# AI Research Orchestrator - Summary

**Overview**
The AI Research Orchestrator is a multi-agent system that automates end-to-end research workflows. Given a research query, it intelligently plans the task, retrieves relevant information, extracts insights, generates summaries, and produces a structured research report. The system uses stateful orchestration to coordinate multiple specialized AI agents through a shared state object.

**Core Architecture**
The system follows a multi-agent pipeline design, where each agent performs a focused responsibility:

1. **Planner Agent** – Decomposes the user query into research topics, search queries, and analysis steps using an LLM.
2. **Web Search Agent** – Retrieves relevant web results via Serper API, extracts content, and indexes it into a FAISS vector store.
3. **Analyzer Agent** – Extracts structured insights such as key findings, supporting evidence, and sources.
4. **Summarizer Agent** – Produces a 300–500 word academic-style summary.
5. **Report Generator Agent** – Generates a structured markdown research report with predefined sections.

**Technology Stack**
- ✓ **LangGraph** – Stateful orchestration with conditional routing and retry logic.
- ✓ **OpenAI API** – GPT-4 for reasoning and text generation; text-embedding-3-small for embeddings.
- ✓ **FAISS** – In-memory vector store for semantic similarity search.
- ✓ **Serper API** – Web search integration.
- ✓ **FastAPI** – REST API backend.
- ✓ **Streamlit** – Conversational chat-based UI.
- ✓ **Pydantic** – Strongly typed shared state model (ResearchState).

**Execution Flow**
User Query → Planner → Web Search → Analyzer → Summarizer → Report Generator → Markdown Output

Conditional logic ensures:
- ➢ Automatic retry if search results are empty.
- ➢ Pipeline termination if no insights are extracted.
- ➢ Clear state transitions (pending → running → completed/error).

**State Management**
A centralized ResearchState object flows across all agents and contains:

- ✓ User query
- ✓ Research topics and search queries
- ✓ Search results (title, source, content, URL)
- ✓ Extracted insights (finding, evidence, source)
- ✓ Summary and final report
- ✓ Status and error fields

**Output**
The system generates a structured markdown report with the following sections:

1. Introduction
2. Background
3. Key Findings
4. Trends
5. Challenges
6. Conclusion
7. References

Reports are saved to the output directory and execution logs are stored for monitoring and debugging.

**Key Features**

- ✓ Intelligent LLM-based planning
- ✓ Web search + content extraction integration
- ✓ FAISS-based vector indexing for RAG capability
- ✓ Structured academic-style output
- ✓ Retry and error-handling logic
- ✓ Comprehensive logging support

**Use Cases**

- ✓ Academic research automation
- ✓ Market and competitive analysis
- ✓ Technical research synthesis