

```

In [2]: import os
        from dotenv import load_dotenv
        from langchain.chat_models import ChatOpenAI
        from langchain.prompts import PromptTemplate
        from langchain.chains import LLMChain
        from langchain.agents import Tool, initialize_agent, AgentType
        from langchain.tools import TavilySearchResults

        # 🗝️ Load API keys
        load_dotenv(".env")
        openai_api_key = os.getenv("OPENAI_API_KEY")
        tavily_api_key = os.getenv("TAVILY_API_KEY")

        # ✨ Initialize LLM
        llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)

        # ✅ Tool 1: Simple QA Tool
        qa_prompt = PromptTemplate.from_template("Answer clearly: {question}")
        qa_chain = LLMChain(llm=llm, prompt=qa_prompt)
        qa_tool = Tool(
            name="Simple QA",
            func=qa_chain.run,
            description="Answers factual questions clearly"
        )

        # ✅ Tool 2: Summarizer Tool
        summary_prompt = PromptTemplate.from_template("Summarize this text:\n\n{text}")
        summary_chain = LLMChain(llm=llm, prompt=summary_prompt)
        summary_tool = Tool(
            name="Summarizer",
            func=summary_chain.run,
            description="Summarizes long paragraphs or text content"
        )

        # ✅ Tool 3: Web Search Tool (Tavily)
        search_tool = Tool(
            name="Web Search",
            func=TavilySearchResults(max_results=3).run,
            description="Search the internet for current and live information"
        )

        # 🔧 Wrap all tools in an agent
        tools = [qa_tool, summary_tool, search_tool]

        agent_executor = initialize_agent(
            tools=tools,
            llm=llm,
            agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
            verbose=True # shows reasoning process
        )

        # 🚀 Run user queries
        queries = [
            "What is LangGraph in LangChain?"

```

```

    # "Summarize this: LangChain is a framework to build LLM apps using prompts"
    # "Latest news about OpenAI GPT-4o"
]

for query in queries:
    print("\n👤 User Query:", query)
    response = agent_executor.run(query)
    print("\n🤖 Agent Response:", response)

```

👤 User Query: What is LangGraph in LangChain?

> Entering new AgentExecutor chain...

I should use the Simple QA tool to answer this factual question.

Action: Simple QA

Action Input: "What is LangGraph in LangChain?"

Observation: LangGraph in LangChain is a graph database that stores and manages language data, including words, phrases, and their relationships. It is a key component of the LangChain platform, allowing for efficient storage, retrieval, and analysis of language-related information.

I now know the final answer

Final Answer: LangGraph in LangChain is a graph database that stores and manages language data.

> Finished chain.

🤖 Agent Response: LangGraph in LangChain is a graph database that stores and manages language data.

In [3]: response

Out[3]: 'OpenAI GPT-4o is a multilingual, multimodal generative pre-trained transformer developed by OpenAI and released in May 2024. It can process and generate text, images, and audio. In August 2024, OpenAI introduced a new feature allowing corporate customers to customize GPT-4o using proprietary company data through fine-tuning.'

In []: