**Artifact 1: Software Design and Engineering**

The Grazioso Salvare Dashboard is a web-based application initially developed to present animal rescue data interactively. Built using the Dash framework and hosted within a Jupyter Notebook environment, the dashboard allows users to filter and visualize animal rescue data through various interactive elements. However, the original implementation was limited in scope, particularly in its design and engineering aspects. The code was monolithic, making it difficult to maintain and extend. This artifact was selected to enhance my software design and engineering proficiency by improving the code’s structure, documentation, and overall usability.

**Original Implementation:** While the Grazioso Salvare Dashboard was functional, it faced significant challenges. The lack of a clear separation of concerns and detailed documentation and the absence of robust error-handling mechanisms made the codebase difficult to manage and extend. These challenges underscored the need for the enhancements that were subsequently made.

**Enhancements Made:** To address these issues, I implemented several key enhancements focused on software design and engineering principles:

1. **Separation of Concerns:** I refactored the code to separate data processing, UI layout, and interactivity into distinct sections. This change improved the maintainability of the code by isolating different functionalities, making the code easier to understand and modify.
2. **Improved Documentation:** I didn't just add comments and docstrings, I added comprehensive ones throughout the code, explaining the purpose of each section and function in detail. This documentation provides valuable context for future developers and users, enhancing the overall readability and maintainability of the code and ensuring that everyone is well-informed and confident in their work.
3. **Enhanced Error Handling:** I didn't just incorporate error handling mechanisms, I made sure they were robust and comprehensive, particularly where user input could cause issues. This includes validating inputs and ensuring that the application fails gracefully in case of errors, thereby improving the software's reliability and ensuring that users feel secure and confident in its performance.

**Technical Explanation:** The refactoring process involved organizing the code into modules, each responsible for a specific dashboard aspect. The data processing logic was encapsulated in a separate function, while the UI components were handled independently. To catch and handle potential errors, error handling was added by incorporating try-except blocks around critical operations, such as data loading and user input processing.

**Outcomes Achieved:** These enhancements align with several key course outcomes, particularly in software design and engineering. The improvements demonstrate my ability to design modular, maintainable, and robust software, focusing on improving code quality and user experience. By implementing these changes, I have shown proficiency in structuring code for better maintainability and reliability, which are crucial skills in software engineering.

The enhancements made to the Grazioso Salvare Dashboard significantly improved its software design and engineering aspects. The code is now better organized, more maintainable, and easier to extend, with improved documentation and error handling. This artifact is a strong example of my software design and engineering capabilities, demonstrating my ability to transform a functional but flawed implementation into a well-structured and reliable application.