# Grazioso Salvare Dashboard Enhanced

**About the Project/Project Title**

*Grazioso Salvare Dashboard –* This dashboard operates as a user interface to interact with data stored in a MongoDB database using a PHP browser interface that also utilizes PHP endpoints as a middleware for interacting with the database. The dashboard facilitates a web interface to view the records retrieved based on filtering and interactions with the widgets and can be adapted to present data for a variety of different datasets.

## Motivation

The primary driving factor behind this project is to provide a contracted client’s desired product of a full stack solution aimed at easily organizing, filtering, and sorting data through a web browser. The dashboard is currently designed for use with a collection of animal records for the Grazioso client but could be easily adapted to serve a variety of records using the PHP server code and the MongoDB database.

## Getting Started

To get a local instance of the project up and running follow the steps below.

1. Start an instance of MongoDB – ‘mongod\_ctl start-noauth’
2. Import a dataset into the local database to interact with – ‘mongoimport --db AAC --collection animals --type=csv --headerline --file= /usr/local/datasets/aac\_shelter\_outcomes.csv --port 5247’ #Replace the port with your local instance port and the specified db and collection as desired for your data
3. Start the mongo command line interface – ‘mongo’ or ‘mongosh’
4. Create a user to connect to the database (AAC in this example) and interact with the data -
   1. ‘use admin’
   2. ‘db.createUser({user: “aacuser”, pwd: passwordPrompt(), roles: [{role: “readWrite”, db “AAC”}]})’
   3. Enter the password for the new user at the prompt.
5. Copy an authentication configuration for the mongod service to the current configuration – ‘sudo cp /etc/mongod\_withauth.conf /etc/mongod.conf’
6. Restart the mongod service to run with authentication required – ‘sudo systemctl restart mongod.service’
7. Host the PHP and HTML code on a web server (Apache, Nginx, IIS, etc.) with PHP installed
   1. Ensure you have the PECL PHP driver installed. Instructions available at: https://www.php.net/manual/en/mongodb.installation.pecl.php
   2. Update the port used to create MongoDB connections in config.php on line 15 if different from 27017
   3. Optional: update any references to localhost within config.php, pro.php, and dashboard.php if you are hosting the database on a different server.
8. Navigate to http://<server>/login.html
9. Enter the user credentials that you created for accessing the MongoDB database and you will be redirected to the project Dashboard available at dashboard.php

This project was created using PHP that functions as both a client interface and middleware to interact with MongoDB. At runtime there may be warnings encountered based on how data is being accessed by the incorporated PHP modules for each of the widgets but none of this impacts the functionality of the dashboard for the intended use. A significant challenge when completing this project was properly formatting the HTML elements within the dashboard while maintaining the AJAX functionality. Ultimately, I would like to further improve this project in the future by using HTML templated pages that are called by the PHP code instead of embedding the HTML rendered within the code itself.

## Installation

This project is written to be run with PHP 7.4.3 or later and relies upon the PECL PHP MongoDB driver being installed. The HTML elements rendered within this project relies upon JavaScript for dynamic and cleanly formatted elements such as the datatable presented on the dashboard the Graph that is created. These are made possible by code available from jQuery.com and Microsoft aspnetcdn.com for Ajax functionality, canvasjs.com, and openstreetmap.org. The licensing for these technologies has been included in this project if required by the entities providing supporting technologies.

This project also relies on connecting to an existing MongoDB instance that is either locally or remotely accessible for accessing and updating data. MongoDB was chosen for the flexibility in scaling, rapid response to queries and ease of integration with Python

## Usage

An instance of MongoDB must be running with imported data following the steps presented above. After hosting the PHP and HTML content on a webserver, the dashboard can be access by navigating to http://<server>/login.html using a modern web browser and logging in with the user credentials you configured in the MongoDB database. The dashboard will automatically populate a table with unfiltered data retrieved from the MongoDB database. Additionally, below the table two widgets will display pie graph representation of breeds presented in the table and a map widget which can be used to view locations of animals in the table. The data presented in the table can be filtered using two methods. First there is a row of four buttons at the top of the table which will apply three filters to present records matching qualities Grazioso Salvare has deemed desirable for dogs acting as ‘Water Rescue’, ‘Mountain or Wilderness Rescue’, and ‘Disaster or Individual Tracking’. These three different filters can be applied by clicking the corresponding button and can be cleared by clicking the ‘Reset Filters’ button. Additionally, the data presented in the table can be filtered and searched by using the search bar located above the table. To clear these filters just delete the filter value and press enter.

### Testing

Testing has been performed for each desired interaction detailed within the client Dashboard Specification documentation. This includes applying filters using the included buttons, filtering data within the table using search strings, clearing filters, paging through the table for multiple outputs, sorting, observing the updated content for both the map and graph as data changed within the table, and engaging the html anchor website from the Grazioso Salvare logo. Examples of these actions can be seen in the accompanying screenshots that follow.

