# CS 340 README Template

## About the Project/Project Title

Animal Rescue is a software application which can very quickly and efficiently sort through large quantities of existing data from several animal shelters to identify and categorize available dogs for potential rescue-animal training.

## Motivation

This project has been created for an innovative international rescue-animal training company, Grazioso Salvare, who identifies dogs that are good candidates for search-and-rescue training. When trained, these dogs are able to find and help to rescue humans or other animals, often in life-threatening conditions.

## Getting Started

To get a local copy up and running, follow the steps in the “Installation” section below for your specific platform. After installation, see the “Usage” section for example code and how to use the Animal Shelter dashboard.

## Installation

Windows: Download and install Python from <https://www.python.org/downloads/windows/> On the customization screen click “Add python.exe to Path,” and then select “Will be installed on local hard drive.” Download get-pip.py from <https://bootstrap.pypa.io/get-pip.py> Open a command prompt and change to the directory containing the get-pip.py file and run the following command:

$ python get-pip.py

MacOS: Python is already installed on MacOS. Install pip using the terminal commands:

$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

$ python3 get-pip.py

Linux: Python is already installed on most Linux distributions, run “python –version” from the terminal and make sure you have at least Python 3.7.x or above. Install pip using the terminal command:

$ sudo apt install python3-pip

All: After installing Python and pip for your platform, install the Python MongoDB driver and JupyterLab using the terminal commands:

$ pip install pymongo

$ pip install jupyterlab

To install the Animal Shelter project, simply copy the source files (“crud.py”, “p2.ipynb”, and “logo.png”) to your project directory.

## Usage

This project uses a Python module which uses MongoDB to create, read, update, and delete data. Python was chosen as it is simple, fast, easy to use, and highly portable. MongoDB was chosen as it is a high-performance highly-scalable database solution. To use the Animal Shelter dashboard, open and run the Jupyter Notebook file “p2.ipynb”

Available CRUD Python functions are as follows:

**create(data)** # inserts *data* in JSON format into the database, returns cursor if success (**C**RUD)

**read(data)** # finds and returns *data* record in JSON format (C**R**UD)

**update(data, updated)** # finds and updates *data* with *updated*, returns True if success (CR**U**D)

**delete(data)** # deletes *data* record, returns True if success (CRU**D**)

The sections below show example uses including setup, code examples, tests, and screenshots.

### Code Example

# Import the AnimalShelter class:

from crud import AnimalShelter

# Create an instance of the class:

animal = AnimalShelter()

# Fetch and display data:

results = animal.read({"name": "Dolly"})

for doc in results:

print(doc)

### Tests

Import test data by running the following commands in the terminal:

# Start mongo with authentication, again make note of the

/usr/local/bin/mongod\_ctl start

# Login as admin:

mongo --authenticationDatabase "admin" -u "admin" -p

# Create a new user (for AAC database) account called “aacuser”

use AAC

db.createUser(

{

user: "aacuser",

pwd: passwordPrompt(), // use password “aacexpert” for test

roles: [ { role: "readWrite", db: "AAC" } ]

}

)

# exit:

exit

# use the new user:

mongo --authenticationDatabase "AAC" -u "aacuser" -p

# import test data

mongoimport --port 35793 --db AAC --collection animals --type=csv --headerline ./aac\_shelter\_outcomes.csv

Open the Jupyter notebook file “crud.ipynb” and select Cell->Run All to run all tests.

### Screenshots

Importing data into MongoDB:

*Text

Description automatically generated*

Creating the “aacuser” account:

*Text

Description automatically generated*

*Text

Description automatically generated*

Testing the crud.py module:

*Graphical user interface, text, application, email

Description automatically generated*

*Graphical user interface, text

Description automatically generated*

*Graphical user interface, text, application, email

Description automatically generated*

Animal Shelter Dashboard:

## Graphical user interface, application Description automatically generated

Using Filters:

Graphical user interface, application

Description automatically generated

Interactive graph and geolocation:Map

Description automatically generated

## Contact

Cameron “Pserkyt” Winningham

[cameron.winningham@snhu.edu](mailto:cameron.winningham@snhu.edu)

<https://pserkyt.com/>