Complete Multi-Agent System Code: Explanation

The code includes **7 intelligent agents** as specified in your project plan:

1. **Supervising Agent** - Central coordinator with local LLM integration
2. **Vision Agent** - Computer vision for mahout recognition and obstacle detection
3. **Audio Agent** - Speech recognition and text-to-speech
4. **Movement Agent** - Controls leg servos for locomotion
5. **Gesture Agent** - Controls expressive movements (trunk, ears, eyes, tail)
6. **Touch Agent** - Handles force-sensing resistors for touch feedback
7. **RFID Agent** - Manages RFID-based mahout identification

Key Features Implemented

- **ROS-based communication** between all agents
- **Local LLM integration** using Ollama for real-time command interpretation
- **Hybrid AI approach** (local + cloud LLM capability)
- **Multi-sensor fusion** (vision, audio, touch, RFID)
- **Servo control** for 15+ different movements
- **Real-time obstacle avoidance**
- **Mahout recognition system**
- **Command mapping and interpretation**

Setup Guide Highlights

The comprehensive setup guide covers:

- **Hardware requirements** and wiring diagrams
- **Software installation** (ROS, OpenCV, LLM tools)
- **GPIO configuration** for all sensors and actuators
- **Calibration procedures** for servos and sensors
- **Testing and debugging** tools
- **Auto-start configuration** for production use
- **System monitoring** and maintenance

Quick Start Instructions

1. **Install the base system** following the setup guide
2. **Wire all hardware** according to the GPIO pin assignments
3. **Copy the agent code** to your Raspberry Pi

4. **Build the ROS workspace** with catkin_make
5. **Launch the system** with roslaunch ai_elephant elephant_system.launch

Hardware Connections Summary

- **15-20 servo motors** for movement and gestures
- **5 touch sensors** (FSRs) for interaction
- **RFID reader** for mahout identification
- **Camera module** for vision processing
- **USB microphone** for voice commands
- **Proximity sensors** for obstacle avoidance

The system is designed to be modular - you can test each agent independently and gradually integrate them. The code includes error handling, logging, and safety features throughout.