# Front End Technologies Week 12 Coding Assignment

**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week’s assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

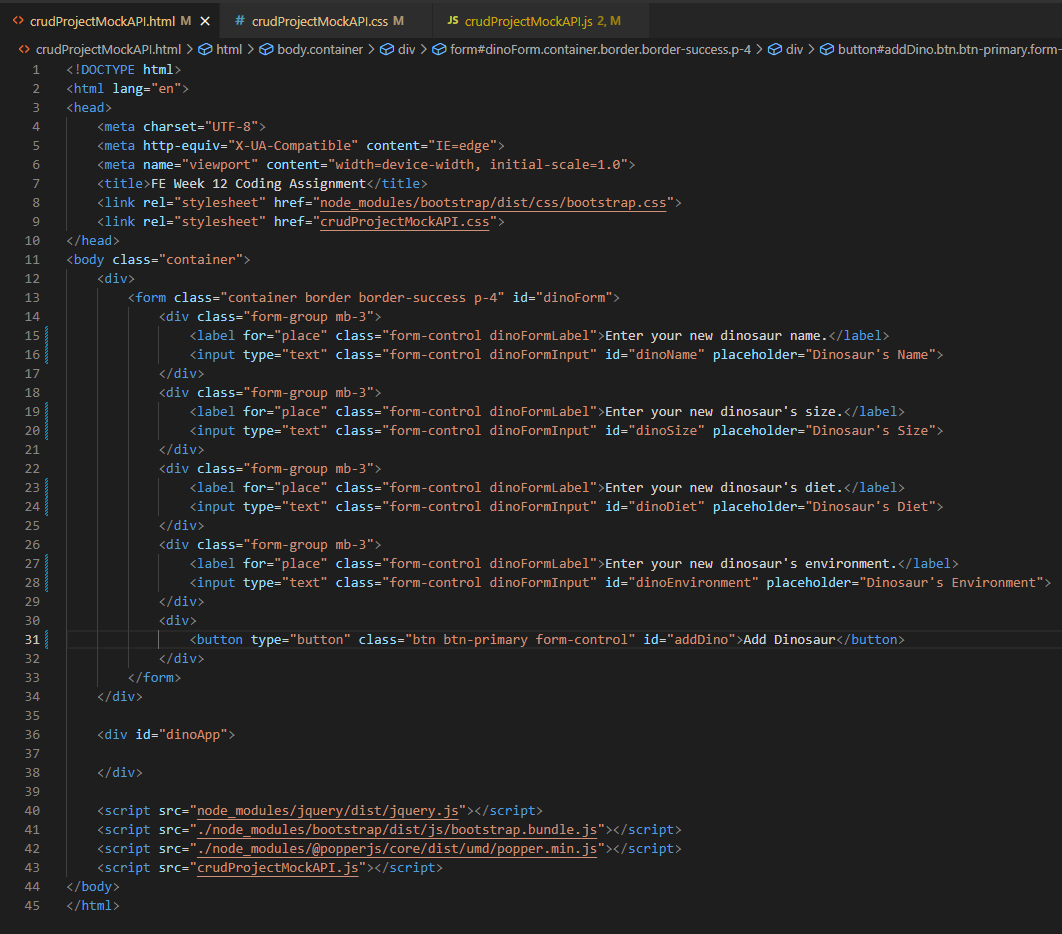
**Coding Steps:**

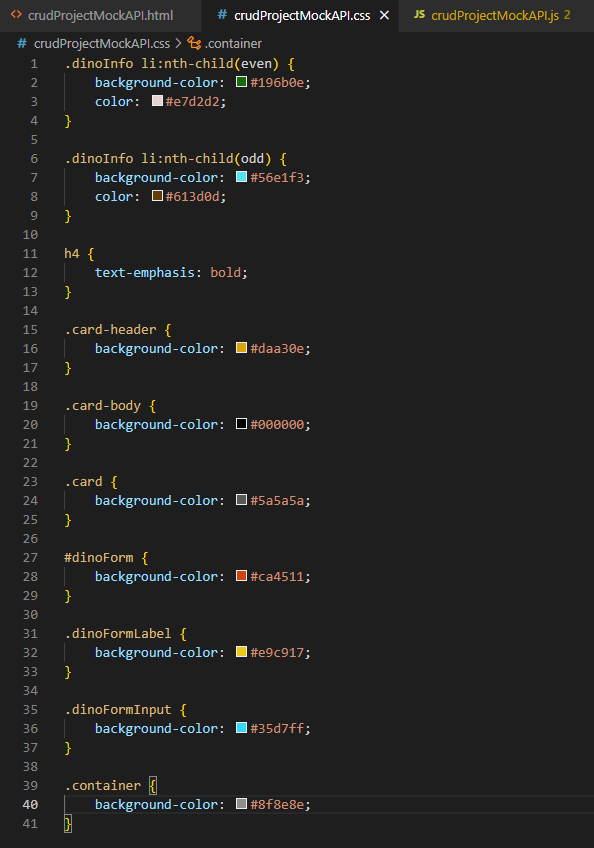
1. Create a full CRUD application of your choice. If you can use an existing API, use AJAX to interact with it. However, you do not have to use an API. If you do not use an API, store the entities you will create, read, update, and delete in an array.
   1. Use a form to add new entities
   2. Build a way for users to update or delete entities
   3. Use Bootstrap and CSS to style your project

