# CS 340 README

## About the Project/Project Title

This project will provide the necessary Create and Read functionalities for the users to be able to interact with the database quickly and easily.

**Future Scope**

This application will eventually provide a user interface and improve additional query functionality.

## Motivation

The motivation behind the creation and upkeep of this project is so in that it’s a requirement of the CS340 Client/Server Development course. The AAC Shelter Outcome project exists with the intention of decreasing the number of stray animals in a given area; Furthermore, after stray retrieval, the stray animals are treated if necessary. Additionally, qualifying stray animals will receive special training that will prepare the animal for a new family life. AAC Shelter Outcome advocates work to find good homes for qualifying strays or otherwise implementing a practical resolution for stray animals that aren’t fit for a family-based lifestyle; This could include some older animals or animals that are terminally ill, and so on.

## Getting Started

This is an example of how you may give instructions on setting up your project locally: “To get a local copy up and running, follow these simple example steps.”

In order to set up the project in a local environment, you will need MongoDB and Python as well as a was a copy of the data set that this project works with.

## Installation

For this project to be developed, there were several resources that were implemented for the client Grazioso Salvare. MongoDB was used for the client database AAC. MongoDB was utilized for the base level of the application; Additionally, JupyterLab/Notebook was used in conjunction with Python programming language to produce the client-side of the application.

JupyterLab/Notebook:

<https://jupyter.org/install>

Python:

<https://www.python.org/downloads/>

<https://wiki.python.org/moin/BeginnersGuide>

MongoDB:

<https://www.mongodb.com/docs/manual/administration/install-community/>

Anaconda:

<https://docs.anaconda.com/anaconda/install/index.html>

## Usage

This project works to increase database query performance. More specifically, these scripts assist in the advanced queries to create and/or retrieve documents from the database with efficiency and speed. Using this program will help the user to query the connected database with efficiency and accuracy. Also, create functionality is also a feature that can be utilized while running this project. Additionally, this interactive web application provides the user a way to visualize the data within the client database.

**Existing Issues/Complications & Screenshots**

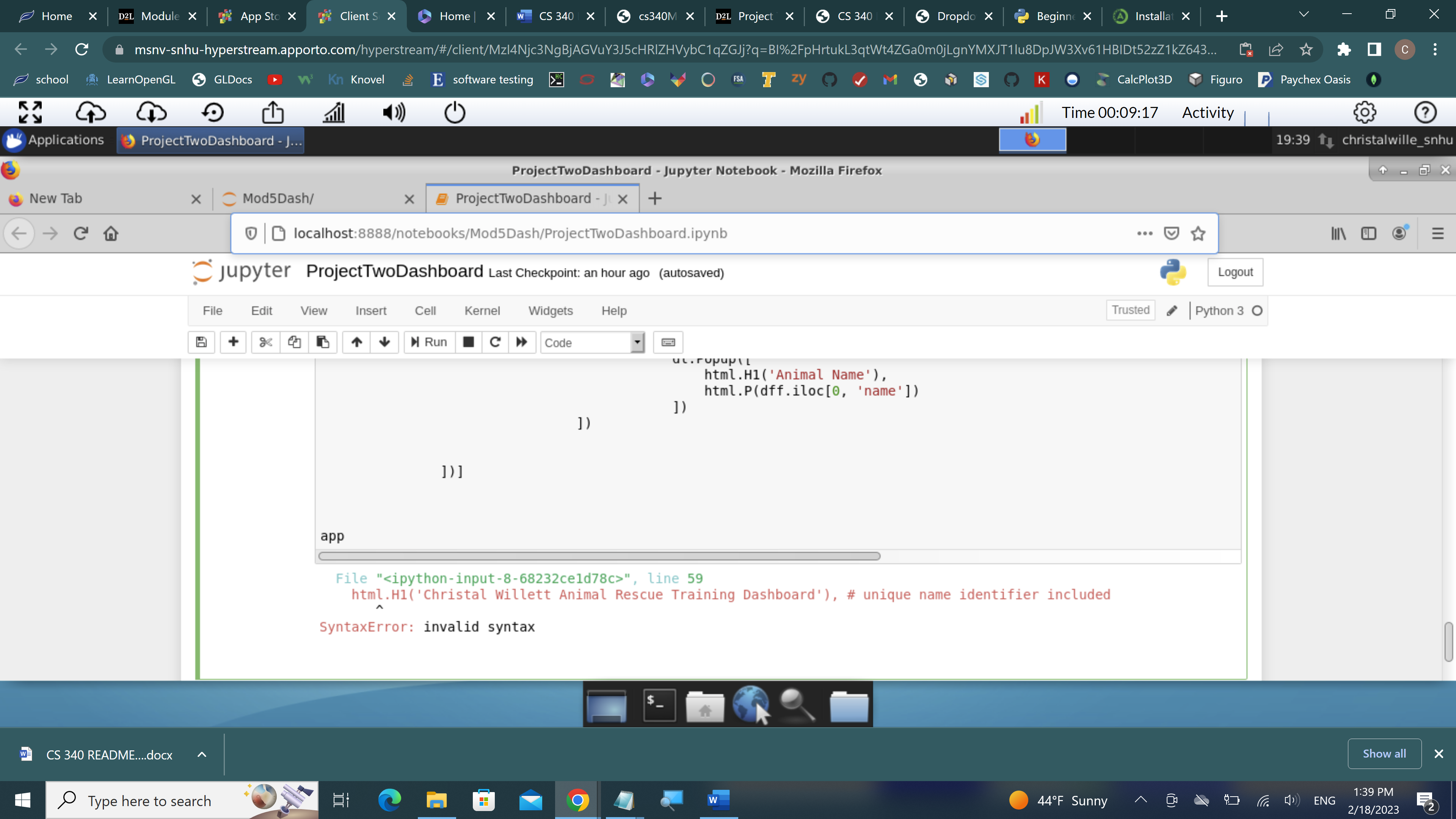
While working on this dashboard, I had the same issue with authentication and enabling access control for a “aacuser” account with read and write privileges assigned to the AAC database. This issue persisted for weeks, however, it has since been resolved and I was able to authenticate and enable security to the varying database users including the admin user. When trying to run my code for each different dash component added, I continued getting an ‘Invalid Syntax” error on app = JupyterDash(‘SimpleExample’), however, once previous issues with authentication were resolved, the error showed below is where I am stuck. Feedback is always appreciated!

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated



### Code Example

The script for this project includes the functionality for the user to Create and Read data from a database. Code examples consist of: def create(self, data): if data is not None: self.database.animals.insert(data) Return true if insert != 0. Features within the ProjectTwo JupyterNotebook include a dropdown dash component using the following logic: @app.callback(Output(‘dd-output-container', ‘children’), Input(‘dropdown-filtering’, ‘value’)) def update\_output(value): return f’Selected {value}’, geolocation chart as defined with @app.callback(Output(‘graph-id’, ‘children’), [Input(‘datatable-id’, ‘derived\_viewport\_data’)]) def update\_graphs(viewData), pie-chart, and filtering options so that the dashboard reflects the options selected by the user.

## Contact

Christal Willett