Pushpa Laxman

Southern New Hampshire University

CS-499-19649-M01 Computer Science Capstone 2025

7- Final Project: Enhancement Three: Databases

Neil Kalinowski  
06/22/2025

For my final project in CS 499, I will showcase my understanding and skill advancements related to the third category of the codebase, which pertains to Databases. This document serves as a review and enhancement of the CS 340 Client/Server development project artifact. This paper aims to convey the upgrades made to the database artifacts through a narrative approach. In preparing my ePortfolio, I will explain how the code operates, pinpoint areas for potential improvement, and detail the modifications made to align with the course objectives.

(CS 340 Project) Client/Server Development

A screenshot of a computer program

AI-generated content may be incorrect.

This artifact describes the need for a Rescue Web App by Grazioso Salvare, a company dedicated to training rescue animals. With this web application, we will create an interactive and operational dashboard for identifying and sorting available dogs based on data from Austin's animal shelters. The CS340 Client/Server development course provided students with the opportunity to conceptualize and develop a web application. This full-stack application utilizes the Model-View-Controller (MVC) development.

A screenshot of a computer program

AI-generated content may be incorrect.

The source code in this module is thoroughly documented and promotes clean coding through clear function and variable names, along with easily maintainable comments. This project performs various database operations using MongoDB, such as querying, retrieving results, writing and deleting data, and executing commands in the database.

A screenshot of a map

Description automatically generated

This artifact was driven by its nature as a full stack application, allowing to engage again with both front-end and back-end development, which will enhance my problem-solving abilities and refine my skills in building data-driven web applications that deliver a smooth and engaging user experience. I was intrigued by the holistic knowledge I gained by switching between front-end and back-end tasks, making informed decisions, and ensuring a cohesive development process that ensures seamless integrations between different application components. This project, CS 340, involved a multi-tier application utilizing a Model-View-Controller (MVC) architecture and a RESTful API to extend the HTTP protocol.

A screenshot of a computer

Description automatically generated

The current version of the application shows a reasonable level of modularity. CRUD operations are organized into separate modules rather than being combined within the main codebase, which improves readability and facilitates easier future maintenance. Multiple instances of this CRUD module are possible. This artifact refers to the development of practices concerning the validation of input data architecture and design, as well as implementing a default denial for attempts to access database records. It features precise, simple, and well-documented source code with a uniform commenting style, promoting clean coding practices through descriptive names for functions and variables. This application uses a CSV file to import existing dogs in shelters into MongoDB and imports dependencies like Python PyMongo driver, Python libraries Dash framework, and Python source code and CRUD modules to manipulate the data imported into MongoDB.

*A computer screen shot of a computer program

Description automatically generated*

*A screen shot of a computer program

Description automatically generated*

*A computer screen shot of a code

Description automatically generated*

*A computer screen with text and numbers

Description automatically generated*

*A computer screen with text and images

Description automatically generated*

From the screenshots you can view that the application features a user-friendly interface and a simple navigation system. The title of the dashboard and the Graziosos icon are displayed in red, directing users to the SNHU page. Users can choose from a range of buttons. They can filter the results by selecting one of four radio buttons according to Mr. Graziosos' preferred dog breeds. The initial four buttons include Water Rescue, Mountain Rescue, Disaster Rescue Team, and Reset. We have incorporated a data table, a pie chart, and a geolocation map for every shelter that provided data. A geo map illustrates the current location of the animal along with the accompanying data table.A screenshot of a computer program

Description automatically generated

I have set up an admin and user account in AAC animals so that the user can read and write within the documents. I have also ensured that the user can access the data via a MongoDB database with the appropriate user authentication (username and password). Also, I have provided authentication that involves creating both an Admin account and an aacuser account for the user.

