Guide to Code

## Root

**Server.js** –

The “server.js” file at the top level of the develop folder is where everything is brought together and served. The connection to the server, connection to the MongoDB, Express Routes, and Passport are all put to use here. At the top is where the NPM packages are “required”. After that, the port is setup and models are required for syncing. The next bit of code is middleware to allow certain things to happen, such as serving static files from the public folder. Then passport is initialized and configured to keep track of user’s session, storing their login status. Next, the express routes are required so that we can send requests on the server. Finally, the connection to the database is create and after that runs, we start listening on port 3000 where we serve our files.

**Package.json** –

To keep it short, the “package.json” file is where you configure information about your projects, the dependencies of your app, and custom scripts to run your code.

## Routes

The routes folder is where you declare routes for sending front-end files and make requests to do things like add to a database.

**Api-routes.js** –

In this file, the routes are set up for signing up, logging in, and getting data about the user to be used on the client side. Passport is used to create a hashed password and authenticate the user throughout their visit. All of these routes interact with the database.

**HTML-routes.js** –

This file checks if the user is authenticated and either servers them the signup/login page if they’re signed out or sends them to the members page if they are signed in.

## Public

This folder is where all the front-end code of the app is stored.

**JS folder** –

This folder has 3 files, login.js, members.js, and signup.js. Login.js manages the functionality of the form and sends a post request to the server to log the user in. Signup.js does the same but for signing up. The members.js file simply displays the user’s info on the members page.

**Stylesheets folder** –

This folder is self-explanatory. It contains all of the CSS for the project.

**Login.html, members.html, signup.html** –

These files represent each page of the project. They hold the elements such as text, forms and buttons. These files are served statically from the server.js file with express static.

## Node Modules

This folder is where all of the dependent node packages are downloaded and stored. This file becomes big so it is in the git ignore of the application. When pulling this project from the GitHub simply run “npm i” to install all of the needed dependencies.

## Models

This folder is where the structure of our database is set up.

**Index.js** –

In this file, all of the needed dependencies are brought in at the top for working with the database and checking the environment variables. This is where the URI for the database is and it imports all of the models to be used by the server.js file.

**User.js** –

This file is where the structure of the User collection is set up. At the top of the file the fields and datatypes are defined for what we’re going to be passing in the database, as well as options such as allowNull which makes a field required. Also in this file, using bcrypt the password of the user is hashed so that the users password is not visible in the database.

## Config

Finally, we have the config folder. This folder is for setting up configurations for different node packages and our project. The config.json file sets up different key value pairs based on if the we’re working in development, test, or production mode. Passport.js is a file that configures our strategy for Passport. This strategy sets up how we want to use Passport. It is then exported to be used in our server.js file. Finally we have isAuthenticated.js, this file is simply for restricting routes if a user is not logged in.