Mark McCormack

CS 499

Milestone Three

Enhancement Two: Algorithms and Data Structure

In this enhancement I will be using the same and only two files in my project that come from CS-340 where I created a Python CRUD module and jupyter notebook dash app to interactively display database information from an animal rescue shelter. I’ve chosen to add complexity to my existing algorithms by implementing asynchronous data loading which can help to improve performance, enhance responsiveness, and facilitate a better user experience. Asynchronous data loading is ideal for I/O-bound tasks, such as reading from a database, as it allows these operations to run in the background without blocking the main thread. Another exciting benefit is that asynchronous tasks can be more resource-efficient since they don't require multiple threads or processes. They can handle many tasks within a single thread, reducing the overhead associated with context switching and memory usage.

To enable this I’ve imported the asyncio Python module and I had to switch from using the pymongo module to the motor module because pymongo does not directly support asynchronous data loading in MongoDB. Also the JupyterDash app now integrates dash-extensions to handle asynchronous callbacks in the notebook.

The biggest challenge for this milestone was understanding the fundamental concepts of asynchronous programming and how it differs from synchronous programming along with successfully implementing asynchronous callbacks to make sure the callbacks are correctly defined and executed without blocking the main event loop. Another pain point was ensuring library component compatibility when adding asyncio and switching from pymongo to the motor module.

The key learning for me was going through the process of converting synchronous code to asynchronous code which provided a deeper understanding of asynchronous programming principles. It’s not as simple as importing asyncio as I learned how to use asynchronous libraries like motor for MongoDB and dash-extensions for Dash, which turned out to be essential for handling non-blocking I/O operations in a Python environment.