**Assignment 6 Pets-R-Us, Part 3**

**Instructions**

**Part 1: MongoDB Atlas Setup**

1. ~~Follow the instructions in the MongoDB Atlas guide and create a new database for the pets-r-us project. Before you proceed to part 2, verify the following has been completed.~~ 
   1. ~~MongoDB Atlas account created.~~
   2. ~~Database created.~~
   3. ~~Custom role created.~~
   4. ~~Custom user created.~~
   5. Customers collection created.

Special note. When you create your web340DB database you will be prompted to create a new database collection. Name it “customers.” This collection will be used in part 2 of this assignment.

**Part 2: Registration**

1. ~~Add a dependency for mongoose (npm package).~~
2. ~~Add a new folder to the pets-r-us project and name it models. All mongoose models will be added to this folder.~~
3. ~~Add a new JavaScript file to the model’s folder named customer.js. And, create a new mongoose model named Customer with properties for customerId and email. Set both properties data types to strings, make them unique, and required.~~
4. Design and build the registration process.

**Additional programming requirements**

1. ~~Build the registration register page. At minimum there should be fields for customerId and email.~~
2. ~~Add an HTTP GET route to display the registration page. This route will display register.ejs.~~
3. ~~Add an HTTP POST route to handle the registration pages form submission. If the call to the create() function is successful, route users back to the landing page.~~

**~~Additional programming requirements~~**

1. ~~Use Mongoose’s built-in create() function to insert a new customer object into the customers collection.~~

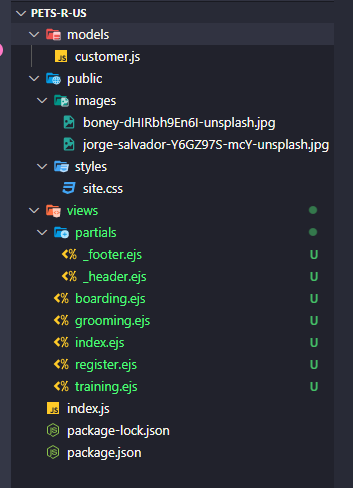
**Special note.** The courses GitHub repository has several examples of how to create a base Node.js project, which can be located [here](https://github.com/buwebdev/web-340/tree/master/week-6). Once you have accessed the repository, you should view and test the projects in the following order:

* 1. mongoose-connection. This project demonstrates how to connect to MongoDB from Node.js. As you review this project pay close attention to line 6-13.
  2. mongoose-model. This project demonstrates how to create a Mongoose model.
  3. fms. This project follows along with the weekly tasks in the pets-r-us project (with some modifications) and should be used as a reference guide while working on the current week’s assignment. Of particular note this week is the additional lines of code added to the index.js file for urlencoded and json(); lines 15-16.

**Special note**. You will need to understand how these projects work to complete this week’s assignment. That means you should download/clone the repositories, open them in VS Code to review their code bases, and run and test them in a web browser. This is the only way you will fully understand how to interact with a MongoDB database through Express.

1. By the end of this week your project folder structure should resemble the following:

**Exhibit 1 Project Folder Structure**



**Special note.** you are building the EJS pages and Node.js routes to interact with the pages. Also, as you can see from the above image, I moved the partial views to a folder named partials. I did this to keep the project organized.