Reverse Engineering Code

- **Server.js -** Creates an express server and creates express app with middleware. Sets app routes. Syncs with database with sequelize and logs connected on PORT.
- **Passport.js -** Sets up passport validation and connects db. Checks if email exists and password validation.
- **Config.json: config -** has local host in development and this is where you set up your production configs.
- **Middleware: isAuthenticated.js -** Checks to see if the user is logged in, else redirects to the login page.
- **Models: index.js -** Sets up sequelize to either use environment vars or local configs. Has associations to associate db.
- **Models: user.js -** Sets up bcrypt, defines user for sequelize, Creates method for user password to check if password can be validated to stored password for user. Before user is created it will hash password.
- **Public: JS: login.js -** sets up all functions for login form.
- **Public: JS: members.js -** get api route to return member name/email.
- **Public: JS: signup.js -** sets up functions for form submit to take in user input and post user to db and handle errs.
- **Routes: api-routes.js -** post functions to set user data to the db, gets for logout route and getting user data.
- **Routes: html-routes.js -** gets for html routes for signup, members, and login.