

AI Integration Reflection (Step 5)

To enhance my Automated Pet Feeder project, I used Microsoft Copilot to assist with refining my Step 4 implementation and exploring real-world hardware integration. I prompted Copilot to review my task sequence and identify potential issues, asking: "Can you suggest improvements for this step-by-step feeding logic, including error handling and alerts?" Copilot recommended adding multiple monitoring intervals for bowl weight, dynamic thresholds for small pets, and alerts for mechanical failures. These suggestions helped me identify weaknesses in my original logic, particularly regarding partial feeding detection and servo malfunction handling.

To improve the decision-making logic in my flowchart, I asked Microsoft Copilot: "Can you propose different ways to structure my feeding system logic or enhance the flowchart?" Copilot suggested using a state machine model to better manage transitions between feeding states, incorporating retry mechanisms for failed dispensing, and adding smart scheduling based on pet behavior. These ideas helped me rethink the rigidity of my current flow and inspired me to explore more modular and adaptive logic structures. As a result, I revised my flowchart to include buffered decision points and more robust error handling.

To understand how my system could be built using actual hardware, I asked Copilot: "How could I implement this system using Arduino or Raspberry Pi with sensors and servo motors?" Copilot provided a clear comparison between the two platforms, outlining the required components such as load cells, RTC modules, and servo motors. It also explained how to structure the code for each platform and suggested a hybrid approach for combining real-time control with cloud-based logging. This guidance helped me visualize the physical setup and consider practical challenges like sensor calibration, power supply, and connectivity.

To professionally document my project, I asked Copilot: "Can you help me write a README.md file and summarize my project for presentation?" Copilot generated a clean, structured README with sections for overview, features, hardware, logic, and future enhancements. It also provided a concise presentation summary that

highlighted the problem, solution, and impact of my system. These resources helped me present my work more clearly and professionally, ensuring that both technical and non-technical audiences could understand the value of my project.

To explore the broader implications of my project, I asked Copilot: “What are the ethical concerns of using AI in automated pet care, and how reliable are AI-generated suggestions?” Copilot explained that while AI can improve efficiency, it must be designed with empathy and accountability. It emphasized the importance of human oversight, data privacy, and adaptive logic to avoid misjudgments. This reflection deepened my understanding of responsible AI design and encouraged me to include safeguards and transparency in my system. It also reinforced the need to treat AI as a support tool rather than a replacement for human care.