7-2 Project Two:

README SUBMISSION

*Christopher Davidson | CS-340 Client/Server Development | 6/22/25*

Grazioso Salvare – AAC Dashboard

About the Project:

This Python Dash web-app lets Grazioso Salvare explore the Austin Animal Center (AAC) outcomes collection and quickly identify adoptable dogs that meet different rescue-team profiles (water, mountain, disaster).

Key widgets

* Interactive Data Table
  + Sort, filter, search, single-row select
* Leaflet Map
  + Pin drops to the selected animal’s found location
* Donut Chart
  + Live outcome-type distribution for current table view
* Age-vs-Outcome Scatter
  + Correlates age (weeks) with outcome types

All data are pulled live from MongoDB through a reusable CRUD module, animal\_shelter.py no CSV exports or manual refreshing are required.

Motivation

Field coordinators asked for a one-stop dashboard that

* Displays all 10 000 + AAC outcome records,
* Filters to each rescue profile with one click, and
* Surfaces location and outcome insights visually.

Duplicating ad-hoc Mongo queries in every notebook was error-prone, so I wrapped the database access in a tiny, testable module and built the Dash front end on top of it.

Getting Started

* Clone repo
  + git clone <your\_git\_url> cs340\_project2 && cd cs340\_project2
* Set environment variables \*once\* per session
  + export MONGO\_USER=root
  + export MONGO\_PASS=iZHac1gYyP
  + export MONGO\_HOST=nv-desktop-services.apporto.com
  + export MONGO\_PORT=31373
* (optional) create / activate virtual environment
  + python -m venv venv && source venv/bin/activate
* Install dependencies
  + pip install -r requirements.txt
* Launch the dashboard
  + jupyter lab dashboard.ipynb

The Dash app starts at <http://127.0.0.1:8050>

Installation:

* MongoDB 6
  + Document store holding \*AAC.animals\*
  + Provided in Apporto
* PyMongo 4.6
  + Official driver; async-safe & well-maintained
  + pip install pymongo`
* Dash 2.16
  + Lightweight web framework for data apps
  + `pip install dash`
* Dash-Leaflet
  + Open-source Mapbox/Leaflet component
  + `pip install dash-leaflet`
* Plotly Express
  + Quick, interactive charts
  + `pip install plotly`

Usage

from animal\_shelter import AnimalShelter

shelter = AnimalShelter(

user="root",

password="iZHac1gYyP",

host="nv-desktop-services.apporto.com",

port=31373,

db\_name="AAC",

col\_name="animals")

#Example: count all dogs

dogs = shelter.read({"animal\_type": "Dog"})

print(len(dogs), "dogs in database")

#Example: wilderness-rescue query (German Shepherd & friends)

query = {

"animal\_type": "Dog",

"breed": {

"$in": [

"German Shepherd", "Alaskan Malamute",

"Old English Sheepdog", "Siberian Husky", "Rottweiler"

]},

"sex\_upon\_outcome": "Intact Male",

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

}

for doc in shelter.read(query, {"\_id": 0, "animal\_id": 1, "breed": 1}):

print(doc)

Tests

The unit tests live in tests/ and are run with pytest:

pytest -q

\*\*Expected output:\*\*

4 passed in 1.15s

test\_create.py – verifies a document can be inserted

test\_read.py – verifies filtered reads return correct keys

test\_update\_delete.py – ensures update & delete report correct counts

Screenshots

A screenshot of a computer

AI-generated content may be incorrect.Default view

A screenshot of a computer

AI-generated content may be incorrect.Filtered by Bird

A screenshot of a computer

AI-generated content may be incorrect.Filtered by Mountain-rescue

For questions, reach me at christopher.davidson@snhu.edu.