# CS 340 README Template

## About the Project/Project Title

*The project is called “AAC Dashboard, Grazioso Salvare edition”  
The project is a software application that can work with existing data from the animal shelters to identify and categorize available dogs.*

## Motivation

*In meeting with the client, they look for certain profiles in dogs to train. For example, search-and-rescue training is generally more effective for dogs that are no more than two years old. Additionally, different breeds of dogs are proficient at different types of rescues, such as water rescue, mountain or wilderness rescue, locating humans after a disaster, or finding a specific human by tracking their scent.*

## Getting Started

install MongoDB depends on what operation system you use.

MongoDB is used to store the data from the animal shelter. <https://www.mongodb.com/home>

*Download and install Python on your machine. To confirm if your installation is right, type python --version in your command line terminal. You should get something like Python 3.9.12*

[*https://www.python.org/*](https://www.python.org/)

*This project is using CRUD code that was coded with Python this allows* MongoDB *to create, read, update, and delete data.*

*The next step is to connect Python and MongoDB*

*PyMongo has a set of packages for Python MongoDB interaction start by creating a virtual environment and activate it by typing*

*python -m venv env*

*source env/bin/activate*

*Now that you are in your virtual environment, you can install PyMongo. In your terminal, type:*

*python -m pip install "pymongo[srv]".*

[*https://pymongo.readthedocs.io/en/stable/*](https://pymongo.readthedocs.io/en/stable/)

*The final step is to download and install Dash Framework, Dash is a framework for building data apps in Python.*

*Dash Enterprise simplifies the development and deployment process in a secure, scalable environment.* <https://plotly.com/dash/>

## Installation

*The programs that need to be installed are MongoDB, Python, PyMongo,* *Dash Framework.  
Preferred the most updated version of those programs for better results.*

## Usage

*update\_dashboard(filter\_type) – Set the dashboard with specific criteria for specific mission*

*update\_graphs – Displays graph of the breeds and their status*

*update\_map(viewData): Displays the location of the dogs*

### Code Example

*def update\_dashboard(filter\_type):*

*if filter\_type == 'Reset':*

*df = pd.DataFrame.from\_records(shelter.read({}))*

*elif filter\_type == 'Water':*

*df = pd.DataFrame.from\_records(shelter.read({'$or':[*

*{'breed': 'Labrador Retriever'},*

*{'breed': 'Newfoundland'}*

*]}))*

*elif filter\_type == 'Mountain':*

*df = pd.DataFrame.from\_records(shelter.read({'$or':[*

*{'breed': 'Bernese Mountain'},*

*{'breed': 'Great Pyrenees'},*

*{'breed': 'Greater Swiss Mountain'},*

*{'breed': 'Kuvasz'},*

*{'breed': 'Newfoundland'}*

*]}))*

*elif filter\_type == 'Disaster':*

*df = pd.DataFrame.from\_records(shelter.read({'$or':[*

*{'breed': 'Belgian Malinois'},*

*{'breed': 'Basset Hound'},*

*{'breed': 'St. Bernard'},*

*{'breed': 'Beagle'},*

*{'breed': 'Labrador Retriever'}*

*]}))*

*columns=[{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns]*

*data=df.to\_dict('records')*

*return (data,columns)*

*def update\_graphs(viewData):*

*graph = pd.DataFrame.from\_dict(viewData)*

*return [*

*dcc.Graph(*

*figure = px.scatter(graph, x='breed', y="outcome\_type"))]*

*def update\_map(viewData):*

*dff = pd.DataFrame.from\_dict(viewData)*

*# Austin TX is at [30.75,-97.48]*

*return [*

*dl.Map(style={'width': '1000px', 'height': '500px'}, center=[30.75,-97.48], zoom=10, children=[*

