Assignment 13 – Putting it All Together  
COS318 – FA2018

Due Date: December 13th, 2018  
Turn in all files using Moodle

You’ve made it; this is the home stretch, the final web programming assignment. Finish strong! It still needs a theme. That’s where you come in. This assignment will bring together many of the topics of web programming and will be themed with something you enjoy. Pick something fun and create your solution to this assignment using that theme.

This assignment will involve you creating a WebAPI or MVC project in Visual Studio with HTML/javascript to go along with it. You have a lot of freedom with this assignment. Be creative. Your assignment must include the following items at a minimum, anything beyond that is up to you.

1. **(20 Points) Controller**
   1. Create a controller with at least three endpoints.
      1. At least one endpoint must create data.
      2. At least one endpoint must retrieve data.
      3. At least one endpoint must edit or delete data.
2. **(20 Points) Model**
   1. Create a model that has at least four properties.
      1. The model properties must be validated using attributes.
   2. Create an in-memory database that contains a list or dictionary of the model objects.
3. **(20 Points) Filter/Middleware**
   1. Create a filter for your controller. This is in addition to a filter that validates your ModelState.
      1. This filter can do whatever you would like, but the best solutions will have the filter doing something useful.
4. **(20 Points) HTML/Javascript**
   1. Create an HTML page that displays data from your controller.
   2. Use forms and buttons and allow a user to read, create, and update data.
      1. The javascript code should make REST requests to each of the endpoints of your server.
   3. The HTML should be styled using CSS classes. Use any colors, layouts, fonts, etc. that you like.
5. **(20 Points)** Code style, formatting, completeness, and quality.

Stretch Levels

Try to complete these stretch levels for extra credit. If you try for the stretch levels, make sure to type it in the comments on Moodle so I don’t miss it.

The stretch levels for this assignment are worth more reputation than any stretch levels for assignments so far. If ever you were going to stretch yourself, now would be the time. Remember, finish strong!

**Bronze Level (something creative here related to your theme)**

The bronze stretch level is up to you. Add something into your assignment that is worthy of a stretch level. Explain why it’s worthy in the Moodle comments.

**Silver Level (another hilarious character or idea from your theme)**

The silver stretch level is up to you. Add something into your assignment that is worthy of a stretch level. Explain why it’s worthy in the Moodle comments.

**Gold Level (the best aspect of your theme)**

The gold stretch level is up to you. Add something into your assignment that is worthy of a stretch level. Explain why it’s worthy in the Moodle comments.

The Rules

1. No inline styles or javascript.
2. Error messages must be “in-page” i.e. no pop-ups or alerts.
3. Any resources not created by you (images, javascript libraries, etc.) must be referenced using a CDN or URL, not directly included in your assignment submission.
4. Service/data/model classes must not have any http, request, or response references.
5. Controller entity classes must not be used directly to store data on the server; translate them into a model (data storage) class before saving the data. Conversely, controllers must not send any model classes to the user; translate them into controller entity classes before sending the response.
6. You may not use any synchronous methods in your C# code wherever there is an async option.
7. All service class instances must be obtained using dependency injection.
8. All requests that submit a body to your server must have their JSON structure validated with ModelState. The controller is not allowed to validate the ModelState directly; this must be done in a filter.
9. All of your controllers/entities must enforce the usage of an api version. Your namespace and folder structure should match the api version of the controller/entities.