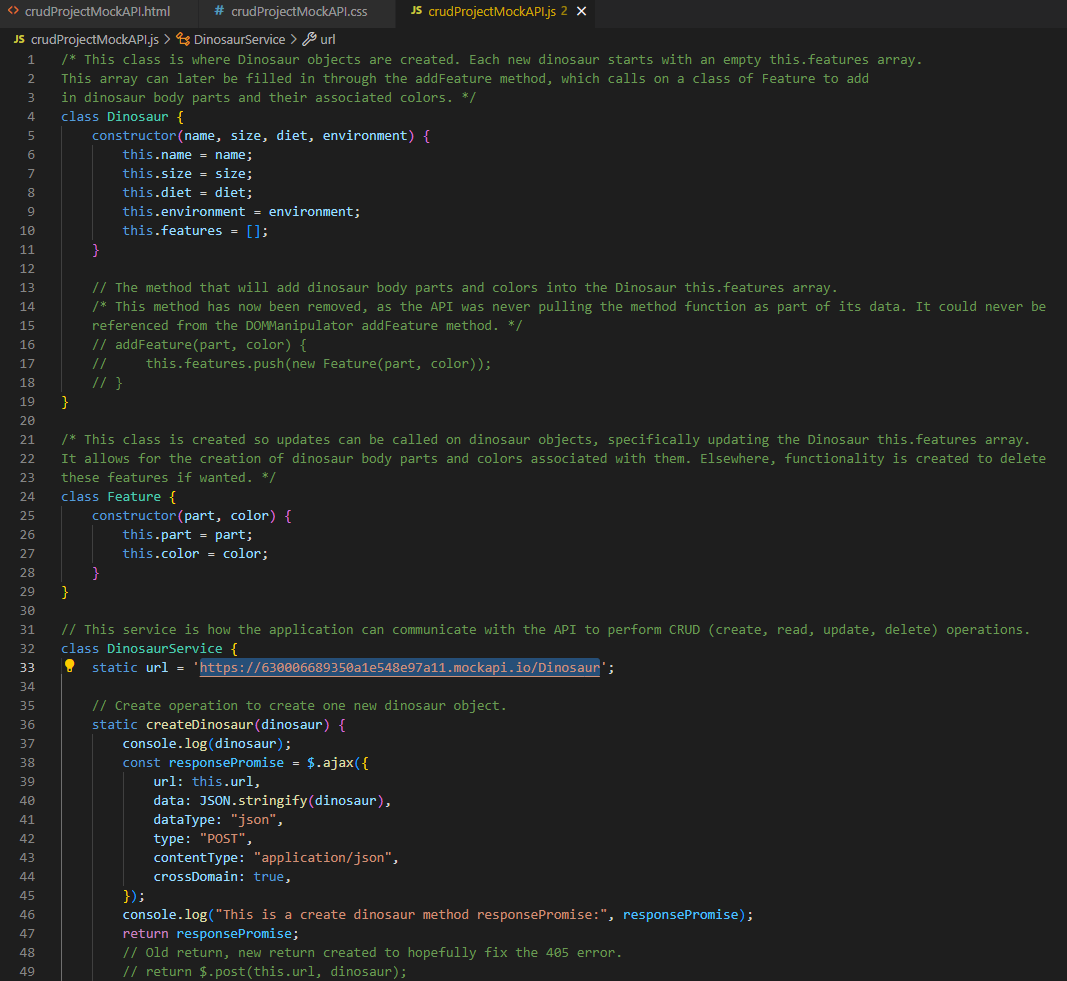
**PLEASE READ:**

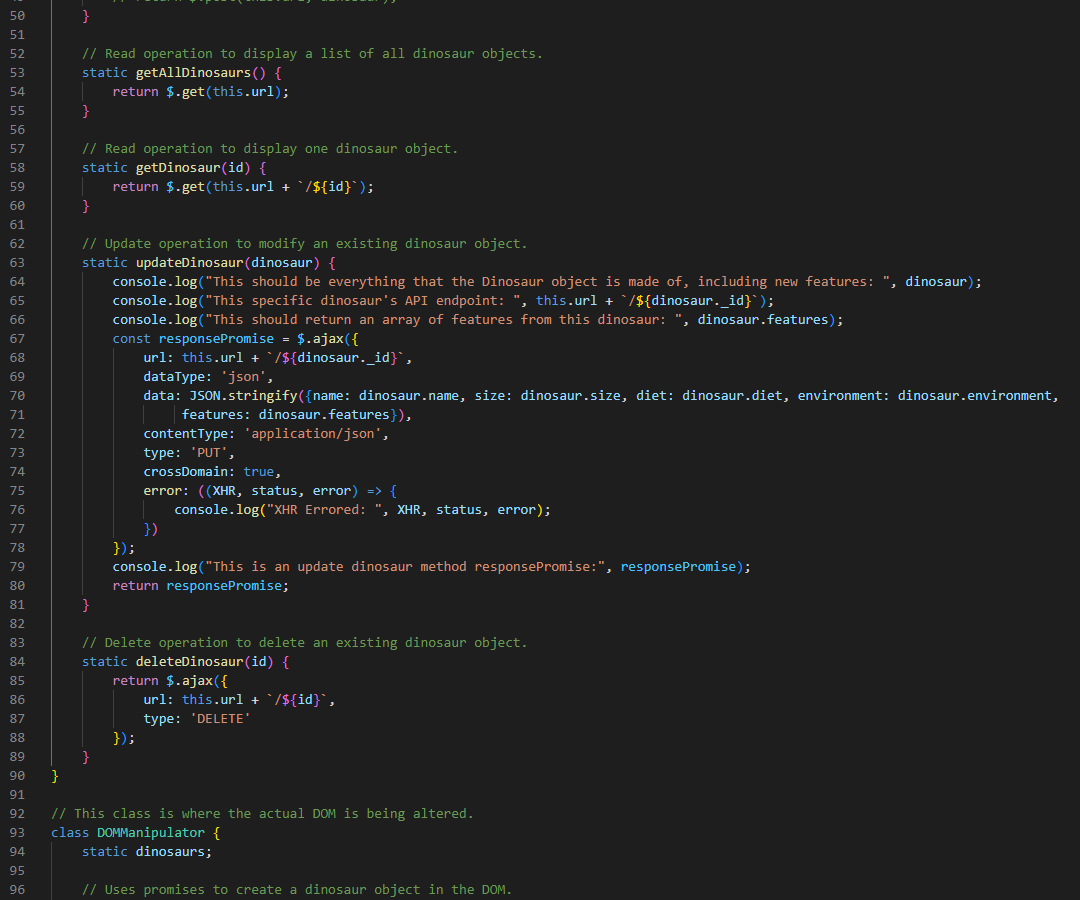
First and foremost: please use the files with the ending MockAPI, as the others are based on an API (crudcrud) that has a limited number of tries (100) and time (1 day) before expiration. The file testingMyHTML reflects how a card in the project SHOULD look and serves as an attempt to see if buttons could be placed into an unorganized list in HTML. Despite switching my API over to MockAPI, I could not get features to be rendered. You will notice many comments and console logs attempting to find where the issue is presenting itself. The method of addFeature in the class DOMManipulator even has a .catch to try to see if there was a ghost error not presenting itself. The updateDinosaur method in the class DinosaurService even has a way to detect an error as part of the ajax promise. This issue is beyond my understanding, and all attempts at outside help have failed. I am submitting the project as is because I want this to be a proper snapshot of what I have learned, and where I feel this week’s videos fundamentally failed to teach a better understanding of how APIs function.

