

## Part 3: AI Integration

### Prompt:

I have a system that is an automatic pet feeder. I have done the flow chart of the same and completed the Word coding, which is as follow Time Reader displays if it is feeding time. If it is feeding time, check if there is enough food in stock, If there is not, trigger low food alert Dispense pre-decided amount of food If the Food is not dispensed, trigger an alert regarding the issue with the Motor Meseure the bowl weight after a while, to see if the food is eaten by pets, if it is not, trigger an alert that Food is not eaten Display Feeding Complete I want your assistance to enhance the word coding with one statement each

### Output

#### Enhanced Word Coding – Automatic Pet Feeder

1. Check Feeding Time → Time Reader checks and displays whether it's the scheduled feeding time.
2. Verify Food Stock → If it is feeding time, check if food stock is sufficient.
3. Trigger Low Food Alert → If food stock is low or empty, activate the low food alert and halt feeding.
4. Dispense Food → If food is available, dispense the pre-decided quantity using the motor.
5. Check Dispensing Status → If food is not dispensed successfully, trigger a motor malfunction alert.
6. Monitor Consumption → After a set delay, measure bowl weight to determine if food was eaten.
7. Trigger Uneaten Food Alert → If bowl weight remains unchanged, trigger a food-not-eaten alert.
8. Complete Feeding Cycle → Display "Feeding Complete" message and log the event.

## Prompt:

I have uploaded the files into Git Repo. I have 5 steps in total. Problem Solving, Organising Data and describing, Flowchart, Word coding and Testing & Refining. I need your assistance in creating a README.md file for the repository, so anybody opening the repository can understand how to navigate through it

## Output

### # 🐾 Automatic Pet Feeder System

Welcome to the repository for an Automatic Pet Feeder, a smart solution designed to automate pet feeding with intelligent alerts and monitoring. This project walks through complete development life-

#### 📁 Repository Structure

1. 🧠 **Problem Solving:** Define the core problem and outline goals of the automatic pet feeder: ~\Contents: Problem statement, user needs, and system expectations.
2. 📊 **Organising Data and Describing:** Identify and describe the key components and data involved: inputs (i.e. time, food stock), outputs, and System variables. System variables.
3. 📈 **Flowchart:** Visualize the logical flow of system. If a step-by-step diagram showing decision points, actions, and alerts.
4. 📝 **Word Coding:** Translate the flowchart into structured pseudo-code. Enhanced word coding with clear logic and alert mechanisms. Contents: Test cases, ob-
5. 🧪 **Testing & Refining:** Simulate and evaluate system behavior, identify issues, refining logic. Test cases, observations, and improvements made.

