### Jean Agnus CS 499 Milestone Two Narrative: Software Design and Engineering Enhancement

#### **1. Description of the Artifact**

The artifact I selected for enhancement is a dashboard web application developed using the Dash framework and MongoDB. This artifact originated from my CS 340 course, where I created a basic web interface to visualize animal shelter data stored in a MongoDB collection. The original version was a single dashboard.py file that loaded data from MongoDB and displayed it in a static table using Dash. At that stage, the application lacked interactive features such as inserting, updating, or deleting animal records from the interface. Additionally, all logic—UI, data retrieval, and database manipulation—was tightly coupled, making the code harder to scale or maintain.

#### **2. Justification for Inclusion in My ePortfolio**

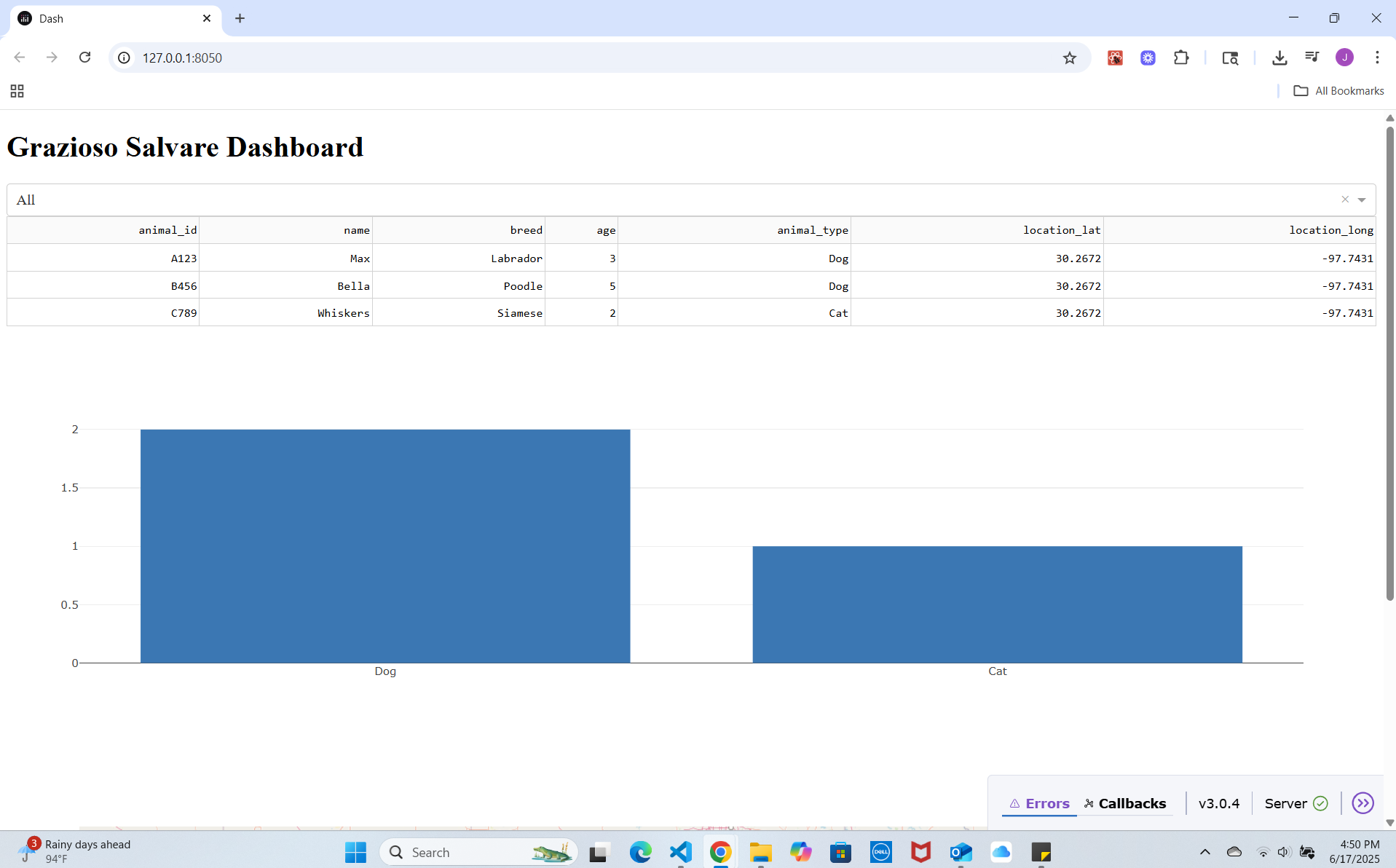
I chose this artifact because it demonstrates my understanding of full-stack web application development, my ability to integrate a front-end (Dash) with a back-end database (MongoDB), and my skills in refactoring and modularization. This artifact is now a stronger representation of my ability to follow software engineering principles, such as **separation of concerns**, **modularity**, and **user input handling**. Enhancing this dashboard shows my ability to solve real-world problems using clean, maintainable, and scalable code.

