**Thomas Martin**

**CS-340-11221-M01 Client/Server Development 2024**

**7-2 Project Two Submission**

**Southern New Hampshire University**

**June 20, 2024**

**About the Project/Grazioso Salvare Animal Rescue Project**

Company Overview: Grazioso Salvare is an innovative international animal rescue training company. They specialize in identifying and training dogs for search-and-rescue operations. These dogs are trained to find and rescue humans or other animals in life-threatening conditions.

Project Objective: The primary objective of this project is to develop a software application that can work with existing data from animal shelters to identify and categorize dogs suitable for search-and-rescue training. The application will provide a user-friendly dashboard for Grazioso Salvare's users to interact with and visualize the data.

**Project Phases**

**Phase 1: Database and CRUD Operations**

**Database Selection:**

MongoDB is a NoSQL database, meaning it does not require a fixed schema. This flexibility allows for storing various types of data without requiring predefined structures, making it ideal for handling diverse and dynamic datasets like the ones from animal shelters. MongoDB can easily scale horizontally by adding more servers to handle increased data load, which is particularly useful for Grazioso Salvare as the volume of data from different shelters grows over time.

The pymongo library provides a seamless interface between MongoDB and Python, allowing for efficient database operations directly from Python scripts. MongoDB's BSON format (Binary JSON) aligns well with Python's native data types, facilitating smooth data exchange and manipulation.

MongoDB's powerful query language supports complex queries, aggregations, and filtering. This capability is essential for retrieving specific subsets of data needed for the interactive dashboard. Indexing in MongoDB ensures that queries are executed quickly, improving the performance of data retrieval operations.

A screenshot of a computer

Description automatically generated

**CRUD Operations:**

Created a Python module for performing CRUD (Create, Read, Update, Delete) operations on the MongoDB database.

This module allows easy interaction with the database, ensuring efficient data management.

