

Grazioso Salvare Rescue Dog Dashboard

README & Project Documentation

Jeremy Ritchie | CS-340 Client/Server Development | SNHU

Author	Jeremy Ritchie
Course	CS-340: Client/Server Development
Institution	Southern New Hampshire University (SNHU)
Client	Grazioso Salvare (via Global Rain)
Stack	Python, Dash, MongoDB, Plotly, Dash Leaflet
Database	MongoDB 7.0 — aac.animals collection (20,000 records)

1. Project Overview

Grazioso Salvare is an international search-and-rescue training company that identifies dogs suitable for specialized rescue training. This project delivers a full-stack web dashboard that connects to a MongoDB database of Austin Animal Center shelter outcomes and allows Grazioso Salvare staff to interactively filter, explore, and visualize dog candidates by rescue type.

The application was built using the MVC (Model-View-Controller) architectural pattern:

- Model: MongoDB database accessed via a custom Python CRUD module (AnimalShelter class)
- View: Dash web framework rendering an interactive DataTable, pie chart, and geolocation map
- Controller: Dash callback functions linking user filter selections to data updates across all widgets

2. Required Functionality

2.1 Dashboard Branding

- Grazioso Salvare logo displayed at the top, wrapped in a hyperlink to www.snhu.edu
- Unique developer identifier: "Developed by Jeremy Ritchie | CS-340 | SNHU"

2.2 Interactive Filter Options

Radio button controls allow filtering of all dashboard widgets by the four required rescue types:

- Water Rescue — Labrador Retriever Mix, Chesapeake Bay Retriever, Newfoundland; Intact Female; 26–156 weeks
- Mountain or Wilderness Rescue — German Shepherd, Alaskan Malamute, Old English Sheepdog, Siberian Husky, Rottweiler; Intact Male; 26–156 weeks
- Disaster or Individual Tracking — Doberman Pinscher, German Shepherd, Golden Retriever, Bloodhound, Rottweiler; Intact Male; 20–300 weeks
- Reset — returns all widgets to their original unfiltered state showing all 20,000 records

2.3 Interactive Data Table

- Displays Austin Animal Center Outcomes data loaded from MongoDB
- Dynamically updates when a rescue type filter is selected
- Features: pagination (10 rows/page), native column sorting, native column filtering, single-row selection, column highlight on click

2.4 Charts

- Pie chart: displays breed distribution of the currently filtered dataset; updates dynamically with filters
- Geolocation map: renders an interactive Leaflet map centered on the selected row's GPS coordinates; shows animal name and breed in a popup marker

3. Dashboard Screenshots

Note: The screenshots below were taken during testing and deployment in the Codio/JupyterLab environment. Each screenshot includes the Grazioso Salvare logo and the developer unique identifier as required by the specification.

Screenshot 1 — Starting State (Reset / Unfiltered)

Shows the full unfiltered dataset (20,000 records across 2,000 pages). The pie chart displays all breeds and the map is centered on Austin, TX.


Grazioso Salvare Rescue Dog Dashboard
Designed by Jersey Eclipse - CS-140 - 2008

Water Rescue Mountain or Wilderness Rescue Disaster or Individual Tracking Reset

ID	res_name	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	dateline	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
1		3 years	A748874	Cat	Domestic Shorthair Mix	Black/White	2014-04-08	2017-04-11 09:00:00	2017-04-11T09:00:00	SOP	Transfer	Neutered Male	Spayed Female	38.5966578739405	-97.3408780721188	156.767873242857
2		3 years	A730324	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-03-12 11:41:00	2016-03-12T11:41:00	Blessing	Adoption	Spayed Female	Spayed Female	38.307046899411	-97.3064330712175	157.504674603187
3		2 months	A656298	Cat	Domestic Shorthair Mix	Tortie	2013-09-05	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor	Adoption	Spayed Female	Spayed Female	38.752316838448	-97.418252918847	14.8989673815875
4		2 years	A718338	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 10:43:00	2015-12-28T10:43:00	Frank	Adoption	Neutered Male	Spayed Female	38.7597518216248	-97.5152750871131	110.314487380785
5		1 year	A656843	Dog	PIT Bull Mix	Brown/White	2013-06-09	2016-04-18 17:24:00	2016-04-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	38.452554397366	-97.4742040292	62.266285742826
6		1 year	A721399	Dog	Dachshund Shorthair Mix	Tan/White	2013-02-23	2016-04-17 17:45:00	2016-04-17T17:45:00	Belle	Adoption	Spayed Female	Spayed Female	38.452554397366	-97.351204751544	52.266285742826
7		3 years	A748868	Cat	Domestic Shorthair Mix	Brown/Tan	2013-04-13	2015-04-17 15:24:00	2015-04-17T15:24:00	Nyla	Return to Owner	Spayed Female	Spayed Female	38.452554397366	-97.351204751544	52.266285742826
8		3 years	A720318	Dog	PIT Bull Mix	Red/White	2013-09-20	2016-07-17 17:12:00	2016-07-17T17:12:00	Person	Partner	Transfer	Neutered Male	38.452554397366	-97.351204751544	148.020454380785
9		2 years	A742387	Dog	Border Collie/Border Collie Mix	Brown/Brindle/White	2015-09-18	2017-03-11 12:36:00	2017-03-11T12:36:00	*Kodi	Adoption	Neutered Male	Spayed Female	38.471148640990	-97.3607780473797	107.91247419408
10		5 years	A721742	Dog	Miniature Schnauzer Mix	Black/White	2012-04-05	2016-04-18 17:27:00	2016-04-18T17:27:00	Gretchen	Adoption	Spayed Female	Spayed Female	38.4733884867566	-97.40883367991	261.812547310985

