Sergio Irianda

CS 499

Enhancement Three

Artifact Overview

The artifact I selected for enhancement is a MongoDB CRUD module originally developed in CS 340 – Client/Server Development. The module used PyMongo to support basic Create, Read, Update, and Delete operations through an AnimalShelter class. For this enhancement, I wrapped the CRUD logic in a RESTful Flask API, enabling web-based interaction and integrating it into a full-stack architecture.

Why This Artifact Was Selected

I chose this artifact for my ePortfolio because it demonstrates backend development, database connectivity, and API design which are all essential skills in modern software engineering. By building a RESTful API layer on top of the original class, I extended the functionality to allow client-side interaction.

Enhancement Description

- Built a RESTful Flask API to expose CRUD functionality over HTTP.
- Added endpoints for GET, POST, PUT, and DELETE operations on animal records.
- Implemented a CSV import method to batch-load animal data into the MongoDB collection.

Outcome Alignment and Improvements

This enhancement aligns with the course outcomes related to algorithmic thinking and innovative problem-solving:

"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."

Reflection on the Enhancement Process

Implementing this data structure in a mobile environment taught me several key skills:

- How to structure Flask routes to follow REST conventions.
- How to test endpoints and validate behavior using Postman and Compass.

One challenge I faced was switching from a preconfigured cloud MongoDB connection (used in the original Apporto lab environment) to a fully local MongoDB Compass instance. I had to set up MongoDB locally and verify that the Flask app could still authenticate and perform operations.