Ava Bard - 101262714
Project Name - AB Art Gallery
Youtube link: https://youtu.be/dZI2RAy81D4

List of files and their purpose:

gallery: folder to store json file of initial data

- gallery.json: json file to store all initial gallery data

models: folder to store schemas

- gallerySchema.js: schema for the gallery (artwork)
- notificationSchema.js: schema for the notifications
- reviewSchema.js: schema for the reviews
- workshopSchema.js: schema for the workshops

public: stores styles and client side of code

- client.js: js file that stores all of the client-side code responsible for sending data to the server. takes information from user and makes the requests from the server-side of the code
- styles.css: for styling the webpages

views: stores the html (pug) information for the webpage

- addArt.pug: page for user to add art
- addWorkshop.pug: page for user to add a workshop
- allArt.pug: page that displays all artwork
- artistPage.pug: page that displays all artist information
- followers.pug: page that displays users followers
- following.pug: page that displays users following
- header.pug: the header for all logged-in pages, allows easy navigation
- individualArt.pug: page that displays individual art page
- liked.pug: page that displays all artworks that user liked
- login.pug: page that asks/allows user to log in
- notifications.pug: page that displays user's notifications
- profile.pug: page that displays user's profile
- registerUser.pug: page that allows user to register for an account
- reviewed.pug: page that displays all user's reviews
- search.pug: page that allows user to search for art
- unauthorized.pug: page that displays when a user is unauthorized (eg. followers list before logging in)
- workshop.pug: page that displays information about a posted workshop

server.js: js file that takes requests from the client side, gets information from that database, or handles the information given from server side, and returns it to the client side using express

gallery-init.js: js file that reads through the gallery.json file and stores the data in the mongoDB database "finalProject" and initializes the collections for gallery, users, workshops, notifications and reviews

Installing, Initializing, and Running the database and server:

1. Extract the folder and in command prompt, go to the directory of the folder
2. Write "npm init" and "npm install express express-session mongoose morgan"
3. Then, type in the command "node gallery-init.js", this will initialize the database
4. After the gallery has been initialized (the command prompt will say if it has), run the command "node server.js"
5. Click the link or type "localhost:3000" into your browser, and you can now run the code!

Discussion and critique of overall design.

1. Inspiration and Design Choices:
    - In my art gallery, I took some inspiration from different social media platforms, mainly Pinterest and VSCO. I chose a simple brown theme because I find white to be overused and a little boring and unimaginative. I think brown is a nice theme that is not too harsh on the eyes, and a nice colour palette overall. When discussing user experience and navigation, I ensured that the user could easily search up art pieces and access the more important information using a simple header that acted as a navigation system.
2. Navigation and User Experience:
    - While creating this website, I conducted a multitude of tests, which included going through all test cases I could think of, such as adding random urls to the end of the localhost:3000 (eg. localhost:3000/abc), and ensured they return what is expected. In order to make sure it was usable, I had some of my friends who are not in this class test out the website and encouraged them to find a way to “break” or “crash” the site. After they found ways around a few parts of the website, I reached a point in which I feel my site functions well enough.
    - By making my friends test out the site, I ensured that my website is user-friendly and functions properly. Some design choices I made were to keep the website simple but still aesthetically pleasing, to make sure users will 1. Be able to understand and use the website and 2. Want to use it/think it looks nice
    - One feature I believe contributes to a user-friendly experience is the header on the website. It is simple and not confusing, so that users know how to find everything on the site.
3. Coding Approach:
    - I chose to keep all of my information on the database using schemas. I think it is an excellent and easy way to store data that make the coding itself much simpler and better coding practice. I had 5 schemas and the objects for each schema were stored in different collections, sometimes referencing eachother. For example, in my gallery schema, I referenced a user as the artist, an array of users for the followers and following, an array of workshops for the posted workshops, and a few more.
4. Visual Consistency:
    - I maintained visual consistency by sticking to a colour theme. All of the elements of my website were some shade of brown to maintain a nice colour scheme. I used styles and containers to keep the consistency to the website and had all of the buttons the same colour so they would stand out
5. Accessibility:
    - I added alternate text for the pictures, and made the text colour contrast from the background colour to improve the readability
6. Security Measures:
    - I used authentication and a database to ensure users could not access the website unless they have an account and are logged in, using a database of users that already have an account
    - I put authentication on all of the web pages to ensure people who do not have an account cannot access anything
7. Usability Improvements:
    - I feel that I could add more to the reviews page to make it look nicer to the user, as well as adding some pagination to the browse art page to make it easier to find artwork
8. Overall Reflection:
    - Overall, I feel that this project highlighted all we learned this semester really well. I feel that this project, although it took a lot of time, was a really rewarding one. It taught me how to user databases and how to properly style websites. A lot more thought was required for this assignment, so I had to put more work into planning this assignment, and actually wrote out what I was going to do, which is not something I typically do. This taught me how to plan out assignments better, and I now know how to work better.

I did not implement any of the suggested bonuses, and have no known ways to break my code