A screen shot of a computer program

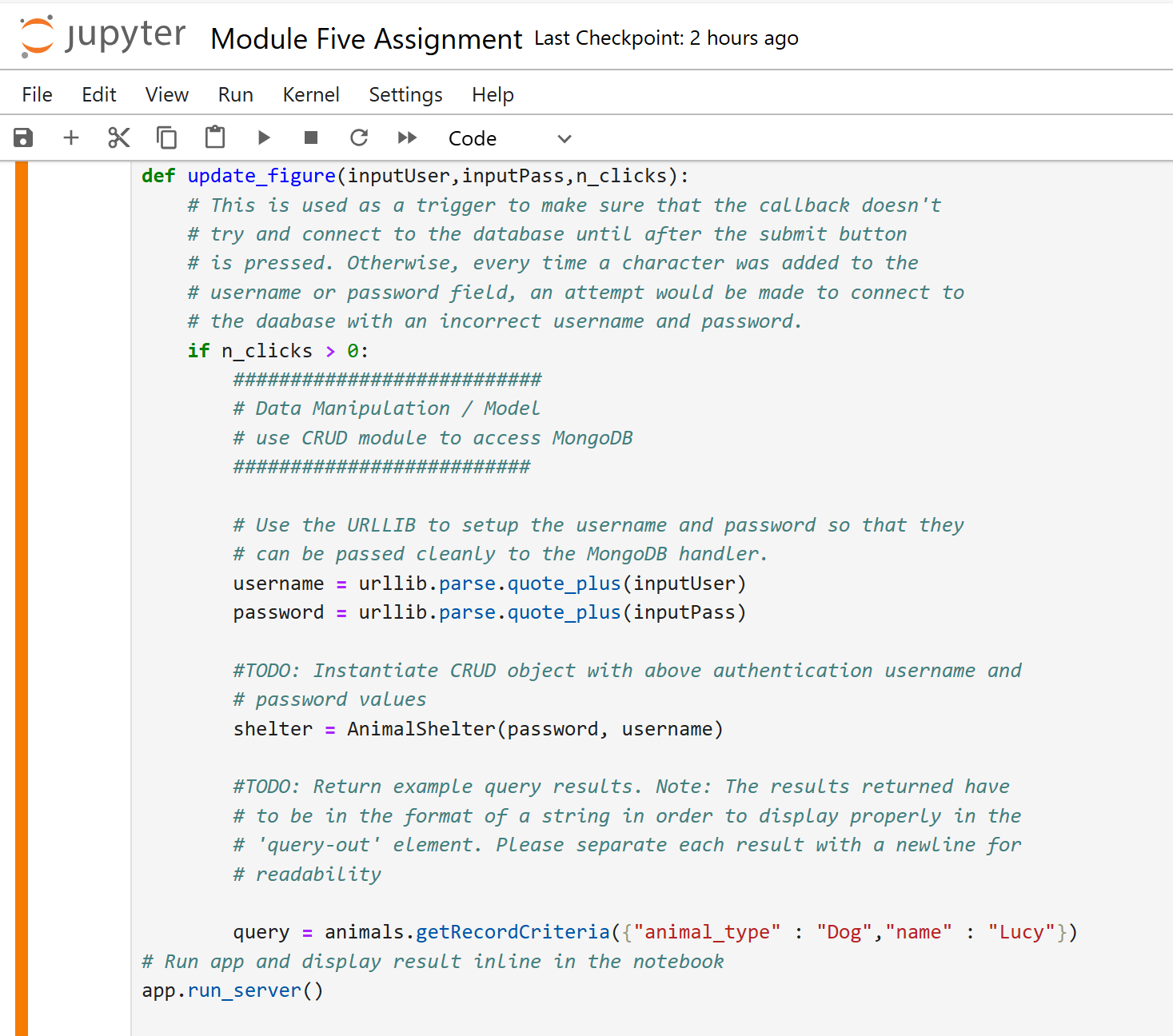
Description automatically generated

A screen shot of a computer program

Description automatically generated

A white rectangular object with a white stripe

Description automatically generated



**Installation**

To use this software, the following tools must be installed:

Python 3.x: Ensure you have Python 3.x installed on your machine. You can download it from the [official Python website](https://www.python.org/downloads/)

pip: Python's package installer should be installed along with Python. Verify its presence by running pip --version in your terminal. If it is not installed, you can download and install it from the [official pip website](https://pip.pypa.io/en/stable/installation/).

MongoDB: Install MongoDB on your machine or set up a cloud instance. Instructions for installing MongoDB can be found on the [MongoDB website](https://www.mongodb.com/try/download/community).

Python Libraries: Install the required Python libraries by running the following command in your terminal: pip install pymongo pandas dash dash-bootstrap-components

CSV Data File: Ensure the Austin Animal Center Outcomes CSV file is available in the project directory. This file will be used to populate the MongoDB database. You can download this file from the supporting materials provided.

**Web Application Dashboard**

After successfully completing previous milestones and Project One, you have created a database and established successful CRUD routines in Python for MongoDB. For Project Two, you will now create a fully functional MongoDB dashboard. This will allow the client, Grazioso Salvare, to interact with and visualize the database.

**Dash Framework**

Dash, a Python framework, is utilized for creating the dashboard. Dash provides a comprehensive framework for building web applications, particularly data-driven dashboards. It seamlessly integrates with popular Python libraries such as Flask, Plotly, and Pandas, making it an ideal choice for this project. Dash simplifies the development process by providing built-in components and abstracting much of the complexity associated with web application development.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Dashboard Features**

The MongoDB dashboard for Grazioso Salvare includes the following features:

Data Table with Custom Filters: This table displays information about the animals at the Austin Animal Shelter. Custom filters enable targeted selection of animals for specific business needs, including:

Water Rescue

Mountain or Wilderness Rescue

Disaster Rescue or Individual Tracking

Pie Chart: A visual representation showing the breakdown of available breeds.

Map: Displays the location of a selected animal.

These features are implemented using the Dash framework, providing an interactive and user-friendly interface for the client to interact with and visualize the data.

A screenshot of a computer

Description automatically generated

**Disaster Rescue or Individual Tracking**

A screenshot of a computer screen

Description automatically generated

**Mountain or Wilderness Rescue**

A screenshot of a computer

Description automatically generated

**Water Rescue**

A screenshot of a computer

Description automatically generated

**Reset**

A screenshot of a computer

Description automatically generated

#### **Challenges Encountered**

During the development of the dashboard, several challenges were encountered and addressed

Integrating MongoDB with Dash for seamless data retrieval and updating required careful design and implementation.

Integrating the Grazioso Salvare logo into the MongoDB with Dash

A dog with a black background

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated