

Name: Victor Glasser
Student ID: 101146336
Course Code: COMP2406
Youtube Link: <https://youtu.be/wawkoYjFvLo>

List of files and purpose:

- Server.js - This is the main server for program with the principle task of finding resources that other files request
- gallery.JSON - This .JSON file has an object which contains all art that is initially on the website
- userList.JSON - This .JSON file has an object which contains all of the initial users on the website
- Header.pug - This is the template for the navigation bar at the top of the screen once the user has logged in
- homePage.pug - This is the template for the page that the user initially sees before they login
- homePage.js - This provides functionality for homePage.pug
- newAccountPage.pug - This is the template for the page that is used to create a new account
- newAccountPage.js - This provides functionality for newAccountPage.pug, and adding a new account to the server
- ArtPiece.pug - This template produces an image of a piece of art, as well as information on the image, such as the artist, the number of people who liked it, and the reviews that were made for that piece
- artPiece.js - Provides functionality for ArtPiece.pug, allowing the user to add a like or review for a piece of art
- ArtBrowse.pug - This template displays all of the art pieces that the server has, with links to their artists' pages
- ArtBrowseMedium.pug - This template displays all of the art pieces that share a specified medium
- ArtBrowseCategory.pug - This template displays all of the art pieces that share a specified category

- AccountWorkshopsPageView.pug - This template displays a list of the workshops that an artists is offering
- accountWorkshopsPageView.js - Provides functionality for AccountWorkshopsPageView.pug, allowing for the user to become a workshop participant
- AccountWorkshopsPage.pug - This template displays a list of the workshops that the user is offering
- AccountReviewsPage.pug - This template displays a list of reviews written by the user
- accountReviewsPage.js - Provides functionality for AccountReviewsPage.pug, allowing the user to delete reviews that they wrote
- AccountPageVisit.pug - This template displays a profile page for artists that the user is viewing
- accountPageVisit.js - Provides functionality for AccountPageVisit.pug, allowing the user to follow or unfollow the artist
- AccountPage.pug - This template displays the user's profile page
- accountPage.js - Provides functionality for AccountPage.pug, allowing the user to toggle between a patron and artist and add art or a workshop if they are an artist
- AccountLikedArtPage.pug - Displays a list of all of the art that the user liked
- accountLikedArtPage.js - Provides functionality for removing likes
- AccountFollowingPage.pug - Displays a list of all of the artists that the user is following
- AccountFollowersPage.pug - Displays a list of all of the artists that are following the user
- AccountArtPage.pug - Displays a list of all of the art that the user has uploaded

Initialization:

Start by unzipping the "VictorGlasser-project.zip" file. In the console, type "npm install", to install all modules required to run this program. Write "node server.js" into the console to run the program

Overall Design:

I did not add any additional features for bonus points. In terms of design principles, RESTful design was achieved to an extent. Error handling took place for retrieving every template, however further error handling could be added for other tasks, such as those found in the POST and DELETE requests. Data transfer was reasonable in size, however it could be minimized even more with additional time and reflection. The code itself is quite organized, with occasional comments to assist the reader in understanding what is occurring. There are some shortcomings, such as the following requirements not being satisfied: users not getting

notifications if a followed artist uploads a new piece, the patron-status user not requiring any uploads to become an artist, and some pages requiring a manual refresh to update after finishing their tasks. There is also a lack of styling, which would make the website much more appealing. MongoDB also was not used, which was expected, so by adding that, the web application would better achieve the project goals.

Known Errors:

Invalid URLs cause crashes and some pages do not refresh and show updates immediately