**Project Two**

**Brandon Goller**

**Southern New Hampshire University**

**Computer Science**

**CS-340: Client/Server Development**

**Tad Kellogg**

**December 7, 2023**

**Grazioso Salvare Animal Shelter Dashboard**

**Project Overview**

As part of a collaboration with Grazioso Salvare, an international rescue-animal training company, Global Rain has developed a comprehensive software solution to identify and categorize dogs from local animal shelters for search-and-rescue training. This project includes a MongoDB-based database, a Python module for CRUD operations, and a client-facing web application dashboard built with the Dash framework.

**Project Functionality**

**Dashboard Features**

1. **Interactive Filtering Options:**
   * Rescue Type Dropdown: Allows users to filter dogs based on rescue type, including Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and a Reset option.
2. **Data Table:**
   * Displays relevant information about the dogs, such as age, breed, and color.
   * Supports single-row selection.
3. **Graphs:**
   * Geolocation Chart: A map visualization with markers for selected dogs, displaying their breed and name.
   * Additional Graph: Another graph displaying relevant data.

**Testing and Deployment Screenshots**

1. **Initial State:**
   * Widgets for interactive options, data table, and charts are visible.

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. **Water Rescue Filter:**
   * Widgets adjusted to show only dogs suitable for water rescue.

A screenshot of a computer

Description automatically generated

1. **Mountain or Wilderness Rescue Filter:**
   * Widgets adjusted to show only dogs suitable for mountain or wilderness rescue.

A screenshot of a computer

Description automatically generated

A screenshot of a map

Description automatically generated

1. **Disaster or Individual Tracking Filter:**
   * Widgets adjusted to show only dogs suitable for disaster or individual tracking.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Reset Filter:**
   * Widgets return to their original, unfiltered state.

A screenshot of a computer

Description automatically generated

**Tools Used**

1. **MongoDB:**
   * **Rationale:** MongoDB was chosen for its flexibility and scalability. It provides seamless integration with Python, allowing for efficient data storage and retrieval.
2. **Dash Framework:**
   * **Rationale:** Dash provides a user-friendly and intuitive framework for building web applications with Python. It enables the creation of interactive dashboards with ease.

**Steps to Reproduce**

1. **Clone Repository:**

bashCopy code

git clone https://github.com/bgoller85/grazioso-salvare-dashboard.git cd grazioso-salvare-dashboard

1. **Install Dependencies:**

bashCopy code

pip install -r requirements.txt

1. **Run the Dashboard:**

bashCopy code

python app.py

1. **Access the Dashboard:** Open your web browser and navigate to <http://127.0.0.1:8050/>

**Challenges and Solutions**

1. **Challenge: Geolocation Chart Error**
   * **Solution:** Adjusted the code to correctly extract latitude and longitude information from the dataset.
2. **Challenge: Widget Alignment**
   * **Solution:** Refactored the layout and styling to ensure proper alignment and visual appeal.