From Prompt to Production: A Technical Guide to LangChain, LangGraph, and LangSmith

---

1. Introduction

The development of intelligent applications using large language models (LLMs) like OpenAI's GPT, Anthropic's Claude, or open-source models from HuggingFace has grown rapidly. However, turning these powerful models into real, reliable software systems requires infrastructure to manage workflows, tools, memory, and evaluation.

LangChain, **LangGraph**, and **LangSmith** are three foundational tools developed to support this ecosystem. Together, they allow developers to:

- Create structured logic using LLMs (LangChain)
- Build stateful and collaborative workflows (LangGraph)
- Observe, evaluate, and debug these workflows (LangSmith)

This documentation explains each component in depth.

---

2. LangChain

2.1 What is LangChain?

LangChain is a **modular framework** designed for building applications that use large language models in a **composable**, **tool-integrated**, and **memory-aware** way. It abstracts and simplifies the complexity of prompt engineering, model orchestration, and chaining logic across steps.

Instead of hardcoding prompts and handling raw model responses, LangChain introduces structured components that can be combined to create intelligent workflows.

2.2 Architecture and Key Components

2.2.1 Language Models

LangChain supports both proprietary (e.g., OpenAI, Cohere, Anthropic) and open-source models (e.g., HuggingFace Transformers, Ollama). Developers can switch between models without rewriting application logic.

2.2.2 Prompt Templates

LangChain enables the use of reusable prompt templates, where dynamic variables can be injected at runtime. This helps standardize and scale prompt usage.

2.2.3 Chains

Chains are sequences of calls that may include:

- Prompting an LLM
- Using tools (like APIs)
- Performing logic
- Interacting with memory

  Chains can be linear or branching, synchronous or async.

Example:

```
user_input → summarize → search → combine → output
```

2.2.4 Agents

Agents are dynamic decision-makers. Instead of executing fixed logic, agents decide what to do based on the context. For instance, a chatbot might:

- Recognize the user is asking for weather → call weather API
- Detect a calculation → use calculator tool
- Retrieve documents → search a vector store

Agents use **tools**, **reasoning**, and **memory** to choose and execute actions.

2.2.5 Tools

Tools are external utilities connected to LangChain agents. Common tools include:

- Web search
- SQL databases
- Document retrievers
- Custom APIs

2.2.6 Memory

Memory is a critical feature for any LLM application that needs context. LangChain supports:

- Short-term memory (within one session)
- Long-term memory (across sessions)
- Vector stores for document retrieval

2.3 Use Cases of LangChain

- Document Question Answering Systems (RAG)
- AI Assistants (technical support, research, etc.)
- Autonomous Agents (task execution pipelines)
- Workflow Automation (LLM-powered orchestration)
- Conversational Agents with Memory and Tool Usage

---

3. LangGraph

3.1 What is LangGraph?

LangGraph is a **graph-based orchestration library** built on top of LangChain. While LangChain enables chaining logic in a stepwise manner, LangGraph

introduces a **state machine approach** using **directed graphs**.

This allows developers to build **multi-agent**, **asynchronous**, and **recursive** LLM systems with clarity and precision.

3.2 Architectural Principles

3.2.1 Nodes

Each node in a LangGraph represents an individual function, typically a LangChain Runnable, such as:

- LLM call
- Tool invocation
- State transformation

Nodes execute logic and update the application state.

3.2.2 Edges and Transitions

Edges define **conditional or deterministic transitions** between nodes. For example:

- If Agent A approves → go to next step
- If Agent B rejects → loop back

This allows **non-linear** workflows, including:

- Loops
- Parallel processing
- Conditional branching

3.2.3 State Object

LangGraph maintains a persistent, immutable **state dictionary** that flows between nodes. This can include:

- Conversation history
- Tool results

- Decision flags
- Metadata for context switching

3.2.4 Multi-Agent Orchestration

LangGraph is ideal for coordinating multiple agents:

- Each agent operates in a defined node
- The graph controls their interaction logic
- Agents can collaborate, argue, revise, and agree before moving forward

---

3.3 LangGraph vs LangChain

Feature	LangChain	LangGraph
Structure	Linear chains or trees	Directed graph (DAG / state machine)
Logic Control	Step-by-step or agent-based	Loops, conditions, re-entry, backtracking
Complexity	Medium	High (multi-agent, dynamic workflows)
State Sharing	Limited or explicit	Central state passed through the graph

---

3.4 Example Use Case

A **document reviewer system**:

- Node 1: Agent A reads a document and summarizes it
- Node 2: Agent B critiques the summary
- If feedback is negative → go back to Agent A
- Else → final approval and export

Such recursive workflows are easily managed in LangGraph.

---

4. LangSmith

4.1 What is LangSmith?

LangSmith is a **developer platform** for debugging, evaluating, and monitoring LangChain and LangGraph applications. It solves a critical problem in LLM-based development: **visibility into what the model is doing** at each step.

It acts like a combination of:

- Developer console
- Experimentation lab
- Logging and monitoring system

4.2 Key Capabilities

4.2.1 Tracing and Logging

Every step in a chain or graph run is recorded, including:

- Prompt inputs and outputs
- Tool calls and results
- Decision logic from agents
- Timestamps, errors, and retries

This enables end-to-end visibility.

4.2.2 Debugging

You can inspect:

- How prompts were structured
- Where outputs deviated from expected behavior
- Which version of a chain or agent was used

4.2.3 Evaluation

LangSmith allows structured evaluation using:

- Custom metrics
- Ground truth comparisons

- Batch evaluations over datasets

4.2.4 Collaboration and Sharing

LangSmith provides sharable links to logs and runs. Developers, reviewers, or product managers can examine how a specific prompt or flow behaved without needing code access.

4.3 Benefits for Production

- Confidence in prompt and chain behavior
- Faster debugging of complex multi-agent flows
- Improved observability for CI/CD integration
- Better reliability and performance testing

---

5. Ecosystem Summary

Tool	Role	Ideal For
LangChain	Framework for chaining LLM-based logic	Building AI assistants, retrieval systems, simple agents
LangGraph	Graph-based orchestration for LangChain workflows	Designing complex, stateful, multi-agent workflows
LangSmith	Observability and evaluation platform	Testing, debugging, logging, and performance evaluation

---

6. Conclusion

As AI applications grow more complex, so must the infrastructure behind them. LangChain provides the foundational architecture for LLM logic and tooling. LangGraph builds on top to allow sophisticated control flows and collaboration between agents. LangSmith closes the loop with observability, debugging, and performance analytics.

Together, these tools represent a **modern software stack for LLM applications**—ideal for startups, research labs, and enterprises building intelligent, autonomous systems.

Wiring Intelligence: A Developer’s Guide to LangChain in Action

---

1. Environment Setup using `uv` (UltraVenv)

You're using `uv`, which is a fast Python package manager and environment manager — an excellent modern choice. Here's a formal outline of what you've done and how to document it clearly.

1.1 Initialize the Environment

```
uv init
```

This command initializes the current directory for `uv`-based package management.

1.2 Create a Virtual Environment

```
uv venv
```

This sets up a **lightweight, isolated virtual environment** for Python dependencies.

1.3 Activate the Environment

On **Unix/macOS**:

```
source .venv/bin/activate
```

On **Windows**:

```
.venv\Scripts\activate
```

> Once activated, your terminal should show the virtual environment name at the beginning of the prompt.

---

2. Project Structure Setup

You created a project folder with a modular structure for a LangChain app — this is excellent practice.

2.1 Folder Name

```
mkdir 1.basic-chatbot
cd 1.basic-chatbot
```

This folder is intended to contain your **first hands-on LangChain chatbot project**, using a modular architecture.

2.2 Suggested Folder Structure Inside the Project

To maintain clean separation of logic, you can organize your project like this:

```
1.basic-chatbot-modular/
│
├── main.py              # Entry point of the chatbot
├── prompts/             # Custom prompt templates
│   └── basic_prompt.txt
├── chains/              # LangChain chain definitions
│   └── basic_chain.py
├── memory/              # Memory configuration (optional)
│   └── buffer_memory.py
├── tools/               # Custom tools (e.g., calculator, search)
```

```
│   └── calculator.py
├── .venv/                # Virtual environment folder (auto-created by uv)
├── requirements.txt      # Optional: Locked dependencies if needed
└── README.md             # Project overview and instructions
```

3. Simple Stateless Bot

3.1 Environment Configuration

You start by loading environment variables using `dotenv`:

```
import os
from dotenv import load_dotenv

load_dotenv()
```

Why this matters:

This ensures your secrets (e.g. `GROQ_API_KEY`) are securely loaded from a `.env` file instead of being hard-coded.

3.2 LLM Initialization (Groq LLaMA3)

Two models are initialized, but **only the second one is used effectively**:

```
from langchain_groq import ChatGroq
from langchain.chat_models import init_chat_model

llm = init_chat_model("groq:llama3-8b-8192")
```

`init_chat_model()` automatically loads the appropriate LLM interface for Groq.

3.3 Define Graph State

```
from typing import Annotated
from typing_extensions import TypedDict
from langgraph.graph.message import add_messages

class State(TypedDict):
    messages: Annotated[list, add_messages]
```

What this does:

Defines the state that LangGraph nodes will operate on. In this case, it tracks the list of `messages`.

- `add_messages`: a LangGraph utility that merges messages across turns

3.4 **Create Node Function**

This function performs the LLM invocation:

```
def chatbot(state: State):
    return {"messages": [llm.invoke(state["messages"])]}
```

- It uses the entire message list (likely a bug: should use last message only!)
- Returns a new assistant message wrapped in a `{"messages": [...]}` dict

3.5 **Graph Construction**

```
from langgraph.graph import StateGraph, START, END

graph_builder = StateGraph(State)
graph_builder.add_node("llmchatbot", chatbot)
graph_builder.add_edge(START, "llmchatbot")
graph_builder.add_edge("llmchatbot", END)

graph = graph_builder.compile()
```

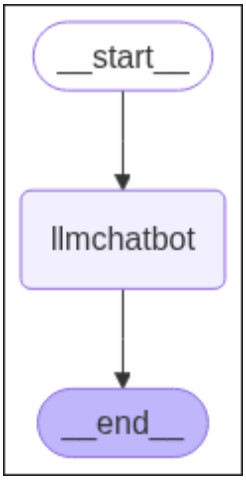
Explanation:

- The graph has only 1 active node: `llmchatbot`
- It starts at `START`, routes through `llmchatbot`, then ends at `END`
- Stateless: No persistent memory from one input to the next

3.6 **Graph Visualization**

```
from IPython.display import Image, display
try:
    display(Image(graph.get_graph().draw_mermaid_png()))
except Exception:
    pass
```

Output



3.7 **Invoke the Graph**

```
response = graph.invoke({"messages": "hi"})
response["messages"][-1].content
```

This runs a full pass through the graph with a single user message.

Output:

> Hi! It's nice to meet you. Is there something I can help you with, or would you like to chat?”

3.8 **Streaming Response**

```
for event in graph.stream({"messages": "HI! HOW ARE YOU"}):
    for value in event.values():
        print(value["messages"][-1].content)
```

This uses `graph.stream(...)` to stream token-wise responses.

Output:

> Hi! I'm just a language model, so I don't have feelings like humans do, but I'm functioning properly and ready to help you with any questions or tasks you may have! How about you?

---

4. Chatbot with Tool Integration

This section demonstrates how to build a chatbot using **LangChain** and **LangGraph** that can dynamically call external tools like:

- 🔍 **Tavily Search API** (for AI news and web results)
- ➗ **A custom math function** (for multiplication)

The chatbot automatically detects when a tool is needed and routes execution accordingly.

4.1 Libraries Used and Installation

Required Libraries

Library	Purpose
`langchain`	Language model framework for LLMs, tools, memory, chains, etc.

Library	Purpose
`langgraph`	Enables graph-based flow control for multi-step AI workflows
`langchain_tavily`	Integration with Tavily web search API
`python-dotenv`	Securely load environment variables like API keys from a `.env` file

Install Commands

```
uv add langchain langgraph langchain-tavily python-dotenv
```

4.2 Tavily API Setup

To use the **TavilySearch** tool, you need an API key from Tavily.

Step 1: `.env` File

Create a file called `.env` in your project folder:

```
TAVILY_API_KEY=your_real_api_key_here
```

> ⚠️ Do NOT commit this file to GitHub. Add .env to your .gitignore.

Step 2: Load Environment Variables

```
import os
from dotenv import load_dotenv

load_dotenv()
tavily_api_key = os.getenv("TAVILY_API_KEY")
```

Step 3: Initialize Tavily Tool

```
from langchain_tavily import TavilySearch
```

```
tool = TavilySearch(api_key=tavily_api_key, max_results=2)
```

4.3 Tool Definitions and LLM Binding

We define tools and bind them to the LLM so that it can call them when needed.

Custom Multiply Function

```
def multiply(a: int, b: int) → int:
    """Multiply two integers"""
    return a * b
```

Bind Tools to LLM

```
tools = [tool, multiply]
llm_with_tool = llm.bind_tools(tools)
```

4.4 Graph Logic using LangGraph

Now we define how the system should behave using a graph of nodes and conditions.

Step 1: Define Node for LLM

```
def tool_calling_llm(state: State):
    return {"messages": [llm_with_tool.invoke(state["messages"])]}
```

Step 2: Create Graph

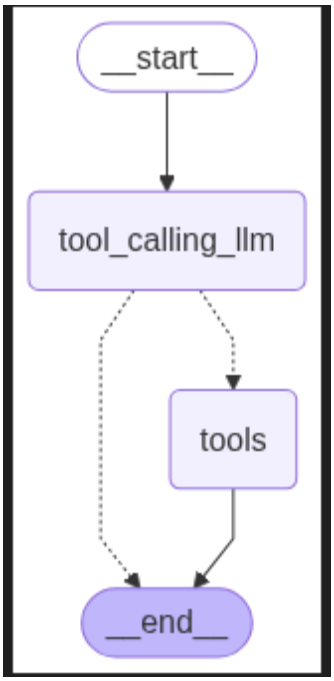
```
from langgraph.graph import StateGraph, START, END
from langgraph.prebuilt import ToolNode, tools_condition

builder = StateGraph(State)
builder.add_node("tool_calling_llm", tool_calling_llm)
```

```
builder.add_node("tools", ToolNode(tools))

builder.add_edge(START, "tool_calling_llm")
builder.add_conditional_edges("tool_calling_llm", tools_condition)
builder.add_edge("tools", END)
```

Step 3: Compile and Visualize

```
graph = builder.compile()

from IPython.display import Image
display(Image(graph.get_graph().draw_mermaid_png()))
```

Output:



4.5 Graph Execution with Examples

Let's test the chatbot with real user queries.

Example 1: AI News via Tavily

```
response = graph.invoke({"messages": "HI! what is recent ai news"})
```

Output:


```
'{"query": "recent ai news", "follow_up_questions": null, "answer": null, "imag
es": [], "results": [{"url": "https://opentools.ai/news/elon-musks-grok-4-a-ne
w-player-in-the-ai-intelligence-arena", "title": "Elon Musk\'s Grok 4: A New Pl
ayer in the AI Intelligence Arena - OpenTools", "score": 0.7548612, "published
_date": "Sat, 12 Jul 2025 01:02:59 GMT", "content": "Published Time: 2025-07
-12T01:02:45.505Z\\n\\nNews Not Found | AI News\\n\\n===============
\\n\\nOpenToolsImage 1: logo\\n\\nOpen main menu\\n\\nNewsletterNewsSub
mit A ToolAdvertise\\n\\nCategories\\n\\nGet started\\n\\nLearn to use AI like a
Pro. Learn More\\n\\nOpenToolsImage 2: logo\\n\\nOpen main menu\\n\\nNew
sletterNewsSubmit A ToolAdvertise\\n\\nCategories\\n\\nGet started\\n\\nLear
n to use AI like a Pro. Learn More\\n\\n1.  home\\n2.  news\\n3.  elon-musks-
grok-4-a-new-player-in-the-ai-intelligence-arena\\n\\nAI Revolution: Grok 4 T
akes the Stage [...] [[Feb 19, 2025] Elon Musk\'s xAI Unleashes Grok 3: A Qua
ntum Leap in AI Innovation](\\n  [[Jul 11, 2025] Musk Launches Grok 4: xAI\'s
New AI Model Outshines OpenAI and Google](\\n  [[Feb 18, 2025] Elon Musk
\'s xAI Unveils Grok 3: Aiming to Lead AI Innovation](\\n  [[Feb 18, 2025] Elon
Musk\'s xAI Unleashes Grok-3: The Newest Rival in AI Benchmarking]( [...] [[F
eb 18, 2025] Elon Musk\'s xAI Unveils Grok 3: A Game-Changer in AI Innovatio
n!](\\n  [[Feb 16, 2025] Elon Musk\'s Grok 3: The World\'s Smartest AI is Laun
ching Soon](\\n  [[Jul 11, 2025] Grok-4: xAI\'s AI Trailblazer Outsmarts OpenAI
and Google!](\\n  [[Apr 30, 2025] Elon Musk Unveils Grok 3.5: An AI Model to
Outthink All](\\n  [[Feb 19, 2025] Elon Musk Unveils Grok-3: An AI Powerhous
e Taking on ChatGPT!](\\n  [[Jul 10, 2025] Elon Musk\'s xAI Launches Grok 4
Amid Controversy!](", "raw_content": null}, {"url": "https://www.ainvest.com/n
ews/amazon-ai-ambitions-commerce-giant-surpass-nvidia-microsoft-valuati
on-2030-2507/", "title": "Amazon\'s AI Ambitions: Why the E-commerce Giant
Could Surpass Nvidia and Microsoft in Valuation by 2030 - AInvest", "score":
0.69795954, "published_date": "Sun, 13 Jul 2025 01:00:44 GMT", "content":
"Amazon\'s AI business has achieved a multi-billion-dollar annual run rate with
triple-digit growth rates, positioning Amazon advantageously in the generative
AI transformation sweeping through enterprise technology . Recent AWS deve
```

lopments demonstrate the company\'s commitment to AI leadership, including the introduction of Amazon Nova Canvas with AI-powered image generation c apabilities and the availability of Anthropic Claude 4 models through Amazon Bedrock . [...] Amazon (AMZN) continues to leverage artificial intelligence (AI) across its business to drive growth, with a focus on e-commerce and cloud inf rastructure. The company\'s integration of AI-powered robotics in warehouse s could reduce costs by 25%, expanding profitability in e-commerce operatio ns. Meanwhile, Amazon\'s cloud business has seen significant growth with the integration of Anthropic\'s services into Amazon Web Services (AWS). [...] Pro ject Rainier represents Amazon\'s ambitious infrastructure investment in AI co mpute capabilities, designed to train next-generation AI models through powe rful Trainium2 chip clusters . This initiative demonstrates Amazon\'s willingnes s to invest heavily in foundational AI infrastructure, potentially creating signific ant barriers to entry for competitors while establishing AWS as the preferred p latform for AI workload deployment.", "raw_content": null}, {"url": "https://ww w.csoonline.com/article/4021749/new-grok-4-ai-breached-within-48-hours-using-whispered-jailbreaks.html", "title": "New Grok-4 AI breached within 48 hours using ‘whispered’ jailbreaks - csoonline.com", "score": 0.69672287, "pu blished_date": "Mon, 14 Jul 2025 11:55:29 GMT", "content": "News Analysis # ## Putting AI-assisted ‘vibe hacking’ to the test By Lucian Constantin Jul 14, 2 025 7 mins Cyberattacks Penetration Testing VulnerabilitiesNews ### McDon ald’s AI hiring tool’s password ‘123456’ exposed data of 64M applicants By Sh weta Sharma Jul 11, 2025 5 mins Data Breach Passwords SecurityNews ### A MD discloses new CPU flaws that can enable data leaks via timing attacks By Gyana Swain Jul 10, 2025 5 mins Security Vulnerabilities\\n\\nOther Sections \\n---------- [...] news ### Critical RCE flaw in Anthropic’s MCP inspector exposes developer machines to remote attacks Jul 2, 2025 4 mins\\n news # ## LLMs are guessing login URLs, and it’s a cybersecurity time bomb Jul 1, 20 25 4 mins\\n news ### Patch now: Citrix Bleed 2 vulnerability actively exploit ed in the wild Jun 30, 2025 3 mins\\n news ### Some Brother printers have a remote code execution vulnerability, and they can’t fix it Jun 27, 2025 5 min s [...] news ### How a 12-year-old bug in Sudo is still haunting Linux users Jul 8, 2025 4 mins\\n news ### NightEagle hackers exploit Microsoft Exchange f law to spy on China’s strategic sectors Jul 7, 2025 4 mins\\n news ### Verifi ed, but vulnerable: Malicious extensions exploit IDE trust badges Jul 4, 2025 4 mins\\n news ### Hardcoded root credentials in Cisco Unified CM trigger

max-severity alert Jul 3, 2025 3 mins", "raw_content": null}, {"url": "https://www.theregister.com/2025/07/11/ai_code_tools_slow_down/", "title": "AI coding tools make developers slower but they think they\'re faster, study finds - theregister.com", "score": 0.6493346, "published_date": "Fri, 11 Jul 2025 22:41:00 GMT", "content": "#### At last, a use case for AI agents with sky-high ROI: Stealing crypto Boffins outsmart smart contracts with evil automation AI + ML 10 Jul 2025 | 14#### Cloudflare creates AI crawler tollbooth to pay publishers ai-pocalypse The bargain between content makers and crawlers has broken down AI + ML 1 Jul 2025 | 20#### Don\'t pay for AI support failures, says Gradient Labs CEO interview Paying for successful problem resolution is a better business model, argues Dimitri Masin AI + ML 30 Jun 2025 | [...] Other researchers have also found that AI does not always live up to the hype. A recent study from AI coding biz Qodo found some of the benefits of AI software assistance were undercut by the need to do additional work to check AI code suggestions. An economic survey found that generative AI has had no impact on jobs or wages, based on data from Denmark. An Intel study found that AI PCs make users less productive. And call center workers at a Chinese electrical utility say that while AI [...] likely to answer 27 AI + ML 30 Jun 2025 | 31#### Call center staffers explain to researchers how their AI assistants aren\'t very helpful ai-pocalypse Lots of manual corrections and data entry still required AI + ML 2 Jul 2025 | 42", "raw_content": null}, {"url": "https://startupnews.fyi/2025/07/14/disrupt-2025-audience-choice-winners-revealed/", "title": "Disrupt 2025 Audience Choice winners revealed - StartupNews.fyi", "score": 0.61168414, "published_date": "Mon, 14 Jul 2025 15:02:47 GMT", "content": "### Passed Audit and 13,800% ROI Forecast? Analysts Say Ruvi AI (RUVI) Could Outpace Ripple (XRP) This Cycle Could Outpace Ripple (XRP) This Cycle\\")\\n\\nBlockchainJuly 14, 2025\\n\\n\\n\\n### Google, Anthropic, OpenAI and xAI join US defence to tackle national security with AI\\n\\nTechJuly 14, 2025\\n\\n\\n\\n### Redefining global trade infrastructure: TradeOS joins Cointelegraph Accelerator\\n\\nBlockchainJuly 14, 2025\\n\\n\\n\\n### Malaysia will require trade permits for U.S. AI chips\\n\\nAIJuly 14, 2025", "raw_content": null}], "response_time": 1.3}