*A computer screen shot of a program code

Description automatically generated*

*A computer screen with white text

Description automatically generated*

## Installation

* *A current version of Python to run both the .py and the .ipynb files MongoDB - to access the database.*
* *Jupyter Notebook is used to write and create ipynb files and run files to check the output of the files.*
* *Pymongo and Objectid are the libraries that I used.*
* *AnimalShelter was the CRUD Python I used.*
* *Install MongoDB: Provide a link to the official installation guide.*

*https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/*

* *Install Python: Follow the steps provided in the installation guide.*

*https://phoenixnap.com/kb/how-to-install-python-3-windows*

* *Install PyMongo: Use pip to install: pip install pymongo*

*https://pypi.org/project/pymongo/*

* *Install Jupyter Notebook: Use pip to install: pip install notebook*

*https://jupyter.org/install*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

While the existing system operates effectively, it can be improved by adding further database functionalities and a more streamlined data retrieval strategy. For this upgrade, I have utilized advanced features of MongoDB, such as implementing indexing on frequently accessed fields like age, animal ID, type, and color, which provides efficient methods for the database to swiftly find information without having to search through the entire table, thus facilitating significant horizontal scalability without requiring changes to the application logic. Furthermore, since we are working with large datasets, we will need to execute several transformations within a single operation, such as filtering, grouping, sorting, reshaping, and computing new fields, which can be quite complex. Therefore, I have employed aggregation pipelines to manage data flowing through the pipeline for animals, concentrating on particular attributes like age, breed, and more, while generating reports on the animals in the shelter.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Using the screenshots above, you can see that the MongoDB database has been created and that all the records have been stored. The data structure implemented is programmatic, allowing stored variable values to be used efficiently in web applications and functions. It improves the design and evaluation of computing solutions by using mathematical principles and computer science practices and standards appropriate to the task. By using mathematical principles and computer science practices and standards appropriate to the task, this approach improves the design and evaluation of computing solutions that solve given problems.

A screen shot of a computer code

AI-generated content may be incorrect.

In the above example of a collection of "animals". $match stage filters the documents by the value in class field i.e. animal\_type: "Dog" in the first stage and passes the document to the second stage. In the Second Stage, the $group stage groups the documents by the breed field to calculate the sum of Dog breeds. In the third stage, the $ sort stage sorts the documents based on field values. Used toarray () to return an array with all the documents from a cursor.

Here, the aggregate () function is used to perform aggregation. It can have three stage operators, an expression and an accumulator. These operators work together to achieve the final desired outcome.

A screenshot of a computer

AI-generated content may be incorrect.

Managing a database requires speed and efficiency. As applications become more complex and handle greater amounts of data, the efficiency of database queries becomes increasingly critical. Implementing indexes is among the most effective methods to enhance query speed. Utilizing indexes is one of the optimal strategies for accelerating query performance.

I used the command createIndex for the breed field and checked the query planner using explain function

db.animals.createIndex({"breed": 1})

db.animals.find({"breed": "Domestic Shorthair Mix"}).explain()

A screenshot of a computer program

Description automatically generated

I used command createIndex for breed, outcome\_type, transfer. Also checked using the explain command with execution stats.

db.animals.createIndex({"breed": 1, "outcome\_type": 1, "Transfer": 1})

A computer screen shot of a program code

Description automatically generated

There are several factors that determine how quickly MongoDB aggregates data, such as the complexity of the aggregation pipeline, the size of the data set, the hardware specifications of the MongoDB server, and the efficiency of the indexes.

A screenshot of a computer screen

AI-generated content may be incorrect.

MongoDB's aggregation pipeline is a powerful way to process and transform data within the database, avoiding the need to fetch large datasets to the application layer. It works by defining a series of stages that process documents, where the output of one stage becomes the input of the next. Each stage performs an operation, such as filtering, grouping, projecting, or transforming data.

The planned enhancement for the CS 340 project aligns with the following course outcomes

 [Course Outcomes: 3]: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices:

It is important to choose the right structure and algorithms when designing a database so that data can be stored and retrieved efficiently while balancing competing demands, like speed, space, and complexity. I have designed and evaluated computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices by carefully considering data volumes, types of data to be stored, their relationships, and the operations that will be performed on them. As part of the proposed improvement, indexing and aggregation have been utilized, as well as leveraging algorithmic principles to enhance performance, which corresponds to CS 340 learning outcomes, particularly with regards to database optimization, data retrieval, and system scalability. By implementing MongoDB indexes and aggregation pipelines, I demonstrated my ability to enhance database performance. To enhance query efficiency, it is essential to establish an index on the animal ID field. Database indexing is a common practice for boosting database performance, which aligns perfectly with the course objective of employing innovative and well-established computing techniques. Adaptable and efficient data retrieval methods reflect my understanding of performance optimization and user-centric queries.

