Grazioso Salvare Database Dashboard

**Functionality**

For this project, the focus was on creating a dashboard that communicated with the database provided. This is to make it available to be accessed by the users, not a software engineer. The dashboard should be easily accessible and understood.

**Tools**

The main tools utilized were MongoDB and Python. MongoDB was chosen as a noSQL solution in processing the varying attributes of the database. Regarding Python, there were libraries that were incorporated that were made with intentions of working with Mongo. PyMongo was used to connect to MongoDB and manipulate the data with ease. The Dash framework was used to design and portray the information in an html style. This made looking and interacting with the database a possibility for clients. The Dash [documentation](https://dash.plotly.com/dash-core-components) was used to properly implement a dropdown selection choice containing different categories of animals. The Python [documentation](https://dash.plotly.com/dash-core-components) was used to create the pie chart containing the breeds.

**Steps**

The first step is to connect MongoDB to the database. This includes importing the database and creating users with roles. The next step was to communicate with Python using the PyMongo library. This is when the CRUD functions are created and used to begin handle the data. Creating the dashboard was done using the Dash framework, and this provides the users with an accessible way to access the data without having to code in queries.

**Challenges**

The biggest challenge I personally faced was design when creating the web page. Having no prior experience with front end design, it was a new challenge to not only ensure that the code functions correctly, but also that it has a certain degree of aesthetics for the users. This step required some extra reading in order to make sure it looked and felt up to par.

**Screenshots**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Sources**

“Dash Core Components.” *Plotly*, https://dash.plotly.com/dash-core-components.

Plotlygraphs. “Pie Charts.” *Pie Charts*, Plotlygraphs, 3 July 2019, https://plotly.com/python/pie-charts/.