

# MULTI-AGENT RESEARCH ASSISTANT

## Assignment Submission

**Pranav Pillai**

Email: [ppillai294@gmail.com](mailto:ppillai294@gmail.com)

Phone: +91 91756 50166

---

## SETUP INSTRUCTIONS

### 1. Open in Google Colab (RECOMMENDED)

- Upload the `.ipynb` file to Google Colab
- Add API keys in Colab Secrets (key icon in left sidebar):
  - `SERPER_API_KEY` (get free at: [serper.dev](https://serper.dev))
  - `GEMINI_API_KEY` (get free at: [ai.google.dev](https://ai.google.dev))

### 2. Run All Cells

- Click **Runtime > Run all**
- Wait 5-10 minutes for complete execution
- Demo queries will execute automatically
- Results are displayed with memory statistics

### 3. API Key Setup (if not using Colab Secrets)

- You'll be prompted to enter keys manually
  - API keys are free tier for both services
- 

## FILE STRUCTURE

### 1. Research\_Document.docx

- Context engineering theory and principles
- Framework comparison (CrewAI vs AutoGen vs LangGraph)
- Architectural decisions and justification

## **2. Multi\_Agent\_Implementation.ipynb**

- Complete working multi-agent system
- 5 demo queries with full results
- Memory system visualization
- Performance metrics

## **3. README.txt (this file)**

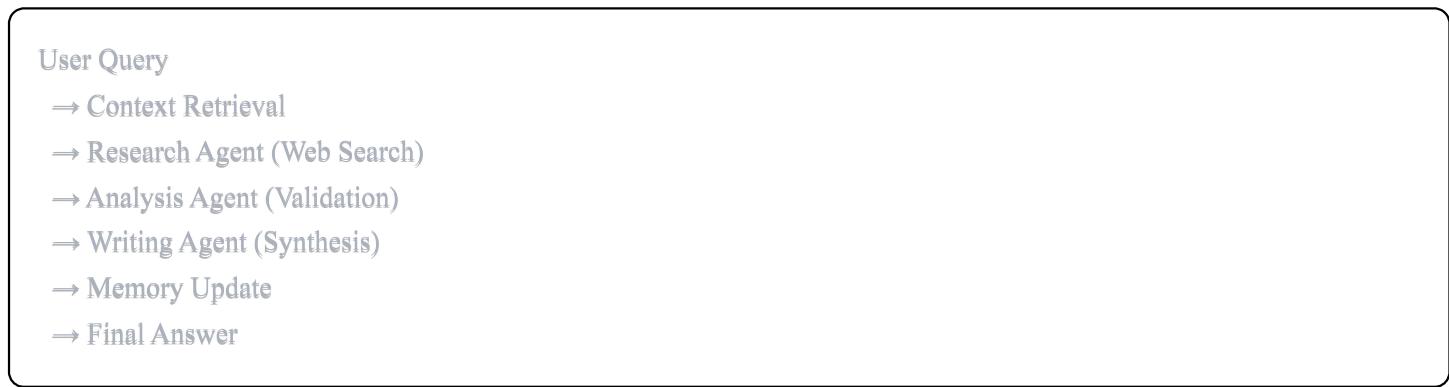
- Setup and execution guide
- 

# **SYSTEM ARCHITECTURE**

## **Components:**

- **Context Engineering Module** (Short-term + Long-term memory)
- **3 Specialized Agents** (Research, Analysis, Writing)
- **Vector Database** (ChromaDB with semantic search)
- **Web Search Integration** (SerperDev API)
- **LLM Backend** (Google Gemini 1.5 Flash)

## **Workflow:**



# **KEY FEATURES DEMONSTRATED**

## **✓ Context Engineering**

- Rolling window short-term memory (10 messages)
- Vector-based long-term memory with semantic retrieval
- Dynamic context assembly based on query relevance

## ✓ Retrieval-Augmented Generation (RAG)

- Prevents hallucinations through web search
- Real-time information beyond training cutoff
- Source attribution and citation

## ✓ Multi-Agent Collaboration

- **Research Agent:** Information gathering
- **Analysis Agent:** Validation and synthesis
- **Writing Agent:** Final output generation
- Sequential process with shared context

## ✓ Memory Management

- Automatic storage of important information
- Semantic search for relevant historical context
- Memory statistics tracking

## ✓ Production-Ready Features

- Error handling and retry logic
  - Rate limiting compliance
  - Scalable architecture
- 

## DEMO QUERIES INCLUDED

### Query 1: "What is CrewAI?"

- Framework explanation with sources
- Demonstrates basic web search integration

### Query 2: "Explain retrieval-augmented generation (RAG)"

- Technical concept explanation
- Shows source citation and synthesis

### Query 3: "What are the latest developments in multi-agent AI systems?"

- Current research trends

- Demonstrates handling of recent information

#### **Query 4: "What is context engineering in AI systems?"**

- Concept definition with examples
- Shows memory retrieval working

#### **Query 5: "Compare CrewAI and AutoGen frameworks"**

- Comparative analysis
  - Demonstrates multi-source synthesis
- 

## **EXPECTED OUTPUT**

Each query produces:

- Web search results with URLs
- Analysis of information quality
- Synthesized answer with citations
- Memory statistics update

### **Performance:**

- Total execution time: ~5-8 minutes for all demos
  - Memory usage: ~6 items stored in long-term memory
- 

## **TECHNICAL SPECIFICATIONS**

### **Dependencies:**

- crewai + crewai-tools
- chromadb (vector database)
- sentence-transformers (embeddings)
- langchain + langchain-huggingface
- google-generativeai (Gemini)
- serper API (web search)

## **Requirements:**

- Python Version: 3.10+
  - GPU: Optional (T4 recommended in Colab)
  - RAM: 12GB+ recommended
- 

## **TROUBLESHOOTING**

Issue	Solution
"API key not found"	Add keys to Colab Secrets or enter manually when prompted
"Rate limit exceeded"	Wait 60 seconds between query batches (already implemented)
"Tool execution failed"	Check internet connection and API key validity
"Memory errors"	Restart runtime and run all cells fresh

---

## **DEMO MEETING PREPARATION**

### **Ready to Demonstrate:**

1. Live query execution (2-3 examples)
2. Memory system visualization
3. Agent collaboration workflow
4. Code walkthrough of key components

### **Questions to Expect:**

- Why CrewAI over alternatives?
  - How does RAG prevent hallucinations?
  - Scalability considerations
  - Production deployment strategy
- 

## **CONTACT INFORMATION**

**Pranav Pillai**

Email: [ppillai294@gmail.com](mailto:ppillai294@gmail.com)

Phone: +91 91756 50166

**Assignment Details:**

Completed: November 2024

Framework: CrewAI (MIT License)

LLM: Google Gemini 1.5 Flash

---

**End of Document**