## Assignment 6 (Updated)

This Assignment is worth 5% of the Web Information Processing CA Component. This is an open-book, graded assignment. You may use online resources for reference purposes only to help with the assignment. Please cite all references as comments in your submissions. You cannot directly reuse HTML/CSS/JS solution code from online sources. You must not engage with another student, in person or electronically (phone, social media, etc.) to secure assistance with this assignment. If you do so you will receive an automatic fail (0%). We will perform similarity checks on submitted assignments to check for collaborative efforts. A reasonable attempt at this assignment will gain you 15% of your continual assignment marks.

You are able to use **EITHER** Express or HTTP Server for this assignment, regardless of your solution to Assignment 5.

**Tasks**

You are required to develop a Node.js application that implements the RESTful API functionality for an application which maintains the following database:

1. **Gym Member:** Personal Details for members of a new gym. This information must include the following: [ID (Integer), Title, First Name, Last Name, Email Address, Premium Membership (Boolean)]
2. **Gym Class:** Details about different classes that the gym offers to its members. This information must include the following: [ID, Class Name, Class Day, Session Length (Hrs) (Integer), Price (Integer), Current Number of Members (Integer)]
3. **Member-Class Information:** Information that links the member of a gym to the several classes that they can take. Gym members must take 3 classes. The information must include the following: [User ID, Class ID].

Your RESTful API should provide CRUD functionality for Creating, Searching (Retrieving), Updating, and Deleting gym member, gym class, and member-class information from a MongoDB database.

There are no requirements to develop a HTML/CSS/JS “front-end” for this assignment. You only need to develop the ”back end” functionality that demonstrates the RESTful API functionality described above. We will test your routes using Postman or Insomnia.

**Assignment 06 - Requirements**

You should include appropriate routes for the each of REST activities for the different entities using the lecture material presented in Lectures 19 and 20. You are required to develop a NodeJS application that implements the functionality outlined above using ExpressJS or HTTP Server, and with the following additional constraints (note that constraints related to your database setup may be carried forward from earlier assignments without change):

1. For this assignment, you should use a MongoDB database to store the information for the user information using either a local installation (or an online MongoDB Cloud Atlas database). Please use NodeJS that has been installed locally. If you do not have NodeJS installed please do so for this assignment, or use http://repl.it. You will also need to install the drivers for accessing MongoDB. If you cannot install these as you do not have a computer that facilitates installation please contact Mark McCormack (mark.mccormack@mu.ie). A video will be posted to Moodle demonstrating how to use online resources for this assignment if you do not have installation privileges.

2. Please note that you should implement best practice when it comes to NoSQL (document) database design for this assignment, i.e., you may choose to have normalised or de-normalised models, or a combination approach.

3. You may write functions to randomly create personal and address data and you may use the name generator functions provided for the Assignment 03 if you wish. You are also welcome to "hard code" data that will be used by the CRUD operations.

4. Titles should include Mx, Ms, Mr, Mrs, Miss, Dr or Other (specify).

5. Your code should include a brief description for the database design (your data modelling approach) and the impact on your code development. This should be included as a comment at the bottom of your code submission.

6. For this assignment you do not need to validate data in the controllers. You may validate if you wish, and have time, as shown in Lessons L20.01 - L20.05 .

7. For this assignment you do not need to generate online forms to collect and validate data sent to the database. Please include an example of routes as comments in your program code.

8. Please "hard-code" your authentication details into the database. If you are using MongoDB please allow access from anywhere (whitelist 0.0.0.0). We will need access to your database in order to correct your assignment. Please note that a MongoDB Atlas database is preferable for this assignment, you must create an account unless you do not have internet access.

**Assignment 06 - Development Notes**

Please adhere to the following development constraints:

1. You must use a RESTful framework (for example, ExpressJS or HTTP Server) for this assignment. You may use either NodeJS's MongoDB or Mongoose modules for manipulating your database (the latter is recommended). You may refer to online resources such as MongoDB, W3schools, of course.

2. You must comment your code, clearly indicating, how your code implements the solution described above in the “Assignment 06 - Requirements” section.

Please note that there are many sample (JS) solutions for implementing similar solutions (REST) functionality available online. While it is fine to consult these, and accompanying articles, for references, you may not re-use code from these projects.

Please cite your reference sources in your codebase. We will search and identify online coding solutions to similar problems for the purposes of checking against submitted solutions in instances where we have concerns about code originality.

**Important Submission Details**

Before submitting your assignment students should check that their solution works in Chrome and/or Firefox. Please indicate the Browser, Operating System (Linux/Windows/MacOS) and Browser version used for testing (as a comment in your submitted code). If you use an online IDE please clearly specify the IDE and provide a link where possible.

All work must be submitted via Moodle (see "Assignments" section for submission). Work submitted via other means will not be accepted unless you have prior arrangements with the Head Demonstrator (Sean Conroy) or the lecturer. All work MUST be submitted by the due-date deadline.

Late submissions will not be accepted. The assignment submission is a zip file named “assignment-06-xxxxxxxxx.zip” (where “xxxxxxxxx” is your student id) containing solution files, e.g. named “assignment-06.js”, “assignment-06.php”, etc. together with any other resources used in the assignment solution.

Please include a dump of the data from your database (as a text file) names “assignment-06.txt”. Please ensure that all external files use relative directory referencing, rather than hard-coding the files’ location. And ensure that you include your subdirectory files. Do not include the node-modules directory.