**Multi-Agent Chat System**

**System Components (As Required)**

**1) Coordinator Agent (Manager)**

* Receives user queries
* Analyzes complexity and determines required agents
* Routes tasks to worker agents
* Merges results and maintains conversation context

**2) Specialized Worker Agents (All Three Implemented)**

* **Research Agent**: Simulates information retrieval using pre-loaded knowledge base
* **Analysis Agent**: Performs comparisons, reasoning, and calculations on retrieved data
* **Memory Agent**: Manages long-term storage, retrieval, and context updates

**3) Enhanced Memory System**

* **Conversation Memory**: Full history with timestamps and metadata
* **Knowledge Base**: Persistent store of facts with provenance
* **Agent State Memory**: Tracks what each agent accomplished per task
* **Search/Retrieval**: Keyword search plus vector similarity

**How to Run**

**Google Colab (Easiest):**

1. Copy code into one cell
2. Run cell
3. Watch all 5 test scenarios execute

**Local:**

python multi\_agent\_system.py

**Sample Test Scenarios (All 5 Included)**

1. **Simple Query**: "What are the main types of neural networks?"
2. **Complex Query**: "Research transformer architectures, analyze their computational efficiency..."
3. **Memory Test**: "What did we discuss about neural networks earlier?"
4. **Multi-step**: "Find recent papers on reinforcement learning, analyze methodologies..."
5. **Collaborative**: "Compare two machine-learning approaches and recommend..."

**Core Requirements ✅**

**A. Agent Communication & Coordination**

* Message passing between agents via Coordinator
* Sequential calls based on dependencies
* Support for multi-agent collaboration

**B. Memory with Context Awareness**

* Structured records with timestamps, topics, source, confidence
* Keyword search implementation
* Memory influences decisions and avoids redundant work

**C. Enhanced Decision-Making**

* Coordinator performs complexity analysis
* Error handling and fallback strategies
* Confidence scoring for agent outputs

**Architecture Flow**

User Query → Coordinator Agent → Worker Agents → Enhanced Memory System

↓

┌─────────────────────────────────────┐

│ Coordinator │

│ (Analyzes & Routes Tasks) │

└─────────────┬───────────────────────┘

│

┌─────────────┼─────────────────────┐

│ │ │

┌───▼──┐ ┌────▼───┐ ┌─────▼─────┐

│Research│ │Analysis│ │ Memory │

│ Agent │ │ Agent │ │ Agent │

└────────┘ └────────┘ └───────────┘

**Key Features**

* **Task Decomposition**: Coordinator implements simple planner
* **Inter-Agent Communication**: Direct method calls with message payloads
* **Logging/Tracing**: Prints agent calls and coordination decisions
* **Structured Memory**: Vector search with metadata storage
* **Confidence Scoring**: Agent outputs include confidence levels