*dl.TileLayer(id="base-layer-id"),*

*# Marker with tool tip and popup*

*dl.Marker(position=[30.75,-97.48], children=[*

*dl.Tooltip(dff.iloc[0,4]),*

*dl.Popup([*

*html.H1("Animal Name"),*

*html.P(dff.iloc[1,9])*

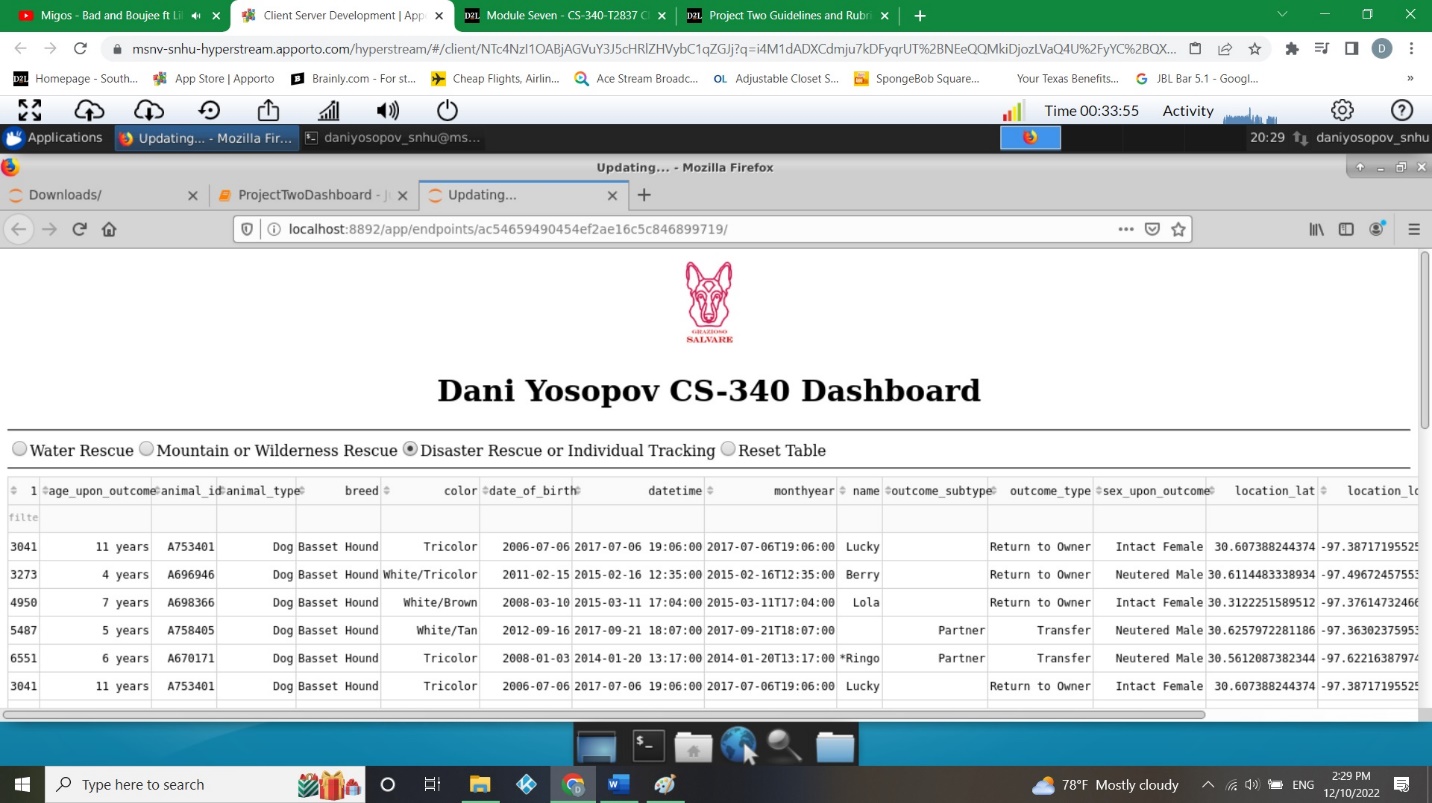
*])*

*])*

*])*

*]*

### Screenshots

**