# Unit-14-Sequelize-Homework-Reverse-Engineering-Code

homework week 14

------------------------------------------------------------------------------------------------------------------------------

--PASSWORD AUTHENTICATION--

This app allows users to create an account, log into the account and sign back out securely. All user data is stored in a mysql

database.

------------------------------------------------------------------------------------------------------------------------------

--USER STORY--

as someone who wants to safely log in to "X", I want to know my personal details are safely stored so that I don't have to worry

about using "X".

------------------------------------------------------------------------------------------------------------------------------

--TECH USED --

-BCRYPTJS

-EXPRESS

-EXPRESS-SESSION

-MYSQL2

-PASSPORT

-PASSPORT-LOCAL

-SEQUELIZE

------------------------------------------------------------------------------------------------------------------------------

--GETTING STARTED--

to begin using this app, please clone this repository into your local storage. Once this is complete, please follow these steps;

1)create a mysql db called "passport\_demo"

2)go into the config file, open config.js and insert your personal data ie username, password etc

3)open terminal in current repo and run "npm i" to install all node packages

4)while in terminal, run "node server.js" and you will successfully connect to server

5)open browser and put "http://localhost:8080" in search bar

6)enjoy using the app!

------------------------------------------------------------------------------------------------------------------------------

--FILES EXPLAINED--

CONFIG

MIDDLEWARE

isAuthenticated.js {

restricts routes that a user is not allowed to visit if not logged in. if user is logged in, it continues with request };

config.json {

connection configuration to connect to server };

passport.js {

contains javascript logic that tells passport we want to log in with an email address and password };

MODELS

index.js {

connects to database and imports users log in data };

user.js {

requires "bcrypt" for password hashing. this makes our database secure even if compromised. Here we have JS that defines what is stored on our database };

ROUTES

api-routes.js {

contains routes for signing in, logging out and getting users specific data to be displayed client side };

html-routes.js {

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

package.json {

contains all package info, node modules used, version info etc };

server.js {

requires packages, sets up PORT, creates express and middleware, creates routes and syncs database / logs message in terminal on successful connection to server };

------------------------------------------------------------------------------------------------------------------------------