Rescue Candidate Breeds

Map showing rescue locations across a geographic area with various rescue stations marked.

Screenshot 2 — Water Rescue Filter Applied

Filter returns 34 records matching Labrador Retriever Mix, Chesapeake Bay Retriever, and Newfoundland breeds; Intact Female; age 26–156 weeks. Pie chart and map update accordingly.


Grazioso Salvare Rescue Dog Dashboard
Designed by Jersey Eclipse - CS-140 - 2008

Water Rescue Mountain or Wilderness Rescue Disaster or Individual Tracking Reset

ID	res_name	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	dateline	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
36		6 months	A709953	Dog	Labrador Retriever Mix	Yellow	2014-12-19	2015-07-06 21:13:00	2015-07-07T06:21:30	Catalina	Medical	Euthanized	Intact Female	38.548886386333	-97.269909895957	38.354886386333
732		2 years	A716982	Dog	Labrador Retriever Mix	Tan/White	2015-05-19	2017-07-21 14:15:00	2017-07-21T14:15:00	Catalina	Return to Owner	Intact Female	Intact Female	38.631830367057	-97.371240476705	114.091860797153
1125		1 year	A717058	Dog	Labrador Retriever Mix	White/Black	2016-08-30	2017-08-31 14:15:00	2017-08-31T14:15:00	Pirate	Return to Owner	Intact Female	Intact Female	38.557236097907	-97.351224228787	52.370288052385
1618		9 months	A716471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17	2016-10-23 17:11:00	2016-10-23T17:11:00	Mika	Adoption	Intact Female	Spayed Female	38.750924803824	-97.730248070564	48.245173815875
1757		7 months	A721767	Dog	Labrador Retriever Mix	Black	2016-06-27	2017-04-01 15:20:00	2017-04-01T15:20:00	Marsay	Return to Owner	Intact Female	Spayed Female	38.488974337324	-97.428867517538	33.234262088227
1898		1 year	A712781	Dog	Labrador Retriever Mix	Black/White	2016-11-27	2017-10-01 13:00:00	2017-10-01T13:00:00	Person	Partner	Transfer	Intact Female	38.288481163013	-97.40883367991	53.478274805138
2842		2 years	A702745	Dog	Labrador Retriever Mix	Black	2013-05-22	2015-09-12 11:00:00	2015-09-12T11:00:00	Abigail	Return to Owner	Intact Female	Intact Female	38.751794295193	-97.412166487972	104.355047431095
2225		2 years	A715341	Dog	Labrador Retriever Mix	Black/White	2015-09-29	2017-09-01 12:37:00	2017-09-01T12:37:00	Abigail	Return to Owner	Intact Female	Intact Female	38.3834245796497	-97.73722770861	109.474974242857
3355		9 months	A671748	Dog	Labrador Retriever Mix	Yellow	2015-12-29	2016-09-01 17:01:00	2016-09-01T17:01:00	Suffering	Euthanized	Intact Female	Intact Female	38.720753488489	-97.391294389924	39.244164821368
4222		1 year	A715553	Dog	Labrador Retriever Mix	Black	2016-09-25	2016-09-27 14:16:00	2016-09-27T14:16:00	Delay	Return to Owner	Intact Female	Intact Female	38.532876805625	-97.40883367991	52.0576805625

Rescue Candidate Breeds

Map showing rescue locations across a geographic area with various rescue stations marked.