[Course Outcomes: 4]: Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals:

A strong understanding of fundamental computing concepts is essential for proving the capacity to provide value in database solutions, as well as the capability to utilize new technologies and adhere to security best practices. In my enhancement, I used MongoDB's indexing and aggregation capabilities to enhance performance and functionality. With the new web application, animal rescue data can now be processed in real-time, providing valuable and useful solutions. By using MongoDB's aggregation framework, I demonstrated my capability to manipulate complex data by producing statistics, such as the number of animals by breed type. The ability to craft and assess algorithms-based computing solutions is necessary for this skill. Through aggregation, data can be transformed into intricate ways, such as filtered, grouped, sorted, and projected, facilitating analysis and reporting. As we are dealing with large datasets, it is essential that we optimize queries and reduce data redundancy to improve performance while maintaining data integrity to ensure accuracy and reliability.

[Course Outcome 5]: "Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities."

To consider the integrity of applications and the reliability of systems, it is crucial to have a secure mindset. It is crucial to validate data in databases to maintain data accuracy, integrity, and reliability, ensuring that the data is reliable and trustworthy.  By preventing errors, inconsistencies, and inaccuracies, it helps ensure a successful project. My goal is to mitigate potential security threats by implementing input validation, such as preventing incorrect data entry, while also ensuring the system's integrity and safety. Also, I ensure that I monitor the application performance, and handle downtime as well as exceptions and issues as soon as possible. Hence, by implementing input validation and exception handling, I have enhanced data security, ensured the protection of account information, while also maintained accessibility, making the app more user-friendly for everyone. With these enhancements, I have developed a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities.

**Reflection, Challenges and Lessons Learned:**

Throughout this project, I had the opportunity to engage in both front-end and back-end development, which refined my problem-solving skills and improved my capacity to build user-friendly and interactive data-driven web applications. The integration of MongoDB with Jupyter Notebooks facilitated a dynamic and collaborative development environment, allowing for easy experimentation between the front-end and back-end. By utilizing Python libraries like PyMongo, data from MongoDB can be effortlessly incorporated into Jupyter Notebooks. The processes of data modeling, querying, and optimization presented significant challenges, but they yielded invaluable lessons learned, leading to improved skills and a deeper understanding of the subject matter.

The original web app was developed on Apporto, a university remote desktop Linux platform; therefore, the artifact was recreated in a Windows OS environment. Since the original code was set up in the Linux environment, it was a challenge to develop the web application. To use a Jupyter Notebook for Dash applications, I used the main Dash package directly, as JupyterDash is deprecated. I have enhanced my skills in creating, developing, and presenting professional written and visual communication materials that are clear, technically accurate, and tailored to specific audiences and contexts.

This project allowed me to explore front-end technologies, including HTML, CSS, and JavaScript, as well as back-end technologies such as Node.js and various server-side languages, alongside fundamental web development concepts. This course established a strong foundation for full-stack web development, utilizing widely used technologies. This artifact utilizes industry-standard Python code best practices, which enhances readability by formatting the code according to the relevant coding standards. Line breaks are included, and the source code is organized effectively, uniformly, and consistently. With the new enhancements, I have implemented valuable and innovative computer solutions that align with industry-specific goals.

This study introduced me to MongoDB's Aggregation, a robust framework for transforming, calculating, and analyzing groups of documents. The aggregation pipeline allows us to gather significant insights by grouping, filtering, sorting, reshaping, and calculating data. As a result of this course, I have gained a deeper understanding of how data analysis works, as well as a greater ability to utilize data analysis tools effectively. The project also emphasized the importance of understanding how data is structured and stored in databases, enabling data scientists to make informed decisions and perform effective analysis. The course taught me to analyze database criteria and specifications and to choose the best data model based on your needs, such as relational, hierarchical, network, or document-based models. In addition to being part of my portfolio, this project will be helpful when I seek employment or attend job interviews. The skills, insights, and experiences I acquired from this course have contributed to the attainment of my personal and professional objectives.

**References**

*GeeksforGeeks. (2025a, February 3). Aggregation in mongodb. https://www.geeksforgeeks.org/aggregation-in-mongodb/*

*Mydbops, & Author, A. the. (2025, May 9). MongoDB aggregation: Beginner’s Guide with examples. RSS. https://www.mydbops.com/blog/mongodb-aggregation-guide*