PROG 3017 Assignment 2 - PHASE 1

**FULL STACK DEVELOPMENT**

# 8% of overall course mark

# Prerequisites

### Required Labs: Lab 2 (NodeJS) and Lab 3 (Express JS)

# Summary

In assignment 1, you were tasked with conceiving and creating a dataset in MongoDB that you would use for your FullStack application. In this portion of the application, you will create the API to interface with this data using Express JS and using the skills that you developed in the previous labs. You will create the beginnings of the API to interface with the data created in Assignment 1.

Before beginning the outlined requirements below, make sure that you have completed the following pre-requirements:

1. You’ll be working from your already cloned FullStackApp repository. (You already submitted your Assignment 1 materials in this repo) The repo will be used moving forward for all work completed for your own unique FullStack application.
2. Be sure to follow the instructions provided in Week 6 lectures to ensure that you have the correct structure and scaffolding of code needed to move forward with the application. Refer to the class recordings for detailed instructions.

# Application Requirements (30 points)

Once you have your repository cloned and your Express application generated, continue with the following requirements

1. **Connect to your Mongo Database via Express. (5 pts)**

Demonstrate that your API application can successfully connect to your local (VM) MongoDB instance.

1. **Create the Mongoose Schema for your Mongo Data Document Structure(s) (10 pts)**

In a folder called ***Models***, define all relevant schemas that relate to your database document structure. (You may have only one). Be sure to modularize your models so that they can be imported in other areas of your application. Mongoose relies on the creation of Schemas which provide structure and a blueprint for the structure of the data that will be stored in your database.

1. **Create The Endpoints For your API (10 pts.)**

Your API will contain all endpoints that enable full CRUD (Create, Read, Update, Delete) functionality. The endpoints should use Mongoose to appropriately interact with your MongoDB data.

Endpoints should include:

* Get all documents
* Get one document by Id
* Create new document
* Update an existing document by Id
* Delete an existing document by Id

All endpoints should respond with the appropriate HTTP codes depending on the result of executing the endpoint. You do not need to apply validation to your endpoints at this point.

1. **Add Environment Variable Capability to your Application (5 pts)**

Configure your application to be able to explicitly set environment variables for your application. Use the ***dotenv*** npm package and create a .env file which will allow you to set your environment variables.

# Instructions

1. Don’t forget that a code review is a **necessary** part of this assignment. You will need to show your code to the instructor in class on the due date while going through an evaluation of the site’s functionality. You will need to explain how the code works and complete the code review part of the rubric. You will need to do this to at least a developing level (see the Note in the rubric below).
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***