Screenshot 3 — Mountain or Wilderness Rescue Filter Applied

Filter returns records matching German Shepherd, Alaskan Malamute, Old English Sheepdog, Siberian Husky, and Rottweiler breeds; Intact Male; age 26–156 weeks.

Grazioso Salvare Rescue Dog Dashboard

Designed by Jersey Etchis - CS-140 - DSDC

View Rescue **Mountain or Wilderness Rescue** Disaster or Individual Tracking Reset

#	res_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	dateline	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
1	3138	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2010-03-23 16:23:00	2010-03-27T23:23:00	Suffering	Euthanized	Intact Male	38.568998448699	-97.32850480325	107.89715984127	
2	5125	2 years	A708726	Dog	Alaskan Malamute	Saliva/White	2013-07-30	2010-09-02 17:24:00	2010-09-07T23:24:00	Papo	Returns to owner	Intact Male	38.43993329138	-97.44862582737	104.8179714287	
3	6821	2 years	A723657	Dog	Rottweiler	Black	2015-05-13	2017-09-21 11:23:00	2017-09-27T23:23:00	Zane	Returns to owner	Intact Male	38.46677208743	-97.55720803642	120.92408879301	
4	6391	2 years	A706183	Dog	Siberian Husky	Black/White	2013-06-05	2015-09-02 16:41:00	2015-09-07T23:41:00	Lobo	Returns to owner	Intact Male	38.42237422575	-97.4309837816861	104.52797094427	
5	6557	6 months	A705681	Dog	German Shepherd	Saliva	2017-07-28	2018-03-22 11:54:00	2018-03-27T23:54:00	Sargent	Returns to owner	Intact Male	38.46698976985	-97.48508813424	26.6422619847639	
6	3138	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2010-03-23 16:23:00	2010-03-27T23:23:00	Suffering	Euthanized	Intact Male	38.568998448699	-97.32850480325	107.89715984127	
7	5125	2 years	A708726	Dog	Alaskan Malamute	Saliva/White	2013-07-30	2010-09-02 17:24:00	2010-09-07T23:24:00	Papo	Returns to owner	Intact Male	38.43993329138	-97.44862582737	104.8179714287	
8	6821	2 years	A723657	Dog	Rottweiler	Black	2015-05-13	2017-09-21 11:23:00	2017-09-27T23:23:00	Zane	Returns to owner	Intact Male	38.46677208743	-97.55720803642	120.92408879301	
9	6391	2 years	A706183	Dog	Siberian Husky	Black/White	2013-06-05	2015-09-02 16:41:00	2015-09-07T23:41:00	Lobo	Returns to owner	Intact Male	38.42237422575	-97.4309837816861	104.52797094427	
10	6557	6 months	A705681	Dog	German Shepherd	Saliva	2017-07-28	2018-03-22 11:54:00	2018-03-27T23:54:00	Sargent	Returns to owner	Intact Male	38.46698976985	-97.48508813424	26.6422619847639	

Rescue Candidate Breeds

Map showing rescue locations in Colorado, USA, with a blue marker indicating the location of the dashboard.

Screenshot 4 — Disaster or Individual Tracking Filter Applied

Filter returns records matching Doberman Pinscher, German Shepherd, Golden Retriever, Bloodhound, and Rottweiler breeds; Intact Male; age 20–300 weeks.

Grazioso Salvare Rescue Dog Dashboard

Designed by Jersey Etchis - CS-140 - DSDC

View Rescue **Mountain or Wilderness Rescue** Disaster or Individual Tracking Reset

#	res_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	dateline	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
1	2801	4 years	A696164	Dog	Rottweiler	Black/Brown	2012-01-01	2015-03-01 24:00:00	2015-03-07T23:25:00	Striker	Returns to owner	Intact Male	38.32997320111	-97.54930843984	208.88899263649	
2	3707	4 years	A712295	Dog	Bloodhound	Red	2012-09-26	2015-09-22 25:00:00	2015-09-27T23:43:00	Buster	Returns to owner	Intact Male	38.17999871387	-97.59320601272	209.8055158736	
3	6821	2 years	A723657	Dog	Rottweiler	Black	2015-05-13	2017-09-21 11:23:00	2017-09-27T23:23:00	Zane	Returns to owner	Intact Male	38.46677208743	-97.55720803642	120.92408879301	
4	2801	4 years	A696164	Dog	German Shepherd	Saliva	2012-01-01	2015-03-01 24:00:00	2015-03-07T23:25:00	Sargent	Returns to owner	Intact Male	38.46698976985	-97.48508813424	26.6422619847639	
5	3707	4 years	A712295	Dog	Bloodhound	Red	2012-09-26	2015-09-22 25:00:00	2015-09-27T23:43:00	Buster	Returns to owner	Intact Male	38.17999871387	-97.59320601272	209.8055158736	
6	6821	2 years	A723657	Dog	Rottweiler	Black	2015-05-13	2017-09-21 11:23:00	2017-09-27T23:23:00	Zane	Returns to owner	Intact Male	38.46677208743	-97.55720803642	120.92408879301	
7	6557	6 months	A705681	Dog	German Shepherd	Saliva	2017-07-28	2018-03-22 11:54:00	2018-03-27T23:54:00	Sargent	Returns to owner	Intact Male	38.46698976985	-97.48508813424	26.6422619847639	

