# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

This project was made for demonstrating how to use Python to interact with a MongoDB database, as well as using Dash to create a localhost to view the database information in an intuitive way. It required knowledge of how to handle information between a client and a server, and how to link them together.

## Motivation

The motivation behind this project was to learn how to develop code that interacts with a local database and then displays the database’s data in an informative way. Learning is a passion and understanding how to interact with the database is vital for future success within the industry.

## Getting Started

To get a local copy up and running, follow these simple steps. Download the repository to your local PC, adjust the credential validation within CRUD.py to a valid user, and run the ProjectTwoDashboard.ipynb file. Click the link at the bottom where it says “Dash app running” to view the database contents using a Dash server.

## Installation

*List the tools you need to use the software and how to install them.*

The tools needed for the software are MongoDB, Jupyter notebook, PyMongo, Dash, and Pandas.

Links to dependencie information:

[Jupyter](https://github.com/jupyter/notebook)

Jupyter was used as the IDE because of its scripting capabilities. It provided an easy workspace to develop the code for this project in.

[MongoDB](https://github.com/mongodb/mongo)

MongoDB was used as the Database because our data is not structured, meaning that not every field for each record contains information. It also has a lot of community made plugins to interact with Python, making it an easy choice for us to use.

[PyMongo](https://github.com/mongodb/mongo-python-driver)

PyMongo is used for the interaction between Python and MongoDB. It lets us write queries to extract information from the database.

[Dash](https://github.com/plotly/dash)

Dash was used to model our data. The client wanted to be able to find specific animals for certain situations such as animals that help during a natural disaster. With Dash, we are able to load all records from the database into a data table, as well as display a map and charts to visualize the information they are looking for. Dash also lets us create pre-defined buttons that execute a specific query when clicked, enabling easy functionality to determine what animals are available for training for a specific need.

[Pandas](https://github.com/pandas-dev/pandas)

Pandas is used to store the records in our database into a data table that Dash can interact with.

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

### Code Example

For a code example, here is a snapshot of code designed to insert a document into an existing database:

*#Implement the C in CRUD*

*def create(self, data):*

*if data is not None:*

*insert\_result = self.database.animals.insert\_one(data) #data should be in dictionary*

*return True if insert\_result.acknowledged else False*

*else:*

*raise Exception("Nothing to save, because data parameter is empty")*

### Tests

For testing the above code, here is an example of a document hard coded into the AnimalTest.py to make sure its functionality is working:

#Test the Create method

print(animals.create({

'age\_upon\_outcome': "6 months",

'animal\_id': 'id\_test',

'animal\_type': 'test',

'breed': 'test breed',

'color': 'test color',

'date\_of\_birth': '2000-01-01',

'datetime': '2000-01-01 12:00:00',

'monthyear': '2000-01-01T12:00:00',

'name': 'test\_name',

'outcome\_subtype': '',

'outcome\_type': 'test',

'sex\_upon\_outcome': 'test',

'location\_lat': 20.050,

'location\_long': -20.050,

'age\_upon\_outcome\_in\_weeks': 155

})

)

### Screenshots

A screenshot of a computer

Description automatically generated

*A screenshot of a computer code

Description automatically generated*

*A screenshot of a computer program

Description automatically generated*

*A screenshot of a computer code

Description automatically generated*

Initial state of the dashboard

*A screenshot of a computer

Description automatically generated*

Dashboard with Water Rescue selected

*A screenshot of a computer

Description automatically generated*

Dashboard with Mountain or Wilderness Rescue selected

*A close-up of a map

Description automatically generated*

Dashboard with Disaster Rescue or Individual Tracking selected

*A screenshot of a computer screen

Description automatically generated*

Dashboard displaying all results after clicking Return All Results

*A screenshot of a computer

Description automatically generated*

Dashboard showing a filter for all animals with age 1 year

A screenshot of a computer screen

Description automatically generated

Dashboard showing the datatable sorted in descending order of color

A screenshot of a computer

Description automatically generated

Dashboard showing a complex command of outcome type equaling adoption and sorted by age

A screenshot of a computer

Description automatically generated

## Roadmap/Features (Optional)

N/A

**Explain the steps that were taken to complete the project:**

The project was completed in iterations. It started with first learning how to import data from a csv file into a mongo database. Next, I learned how to write Python code to interact with the database, and display CRUD functionalities (Create, Read, Update, Delete). After that came learning how to use Dash to host the information on a server and display information in a friendly and intuitive way.

**Identify any challenges that were encountered and explain how those challenges were overcome:**

The biggest challenge I had was trying to get the map to display in dash. I looked over my code multiple times for any spelling mistakes or syntax errors only to discover there was a problem with how I configured my datatable. I made the fix and got the map working.

Another challenge I had was getting the datatable to update with the selected radio buttons for disaster types. I checked the callbacks and found that my datatable wasn’t being accessed properly, therefore not updating the table. I corrected the code, and it now works.

## Contact

Your name: Bryce Cooperrider