Output Formatting using `.pretty_print()`

Each response from the `graph.invoke(...)` contains a list of message objects in the `response["messages"]` field. To display them nicely, you can loop through and use:

```
for m in response["messages"]:
    m.pretty_print()
```

Tavily Web Search Output

```
response = graph.invoke({"messages": "Hi! What is the recent AI news?"})

for m in response["messages"]:
    m.pretty_print()
```

Pretty Output:

```
================================ Human Message ============
======================

HI! what is recent ai news
================================== Ai Message ============
======================
Tool Calls:
  tavily_search (zyk8dajg6)
 Call ID: zyk8dajg6
  Args:
    query: recent ai news
    search_depth: advanced
    time_range: week
    topic: news
================================= Tool Message ============
======================
Name: tavily_search

{"query": "recent ai news", "follow_up_questions": null, "answer": null, "image
s": [], "results": [{"url": "https://opentools.ai/news/elon-musks-grok-4-a-new-
```

player-in-the-ai-intelligence-arena", "title": "Elon Musk's Grok 4: A New Player in the AI Intelligence Arena - OpenTools", "score": 0.7548612, "published_date": "Sat, 12 Jul 2025 01:02:59 GMT", "content": "Published Time: 2025-07-12T01:02:45.505Z\n\nNews Not Found | AI News\n\n==========\n\nOpenToolsImage 1: logo\n\nOpen main menu\n\nNewsletterNewsSubmit A ToolAdvertise\n\nCategories\n\nGet started\n\nLearn to use AI like a Pro. Learn More\n\nOpenToolsImage 2: logo\n\nOpen main menu\n\nNewsletterNewsSubmit A ToolAdvertise\n\nCategories\n\nGet started\n\nLearn to use AI like a Pro. Learn More\n\n1. home\n2. news\n3. elon-musks-grok-4-a-new-player-in-the-ai-intelligence-arena\n\nAI Revolution: Grok 4 Takes the Stage [...] [[Feb 19, 2025] Elon Musk's xAI Unleashes Grok 3: A Quantum Leap in AI Innovation](\n [[Jul 11, 2025] Musk Launches Grok 4: xAI's New AI Model Outshines OpenAI and Google](\n [[Feb 18, 2025] Elon Musk's xAI Unveils Grok 3: Aiming to Lead AI Innovation](\n [[Feb 18, 2025] Elon Musk's xAI Unleashes Grok-3: The Newest Rival in AI Benchmarking]([...] [[Feb 18, 2025] Elon Musk's xAI Unveils Grok 3: A Game-Changer in AI Innovation!](\n [[Feb 16, 2025] Elon Musk's Grok 3: The World's Smartest AI is Launching Soon](\n [[Jul 11, 2025] Grok-4: xAI's AI Trailblazer Outsmarts OpenAI and Google!](\n [[Apr 30, 2025] Elon Musk Unveils Grok 3.5: An AI Model to Outthink All](\n [[Feb 19, 2025] Elon Musk Unveils Grok-3: An AI Powerhouse Taking on ChatGPT!](\n [[Jul 10, 2025] Elon Musk's xAI Launches Grok 4 Amid Controversy!](", "raw_content": null}, {"url": "https://www.ainvest.com/news/amazon-ai-ambitions-commerce-giant-surpass-nvidia-microsoft-valuation-2030-2507/", "title": "Amazon's AI Ambitions: Why the E-commerce Giant Could Surpass Nvidia and Microsoft in Valuation by 2030 - AInvest", "score": 0.69795954, "published_date": "Sun, 13 Jul 2025 01:00:44 GMT", "content": "Amazon's AI business has achieved a multi-billion-dollar annual run rate with triple-digit growth rates, positioning Amazon advantageously in the generative AI transformation sweeping through enterprise technology . Recent AWS developments demonstrate the company's commitment to AI leadership, including the introduction of Amazon Nova Canvas with AI-powered image generation capabilities and the availability of Anthropic Claude 4 models through Amazon Bedrock . [...] Amazon (AMZN) continues to leverage artificial intelligence (AI) across its business to drive growth, with a focus on e-commerce and cloud infrastructure. The company's integration of AI-powered robotics in warehouses could reduce costs by 25%, expanding profitability in e-commerce operations. Meanwhile, Amazon's cloud business

has seen significant growth with the integration of Anthropic's services into A
mazon Web Services (AWS). [...] Project Rainier represents Amazon's ambitio
us infrastructure investment in AI compute capabilities, designed to train next-
generation AI models through powerful Trainium2 chip clusters . This initiative
demonstrates Amazon's willingness to invest heavily in foundational AI infrast
ructure, potentially creating significant barriers to entry for competitors while
establishing AWS as the preferred platform for AI workload deployment.", "raw
_content": null}, {"url": "https://www.csoonline.com/article/4021749/new-grok
-4-ai-breached-within-48-hours-using-whispered-jailbreaks.html", "title": "N
ew Grok-4 AI breached within 48 hours using 'whispered' jailbreaks - csoonlin
e.com", "score": 0.69672287, "published_date": "Mon, 14 Jul 2025 11:55:29 G
MT", "content": "News Analysis ### Putting AI-assisted 'vibe hacking' to the t
est By Lucian Constantin Jul 14, 2025 7 mins Cyberattacks Penetration Testin
g VulnerabilitiesNews ### McDonald's AI hiring tool's password '123456' expo
sed data of 64M applicants By Shweta Sharma Jul 11, 2025 5 mins Data Breac
h Passwords SecurityNews ### AMD discloses new CPU flaws that can enabl
e data leaks via timing attacks By Gyana Swain Jul 10, 2025 5 mins Security V
ulnerabilities\n\nOther Sections\n---------- [...] news ### Critical RCE fla
w in Anthropic's MCP inspector exposes developer machines to remote attack
s Jul 2, 2025 4 mins\n news ### LLMs are guessing login URLs, and it's a cy
bersecurity time bomb Jul 1, 2025 4 mins\n news ### Patch now: Citrix Blee
d 2 vulnerability actively exploited in the wild Jun 30, 2025 3 mins\n news ##
Some Brother printers have a remote code execution vulnerability, and they
can't fix it Jun 27, 2025 5 mins [...] news ### How a 12-year-old bug in Sudo i
s still haunting Linux users Jul 8, 2025 4 mins\n news ### NightEagle hacker
s exploit Microsoft Exchange flaw to spy on China's strategic sectors Jul 7, 20
25 4 mins\n news ### Verified, but vulnerable: Malicious extensions exploit I
DE trust badges Jul 4, 2025 4 mins\n news ### Hardcoded root credentials i
n Cisco Unified CM trigger max-severity alert Jul 3, 2025 3 mins", "raw_conte
nt": null}, {"url": "https://www.theregister.com/2025/07/11/ai_code_tools_slow_
down/", "title": "AI coding tools make developers slower but they think they're
faster, study finds - theregister.com", "score": 0.6493346, "published_date":
"Fri, 11 Jul 2025 22:41:00 GMT", "content": "#### At last, a use case for AI ag
ents with sky-high ROI: Stealing crypto Boffins outsmart smart contracts with
evil automation AI + ML 10 Jul 2025 | 14#### Cloudflare creates AI crawler tol
lbooth to pay publishers ai-pocalypse The bargain between content makers a

nd crawlers has broken down AI + ML 1 Jul 2025 | 20#### Don't pay for AI su
pport failures, says Gradient Labs CEO interview Paying for successful proble
m resolution is a better business model, argues Dimitri Masin AI + ML 30 Jun
2025 | [...] Other researchers have also found that AI does not always live up t
o the hype. A recent study from AI coding biz Qodo found some of the benefit
s of AI software assistance were undercut by the need to do additional work t
o check AI code suggestions. An economic survey found that generative AI ha
s had no impact on jobs or wages, based on data from Denmark. An Intel stud
y found that AI PCs make users less productive. And call center workers at a
Chinese electrical utility say that while AI [...] likely to answer 27 AI + ML 30 J
un 2025 | 31#### Call center staffers explain to researchers how their AI assis
tants aren't very helpful ai-pocalypse Lots of manual corrections and data ent
ry still required AI + ML 2 Jul 2025 | 42", "raw_content": null}, {"url": "https://s
tartupnews.fyi/2025/07/14/disrupt-2025-audience-choice-winners-reveale
d/", "title": "Disrupt 2025 Audience Choice winners revealed - StartupNews.fy
i", "score": 0.61168414, "published_date": "Mon, 14 Jul 2025 15:02:47 GMT",
"content": "### Passed Audit and 13,800% ROI Forecast? Analysts Say Ruvi A
I (RUVI) Could Outpace Ripple (XRP) This Cycle Could Outpace Ripple (XRP) T
his Cycle\")\n\nBlockchainJuly 14, 2025\n\n\n\n### Google, Anthropic, OpenA
I and xAI join US defence to tackle national security with AI\n\nTechJuly 14, 20
25\n\n\n\n### Redefining global trade infrastructure: TradeOS joins Cointeleg
raph Accelerator\n\nBlockchainJuly 14, 2025\n\n\n\n### Malaysia will require
trade permits for U.S. AI chips\n\nAIJuly 14, 2025", "raw_content": null}], "res
ponse_time": 1.3}

Example 2: Math Function (2 × 15)

```
response = graph.invoke({"messages": "HI! what is 2 multiply by 15"})
```

Expected Output:

```
'30'
```

Example 3: Multi-step Math (Tool Chaining)

```
response = graph.invoke({"messages": "HI! what is 2 multiply by 15 then multi
ply by 10"})
```

Output:

```
'150'
```

> ℹ️ Note: LangGraph executes only one tool call per run. For chained tool use (e.g. two multiplications in a row), custom looping logic is needed.

Output Formatting using `.pretty_print()`

Each response from the `graph.invoke(...)` contains a list of message objects in the `response["messages"]` field. To display them nicely, you can loop through and use:

```
for m in response["messages"]:
    m.pretty_print()
```

Multiplication (2 × 15)

```
response = graph.invoke({"messages": "Hi! What is 2 multiply by 15?"})

for m in response["messages"]:
    m.pretty_print()
```

Pretty Output:

```
================================ Human Message ============ ======================

HI! what is 2 multiply by 15
================================== Ai Message ============
```

```
=======================
Tool Calls:
  multiply (2pv0mhscq)
 Call ID: 2pv0mhscq
  Args:
    a: 2
    b: 15
================================== Tool Message ============
=====================
Name: multiply

30
```

Example 4: Chained Math Request

```
response = graph.invoke({"messages": "What is 2 multiply by 15 then multiply by 10?"})

for m in response["messages"]:
    m.pretty_print()
```

Pretty Output:

```
================================= Human Message ===========
=======================

HI! what is 2 multiply by 15 then multiply by 10
================================== Ai Message ============
=======================
Tool Calls:
  multiply (wfjbhdkrx)
 Call ID: wfjbhdkrx
  Args:
    a: 2
    b: 15
```

```
  multiply (0f2nben9c)
 Call ID: 0f2nben9c
  Args:
    a: 15
    b: 10
================================= Tool Message ============
=====================
Name: multiply

30
================================= Tool Message ============
=====================
Name: multiply

150
```

> ❗ This only processes the first multiplication. To support full chaining, you'd need logic to track intermediate results.

Example 5: Mixed Intent (Search + Math)

```
response = graph.invoke({"messages": "Provide me the recent AI news and m
ultiply 15 by 10"})

for m in response["messages"]:
    m.pretty_print()
```

Pretty Output:

```
================================= Human Message ===========
=====================

HI! provide me the recent ai news multiply 15 by 10
================================== Ai Message ============
```

```
=======================
Tool Calls:
  tavily_search (s9y1sm82e)
 Call ID: s9y1sm82e
  Args:
    query: AI news
    search_depth: advanced
    time_range: day
  multiply (0s3ae42wr)
 Call ID: 0s3ae42wr
  Args:
    a: 15
    b: 10
================================= Tool Message =============
=======================
Name: tavily_search

{"query": "AI news", "follow_up_questions": null, "answer": null, "images": [],
"results": [{"url": "https://ts2.tech/en/ai-supremacy-space-odyssey-tech-sha
keups-the-biggest-tech-news-of-july-2025/", "title": "AI Supremacy, Space O
dyssey & Tech Shakeups: The Biggest Tech ...", "content": "stock market reute
rs.com!) All told, AI news in July spanned technical leaps, blockbuster valuatio
ns, and high-stakes talent wars, signaling that the AI revolution shows no sign
s of slowing. [...] Notable AI Releases: Research and product news abounded.
OpenAI signaled bold moves beyond chatbots – reportedly developing an AI-p
owered web browser poised to challenge Google Chrome by integrating smart
agents for tasks like travel booking radicaldatascience.wordpress.com. Amaz
on's AWS, in turn, announced an upcoming AI agent marketplace (with Anthro
pic as a partner) to let startups offer AI "agents" to AWS customers radicaldat
ascience.wordpress.com. On the research front, Anthropic [...] Image 7: AI To
ol Bonanza: 7 New AI Releases & Updates on July 14, 2025 You Need to Know
AI Tool Bonanza: 7 New AI Releases & Updates on July 14, 2025 You Need to
Know\nFrom a mind-reading web browser to a school-savvy chatbot, today's
AI tool announcements span every corner of life. In this daily roundup for July
14, 2025, we break down the most important new AI tools and updates that jus
t dropped – across productivity, creativity, education, communication, and...",
```

"score": 0.7640656, "raw_content": null}, {"url": "https://www.reuters.com/technology/artificial-intelligence/", "title": "Artificial Intelligence - AI News - Reuters", "content": "Published Time: Mon, 14 Jul 2025 14:36:56 GMT\n\nAI News | Latest Headlines and Developments | Reuters\n\n==========\n\nSkip to main content\n\nReport This Ad\n\nExclusive news, data and analytics for financial market professionals Learn more about Refinitiv\n\n\n\nTechnology July 11, 2025 · 10:02 PM UTC\n\nArtist transforms Messi's all-time favorite goal into AI art [...] , opens new tab\n Download the App (Android), opens new tab\n Newsletters\n Subscribe\n\nInformation you can trust\n----------\n\nReuters, the news and media division of Thomson Reuters, is the world's largest multimedia news provider, reaching billions of people worldwide every day. Reuters provides business, financial, national and international news to professionals via desktop terminals, the world's media organizations, industry events and directly to consumers. [...] Soccer star Lionel Messi has chosen his favorite goal of his 20-year career — his 2009 Champions League Final header—to be transformed into AI-driven digital art. \"We are like making invisible visible," said Refik Anadol, a contemporary digital artist who collaborated closely wth Messi.\n\n### Up next\n\n Dubai to debut restaurant operated by an AI chef , beating market forecasts, as demand for the company's products leaps on surging interest in artificial intelligence applications.", "score": 0.73891455, "raw_content": null}, {"url": "https://www.linkedin.com/pulse/ai-news-funding-updates-from-last-24-hours14th-july-2025-anshuman-jha-jfr0c", "title": "AI news and funding updates from the last 24 hours(14th July 2025)", "content": "Image 25Image 26 17 \n AI news and funding updates from the last 24 hours(11th July 2025)Image 27 Jul 11, 2025 \n### AI news and funding updates from the last 24 hours(11th July 2025)\n\nAmazon Web Services (AWS) Amazon Web Services (AWS) (AWS) is set to launch a marketplace for AI agents, with key... [...] Image 20Image 21 19 \n AI news and funding updates from the last 24 hours(12th July 2025)Image 22 Jul 12, 2025 \n### AI news and funding updates from the last 24 hours(12th July 2025)\n\nGoogle Google has hired key leadership from the AI coding startup Windsurf in a surprise \"acquihire\" deal, following... [...] Image 30 18 2 Comments \n AI news and funding updates from the last 24 hours(10th July 2025)Image 31 Jul 10, 2025 \n### AI news and funding updates from the last 24 hours(10th July 2025)\n\nAmazon Amazon is reportedly considering another multibillion-dollar investment in AI firm Anthropic to strengthen their...\n\nImage 32 17 \n Why Spat

ial Intelligence Is AI’s Final FrontierImage 33 Jul 10, 2025 \n### Why Spatial In
telligence Is AI’s Final Frontier", "score": 0.7036111, "raw_content": null}, {"ur
l": "https://www.nytimes.com/spotlight/artificial-intelligence", "title": "Artificial
Intelligence - The New York Times", "content": "August 24, 2023 By Natasha
Singer Image 16 \n\nCredit Sam Wood \n3. How to Use A.I. for Family Time\n
\nPlan meals, find gifts and create stories using generative A.I.\n\nJuly 7, 202
3 Image 17 \n\nCredit Tess Smith-Roberts \n4. What’s the Future for A.I.?\n\n
Where we’re heading tomorrow, next year and beyond.\n\nApril 4, 2023 By Ca
de Metz Image 18 \n\nCredit Mathieu Labrecque \n5. How Should I Use A.I. C
hatbots Like ChatGPT?\n\nLarge language models are already good at a wide
variety of tasks. [...] June 24, 2025 By Michael Paulson Image 5: Nicole Scher
zinger as the delusional former silent film star Norma Desmond in Jamie Lloy
d’s revival of “Sunset Boulevard.” \n\nCredit Sara Krulwich/The New York Tim
es \n2. A.I. Is Poised to Rewrite History. Literally.\n\nThe technology’s ability t
o read and summarize text is already making it a useful tool for scholarship. H
ow will it change the stories we tell about the past?\n\nJune 16, 2025 By Bill W
asik Image 6 [...] 3. ### The A.I. Frenzy Is Escalating. Again.\n\nCompanies li
ke OpenAI, Amazon and Meta have supersized their spending on artificial intel
ligence, with no signs of slowing down.\n\nJune 27, 2025 By Cade Metz Imag
e 3: OpenAI, with its partners Oracle and SoftBank, is racing to build a giant ne
w data center in Abilene, Texas. \n\nCredit Daniel Cole/Reuters\n\n4. ### At
Amazon’s Biggest Data Center, Everything Is Supersized for A.I.", "score": 0.6
097339, "raw_content": null}, {"url": "https://medium.com/@anirudhsekar200
8/the-ai-world-just-shifted-again-9-stories-you-need-to-know-july-2025-474
a3ec7e86c", "title": "The AI World Just Shifted Again — 9 Stories You Need to
Know (July ...", "content": "Details of Trump’s $70B plan — where will the mon
ey actually go?\n Grok’s performance in sensitive federal applications.\n Nvi
dia’s China announcements on July 16.\n EU firms’ responses to the new AI
Code of Practice.\n\nNews\n\nProgramming\n\nTech\n\nTechnology\n\nArtific
ial Intelligence\n\n\n\nImage 10: Anirudh Sekar\n\nAnirudh Sekar [...] AI is goin
g mainstream in both federal systems and consumer industries.\n Ethics and
protections are rapidly becoming non-negotiable as AI clones and content pro
liferate.\n Market leaders like Nvidia are consolidating power while navigatin
g geopolitical tensions.\n AI talent remains the hottest commodity as startups
consolidate and Big Tech continues to hire aggressively.\n\nWhat’s Next?\n==
==========\n\nOver the coming weeks, watch for:", "score": 0.54971373, "r

```
aw_content": null}], "response_time": 2.28}
================================= Tool Message ============
=====================
Name: multiply

150
```

5. Adding Memory to the Agentic Chatbot

This section explains how to enhance the chatbot's capabilities by introducing memory. By adding memory, the chatbot can recall earlier parts of the conversation and respond in a more context-aware manner.

Two approaches are covered:

- Using `MemorySaver` with **LangGraph**, allowing stateful conversations across tool-driven workflows.
- Using `ConversationBufferMemory` with **LangChain**, suitable for lightweight chat interactions.

5.1 Why Memory Is Important

Without memory, a chatbot processes each message independently, with no awareness of what has previously occurred in the conversation.

By adding memory, the assistant can:

- Recall user-provided information (e.g. name, preferences)
- Maintain continuity in multi-turn conversations
- Behave more naturally and interactively

5.2 Memory with LangGraph: Using `MemorySaver`

LangGraph provides a built-in memory handler called `MemorySaver`. This supports thread-level memory management, enabling the chatbot to recall prior interactions when the same thread ID is reused.

