🎨 Creative Project #4

Re-submit Assignment

Due Apr 9 by 11:59pm **Points** 100 **Submitting** a website url

For this lab, you will create a website that uses Vue for the front end and Node, Express, and Mongo for the back end. The front end should use an API to communicate with the back end, including creating, reading, updating, and deleting data stored on the back end. This website can be for anything you want, but should not be similar to the Vue apps we have already built so far. You must create your own original work. Your creative projects can build on each other.

Pair Programming

You are welcome to use pair programming for this assignment. With pair programming, two students work together at one workstation. One takes the role of the "driver" and writes code while the other takes the role of the "observer" to review each line of code as it is typed in and offer suggestions. The two students switch roles frequently during the project. The two students should have relatively equal levels of experience.

Pair programming is **not** assigning different parts of the project to two different people, and then meeting later to put the work together. You should follow the above guidelines for pair programming to create a better learning experience.

Requirements

This is a creative project, so you are welcome to build what you would like. The basic requirements are:

- The site uses Vue CLI for the front end
- The site uses Node, Express, and Mongo on the back end
- The site uses an API between the front end and back end, and supports create, read, update, and delete operations
- The site has multiple pages (multiple views in the Router)
- The site contains a footer that links to your public GitHub repository

In addition, the **site should have a robust data model**. Ideally, this should be multiple collections of documents and a detailed schema for each collection. For example, if you were building a recipe site, you could have a collection for recipes and for people. The recipes should have a robust schema, such as name, description, ingredient list, photo. The people should have a robust schema, such as name, interests, favorite recipes, etc. You want to avoid having an overly simplistic site.

The GitHub repository should be in your own GitHub account.

Submission

Please submit to Canvas the URL for your web server, where your code is running (e.g. on your DigitalOcean server). On this website, in a footer, you must include a link to your GitHub repository. Use a unique URL for the project, e.g. cp3.emmasmith.org.

If you worked with a pair programming partner, both of you should submit the assignment and both of you should list the other person's name in the website footer and in the assignment submission.

Rubric for Grading

We will grade your projects based on:

- you used the Vue CLI for the front end, with multiple pages
- you used Node, Express, and Mongo for the back end,
- you have an API that includes creating, reading, updating and deleting data on the back end,
- you have a robust data model,
- how well you follow web design principles: spacing, typography, color, consistency, and responsive design.

TAs will assign scores in these ranges:

- 90-100: Excellent use of Vue CLI, Node, Express, and Mongo, robust data model, and clearly follows web design principles
- 80-90: Uses Vue CLI, Node, Express and Mongo, good effort was made, but needs a little richer data model or a more complete API or web design principles were completely ignored
- 70-80: Fulfills the minimum requirement of using Vue CLI, Node, Express, and Mongo, but the site, data model, and/or API are too simple