Andy Li
101264096
Andy’s Open Gallery
COMP2406
Youtube Video: https://youtu.be/itqX6Ng4ABw

2.

- database-initializer.js : initializes the database, includes data.
- UserModel and ArtModel are mongo Schemas
- styles.css & browse.css : styling for login and register and browse pages respectively.
- Pug files for client-side interactivity. Some partials.
- client.js - Handles all front-end Javascript.
- server.js - Handles all back-end javascript, including routing.
- gallery.json - supplies default art
- package-lock.json and package.json - give details on installed modules

3.

- In the same directory as server.js, enter “npm init”
- Enter “npm i mongodb”, “npm i mongoose”, “npm i express”, “npm i express-session”,
- Initialize the database using “node database-initializer.js”
- Run the server using “node ./server.js”
- Navigate to localhost:3000 on a web browser.

4. Overall Design and Implementation Quality

1. RESTful API implementation.
    - Adheres to RESTful guidelines. Program is stateless and effectively utilizes meaningful URLs and employs uniform interface.

2. Authentication
    - Authentication is performed selectively depending on routes and functionality. Users not signed in cannot view art pages or profiles, but can browse the general feed.

3. HTTP Status Codes:
    - Unless provided with an invalid URL, status code 200 is often used for successful operations. Pre-emptive checks are used to reroute unauthorized users to the login page rather than generating a 404 error.

4. Schemas:
    - Schemas were designed with little information, employing arrays of strings and condensed versions of some properties for optimization.

5. Modular design:

    - Use of middleware increases modularity and separates code neatly so that each piece of middleware handles one task specifically. Also helps with code reuse and scaleability.

6. Use of Mongoose:
    a. Simplifying interactions with Mongodb through the use of mongoose.
    b. Applied asynchronous functions and avoided buffer issues using the await keyword.

7. Improvement: file management and routing through multiple files. Error handling could be more specific and frequent. As the code currently is, the assumption is that everything exists as it should.

5. Design Decisions and Functionality

- Verification implemented by storing objectIds when following, liked, and comments in multiple locations, such as the patron’s account and the art piece itself. That way they can cross-reference one another and if one is not found it can be handled appropriately.

6.

- No crashes, I think.
- Notifications not implemented.
- Workshops not implemented.