🔬 Lab 8 Environment setting and LangSmith Tutorial

Part 1: Environment Setup

Before working with LangSmith, ensure your environment is properly configured with the following prerequisites:

- Python `3.11` or higher (3.11+ required for LangSmith compatibility)

  - Environment Setup Steps can be found here in Lab 0.

- Project dependencies from `requirements.txt` (separate files for MacOS and Windows)

  - Download the file to the same directory as your .env file.

  - Rename the downloaded file to `requirements.txt`

  - Activate the your virtual environment by running the code below:

    pip install -r requirements.txt

- `LangSmith API key` (see below 2.1)

- Access to `CGU VPN` for PGVector connection

Part 2: LangSmith Tutorial

LangSmith is a powerful platform for developing, evaluating, and monitoring language model applications. This tutorial will guide you through:

- Setting up LangSmith with your project

- Creating and tracking chains and agents

- Evaluating model performance

- Debugging complex workflows

- Monitoring your applications

Go to https://www.langchain.com/langsmith to create an free account.

1. Setup and Authentication

- To create an API key head to the LangSmith settings page. Then click **Create API Key.**

- Copy and paste the API key into your .env file:

```
LANGSMITH_API_KEY = "lsv2_                                    de"
```

- Set up the configuration in your source code, here we use an example from lab 8

```python
langsmith_api_key = os.getenv("LANGSMITH_API_KEY")

# Enable LangSmith tracing for observability/debugging
os.environ["LANGCHAIN_TRACING"] = "true"
# Set the project name for LangSmith, it will create a new project if it doesn't exist
os.environ["LANGCHAIN_PROJECT"] = "GenAI-Class-Lab8"
```

- (Optional) You may use this as a python decorator if you define a function that is not integrated into LangChain, LangGraph function or class.

> # Import the `trace` decorator from LangSmith to enable tracing of some individual customized function calls and metadata for observability/debugging.
> from langsmith import trace

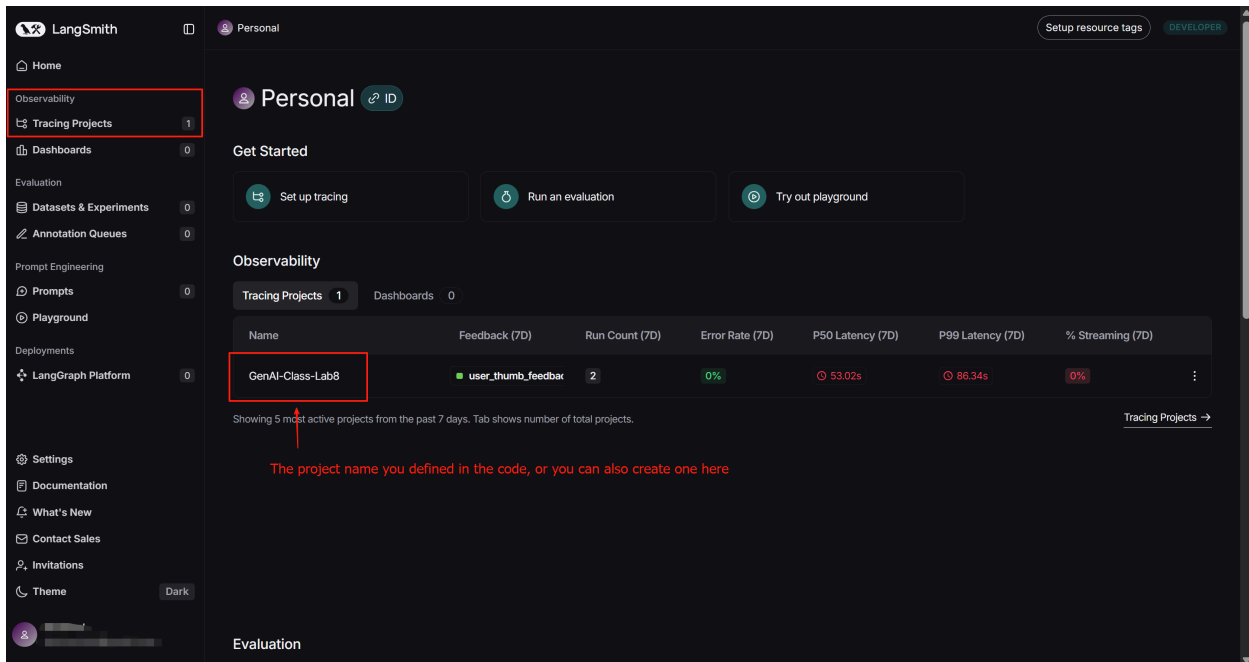# 2. Observability, Monitoring and Evaluation

Concepts | 🦜🛠️ LangSmith

This conceptual guide covers topics that are important to understand when logging traces to LangSmith. A Trace is essentially a series of steps that your application takes to go

🦜 https://docs.smith.langchain.com/observability/concepts

- Go to the tracing projects in your LangSmith account

All the runs for this project name will be recorded here. You can check the basic statistics, like the number of runs, error rates, and latency.

- Open one of the tracing projects

You will see all the runs' basic stats, like input, output, token usage, time, and etc.

- Open one of the runs

You will see all the intermediate results and stats for each node. (If you use the trace decorator, it will be displayed here as a node as well.)

You can expand one of the nodes to see more granular details.

For more details, check their official documentation at
https://docs.smith.langchain.com/