# Jonathan Wolanyk

12/11/2022

# CS 340 README

## About the Project/Project Title

This project is designed to help the Austin Animal Center (AAC) have an efficient system that stores information about animals in their care. The end-goal would be for the AAC to have a system that allows users to create, read, update, and delete animal records. The AAC also requested certain premade filters be available to quickly search through their database. Specifically, the database should have a menu that allows quick searches for water, mountain or wilderness, and disaster relief animals.

MongoDB was utilized as the model component for development because of its flexibility and simple data storage structure. Additionally, queries in MongoDB are quite easy for new developers to grasp, making it ideal for volunteers that may be new to programming but want to help the AAC.

The Dash framework was used for viewports and controls for the web application. It is a great tool for this project because it supports analytical web applications without advanced knowledge of development. Since new developers will likely be working on the dashboard on a volunteer basis, the Dash framework makes sense for this program.

## Motivation

The AAC is constantly bringing and finding homes for new animals. Without an electronic system to manage all of the animals in their network, the organization would be bogged down with recordkeeping instead of helping animals and hopeful pet owners.

## Getting Started

To begin using the program, one will need Python and Jupyter Notebook installed on his/her computer to run the necessary files. We hope to create a standalone copy that users will be able to use without Python. Once installed, users can run the attached files to utilize the program.

## Installation

This application has been developed as a web-based application. As a result, as long as the user has access to the internet, they will be able to use the dashboard without installation. For developers, Python and Jupyter Notebook must be installed on their machines to work on the dashboard.

In order to install Python, users should visit the official website: <https://www.python.org/downloads/> and click the download link on the main page. Once installed, users can download the files attached to this document to begin using the database program. To install Jupyter Notebook, one can visit <https://jupyter.org/> and follow the prompts to install the software.

**Steps Taken to Complete:**

To complete this dashboard, quite a bit of research into Python, MongoDB, and the Dash framework was conducted. I struggled with the Dash framework the most initially since I had not used the framework in the past. I had experience with Python and MongoDB, so it was only natural to struggle with the unfamiliar portion of the project. The main issue I encountered was utilizing the @app.callback portion of code. Once I realized that it was providing an easy place to list the input and output of the following function, it became much easier to use.

## Usage

This program allows users to view animals in the database using the interactive table dropdown menu. It is also able to show where specific animals are located on the interactive map. An example of a dropdown menu search using the “Mountain or Wilderness” option is included on the next page. Additionally, users can type in the first row of any column to perform a keyword search for animals with specific characteristics. Please note that additional screenshots are listed later in this document with all of the current options for the dropdown list:

Graphical user interface, application

Description automatically generated

This program also supports create, read, update and delete methods for records. Some examples of functions that can be utilized are listed below:

### Code Example

Records can be created using the following syntax:

testAnimal = {

'animal\_id': 'Test',

'animal\_type': 'Dog',

'breed': 'Boxer',

'color': 'Brown',

'date\_of\_birth': '11-16-2020',

'name': 'Lester',

'sex\_upon\_outcome': 'Male',

'age\_upon\_outcome\_in\_weeks': '2'

}

a1.create(testAnimal)Tests

Records can be viewed using the following syntax, which reads data from an animal named Lester:

data = {'name' : ‘Lester’}

a1.read(data)

Records can be updated using the following syntax which updates a record for an animal named Lester to Bobby:

searchTerm = {‘name’ : ‘Lester’}

data = (‘$set’: {‘name’ : ‘Bobby’})

a1.update(searchTerm, data)

Records can be deleted using the following syntax, which deletes a record of an animal named Bobby:

deleteTerm = {‘name’ : ‘Bobby’}

a1.delete(deleteTerm)

**Tests**

Tests can be conducted by creating an animal record before attempting to view the record using the method listed above. The following screenshots show images of conducted tests as well as how to replicate tests for future developers of this program.

