Project 14 - FitnessTrac-kr Back-End

INDIVIDUAL REPORT FOR

Student Name:

	Total Score	Total Possible	Common - 30% Game - 70% **Final Grade**
Total Grade	0	90	**0.00%**

Overall Comments

Generally excellent work! There are some technical problems remaining, which I was surprised to see. Comments on each of these remaining issues has been sent to you as a group DM in Slack. With the remaining endpoints fixed, this API could reasonably be used for the front end project.

Common Requirements (30% of total)

Total Score	Total Possible	Grade For Common Requirements (30%)
0	58	0.00%

Grade

Criteria	Score	Possible	Comments
As an instructor I want to see you demonstrate mastery (when appropriate) of:			
NodeJS Concepts			
the require & module.exports module system as organization	0	2	
usage of process.env when necessary	0	2	
Asynchronous coding			
try / catch blocks	0	2	
appropriate use of async and await	0	2	
correctly returning data from an async function	0	2	
correctly throwing and catching errors from an async function	0	2	
Database concepts (SQL Focused)			
table creation	0	2	
inserting data	0	2	
removing data	0	2	
updating data	0	2	
querying single tables	0	2	
querying joined tables	0	2	
Database adapter concepts (functions which interact with th			
correct translation of passed data (to function) to form database queries	0	2	
correct return from function of data types	0	2	
Express concepts			
using middleware correctly	0	2	
setting up routes correctly	0	2	
building the server from a collection of routes	0	2	
using modules like bodyParser correctly	0	2	
incorporating JSON Web Tokens to authenticate users when necessary	0	2	
setting up an error handling middleware to alert API users of potential errors	0	2	
Routing concepts			
using the correct verbs on routes (e.g. GET vs PU	0	2	
using sub-routes for collections of data (e.g. /user	0	2	
Deployment			
correctly setting environment variables	0	2	
correctly deploying local repo to Heroku so that th	0	2	n/a
As an engineering manager I want to see code (HTML, CSS, and JS) that:			
is cleanly written, in separate files with a singular goal when possible	0	2	
has no unused functions or variables	0	2	
has expressive variable, function, and class names	0	2	
is organized into a coherent flow	0	2	
has a well-developed seed file which will rebuild the appropriate tables, and populate some initial data	0	2	

Application Specific Requirements (70%)

Total Score	Total Possible	Grade For Application Specific Requirements (70%)
0	32	0.00%

Grade

Criteria	Score	Possible	Comments
As a consumer of your API I want to:			
be able to register for an account with a username and password such that:			
- no duplicate username can be registered	0	2	
- no password under 8 characters in length can be used	0	2	
be secure knowing that my password will not be returned in any response when I hit any API endpoint	0	2	
be able to login with my correct username/password combination and to be returned a JSON Web Token for future requests	0	2	
be able to retrieve a list of all activities (exercises) from the database	0	2	
be able to retrieve a list of all routines (collections of activities) from the database, and each routine should have an array of the activities that it contains	0	2	
be able to retrieve a list of all routines that a specific user has created	0	2	
be able to retrieve a list of all routines that feature a specific activity	0	2	
be able to create a new activity, only if I am logged in	0	2	
be able to update an activity, only if I am logged in (even if I was not the creator)	0	2	
be able to create a new routine, only if I am logged in	0	2	
be able to update or delete a routine, only if I am logged in _as_ the routine creator	0	2	
be able to add an activity to a routine, only if it does not currently contain it and only if I am logged in _as_ the routine creator	0	2	
be able to update the number of times or duration that an activity has in a certain routine, only if I am logged in _as_ the routine creator	0	2	
be able to remove an activity from a routine, only if I am logged in _as_ the routine creator	0	2	
be able to receive descriptive errors when I have made a mistake	0	2	
Extra Credit:			
be secure knowing that my password is not stored as plain-text, but rather it is hashed before being stored	0	2	