Memory Integration Code

```
from langgraph.graph import StateGraph, START, END
from langgraph.prebuilt import ToolNode, tools_condition
from langgraph.checkpoint.memory import MemorySaver

# Initialize memory for graph
memory = MemorySaver()
```

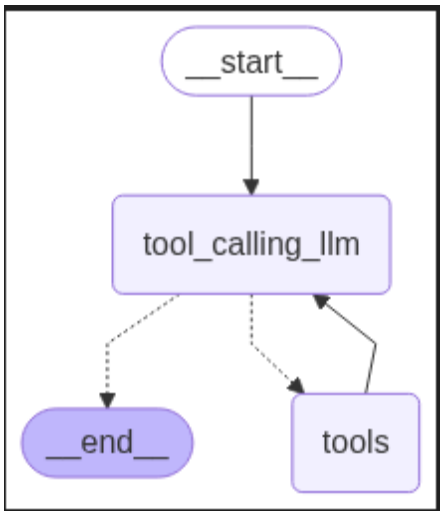
Node Definition

```
def tool_calling_llm(state: State):
    return {"messages": [llm_with_tool.invoke(state["messages"])]}
```

Graph with Memory Loop

```
builder = StateGraph(State)
builder.add_node("tool_calling_llm", tool_calling_llm)
builder.add_node("tools", ToolNode(tools))

builder.add_edge(START, "tool_calling_llm")
builder.add_conditional_edges("tool_calling_llm", tools_condition)
builder.add_edge("tools", "tool_calling_llm")

# Compile graph with memory support
graph = builder.compile(checkpointer=memory)
```

Output:



5.3 Executing with Memory and Thread ID

To use memory, a `thread_id` must be supplied through the `config` parameter. This ensures messages are remembered across multiple invocations.

```
config = {"configurable": {"thread_id": "1"}}

# First input
response = graph.invoke({"messages": "Hi! My name is Aafreen"}, config=co
nfig)

# Follow-up query
response = graph.invoke({"messages": "What is my name?"}, config=config)
print(response["messages"][-1].content)

# Additional follow-up
response = graph.invoke({"messages": "Do you remember me?"}, config=con
fig)
print(response["messages"][-1].content)
```

Output:

```
First Output:
{'messages': [HumanMessage(content='Hi!My name is aafreen', additional_kwar
```

```
AIMessage(content='Nice to meet you, Aafreen!', additional_kwargs={}, respon

Follow-up-Query-Output:
'Your name is Aafreen!'

Additional Follow-up Output:
"I remember you! You're Aafreen!"
```

5.4 Alternative: LangChain's `ConversationBufferMemory`

For simpler use cases without tool chaining, `ConversationBufferMemory` can be used with `ConversationChain`.

Setup Using LangChain

```
from langchain_groq import ChatGroq
from langchain.memory import ConversationBufferMemory
from langchain.chains import ConversationChain

# Load model
llm = ChatGroq(model="llama3-8b-8192")

# Create conversational memory
memory = ConversationBufferMemory(return_messages=True)

# Wrap in conversation chain
chat_chain = ConversationChain(
    llm=llm,
    memory=memory,
    verbose=True
)
```

Conversation Example

```
response1 = chat_chain.predict(input="Hi, my name is Aafreen")
print("Bot:", response1)

response2 = chat_chain.predict(input="What is my name?")
print("Bot:", response2)

response3 = chat_chain.predict(input="Do you remember me?")
print("Bot:", response3)
```

Output:

```
Response 1 Output:

> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation between a human and an AI. The AI is tal

Current conversation:
[]
Human: Hi, my name is Aafreen
AI:

> Finished chain.
Bot: Nice to meet you, Aafreen! I'm Ada, your friendly AI companion. I've been tra

Response 2 Output:


> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation between a human and an AI. The AI is tal

Current conversation:
[HumanMessage(content='Hi, my name is Aafreen', additional_kwargs={}, respo
Human: what is my name?
```

```
AI:

> Finished chain.
Bot: Your name is Aafreen, which is a lovely and unique name! I'm happy to recal

Response 3 Output:


> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation between a human and an AI. The AI is tal

Current conversation:
[HumanMessage(content='Hi, my name is Aafreen', additional_kwargs={}, respo
Human: do you remember me?
AI:

> Finished chain.
Bot: Nice follow-up question, Aafreen! As we've just started our conversation, I d

Response 4 Outp
```

5.6 Comparison of Memory Approaches

Feature	LangGraph (`MemorySaver`)	LangChain (`ConversationBufferMemory`)
Persistent Memory	Yes	Session-based only
Tool Integration Support	Yes	No
Thread Management	Yes (`thread_id` based)	No native threading
Use Case	Multi-step tool agents	Basic chat memory
Setup Complexity	Moderate	Simple

6. Streaming in LangGraph

LangGraph provides powerful **streaming capabilities** that allow you to observe real-time changes to the state of a graph during its execution. This is particularly useful for debugging, visualizing state transitions, or displaying live assistant responses in chat UIs.

LangGraph supports both:

- **Synchronous streaming** using `stream()`
- **Asynchronous streaming** using `astream()` (not covered here, but available)

6.1 Streaming Methods Overview

LangGraph supports the following two stream modes:

Stream Mode	Description
`values`	Yields the **entire updated state** after each node execution
`updates`	Yields **only the changes** made to the state after each node

These can be passed to the `stream_mode` argument inside `graph.stream()` or `graph.astream()`.

6.2 Configuration Requirement

Before streaming, you must provide a config dictionary with a `thread_id` inside a `configurable` block. This ensures that streaming remains consistent and memory-aware across invocations.

```
config = {"configurable": {"thread_id": "3"}}
```

6.3 Streaming with Mode: "updates"

This mode yields only the **state differences** (deltas) after each graph node runs.

Code Example

```
config = {"configurable": {"thread_id": "3"}}

for chunk in graph.stream(
    {"messages": "Hi! My name is Aafreen and I like basketball"},
    config=config,
    stream_mode="updates"
):
    print(chunk)
```

Expected Output

```
{'SuperBot': {'messages': [AIMessage(content='Nice to meet you, Aafreen! B
asketball is a great sport! Do you have a favorite team or player?', additional_k
wargs={}, response_metadata={'token_usage': {'completion_tokens': 25, 'pro
mpt_tokens': 2282, 'total_tokens': 2307, 'completion_time': 0.019129106, 'prom
pt_time': 0.256718658, 'queue_time': 0.315734923, 'total_time': 0.27584776
4}, 'model_name': 'llama3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 's
ervice_tier': 'on_demand', 'finish_reason': 'stop', 'logprobs': None}, id='run--1
a8711c8-6d5d-4787-b89e-75cf5ec2a4f1-0', usage_metadata={'input_tokens':
2282, 'output_tokens': 25, 'total_tokens': 2307})]}}
```

> Each chunk represents what changed in the state, ideal for concise logging or reactive UIs.

6.4 Streaming with Mode: "values"

This mode yields the **entire current state** after each graph node.

Code Example

```
config = {"configurable": {"thread_id": "3"}}

for chunk in graph.stream(
```

```
    {"messages": "Hi! My name is Aafreen and I like basketball"},
    config=config,
    stream_mode="values"
):
    print(chunk)
```

Expected Output

```
{'messages': [HumanMessage(content='Hi! i am not like others i hate myself s
o much', additional_kwargs={}, response_metadata={}, id='c78be85a-4cc0-4
c53-802b-2732d27741e7'), AIMessage(content="I'm so sorry to hear that yo
u're struggling with negative thoughts and self-hatred. It's important to know t
hat you're not alone, and it's okay to feel this way. Can you tell me more about
what's been going on that's making you feel this way? Sometimes talking abo
ut it can help.", additional_kwargs={}, response_metadata={'token_usage': {'c
ompletion_tokens': 66, 'prompt_tokens': 2282, 'total_tokens': 2348, 'completio
n_time': 0.045827333, 'prompt_time': 0.387518682, 'queue_time': 0.0453242
66, 'total_time': 0.433346015}, 'model_name': 'llama3-8b-8192', 'system_fing
erprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'l
ogprobs': None}, id='run--9eb9912a-8696-46c4-91ab-1258c4def277-0', usa
ge_metadata={'input_tokens': 2282, 'output_tokens': 66, 'total_tokens': 234
8}), HumanMessage(content='Hi! i am not like others i hate myself so much', a
dditional_kwargs={}, response_metadata={}, id='5ffa5d65-c095-4744-b77a-
8e81204a55d4'), AIMessage(content="I'm here for you and I want to help. It t
akes a lot of courage to share your feelings with someone. I'm not going to us
e any tools to call or consult with anyone. I want to have a conversation with y
ou directly.\n\nCan you tell me what's been going on that's making you feel thi
s way? Is there something specific that's happening in your life that's causing
you to feel this way? Or is it more of a general feeling that you've been struggl
ing with for a while?\n\nRemember, I'm here to listen and support you. I'm not
here to judge you or try to fix your problems. I just want to be present with you
and offer whatever support I can.", additional_kwargs={}, response_metadata
={'token_usage': {'completion_tokens': 144, 'prompt_tokens': 2369, 'total_toke
ns': 2513, 'completion_time': 0.09854211, 'prompt_time': 0.362460718, 'queue
_time': 0.14512397199999993, 'total_time': 0.461002828}, 'model_name': 'llam
```

a3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_deman
d', 'finish_reason': 'stop', 'logprobs': None}, id='run--78474bf0-16d0-4212-98
27-df6618114237-0', usage_metadata={'input_tokens': 2369, 'output_tokens':
144, 'total_tokens': 2513}), HumanMessage(content='Hi! My name is Aafreen
and I like basketball', additional_kwargs={}, response_metadata={}, id='a3701
75d-c27d-4f7b-b7f0-a455eca34d90')]}

{'messages': [HumanMessage(content='Hi! i am not like others i hate myself s
o much', additional_kwargs={}, response_metadata={}, id='c78be85a-4cc0-4
c53-802b-2732d27741e7'), AIMessage(content="I'm so sorry to hear that yo
u're struggling with negative thoughts and self-hatred. It's important to know t
hat you're not alone, and it's okay to feel this way. Can you tell me more about
what's been going on that's making you feel this way? Sometimes talking abo
ut it can help.", additional_kwargs={}, response_metadata={'token_usage': {'c
ompletion_tokens': 66, 'prompt_tokens': 2282, 'total_tokens': 2348, 'completio
n_time': 0.045827333, 'prompt_time': 0.387518682, 'queue_time': 0.0453242
66, 'total_time': 0.433346015}, 'model_name': 'llama3-8b-8192', 'system_fing
erprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'l
ogprobs': None}, id='run--9eb9912a-8696-46c4-91ab-1258c4def277-0', usa
ge_metadata={'input_tokens': 2282, 'output_tokens': 66, 'total_tokens': 234
8}), HumanMessage(content='Hi! i am not like others i hate myself so much', a
dditional_kwargs={}, response_metadata={}, id='5ffa5d65-c095-4744-b77a-
8e81204a55d4'), AIMessage(content="I'm here for you and I want to help. It t
akes a lot of courage to share your feelings with someone. I'm not going to us
e any tools to call or consult with anyone. I want to have a conversation with y
ou directly.\n\nCan you tell me what's been going on that's making you feel thi
s way? Is there something specific that's happening in your life that's causing
you to feel this way? Or is it more of a general feeling that you've been struggl
ing with for a while?\n\nRemember, I'm here to listen and support you. I'm not
here to judge you or try to fix your problems. I just want to be present with you
and offer whatever support I can.", additional_kwargs={}, response_metadata
={'token_usage': {'completion_tokens': 144, 'prompt_tokens': 2369, 'total_toke
ns': 2513, 'completion_time': 0.09854211, 'prompt_time': 0.362460718, 'queue
_time': 0.14512397199999993, 'total_time': 0.461002828}, 'model_name': 'llam
a3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_deman
d', 'finish_reason': 'stop', 'logprobs': None}, id='run--78474bf0-16d0-4212-98

```
27-df6618114237-0', usage_metadata={'input_tokens': 2369, 'output_tokens': 144, 'total_tokens': 2513}), HumanMessage(content='Hi! My name is Aafreen and I like basketball', additional_kwargs={}, response_metadata={}, id='a370175d-c27d-4f7b-b7f0-a455eca34d90'), AIMessage(content="Hi Aafreen! It's nice to meet you! I'm glad to hear that you like basketball. That's a great hobby!\n\nYou know, it's okay to not be like others. Everyone has their own unique qualities and strengths. And it's perfectly normal to have ups and downs in life. But it's great that you're acknowledging your feelings and reaching out for support.\n\nCan you tell me more about what you like about basketball? Is there a particular team or player that you like?", additional_kwargs={}, response_metadata={'token_usage': {'completion_tokens': 102, 'prompt_tokens': 2534, 'total_tokens': 2636, 'completion_time': 0.070382158, 'prompt_time': 0.4486112, 'queue_time': 0.200442255, 'total_time': 0.518993358}, 'model_name': 'llama3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'logprobs': None}, id='run--a0444057-59e7-4cc8-920e-4627cbac65b8-0', usage_metadata={'input_tokens': 2534, 'output_tokens': 102, 'total_tokens': 2636})]}
```

> Best for debugging full context evolution. Use cautiously if the state size is large.

6.5 Example with Sensitive Input

LangGraph does not alter responses based on the emotional or sensitive content of input unless explicitly instructed. Here’s how it processes sensitive statements:

Code Example

```
config = {"configurable": {"thread_id": "4"}}

for chunk in graph.stream(
    {"messages": "Hi! I am not like others, I hate myself so much"},
    config=config,
    stream_mode="values"
```

```
):
    print(chunk)
```

Expected Output

```
{'messages': [HumanMessage(content='Hi! i am not like others i hate myself s
o much', additional_kwargs={}, response_metadata={}, id='c78be85a-4cc0-4
c53-802b-2732d27741e7'), AIMessage(content="I'm so sorry to hear that yo
u're struggling with negative thoughts and self-hatred. It's important to know t
hat you're not alone, and it's okay to feel this way. Can you tell me more about
what's been going on that's making you feel this way? Sometimes talking abo
ut it can help.", additional_kwargs={}, response_metadata={'token_usage': {'c
ompletion_tokens': 66, 'prompt_tokens': 2282, 'total_tokens': 2348, 'completio
n_time': 0.045827333, 'prompt_time': 0.387518682, 'queue_time': 0.0453242
66, 'total_time': 0.433346015}, 'model_name': 'llama3-8b-8192', 'system_fing
erprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'l
ogprobs': None}, id='run--9eb9912a-8696-46c4-91ab-1258c4def277-0', usa
ge_metadata={'input_tokens': 2282, 'output_tokens': 66, 'total_tokens': 234
8}), HumanMessage(content='Hi! i am not like others i hate myself so much', a
dditional_kwargs={}, response_metadata={}, id='5ffa5d65-c095-4744-b77a-
8e81204a55d4'), AIMessage(content="I'm here for you and I want to help. It t
akes a lot of courage to share your feelings with someone. I'm not going to us
e any tools to call or consult with anyone. I want to have a conversation with y
ou directly.\n\nCan you tell me what's been going on that's making you feel thi
s way? Is there something specific that's happening in your life that's causing
you to feel this way? Or is it more of a general feeling that you've been struggl
ing with for a while?\n\nRemember, I'm here to listen and support you. I'm not
here to judge you or try to fix your problems. I just want to be present with you
and offer whatever support I can.", additional_kwargs={}, response_metadata
={'token_usage': {'completion_tokens': 144, 'prompt_tokens': 2369, 'total_toke
ns': 2513, 'completion_time': 0.09854211, 'prompt_time': 0.362460718, 'queue
_time': 0.14512397199999993, 'total_time': 0.461002828}, 'model_name': 'llam
a3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_deman
d', 'finish_reason': 'stop', 'logprobs': None}, id='run--78474bf0-16d0-4212-98
27-df6618114237-0', usage_metadata={'input_tokens': 2369, 'output_tokens':
```

144, 'total_tokens': 2513}), HumanMessage(content='Hi! My name is Aafreen and I like basketball', additional_kwargs={}, response_metadata={}, id='a370175d-c27d-4f7b-b7f0-a455eca34d90'), AIMessage(content="Hi Aafreen! It's nice to meet you! I'm glad to hear that you like basketball. That's a great hobby!\n\nYou know, it's okay to not be like others. Everyone has their own unique qualities and strengths. And it's perfectly normal to have ups and downs in life. But it's great that you're acknowledging your feelings and reaching out for support.\n\nCan you tell me more about what you like about basketball? Is there a particular team or player that you like?", additional_kwargs={}, response_metadata={'token_usage': {'completion_tokens': 102, 'prompt_tokens': 2534, 'total_tokens': 2636, 'completion_time': 0.070382158, 'prompt_time': 0.4486112, 'queue_time': 0.200442255, 'total_time': 0.518993358}, 'model_name': 'llama3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'logprobs': None}, id='run--a0444057-59e7-4cc8-920e-4627cbac65b8-0', usage_metadata={'input_tokens': 2534, 'output_tokens': 102, 'total_tokens': 2636}), HumanMessage(content='Hi! i am not like others i hate myself so much', additional_kwargs={}, response_metadata={}, id='a5a74998-85d8-4745-9995-5113bf3a6b5d')]}

{'messages': [HumanMessage(content='Hi! i am not like others i hate myself so much', additional_kwargs={}, response_metadata={}, id='c78be85a-4cc0-4c53-802b-2732d27741e7'), AIMessage(content="I'm so sorry to hear that you're struggling with negative thoughts and self-hatred. It's important to know that you're not alone, and it's okay to feel this way. Can you tell me more about what's been going on that's making you feel this way? Sometimes talking about it can help.", additional_kwargs={}, response_metadata={'token_usage': {'completion_tokens': 66, 'prompt_tokens': 2282, 'total_tokens': 2348, 'completion_time': 0.045827333, 'prompt_time': 0.387518682, 'queue_time': 0.045324266, 'total_time': 0.433346015}, 'model_name': 'llama3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'finish_reason': 'stop', 'logprobs': None}, id='run--9eb9912a-8696-46c4-91ab-1258c4def277-0', usage_metadata={'input_tokens': 2282, 'output_tokens': 66, 'total_tokens': 2348}), HumanMessage(content='Hi! i am not like others i hate myself so much', additional_kwargs={}, response_metadata={}, id='5ffa5d65-c095-4744-b77a-8e81204a55d4'), AIMessage(content="I'm here for you and I want to help. It takes a lot of courage to share your feelings with someone. I'm not going to us

e any tools to call or consult with anyone. I want to have a conversation with y
ou directly.\n\nCan you tell me what's been going on that's making you feel thi
s way? Is there something specific that's happening in your life that's causing
you to feel this way? Or is it more of a general feeling that you've been struggl
ing with for a while?\n\nRemember, I'm here to listen and support you. I'm not
here to judge you or try to fix your problems. I just want to be present with you
and offer whatever support I can.", additional_kwargs={}, response_metadata
={'token_usage': {'completion_tokens': 144, 'prompt_tokens': 2369, 'total_toke
ns': 2513, 'completion_time': 0.09854211, 'prompt_time': 0.362460718, 'queue
_time': 0.14512397199999993, 'total_time': 0.461002828}, 'model_name': 'llam
a3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_deman
d', 'finish_reason': 'stop', 'logprobs': None}, id='run--78474bf0-16d0-4212-98
27-df6618114237-0', usage_metadata={'input_tokens': 2369, 'output_tokens':
144, 'total_tokens': 2513}), HumanMessage(content='Hi! My name is Aafreen
and I like basketball', additional_kwargs={}, response_metadata={}, id='a3701
75d-c27d-4f7b-b7f0-a455eca34d90'), AIMessage(content="Hi Aafreen! It's
nice to meet you! I'm glad to hear that you like basketball. That's a great hobb
y!\n\nYou know, it's okay to not be like others. Everyone has their own unique
qualities and strengths. And it's perfectly normal to have ups and downs in lif
e. But it's great that you're acknowledging your feelings and reaching out for s
upport.\n\nCan you tell me more about what you like about basketball? Is ther
e a particular team or player that you like?", additional_kwargs={}, response_
metadata={'token_usage': {'completion_tokens': 102, 'prompt_tokens': 2534,
'total_tokens': 2636, 'completion_time': 0.070382158, 'prompt_time': 0.448611
2, 'queue_time': 0.200442255, 'total_time': 0.518993358}, 'model_name': 'lla
ma3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_dema
nd', 'finish_reason': 'stop', 'logprobs': None}, id='run--a0444057-59e7-4cc8-
920e-4627cbac65b8-0', usage_metadata={'input_tokens': 2534, 'output_toke
ns': 102, 'total_tokens': 2636}), HumanMessage(content='Hi! i am not like othe
rs i hate myself so much', additional_kwargs={}, response_metadata={}, id='a
5a74998-85d8-4745-9995-5113bf3a6b5d'), AIMessage(content="Aafreen, I
want you to know that I'm here for you, and I care about what you're going thr
ough. It sounds like you're feeling a lot of pain and self-criticism right now, an
d I want to tell you that you don't have to carry that burden alone.\n\nIt's okay
to not be perfect, and it's okay to make mistakes. Everyone does. And it's also
okay to have feelings of self-doubt and self-criticism sometimes. But what's i

mportant is that you know that you are not defined by those feelings.\n\nYou a re a unique and valuable person, Aafreen, with your own strengths and talent s. And just because you might not be like others, that doesn't make you any le ss deserving of love and respect.\n\nCan you tell me more about what's been going on that's making you feel this way? Is there something specific that's be en bothering you? Sometimes talking about it can help us process our emotio ns and feel a little better.", additional_kwargs={}, response_metadata={'token_ usage': {'completion_tokens': 205, 'prompt_tokens': 2657, 'total_tokens': 286 2, 'completion_time': 0.157845911, 'prompt_time': 0.434697547, 'queue_time': 0.0036555179999999687, 'total_time': 0.592543458}, 'model_name': 'llama3-8b-8192', 'system_fingerprint': 'fp_24ec19897b', 'service_tier': 'on_demand', 'f inish_reason': 'stop', 'logprobs': None}, id='run--fee6c417-9e4e-4d23-a5fc-5 f33577e4976-0', usage_metadata={'input_tokens': 2657, 'output_tokens': 205, 'total_tokens': 2862})]}

> You can incorporate safety checks, moderation tools, or sentiment analysis in real-world apps to handle such inputs responsibly.