Rescue Candidate Breeds

Map showing rescue locations in Colorado, USA, with a blue marker indicating the location of the dashboard.

4. Tools Used & Rationale

4.1 MongoDB — Model Component

MongoDB was selected as the database for several reasons that align specifically with this project's requirements:

- Document-oriented storage: Animal shelter records are naturally document-shaped (heterogeneous fields, nested data). MongoDB's BSON document model stores each animal as a self-contained record without requiring a rigid relational schema.
- Native Python integration: The PyMongo driver provides a first-class Python API. Query results are returned as Python dictionaries, which convert directly to pandas DataFrames via `pd.DataFrame.from_records()` with zero transformation overhead.
- Flexible querying: MongoDB's query language supports the `$in`, `$gte`, and `$lte` operators required by the Grazioso Salvare breed/age/sex filter specifications natively, without joins or complex SQL.
- Scalability: The 20,000-record Austin Animal Center dataset fits comfortably in MongoDB and the architecture would scale to larger shelter networks without schema changes.

4.2 Dash Framework — View & Controller

Dash (by Plotly) was chosen as the web application framework because it provides both the View and Controller layers of the MVC architecture in a single Python-native package:

- View layer: Dash's `html` and `dcc` components generate the full HTML/CSS/JS interface from Python, eliminating the need to write separate frontend code. The `dash_table.DataTable` component provides a production-ready interactive grid.
- Controller layer: Dash's `@app.callback` decorator system implements reactive data binding. Callbacks automatically re-execute when Input component values change and push results to Output components — this is the mechanism that makes the table, pie chart, and map all update simultaneously when a filter radio button is clicked.
- Plotly charts: Dash integrates natively with Plotly Express (`px.pie`) for chart generation, enabling the breed distribution pie chart to re-render from filtered DataTable state.
- JupyterDash: The `jupyter_dash` variant runs the Dash server inline within Jupyter/JupyterLab (as used in the Codio environment), enabling rapid development and testing without a separate server process.

4.3 Full Tool Stack

Tool / Technology	Purpose	Version
Python 3.11	Core application language	3.11.2
MongoDB	Document database (Model layer)	7.0.21
PyMongo	Python MongoDB driver	Latest
Dash / JupyterDash	Web framework (View + Controller)	Latest
Dash Leaflet	Interactive geolocation map widget	Latest
Plotly Express	Pie chart and data visualization	Latest
pandas	DataFrame manipulation and query results	Latest

Tool / Technology	Purpose	Version
dash_table	Interactive DataTable widget	Latest
base64	Logo image encoding for inline display	stdlib
Codio / JupyterLab	Cloud development environment	N/A

4.4 Resource Links

[Dash Documentation](#)

[Dash Leaflet Documentation](#)

[PyMongo Documentation](#)

[MongoDB Query Operators Reference](#)

[Plotly Express Documentation](#)

[Austin Animal Center Outcomes Dataset](#)

[Grazioso Salvare \(SNHU\)](#)

5. Steps Taken to Complete the Project

Step 1: CRUD Python Module (Project One)

Built the AnimalShelter class in CRUD_Python_Module.py implementing the four CRUD operations against the MongoDB aac.animals collection. The class accepts username and password at instantiation and connects via PyMongo's MongoClient. The read() method accepts a MongoDB query dict and returns a list of matching documents, which was the foundational interface used by all dashboard callbacks.

Step 2: Database Setup & Data Import

Imported the Austin Animal Center Outcomes CSV dataset into MongoDB using mongoimport, creating the aac database and animals collection with 20,000 records. Created the aacuser account with read/write permissions on the aac database. Verified connectivity using mongosh.

```
mongoimport --username aacuser --password SNHU12345 --authenticationDatabase admin --
db aac --collection animals --type csv --headerline --file aac_shelter_outcomes.csv
```

Step 3: Dashboard Layout

Constructed the app.layout using Dash html and dcc components. Key layout decisions included: wrapping the logo in an html.A anchor tag pointing to www.snhu.edu (per spec), using dcc.RadioItems for the four filter options, configuring the DataTable with pagination/filtering/row-selection, and arranging the pie chart and map side-by-side using a flex div.

