Charles Campbell

Enhancement Two

27 July 2025

Algorithms & Data Structure Narrative

The original Python CRUD module used in my old client/server development project I made back in February – April of 2025 featured some basic implementations. The module would connect to the MongoDB, execute specific Read algorithms (such as only pulling all instances of 'dogs' from the database), and display them to the front-end client. While other algorithms such as the Create, Update, and Delete were written in the module, they were never implemented and utilized for the user to create a new item to be added to the collection, update any data for the entries, or be able to delete them should the user not need that data anymore. This presented an ample opportunity to not only improve the functionality for a full CRUD module, but would allow me to test my ability in converting it from Python to JavaScript in the MEAN stack.

CRUD operations are an essential part of any piece of software. The necessity to manipulate data for seamless operations are the cornerstone of any application. This is just as true in full-stack development. Showcasing my ability to not only migrate the existing logic from the Python module into the JavaScript environment, but also to enhance each operation for full use in the overall project would be a great example of my front-end, back-end and data manipulation skills. What's more, this also broadcasts my design skills and abilities by creating code that's modular, organized, and scalable by ensuring that each individual algorithm of CRUD can handle the data as intended.

I was able to successfully implement a full CRUD module, where a user can easily click a button to add a new item to the glossary list, click on an existing item and quickly either update it or delete it with a press of a button. By creating a JSON schema, and designing algorithmic operations to handle that data structure and information held within, I was able to cover the course outcomes I sought to meet at the beginning of this enhancement journey. This artifact showcases my ability to deliver a professional-quality technically sound system, where I was able to solve a problem through appropriate computer science practices using algorithmic principles, and utilized innovative techniques and industry standard tools to accomplish solutions and goals.

One of the biggest challenges was the different structure of how each CRUD algorithm is designed in the two different systems. In the Python module, many of the algorithms were almost identical to the terminal commands one would run to access data in the MongoDB. Meanwhile in JavaScript, especially with implementing the JSON schema within the code itself, required a lot more research and testing than in the older artifact. Having to design a full API that could transfer the backend to the front-end required extensive debugging and refactoring. The process did help me learn how important modularity is, even for this SPA that I've designed. Each individual portion allows me to quickly test and locate errors that might arise, while ensuring that potential future growth can be easily done. Not only by myself, but if I was working on a team or another developer took on the project, they could easily understand each algorithms purpose and get a sense of the application as a whole.

I'm looking forward to the next step of enhancing the MongoDB, where I'll be ensuring that the overall data security is well thoughtout and accounted for in the design of the code, while also increasing the user experience by further enhancing the front-end features of the SPA.