# Environment Setup – Detailed Summary

This document explains everything we have done so far to create a clean, working environment for the **Financial Analyst Copilot (AI-Powered RAG System)** project, in simple and clear steps.

## 1■■ Installed Python 3.14

We installed the latest version of Python (3.14.0) on macOS.

This ensures compatibility with modern AI libraries like **LangChain**, **FAISS**, and **SentenceTransformers**.

Why:

Newer Python versions improve performance and prevent library version errors.

## 2■■ Created a Virtual Environment (venv)

We created a private sandbox using:

```

python3 -m venv venv
```

This isolates all project libraries so that they don’t interfere with the system Python or other projects.

Why:

Each project gets its own clean space — like a separate box just for its dependencies.

## 3■■ Activated the Environment

We activated the venv with:

```

source venv/bin/activate
```

After activation, `(venv)` appears in the terminal prompt, meaning every install command (`pip install`) goes into this project only.

Why:

To keep all libraries installed inside this project’s sandbox.

## 4■■ Verified Python Version Inside venv

We confirmed that the environment is using Python 3.14:

```
python --version
```

Output showed `Python 3.14.0`.

Why:

To ensure our environment uses the correct Python version.

## 5■■ Installed Required Libraries

We installed the main tools needed for the Copilot:

```
pip install --no-deps langchain faiss-cpu sentence-transformers streamlit requests beautifulsoup4 pandas
```

Library Purposes:

- **LangChain:** Framework for connecting LLM + retrieval system (RAG).
- **FAISS:** Vector database for fast semantic search.
- **SentenceTransformers:** Creates embeddings (numerical text meaning).
- **Streamlit:** Web-based UI for the Copilot.
- **Requests:** Downloads SEC filings.
- **BeautifulSoup4:** Cleans HTML filings into plain text.
- **Pandas:** Handles structured data and tables.

## 6■■ Skipped PyArrow

The package `pyarrow` failed because it has **no prebuilt wheels for Python 3.14 yet.**
We checked and confirmed that **PyArrow is not required** for this project.

Why:
Your Copilot project processes text (not Parquet or Arrow data), so we safely skipped PyArrow.

## 7■■ Installed CMake (System Tool)

We installed CMake globally using:

```
brew install cmake
```

This allows Python libraries that need compilation (like FAISS) to work properly.

## 8■■ Verified Everything Works

We ran this test:

```
python -c "import langchain, faiss, pandas; print('■ Setup successful!')"
```

and got the message:

```
■ Setup successful!
```

This confirmed all major components are installed correctly.

## 9■■ Current Environment Status

Component	Status	Description
Python 3.14.0	■ Installed	Latest version, ready for AI workloads
Virtual Environment	■ Active	Isolated sandbox for your project

Core Libraries	■ Installed	LangChain, FAISS, etc.
CMake	■ Installed	Enables package compilation
PyArrow	■ Skipped	Not needed for this project
VS Code Interpreter	■ Configured	Uses the correct `venv` Python

## 10■■ What Happens Next

Now that your environment is ready, you can start writing your **project code** in these steps:

1. **ETL Pipeline (`etl_sec_ingestor.py`)** – Fetch and clean SEC filings.
2. **Embedding & Indexing (`embed_indexer.py`)** – Convert text into searchable vectors.
3. **RAG Chain (`rag_chain.py`)** – Connect retriever + LLM.
4. **Evaluation (`evaluator.py`)** – Check accuracy and response quality.
5. **Streamlit UI (`app.py`)** – Create the chat interface for the Copilot.

■ **Summary:**

You now have a modern, clean, and isolated Python environment fully prepared for the **Financial Analyst Copilot** project.

All tools are installed, tested, and ready to use for the next step — building the ETL pipeline.