*Continued on next page…*

**Proof of “All” Selection Dashboard:**

**Graphical user interface, application

Description automatically generated**

**Proof of “Water” Selection Dashboard:**

**Graphical user interface, application

Description automatically generated**

**Proof of “Mountain or Wilderness Rescue” Selection Dashboard:**

**Graphical user interface, application

Description automatically generated**

**Proof of “Reset” Selection Dashboard:**

**Graphical user interface, application

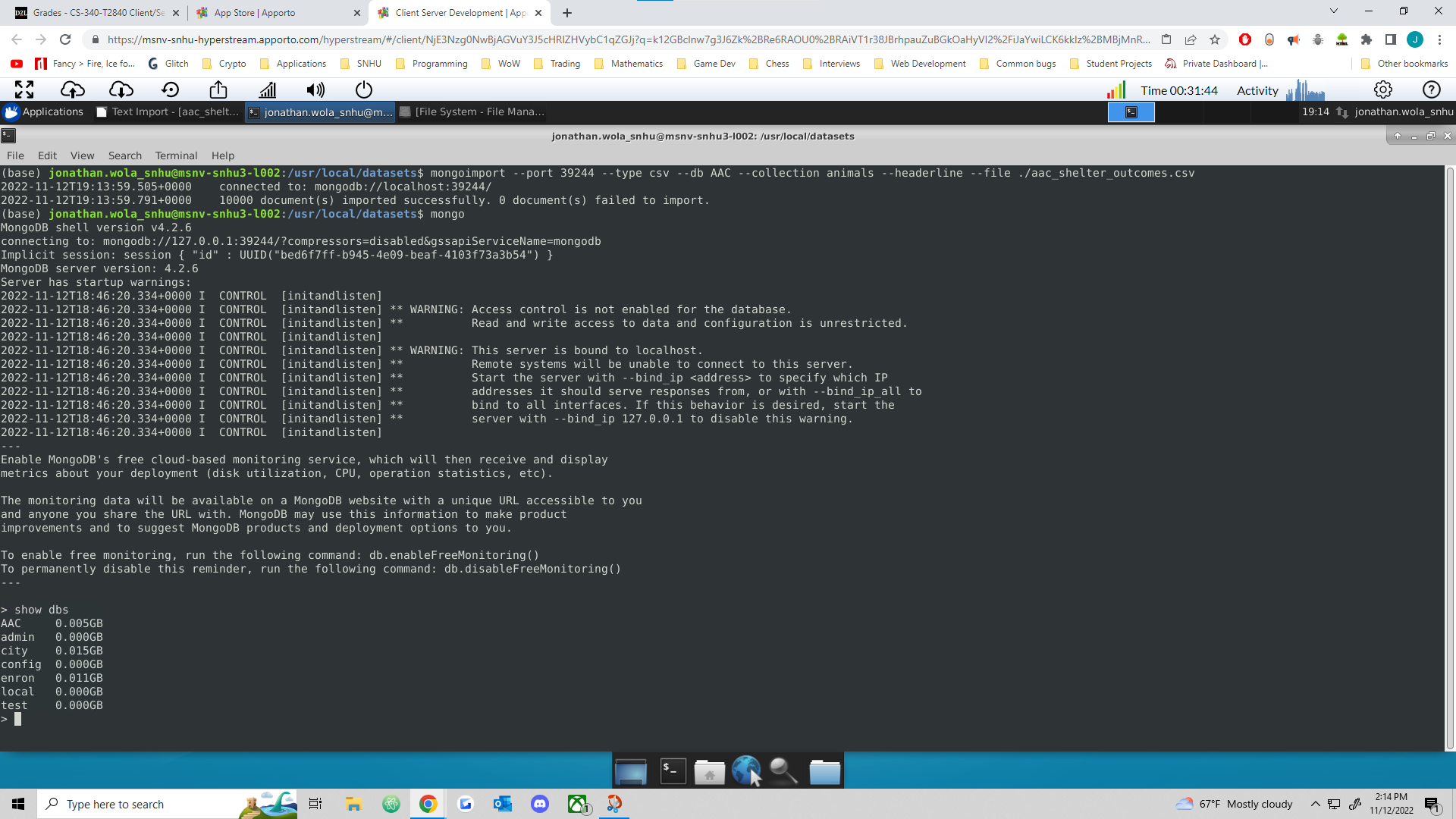
Description automatically generated**

**Jupyter Notebook Tests**

## Graphical user interface, text, application Description automatically generated

Text

Description automatically generated**Proof of CRUD:**

**Proof of AAC CSV Import**

**Proof of AAC CSV Load**

A screenshot of a computer

Description automatically generated

**Proof of User Account creation**

A screenshot of a computer

Description automatically generated

## Contact

Jonathan Wolanyk

Jonathan.wolanyk@snhu.edu