Angel Baez

CS-340 Client/Server Development

Prof. James Webb

Grazioso Salvare Dashboard

README

**Project Overview** The Grazioso Salvare organization dashboard is a web application designed to help identify available rescue animals based on their function, type, and location. The project is implemented using MongoDB, Dash, Plotly, and Leaflet. The interactive application provides a data table containing key information for the organization's operations, dynamically updating based on selected filters.

**Steps and Installation**

* Install MongoDB.
* Set up a suitable development environment (PyCharm).
* Install Jupyter Notebook using the command: pip install notebook.
* Import the data into MongoDB from the file “aac\_shelter\_outcomes.xls.”
* Create a user in MongoDB with the necessary permissions.
* Run Jupyter Notebook using the command: jupyter notebook in the PyCharm Windows terminal.
* Develop a \*.py file containing methods for connecting to MongoDB and data manipulation.
* Create a \*.ipynb file to verify the DB connection and methods for manipulating DB documents.
* Implement the dashboard in a new \*.ipynb file, configuring the interactive table, map, and chart.

**Tools Used and Justification**

MongoDB: Used along with the Pymongo library, facilitating connection and data manipulation.

JupyterDash: Allows the development of an interactive interface without requiring advanced web development knowledge.

Jupyter Notebook: Provides an ideal development environment for organizing and testing the project's various modules.

Plotly Express: Used for creating interactive charts, such as the pie chart that displays the number of available animals based on the selected filter.

Dash Leaflet: Enables interactive visualization of geographic locations within the dashboard.  
Pandas: Used to extract, filter, and sort MongoDB documents, facilitating their display in the interactive table.

**Dash Framework, Controller, and Application**

The Dash framework is a Python library that enables the creation of interactive web applications with data visualization capabilities, including charts and maps, without requiring extensive blocks of coding like HTML, JavaScript coding, etc. The view of the interactive web is defined with the “Layout” attribute, using components such as “HTML.Dic(), dcc.Graph and dash\_table.DataTable() to structure the UI. The controller uses the @app.callback(), which dynamically updates the table, chart, and graph.

**Challenges**

Different challenges should be experienced during the development of the process. In my case, I have a connection issue with MongoDB. When creating the user account in MongoDB, you must ensure that the proper access and attributes are given to the user account, or the user won’t be able to access the DB resources. Another possible scenario is the libraries in Jupyter. I must update and reinstall some libraries using pip install, update dash, and Plotly dash-leaflet because my server could not display the different elements in the interactive web. Good practice is always to ensure you are using the newest library when importing them since some of them could be depreciated.

**Conclusion**

The Grazioso Salvare Dashboard is implemented using MongoDB and Dash through Python to provide an interactive experience, allowing users to filter, visualize, and analyze rescue animal information in real-time. The project depicts the capability of Dash and MongoDB to develop interactive web with the necessity of advanced web development tools.

**Steps and Screenshots**

Importing aac\_shelter\_outcomes.xls to MongoDB

A screenshot of a computer screen

AI-generated content may be incorrect.

Creating the MongoDB user to access the DB

A screenshot of a computer

AI-generated content may be incorrect.

Python code in Jupyter environnement local server for DB interaction : \*.py file.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

Testing codes: Connection and the different methods for DB manipulation. \*ipynb file.

*A screenshot of a computer

AI-generated content may be incorrect.*

A screenshot of a computer program

AI-generated content may be incorrect.

Dashboard for interactive web: \*.ipynb file.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a code

AI-generated content may be incorrect.

Running Dashboard in Jupyter web server

A computer code with text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A map with a map and a map with a map and a map with a map and a map with a map and a map with a map and a map with a map and a map with

AI-generated content may be incorrect.