

**Youtube Video:**

<https://youtu.be/YpHuwL1ylOg>

**List of files and their purpose:**

database-initializer.js

- To init the database

package.json

- Use for npm install, holds the dependencies

server.js

- All the server code to handle requests

gallery.json

- To hold all the data files for the artworks

artist.js

- Client js file for artist page

artwork.js

- Client js file for artwork page

artworks.js

- Client js file for artworks page

createAccount.js

- Client js file to create account

createArtwork.js

- Client js file to create artwork

profile.js

- Client js file for profile page

artist.pug

- Template file for artist page

artistProfile.pug

- Template file for artist profile page

artwork.pug

- Template file for artwork page

artworks.pug

- Template file for artworks page

createArtwork.pug

- Template file for creating artworks

createWorkshop.pug

- Template file for creating workshops

index.pug

- Template file for front page

login.pug

- Template file for login page

patronProfile.pug

- Template file for patron profile page

Workshops.pug

- Template file for workshops page

header.pug

- Template file for header links

### **how to install, initialize, and run your database and server:**

Installs correct dependencies:

**npm install**

Initiates database:

**node database-initializer.js**

Runs server:

**node server.js**

### **Discussion and critique of your overall design:**

I like how simplistic my design is and in my opinion the code is broken down and more readable. Because of this the code is not as optimal and will take more resources for storage and computing with communication with the database. For a project of this size it shouldn't matter but for more users I'd rewrite my mongo commands/ switch to mongoose to fix latency. On the other hand scaling might be more easy as since the documents are separate dispersing the documents to many servers might be easier. Maintenance would be an area of improvement as adding routes with mongodb is hellish so mongoose is a must. Security is a huge fix as passwords are in plain text. Lastly, adding mobile integration would be cool.

### **Explanation of any design decisions and extra functionality**

It's really hard to pinpoint some design decisions on paper as this project was really open ended for interpretation. Viewing the video should maybe show the choices with how I chose to implement the features. For the structure of the code I wanted to make it more readable and to break down tougher steps into smaller functions. Even the database is broken down pretty nicely.

### **List of any known errors or control sequences that work not as expected or crash your code**

I don't know of any errors of such in my program and I hope there isn't any