CS 499 Milestone Four: Database Enhancement Narrative

Name: Femi Abdul
Course: CS 499
Assignment: Milestone Four – Databases

Artifact Narrative: Pirate Intelligent Agent – Database Enhancement

The artifact I selected for the Databases enhancement is the Pirate Intelligent Agent project, which was originally developed during CS 370: Artificial Intelligence. This artifact simulates an agent navigating a maze to reach a treasure while avoiding traps, utilizing reinforcement learning with a Q-learning algorithm. The original implementation stored Q-values only in memory during the training phase, meaning the agent would lose all learned experience after every run.

I chose this artifact because it offered a natural opportunity to incorporate persistent storage of training data. The enhancement showcases my ability to store, retrieve, and manage structured data by simulating a Q-table database. While the project does not use a full-fledged SQL database due to environment limitations, I implemented Q-table persistence through JSON file operations to replicate core database functionality. These additions demonstrate core database design principles such as data serialization, schema-like structuring, read/write operations, and modular access functions.

The artifact was improved by introducing two key functions: `save_q_table()` and `load_q_table()`. These enable the agent to store and reload its Q-values across sessions, effectively simulating persistent storage. Additionally, I restructured the code to separate the training logic from the data storage layer, which follows good design principles and enhances code reusability. I also added exception handling and basic validation to ensure the integrity of the saved data.

This enhancement aligns with the CS 499 course outcome of using innovative tools and techniques in computing practices. I demonstrated the ability to manipulate structured data and implement persistence mechanisms. The use of JSON allowed me to showcase database-like logic in a constrained environment, while still achieving my project goal.

Through this process, I learned how persistent storage affects agent performance, especially in long-term learning. One of the challenges I faced was simulating database behavior without an actual SQL engine. I overcame this by using Python's `json` module to structure data similarly to key-value storage systems. Testing the accuracy of stored Q-values after reloading required careful validation of state-action pairs.

Overall, this enhancement allowed me to evolve the project from a single-run learning model into a persistently improving system, reflecting real-world AI deployment scenarios. The modular

Q-table class, combined with data saving and loading capabilities, demonstrates my understanding of database concepts and my ability to adapt them to machine learning contexts.