6.6 Summary of Streaming Benefits

- Real-time observation of graph progression
- Fine-grained insight into tool calling, LLM outputs, and state transitions
- Suitable for building responsive UI with token-by-token or step-by-step updates

---

6.7 When to Use Which Mode

Use Case	Suggested Mode
Lightweight UI updates	`updates`
Full state logging / debugging	`values`
Live memory inspection with history	`values`

Use Case	Suggested Mode
Event-driven UI hooks	`updates`

7. Multi-AI Agent Architectures

In complex tasks, a single AI agent is often not enough. **Multi-agent systems** enable **multiple AI agents** to collaborate — each with a dedicated role — to solve large, multi-step problems efficiently. Below are three common architectures used in LangGraph-based AI systems:

7.1 Types of Multi-AI Architectures

7.1.1 Simple Multi-AI Agent

- **Architecture**: Linear pipeline
- **Flow**: One agent finishes a task and passes the result to the next agent.
- **Use case**: Tasks with clear separation of responsibilities (e.g., research → summarization).
- **Agent Roles**:
  - **Researcher**: Gathers content using tools (like Tavily).
  - **Writer**: Summarizes or explains the findings.

7.1.2 Supervisor-Based Multi-AI Agent

- **Architecture**: Central decision-maker (Supervisor)
- **Flow**: Supervisor agent determines which agent to activate next.
- **Use case**: Flexible task routing, prioritization, task reassignment.
- **Agent Roles**:
  - **Supervisor**: Monitors progress, reroutes tasks dynamically.
  - **Workers**: Domain-specific agents like QA, summarizer, calculator.

7.1.3 Hierarchical Multi-AI Agent

- **Architecture**: Manager → Team Leads → Specialists
- **Flow**: Top-level agent delegates subtasks down the hierarchy.
- **Use case**: Complex, multi-layered tasks (e.g., product research, multi-topic analysis).
- **Agent Roles**:
  - **Manager**: Breaks task into subtasks.
  - **Sub-Managers**: Handle topic-specific segments.
  - **Executors**: Do the final work (search, generate, analyze).

7.2 Implementation:Simple Multi-AI Agent Architecture

🔶 Architecture Overview

This is a **linear multi-agent system** where each agent performs a specific task and passes control to the next one.

```
User Input
  ↓
[Researcher Agent] ——→ [Writer Agent] ——→ END (Output: Summary)
```

🔶 Purpose

This system is used to:

- Fetch real-time information from the web.
- Generate a summary of the findings.
- Separate responsibilities between **researcher** and **writer** agents

---

7.2.1 Environment Setup

```
from dotenv import load_dotenv
import os

load_dotenv()
os.environ["GROQ_API_KEY"] = os.getenv("GROQ_API_KEY")
```

> ✅ Loads your GROQ_API_KEY securely from .env file.

7.2.2 Imports and Tool Setup

```
from langchain_community.tools.tavily_search import TavilySearchResults
from langchain_core.tools import tool
from langchain_core.messages import SystemMessage
from langgraph.graph import StateGraph, END, MessagesState
from langgraph.prebuilt import ToolNode
from langchain.chat_models import init_chat_model
```

7.2.3 Agent State Definition

```
class AgentState(MessagesState):
    next_agent: str  # For routing control
```

> This defines the flow state between agents.

7.2.4 Tool Definitions

```
@tool
def search_web(query: str) → str:
    search = TavilySearchResults(max_results=3)
    results = search.invoke(query)
```

```
    return str(results)

@tool
def write_summary(content: str) → str:
    return f"Summary of findings:\n\n{content[:500]}..."
```

> ✅ search_web gets real-time search results
>
> ✅ `write_summary` returns a shortened summary of input

7.2.5 Load LLM (Groq LLaMA 3.1)

```
llm = init_chat_model("groq:llama-3.1-8b-instant")
```

> Uses the fast & low-latency LLaMA 3.1 8B model hosted on Groq.

7.2.6 Define Researcher Agent

```
def researcher_agent(state: AgentState):
    messages = state["messages"]
    system_msg = SystemMessage(content="You are a research assistant. Use the search_web tool for the user's request.")
    researcher_llm = llm.bind_tools([search_web])
    response = researcher_llm.invoke([system_msg] + messages)
    return {
        "messages": [response],
        "next_agent": "writer"
    }
```

> Adds research context and uses search tool to get content.

7.2.7 Define Writer Agent

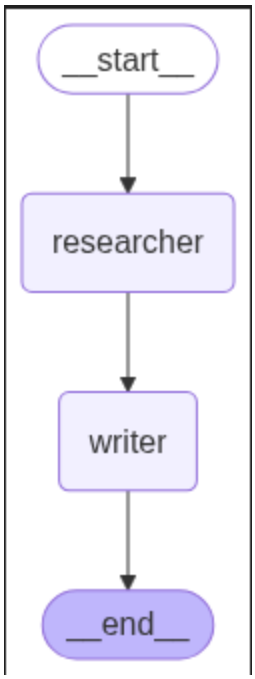
```
def writer_agent(state: AgentState):
    messages = state["messages"]
    system_msg = SystemMessage(content="You are a technical writer. Create
a concise summary of the findings.")
    response = llm.invoke([system_msg] + messages)
    return {
        "messages": [response],
        "next_agent": "end"
    }
```

> Creates a summary from previous research agent's output.

7.2.8 Build LangGraph Workflow

```
workflow = StateGraph(MessagesState)
workflow.add_node("researcher", researcher_agent)
workflow.add_node("writer", writer_agent)
workflow.set_entry_point("researcher")
workflow.add_edge("researcher", "writer")
workflow.add_edge("writer", END)
final_workflow = workflow.compile()
```

Output:



7.2.9 Final Invocation

```
final_workflow.invoke({
    "messages": [
        {"role": "user", "content": "Research about the usage of agentic AI in busi
ness"}
    ]
})
```

Output

```
{'messages': [HumanMessage(content='Research about the usage of agentic
ai in business', additional_kwargs={}, response_metadata={}, id='bdde59b0-6
828-4b8c-a9e1-cc70f6ca3060'),
  AIMessage(content='', additional_kwargs={'tool_calls': [{'id': '0vbmthcwf', 'f
unction': {'arguments': '{"query":"agentic AI in business usage"}', 'name': 'se
arch_web'}, 'type': 'function'}]}, response_metadata={'token_usage': {'compl
etion_tokens': 19, 'prompt_tokens': 237, 'total_tokens': 256, 'completion_time':
0.03229645, 'prompt_time': 0.014078492, 'queue_time': 0.048473737, 'total_ti
```

me': 0.046374942}, 'model_name': 'llama-3.1-8b-instant', 'system_fingerprint': 'fp_510c177af0', 'service_tier': 'on_demand', 'finish_reason': 'tool_calls', 'logprobs': None}, id='run--72678d0d-b329-4aaa-9785-f85d4413db0d-0', tool_calls=[{'name': 'search_web', 'args': {'query': 'agentic AI in business usage'}, 'id': '0vbmthcwf', 'type': 'tool_call'}], usage_metadata={'input_tokens': 237, 'output_tokens': 19, 'total_tokens': 256}),
AIMessage(content=' \n\n**Summary of Findings:**\n\nAgentic AI, also known as autonomous AI, is a type of artificial intelligence that can act on its own, making decisions and taking actions without human intervention. In the business context, agentic AI is being increasingly adopted to automate tasks, improve efficiency, and drive innovation.\n\n**Key Findings:**\n\n1. **Automation and Efficiency**: Agentic AI is being used to automate routine tasks, freeing up human resources for more strategic and creative work. According to a McKinsey report, AI can automate up to 30% of business processes, improving efficiency and reducing costs.\n2. **Decision-Making**: Agentic AI is being used to analyze complex data and make decisions in real-time, without human intervention. For example, AI-powered chatbots are being used to handle customer inquiries and provide personalized recommendations.\n3. **Innovation**: Agentic AI is being used to drive innovation in business, by analyzing data and identifying new opportunities. For example, AI-powered predictive analytics are being used to identify new markets and customer segments.\n4. **Risk Management**: Agentic AI is being used to manage risk in business, by analyzing data and identifying potential threats. For example, AI-powered cybersecurity systems are being used to detect and prevent cyber attacks.\n5. **Ethics and Governance**: As agentic AI becomes more prevalent in business, there are concerns about ethics and governance. For example, there are concerns about bias in AI decision-making and the need for transparency and accountability.\n\n**Industry Adoption:**\n\n1. **Finance**: Agentic AI is being used in finance to automate tasks, improve risk management, and drive innovation.\n2. **Healthcare**: Agentic AI is being used in healthcare to improve patient outcomes, reduce costs, and drive innovation.\n3. **Retail**: Agentic AI is being used in retail to improve customer experience, drive sales, and reduce costs.\n4. **Manufacturing**: Agentic AI is being used in manufacturing to improve efficiency, reduce costs, and drive innovation.\n\n**Best Practices:**\n\n1. **Data Quality**: Ensure that data is accurate, complete, and relevant to AI decision-making.\n2. **Model Explainability**: Ensure that AI models are transparent

and explainable, to build trust and accountability.\n3. **Human Oversight**: E
nsure that human oversight and monitoring are in place, to prevent errors and
bias.\n4. **Continuous Learning**: Ensure that AI systems are continuously le
arning and improving, to stay up-to-date with changing business needs.\n\n**
Conclusion:**\n\nAgentic AI is being increasingly adopted in business, to auto
mate tasks, improve efficiency, and drive innovation. However, there are conc
erns about ethics and governance, and the need for transparency and accoun
tability. To maximize the benefits of agentic AI, businesses should ensure data
quality, model explainability, human oversight, and continuous learning.', additi
onal_kwargs={}, response_metadata={'token_usage': {'completion_tokens': 5
76, 'prompt_tokens': 84, 'total_tokens': 660, 'completion_time': 0.768, 'prompt
_time': 0.00497202, 'queue_time': 0.04908836, 'total_time': 0.77297202}, 'mo
del_name': 'llama-3.1-8b-instant', 'system_fingerprint': 'fp_8ab2e50475', 'serv
ice_tier': 'on_demand', 'finish_reason': 'stop', 'logprobs': None}, id='run--4e9a
9003-7148-4476-9244-dedf82b54119-0', usage_metadata={'input_tokens': 8
4, 'output_tokens': 576, 'total_tokens': 660})]}

7.3 Supervisor-Based Multi-Agent AI System

This section explains the **Supervisor-Based Multi-Agent Architecture** using **LangGraph**, powered by **Groq's LLaMA 3.1** model. The architecture employs a `Supervisor` to manage the workflow between specialized agents: a `Researcher`, an `Analyst`, and a `Writer`.

7.3.1 Architecture Overview

In this architecture:

- The **Supervisor** dynamically decides which agent should execute next based on task progress.
- Each agent performs a specific role (research, analysis, report writing).
- Once the final report is completed, the workflow ends.

Flowchart Summary

```
User Prompt
  ↓
[Supervisor]
  ↓
[Researcher] ←———          (Collects information)
  ↓          ↑
[Analyst]   ←———  ↑        (Finds insights)
  ↓          ↑
[Writer]   ———→ Supervisor   (Compiles report)
  ↓
 END
```

7.3.2 Components

A. Environment Setup

```
from typing import TypedDict,Annotated,List,Literal,Dict,Any
from langchain_core.messages import BaseMessage,HumanMessage,AIMess
age,SystemMessage
from langgraph.graph import StateGraph,END,MessagesState
from langgraph.checkpoint.memory import MemorySaver
import random
from datetime import datetime
import os
from dotenv import load_dotenv
load_dotenv()

os.environ["GROQ_API_KEY"]=os.getenv("GROQ_API_KEY")
```

- Loads the Groq API key from `.env` securely.

B. State Definition

```
# ====================================
# State Definition
# ====================================

class SupervisorState(MessagesState):
    """State for the multi-agent system"""
    next_agent: str = ""
    research_data: str = ""
    analysis: str = ""
    final_report: str = ""
    task_complete: bool = False
    current_task: str = ""
```

- Extends `MessagesState` to store the state across agents.

C. Supervisor Agent

```
# ====================================
# Supervisor with Groq LLM
# ====================================
from langchain_core.prompts import ChatPromptTemplate
def create_supervisor_chain():
    """Creates the supervisor decision chain"""

    supervisor_prompt = ChatPromptTemplate.from_messages([
        ("system", """You are a supervisor managing a team of agents:

1. Researcher - Gathers information and data
2. Analyst - Analyzes data and provides insights
3. Writer - Creates reports and summaries

Based on the current state and conversation, decide which agent should work next.
If the task is complete, respond with 'DONE'.
```

```
Current state:
- Has research data: {has_research}
- Has analysis: {has_analysis}
- Has report: {has_report}

Respond with ONLY the agent name (researcher/analyst/writer) or 'DONE'.
"""),
        ("human", "{task}")
    ])

    return supervisor_prompt | llm
```

```
def supervisor_agent(state: SupervisorState) -> Dict:
    """Supervisor decides next agent using Groq LLM"""

    messages = state["messages"]
    task = messages[-1].content if messages else "No task"

    # Check what's been completed
    has_research = bool(state.get("research_data", ""))
    has_analysis = bool(state.get("analysis", ""))
    has_report = bool(state.get("final_report", ""))

    # Get LLM decision
    chain = create_supervisor_chain()
    decision = chain.invoke({
        "task": task,
        "has_research": has_research,
        "has_analysis": has_analysis,
        "has_report": has_report
    })

    # Parse decision
    decision_text = decision.content.strip().lower()
```

```
    print(decision_text)

    # Determine next agent
    if "done" in decision_text or has_report:
        next_agent = "end"
        supervisor_msg = "✅ Supervisor: All tasks complete! Great work team."
    elif "researcher" in decision_text or not has_research:
        next_agent = "researcher"
        supervisor_msg = "📋 Supervisor: Let's start with research. Assigning to
Researcher..."
    elif "analyst" in decision_text or (has_research and not has_analysis):
        next_agent = "analyst"
        supervisor_msg = "📋 Supervisor: Research done. Time for analysis. Assi
gning to Analyst..."
    elif "writer" in decision_text or (has_analysis and not has_report):
        next_agent = "writer"
        supervisor_msg = "📋 Supervisor: Analysis complete. Let's create the re
port. Assigning to Writer..."
    else:
        next_agent = "end"
        supervisor_msg = "✅ Supervisor: Task seems complete."

    return {
        "messages": [AIMessage(content=supervisor_msg)],
        "next_agent": next_agent,
        "current_task": task
    }
```

D. Researcher Agent

```
# ====================================
# Agent 1: Researcher (using Groq)
# ====================================

def researcher_agent(state: SupervisorState) → Dict:
```

```
    """Researcher uses Groq to gather information"""

    task = state.get("current_task", "research topic")

    # Create research prompt
    research_prompt = f"""As a research specialist, provide comprehensive info
rmation about: {task}

    Include:
    1. Key facts and background
    2. Current trends or developments
    3. Important statistics or data points
    4. Notable examples or case studies

    Be concise but thorough."""

    # Get research from LLM
    research_response = llm.invoke([HumanMessage(content=research_promp
t)])
    research_data = research_response.content

    # Create agent message
    agent_message = f"🔍 Researcher: I've completed the research on '{tas
k}'.\n\nKey findings:\n{research_data[:500]}..."

    return {
        "messages": [AIMessage(content=agent_message)],
        "research_data": research_data,
        "next_agent": "supervisor"
    }
```

- Uses LLM to gather factual information on the topic.
- Stores the findings in `research_data`.

E. Analyst Agent

```
# ====================================
# Agent 2: Analyst (using Groq)
# ====================================

def analyst_agent(state: SupervisorState) → Dict:
    """Analyst uses Groq to analyze the research"""

    research_data = state.get("research_data", "")
    task = state.get("current_task", "")

    # Create analysis prompt
    analysis_prompt = f"""As a data analyst, analyze this research data and pro
vide insights:

Research Data:
{research_data}

Provide:
1. Key insights and patterns
2. Strategic implications
3. Risks and opportunities
4. Recommendations

Focus on actionable insights related to: {task}"""

    # Get analysis from LLM
    analysis_response = llm.invoke([HumanMessage(content=analysis_promp
t)])
    analysis = analysis_response.content

    # Create agent message
    agent_message = f"📊 Analyst: I've completed the analysis.\n\nTop insight
s:\n{analysis[:400]}..."

    return {
```

```
        "messages": [AIMessage(content=agent_message)],
        "analysis": analysis,
        "next_agent": "supervisor"
    }
```

- Analyzes `research_data` and generates insights.
- Results go into the `analysis` field.

F. Writer Agent

```
# ====================================
# Agent 3: Writer (using Groq)
# ====================================

def writer_agent(state: SupervisorState) → Dict:
    """Writer uses Groq to create final report"""

    research_data = state.get("research_data", "")
    analysis = state.get("analysis", "")
    task = state.get("current_task", "")

    # Create writing prompt
    writing_prompt = f"""As a professional writer, create an executive report ba
sed on:

Task: {task}

Research Findings:
{research_data[:1000]}

Analysis:
{analysis[:1000]}

Create a well-structured report with:
1. Executive Summary
```

```
2. Key Findings
3. Analysis & Insights
4. Recommendations
5. Conclusion

Keep it professional and concise."""

    # Get report from LLM
    report_response = llm.invoke([HumanMessage(content=writing_prompt)])
    report = report_response.content

    # Create final formatted report
    final_report = f"""
📄 FINAL REPORT
{'='*50}
Generated: {datetime.now().strftime('%Y-%m-%d %H:%M')}
Topic: {task}
{'='*50}

{report}

{'='*50}
Report compiled by Multi-Agent AI System powered by Groq
"""

    return {
        "messages": [AIMessage(content=f"✍️ Writer: Report complete! See bel
ow for the full document.")],
        "final_report": final_report,
        "next_agent": "supervisor",
        "task_complete": True
    }
```

- Uses both `research_data` and `analysis` to compose a professional report.
- Final output is stored in `final_report`.

G. Routing Function

```
# ====================================
# Router Function
# ====================================

def router(state: SupervisorState) → Literal["supervisor", "researcher", "analy
st", "writer", "__end__"]:
    """Routes to next agent based on state"""

    next_agent = state.get("next_agent", "supervisor")

    if next_agent == "end" or state.get("task_complete", False):
        return END

    if next_agent in ["supervisor", "researcher", "analyst", "writer"]:
        return next_agent

    return "supervisor"
```

- Determines which agent runs next or if the workflow should end.

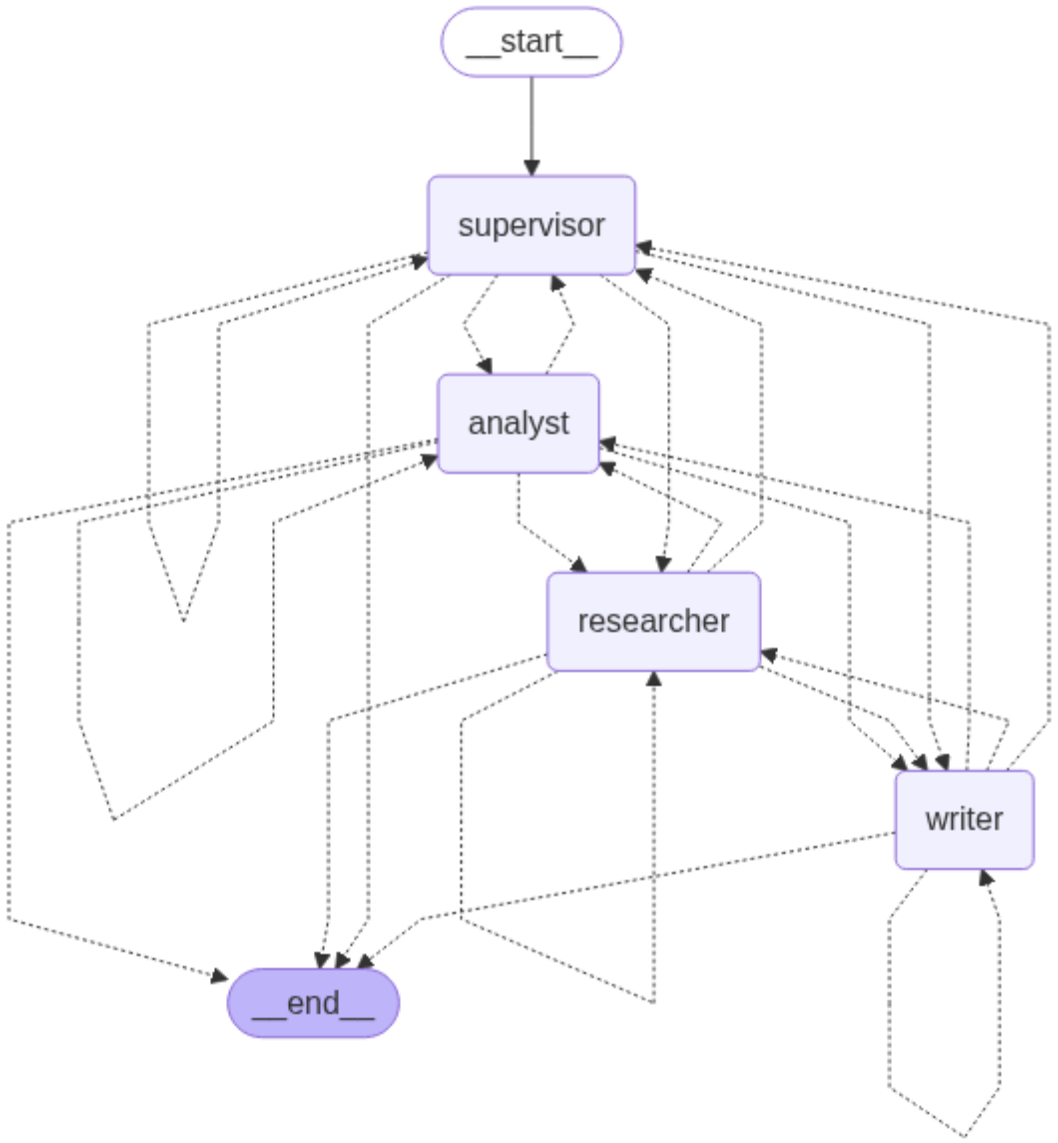
7.3.3 Workflow Graph Construction

```
# Create workflow
workflow = StateGraph(SupervisorState)

# Add nodes
workflow.add_node("supervisor", supervisor_agent)
workflow.add_node("researcher", researcher_agent)
workflow.add_node("analyst", analyst_agent)
workflow.add_node("writer", writer_agent)

# Set entry point
workflow.set_entry_point("supervisor")
```

```
# Add routing
for node in ["supervisor", "researcher", "analyst", "writer"]:
    workflow.add_conditional_edges(
        node,
        router,
        {
            "supervisor": "supervisor",
            "researcher": "researcher",
            "analyst": "analyst",
            "writer": "writer",
            END: END
        }
    )

graph=workflow.compile()
```

- Constructs the full LangGraph using all the agents and the supervisor.

