**Project Two**

Jeffrey Carlson

Southern New Hampshire University

Client/Server Development

Hari Paruchuri

8/12/2023

Introduction

This document presents the Grazioso Salvare Dashboard project, which involves creating a web-based dashboard for exploring and interacting with data from an animal shelter database. The project utilizes two main tools: Dash, a web application framework, and MongoDB, a database system.

Dashboard Features

The Grazioso Salvare Dashboard offers the following features:

- Filtering: Users can select a rescue type, such as water rescue or mountain rescue, and the dashboard will display relevant animal data.

- Data Table: The dashboard includes a table that presents information about animals. Users can select specific entries for more details.

- Chart: A chart is integrated into the dashboard, displaying the distribution of animal types in the shelter.

- Map: The dashboard incorporates an interactive map with markers that indicate animal locations.

Tools and Rationale

MongoDB was chosen as the data storage solution due to its capabilities and seamless integration with Python. It enables efficient handling of various data structures, a requirement for managing animal shelter records. Dash was selected for building the dashboard due to its capacity to create interactive web-based data visualizations and compatibility with Python libraries such as Pandas and Plotly.

Project Workflow

The project proceeded through the following stages:

1. MongoDB Setup: Establishing a connection to MongoDB using the “pymongo” driver to facilitate interactions with the animal shelter database.

2. Dashboard Design: Designing the dashboard layout using Dash's HTML and Core components, incorporating interactive features.

3. Interactivity: Implementation of Dash callbacks to enable data filtering, chart updates, and map visualizations based on user interactions.

4. Chart Creation: Utilizing Plotly Express to generate a histogram chart illustrating the distribution of animal types.

5. Map Visualization: Integrating Dash Leaflet to create an interactive map featuring markers that pinpoint animal locations.

6. Testing and Deployment: Rigorous testing of the dashboard on a local environment, followed by deployment using the “app.run\_server” method.

Challenges and Solutions

Throughout the project, challenges emerged and were addressed:

- Data Presentation: Designing an effective data representation for user-friendly display demanded thoughtful consideration. Leveraging Dash components, tables, and charts were created to convey information efficiently.

- Map Integration: Integrating the interactive map with accurate markers posed a challenge, which was overcome by employing Dash Leaflet to achieve the desired interactivity.