

Author: Brandon Wu
Student ID: 101260155
Project Name (Openstack): Brandon Final
Youtube Link: <https://youtu.be/aULg25ssyss>

List of Files

server.js: Sets up a web server using Express, connects to a MongoDB database using the MongoDB Node.js driver, and defines various routes and middleware for handling HTTP requests. It includes functionality for user authentication, posting and viewing artwork, managing user profiles, workshops, comments, likes, and notifications.

database-initializer.js: Connects to a MongoDB database using the MongoDB Node.js driver and drops existing collections ("artCollection," "artistCollection," "userCollection," "commentCollection," "likesCollection," "workshopCollection," and "followCollection").

addArt.js: It retrieves input values from HTML elements representing the title, year, category, medium, description, and poster of the artwork. Input values are then cleared and the file sends a PUT request to check if the given title exists already. If the server gives it the heads up, it will send a POST request to the gallery.

addWorkShop.js: Similar to addArt, but it only sends a POST request to add a new workshop to /workshop. If statements to alert the outcome.

enroll.js: Extracts the current URL and parses it to obtain the last segment. Sends a POST request to the "/enroll" endpoint, including the constructed data object in the request body. It displays an alert message based on the outcome of the request. If the response is successful (HTTP status 200), it shows a success alert; otherwise, it displays an error alert.

follow.js: Same functionality as enroll.js

galleryData.js: comment() retrieves comments from the HTML element with the ID comments. Clears input field after retrieving value, extracts current URL, constructs an object and sends a POST request to the server. likes() does a similar thing to comments

login.js: Retrieves the user's username and password from HTML elements and clears them. Creates an object loginInfo that contains the information. Sends a PUT request to /login and awaits response.

newAccounr.js: Same as login.js

profile.js: removeLike() constructs an object containing title of artwork in order to remove the like. A DELETE request is sent to the server and awaits response. If successful the program redirects user to the current page with window.location.href. unfollow() and removeComment()

does the same thing as `removeLike()`. `switchAccount()` also creates an empty object, but sends a POST request to `/switchAccount()`.

`search.js`: Retrieves user input for medium and category. Clears input fields and creates an empty object. If user provides a value for medium or category, it adds the respective property to the object. Sends a PUT request to `/search` and awaits response.

`styles.css`: CSS file that creates styling to the website

pug files: templates for different html

How to activate the website on openstack

1. Go to `http://134.117.134.43:3000/`

Instructions on how to activate the website on localhost:3000

1. Locate file in the folder
2. type `npm install express` in the terminal
3. type `npm install pug` in the terminal
4. type `npm install mongodb` in the terminal
5. type `npm install express-session` in the terminal
6. After everything is installed enter `node database-initializer.js`
7. Go to `http://localhost:3000`
8. Main page will welcome the user
9. Gallery shows a list of artists and their artworks
10. Login allows you to log in if you haven't already
11. New account allows people to make a new account
12. Artists shows all artists
13. Add Art lets people create an artwork of their own
14. Add workshop lets people create a workshop
15. Workshops shows all existing workshops
16. Search lets people find artworks through medium and category
17. Profile allows people to see what kind of account they have, option to switch account, all likes, reviews and accounts followed. As well as a log out option

Stylistic Design Decisions:

Div is use to create distinct parts of the website

Turquoise color is used to make the website colorful

Error that can crash my code:

Program runs perfectly fine without crashing, no bugs will crash the server.

