# CS 340 README Template

## About the Project/Project Title

This project has two major components: the CRUD back-end and the Jupyter front-end.

The CRUD project is a python-based backend of the application that allows a user to interact with a MongoDB database. These interactions fall under the CRUD principles, which stands for: Create, Read, Update, Delete.

The front end is a web-based application that allows for the user to view all animals in the Grazioso Salvare database. This data can be filtered on to show all animals or show what animals are best suited for each type of rescue operations. The front end uses the Dash framework which provides for the view and controller structure of the application. The Dash framework is a common, widely used framework for web applications and is kept up to date. It contains many more components that are not used in this phase of the project but can be added later in future phases if expansion is required. More information on the Dash framework can be found here: <https://plotly.com/dash/>.

## Motivation

This application serves the role as a database maintainer and allows for ease of use with upkeep. It also allows for users to quickly find the best candidates for rescue operations and potential trainings. This application allows for the convenience of storing login credentials, connection strings and providing methods to handle all CRUD functionality.

## Getting Started

The only steps required to get started is to:

* Be on the GS network
* Navigate to: animalsheltergsapplication.com
* Begin viewing data. Data can be filtered based on the buttons above the data table, and below you will see the geographical location of the animal as well as a graph with filtered breed percentages.

## Installation

N/A

## Usage

### Code Example

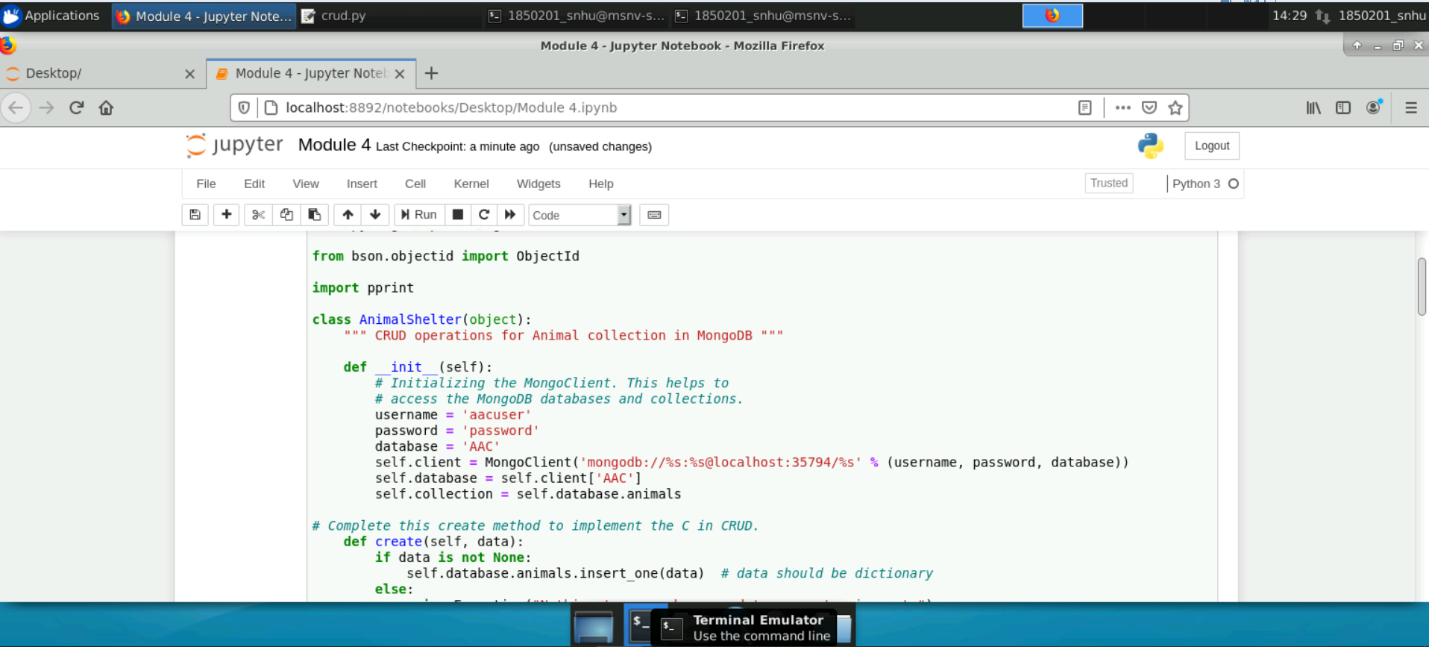
Text, letter

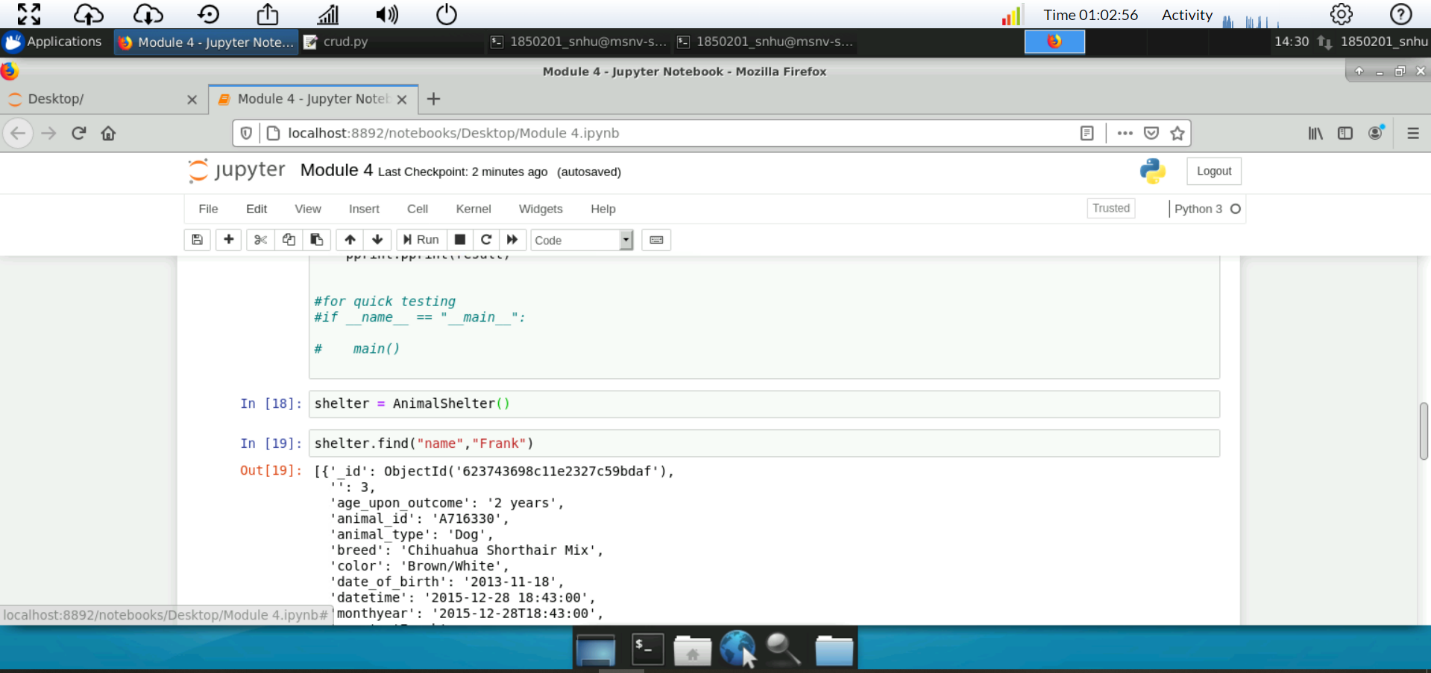
Description automatically generated

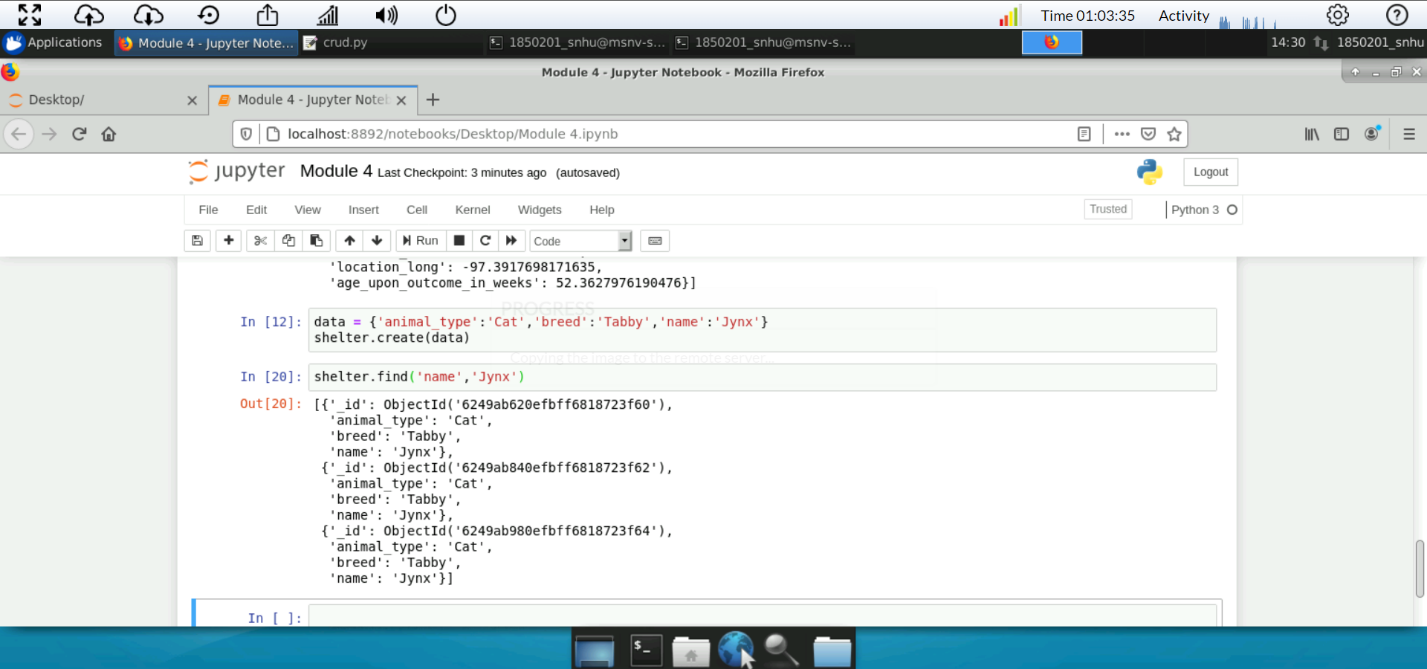
### Tests

I tested my code in a Jupyter notebook. You can do this by installing Jupyter Notebook onto your machine and copy/paste the code in to run it. From there, attempt to create a class object, find documents and create/return documents.

### Screenshots

**

**

**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

**Steps**

This project was completed iteratively. Firstly, the database was setup and configured. Secondly, the backend application was developed to ensure that the database could be hit and “CRUDed” appropriately. Lastly, the graphical front end was developed.

**Challenges**

It was challenging using the Jupyter environment. I did like that it had output consoles, but anything more than simply printing data became cumbersome and difficult to use. Also, there was very little error help or logging. The iterative approach helped mitigate time spent on issues, since smaller pieces were developed to completion first, but it was still difficult.

## Contact

Stuart Williams