#### **3. Improvements Made to the Artifact**

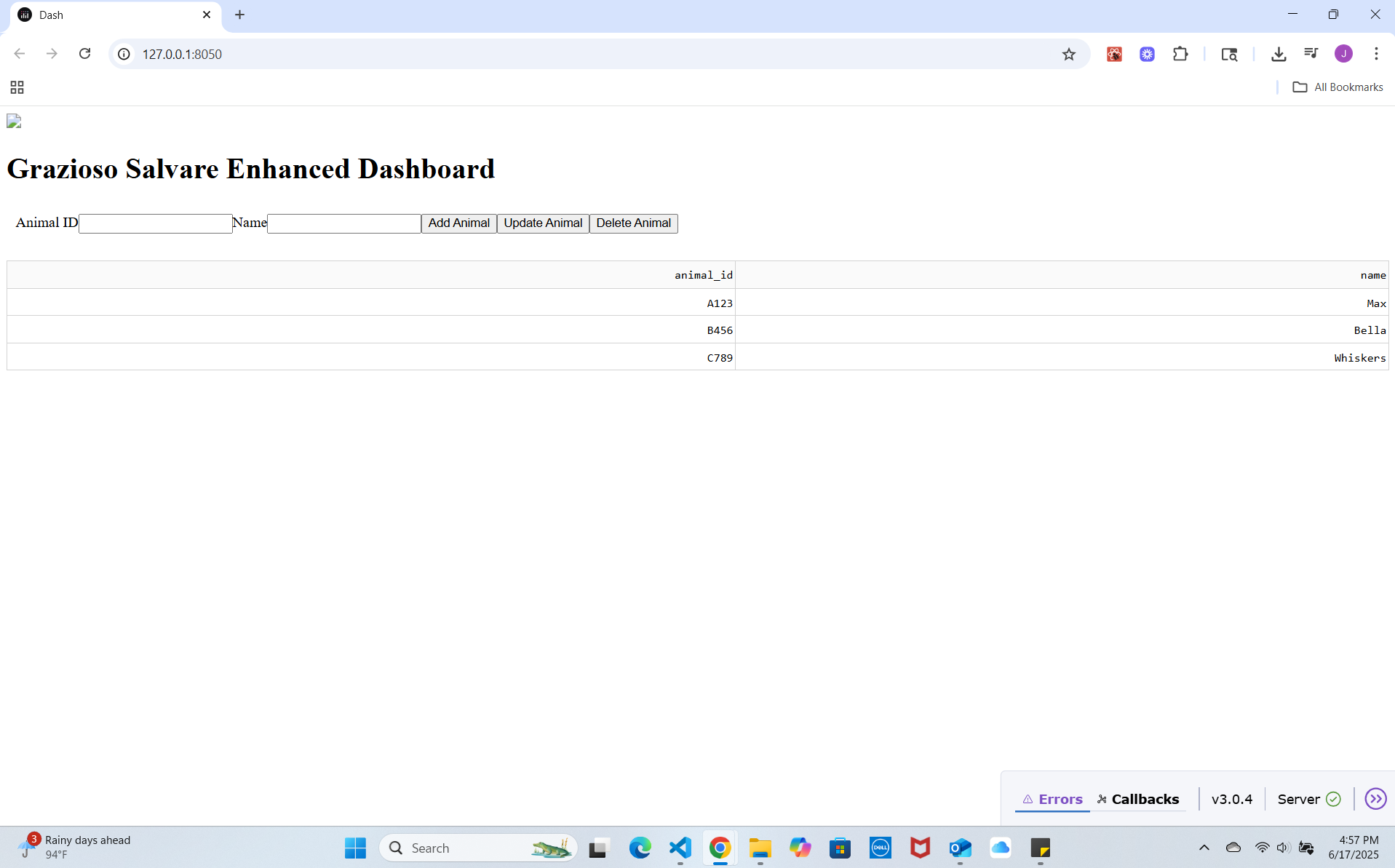
For this milestone, I significantly improved the dashboard in the following ways:

* **Refactored the codebase** to separate database logic from UI code by creating a crud\_module.py file that contains functions for reading, inserting, updating, and deleting records from MongoDB.
* **Improved modularity** by importing those functions into the main dashboard.py, reducing code duplication and improving clarity.
* **Added form elements** to allow users to insert new animal records via input fields and an “Add Animal” button.
* **Added editing functionality**, enabling users to select a record from the table, modify it, and update the database in real time.
* **Implemented delete functionality** with success feedback messages.
* **Improved input validation** and error handling for a more reliable user experience.
* Documented “before” and “after” screenshots to demonstrate visual and functional improvements.

