SIMPLE MODELS

This assignment is intended to get you working with Models in Node.js using Mongoose.

Due by due date on the dropbox.

To do this assignment locally, you will need MongoDB running on your machine. Instructions on how to do this on a lab machine or your home machine can be found in the "Setting Up MongoDB Locally" guide on MyCourses.

To deploy this assignment to Heroku, you will need MongoDB running in the cloud. Instructions on how to do this can be found in the "MongoDB Cloud Setup" guide on MyCourses.

Assignment Details

In this assignment, you will add a new model to our existing MVC example.

1. Download the finished code from our Simple MVC Example
   https://github.com/IGM-RichMedia-at-RIT/simple-mvc-example-done

2. Make sure that you can run it. You will need MongoDB running locally to do this (see above).

3. Create a new model file called Dog.js

   Every Dog should have a
   - name (string)
   - breed (string)
   - age (number)
   - createdDate (Date);

4. In the page3.handlebars view

   - Create a form that allows for the creation of a new Dog. The user should be able to enter the name, breed and age.

   - Create a form that allows for the lookup of a Dog by name and increases its age. The user should be able to enter the name of a dog and have it increase the age of only that dog. It should also return an error if the dog does not exist.

-- Continued on Next Page --

5. In your controller code, add the following logic (note: you'll also need to add routes with the proper methods and URLs to make this functionality work).

    - A function for creating a new dog
        - If the name, breed, and/or age is missing, the function won't create a dog and will instead send the client an error.

    - A function for looking up a dog by name
        - If the dog exists, it's age is increased by 1.
            - Note that to do this will involve either 1) utilizing techniques covered in two separate functions from the demo code, or 2) utilizing functionality of mongoose not covered in class. Feel free to check out https://mongoosejs.com/docs/api/model.html
        - If it doesn't exist, return a message to the user that the dog doesn't exist.

6. Create a page4.handlebars view and a route for /page4 that renders this new page. The page4 view should:

    - Show all dogs in a list view. For each dog, there should be a <li> tag. This should continue to work as more dogs get added.

    Refer to the handlebars documentation if needed. http://handlebarsjs.com/

-- Submission Instructions on Next Page --

ESLint & Code Quality

You are required to use ESLint with the AirBnB configuration for code quality checks. The eslintrc config file was given to you in the starter files. Your **server** code must pass ESLint to get credit for this part. Warnings are okay, but errors must be fixed. I won't require ESLint on the client-side code in this assignment. ***If you cannot fix an error or need help, please ask us!***

You are also required to use the "pretest" script in order to run ESLint. I should be able to run "*npm test*" and have ESLint scan your code for me.

Continuous Integration

You are required to use CircleCI for continuous integration. Your build must succeed and be handed in with the assignment. The circle.yml config file was given to you in the starter files. CircleCI will build based on your *npm test* and ESLint configuration.

For instructions on how to do this, look at the ***CircleCI Setup*** instructions on mycourses.

-- Rubric on Next Page --

SIMPLE MODELS

Rubric

DESCRIPTION	SCORE	VALUE %
App works correctly – all pages, forms and functions work correctly.		**20**
Dog model has the correct data types and is functional – The Dog.js model should contain the correctly made model, schema, and exports.		**20**
page3.handlebars has functional form to allow creation of a new dog – page3's dog creation form should submit correctly and have the correct types for name, breed and age. The server should receive the information and handle it appropriately.		**10**
page3.handlebars has a functional form that allows lookup of dog by name and increases its age – page3's dog lookup form should submit correctly and have the correct type for name. The server should receive the information and handle it appropriately.		**20**
Controllers have correct logic for Dog – Controller should allow for dog creation, error handling for invalid data or database errors, looking up a dog by name to increase its age and messaging the user if a dog does not exist.		**15**
page4.handlebars correctly shows all dogs in a list – page4 should show an itemized list of all of the current Dogs in list items. If a dog is made, then going back to page4 should show the updated results.		**15**
Git Penality – If your code is not in a Git repo or the link is not handed in, there will be a penalty.		**-30% (this time)**
Heroku Penalty – If your app is not functional on Heroku, there will be a penalty. This means you need to correctly hookup the addons on Heroku.		**-30% (this time)**
CircleCI Penalty – If your build is not successful on CircleCI, there will be a penalty.		**-20% (this time)**
ESLint Penalties – I should be able to run 'npm test' and have ESLint scan your code without errors. Warnings are acceptable. **You may not alter the rules in the .eslintrc file provided.** **Check your code with ESLint often during development!** **If you cannot fix an error, please ask us!**	1 Error 2 Errors 3 Errors 4 Errors 5+ Errors	**-10%** **-15%** **-20%** **-25%** **-35%**
Additional Penalties – These are point deductions for run time errors, poorly written code or improper code. There are no set values for penalties. The more penalties you make, the more points you will lose.		
TOTAL	**TOTAL**	**100%**

Submission:

By the due date please submit a zip of your project to the dropbox. **Do not** include the node_modules folder in the zip or in your git repo.

In the submission comments include the following:

- *a link to your Heroku App (not the dashboard but the working web link from 'open app'),*
- *a link to your Github repo (if private, please add the TA and myself).*
- a link to your circleCI build *in the submission comments.*