# CS 340 README Template

## About the Project/Project Title

This project displays a user interface connected to a database of animals. Within the interface, a user may filter and search by specific criteria to find what they need.

## Motivation

Our client Grazioso Salvare needed a full stack application to manage a database of service animals.

## Getting Started

To get setup:

1. Download the “AnimalShelter.py” file
2. Download the “FinalProject.ipnyb” file
3. Start mongoDB with user authentication under “aacuser”
4. Run “jupyter notebook” in the kernel to start jupyter notebook
5. Run the “FinalProject.ipnyb” file

## Installation

You will need:

* Python: See <https://www.python.org/downloads/>
* Jupyter Notebook: See <https://jupyter.org/install>
* MongoDB: See <https://www.mongodb.com/docs/manual/installation/>
* Dash: See <https://dash.plotly.com/>

**Module Uses**

The Python driver used to connect to MongoDB is PyMongo and is used for synchronous Python applications. CRUD was implemented to be the “glue” between Pyrthon and MongoDB. This allows us to create, read, update, or delete anything in our database.

The Dash Framework allows for the creation of UI and webpage elements and uses functions such as callbacks to determine user input.

Pandas API allows for the use of a pandas data frame which formats the data and creates more functionality.

An MVC (model-view-controller) pattern was used to demonstrate the functionality of this web application. This model was chosen as it best represents the division of the front end and back end work being done in python.

### Code Example/Tests/Screenshots:

**User Interface:**

Company name

Description automatically generated

Table

Description automatically generated

Map

Description automatically generated with medium confidence

**AnimalShelter.ipnyb:**

**View**: This includes the application’s layout and how it looks on a page

Text

Description automatically generated Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Model**: This takes in the data and reads it is an a pandas dataframe. Authentication also takes place here.

Graphical user interface, text, application

Description automatically generated

**Controller**: This is where the application callbacks are made and where the user interaction takes place.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**CRUD Functions:**

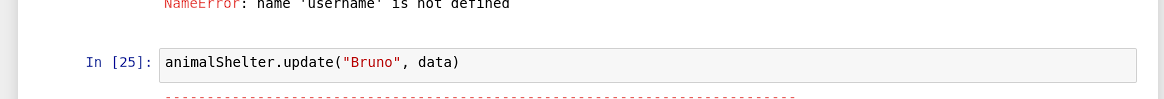
Create – This screenshot demonstrates the “Create” function in CRUD by creating a data set in Animal Shelter



Read – This screenshot demonstrates the “Read” function in CRUD by reading through the data set to try to find “Bruno.”



Update – This screenshot demonstrates the “Update” function in CRUD by finding an existing entry and updating it with the new data. In this case, the existing entry is “Bruno” and data is what will replace those fields.



Delete – This screenshot demonstrates the “Delete” function in CRUD by finding an existing entry and deleting it. In this case, the existing entry is “Bruno” and will be deleted from the database.



**MongoImport**: This screenshot demonstrates importing the csv file into MongoDB

Graphical user interface, text, application

Description automatically generated

**Authentication**/**User** **Creation:** This screenshot demonstrates logging into MongoDB as both an administrator and user named “aacuser”

Text

Description automatically generated

## Contact

Your name: Austin Palmer