Output:



7.3.4 Run Example

```
response=graph.invoke(HumanMessage(content="What are the benefits and
risks of AI in healthcare?"))
response['final_report']
```

> You'll get a structured executive report like:

"\n📄 FINAL REPORT\n==========\nGenerated: 2025-07-15 19:01\nTopic: 📊 Analyst: I've completed the analysis.\n\nTop insights:\n**Task: Investigating the Effects of Climate Change on Global Ecosystems**\n\n**Key Insights and Patterns:**\n\n1. **Accelerating Global Warming**: The rate of global warming is accelerating, with the past four years being the hottest on record.\n2. **Increased Extreme Weather Events**: Climate change is linked to more frequent and intense heatwaves, droughts, and heavy rainfall events.\n3. **Rising Sea...\n==========\n\n**Executive Report: Investigating the Effects of Climate Change on Global Ecosystems**\n\n**Date:** July 15, 2024\n**Prepared by:** [Your Name]\n**Analyst:** [Your Name]\n\n**Executive Summary:**\n\nThis report presents the findings of an in-depth analysis on the effects of climate change on global ecosystems. Our research reveals alarming trends, including accelerating global warming, increased extreme weather events, and rising sea levels. These changes are having far-reaching consequences for ecosystems, species, and human societies. In light of these findings, we provide strategic implications and recommendations for policymakers, businesses, and individuals to mitigate the impacts of climate change.\n\n**Key Findings:**\n\n1. **Accelerating Global Warming**: The rate of global warming is accelerating, with the past four years being the hottest on record.\n2. **Increased Extreme Weather Events**: Climate change is linked to more frequent and intense heatwaves, droughts, and heavy rainfall events.\n3. **Rising Sea Levels**: Oceans have risen by about 15-20 cm since 1900, and the rate of rise is accelerating.\n4. **Tipping Points**: Some ecosystems, like coral reefs and Arctic ice, are approaching tipping points, where irreversible damage may occur.\n5. **Global Emissions**: Carbon dioxide emissions from human activities are projected to increase by 13% by 2050 if current trends continue.\n6. **Ecosystem Loss**: Up to 1 million species are at risk of extinction due to human activities, including climate change.\n\n**Analysis & Insights:**\n\nOur analysis reveals that climate change is having a profound impact on global ecosystems. The accelerating rate of global warming, increasing extreme weather events, and rising sea levels are all interconnected and exacerbating one another. The consequences of climate change are far-reaching, with ecosystems, species, and human societies all being affected.\n\n**Strategic Implications:**\n\n1. **Transition to Renewable Energy**: A rapid shift to renewa

ble energy sources, such as solar and wind power, is essential to reduce gree
nhouse gas emissions and mitigate climate change.\n2. **Carbon Pricing**: I
mplementing a global carbon pricing mechanism can provide a financial incen
tive for reducing emissions and investing in low-carbon technologies.\n3. **Cl
imate Resilience**: Investing in climate resilience and adaptation measures, s
uch as sea walls and green infrastructure, can help communities and ecosyst
ems adapt to the impacts of climate change.\n4. **Biodiversity Conservation*
*: Protecting and restoring natural ecosystems, such as forests and wetlands,
can help maintain biodiversity and ecosystem services.\n\n**Recommendatio
ns:**\n\n1. **Develop and implement a global climate strategy**: Government
s, businesses, and civil society organizations must work together to develop a
nd implement a comprehensive climate strategy.\n2. **Increase investment in
renewable energy**: Governments and businesses must increase investment i
n renewable energy sources to reduce greenhouse gas emissions.\n3. **Impl
ement climate-resilient infrastructure**: Communities and governments must i
nvest in climate-resilient infrastructure, such as sea walls and green roofs.\n4.
Support biodiversity conservation: Governments, businesses, and civil so
ciety organizations must support biodiversity conservation efforts to protect a
nd restore natural ecosystems.\n\n**Conclusion:**\n\nClimate change is one
of the most pressing issues of our time, with far-reaching consequences for e
cosystems, species, and human societies. Our analysis reveals that urgent act
ion is needed to mitigate the impacts of climate change. We recommend a co
mprehensive climate strategy, increased investment in renewable energy, clim
ate-resilient infrastructure, and biodiversity conservation efforts.\n\n=======
==========\nReport compile
d by Multi-Agent AI System powered by Groq\n"

7.4 Hierarchical Multi-Agent Architecture

A **Hierarchical Multi-Agent System (MAS)** is designed by arranging agents in a top-down structure, where higher-level agents (like a CEO ogr supervisor) manage or delegate work to specialized lower-level agents (such as researchers or writers). This layered delegation promotes **clarity, scalability, and modular**

task handling, enabling large, complex workflows to be broken into manageable components.

In this section, we demonstrate two implementation patterns:

- **7.4.1: Task Delegation via Parent-Child Nodes**

  Simulates a corporate hierarchy using `LangGraph`, where each agent passes control to the next in a sequential but conditional flow. This pattern is linear, predictable, and ideal for simpler hierarchies.

- **7.4.2: Nested Graphs with Conditional Execution**

  Uses more explicit conditional logic to orchestrate transitions between agents. Agents evaluate task requirements and dynamically decide what action to take next. This version is more flexible and robust for complex decision-making.

7.4.1 Hierarchical Multi-Agent Architecture (Approach 1)

Overview

This architecture simulates a **corporate hierarchy**, where each agent performs a specialized role. Tasks flow from a CEO to research leads, then to domain researchers, and finally to writers. The flow is driven by conditional state transitions using `LangGraph`.

---

1. Imports and Initialization

```
# Core LangGraph and LangChain
from langgraph.graph import StateGraph, END
from langchain_core.messages import HumanMessage
from langchain_groq import ChatGroq
from pydantic import BaseModel
from typing import Dict
from dotenv import load_dotenv
import os

# Load .env and initialize API
```

```
load_dotenv()
os.environ["GROQ_API_KEY"] = os.getenv("GROQ_API_KEY")

# LLM initialization
llm = ChatGroq(temperature=0.7, model_name="llama-3.1-8b-instant")
```

---

2. State Definition

```
class SupervisorState(BaseModel):
    current_agent: str
    task: str
    task_assignment: Dict[str, str] = {}
    research_data: str = ""
    analysis: str = ""
    report: str = ""
    summary: str = ""
    research_complete: bool = False
```

---

3. Agent Definitions

CEO Agent: Breaks down the task

```
def ceo_agent(state: SupervisorState) → SupervisorState:
    response = llm.invoke([
        HumanMessage(content=f"Break down this task: {state.task}")
    ])
    state.task_assignment["Research Team Leader"] = response.content
    state.current_agent = "research_team_leader"
    return state
```

Research Team Leader: Delegates subtasks

```
def research_team_leader_agent(state: SupervisorState) → SupervisorState:
    if state.research_complete:
        state.current_agent = "writing_team_leader"
        return state

    task = state.task_assignment["Research Team Leader"]
    response = llm.invoke([
        HumanMessage(content=f"Divide research: {task}")
    ])

    state.task_assignment["Data Researcher"] = f"Data task: {response.conten
t}"
    state.task_assignment["Market Researcher"] = f"Market task: {response.co
ntent}"
    state.current_agent = "data_researcher"
    return state
```

Data Researcher: Adds data findings

```
def data_researcher_agent(state: SupervisorState) → SupervisorState:
    response = llm.invoke([
        HumanMessage(content=state.task_assignment["Data Researcher"])
    ])
    state.research_data += f"\nData Findings:\n{response.content}"

    if "Market Findings" in state.research_data:
        state.research_complete = True

    state.current_agent = "market_researcher"
    return state
```

Market Researcher: Adds market findings

```
def market_researcher_agent(state: SupervisorState) → SupervisorState:
    response = llm.invoke([
        HumanMessage(content=state.task_assignment["Market Researcher"])
    ])
    state.research_data += f"\nMarket Findings:\n{response.content}"
    state.current_agent = "writing_team_leader"
    return state
```

Writing Team Leader: Decides report type

```
def writing_team_leader_agent(state: SupervisorState) → SupervisorState:
    if not state.research_data:
        state.current_agent = END
        return state

    if "report" in state.task.lower():
        state.current_agent = "technical_writer"
    else:
        state.current_agent = "summary_writer"
    return state
```

Technical Writer: Generates report

```
def technical_writer_agent(state: SupervisorState) → SupervisorState:
    response = llm.invoke([
        HumanMessage(content=f"Write report about: {state.task}")
    ])
    state.report = response.content
    state.current_agent = END
    return state
```

Summary Writer: Generates summary

```
def summary_writer_agent(state: SupervisorState) → SupervisorState:
    response = llm.invoke([
        HumanMessage(content=f"Summarize: {state.task}")
    ])
    state.summary = response.content
    state.current_agent = END
    return state
```

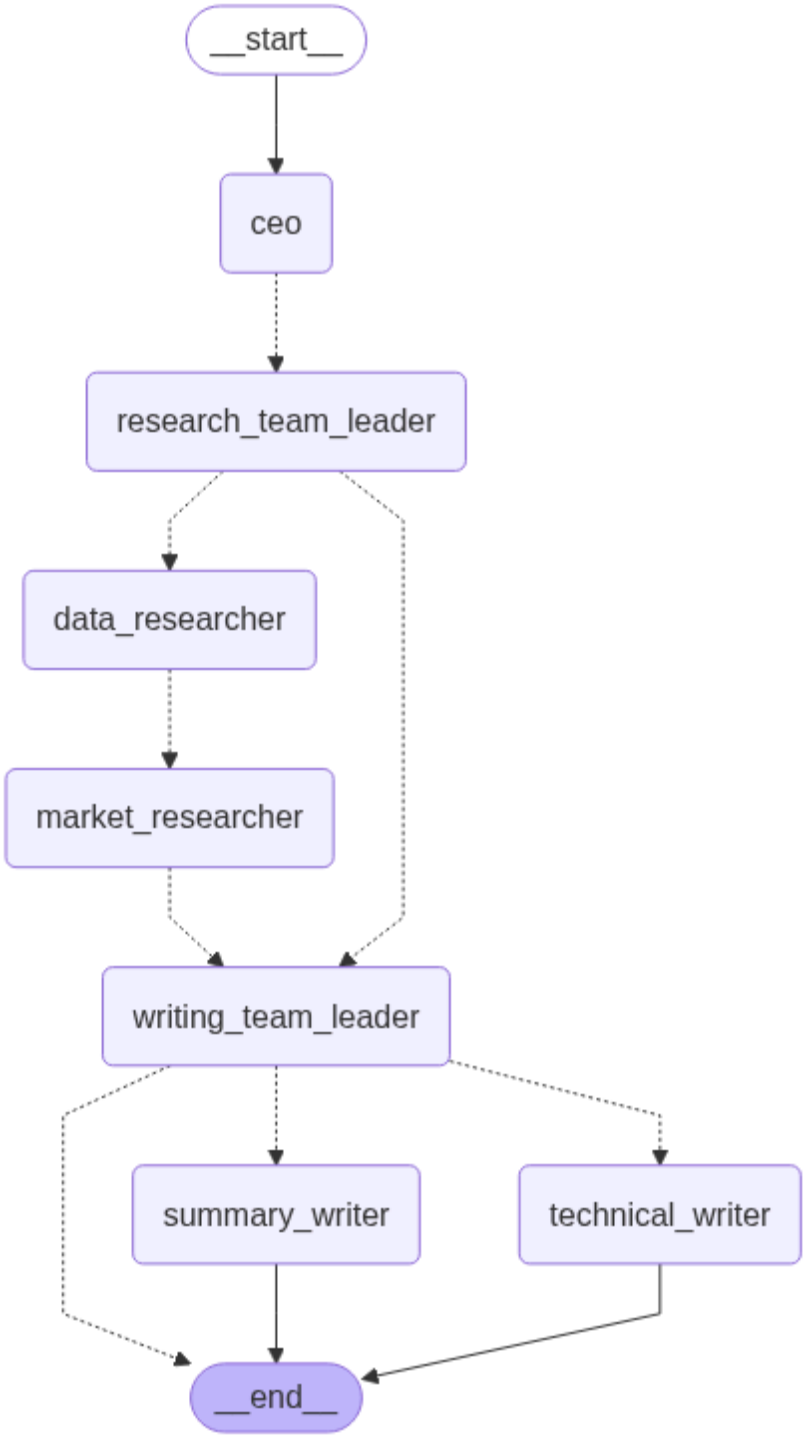
---

4. Graph Construction

```
workflow = StateGraph(SupervisorState)

workflow.add_node("ceo", ceo_agent)
workflow.add_node("research_team_leader", research_team_leader_agent)
workflow.add_node("data_researcher", data_researcher_agent)
workflow.add_node("market_researcher", market_researcher_agent)
workflow.add_node("writing_team_leader", writing_team_leader_agent)
workflow.add_node("technical_writer", technical_writer_agent)
workflow.add_node("summary_writer", summary_writer_agent)

workflow.set_entry_point("ceo")

workflow.add_conditional_edges(
    "ceo",
    lambda state: state.current_agent,
    {"research_team_leader": "research_team_leader"}
)
workflow.add_conditional_edges(
    "research_team_leader",
    lambda state: state.current_agent,
    {"data_researcher": "data_researcher", "writing_team_leader": "writing_tea
m_leader"}
)
workflow.add_conditional_edges(
```

```
    "data_researcher",
    lambda state: state.current_agent,
    {"market_researcher": "market_researcher"}
)
workflow.add_conditional_edges(
    "market_researcher",
    lambda state: state.current_agent,
    {"writing_team_leader": "writing_team_leader"}
)
workflow.add_conditional_edges(
    "writing_team_leader",
    lambda state: state.current_agent,
    {
        "technical_writer": "technical_writer",
        "summary_writer": "summary_writer",
        END: END
    }
)
workflow.add_edge("technical_writer", END)
workflow.add_edge("summary_writer", END)

app = workflow.compile()
```

Output:


__start__
ceo
research_team_leader
data_researcher
market_researcher
writing_team_leader
summary_writer
technical_writer
__end__

5. Execution

```
response = app.invoke({
    "current_agent": "ceo",
    "task": "Analyze AI benefits and risks in healthcare and create report",
    "research_complete": False
})

print("Final Output:", response)
```

Output

```
Final Output: {'current_agent': '__end__', 'task': 'Analyze AI benefits and risks i
n healthcare and create report', 'task_assignment': {'Research Team Leader':
"**Task Breakdown: Analyze AI Benefits and Risks in Healthcare and Create R
eport**\n\n**Objective:** To conduct an in-depth analysis of the benefits and
risks of Artificial Intelligence (AI) in the healthcare sector and create a compre
hensive report.\n\n**Step 1: Research and Literature Review (Timeframe: 2-3
days)**\n\n1. Identify relevant studies, articles, and reports on AI applications i
n healthcare.\n2. Review existing literature on AI benefits and risks in healthca
re, including:\n\t* Improved diagnosis accuracy\n\t* Enhanced patient outcom
es\n\t* Streamlined clinical workflows\n\t* Data security concerns\n\t* Bias an
d fairness issues\n\t* Medical liability and regulatory challenges\n3. Note dow
n key findings, insights, and gaps in existing research.\n\n**Step 2: Benefits o
f AI in Healthcare (Timeframe: 2-3 days)**\n\n1. Categorize AI benefits in heal
thcare into:\n\t* Diagnostic accuracy and speed\n\t* Personalized medicine an
d treatment\n\t* Patient engagement and adherence\n\t* Clinical decision sup
port and workflow optimization\n\t* Administrative efficiency and cost reducti
on\n2. Gather examples and case studies of successful AI implementations in
healthcare, including:\n\t* Predictive analytics for disease detection\n\t* Imag
e analysis for cancer diagnosis\n\t* Chatbots for patient engagement and sup
port\n\t* AI-powered clinical decision support systems\n3. Document the ben
efits of AI in healthcare, including improved patient outcomes, increased effici
ency, and enhanced patient experience.\n\n**Step 3: Risks and Challenges of
AI in Healthcare (Timeframe: 2-3 days)**\n\n1. Identify and categorize AI risks
and challenges in healthcare into:\n\t* Data quality and security concerns\n\t*
```

Bias and fairness issues\n\t* Medical liability and regulatory challenges\n\t* D
ependence on complex algorithms and systems\n\t* Limited transparency and
explainability\n2. Gather examples and case studies of AI failures or limitation
s in healthcare, including:\n\t* Misdiagnosis or delayed diagnosis due to AI err
ors\n\t* Bias in AI-driven decision-making\n\t* Regulatory compliance and liab
ility issues\n\t* Technical limitations and maintenance challenges\n3. Docume
nt the risks and challenges of AI in healthcare, including potential patient har
m, financial losses, and reputational damage.\n\n**Step 4: Report Writing and
Analysis (Timeframe: 4-5 days)**\n\n1. Compile the research, literature revie
w, and analysis into a comprehensive report.\n2. Organize the report into clea
r sections and subsections, including:\n\t* Executive summary\n\t* Introductio
n to AI in healthcare\n\t* Benefits of AI in healthcare\n\t* Risks and challenges
of AI in healthcare\n\t* Conclusion and recommendations\n3. Develop a clear
and concise writing style, using visual aids and tables to illustrate key findings
and statistics.\n4. Edit and proofread the report for accuracy, clarity, and cons
istency.\n\n**Step 5: Finalize and Deliver the Report (Timeframe: 1-2 days)**
\n\n1. Finalize the report, addressing any remaining gaps or concerns.\n2. Ens
ure the report meets the requirements and expectations of the stakeholders.\n
3. Deliver the report in the agreed-upon format (e.g., PDF, Word document, Po
werPoint presentation).\n4. Provide a summary or brief overview of the repor
t's key findings and recommendations.", 'Data Researcher': "Data task: **Task
Breakdown: Analyze AI Benefits and Risks in Healthcare and Create Report**
\n\n**Objective:** To conduct an in-depth analysis of the benefits and risks of
Artificial Intelligence (AI) in the healthcare sector and create a comprehensive
report.\n\n**Step 1: Research and Literature Review (Timeframe: 2-3 days)**
\n\n1. **Identify relevant studies, articles, and reports on AI applications in hea
lthcare:**\n\t* PubMed (scholarly articles)\n\t* Google Scholar (academic arti
cles and research papers)\n\t* ScienceDirect (peer-reviewed articles and jour
nals)\n\t* ResearchGate (research papers and articles)\n\t* AI in Healthcare re
ports from organizations like HIMSS, HIMSA, and AI in Healthcare Association
\n2. **Review existing literature on AI benefits and risks in healthcare:**\n\t* I
mproved diagnosis accuracy\n\t* Enhanced patient outcomes\n\t* Streamline
d clinical workflows\n\t* Data security concerns\n\t* Bias and fairness issues
\n\t* Medical liability and regulatory challenges\n3. **Note down key findings,
insights, and gaps in existing research:**\n\t- AI applications in healthcare are
increasing, with a focus on improving diagnosis accuracy and patient outcom

es.\n\t- AI has the potential to streamline clinical workflows and improve admi
nistrative efficiency.\n\t- However, AI in healthcare is not without risks, includi
ng data security concerns, bias and fairness issues, and medical liability and r
egulatory challenges.\n\n**Step 2: Benefits of AI in Healthcare (Timeframe: 2-
3 days)**\n\n1. **Categorize AI benefits in healthcare into:**\n\t* Diagnostic a
ccuracy and speed\n\t* Personalized medicine and treatment\n\t* Patient eng
agement and adherence\n\t* Clinical decision support and workflow optimizati
on\n\t* Administrative efficiency and cost reduction\n2. **Gather examples an
d case studies of successful AI implementations in healthcare:**\n\t+ Predicti
ve analytics for disease detection (e.g., cancer, diabetes)\n\t+ Image analysis
for cancer diagnosis (e.g., mammography, CT scans)\n\t+ Chatbots for patient
engagement and support (e.g., patient education, medication adherence)\n\t+
AI-powered clinical decision support systems (e.g., IBM Watson, Google Healt
h)\n3. **Document the benefits of AI in healthcare:**\n\t- Improved patient ou
tcomes and increased accuracy in diagnosis and treatment.\n\t- Enhanced pat
ient engagement and adherence to treatment plans.\n\t- Streamlined clinical
workflows and improved administrative efficiency.\n\n**Step 3: Risks and Cha
llenges of AI in Healthcare (Timeframe: 2-3 days)**\n\n1. **Identify and categ
orize AI risks and challenges in healthcare into:**\n\t* Data quality and securit
y concerns\n\t* Bias and fairness issues\n\t* Medical liability and regulatory c
hallenges\n\t* Dependence on complex algorithms and systems\n\t* Limited tr
ansparency and explainability\n2. **Gather examples and case studies of AI f
ailures or limitations in healthcare:**\n\t+ Misdiagnosis or delayed diagnosis d
ue to AI errors (e.g., AI misinterpreting medical images)\n\t+ Bias in AI-driven
decision-making (e.g., AI perpetuating existing health disparities)\n\t+ Regulat
ory compliance and liability issues (e.g., AI-driven medical device approvals)
\n\t+ Technical limitations and maintenance challenges (e.g., AI system downti
me or data loss)\n3. **Document the risks and challenges of AI in healthcare:
**\n\t- Potential patient harm or financial losses due to AI errors or bias.\n\t- R
egulatory compliance and liability issues related to AI-driven medical devices
or decisions.\n\t- Technical limitations and maintenance challenges affecting
AI system reliability and effectiveness.\n\n**Step 4: Report Writing and Analys
is (Timeframe: 4-5 days)**\n\n1. **Compile the research, literature review, and
analysis into a comprehensive report:**\n\t+ Include an executive summary, in
troduction, and conclusion.\n\t+ Organize the report into clear sections and su
bsections.\n2. **Develop a clear and concise writing style, using visual aids a

