LangSmith Demo

Prerequisites

• Python 3.10+ (3.11 recommended).

• A terminal (macOS Terminal / Windows PowerShell / Linux shell).

• Your Demo2 script (e.g., LangGraph_Demo2_LangSmith.py or Demo2.py).

• If using Ollama: Ollama installed and a model pulled (example: llama3.2:3b).

• A LangSmith account + API key (if you want the website traces).

Step 1 — Create a project folder and virtual environment

macOS / Linux

```
mkdir LangSmith_Demo2
cd LangSmith_Demo2
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
```

Windows (PowerShell)

```
mkdir LangSmith_Demo2
cd LangSmith_Demo2
python -m venv .venv
.\.venv\Scripts\Activate
python -m pip install --upgrade pip
```

Step 2 — Install required packages

Install LangGraph + LangChain + LangSmith + your LLM integration (Ollama example below).

```
pip install -U langgraph langchain langsmith langchain-ollama
python-dotenv typing_extensions
```

Step 3 — (Optional) Set up Ollama for local LLMs

If your Demo2 uses Ollama (ChatOllama), verify it works and pull a model.

```
ollama --help
ollama pull llama3.2:3b
ollama run llama3.2:3b "Say hello in one sentence."
```

Step 4 — Enable LangSmith tracing

LangSmith tracing is controlled by environment variables. You can set them via a .env file.

Use a .env file.

Create a file named .env in the same folder as your script and add:

```
LANGCHAIN_TRACING=true
LANGCHAIN_API_KEY=lsv2_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LANGCHAIN_PROJECT=LangSmith
# Optional (sometimes helpful):
# LANGCHAIN_ENDPOINT=https://api.smith.langchain.com
```

Your Python code must load this file near the top:

```
from dotenv import load_dotenv
load_dotenv()
```

Step 5 — Run your Demo2 script

From the same folder (with your virtual environment activated), run:

```
python *Demo2.py
```

```
=== RUN: QUESTION ===
FINAL RESULT:
 {'payload': [{'time_of_comment': '2025-01-20', 'customer_remark': 'Why has the packaging changed?', 'social_media_channel': 'faceb
ook', 'number_of_likes': 100}], 'text': 'Why has the packaging changed?', 'route': 'question', 'answer': 'We appreciate you bringin
g this to our attention and are looking into the matter.'}

STREAM:

{'extract_content': {'text': 'Why has the packaging changed?'}}
{'llm_route': {'route': 'question'}}
{'run_question_code': {'answer': 'Thanks for your question. We will look into it.'}}
{'beautify_llm': {'answer': 'We appreciate you bringing this to our attention and are looking into the matter.'}}

=== RUN: COMPLIMENT ===
FINAL RESULT:
 {'payload': [{'time_of_comment': '2025-01-21', 'customer_remark': 'I love your product—great job!', 'social_media_channel': 'insta
gram', 'number_of_likes': 42}], 'text': 'I love your product—great job!', 'route': 'compliment', 'answer': 'I appreciate your kind
words about my service.'}

STREAM:

{'extract_content': {'text': 'I love your product—great job!'}}
{'llm_route': {'route': 'compliment'}}
{'run_compliment_code': {'answer': 'Thanks for the compliment.'}}
{'beautify_llm': {'answer': 'I appreciate your kind words about my service.'}}
```

Step 6 — View runs on the LangSmith website

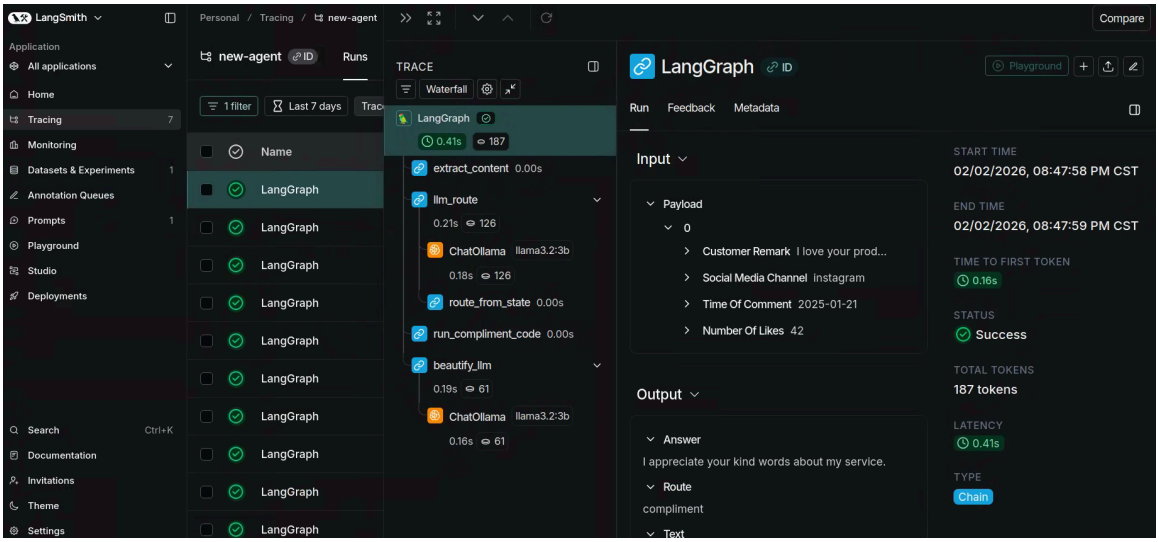
Go to the LangSmith site: https://smith.langchain.com

In the left sidebar, click Tracing (or Projects → Tracing).

Open the project named exactly like LangSmith.

You will see a table of Runs. Click any row (green check) to open the run.

In the run details page, expand the trace tree on the left to see node-level steps and LLM calls.



Tip: The table view shows only one row per run. The node-by-node details appear after you click into a run.

References

• LangSmith docs: search for 'LangSmith tracing' in LangChain documentation.

• Ollama install docs: https://ollama.com/download