Jean Agnus  
CS 499 Milestone Three Enhancement Narrative  
Professor: Nembhard  
5/28/25

Enhancement Overview:

For this milestone, I enhanced my CS 340 Grazioso Salvare Dashboard project by adding a **Binary Search Tree BST** to it for better search features on my application. This replacement provided a more optimal algorithmic addition, enabling faster access to animal records by name.

Rationale for Enhancement:

The animal save dashboard was nontrivial in that it was also built with MongoDB and Dash but had no sophisticated search algorithms or clever data structures. What I had done in response to some feedback from Milestone One about “Using meaningful data structures” was to create a Binary Search Tree, and it had really helped to up the performance and complexity of the game. Searching in a BST takes O(log n) time, so it is faster on large sets than linear search.

Implementation Summary:

* **Created a new module (bst\_module.py)** that defines a BinarySearchTree and Node class.
* **Modified dashboard.py** to construct a BST during loading:
  + On every node, an animal name is the key and its full record is the value.
* **Added a search input and button** to the Dash UI that allows users to search for animals by name.
* **Updated the update\_table() callback** so that if a name is entered, the BST is searched instead of the DataFrame.
* Kept everything else the same for the core dashboard functionality (charts, maps, filters) to give backward compatibility and for usability sake.

Improvements Achieved:

* **Performance**: Faster name search since it uses the BST
* **Scalability**: This structure shines when datasets are larger than can be supported by linear scans.
* **Modularity**: The enhancement follows clean code principles by isolating logic into its own module.
* **User Experience**: The new name search capability is a plus for usability and play levels.

Conclusion:  
  
In the context of real-world software these are the implementation of an algorithm, and I believe this improvement shows my mastery of algorithmic thought and applying data structure to real software. It is a proof that I can pinpoint performance bottlenecks, introduce computer science to the problem, and clean legacy code which is necessary for developer roles on a higher level.