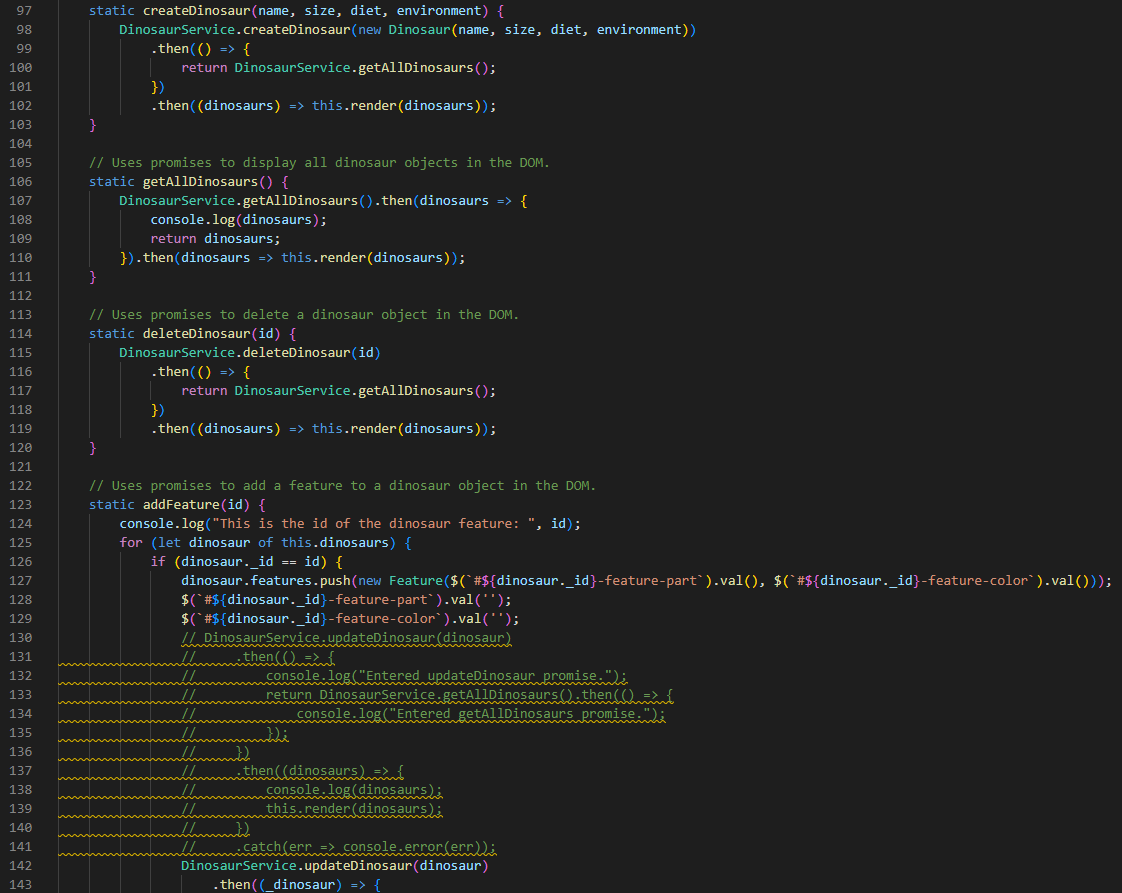
**Screenshots of Code:**

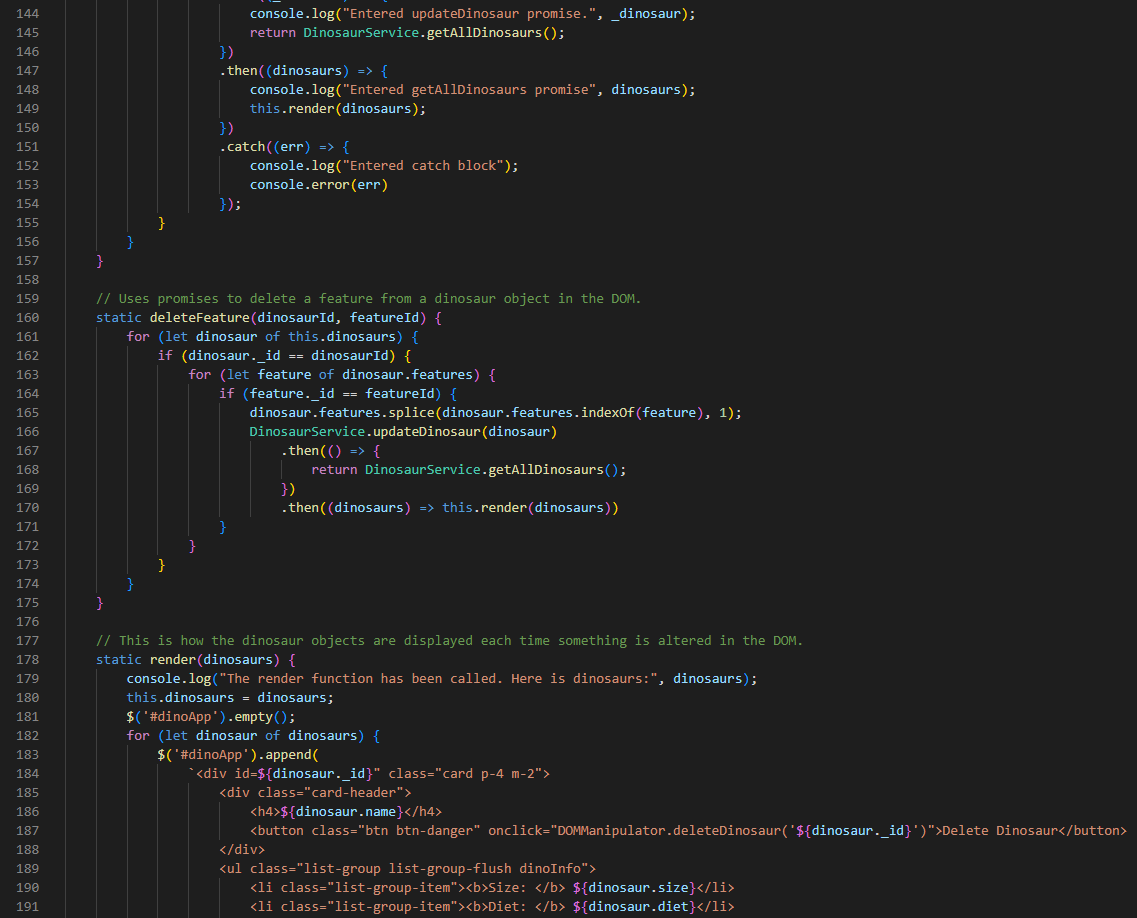


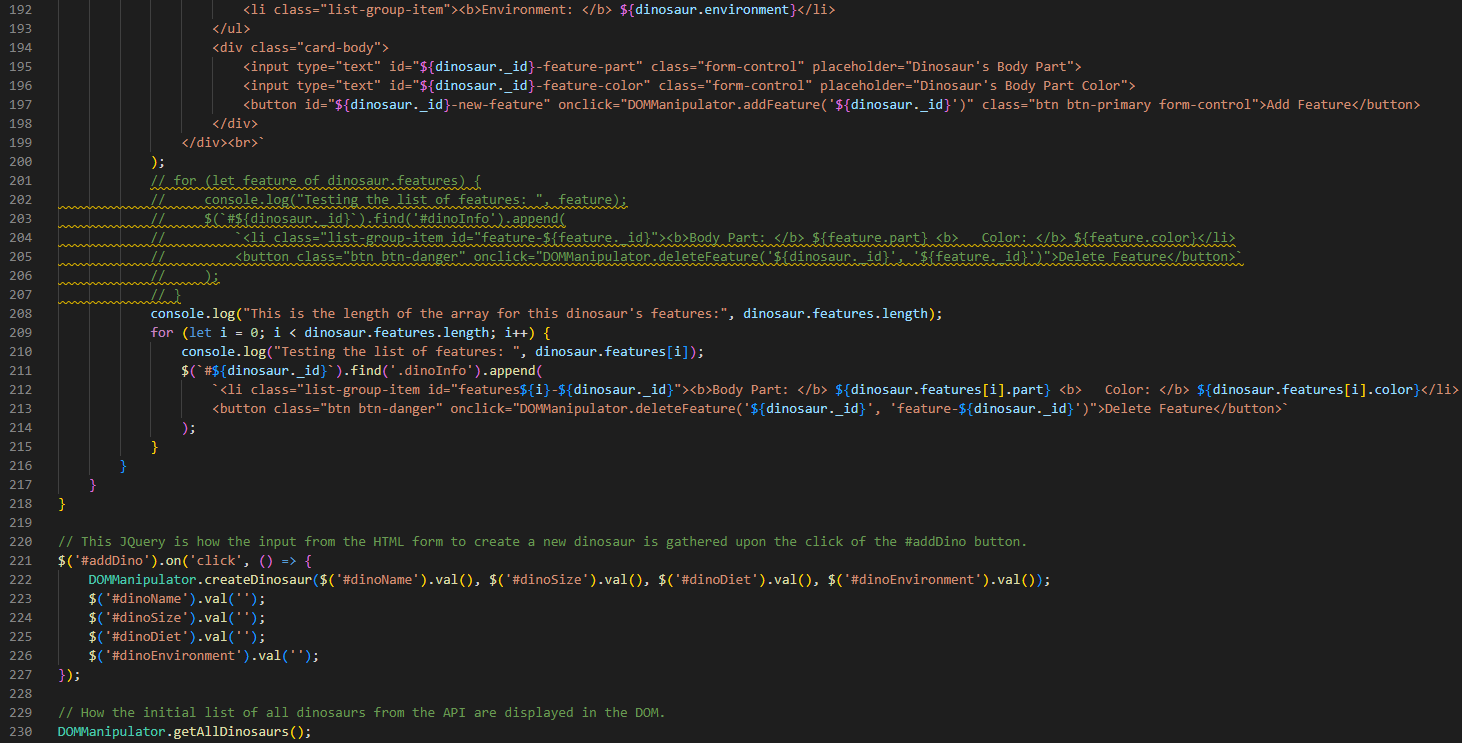






