

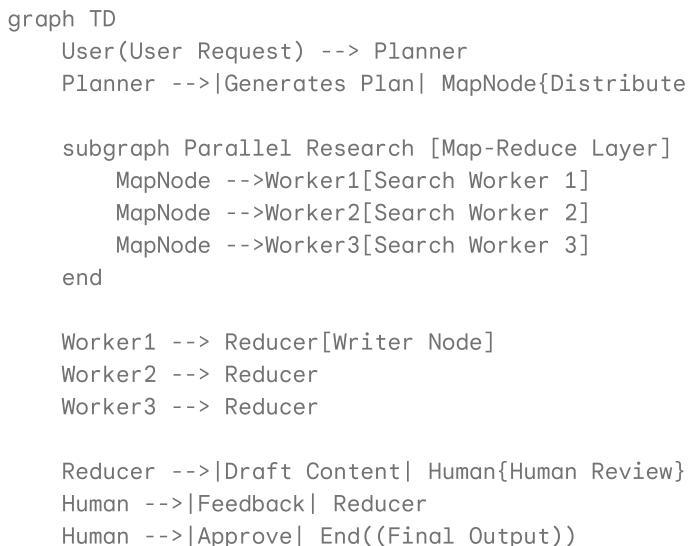
Nexus: Deep Research & Content Architect

An autonomous agent that thinks, plans, researches in parallel, and adapts to you.

Nexus is a production-grade Deep Agent built with **LangGraph**. Unlike standard chatbots, Nexus employs a **Map-Reduce architecture** to perform parallelized research, maintains **long-term memory** of user preferences, and includes a **Human-in-the-Loop** workflow for quality assurance.

Architecture

Nexus operates on a cyclic graph designed for complex reasoning tasks.



Key Features (Resume Highlights)

- **Distributed Map-Reduce Pattern:** Dynamically spawns parallel worker nodes based on task complexity using LangGraph's `Send` API.
- **Long-Term Memory (Persistence):** Uses a persistent JSON store (extensible to VectorDB) to remember user style preferences across different sessions.
- **Human-in-the-Loop (HITL):** pauses execution before finalization, allowing users to critique and steer the agent's output in real-time.
- **Strict Schema Validation:** Fully typed state management using **Pydantic** to ensure production-grade reliability.
- **State Management:** Utilizes LangGraph's `StateGraph` and `Checkpointers` to handle complex, multi-turn conversational states.

Getting Started

Prerequisites

- Python 3.9+
- OpenAI API Key

- (Optional) Tavily API Key for real web search (Mock mode enabled by default)

Installation

1. Clone the repository:

```
git clone [https://github.com/yourusername/nexus-agent.git](https://github.com/your  
cd nexus-agent
```

2. Install dependencies:

```
pip install -r nexus_agent/requirements.txt
```

3. Configure Environment:

Create a `.env` file in the `nexus_agent` directory:

```
OPENAI_API_KEY="sk-..."  
TAVILY_API_KEY="tvly-..." # Optional
```

Usage

Run the entry point script to start an interactive session:

```
python nexus_agent/main.py
```

Example Workflow:

- Input:** "Research the latest advancements in Solid State Batteries."
- Planning:** Nexus breaks this down into "Cathode materials," "Manufacturing costs," and "EV adoption."
- Execution:** You will see logs of parallel workers fetching data simultaneously.
- Review:** Nexus presents a draft.
- Critique:** You type: *"Make it more technical and focus on Toyota's roadmap."*
- Refinement:** Nexus rewrites the specific sections and updates the draft.

📁 Project Structure

A clean, modular structure separating concerns for scalability.

```
nexus_agent/  
|__ main.py          # CLI Entry point & Human-in-the-loop logic  
|__ requirements.txt # Project dependencies  
└__ src/  
    |__ config.py     # LLM & API Configuration  
    |__ graph.py      # Core LangGraph definition (Nodes & Edges)  
    |__ memory_store.py # Long-term persistence logic (JSON/DB)
```

```
├── state.py          # Pydantic models for Graph State
├── tools.py          # Search tools (Tavily/Mock)
└── nodes/
    ├── planner.py    # Individual Agentic Components
    ├── researcher.py # Task decomposition agent
    └── writer.py     # Parallel worker agent
                      # Synthesis & Drafting agent
```

🧠 Memory & Adaptation

Nexus creates a file named `user_long_term_memory.json` in your root directory.

- If you say: *"I prefer bullet points."*
- Nexus saves: `{"preferences": ["Use concise, bulleted format"]}`
- **Next Session:** Nexus automatically applies this formatting without being asked again.

🔮 Future Roadmap

- [] **Vector Database:** Migrate JSON memory to Pinecone/Weaviate for semantic recall.
- [] **Multi-Modal:** Add support for analyzing images and charts during research.
- [] **Slack/Discord Bot:** Wrap `main.py` in a FastAPI server for external webhooks.

Built with ❤️ using LangGraph & Python.