

OPENGALLERY REPORT

Name: Ethan Abraham

Student ID: 101324864

INCLUDED FILES

- package.json: json with all the dependencies for the server
- server.js: node js server

DATA

- data-initializer.js : initializer for the database
- gallery.json: json file storing the initial data for database

PUBLIC

SCRIPTS

- account.js: JavaScript file for account page
- artwork.js: JavaScript file for artworks partial
- gallery.js: Javascript file for gallery page
- header.js: JavaScript file for header partial
- login.js: Javascript file for login page
- new-art.js: Javascript file for new-art page
- register.js: Javascript file for register page
- reviews.js: Javascript file for reviews partial

STYLES

- account.css: CSS for account page
- artwork.css: CSS for artworks partial
- footer.css: CSS for footer partial
- gallery.css: CSS for gallery page
- header.css: CSS for header partial
- login.css: CSS for login page
- new-art.css: CSS for new-art page
- register.css: CSS for register page
- reviews.css: CSS for reviews partial
- search.css: CSS for search page

ROUTERS

- account-router.js: router for /account

- artwork-router.js: router for /artworks
- search-router.js: router for /search
- workshop-router.js: router for /workshop

SCHEMAS

- Art-Model.js: schema for artworks collection
- User-Model.js: schema for user collection
- Workshop-Model.js: schema for workshop collection

VIEWS

PAGES

- account.pug: account page
- error.pug: error page
- gallery.pug: view artwork page
- home.pug: home page
- login.pug: login page
- new-art.pug: add new art page
- register.pug: register account page
- search.pug: search results page

PARTIALS

- artworks.pug: partial for all artworks
- footer.pug: partial for footer
- header.pug: partial for header
- reviews.pug: partial for reviews

INSTALLATION AND RUNNING THE SERVER

1. Open a terminal in the openGallery directory.
2. Type `npm install` into the command line.
3. To initialize the database if not done before, type `node ./data/data-initializer.js` in the same terminal
4. To start the server, in the same terminal, type `node server.js`
5. Hold CTRL and select the link or, type the link into a browser to view the site.

DESIGN

For this application. The Design is very heavily focused on the server completing all the procession and check while the client just reads the data given from the server. In other words, the only data that is consistently stored is the user's role, ID and username in their cookie. This way all processing can be done by the server and is not limited by the hardware on the client side.

There is a few critiques on this design though. Firstly, since there is not a lot of data being sent to the client from the server, user experience is now dependent on the internet connection/bandwidth of the client. Furthermore, with the server storing all the data for the pages to load, there can be pages with large JSONs being

sent to them. Thus causing clutter to form with sending a Json on the server side and from all the searches done in the database to build the Json.

EXTRA FUNTIONALITY

- Added a homepage where users can like and go to review artwork from all artists instead of only on the artist page
- User is able to view how many users follow and following the account currently being viewed and a list of who those users are as well with hyperlinks to those accounts

KNOWN ERRORS

- When embedding large images, the site struggles to load them quickly. Every other element loads before the photos
 - o The limit seems to be a photo width of 720 pixel max for large jpeg 100 files.

YOUTUBE LINK

https://youtu.be/OkeuM_phwKM