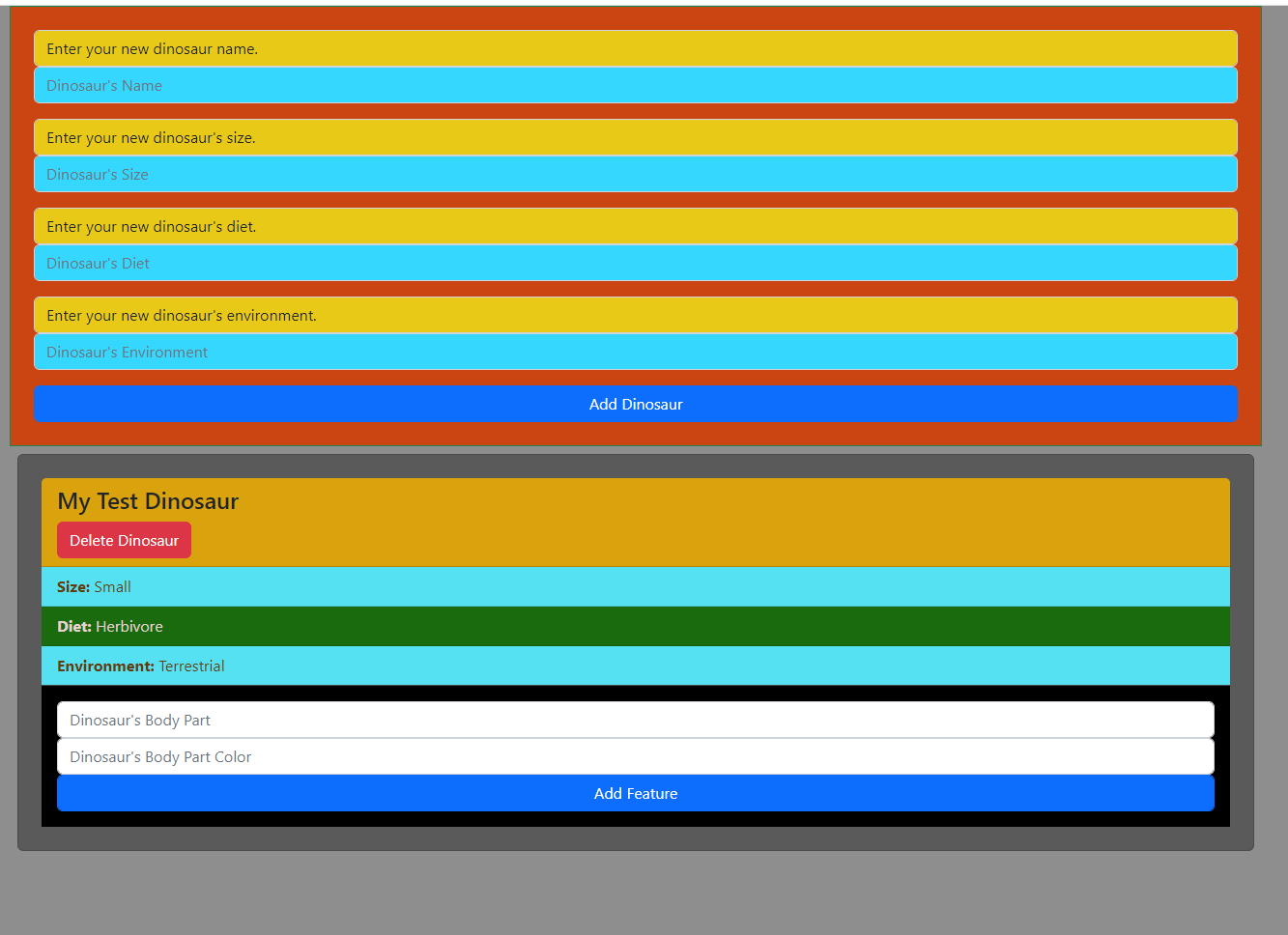


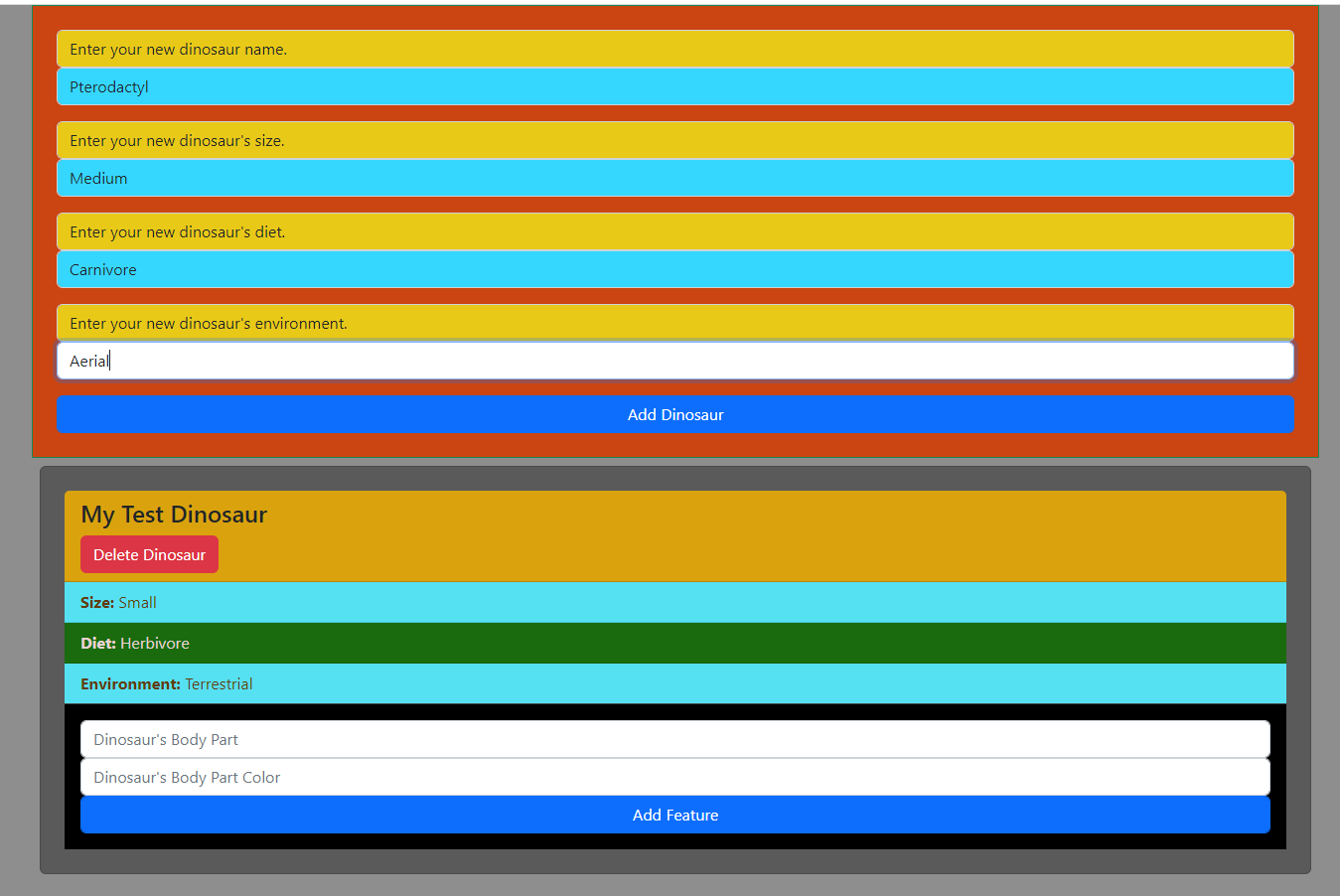




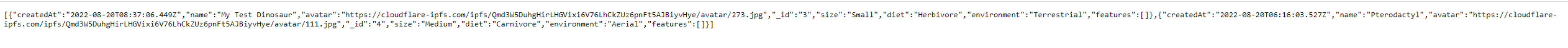
**Screenshots of Running