COMPX322-23A: Assignment Three

Due Date: Monday May 22nd, 10 am

Web Services: REST API for Project Management Application

For this coursework you are required to implement REST API for the Project Management Application, which allows users to manage their projects. You will use:

- HTTP Verbs to make request

  [**C**(POST)

  R(GET)

  U(PUT)

  D(DELETE)]

- HTTP Response codes to indicate status
- MySQL database
- Node.js and Express.js back-end

Application Description

This is a simple Node.js RESTful CRUD API using Express to interact with a MySQL database. This RESTful API should be able to respond to the following requests:

- Create new projects and save it back to the database.
- Get list of all projects available in the database.
- Get projects by Project ID.
- Get projects by Project Name.
- Update project information by Project ID.
- Delete project by Project ID.
- Delete all projects from the database.

Implementation

You have been provided with a skeleton implementation on Moodle, along with a **ProjectsDatabase.sql** file to create the database table.

You should extend the skeleton code rather than develop a completely new solution yourself. For the directory of your assignment implementation, use *restful-assn3.*

What you need to do:

1. Download the skeleton code. In the terminal, run **npm install** to install all the required dependencies.
2. Review **server.js** to understand how the Express web-server has been setup.
3. In the **config** directory, copy **db.config.js.in** file to **db.config.js**, and replace the username, password and database entries with your own credentials.
4. Ingest **ProjectsDatabase.sql**, into your MySQL instance, to create a *projects* table with five rows inserted in the table.
5. In the **models** directory, in the **projects.model.js** file, expand upon the minimal constructor for Project that is provided, and use the database connection to write CRUD functions:
   You need to support the following functions [user *query()* method]:
   - Create Projects
   - Retrieve All Projects
   - Retrieve Projects by ID
   - Retrieve Projects by Project Name
   - Update Projects by ID
   - Delete Projects by ID
   - Delete All Projects
6. In the **routes** directory, in a **projects.routes.js** file, set all the endpoints.
7. In controller folder, in **projects.controller.js** file, write the controllers with CRUD functions.
8. Test API using POSTMAN.

What to Submit and How

All pertinent material you have developed for this assignment must be submitted electronically using Moodle. The submitted files must be sufficient to recreate your app by running *npm install* followed by *npm start*. **Do not include your *node_modules* directory, as this is not needed to reconstitute your project using *npm install*. Moreover, it could contain binaries files specific to the computer platform you developed the assignment on that are incompatible with the computer system used to test your submitted assignment solution.**

You may choose between submitting a ZIP file or a 'tar-ball' (*.tar.gz*). For the former, use the name *restful-assn4.zip* and *restful-assn4.tar.gz* for the latter. See the Week 8 Lab Exercise for instructions on how to 'tar up' your files, if you are unfamiliar with this option. **Marks will be deducted for submitted assignments that do meet these requirements.**

In the COMPX322 Moodle site, you will see an *Assignment 3* hyperlink to the submission page. This link allows you to upload your *tar.gz*/*zip* file. You can do this as many times as you want up to the submission deadline for the assignment.

When you submit a file, Moodle will ask you to confirm that what you have submitted is your own work, and will provide you with a 'receipt' that establishes that you have indeed submitted something. No other mechanism for submission will be accepted.

How Your Work will be Assessed

The assignment will be marked out of 50 as follows:

API meets functional requirements: • Create Projects • Retrieve All Projects • Retrieve projects by Project ID • Retrieve projects by Project Name • Update projects by Project ID • Delete projects by Project ID • Delete all Projects	35 marks
Set up the Routes	5 marks
API Testing using Postman *Include indicative comments in the code for each RESTful 'noun' you have implemented, as to the testing you have done with Postman*	5 marks
Code is clearly formatted and commented	5 marks

The deduction for incorrectly submitted files is capped at 2 marks.