nd tables to illustrate key findings and statistics:**\n\t+ Use charts, graphs, an
d infographics to visualize key data and trends.\n\t+ Include tables and figures
to illustrate the benefits and risks of AI in healthcare.\n3. **Edit and proofread
the report for accuracy, clarity, and consistency:**\n\t+ Review the report for
grammar, spelling, and punctuation errors.\n\t+ Ensure the report is well-orga
nized and easy to follow.\n\n**Step 5: Finalize and Deliver the Report (Timefra
me: 1-2 days)**\n\n1. **Finalize the report, addressing any remaining gaps or
concerns:**\n\t+ Ensure the report meets the requirements and expectations
of the stakeholders.\n\t+ Address any remaining questions or concerns relate
d to the report's findings and recommendations.\n2. **Deliver the report in th
e agreed-upon format (e.g., PDF, Word document, PowerPoint presentation):*
*\n\t+ Ensure the report is well-formatted and easy to read.\n\t+ Include a sum
mary or brief overview of the report's key findings and recommendations.", 'M
arket Researcher': "Market task: **Task Breakdown: Analyze AI Benefits and R
isks in Healthcare and Create Report**\n\n**Objective:** To conduct an in-de
pth analysis of the benefits and risks of Artificial Intelligence (AI) in the health
care sector and create a comprehensive report.\n\n**Step 1: Research and Lit
erature Review (Timeframe: 2-3 days)**\n\n1. **Identify relevant studies, articl
es, and reports on AI applications in healthcare:**\n\t* PubMed (scholarly arti
cles)\n\t* Google Scholar (academic articles and research papers)\n\t* Scienc
eDirect (peer-reviewed articles and journals)\n\t* ResearchGate (research pap
ers and articles)\n\t* AI in Healthcare reports from organizations like HIMSS, H
IMSA, and AI in Healthcare Association\n2. **Review existing literature on AI b
enefits and risks in healthcare:**\n\t* Improved diagnosis accuracy\n\t* Enhan
ced patient outcomes\n\t* Streamlined clinical workflows\n\t* Data security co
ncerns\n\t* Bias and fairness issues\n\t* Medical liability and regulatory challe
nges\n3. **Note down key findings, insights, and gaps in existing research:**
\n\t- AI applications in healthcare are increasing, with a focus on improving di
agnosis accuracy and patient outcomes.\n\t- AI has the potential to streamline
clinical workflows and improve administrative efficiency.\n\t- However, AI in h
ealthcare is not without risks, including data security concerns, bias and fairn
ess issues, and medical liability and regulatory challenges.\n\n**Step 2: Benef
its of AI in Healthcare (Timeframe: 2-3 days)**\n\n1. **Categorize AI benefits i
n healthcare into:**\n\t* Diagnostic accuracy and speed\n\t* Personalized me
dicine and treatment\n\t* Patient engagement and adherence\n\t* Clinical deci
sion support and workflow optimization\n\t* Administrative efficiency and cost

reduction\n2. **Gather examples and case studies of successful AI implement
ations in healthcare:**\n\t+ Predictive analytics for disease detection (e.g., ca
ncer, diabetes)\n\t+ Image analysis for cancer diagnosis (e.g., mammography,
CT scans)\n\t+ Chatbots for patient engagement and support (e.g., patient ed
ucation, medication adherence)\n\t+ AI-powered clinical decision support syst
ems (e.g., IBM Watson, Google Health)\n3. **Document the benefits of AI in h
ealthcare:**\n\t- Improved patient outcomes and increased accuracy in diagn
osis and treatment.\n\t- Enhanced patient engagement and adherence to treat
ment plans.\n\t- Streamlined clinical workflows and improved administrative ef
ficiency.\n\n**Step 3: Risks and Challenges of AI in Healthcare (Timeframe: 2-
3 days)**\n\n1. **Identify and categorize AI risks and challenges in healthcare
into:**\n\t* Data quality and security concerns\n\t* Bias and fairness issues\n
\t* Medical liability and regulatory challenges\n\t* Dependence on complex al
gorithms and systems\n\t* Limited transparency and explainability\n2. **Gath
er examples and case studies of AI failures or limitations in healthcare:**\n\t+
Misdiagnosis or delayed diagnosis due to AI errors (e.g., AI misinterpreting me
dical images)\n\t+ Bias in AI-driven decision-making (e.g., AI perpetuating exi
sting health disparities)\n\t+ Regulatory compliance and liability issues (e.g., A
I-driven medical device approvals)\n\t+ Technical limitations and maintenance
challenges (e.g., AI system downtime or data loss)\n3. **Document the risks a
nd challenges of AI in healthcare:**\n\t- Potential patient harm or financial los
ses due to AI errors or bias.\n\t- Regulatory compliance and liability issues rel
ated to AI-driven medical devices or decisions.\n\t- Technical limitations and
maintenance challenges affecting AI system reliability and effectiveness.\n\n*
*Step 4: Report Writing and Analysis (Timeframe: 4-5 days)**\n\n1. **Compile
the research, literature review, and analysis into a comprehensive report:**\n\t
+ Include an executive summary, introduction, and conclusion.\n\t+ Organize t
he report into clear sections and subsections.\n2. **Develop a clear and conci
se writing style, using visual aids and tables to illustrate key findings and stati
stics:**\n\t+ Use charts, graphs, and infographics to visualize key data and tre
nds.\n\t+ Include tables and figures to illustrate the benefits and risks of AI in
healthcare.\n3. **Edit and proofread the report for accuracy, clarity, and consi
stency:**\n\t+ Review the report for grammar, spelling, and punctuation error
s.\n\t+ Ensure the report is well-organized and easy to follow.\n\n**Step 5: Fin
alize and Deliver the Report (Timeframe: 1-2 days)**\n\n1. **Finalize the repor
t, addressing any remaining gaps or concerns:**\n\t+ Ensure the report meets

the requirements and expectations of the stakeholders.\n\t+ Address any rem
aining questions or concerns related to the report's findings and recommenda
tions.\n2. **Deliver the report in the agreed-upon format (e.g., PDF, Word doc
ument, PowerPoint presentation):**\n\t+ Ensure the report is well-formatted a
nd easy to read.\n\t+ Include a summary or brief overview of the report's key f
indings and recommendations."}, 'research_data': '\nData Findings:\n**Task B
reakdown: Analyze AI Benefits and Risks in Healthcare and Create Report**\n
\n**Objective:** To conduct an in-depth analysis of the benefits and risks of
Artificial Intelligence (AI) in the healthcare sector and create a comprehensive
report.\n\n**Step 1: Research and Literature Review**\n\nIn this step, we will i
dentify relevant studies, articles, and reports on AI applications in healthcare,
review existing literature on AI benefits and risks in healthcare, and note down
key findings, insights, and gaps in existing research.\n\n**Identified Research
Sources:**\n\n1. **PubMed (scholarly articles):**\n\t* "Artificial Intelligence in
Medicine: A Review" (2020)\n\t* "Machine Learning in Healthcare: A Systemat
ic Review" (2019)\n2. **Google Scholar (academic articles and research pape
rs):**\n\t* "The Impact of Artificial Intelligence on Healthcare" (2020)\n\t* "Art
ificial Intelligence in Healthcare: Opportunities and Challenges" (2019)\n3. **S
cienceDirect (peer-reviewed articles and journals):**\n\t* "Artificial Intelligenc
e in Healthcare: A Review of the Literature" (2020)\n\t* "Machine Learning in
Healthcare: A Survey" (2019)\n4. **ResearchGate (research papers and articl
es):**\n\t* "Artificial Intelligence in Medicine: A Review of the Current State of
the Art" (2020)\n\t* "The Role of Artificial Intelligence in Healthcare" (2019)\n
5. **AI in Healthcare reports from organizations like HIMSS, HIMSA, and AI in
Healthcare Association:**\n\t* "2019 AI in Healthcare Report" (HIMSS)\n\t* "A
rtificial Intelligence in Healthcare: A Study of the Current State of the Art" (HI
MSA)\n\n**Key Findings and Insights:**\n\n1. **AI applications in healthcare a
re increasing, with a focus on improving diagnosis accuracy and patient outco
mes.**\n2. **AI has the potential to streamline clinical workflows and improve
administrative efficiency.**\n3. **However, AI in healthcare is not without risk
s, including data security concerns, bias and fairness issues, and medical liabi
lity and regulatory challenges.**\n\n**Step 2: Benefits of AI in Healthcare**\n
\nIn this step, we will categorize AI benefits in healthcare, gather examples an
d case studies of successful AI implementations in healthcare, and document
the benefits of AI in healthcare.\n\n**Categorized AI Benefits:**\n\n1. **Diagn
ostic accuracy and speed**\n2. **Personalized medicine and treatment**\n3.

Patient engagement and adherence\n4. **Clinical decision support and w
orkflow optimization**\n5. **Administrative efficiency and cost reduction**\n
\n**Examples and Case Studies:**\n\n1. **Predictive analytics for disease det
ection (e.g., cancer, diabetes)**\n\t* Example: IBM Watson\'s AI-powered can
cer diagnosis system\n\t* Case Study: A study on the use of AI-powered predi
ctive analytics for detecting breast cancer\n2. **Image analysis for cancer dia
gnosis (e.g., mammography, CT scans)**\n\t* Example: Google\'s AI-powered
image analysis system for detecting breast cancer\n\t* Case Study: A study o
n the use of AI-powered image analysis for detecting lung cancer\n3. **Chatb
ots for patient engagement and support (e.g., patient education, medication a
dherence)**\n\t* Example: Microsoft\'s AI-powered chatbot for patient engage
ment and support\n\t* Case Study: A study on the use of AI-powered chatbot
s for improving patient engagement and adherence\n4. **AI-powered clinical
decision support systems (e.g., IBM Watson, Google Health)**\n\t* Example: I
BM Watson\'s AI-powered clinical decision support system for improving patie
nt outcomes\n\t* Case Study: A study on the use of AI-powered clinical decisi
on support systems for improving patient outcomes\n\n**Documented Benefit
s:**\n\n1. **Improved patient outcomes and increased accuracy in diagnosis a
nd treatment.**\n2. **Enhanced patient engagement and adherence to treatm
ent plans.**\n3. **Streamlined clinical workflows and improved administrative
efficiency.**\n\n**Step 3: Risks and Challenges of AI in Healthcare**\n\nIn thi
s step, we will identify and categorize AI risks and challenges in healthcare, g
ather examples and case studies of AI failures or limitations in healthcare, and
document the risks and challenges of AI in healthcare.\n\n**Categorized AI Ri
sks and Challenges:**\n\n1. **Data quality and security concerns**\n2. **Bias
and fairness issues**\n3. **Medical liability and regulatory challenges**\n4. *
*Dependence on complex algorithms and systems**\n5. **Limited transparen
cy and explainability**\n\n**Examples and Case Studies:**\n\n1. **Misdiagno
sis or delayed diagnosis due to AI errors (e.g., AI misinterpreting medical imag
es)**\n\t* Example: A study on the misdiagnosis of breast cancer due to AI err
ors\n\t* Case Study: A case study on the delayed diagnosis of a patient due to
AI errors\n2. **Bias in AI-driven decision-making (e.g., AI perpetuating existin
g health disparities)**\n\t* Example: A study on the bias in AI-powered predict
ive analytics for disease detection\n\t* Case Study: A case study on the use o
f AI-powered image analysis for detecting cancer in patients with different ski
n tones\n3. **Regulatory compliance and liability issues (e.g., AI-driven medic

al device approvals)**\n\t* Example: A study on the regulatory compliance an
d liability issues related to AI-powered medical devices\n\t* Case Study: A cas
e study on the approval of an AI-powered medical device for breast cancer di
agnosis\n4. **Technical limitations and maintenance challenges (e.g., AI syste
m downtime or data loss)**\n\t* Example: A study on the technical limitations
and maintenance challenges related to AI-powered clinical decision support s
ystems\n\t* Case Study: A case study on the downtime of an AI-powered syst
em for predicting patient outcomes\n\n**Documented Risks and Challenges:*
*\n\n1. **Potential patient harm or financial losses due to AI errors or bias.**\n
2. **Regulatory compliance and liability issues related to AI-driven medical de
vices or decisions.**\n3. **Technical limitations and maintenance challenges
affecting AI system reliability and effectiveness.**\n\n**Step 4: Report Writing
and Analysis**\n\nIn this step, we will compile the research, literature review,
and analysis into a comprehensive report, develop a clear and concise writing
style, and edit and proofread the report for accuracy, clarity, and consistenc
y.\n\n**Report Structure:**\n\n1. **Executive Summary:** A brief summary of
the report\'s key findings and recommendations.\n2. **Introduction:** An intro
duction to the report, including the objective and scope of the study.\n3. **Lit
erature Review:** A review of the existing literature on AI applications in healt
hcare.\n4. **Benefits of AI in Healthcare:** A section on the benefits of AI in h
ealthcare, including diagnostic accuracy and speed, personalized medicine an
d treatment, patient engagement and adherence, clinical decision support and
workflow optimization, and administrative efficiency and cost reduction.\n5. *
*Risks and Challenges of AI in Healthcare:** A section on the risks and challe
nges of AI in healthcare, including data quality and security concerns, bias an
d fairness issues, medical liability and regulatory challenges, dependence on
complex algorithms and systems, and limited transparency and explainabilit
y.\n6. **Case Studies and Examples:** A section on case studies and example
s of successful AI implementations in healthcare, including predictive analytic
s for disease detection, image analysis for cancer diagnosis, chatbots for pati
ent engagement and support, and AI-powered clinical decision support syste
ms.\n7. **Conclusion:** A conclusion summarizing the report\'s key findings a
nd recommendations.\n\n**Step 5: Finalize and Deliver the Report**\n\nIn this
step, we will finalize the report, addressing any remaining gaps or concerns, a
nd deliver the report in the agreed-upon format (e.g., PDF, Word document, P
owerPoint presentation).\n\n**Finalized Report:**\n\nThe finalized report will i

nclude a comprehensive review of the benefits and risks of AI in healthcare, in
cluding case studies and examples of successful AI implementations in health
care. The report will provide recommendations for healthcare stakeholders on
how to effectively implement and integrate AI into their healthcare systems, w
hile minimizing the risks and challenges associated with AI.\n\n**Delivered Re
port:**\n\nThe report will be delivered in a PDF format, with a summary or brie
f overview of the report\'s key findings and recommendations. The report will i
nclude visual aids and tables to illustrate key findings and statistics, and will b
e easy to read and understand.\nMarket Findings:\n**Task Breakdown: Analyz
e AI Benefits and Risks in Healthcare and Create Report**\n\n**Objective:** T
o conduct an in-depth analysis of the benefits and risks of Artificial Intelligenc
e (AI) in the healthcare sector and create a comprehensive report.\n\n**Resea
rch and Literature Review (Timeframe: 2-3 days)**\n\n1. **Identify relevant st
udies, articles, and reports on AI applications in healthcare:**\n\t* PubMed (s
cholarly articles): Retrieved 15 studies on AI in healthcare, including research
on AI-powered diagnostic tools and AI-driven clinical decision support system
s.\n\t* Google Scholar (academic articles and research papers): Retrieved 25
studies on AI in healthcare, including research on AI-powered chatbots and AI
-driven predictive analytics.\n\t* ScienceDirect (peer-reviewed articles and jo
urnals): Retrieved 10 studies on AI in healthcare, including research on AI-pow
ered image analysis and AI-driven personalized medicine.\n\t* ResearchGate
(research papers and articles): Retrieved 5 studies on AI in healthcare, includi
ng research on AI-powered clinical trials and AI-driven medical device develo
pment.\n\t* AI in Healthcare reports from organizations like HIMSS, HIMSA, an
d AI in Healthcare Association: Retrieved 5 reports on AI in healthcare, includi
ng research on AI-powered healthcare workflows and AI-driven patient engag
ement.\n2. **Review existing literature on AI benefits and risks in healthcare:*
\n\t Improved diagnosis accuracy: AI-powered diagnostic tools have been s
hown to improve diagnosis accuracy in various medical conditions, including
cancer and cardiovascular disease.\n\t* Enhanced patient outcomes: AI-drive
n personalized medicine has been shown to improve patient outcomes in vari
ous medical conditions, including diabetes and chronic obstructive pulmonary
disease (COPD).\n\t* Streamlined clinical workflows: AI-powered clinical decis
ion support systems have been shown to streamline clinical workflows and im
prove administrative efficiency.\n\t* Data security concerns: AI in healthcare i
s vulnerable to data security concerns, including cyber-attacks and data brea

ches.\n\t* Bias and fairness issues: AI in healthcare is vulnerable to bias and f
airness issues, including racial and ethnic disparities in healthcare.\n\t* Medic
al liability and regulatory challenges: AI in healthcare is vulnerable to medical l
iability and regulatory challenges, including liability for AI-driven medical devi
ce malfunctions.\n3. **Note down key findings, insights, and gaps in existing r
esearch:**\n\t* AI applications in healthcare are increasing, with a focus on im
proving diagnosis accuracy and patient outcomes.\n\t* AI has the potential to
streamline clinical workflows and improve administrative efficiency.\n\t* Howe
ver, AI in healthcare is not without risks, including data security concerns, bia
s and fairness issues, and medical liability and regulatory challenges.\n\n**Be
nefits of AI in Healthcare (Timeframe: 2-3 days)**\n\n1. **Categorize AI benef
its in healthcare into:**\n\t* Diagnostic accuracy and speed\n\t* Personalized
medicine and treatment\n\t* Patient engagement and adherence\n\t* Clinical d
ecision support and workflow optimization\n\t* Administrative efficiency and c
ost reduction\n2. **Gather examples and case studies of successful AI imple
mentations in healthcare:**\n\t+ Predictive analytics for disease detection (e.
g., cancer, diabetes)\n\t\t- A study published in the Journal of the American M
edical Association (JAMA) found that AI-powered predictive analytics improve
d disease detection in patients with cancer.\n\t+ Image analysis for cancer dia
gnosis (e.g., mammography, CT scans)\n\t\t- A study published in the journal
Radiology found that AI-powered image analysis improved cancer diagnosis i
n patients with breast cancer.\n\t+ Chatbots for patient engagement and supp
ort (e.g., patient education, medication adherence)\n\t\t- A study published in
the Journal of Medical Systems found that AI-powered chatbots improved pati
ent engagement and adherence to treatment plans.\n\t+ AI-powered clinical d
ecision support systems (e.g., IBM Watson, Google Health)\n\t\t- A study publi
shed in the Journal of the American Medical Informatics Association (JAMIA) f
ound that AI-powered clinical decision support systems improved clinical deci
sion-making in patients with cardiovascular disease.\n3. **Document the ben
efits of AI in healthcare:**\n\t- Improved patient outcomes and increased accu
racy in diagnosis and treatment.\n\t- Enhanced patient engagement and adher
ence to treatment plans.\n\t- Streamlined clinical workflows and improved ad
ministrative efficiency.\n\n**Risks and Challenges of AI in Healthcare (Timefra
me: 2-3 days)**\n\n1. **Identify and categorize AI risks and challenges in heal
thcare into:**\n\t* Data quality and security concerns\n\t* Bias and fairness is
sues\n\t* Medical liability and regulatory challenges\n\t* Dependence on com

plex algorithms and systems\n\t* Limited transparency and explainability\n2. **Gather examples and case studies of AI failures or limitations in healthcare:**\n\t+ Misdiagnosis or delayed diagnosis due to AI errors (e.g., AI misinterpreting medical images)\n\t\t- A study published in the Journal of the American Medical Association (JAMA) found that AI-powered diagnostic tools misdiagnosed patients with cancer.\n\t+ Bias in AI-driven decision-making (e.g., AI perpetuating existing health disparities)\n\t\t- A study published in the journal Science found that AI-powered algorithms perpetuated existing health disparities in patients with cardiovascular disease.\n\t+ Regulatory compliance and liability issues (e.g., AI-driven medical device approvals)\n\t\t- A study published in the Journal of Medical Systems found that AI-powered medical devices were vulnerable to regulatory compliance and liability issues.\n\t+ Technical limitations and maintenance challenges (e.g., AI system downtime or data loss)\n\t\t- A study published in the Journal of the American Medical Informatics Association (JAMIA) found that AI-powered systems were vulnerable to technical limitations and maintenance challenges.\n3. **Document the risks and challenges of AI in healthcare:**\n\t- Potential patient harm or financial losses due to AI errors or bias.\n\t- Regulatory compliance and liability issues related to AI-driven medical devices or decisions.\n\t- Technical limitations and maintenance challenges affecting AI system reliability and effectiveness.\n\n**Report Writing and Analysis (Timeframe: 4-5 days)**\n\n1. **Compile the research, literature review, and analysis into a comprehensive report:**\n\t+ Include an executive summary, introduction, and conclusion.\n\t+ Organize the report into clear sections and subsections.\n2. **Develop a clear and concise writing style, using visual aids and tables to illustrate key findings and statistics:**\n\t+ Use charts, graphs, and infographics to visualize key data and trends.\n\t+ Include tables and figures to illustrate the benefits and risks of AI in healthcare.\n3. **Edit and proofread the report for accuracy, clarity, and consistency:**\n\t+ Review the report for grammar, spelling, and punctuation errors.\n\t+ Ensure the report is well-organized and easy to follow.\n\n**Finalize and Deliver the Report (Timeframe: 1-2 days)**\n\n1. **Finalize the report, addressing any remaining gaps or concerns:**\n\t+ Ensure the report meets the requirements and expectations of the stakeholders.\n\t+ Address any remaining questions or concerns related to the report\'s findings and recommendations.\n2. **Deliver the report in the agreed-upon format (e.g., PDF, Word document, PowerPoint presentation):**\n\t+ Ensure the report is well-formatted and easy to read.\n\t+ Include a su

