Project Setup Steps for Product Price Checker App with LLM Integration

1. Obtain API Keys

1.1. Acquire a **LangSmith API key** from the LangSmith platform: https://smith.langchain.com
 1.2. Obtain an **OpenAI API key** from the OpenAI platform:
https://platform.openai.com/account/api-keys
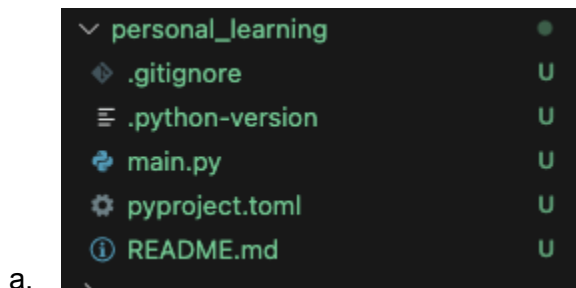
---

2. Set Up the Project Structure Using uv

2.1. Install uv if not already installed:

```
pip install uv
```

2.2. Navigate to the desired project directory and initialize the project:

```
uv init
```

> This command will automatically create a `.venv`, `pyproject.toml`, and other necessary files.

a.

---

3. Install Dependencies

3.1. Add LangChain to the project:

```
uv add langchain
```

> This installs the package and updates `pyproject.toml` accordingly.

```toml
1    [project]
2    name = "personal-learning"
3    version = "0.1.0"
4    description = "Add your description here"
5    readme = "README.md"
6    requires-python = ">=3.12"
7    dependencies = [
8        "langchain>=0.3.25",
9    ]
10
```

PROBLEMS   TERMINAL   ···   ▣ Python - personal_learning   + ∨   ▯   🗑   ···

```
Successfully installed uv-0.7.8
(base) avijit@SPHDELCMLT035 personal_learning % pwd
/Users/avijit/Desktop/local_codes/code/personal_learning
(base) avijit@SPHDELCMLT035 personal_learning % uv init
Initialized project `personal-learning`
(base) avijit@SPHDELCMLT035 personal_learning % uv add langchain
Using CPython 3.12.7 interpreter at: /opt/anaconda3/bin/python3.12
Creating virtual environment at: .venv
Resolved 32 packages in 338ms
Prepared 28 packages in 448ms
Installed 28 packages in 35ms
 + annotated-types==0.7.0
 + anyio==4.9.0
 + certifi==2025.4.26
 + charset-normalizer==3.4.2
 + h11==0.16.0
 + httpcore==1.0.9
 + httpx==0.28.1
 + idna==3.10
 + jsonpatch==1.33
 + jsonpointer==3.0.0
 + langchain==0.3.25
 + langchain-core==0.3.61
 + langchain-text-splitters==0.3.8
 + langsmith==0.3.42
 + orjson==3.10.18
 + packaging==24.2
 + pydantic==2.11.5
 + pydantic-core==2.33.2
 + pyyaml==6.0.2
 + requests==2.32.3
 + requests-toolbelt==1.0.0
 + sniffio==1.3.1
 + sqlalchemy==2.0.41
 + tenacity==9.1.2
```

3.2. Install other required dependencies using the same approach:

```
uv add streamlit
uv add python-dotenv
uv add openai
```

(Include any additional libraries required for your application.)

---

4. Set Environment Variables

4.1. Create a `.env` file in the root of your project.
 4.2. Add the necessary environment variables, e.g.:

```
OPENAI_API_KEY=your_openai_api_key
LANGCHAIN_API_KEY=your_langsmith_api_key
LANGCHAIN_TRACING_V2=true
LANGCHAIN_PROJECT=product-price-checker
```

---

5. Develop the Application

5.1. Implement the **Product Price Checker** backend logic using LangChain or LLM APIs.
 5.2. Create a user interface using **Streamlit**, for example:

```python
# main.py
import streamlit as st

st.title("Product Price Checker")
# UI logic here
```

---

6. Add Code Formatting and Linting Tools

6.1. Install `ruff` using `uv`:

```
uv tool install ruff
uv tool update-shell
```

6.2. Restart your shell/terminal session.

6.3. Auto-fix formatting issues:

```
ruff check --fix .
```

---

7. Activate Virtual Environment

Activate the virtual environment manually:

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

---

8. Run the Application

Start the Streamlit application:

```
streamlit run main.py
```

Access the frontend through the provided local URL in your terminal output.

b.
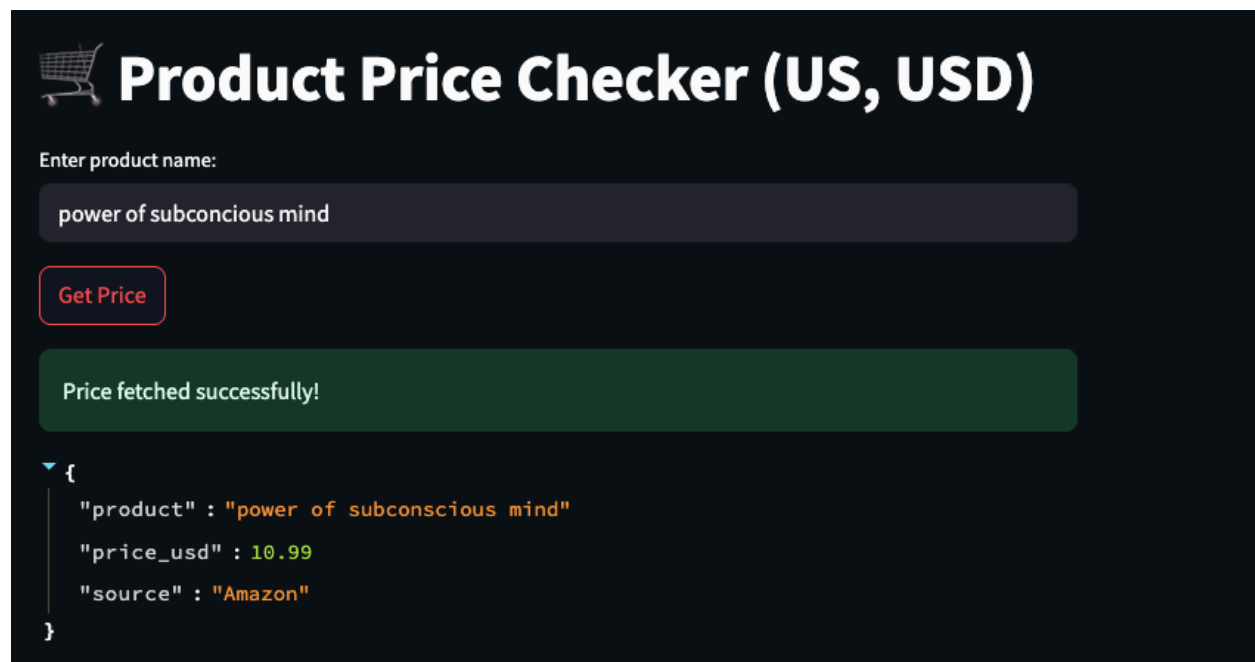
c.

d.

9. Validate in LangSmith UI (Optional)

2.

If logging is enabled for your LangChain workflows, verify traces and run outputs in the **LangSmith UI** at https://smith.langchain.com