Project Report:
Name: Ryushen Tan
Student Id: 101269064

Project name: Term Project

Course Number: COMP 2406

youtube link: https://youtu.be/p9D4CefbGIM

List of files:
artwork.css | css file for artwork page
general.css | css file for most pages
home.css | css file for home page

gallery.json | Provided json file with image data

heart.png | image of a heart that was free from the internet
account.js | js file handling account pug page button requests
accounts.js | js file handling accounts page functionality and sending requests to the server side
artwork.js | js file handling artwork display functionality and sending requests to the server
register.js | js file handling registering functionality and sending requests to the server
search.js | js file responsible for search result functionality and displaying the html
account.pug | pug page responsible for rendering account page
accounts.pug | pug page reponsible for rendering artist pages
addart.pug | pug page responsible for using adding art
artwork.pug | responsible for individual artwork pages
artworks.pug | responsible for a list of artwork pages
index.pug | home entry page
register.pug | front end page responsible for user registration
search.pug | front end page responsible for rendering use search functionality
workshop.pug | front end page responsible for user to add workshops
database-initializer.js | initializes database
server.js | server side code with routing
Instructions: do npm install, npm install mongoDB, npm install express, npm install express-session, make sure u have the essentials, than do node database-initializer.js then u can just do node server.js

Overall Design: the implementation quality is decent i use the proper routes, post if im adding anything put if im updating any data, gets if im getting data, delete if im deleting anything. I implimented proper json response codes for ex. 404 or an response 201, I use fetch so a lot of my functions are async, and I use await for proper database usage. Some possible improvements would be more consistent requests cause I ended up learning half way through, or less pages cause I had a lot of extra space in the end. The design provided a very simple and visually appealing website, where there is colours that are soft on your eyes, and make the text visible, as well as interactive features like with the buttons, or a(href)'s where there are hover css effects applied.

Decisions: a notable decision would be not to include the review and likes for the artist artworks on the artist page cause I believe it would make it too crowded, so instead I leave it for when the user taps on the artwork, it also leaves out the problem with users having an unconscious bias towards art because of previous opinions. I also have two artworks pages, one where users can ctrl+f the whole thing and another with pagination for simple browsing.