Open my sql and create a database with the same name as the one listed in one of the files. In connection.js, in the configuration folder, there is a config.json file with 3 nested objects inside. One is called development which is for the development phase of the project. In there you put your mySQL password and database name in the placeholder. Then you npm i so the package.json file can be read and the libraries are downloaded. Then run npm i and the server will start. Open a browser and type localhost: followed by the correct port number. The html.routes file will route you to the signup.html file. From here you enter your email and password. When you hit enter, an ajax call to the signup.js will make a post request. This gets sent to the server and the authentication gets sent back to the html routes, through members.js and the members.html page loads. You can then click signout. If you sign-out, the html routes will send you back to the sign-in page. You can click the log-in button and the html-routes file will send you to the login page. If you enter your log-in credentials the login.js will send a get request to the server via the api-routes file which then goes back through the middleware file for authentication via the index.js file in the models folder. If the credentials are authentic that file will send that message back to the html.routes file which will send the user to the members.html page. There are models for the data to pass through so it is readable. SQL is stored as strings, and javascript data is an object with lots of data types. This app uses several libraries. These are: express for the server and routing, express-session for session storage of user credentials, bcryptjs to encrypt password information during the authentication, sequelize, an orm to change the data from the database into a format which is understandable for the javascript code, mysql which is a mysql driver for node, passport.js for authentication of user credentials, via passport-local, which is a configuration strategy library.

User enters the app at signup. The app has a form and 2 buttons. One for sign-up, one for log in. If you log in you go to members.html. From there you can log out and you go back to sign-in.html If you sign-up, you enter your credentials and you go to members.html.