# 5-2 Milestone Four: Enhancement Three: Databases

# Artifact Description

The artifact I selected for the databases enhancement is my Arrow Grain Calculator App. Originally built as a simple client-side tool to calculate total arrow grain weight, it has been expanded into a full-stack application. In its enhanced form, it now supports persistent data storage using MongoDB, allowing users to save, load, update, and delete arrow builds with complete tracking of components and configurations.

# Justification for Inclusion in ePortfolio

I selected this artifact because it showcases my full-stack development skills with a focus on backend data management and database operations. The enhancements demonstrate how I designed and implemented RESTful API endpoints in Express.js for creating, reading, updating, and deleting arrow build documents stored in a MongoDB database. Each build includes the user-defined name, the list of arrow components and their grain weights, GPI (grains per inch), arrow length, and the automatically calculated total grain weight.

In addition to standard CRUD operations, I used Mongoose to define and validate a schema for arrow builds in `ArrowBuild.js`. The application tracks `createdAt` timestamps for build history, and displays saved builds in a visually accessible layout. Users can load a build into the calculator UI to modify or delete it, reflecting robust read/update integration between the frontend and backend. These features demonstrate my ability to work with NoSQL data models, backend validation, RESTful routing, and client-server synchronization.

# Course Outcomes Addressed

This enhancement aligns with the following computer science program outcomes:  
  
- Demonstrating well-founded techniques and tools in computing practices to deliver value through persistent data solutions.  
- Designing and implementing backend APIs with database connectivity to enable secure and dynamic data storage.  
- Delivering a user-focused interface with seamless integration of persistent backend functionality.

The enhancement also complements earlier work in UI design and algorithm logic, making the application well-rounded. The original plan to showcase database interaction was fully realized, and the app now operates with a reliable backend that supports meaningful long-term data use.

# Reflection on the Enhancement Process

While enhancing the app for database integration, I learned to design MongoDB schemas that reflect user-driven data structures and efficiently support both reads and writes. One of the most challenging aspects was ensuring that frontend state and backend updates stayed in sync, especially when editing builds or calculating updated totals. Error handling and form validation became essential to ensure data consistency and prevent broken or malformed builds.

Overall, this enhancement significantly strengthened the app's utility and scalability. It deepened my experience in full-stack application development, RESTful API design, and NoSQL database management, all critical areas in real-world software engineering.

A screenshot of a computer

AI-generated content may be incorrect.