*Application*

This application is built so the user can securely log in and out of a system. Data is stored within the system.

*User Story*

As a user I want to be able to securely log in so that I can store my information and then log out so no one else can access it

*Using the app*

* Before running the application make sure to run “npm install” to install all required node modules.
* Check the port used in server.js is not being currently used on your local system e.g. if another app is running on PORT8080 make sure to alter the code to use a different port.

*Code elements*

* JavaScript
* jQuery
* CSS
* HTML
* Npm Express
* Npm express-session
* Npm fs
* Npm path
* Npm sequelize
* Npm bycryptjs

*Code walkthrough*

**Develop folder**

* This folder is used to allow for quick navigation when accessing the application in the terminal

**Config**

* **Middleware:**
  + Contains: ‘isAuthenticated.js’
    - This file restricts the user from accessing certain aspects of the application if the user is not logged in
    - If the user is logged in it allows the application to proceed to the restricted routes
* ‘config.json’
  + Is a configuration file used to create a connection
* ‘passport.js’
  + This file tells the system that we want to login using a username and a password
  + If the correct details are used the login details are returned
  + If incorrect details are used an error message appears

**Models**

* ‘Index.js’
  + This file contains variables containing the npm install variables and variables requiring links to other files
  + It uses module.exports to export from the db folder
* ‘user.js’
  + Requires bycrypt for password hashing
    - This is used to secure database
  + Dictates the email and password cannot be null values
  + Checks that what is entered by the user as a password is within the password database

**Public**

* **JS**
  + ‘login.js’
    - Contains front end JavaScript for the login process.
    - When someone enters their login information this file checks to see if an email and password have been entered
    - If appropriate information is entered then the members page is opened
  + ‘Members.js
    - This file contains a GET request to see who has logged in
    - It also updates the pages HTML
  + ‘singup.js’
    - Contains front end JavaScript for the signup process.
    - Allows a new user to signup
    - If user successfully signs up it redirects them to the members page
* **Stylesheet** 
  + ‘Style.css’
    - Contains front end CSS for all classes and id’s within front end
* ‘Login.html’
  + Contains front end html for the login page
* ‘members.html’
  + Contains front end html for the members page
* Signup.html
  + Contains front end html for the signup page

**Routes**

* ‘api-routes.js’
  + Exports information throughout the app
  + Handles sending users to the members page if credentials are valid
  + Contains the routes for api functions these include:
    - Sign up
    - Logging out
    - Getting data about user to be used client side
* ‘html-routes.js’
  + Exports information throughout the app
  + Sends user to the correct html page based on whether they’re a member, signing up or logging in

‘Server.js’

* Creates the port that the app will run on
* Configures middleware