Cristian Chavez

2/23/23

CS 340

**Grazioso Salvare**

The Grazioso Salvare Application is meant to provide users the ability to easily access animal center databases and search for animals that will fit their desired purpose.

**Application Components**

To create the dashboard, MongoDB was used for our database as controlled user and admin functionality. Python was used in order to execute Create, Read, Update, and Delete functions on the MongoDB database. And the Dash framework was to create an interface that users could easily interact with. There was some help from the Pandas dataframe which was used alongside Dash to help more easily move data.

## Setting Up The Python Code

* To get started, you’ll want to login to the SNHU Virtual Lab with Apporto. Next, open the terminal. Start mongo with no authorization, use “mongod\_ctl start no-auth”.
* Once mongo is running, import the .csv file on your machine using the appropriate mongoimport method.
* We will now add user authorization. Create an admin account. Once the admin account is created, while signed in as admin, create a user named “aacuser” belonging to the database “AAC”. You have now set up the data and the user authorization!
* Moving onto the Create and Read functionality, we will use Jupyter Notebook. Before you open it, right click on the app and click “edit launcher”, ensure that “Run in Terminal” is checked.
* In Jupyter, create a new Text file. Rename this textfile with a .py ending, this will turn it into a python file we can work with. In this file implement the Create and Read functions, ensure that in the “\_init” function that you declare the username of “aacuser” and the password you gave it.
* Lastly, create a new Python 3 notebook file, here is where testing will be conducted. You can test the function of the python written by using the create function you made with data that’s hardcoded in. Ensure this data is in the database by using the read function that was also created to display the inserted data.

**Issues**

During the creation of the user dashboard, reviewing the dash documentation on all their components was necessary, along with reading mongoDB documentation on how to create the appropriate queries to the database using python.

**Functionality**

Below are a few examples of the dashboard, and how it is changes depending on the filtering option chosen

Graphical user interface, application

Description automatically generated

**Water Rescue**

Graphical user interface, application

Description automatically generated

**Mountain Rescue**

**Graphical user interface, application

Description automatically generated**

**Disaster Rescue**

Graphical user interface, application

Description automatically generated

**Original, Unfiltered Data**