Step 4: Filter Callbacks

Implemented the update_dashboard() callback function, which receives the selected radio button value and constructs the appropriate MongoDB query using the exact breed lists, sex values, and age ranges from the Grazioso Salvare specification document. The filtered results are returned as a list of dicts to update the DataTable's data property.

Step 5: Chart Callbacks

Implemented update_graphs() to render a Plotly Express pie chart from the DataTable's derived_virtual_data (the post-filter, post-sort view). Implemented update_map() to render a Dash Leaflet map centered on the GPS coordinates of the selected DataTable row, with a popup marker showing the animal's name and breed.

Step 6: Testing & Deployment

Ran the notebook in Codio's JupyterLab environment using app.run_server(). Tested all four filter states (Water Rescue, Mountain Rescue, Disaster Rescue, Reset) and verified that the DataTable, pie chart, and map all updated correctly for each selection. Captured screenshots of each filter state for documentation.

6. Challenges & Solutions

Challenge 1: KeyError on df.drop(columns=['_id'])

The dashboard crashed on startup with a KeyError because the MongoDB driver version in the Codio environment did not include the _id field in query results by default, so there was nothing to drop.

Solution: Wrapped the drop call in an existence check: if '_id' in df.columns: df.drop(columns=['_id'], inplace=True)

Challenge 2: Incorrect Password — Empty DataTable

The dashboard loaded but displayed no data. The CRUD module was instantiated with password "SNHU1234" (4 digits) while the actual aacuser account password was "SNHU12345" (5 digits). MongoDB's authentication failure returned an empty list silently.

Solution: Verified credentials directly in mongosh, then corrected the password constant in the dashboard file.

Challenge 3: Callback Error — selected_columns Is None on Load

The update_styles() callback crashed immediately on page load because Dash fires all callbacks on startup, and selected_columns arrives as None before any user interaction. Iterating over None raises a TypeError.

Solution: Added an early return guard: if not selected_columns: return []

Challenge 4: Map Crash Due to Integer Column Indexing

The update_map() callback used `df.iloc[row, 13]` and `df.iloc[row, 14]` to access latitude and longitude. When the DataFrame column order changed after filtering, these integer positions pointed at wrong columns and raised IndexError.

Solution: Switched to named column access (`df.iloc[row]['location_lat']`) and added a pre-flight check that all required columns exist before rendering the map.

Challenge 5: Typo df.filtered.drop Causing Runtime Error

A typo in the original starter code wrote `df.filtered.drop()` instead of `df_filtered.drop()`. Python interpreted `.filtered` as an attribute lookup on the DataFrame, raising AttributeError: 'DataFrame' object has no attribute 'filtered'. The error was masked by a broad try/except that fell back to the unfiltered `df`, making filters appear to have no effect.

Solution: Located the typo via grep and corrected `df.filtered.drop` to `df_filtered.drop`. Removed the masking fallback and replaced it with targeted error handling that prints the actual exception message to the Jupyter output.

7. Reproducing the Project

Prerequisites

- Codio account with a workspace containing JupyterLab
- Python 3.11+ with packages: jupyter-dash, dash, dash-leaflet, dash[diskcache], plotly, pandas, pymongo
- MongoDB 7.0 running on localhost:27017
- Austin Animal Center Outcomes CSV dataset

Setup Steps

1. Start MongoDB: run `mongod --auth --bind_ip localhost` in the Codio terminal
2. Import data: run the `mongoimport` command shown in Step 2 above
3. Place `CRUD_Python_Module.py` and `Grazioso Salvare Logo.png` in the same directory as the notebook
4. Open `ProjectTwoDashboard.ipynb` in JupyterLab
5. Select Kernel > Restart & Run All
6. Click the Dash app URL printed in the output cell to open the dashboard

File Structure

```
workspace/
    ├── ProjectTwoDashboard.ipynb      # Main dashboard notebook
    └── CRUD_Python_Module.py          # AnimalShelter CRUD class
```

```
|   └── Grazioso Salvare Logo.png      # Company logo (branding)
    └── aac_shelter_outcomes.csv      # Source data for import
```

Grazioso Salvare Rescue Dog Dashboard — Developed by Jeremy Ritchie — CS-340 SNHU