Application:**



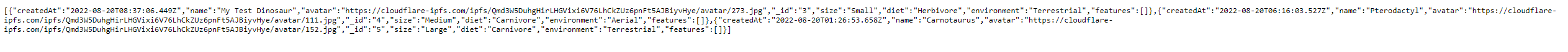


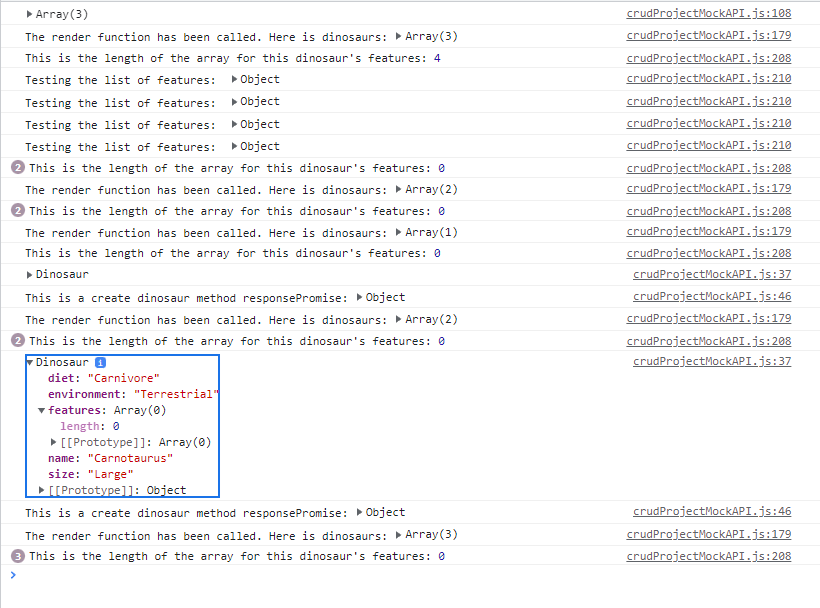


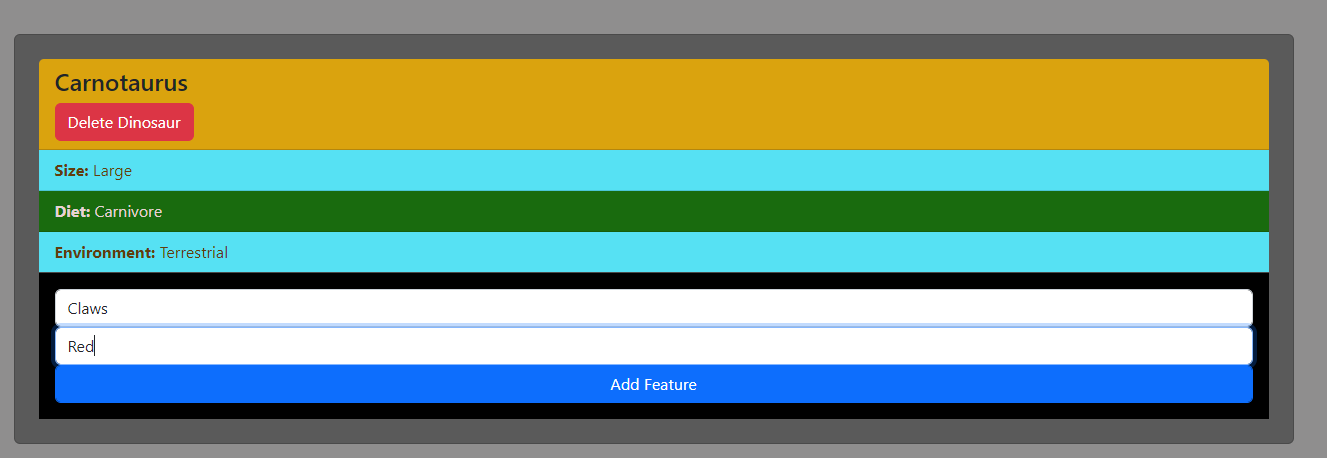


