Mark McCormack

CS 499

Milestone Four

Enhancement Three: Databases

In this final enhancement I am working with the same python CRUD module and jupyter notebook as before that I created in my CS 340 class. For the database enhancement I have successfully implemented MongoDB batch aggregation. To do this I had to add an aggregation method to my CRUD module. Aggregation frameworks in MongoDB offer more capability and value by enabling advanced querying for analytics that involve grouping, filtering, and sorting data. Aggregation pipelines in MongoDB enable enhanced data processing by performing multiple stages of data processing in a single query and reducing round trips between the dash app and the database. Aggregations can help with analyzing data trends, patterns, and statistical calculations which enable modern data-driven decision-making processes.

The most exciting thing I learned about during this database enhancement work has been about performance optimization, which includes dynamic filtering and scalability. In my jupyter notebook I had to retrofit the update\_dashboard callback to use the aggregation pipeline and the fetch\_aggregated\_data function which encapsulates the aggregation logic. Aggregation pipelines are processed on the database server, which can be optimized for such operations, but even if the server isn’t optimized it is still much faster and more efficient at data retrieval compared to client-side processing. As the data grows, using aggregation pipelines ensures that complex data operations remain performant, leveraging MongoDB’s optimized aggregation framework.