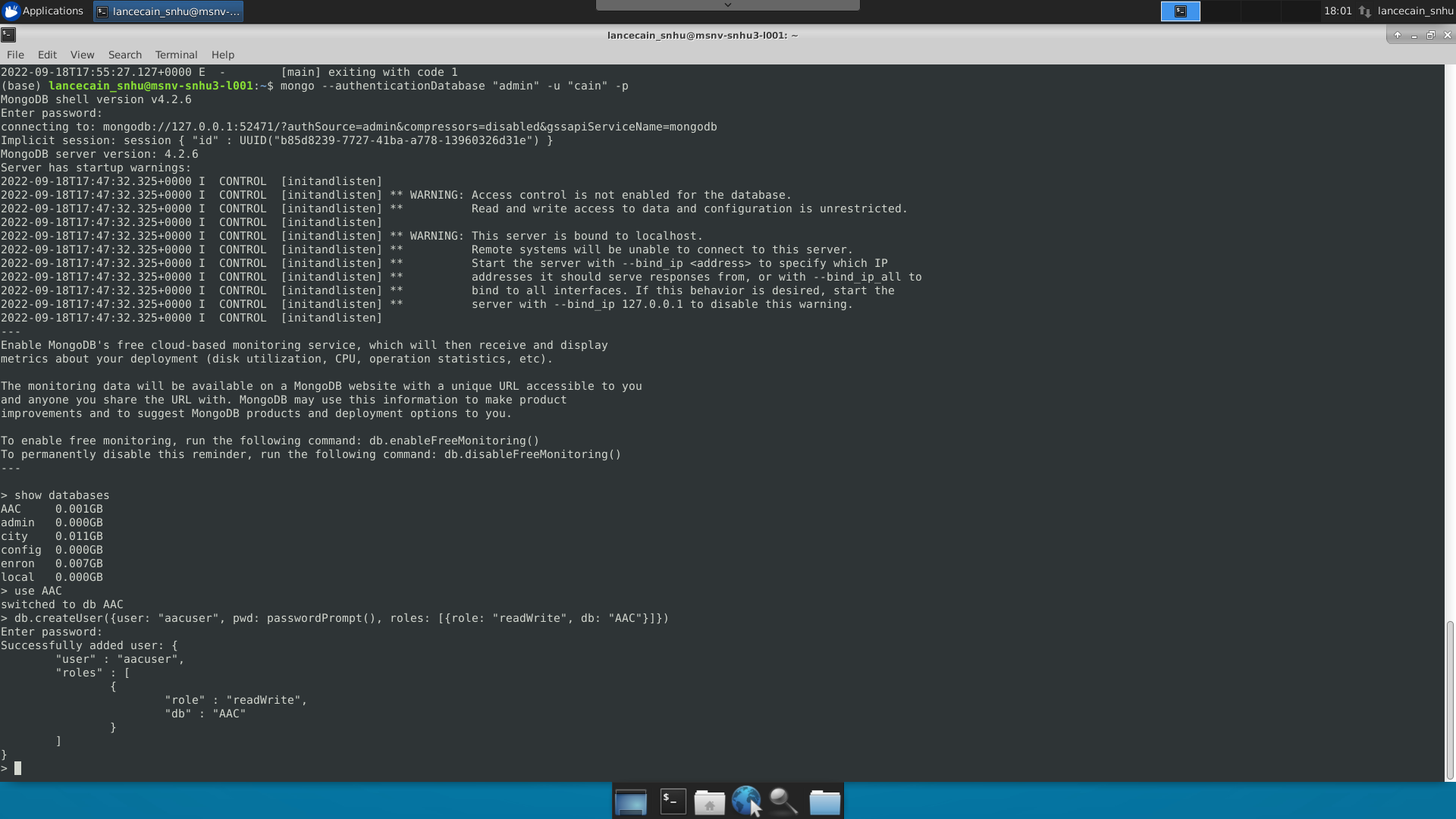
### Screenshots

Importing the Austin Animal Center Outcomes data CSV using the MongoDB import tool.

Graphical user interface

Description automatically generated

Authenticating as an administrator account and “aacuser” user account



Graphical user interface, text

Description automatically generated

Starting MongoDB with user authentication and loaded data

Graphical user interface, text

Description automatically generated

Dashboard login page



Accessing the dashboard after login

Graphical user interface, application

Description automatically generated

Opening the Grazioso Salvare logo html anchor in a new tab

Graphical user interface

Description automatically generated

New tab to [www.snhu.edu](http://www.snhu.edu) opened

Graphical user interface, website

Description automatically generated

View of the data table graph and map widgets

Graphical user interface, application

Description automatically generated

Water Rescue Filter Applied

Table

Description automatically generated

Updated widgets for filter application

Graphical user interface, application

Description automatically generated

Disaster Rescue Filter Applied

Graphical user interface, application

Description automatically generated

Reset Filters

Graphical user interface, application, table

Description automatically generated

Filter data viewed in the table using search

Graphical user interface, application

Description automatically generated

Reset by clearing search string

Graphical user interface, application, table

Description automatically generated

Sort by column

Graphical user interface, application

Description automatically generated

## Additional Resources

This project would not be possible without the helpful insights provided at the following links and project documentation made available for the technologies employed like MongoDB and PHP MongoDB integration instructions. I encourage everyone to review these items to get a better understanding of the inner workings of this project.

https://www.mongodb.com/docs/manual/reference/method/

https://www.mongodb.com/docs/php-library/current/tutorial/install-php-library/

https://canvasjs.com/javascript-charts/

https://www.openstreetmap.org/about

https://learn.microsoft.com/en-us/aspnet/ajax/

https://learn.microsoft.com/en-us/aspnet/ajax/cdn/overview

## Contact

Lance Cain: [lance.cain@snhu.edu](mailto:lance.cain@snhu.edu)