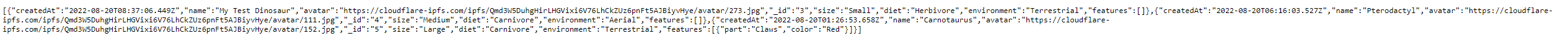




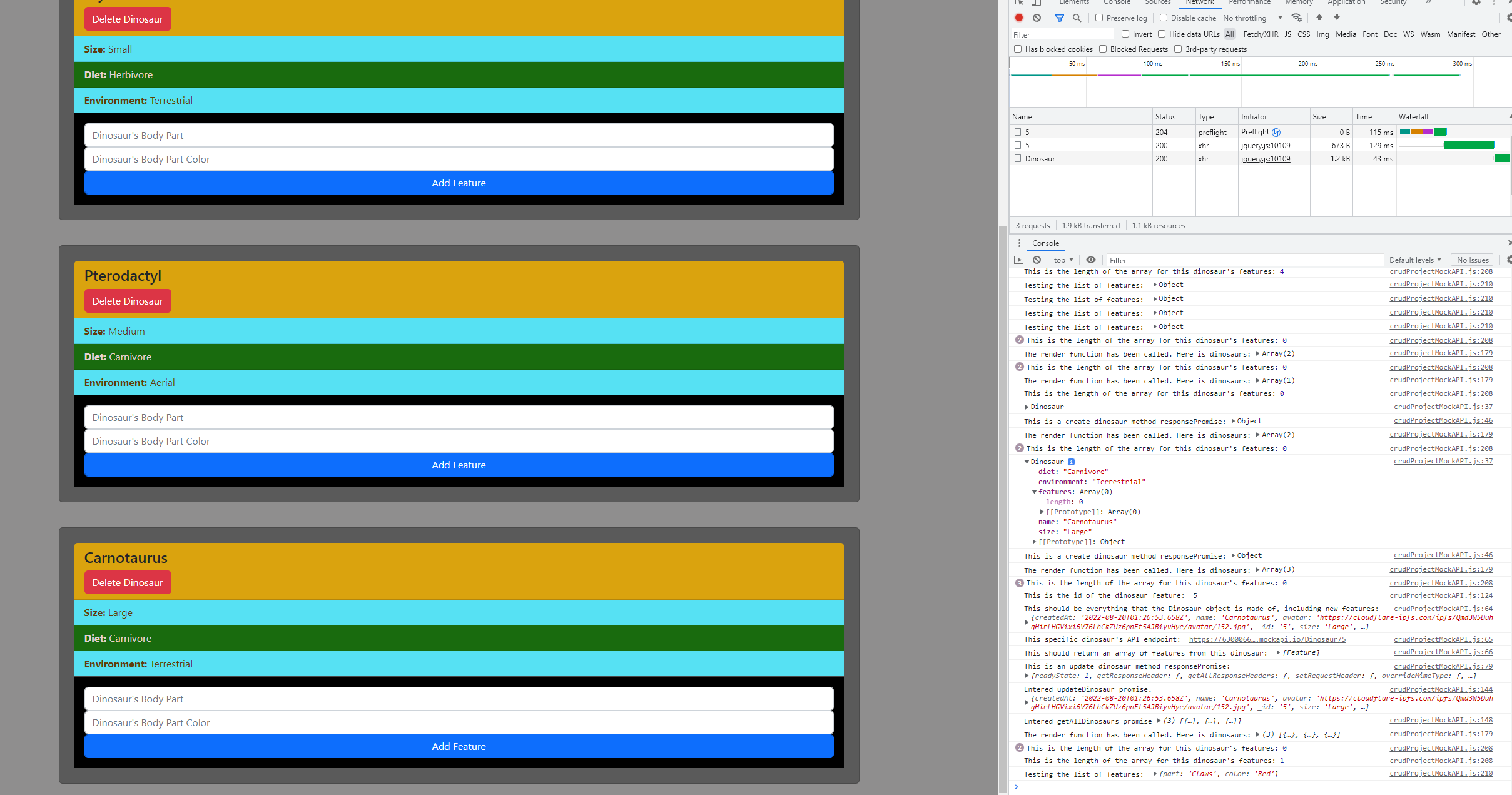








Look at the screenshot below. It will not render the features along with the delete button. I cannot figure out why, and no one else has either.



**URL to GitHub Repository:** <https://github.com/spobrien1116/FEWeek12CodingAssignment>