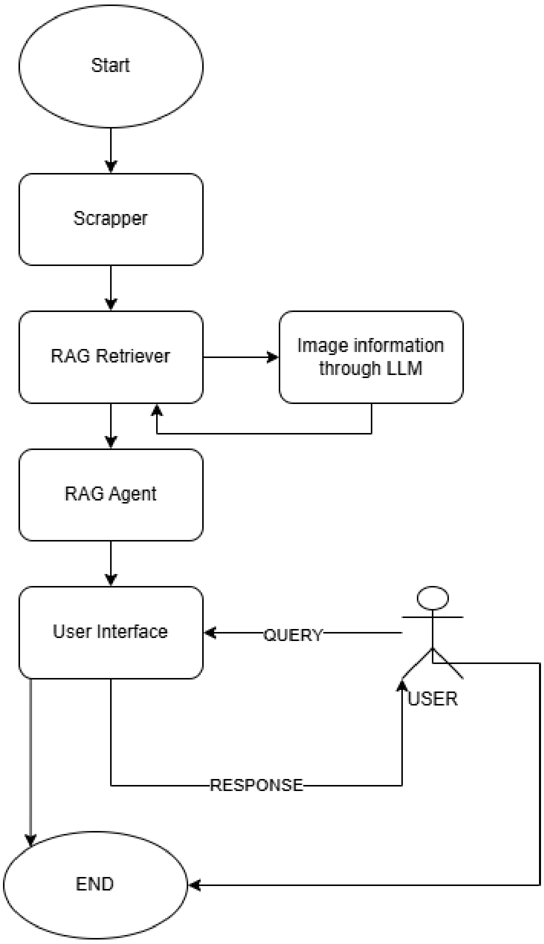
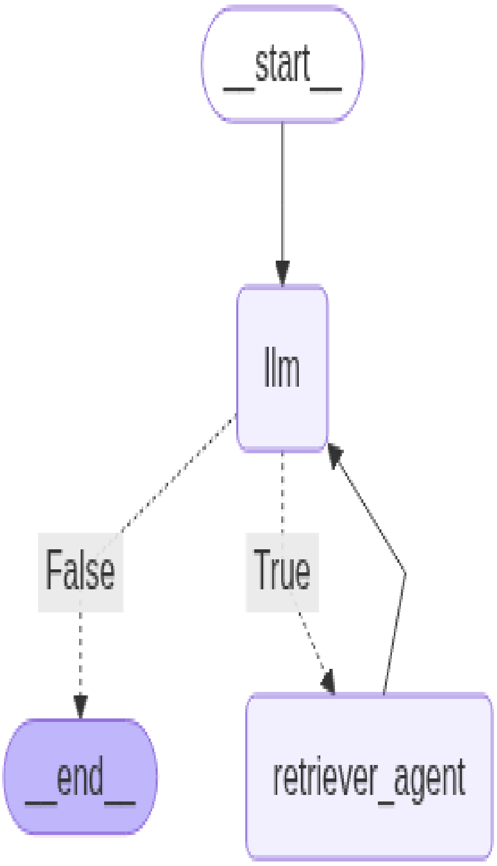
RAG Research Assistant for AI Literature Analysis

🏗️ System Architecture & Workflow





End-to-End Process Flow:

1. START → SCRAPER → Automated content extraction from research papers
2. RAG RETRIEVER → Document processing, chunking, and vector embedding
3. IMAGE INFORMATION → Multi-modal analysis via vision-language models
4. RAG AGENT → Intelligent query processing and response generation
5. USER INTERFACE → Interactive chat experience with streaming responses
6. END → Complete research analysis cycle

🤖 AI Models & Core Technologies

Primary AI Models

Component	Model/Service	Purpose
Language Model	Azure OpenAI GPT-4o	Primary reasoning, response generation, and complex query understanding
Embeddings	text-embedding-3-small	Document vectorization and semantic similarity matching

Component	Model/Service	Purpose
Vision Model	GPT-4o Vision	Image analysis and description generation for multi-modal content

Core Framework Stack

Layer	Technology	Function
Agent Framework	LangGraph	Multi-step reasoning, state management, and tool orchestration
LLM Integration	LangChain	Model abstraction, prompt engineering, and tool calling
Vector Database	ChromaDB	High-performance similarity search and document storage
Web Interface	Streamlit	Interactive chat interface with real-time streaming
Web Scraping	FireCrawl	Professional content extraction from research websites
Document Processing	Unstructured	Intelligent markdown parsing and text chunking

📊 Technical Architecture Specifications

Data Processing Pipeline

- Chunk Size: 1000 characters with 200-character overlap for optimal context retention
- Retrieval Strategy: Top-20 similarity search with configurable parameters
- Memory Management: Persistent conversation state via MemorySaver
- Streaming: Real-time response generation with 20ms word delays

Infrastructure Components

- Runtime: Python 3.13+ with UV package management
- Containerization: Docker with multi-stage builds and health checks
- Deployment: Docker Compose with nginx proxy support
- Storage: Local filesystem with volume mounting for artifacts

🎯 Specialized Use Cases & Capabilities

Research Analysis Features

- Circuit Analysis: Deep understanding of transformer neural circuits and mechanisms
- Methodology Extraction: Automated identification of experimental approaches
- Performance Metrics: Analysis of model benchmarks and comparative studies
- Safety Implications: Assessment of AI safety and interpretability findings

Multi-Modal Intelligence

- Document + Image Processing: Simultaneous analysis of text and visual research content

- Contextual Understanding: Integration of figures, diagrams, and textual explanations
- Comprehensive Insights: Holistic research paper analysis beyond text-only approaches

🔧 Production-Ready Features

Scalability Considerations

- Modular Design: Independent components for scraping, processing, and querying
- Error Handling: Comprehensive exception management and graceful degradation
- Configuration Management: Environment-based settings for different deployment scenarios
- Health Monitoring: Built-in health checks and logging for production deployment

Security & Reliability

- API Key Management: Secure environment variable handling
- Resource Optimization: Efficient memory usage and processing patterns
- Fault Tolerance: Robust error recovery and user feedback systems