Fisal Ikhmayes

CS340

04/21/2024

Module 7

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Grazioso Salvare Dashboard Project**

**Description**

This project involves the development of a web application dashboard for Grazioso Salvare, a rescue-animal company. The dashboard allows users to interact with and visualize data from animal shelters, helping to identify dogs for search-and-rescue training.

**Functionality**

The dashboard includes the following functionality:

* Interactive data table displaying unfiltered view of Austin Animal Center Outcomes data set.
* Filtering options for users to select specific rescue types and preferred dog breeds.
* Interactive data table that responds to filter selections.
* Geolocation chart and a second chart of choice that dynamically respond to filter selections.

**Tools Used**

* **Python**: Used for backend development and interfacing with MongoDB.
* **MongoDB**: Used as the database to store and retrieve data.
* **Dash Framework**: Provides the view and controller structure for the web application.

**Why MongoDB?**

MongoDB was chosen as the database due to its flexibility and scalability. It provides a document-based data model that integrates well with Python, making it easy to work with complex data structures.

**Why Dash Framework?**

Dash was chosen for its simplicity and ease of use. It allows for the creation of interactive web applications using Python, without the need for additional frontend frameworks like JavaScript.

**Resources Used**

**MongoDB**

**Dash Frameworl**

**Steps Taken**

1. Reviewed Dashboard Specifications Document to understand requirements.
2. Created data table to display unfiltered data.
3. Developed database queries for filtering functionality.
4. Created interactive options for filtering.
5. Modified data table and created charts for dynamic data display.
6. Tested dashboard, capturing screenshots for documentation.

**Challenges**

* Challenge 1: Implementing interactive options for filtering.
  + Solution: Utilized Dash Core Components to create radio items and drop-down menus for filtering.
* Challenge 2: Ensuring charts dynamically respond to filter selections.
  + Solution: Used callback functions in Dash to update charts based on filter selections.