**REVERSE ENGINEERING**

**Purpose:** This is a tutorial for Node JS application using Sequelize and Passport and showing a thorough walkthrough to understand each codebase.

How to Install

First clone or download the repository into your computer. Create SQL Scripts for databases as named in the config.json file and open and run the script in MySQL.

Edit the config.js file and add your password for your local MySQL password.

If gitignore is added to file with "node modules" inside the folder, node modules folder will not show up. Open preferred terminal and type “NPM install” to install the required node modules for this project.

If there are any error showing in terminal, run NPM audit fix if any issues.

Type "Node Sever.js" in your terminal and to test your connection. Once your connection is connected, open the local server link in your web browser.

Step by Step walkthrough of the codebase

What is the Middleware? The middleware file restricts the routes so that user cannot reach another user’s page unless they have login access.

**config.json File**

This file is a middleware connection configuration that help connects to the server.

**passport.js**

This is a Javascript that tells passport that we want to log in using a specific email address and password.

**index.js**

This Javascript allows you to connects to the database and imports each user’s data

**user.js**

This Javascript requires "bcrypt" module for password hashing. This allows our database to be more secure even if compromised. This Javascript describes what is use in our database.

**api-routes.js**

This gives us the route for signing up a user or signing in and logging out. This allows the user to give specific data to be displayed client on the browser side.

**html-routes.js**

Routes that check whether the user is signed in, whether user already has account etc and sends them to the correct html page.

**.gitignore**

This hides unwanted items or password if you have one so that it does not show up when uploaded into github repo.

**package.json**

This explains all the packages info, node modules used and version installed

**server.js**

Requires all the necessary npm packages and sets up PORT and connects to express and middleware. This Javascript will pull all routes and syncs database to successfully connection to server.