#### 🚀 Getting Started

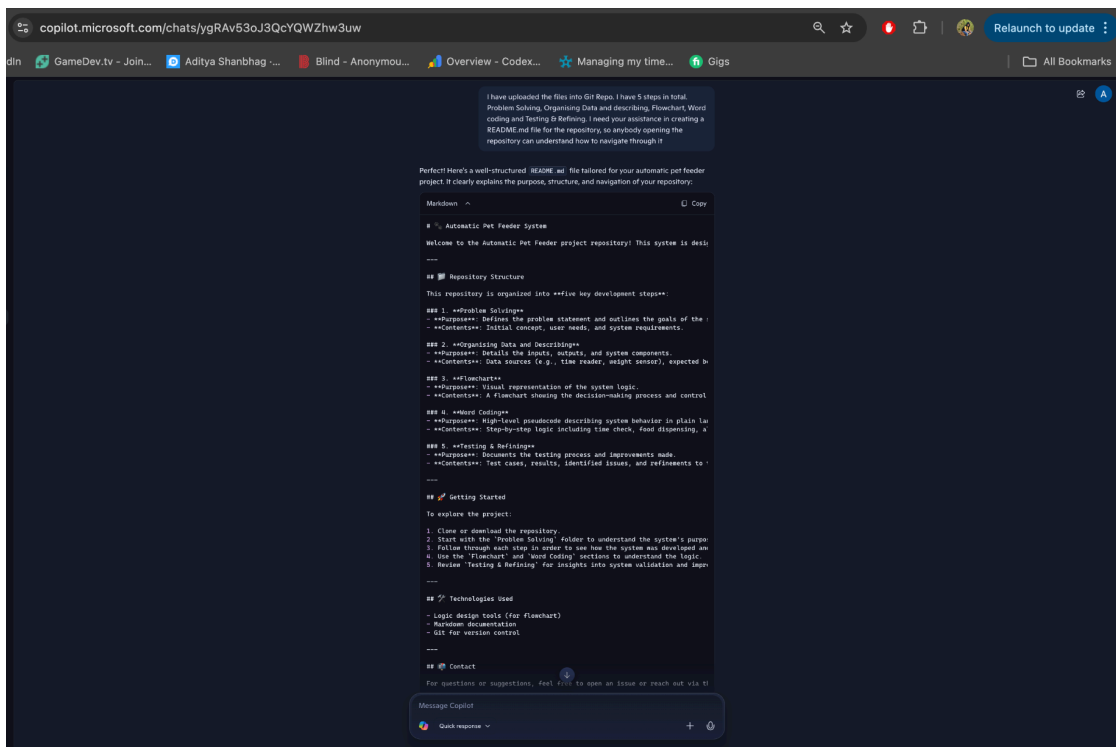
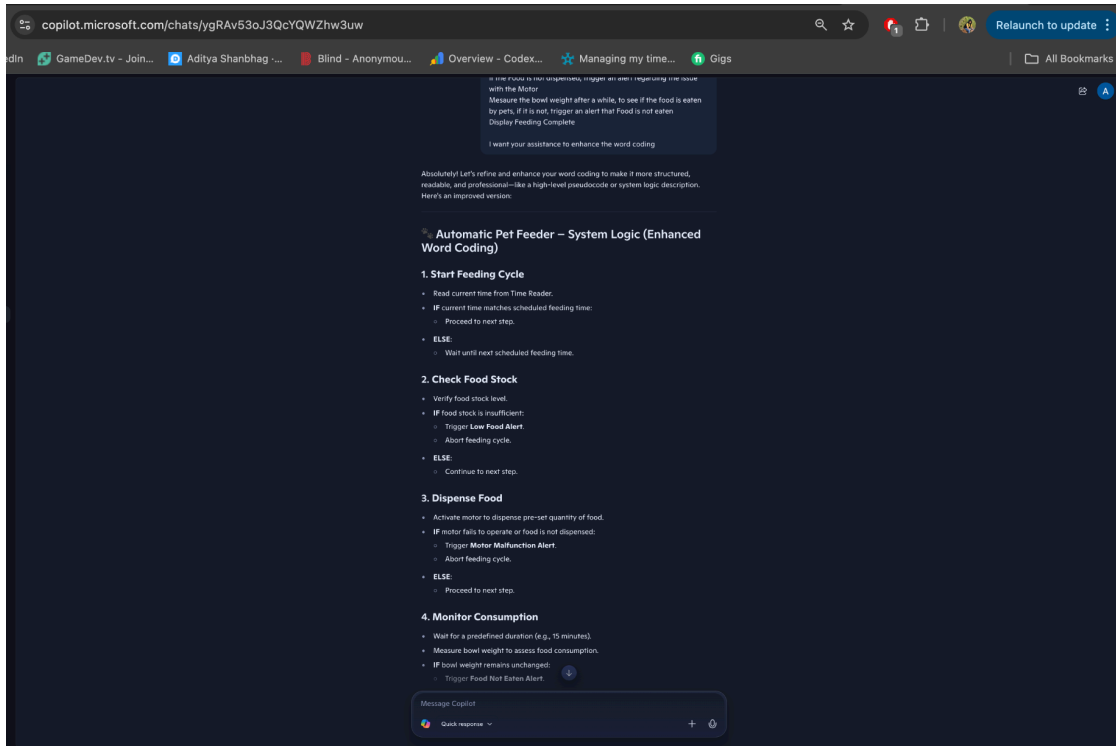
To explore the project:

1. Clone the repository. Start with the "Problem:" solving the system's purpose.
2. Start with the 'Problem Solving' folder to understand the system's sequentially to see how the idea evolves into final.
3. Review 'Testing & Refining' to understand how the system is validated.

#### 🌟 Highlights

- Smart alert system for low food and motor issues.
- Consumption monitoring using bowl weight.
- Modular design for easy upgrades and sensor integration.

# Screenshots of the conversation with the co-pilot



## **Insights & Influence of CoPilot**

The AI agent has assisted significantly in enhancing my logical reasoning by suggesting enhancements such as the introduction of retry mechanisms, which has made the code more efficient and reliable in general.

Moreover, its significance of the concept of coding provided me with a clearer concept of how to portray the inherent logic in a structured and comprehensible manner.

This not only reinforced my knowledge of programming practice but also enabled me to write a readable and explanatory README file for my GitHub repository.