AI Integration & Reflection

To refine this project, I used Microsoft Copilot as a supportive tool.

First, I asked:How can I improve my feeding algorithm to handle errors and alerts? Copilot suggested dynamic thresholds, multiple monitoring intervals, and a feedback mechanism for servo malfunctions. These recommendations helped me see flaws I had overlooked , for example, how a small pet eating just 2–3g might be misclassified as not eating, or how a motor failure could go undetected.

Next, I explored implementation by asking:What hardware would I need to build this with Arduino or Raspberry Pi? Copilot outlined servo motors, weight sensors, and possible coding frameworks. This shifted my perspective from a purely theoretical design to a practical system, making me consider issues such as sensor calibration, power reliability, and data storage.

Beyond the technical details, Copilot also helped me present my work more clearly. It guided me in restructuring my pseudocode with modular steps, improved naming conventions, and more precise comments, making the logic easier to follow.

Overall, AI acted like a second reviewer rather than a replacement for my own thinking. It pushed me to critically evaluate my initial design, broaden my awareness of real-world constraints, and polish my documentation. This experience showed me how AI can serve as both a problem-solving assistant and a professional development tool, deepening my understanding of how to design reliable, real-world automated systems.