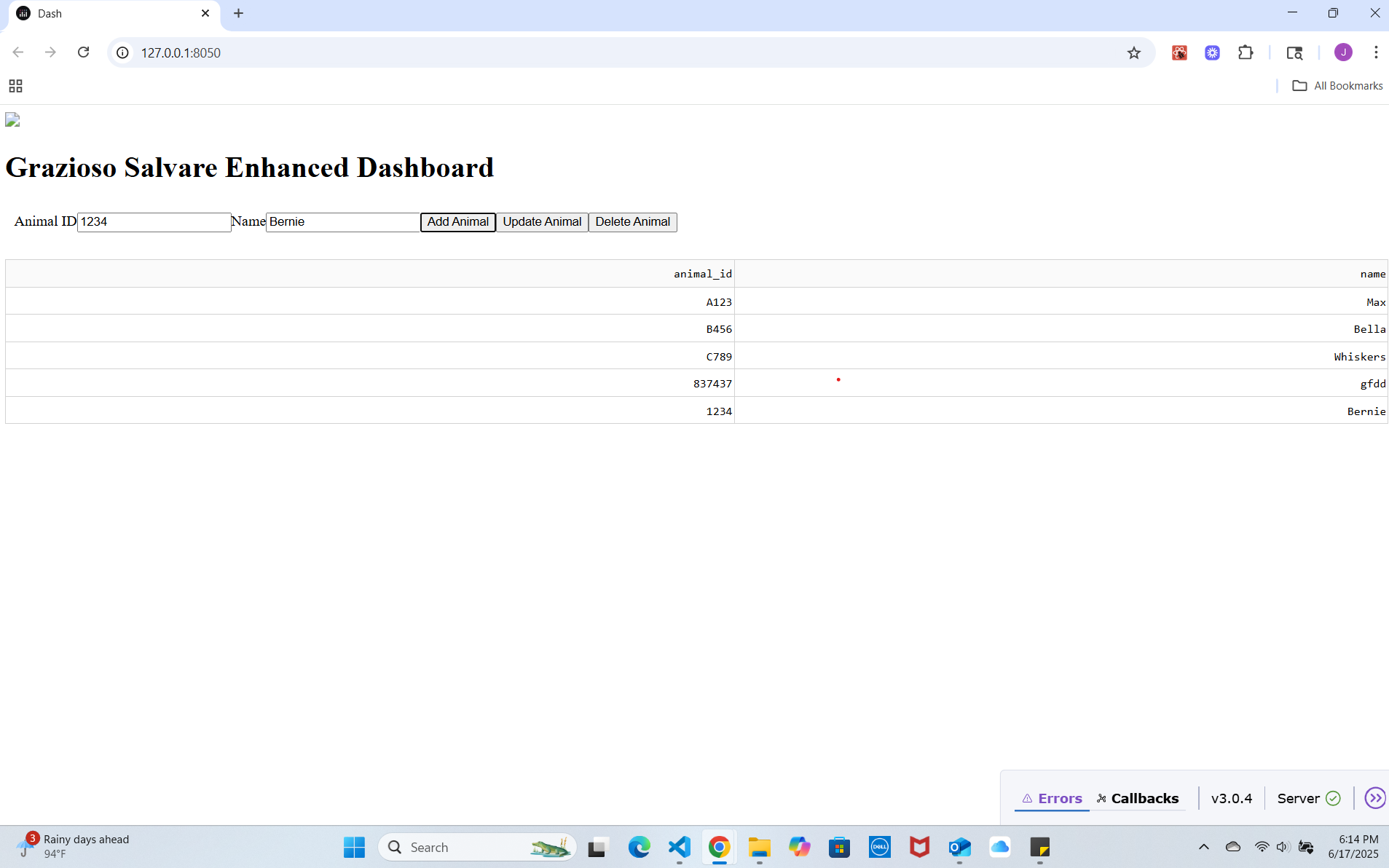
Before Dashboard



After add\_form



After add\_success



#### **4. Course Outcome Alignment**

This enhancement aligns with the following **CS 499 Course Outcome**:

**“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 (Software Design/Engineering/Databases).”**

By refactoring my code to follow software engineering best practices, I’ve demonstrated clean coding principles, modular structure, and user-centered design. The interactive CRUD dashboard delivers value for managing shelter data efficiently.

#### **5. Reflections on the Process**

Enhancing this artifact was both challenging and rewarding. The biggest challenge was learning how to separate concerns between UI and data logic in Dash, which does not follow traditional MVC patterns. I also had to think carefully about user flow—how a user would add, edit, or delete animals—and how to reflect those changes dynamically in the DataTable. I gained deeper insight into **callback chaining**, **state management in Dash**, and **using pymongo more effectively**.

What I’m most proud of is turning a static read-only dashboard into a fully functional web application. I now feel much more confident in my ability to design scalable Python-based apps that connect to real-world databases and support CRUD operations. These are key skills that I will apply in my future career as a software or AI engineer.