# CS 340 README PROJECT TWO

## About the Project/Project Title

*This project completes the second phase of the full-stack application developed for Grazioso Salvare, an organization that identifies and trains dogs for search-and-rescue missions.*

*In Project Two, a fully functional interactive dashboard was created using Python, Dash, and MongoDB in the Codio cloud environment. The dashboard allows users to visualize, filter, and explore animal data from the Austin Animal Center database. It connects directly to the CRUD Python module from Project One and provides a graphical client interface that supports real-time data retrieval and filtering based on rescue types.*

## Motivation

*Grazioso Salvare needed a visual, data-driven tool to help staff quickly identify dogs suitable for different types of rescue training. The motivation behind this phase was to extend the secure backend developed in Project One into a user-friendly dashboard that could present the data graphically.*

*The result is an intuitive interface that enables filtering, mapping, and visual analysis, reducing user error and improving decision-making when evaluating shelter animals.*

## Getting Started

*Before running the dashboard, ensure that MongoDB is active and that the aac database and animals collection are populated with the Austin Animal Center dataset.*

*mongoimport --type=csv --headerline --db aac --collection animals --drop ./datasets/aac\_shelter\_outcomes.csv*

*Create a database user in mongosh:*

*use admin*

*db.createUser({*

*user: "aacuser",*

*pwd: "SNHU1234",*

*roles: [{ role: "readWrite", db: "aac" }]*

*})*

*Verify project files in Codio:*

*CRUD\_Python\_Module.py*

*ProjectTwoDashboard.ipynb*

*Grazioso\_Salvare\_Logo.png*

*Open Codio > Jupyter Lab > ProjectTwoDashboard.ipynb*

*Run all cells; the dashboard will launch on your proxy URL (Dash app running on https://lakeuranium-culturesaga-3000.codio.io/proxy/8050/)*

## Installation

*The following tools are required to run the project:*

* ***Codio: Primary cloud-based IDE used to develop, run, and test all Python and MongoDB components.***
* ***Python 3.x: Core programming language for CRUD logic and Dash framework.***
* ***MongoDB Community Server 7.x: Database engine storing Austin Animal Center data.***
* ***PyMongo: MongoDB driver enabling Python-to-database connectivity.***
* ***Jupyter Lab (in Codio): Interactive environment for running and testing the dashboard.***

## Usage

### *Model: MongoDB database accessed via the AnimalShelter class in the CRUD module.*

### *View: Dash components (interactive data table, bar chart, and geolocation map).*

### *Controller: Callback functions linking user input to query updates and widget outputs.*

### Code Example

*from CRUD\_Python\_Module import AnimalShelter*

*db = AnimalShelter("aacuser", "SNHU1234")*

*# Example: Retrieve dogs for Water Rescue training*

*query = {*

*"animal\_type": "Dog",*

*"breed": {"$in": [*

*"Labrador Retriever Mix",*

*"Chesapeake Bay Retriever",*

*"Newfoundland"*

*]},*

*"sex\_upon\_outcome": "Intact Female",*

*"age\_upon\_outcome\_in\_weeks": {"$gte": 26, "$lte": 156}*

*}*

*results = db.read(query)*

*print("Dogs suitable for Water Rescue:", len(results))*

### Tests

*All CRUD functions were tested in the Jupyter Notebook file ProjectOneTestScript.ipynb.*

*Each operation produced the expected output:*

Create -> True

Read -> found 1 document(s)

Update -> modified 1

Delete -> removed 1

### Screenshots

*A screenshot of a computer

AI-generated content may be incorrect.*

*A white sheet with black text

AI-generated content may be incorrect.*

*A screenshot of a computer

AI-generated content may be incorrect.*

## Contact

Dusty Cook