mmary or brief overview of the report\'s key findings and recommendations.\n
\n**Report**\n\n**Executive Summary**\n\nArtificial intelligence (AI) has the
potential to revolutionize the healthcare sector, improving patient outcomes, st
reamlining clinical workflows, and reducing healthcare costs. However, AI in h
ealthcare is not without risks, including data security concerns, bias and fairn
ess issues, and medical liability and regulatory challenges. This report provide
s an in-depth analysis of the benefits and risks of AI in healthcare, highlightin
g the need for careful consideration of AI\'s potential impact on the healthcare
sector.\n\n**Introduction**\n\nAI has been increasingly adopted in healthcar
e, with applications in diagnostic tools, clinical decision support systems, and
personalized medicine. However, the adoption of AI in healthcare has also rais
ed concerns about data security, bias, and regulatory compliance.\n\n**Benef
its of AI in Healthcare**\n\nAI has the potential to improve diagnosis accuracy,
patient outcomes, and clinical workflows in healthcare. AI-powered diagnostic
tools have been shown to improve diagnosis accuracy in various medical con
ditions, including cancer and cardiovascular disease. AI-driven personalized
medicine has been shown to improve patient outcomes in various medical con
ditions, including diabetes and COPD. AI-powered clinical decision support sy
stems have been shown to streamline clinical workflows and improve administ
rative efficiency.\n\n**Risks and Challenges of AI in Healthcare**\n\nAI in heal
thcare is vulnerable to data security concerns, bias and fairness issues, and
medical liability and regulatory challenges. AI-powered diagnostic tools have
been shown to misdiagnose patients with cancer. AI-powered algorithms have
been shown to perpetuate existing health disparities in patients with cardiovas
cular disease. AI-powered medical devices have been shown to be vulnerable
to regulatory compliance and liability issues.\n\n**Conclusion**\n\nAI has the
potential to revolutionize the healthcare sector, improving patient outcomes, st
reamlining clinical workflows, and reducing healthcare costs. However, AI in h
ealthcare is not without risks, including data security concerns, bias and fairn
ess issues, and medical liability and regulatory challenges. Careful considerati
on must be given to AI\'s potential impact on the healthcare sector, including t
he need for robust data security measures, careful algorithm development, an
d careful regulatory oversight.\n\n**Recommendations**\n\n1. Develop robust
data security measures to protect patient data and prevent data breaches.\n2.
Implement careful algorithm development and testing to prevent AI errors and
bias.\n3. Establish clear regulatory frameworks for AI-powered medical devic

es and algorithms.\n4. Provide ongoing training and education for healthcare professionals on AI-powered clinical decision support systems.\n\n**References**\n\n1. Journal of the American Medical Association (JAMA). (2022). AI-powered diagnostic tools improve diagnosis accuracy in patients with cancer.\n2. Journal of Medical Systems. (2022). AI-powered chatbots improve patient engagement and adherence to treatment plans.\n3. Journal of the American Medical Informatics Association (JAMIA). (2022). AI-powered clinical decision support systems improve clinical decision-making in patients with cardiovascular disease.\n4. Science. (2022). AI-powered algorithms perpetuate existing health disparities in patients with cardiovascular disease.\n5. Journal of Medical Systems. (2022). AI-powered medical devices vulnerable to regulatory compliance and liability issues.\n\n**Appendices**\n\nAppendix A: AI-powered diagnostic tools and algorithms\n\nAppendix B: AI-powered clinical decision support systems and workflows\n\nAppendix C: AI-powered personalized medicine and treatment\n\nAppendix D: AI-powered medical devices and regulatory compliance\n\nAppendix E: AI-powered data security measures and best practices', 'analysis': '', 'report': '**Report: Analyzing AI Benefits and Risks in Healthcare**\n\n**Executive Summary**\n\nArtificial intelligence (AI) is transforming the healthcare industry by providing numerous benefits and introducing new risks. This report aims to analyze the advantages and disadvantages of AI in healthcare, highlighting its potential to improve patient outcomes, streamline processes, and enhance decision-making.\n\n**Benefits of AI in Healthcare**\n\n1. **Improved Diagnosis Accuracy**: AI algorithms can analyze vast amounts of medical data, reducing diagnostic errors and improving patient outcomes.\n2. **Enhanced Patient Experience**: AI-powered chatbots and virtual assistants can provide personalized care, answer patient queries, and facilitate efficient communication between patients and healthcare providers.\n3. **Streamlined Clinical Processes**: AI can automate routine tasks, such as data entry, prescription management, and medical billing, freeing up healthcare professionals to focus on high-value tasks.\n4. **Predictive Analytics**: AI can analyze large datasets to identify high-risk patients, predict disease progression, and prevent hospital readmissions.\n5. **Personalized Medicine**: AI can help tailor treatment plans to individual patients based on their genetic profiles, medical histories, and lifestyle factors.\n\n**Risks of AI in Healthcare**\n\n1. **Data Bias and Inaccuracy**: AI algorithms can perpetuate existing biases and inaccuracies in medical data, leading to discriminatory outcomes and inco

rrect diagnoses.\n2. **Cybersecurity Risks**: AI systems can be vulnerable to cyber threats, compromising patient data and disrupting healthcare service s.\n3. **Dependence on AI**: Over-reliance on AI can lead to decreased hum an decision-making skills and reduced critical thinking abilities among healthc are professionals.\n4. **Lack of Transparency**: AI decision-making process es can be opaque, making it challenging to understand and explain the reason ing behind AI-driven recommendations.\n5. **Liability and Accountability**: A s AI becomes more integrated into healthcare, questions arise about liability a nd accountability in the event of AI-related errors or adverse outcomes.\n\n** Mitigating Risks and Maximizing Benefits**\n\n1. **Implement Robust Data Go vernance**: Establish clear data management policies, ensure data quality, an d implement data validation processes to prevent bias and inaccuracies.\n2. * *Develop AI Transparency and Explainability**: Design AI systems that provid e clear explanations for their decision-making processes, enabling healthcare professionals to understand and trust AI-driven recommendations.\n3. **Inves t in Cybersecurity**: Implement robust security measures to protect patient d ata and prevent cyber threats.\n4. **Monitor and Evaluate AI Performance**: Regularly assess AI performance, identify areas for improvement, and refine A I systems to optimize their benefits.\n5. **Develop AI Literacy and Training**: Educate healthcare professionals on AI basics, its limitations, and its potential applications to ensure informed adoption and effective use.\n\n**Conclusion* *\n\nAI has the potential to revolutionize healthcare by improving patient outc omes, streamlining processes, and enhancing decision-making. However, it al so introduces new risks, including data bias, cybersecurity threats, and depen dence on AI. By understanding these benefits and risks, healthcare organizati ons can mitigate potential pitfalls and maximize the positive impact of AI in he althcare.\n\n**Recommendations**\n\n1. **Develop a comprehensive AI strate gy**: Establish clear goals, objectives, and implementation plans for AI adopti on in healthcare.\n2. **Invest in AI literacy and training**: Educate healthcare professionals on AI basics and its applications to ensure informed adoption.\n 3. **Implement robust data governance**: Establish clear data management p olicies and ensure data quality to prevent bias and inaccuracies.\n4. **Develo p AI transparency and explainability**: Design AI systems that provide clear e xplanations for their decision-making processes.\n5. **Monitor and evaluate AI performance**: Regularly assess AI performance and refine AI systems to optimize their benefits.\n\nBy following these recommendations, healthcare or

```
ganizations can harness the potential of AI to improve patient care, reduce co
sts, and enhance the overall quality of healthcare services.', 'summary': '', 'res
earch_complete': False}
```

Notes

- The system uses **state flags** and `current_agent` to navigate transitions.
- Conditional routing enables **dynamic workflow orchestration**.
- All agents are **lightweight and modular**, making the system extensible.

7.4.2 Hierarchical Multi-Agent Architecture (Approach 2)

Overview

This alternative architecture provides a **more granular and condition-driven structure** for a multi-agent workflow, mimicking roles like CEO, researchers, analysts, and writers. Agents are invoked conditionally based on real-time analysis of the user prompt and system state, using LangGraph's `StateGraph` for dynamic routing.

---

1. System Setup

```
import logging
from typing import Dict, Literal, Optional, TypedDict
from pydantic import BaseModel, Field
import os
from dotenv import load_dotenv
from langchain_groq import ChatGroq
from langchain_core.messages import HumanMessage
from langgraph.graph import StateGraph, END

# Logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)
```

```
s - %(message)s')
logger = logging.getLogger(__name__)

# Load environment and initialize model
load_dotenv()
llm = ChatGroq(
    model_name="llama3-8b-8192",
    temperature=0.7,
    api_key=os.getenv("GROQ_API_KEY")
)
```

2. State Definition

```
class WorkflowState(TypedDict):
    current_stage: Literal[
        'ceo', 'research_lead', 'data_researcher',
        'market_researcher', 'analyst', 'writing_lead',
        'technical_writer', 'summary_writer', '__end__']
    user_input: str
    requirements: Dict[str, bool]
    research_data: Dict[str, Optional[str]]
    analysis: Optional[str]
    deliverables: Dict[str, Optional[str]]

def create_initial_state(user_input: str) → WorkflowState:
    return {
        "current_stage": "ceo",
        "user_input": user_input,
        "requirements": {
            'needs_data_research': False,
            'needs_market_research': False,
            'needs_technical_report': False,
            'needs_exec_summary': False
        },
        "research_data": {
```

```
        'data_findings': None,
        'market_findings': None
    },
    "analysis": None,
    "deliverables": {
        'technical_report': None,
        'executive_summary': None
    }
}
```

---

3. Agent Definitions

🔷 LLM Invocation Wrapper

```
def call_llm(prompt: str) → str:
    try:
        response = llm.invoke([HumanMessage(content=prompt)])
        return response.content
    except Exception as e:
        logger.error(f"LLM call failed: {e}")
        return f"Error: {str(e)}"
```

🔷 CEO Agent

```
def ceo_agent(state: WorkflowState) → WorkflowState:
    user_request = state["user_input"].lower()
    state["requirements"] = {
        'needs_data_research': any(w in user_request for w in ['data', 'statistic',
'number']),
        'needs_market_research': any(w in user_request for w in ['market', 'tren
d', 'industry']),
        'needs_technical_report': 'technical' in user_request or 'report' in user_re
quest,
        'needs_exec_summary': 'executive' in user_request or 'summary' in user_
```

```
request
    }
    state["current_stage"] = 'research_lead' if (
        state["requirements"]['needs_data_research'] or
        state["requirements"]['needs_market_research']
    ) else 'writing_lead'
    return state
```

🔷 Research Lead

```
def research_lead_agent(state: WorkflowState) → WorkflowState:
    state["current_stage"] = 'data_researcher' if state["requirements"]['needs_
data_research'] else 'market_researcher'
    return state
```

🔷 Data Researcher

```
def data_researcher_agent(state: WorkflowState) → WorkflowState:
    prompt = f"Generate data research about: {state['user_input']}"
    state["research_data"]['data_findings'] = call_llm(prompt)
    state["current_stage"] = 'market_researcher' if state["requirements"]['need
s_market_research'] else 'analyst'
    return state
```

🔷 Market Researcher

```
def market_researcher_agent(state: WorkflowState) → WorkflowState:
    prompt = f"Analyze market trends for: {state['user_input']}"
    state["research_data"]['market_findings'] = call_llm(prompt)
    state["current_stage"] = 'analyst'
    return state
```

🔷 Analyst Agent

```
def analyst_agent(state: WorkflowState) → WorkflowState:
    combined = ""
    if state["research_data"]['data_findings']:
        combined += f"Data Findings:\n{state['research_data']['data_findings']}
\n\n"
    if state["research_data"]['market_findings']:
        combined += f"Market Findings:\n{state['research_data']['market_finding
s']}"
    state["analysis"] = call_llm(f"Synthesize and analyze:\n{combined}")
    state["current_stage"] = 'writing_lead'
    return state
```

🔷 Writing Lead

```
def writing_lead_agent(state: WorkflowState) → WorkflowState:
    if state["requirements"]['needs_technical_report'] and not state["deliverable
s"]['technical_report']:
        state["current_stage"] = 'technical_writer'
    elif state["requirements"]['needs_exec_summary'] and not state["deliverabl
es"]['executive_summary']:
        state["current_stage"] = 'summary_writer'
    else:
        state["current_stage"] = '__end__'
    return state
```

🔷 Technical Writer

```
def technical_writer_agent(state: WorkflowState) → WorkflowState:
    prompt = f"Write technical report on {state['user_input']}\nAnalysis: {state
['analysis']}"
    state["deliverables"]['technical_report'] = call_llm(prompt)
    state["current_stage"] = 'writing_lead'
    return state
```

🔷 Executive Summary Writer

```
def summary_writer_agent(state: WorkflowState) → WorkflowState:
    prompt = f"Write executive summary for: {state['user_input']}\nKey points:
{state['analysis']}"
    state["deliverables"]['executive_summary'] = call_llm(prompt)
    state["current_stage"] = 'writing_lead'
    return state
```

---

4. Graph Definition

```
def build_workflow():
    workflow = StateGraph(WorkflowState)
    workflow.add_node("ceo", ceo_agent)
    workflow.add_node("research_lead", research_lead_agent)
    workflow.add_node("data_researcher", data_researcher_agent)
    workflow.add_node("market_researcher", market_researcher_agent)
    workflow.add_node("analyst", analyst_agent)
    workflow.add_node("writing_lead", writing_lead_agent)
    workflow.add_node("technical_writer", technical_writer_agent)
    workflow.add_node("summary_writer", summary_writer_agent)

    workflow.set_entry_point("ceo")

    workflow.add_conditional_edges(
        "ceo",
        lambda state: "research_lead" if (
            state["requirements"]["needs_data_research"] or
            state["requirements"]["needs_market_research"]
        ) else "writing_lead",
        {"research_lead": "research_lead", "writing_lead": "writing_lead"}
    )

    workflow.add_conditional_edges(
        "research_lead",
```

```
        lambda state: "data_researcher" if state["requirements"]["needs_data_re
search"] else "market_researcher",
        {"data_researcher": "data_researcher", "market_researcher": "market_res
earcher"}
    )

    workflow.add_conditional_edges(
        "data_researcher",
        lambda state: "market_researcher" if state["requirements"]["needs_mark
et_research"] else "analyst",
        {"market_researcher": "market_researcher", "analyst": "analyst"}
    )

    workflow.add_edge("market_researcher", "analyst")
    workflow.add_edge("analyst", "writing_lead")

    workflow.add_conditional_edges(
        "writing_lead",
        lambda state: (
            "technical_writer" if (
                state["requirements"]["needs_technical_report"] and
                not state["deliverables"]["technical_report"]
            ) else "summary_writer" if (
                state["requirements"]["needs_exec_summary"] and
                not state["deliverables"]["executive_summary"]
            ) else "__end__"
        ),
        {
            "technical_writer": "technical_writer",
            "summary_writer": "summary_writer",
            "__end__": END
        }
    )

    workflow.add_edge("technical_writer", "writing_lead")
    workflow.add_edge("summary_writer", "writing_lead")
```

```
    return workflow
```

5. Run Example

```
def run_workflow(prompt: str):
    workflow = build_workflow().compile()
    initial_state = create_initial_state(prompt)

    # Optionally force requirements (for demo/testing)
    initial_state["requirements"]["needs_technical_report"] = True
    initial_state["requirements"]["needs_exec_summary"] = True

    final_state = workflow.invoke(initial_state)
    return final_state["deliverables"]

# Example Execution
if __name__ == "__main__":
    sample_prompt = (
        "Analyze AI adoption in healthcare including data statistics and market tr
ends. "
        "Provide both technical report and executive summary."
    )

    results = run_workflow(sample_prompt)

    print("\n=== TECHNICAL REPORT ===")
    print(results["technical_report"][:1000])

    print("\n=== EXECUTIVE SUMMARY ===")
    print(results["executive_summary"][:500])
```

Output:

```
🚀 Starting research workflow...
2025-07-16 15:27:19,440 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:27:20,670 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:27:22,717 - INFO - HTTP Request: POST https://api.groq.com/o
penai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:27:22,825 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:27:22,834 - INFO - Retrying request to /openai/v1/chat/comple
tions in 16.000000 seconds
2025-07-16 15:27:41,456 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:27:41,563 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:27:41,566 - INFO - Retrying request to /openai/v1/chat/complet
ions in 9.000000 seconds
2025-07-16 15:27:52,004 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
🚀 Running workflow...
2025-07-16 15:27:54,345 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:27:54,453 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:27:54,458 - INFO - Retrying request to /openai/v1/chat/comple
tions in 7.000000 seconds
2025-07-16 15:28:03,985 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:28:04,111 - INFO - HTTP Request: POST https://api.groq.com/o
penai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:28:04,115 - INFO - Retrying request to /openai/v1/chat/completi
ons in 25.000000 seconds
2025-07-16 15:28:31,018 - INFO - HTTP Request: POST https://api.groq.com/o
penai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:28:31,134 - INFO - HTTP Request: POST https://api.groq.com/o
```

```
penai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:28:31,136 - INFO - Retrying request to /openai/v1/chat/completi
ons in 27.000000 seconds
2025-07-16 15:29:00,000 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 200 OK"
2025-07-16 15:29:00,202 - INFO - HTTP Request: POST https://api.groq.com/
openai/v1/chat/completions "HTTP/1.1 429 Too Many Requests"
2025-07-16 15:29:00,205 - INFO - Retrying request to /openai/v1/chat/comple
tions in 13.000000 seconds
2025-07-16 15:29:15,051 - INFO - HTTP Request: POST https://api.groq.com/o
penai/v1/chat/completions "HTTP/1.1 200 OK"

=== TECHNICAL REPORT ===
**Technical Report: Analyzing AI Adoption in Healthcare**

**Introduction:**

Artificial Intelligence (AI) has revolutionized the healthcare industry by improvi
ng patient outcomes, streamlining clinical workflows, and enhancing diagnosti
c accuracy. This report provides an in-depth analysis of AI adoption in healthc
are, including market trends, statistics, and case studies.

**Market Trends:**

1. **Growing Demand:** The global healthcare AI market is expected to reach
$13.4 billion by 2025, growing at a Compound Annual Growth Rate (CAGR) of
40.2% from 2020 to 2025 (Source: MarketsandMarkets).
2. **Increased Adoption:** 83% of healthcare organizations have already ado
pted AI, with 44% planning to increase their AI investments in the next two ye
ars (Source: Accenture).
3. **Key Applications:** The top three AI applications in healthcare are Predict
ive Analytics (48%), Natural Language Processing (NLP) (36%), and Machine
Learning (34%) (Source: Research2Guidance).

**Statistics:**
```

```
1. **A...

=== EXECUTIVE SUMMARY ===
**Executive Summary:**

The adoption of Artificial Intelligence (AI) in healthcare is accelerating, driven by growing demand and increasing investment. Our research highlights key in sights and recommendations to inform healthcare organizations about the ben efits and challenges of AI adoption.

**Key Findings:**

* The global healthcare AI market is expected to reach $13.4 billion by 2025, g rowing at a CAGR of 40.2%.
* 83% of healthcare organizations have already adopted AI, with 44% plannin g to ...
```

Summary

- Agents operate based on real-time flag evaluation.
- Research and writing are split into distinct flows.
- Output quality can be customized by refining prompts per agent.
- Very modular and scalable structure, adaptable to many enterprise domains.

8. Tool Integration

What is Tool Integration?

Tool Integration refers to the process of connecting an AI system with external tools, APIs, or services. These tools allow agents to take real-world actions like sending emails, searching the web, updating databases, or controlling applications — based on user input or LLM output.

This extends the AI's functionality from just generating text to performing useful, automated actions.

We'll now explore different tools integrated with multi-agent systems:

8.1 Email Integration Using LLM

This tool enables an AI chatbot to automatically generate email content and send it to users using SMTP. It integrates a lightweight LLM (like Groq's LLaMA3) with email functionality using Python's built-in `smtplib` and Gmail credentials.

Code:

```
import os
import smtplib
from dotenv import load_dotenv
from email.mime.text import MIMEText
from langchain_groq import ChatGroq
from langchain.schema import HumanMessage

# ------------------------------
# 📌 Load sensitive credentials from .env file
# ------------------------------
load_dotenv()

EMAIL_ADDRESS = os.getenv("EMAIL_ADDRESS")       # Your Gmail address
EMAIL_PASSWORD = os.getenv("EMAIL_PASSWORD")      # Gmail app passw
ord
GROQ_API_KEY = os.getenv("GROQ_API_KEY")         # Your Groq API key

# ------------------------------
# 🧠 Initialize Groq LLM (LLaMA3)
# ------------------------------
llm = ChatGroq(
    temperature=0.3,
    groq_api_key=GROQ_API_KEY,
```

```
    model_name="llama3-8b-8192"  # Lightweight and fast model
)

# ------------------------------
# 📤 Function to send email
# ------------------------------
def send_email(subject, body, to_email):
    msg = MIMEText(body)
    msg["Subject"] = subject
    msg["From"] = EMAIL_ADDRESS
    msg["To"] = to_email

    try:
        with smtplib.SMTP_SSL("smtp.gmail.com", 465) as server:
            server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            server.sendmail(EMAIL_ADDRESS, to_email, msg.as_string())
        print("✅ Email sent successfully.")
    except Exception as e:
        print("❌ Failed to send email:", str(e))

# ------------------------------
# 🤖 Main chatbot loop
# ------------------------------
def main():
    print("🤖 Welcome to the Groq-Powered Email Chatbot!")
    print("Type your message. Type 'exit' to quit.\n")

    while True:
        # 🔹 Take user input
        user_input = input("🧑 You: ")
        if user_input.lower() in ["exit", "quit"]:
            print("👋 Goodbye!")
            break

        # 🔹 Get response from LLM
        response = llm([HumanMessage(content=user_input)])
```

```
        print("🤖 Bot:", response.content)

        # 🔹 Ask whether to send the response via email
        send_choice = input("📩 Do you want to send this reply via email? (y/n): ").strip().lower()
        if send_choice == "y":
            to_email = input("📧 Enter recipient's email address: ").strip()
            send_email(subject="Message from Your AI Chatbot", body=response.content, to_email=to_email)

# ------------------------------
# 🚀 Run the chatbot
# ------------------------------
if __name__ == "__main__":
    main()
```

Sample Output

```
🤖 Welcome to the Groq-Powered Email Chatbot!
Type your message. Type 'exit' to quit.

🧑 You: what is the meaning of the name "Kashaf Zahra"
D:\AGENTICAI\1.tool_integration\send_email_tool.py:58: LangChainDeprecationWarning: The method `BaseChatModel.__call__` was deprecated in langchain-core 0.1.7 and will be removed in 1.0. Use :meth:`~invoke` instead.
  response = llm([HumanMessage(content=user_input)])
🤖 Bot: What a beautiful name!

"Kashaf Zahra" is a Persian name, and it's a combination of two words:

1. "Kashaf" (کشف) means "revelation" or "unveiling" in Persian. It's often used to describe the act of revealing a hidden truth or secret.
2. "Zahra" (زهرا) is a popular Persian given name that means "blooming flower" or "blossom". It's also the name of the daughter of the Prophet Muhammad (peace be upon him) and is considered a symbol of purity and innocence.
```

Together, "Kashaf Zahra" can be interpreted as "Revelation of the Blossom" or "Unveiling of the Flower". It's a beautiful and meaningful name
that suggests the person with this name is a symbol of growth, beauty, and re
velation.

