


Carlie Abraham on 31.03.2017 at 05:20: This login javascript code processes the data entered into LoginTest.html. When the submit button is pressed, a function is called that retrieves the username and password info from the form. A call is made to the server to validate the username and password. If the username and password matched with a user in the database, the page redirects to another page on the website. If not, the user is alerted and an attempt counter is decremented. After 3 false logins, the login form is disabled, not allowing the user to continue to enter more false information.

 Login Test HTML Code [*LoginTest.html*]
Uploaded by Carlie Abraham on 31.03.2017 05:03.

 Comments for result Login Test HTML Code

Carlie Abraham on 31.03.2017 at 05:15: This HTML tests the login features that were added to the server code and provides a basic layout for a login page. A form is created for the user to input his/her username and password (the password is dotted out when the user types). When submitted, the information is passed to a javascript file which takes care of all of the processing of the data. This will be discussed further in the explanation of the code login.js which takes care of the processing.

Carlie Abraham on 31.03.2017 at 05:16: A few tests were performed to make sure the code works. First, the form was left blank and then submitted. This returned a false login, as expected. Second, a viable username was entered with an incorrect password, and again false was returned. Third, an unviable username was tested with a fake password, which resulted in a false login. And finally, a viable username and password was entered, and the server returned that login was successful.

 Display Song from Database, screenshot [*display_songs.png*]
Uploaded by Tong Yu on 30.03.2017 23:57.

 **Dispaly Song from Database** [*Tongtest.html*]

Uploaded by Tong Yu on 30.03.2017 23:42.

 **Comments for result Dispaly Song from Database**

Tong Yu on 30.03.2017 at 23:55: Using Carlie's SocketIO.on function, I added in scripts that displays the entire list of songs in the collection of songs in the database in a checklist format. In the code, I created two empty arrays, "songList" and "songID". The name and ObjectID are extracted from the result of the original function, and a loop was created where two new elements, checkbox and the label, would be made for each element in the array. The two were appended and positioned in an empty form in the body. I also made a clearbutton that erases the innerHTML of that empty form each time. The idea was to do the same for the Songboards as well, and then copy and paste this script into the Teacher sites. The teacher would be able to check off the songs, and then the checked off songs would be collected in an array, and then the selected songboard would be updated with these new song included.

 **Domain Name Registration** Created by Yanlin Ho on 31.03.2017 15:29.

Our domain name was registered with Amazon Web Services for \$12. Registering with Amazon allows us to use their Route 53 service, which directs traffic on the registered domain name webpage to the EC2 server that we have registered on AWS as well. This allow us moving forward to configure a public IP address to the website so that we can interact with content using copyrights, such as embedding music onto the website.