

## # Project Report

### ## Personal Information

- **Name**: Soham Gherwada
- **Student ID**: 101257974
- **Course Number**: COMP 2406B
- **YouTube Video Link**: <https://youtu.be/R38OgJHN5o8>

### ## List of Files

1. **File 1**: data
  1. gallery.json
2. **File 2**: views
  1. **File1**: userpages
    1. **File1**:img
      1. artgallery.jpg
    2. addApost.ejs
    3. artist.css
    4. artist.ejs
    5. artistUser.css
    6. artistUserpage.ejs
    7. artwork.css
    8. artwork.ejs
    9. home.css
    - 10.homepage.ejs
    - 11.login.ejs
    - 12.logout.ejs
    - 13.notification.ejs
    - 14.search.ejs
    - 15.style.css
    - 16.user.css
    - 17.user.ejs
    - 18.workshop.ejs
    - 19.WorkshopCreation.ejs
3. database-initializer.js
4. package.json
5. server.js
6. README

## ## Installation and Initialization Steps

1. Step 1: npm install
2. Step 2: node database-initializer.js
3. Step 3: node server.js

## ## Overall Design and Implementation Quality

- **Design Discussion**: The server is built to handle all the request in a elegant way. Each request made to the server only sends the required data needing for the functionality of the page to work each post/put request are equipped with only the data that they need to post. The server also has a variety of of post put get request that make the program have good readability and easy to navigate.
- **Critique**: The website could have had a better visuals for the for the post page and the workshop page.
- **Scalability**: There are several features that I could add like texting or video calling or a calendar for the workshops. Also the artist could reply to the comments. This would have been a great feature.
- **Extra Functionality 1**: The user when they go to check out a singular artwork they are able to use the cursor as a magnifying glass across the image to check out the details of the image.
  - **Extra Functionality 2**: The Artist is able to view all the users that have joined the workshop.
  - **Extra Functionality 3**: The Artist is able to view the post live as a preview before pressing the submit button.

## Server.js

### **Login and Registration:**

- /login: Handles user login. Verifies credentials and sets session data.
- /register: Manages user registration. Checks if the user exists, creates a new user, and sets session data.

### **Static Files Serving:**

- /home.css, /login/user.css, /artistprofile/artistUser.css, etc.: Serve CSS files for specific pages.

- /img/artgallery.jpg: Serves an image file.

### **Page Rendering and Data Fetching:**

- /gallery, /login/:user, /artist/:artist, etc.: Render pages and fetch necessary data from the database to display on these pages.

### **Post-related Actions:**

- /send: Handles the submission of new posts/artworks.
- /like, /unlike: Allows users to like/unlike posts.
- /comment/:artwork: Manages comments on artworks.
- /deleteComment/:artwork: Handles the deletion of comments on artworks.

### **User Actions:**

- /logout: Logs the user out and clears session data.
- /follow, /unfollow: Allows users to follow/unfollow artists.
- /switch: Allows users to switch between being an artist and a patron.

### **Search Functionality:**

- /search: Allows users to search for posts based on various parameters.

### **Workshop-related Actions:**

- /workshop/:artist, /WorkshopCreation/:artist, etc.: Manages workshops, their creation, joining, and notification sending to users.

### **Notification Handling:**

- /notification: Renders a page displaying user notifications.

### **Client-side Script and Styling:**

- /client.js, /style.css: Serve client-side JavaScript and CSS files.

### **Server Setup:**

- Starts the server on port 3000 and listens for incoming requests.