Pushpa Laxman

Southern New Hampshire University

CS-499-19649-M01 Computer Science Capstone 2025

5-2 Milestone Four: Enhancement Three: Databases

Neil Kalinowski  
06/08/2025

Q. Briefly describe the artifact. What is it? When was it created?

My name is Pushpa Laxman, and I will evaluate the code for Milestone Four as a part of my final project, CS 499. This document provides a review and enhancement of an artifact CS 340 Client/ Server development project that demonstrates my knowledge and skill enhancements to the third category of the codebase, which covers Databases. This paper aims to illustrate the improvements made to database artifacts with a narrative. I will outline how the code functions, identify potential areas for improvement and elaborate on how it has been modified to fulfill the course objectives and prepare it for my ePortfolio.

Selected Artifact: animal\_main.py, animal\_module.py

(CS 340 Project) Client/Server Development

A screenshot of a computer program

AI-generated content may be incorrect.

Q. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?

The chosen artifact for the databases category showcases Grazioso Salvare, a company dedicated to training rescue animals that focuses on finding dogs suitable for search-and-rescue training programs and is looking for a Rescue Web App. The goal of this web application is to develop an interactive and fully operational dashboard that can identify and sort available dogs based on existing data from animal shelters in Austin. A web application was conceptualized and created during the CS340 Client/Server development course. The application use Model, View, and Controller development patterns for this full-stack application.

Model: This layer represents the underlying data/database used in development -View: This layer represents the user-facing visual representation. Communication between the user and the data requested is handled by the user interface (UI). This project uses MongoDB as its database. The view layer is built with Dash Plotly, while the controller layer utilizes Python with Pymongo, relying on Dash Leaflet. A variety of database operations are performed through this application, including querying, retrieving results, writing and deleting data, and running database commands.

A screenshot of a computer program

AI-generated content may be incorrect.

Before this term, I had a course CS 465 Full stack develoopment. The project was to create a Travlr website Using the Model-View-Controller (MVC) approach. The application uses MEAN stack architecture, with MongoDB as a database, Express as a web framework, Angular as a front-end framework, and Node.js as a web server. During this project, I had the opportunity to learn about the core principles of web development, including front-end technologies like HTML, CSS, and JavaScript, and back-end technologies like Node.js and server-side languages. This course provided a solid foundation in full-stack web development using a popular and efficient set of technologies.

A screenshot of a map

Description automatically generated

The choice to select this artifact was driven by its nature as a full stack application, allowing me 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. Upon running the admin app file in the computer terminal, a new window opens in the internet browser with the address of the corresponding app (e.g. http:/127.0.1:31572). A dashboard for the client-facing web application appears as soon as the browser loads.

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*

With its user-friendly interface and intuitive navigation, this application is easy to use. The Graziosos icon appears red in the application, linking to the SNHU page and displaying the dashboard title. An array of buttons can be selected by the user. Four radio buttons allow users to filter results based on Mr. Graziosos' preferences for the breed of dog. Water Rescue, Mountain Rescue, Disaster Rescue Team, and Reset are the first four buttons, as you can see. For each of the five shelters that contributed information, you will find a data table, pie chart, and geolocation chart. Here's a pie chart showing how many of each breed are used in each filter. The data table is accompanied by a geo map that shows the animal's current location.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

However, some areas needed improvement:

**Limitations**: Although the original system functions well, it can be enhanced with additional database features and a more efficient data access strategy. For this enhancement, I will incorporate advanced MongoDB features such as:

**Plan**: Implementing indexing to frequently queried fields like age, animal ID, type, and color offers efficient methods that allow the database to swiftly find data without needing to search through the entire table, facilitating significant horizontal scalability without making adjustments to application logic.

**Limitation**: Since we are dealing with large datasets, we must perform multiple transformations within a single operation, such as filtering, grouping, sorting, reshaping, and calculating new fields, which is a challenging job.

**Plan**: In my plan, I intend to use aggregation pipelines to handle the data flowing through the pipeline for animals, focusing on specific characteristics such as age, breed, and more, and producing reports on the animals in the shelter.

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.

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.

In the above screenshots, you can see the database has been created and all the records have been stored in the MongoDB database. I have also used the find command for the animal type as dog with the limit of 1. The count function is getting the total records as 1000. The implemented data structure is programmatic, where stored variable values can be used efficiently in different functions and callbacks through web applications. 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.

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.

Q. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?.

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:

As part of the proposed improvement, indexing and aggregation will be 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. Optimizing query performance requires creating an index on the animal ID field. Database indexing is a widely accepted method of increasing database effectiveness, which perfectly aligns with the course outcome of using innovative and well-founded computing methods.

Adaptable and efficient data retrieval methods reflect my understanding of performance optimization and user-centric queries. Developing computing solutions that navigate the tradeoffs involved in design decisions supports the course outcome.

[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:

In my enhancement, I use 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 in intricate ways, such as filtered, grouped, sorted, and projected, facilitating analysis and reporting.

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

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.

Q. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

Combining MongoDB with Jupyter Notebooks enabled flexible and interactive development, enabling easy collaboration and experimentation. Using Python libraries like PyMongo, Jupyter Notebooks seamlessly integrate with MongoDB data. Modeling, querying, and optimizing data were challenging processes, but valuable lessons were learned, resulting in enhanced skills and a deeper understanding in these areas.

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 must use 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 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, implement valuable and innovative computer solutions that align with industry-specific goals.

In the course of this study, I discovered Aggregation in MongoDB, which serves as a powerful framework for executing complex data transformations, calculations, and analyses on groups of documents. By utilizing the aggregation pipeline, we can derive significant insights through grouping, filtering, sorting, reshaping, and calculating data.

My understanding of the data analysis process has been enhanced by this course, as well as my ability to utilize these tools effectively. This project will also be beneficial when I seek employment or attend interviews, in addition to being part of my portfolio. Through this work, I have gained knowledge, skills, and experience, enabling me to meet my personal and professional objectives.

**References**

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