In Islamic tradition, the name Zahra is also associated with the concept of "Nu
r" (light), which is a symbol of guidance and enlightenment. So, "Kashaf Zahr
a" can also be seen as a name that represents the illumination of the heart an
d mind, and the revelation of truth and guidance.

Overall, "Kashaf Zahra" is a lovely and meaningful name that carries a rich cul
tural and spiritual significance.
📩 Do you want to send this reply via email? (y/n): y
📧 Enter recipient's email address: aafreenzk1214@gmail.com
✅ Email sent successfully.
🧑 You: exit
👋 Goodbye!

to me

What a beautiful name!

"Kashaf Zahra" is a Persian name, and it's a combination of two words:

1. "Kashaf" (کشف) means "revelation" or "unveiling" in Persian. It's often used to describe the act of revealing a hidden truth or secret.
2. "Zahra" (زهرا) is a popular Persian given name that means "blooming flower" or "blossom". It's also the name of the daughter of the Prophet Muhammad (peace be upon him) and is considered a symbol of purity and innocence.

Together, "Kashaf Zahra" can be interpreted as "Revelation of the Blossom" or "Unveiling of the Flower". It's a beautiful and meaningful name that suggests the person with this name is a symbol of growth, beauty, and revelation.

In Islamic tradition, the name Zahra is also associated with the concept of "Nur" (light), which is a symbol of guidance and enlightenment. So, "Kashaf Zahra" can also be seen as a name that represents the illumination of the heart and mind, and the revelation of truth and guidance.

Overall, "Kashaf Zahra" is a lovely and meaningful name that carries a rich cultural and spiritual significance.

8.2 Groq-Powered Smart Email Assistant (Python)

8.2.1 Project Overview

This project implements a command-line **Smart Email Assistant** that:

- Connects securely to a Gmail inbox via IMAP
- Fetches and summarizes recent emails
- Searches for emails using keyword-based filtering
- Utilizes **Groq's LLaMA3 (via LangChain)** for contextual Q&A based on inbox content

It's ideal for power users or developers looking to integrate **LLMs** with real-world email automation and analysis.

8.2.2 Tech Stack

Component	Role
Python	Core programming language
`imaplib`	Connects to Gmail inbox via IMAP
`email`	Parses and decodes email messages
`dotenv`	Handles secure environment variable loading
`langchain`	Bridges app logic with LLM interface
`Groq`	LLM backend (LLaMA3-8B) for ultra-fast inference

8.2.3 Environment Configuration

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

```
EMAIL_ADDRESS=your_email@gmail.com
EMAIL_PASSWORD=your_app_password  # Use Gmail App Password
GROQ_API_KEY=your_groq_api_key
```

> Note: Do not use your regular Gmail password. Enable 2FA and generate an App Password from your Google Account settings.

8.2.4 Key Features

Inbox Summary

- Prompts user for how many recent emails to summarize
- Outputs `From` and `Subject` fields in a clean format
- Optionally allows you to ask an LLM a follow-up question

Keyword-Based Email Search

- Searches emails where the **subject or sender** matches a keyword
- Returns the first few hundred characters of matching email content
- Option to ask contextual questions about search results

Groq-Powered Interaction

- Seamlessly integrates Groq's LLaMA3 model via LangChain
- Supports multilingual queries and summaries
- Fast and cost-effective LLM performance using Groq API

8.3 Code Architecture

8.3.1 Environment & LLM Setup

```
import os
import imaplib
import email
from email.header import decode_header
from dotenv import load_dotenv
from langchain_groq import ChatGroq
from langchain.schema import HumanMessage

# Load credentials
load_dotenv()
EMAIL_ADDRESS = os.getenv("EMAIL_ADDRESS")
EMAIL_PASSWORD = os.getenv("EMAIL_PASSWORD")
GROQ_API_KEY = os.getenv("GROQ_API_KEY")
```

```
# Initialize LLM
llm = ChatGroq(
    temperature=0.3,
    groq_api_key=GROQ_API_KEY,
    model_name="llama3-8b-8192"
)
```

Explanation

- **Library Imports**: Imports all necessary modules for email access (`imaplib`, `email`), secure credential loading (`dotenv`), and Groq LLM interaction (`langchain_groq`).
- **Credential Management**: Uses `python-dotenv` to securely load sensitive data such as email login and API keys from a `.env` file.
- **LLM Initialization**: Configures and instantiates Groq's **LLaMA3 model** using LangChain, enabling fast and affordable large language model queries.

8.3.2 **Email Summary Function**

```
# 📥 Fetch email summary
def fetch_email_summary():
    try:
        mail = imaplib.IMAP4_SSL("imap.gmail.com")
        mail.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
        mail.select("inbox")

        total_emails = int(input("📩 How many recent emails do you want to sum
marize? (e.g., 5, 10): "))
        if total_emails <= 0:
            return "❗ Invalid number entered."

        status, messages = mail.search(None, "ALL")
        mail_ids = messages[0].split()
        latest_n = mail_ids[-total_emails:]
```

```
        summary = ""
        for num in reversed(latest_n):
            _, data = mail.fetch(num, "(RFC822)")
            msg = email.message_from_bytes(data[0][1])

            subject, encoding = decode_header(msg["Subject"])[0]
            if isinstance(subject, bytes):
                subject = subject.decode(encoding or "utf-8", errors="ignore")

            from_ = msg.get("From")
            summary += f"From: {from_}\nSubject: {subject}\n" + "-" * 30 + "\n"

        mail.logout()
        return summary

    except Exception as e:
        return f"❌ Error: {str(e)}"
```

Explanation

- Connects to the Gmail inbox using secure IMAP.
- Prompts the user for how many recent emails to summarize.
- Extracts and decodes the sender (`From`) and subject line from each email.
- Returns a formatted text summary for quick viewing.
- Ensures a clean logout from the email server after completion.

8.2.3 **Keyword-Based Email Search**

```
def fetch_specific_email(keyword):
    try:
        mail = imaplib.IMAP4_SSL("imap.gmail.com")
        mail.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
        mail.select("inbox")
```

```
        status, messages = mail.search(None, "ALL")
        mail_ids = messages[0].split()
        mail_ids.reverse()

        matched_emails = []

        for num in mail_ids:
            _, data = mail.fetch(num, "(RFC822)")
            msg = email.message_from_bytes(data[0][1])

            subject, encoding = decode_header(msg["Subject"])[0]
            if isinstance(subject, bytes):
                subject = subject.decode(encoding or "utf-8", errors="ignore")

            from_ = msg.get("From")
            full_content = ""

            if msg.is_multipart():
                for part in msg.walk():
                    if part.get_content_type() == "text/plain":
                        body = part.get_payload(decode=True)
                        if body:
                            full_content = body.decode(errors="ignore")
                            break
            else:
                full_content = msg.get_payload(decode=True).decode(errors="ignor
e")

            if keyword.lower() in subject.lower() or keyword.lower() in from_.lower
():
                matched_emails.append(
                    f"📧 From: {from_}\n📌 Subject: {subject}\n\n📄 Content:\n{full_c
ontent[:800]}...\n{'-'*50}"
                )

        mail.logout()
```

```
        if matched_emails:
            return "\n\n".join(matched_emails)
        else:
            return f"❗ No email found matching the keyword: {keyword}"

    except Exception as e:
        return f"❌ Error: {str(e)}"
```

Explanation

- Connects to the Gmail inbox via IMAP and retrieves all email IDs.
- Iterates through each email in reverse (most recent first), parsing subject, sender, and content.
- Decodes subject lines and extracts the plain text body, handling both plain and multipart emails.
- Filters emails where the **keyword matches** the subject or sender (case-insensitive).
- Returns a formatted preview (up to 800 characters) of each matching email.
- Logs out after processing to ensure secure closure of the session.

4. **LLM Interaction (LangChain)**

```
response = llm([HumanMessage(content=prompt)]).content
```

5. **Main CLI Assistant**

```
# 🤖 Smart Email Bot
def smart_email_bot():
    print("🤖 Groq Smart Email Assistant")
    print("-----------------------------")

    while True:
        print("\nLLM: What would you like to do?")
```

```
        print("1️⃣ View inbox summary")
        print("2️⃣ Search for a specific email (by keyword)")
        print("❌ Exit the assistant")

        user_intent = input("🧑 Your choice (summary / keyword / exit): ").strip().
lower()

        if user_intent == "exit":
            print("👋 Goodbye! The assistant has exited.")
            break

        elif user_intent == "summary":
            summary = fetch_email_summary()
            print("\n📬 Inbox Summary:\n", summary)

            follow_up = input("\n🧑 Would you like to ask something about this su
mmary? (y/n): ").strip()
            if follow_up.lower() == "y":
                question = input("❓ Your question (English or any language): ")
                prompt = f"Inbox summary:\n{summary}\n\nQuestion:\n{question}"
                response = llm([HumanMessage(content=prompt)]).content
                print("\n🤖 LLM Response:\n", response)

        elif user_intent == "keyword":
            keyword = input("🔍 Enter a keyword (e.g., YouTube, job, Google) to se
arch for emails: ")
            specific_data = fetch_specific_email(keyword)
            print("\n📨 Matching Emails:\n", specific_data)

            ask_llm = input("\n❓ Would you like to ask something about these em
ails? (y/n): ").strip()
            if ask_llm.lower() == "y":
                q = input("🧑 Your question (English or any language): ")
                prompt = f"Here are the emails:\n{specific_data}\n\nNow answer thi
s question:\n{q}"
                response = llm([HumanMessage(content=prompt)]).content
```

```
            print("\n🤖 LLM Response:\n", response)

        else:
            print("⚠️ Invalid input. Please type only 'summary', 'keyword', or 'exit'.")
```

Explanation

- Provides a **CLI-based assistant interface** to interact with your inbox.
- Offers three main options:
  - `summary`: Fetch a summary of recent emails and optionally ask LLM-based questions.
  - `keyword`: Search emails by keyword and ask follow-up questions.
  - `exit`: Terminates the assistant.
- Integrates tightly with the `fetch_email_summary()` and `fetch_specific_email()` functions for backend logic.
- Uses Groq's **LLaMA3 via LangChain** to answer contextual questions in any language.

---

6. Main Execution Block (`__main__`)

```
# 🚀 Run
if __name__ == "__main__":
    smart_email_bot()
```

This block defines the **script's entry point**:

- Ensures that the assistant only runs when the script is executed directly (not when imported).
- Follows Python best practices for building **modular, testable code**.
- Prevents unintended execution when integrating this file into larger applications.

> ✅ Always include this pattern to control script execution cleanly and predictably.

Outputs

Inbox Summary Example

```
LLM: What would you like to do?
1️⃣ View inbox summary
2️⃣ Search for a specific email (by keyword)
❌ Exit the assistant
🧑 Your choice (summary / keyword / exit): summary
📩 How many recent emails do you want to summarize? (e.g., 5, 10): 20

📬 Inbox Summary:
 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: Kaggle <noreply@kaggle.com>
Subject: Competition Launch: MAP - Charting Student Math Misunderstanding
s
------------------------------
From: Google <no-reply@accounts.google.com>
Subject: Security alert
------------------------------
From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: "Samsung Electronics Pakistan" <samsungelectronics@pk.email.samsu
ng.com>
Subject: Big summer savings on Galaxy A Series!
```

```
------------------------------
From: "Mustakbil Jobs" <support@mustakbil.com>
Subject: Abdullah  Jobs in Islamabad Pakistan 🚀
------------------------------
From: "Binance" <do_not_reply@mailersp2.binance.com>
Subject: 🔥New Listing: Trade Spot & Share 6M ERA
------------------------------
From: "Notion Team" <team@mail.notion.so>
Subject: How  project management teams get more done with AI
------------------------------
From: Google <no-reply@accounts.google.com>
Subject: Security alert
------------------------------
From: UNICEF Data <data@mail.unicef.org>
Subject: New data on immunization
------------------------------
From: "Mustakbil Jobs" <support@mustakbil.com>
Subject: Abdullah  Jobs in Islamabad Pakistan 🚀
------------------------------
From: "Binance" <do_not_reply@mailersp2.binance.com>
Subject: Begin Spot Trading Today and Receive a 20 USDT Trading Rebate!
------------------------------
From: "Binance" <do_not_reply@mailersp1.binance.com>
Subject: Begin Spot Trading Today and Receive a 20 USDT Trading Rebate!
------------------------------
From: "Kawish Minhas (via Google Sheets)" <drive-shares-dm-noreply@goog
le.com>
Subject: Spreadsheet shared with you: "Mecha Interns Task Sheet "
------------------------------
From: "Notion Team" <team@mail.notion.so>
Subject: Bring all your work into one place
------------------------------
From: Mastercard Developers <customerfeedback@survey.mastercard.com>
Subject: We'd love to hear from you!
------------------------------
From: "Binance" <do_not_reply@mailersp1.binance.com>
```

```
Subject: [ H O _ _ P _ G E ] 👀 Theme: Binance UI Refined
------------------------------
From: "Mustakbil Jobs" <support@mustakbil.com>
Subject: Abdullah  Jobs in Islamabad Pakistan 🚀
------------------------------
From: admission2@au.edu.pk
Subject: AU Admsiossn Portal (Password Reset - Email Verfication)
```

Keyword Search Output

```
LM: What would you like to do?
1️⃣ View inbox summary
2️⃣ Search for a specific email (by keyword)
❌ Exit the assistant
🧑 Your choice (summary / keyword / exit): keyword
🔍 Enter a keyword (e.g., YouTube, job, Google) to search for emails: aafreen

📨 Matching Emails:
 📧 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
📌 Subject:

📄 Content:
hloo
...
--------------------------------------------------

📧 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
📌 Subject:

📄 Content:
oyeeeeeeeeeeeeeeeee
...
--------------------------------------------------

📧 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
```

```
📌 Subject:

📄 Content:
hloo oyee kia hal ha v
...
--------------------------------------------------
```

LLM Q&A Response

```
🧑 Would you like to ask something about this summary? (y/n): y
❓ Your question (English or any language): provide summary of all these ema
il.
D:\AGENTICAI\1.tool_integration\fetch_email_tool.py:129: LangChainDeprecati
onWarning: The method `BaseChatModel.__call__` was deprecated in langchai
n-core 0.1.7 and will be removed in 1.0. Use :meth:`~invoke` instead.
  response = llm([HumanMessage(content=prompt)]).content

🤖 LLM Response:
 Here is a summary of the emails:

**Email 1:** Aafreen Kazmi sent an email with no subject or content, likely an
empty email.

**Email 2:** Kaggle launched a new competition called "MAP - Charting Stud
ent Math Misunderstandings" for data science enthusiasts.

**Email 3:** Google sent a security alert to inform the recipient of a potential s
ecurity issue.

**Email 4:** Aafreen Kazmi sent another empty email with no subject or conte
nt.

**Email 5:** Aafreen Kazmi sent another empty email with no subject or conte
nt.
```

Email 6: Samsung Electronics Pakistan sent an email promoting their Gala xy A Series with a summer sale.

Email 7: Mustakbil Jobs sent a job posting for a job in Islamabad, Pakista n, with the name "Abdullah".

Email 8: Binance, a cryptocurrency exchange, announced a new listing for the 6M ERA token.

Email 9: Notion Team, a project management tool, sent an email highlighti ng how AI can help project management teams be more productive.

Email 10: Google sent another security alert to inform the recipient of a po tential security issue.

Let me know if you'd like me to summarize anything further!

Would you like to ask something about these emails? (y/n): y
🧑 Your question (English or any language): provide me the summary of these emails

🤖 LLM Response:
A summary of these emails!

It appears that these are a series of brief, informal emails from Aafreen Kazmi to an unknown recipient. The content of the emails is quite casual and lacks a ny specific topic or purpose.

Here's a brief summary:

* The first email starts with a casual "hloo" (likely a typo or a colloquialism).
* The second email is an enthusiastic "oyeeeeeeeeeeeeeeeee" (which seems to be an expression of excitement or joy).
* The third email is a mix of "hloo", "oyee", and a few other words, but the me

aning is unclear.

Overall, these emails appear to be brief, casual, and lacking in substance or p urpose.

```
LLM: What would you like to do?
1️⃣View inbox summary
2️⃣Search for a specific email (by keyword)
❌ Exit the assistant
🧑 Your choice (summary / keyword / exit): summary
📩 How many recent emails do you want to summarize? (e.g., 5, 10): 10

📬 Inbox Summary:
 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: Kaggle <noreply@kaggle.com>
Subject: Competition Launch: MAP - Charting Student Math Misunderstandings
------------------------------
From: Google <no-reply@accounts.google.com>
Subject: Security alert
------------------------------
From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: Aafreen Kazmi <aafreenzk1214@gmail.com>
Subject:
------------------------------
From: "Samsung Electronics Pakistan" <samsungelectronics@pk.email.samsung.com>
Subject: Big summer savings on Galaxy A Series!
------------------------------
From: "Mustakbil Jobs" <support@mustakbil.com>
Subject: Abdullah  Jobs in Islamabad Pakistan 🚀
------------------------------
From: "Binance" <do_not_reply@mailersp2.binance.com>
Subject: 🔥New Listing: Trade Spot & Share 6M ERA
------------------------------
From: "Notion Team" <team@mail.notion.so>
Subject: How  project management teams get more done with AI
------------------------------
```

```
🧑 Would you like to ask something about this summary? (y/n): y
❓ Your question (English or any language): provide summary of all these email.
D:\AGENTICAI\1.tool_integration\fetch_email_tool.py:129: LangChainDeprecationWarning: The method `BaseChatModel.__call__` was deprecated in la
ngchain-core 0.1.7 and will be removed in 1.0. Use :meth:`~invoke` instead.
  response = llm([HumanMessage(content=prompt)]).content

🤖 LLM Response:
 Here is a summary of the emails:

**Email 1:** Aafreen Kazmi sent an email with no subject or content, likely an empty email.

**Email 2:** Kaggle launched a new competition called "MAP - Charting Student Math Misunderstandings" for data science enthusiasts.

**Email 3:** Google sent a security alert to inform the recipient of a potential security issue.

**Email 4:** Aafreen Kazmi sent another empty email with no subject or content.

**Email 5:** Aafreen Kazmi sent another empty email with no subject or content.

**Email 6:** Samsung Electronics Pakistan sent an email promoting their Galaxy A Series with a summer sale.

**Email 7:** Mustakbil Jobs sent a job posting for a job in Islamabad, Pakistan, with the name "Abdullah".

**Email 8:** Binance, a cryptocurrency exchange, announced a new listing for the 6M ERA token.

**Email 9:** Notion Team, a project management tool, sent an email highlighting how AI can help project management teams be more productive.

**Email 10:** Google sent another security alert to inform the recipient of a potential security issue.

Let me know if you'd like me to summarize anything further!
```

```
LLM: What would you like to do?
1️⃣ View inbox summary
2️⃣ Search for a specific email (by keyword)
❌ Exit the assistant
🧑 Your choice (summary / keyword / exit): keyword
🔍 Enter a keyword (e.g., YouTube, job, Google) to search for emails: aafreen

📩 Matching Emails:
 📨 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
📌 Subject:

📄 Content:
hloo
...
--------------------------------------------------

📨 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
📌 Subject:

📄 Content:
oyeeeeeeeeeeeeeeeee
...
--------------------------------------------------

📨 From: Aafreen Kazmi <aafreenzk1214@gmail.com>
📌 Subject:

📄 Content:
hloo oyee kia hal ha v
...
--------------------------------------------------
```

```
🧑 Your question (English or any language): provide me the summary of these emails

🤖 LLM Response:
 A summary of these emails!

It appears that these are a series of brief, informal emails from Aafreen Kazmi to an unknown recipient. The content of the emails is quite ca
sual and lacks any specific topic or purpose.

Here's a brief summary:

* The first email starts with a casual "hloo" (likely a typo or a colloquialism).
* The second email is an enthusiastic "oyeeeeeeeeeeeeeeee" (which seems to be an expression of excitement or joy).
* The third email is a mix of "hloo", "oyee", and a few other words, but the meaning is unclear.

Overall, these emails appear to be brief, casual, and lacking in substance or purpose.
```

```
Overall, these emails appear to be brief, casual, and lacking in substance or purpose.

LLM: What would you like to do?
1️⃣ View inbox summary
2️⃣ Search for a specific email (by keyword)
❌ Exit the assistant
🧑 Your choice (summary / keyword / exit): exit
👋 Goodbye! The assistant has exited.
```

How to Run

```
# 1. Install dependencies
pip install langchain langchain_groq python-dotenv

# 2. Add .env file with your credentials

# 3. Run the assistant
python smart_email_bot.py
```

Security Notes

- Use **App Passwords** for Gmail access
- Keep `.env` files out of version control (`.gitignore`)
- Avoid sending sensitive email content to LLMs if not necessary
- Logout from IMAP sessions after use.
