Why We Use a Virtual Environment (venv)
Quick Summary
We use a virtual environment (venv) to create a clean, isolated workspace for the Financial Analyst Copilot (AI-Powered RAG System) project.
It ensures that all libraries, tools, and dependencies are installed locally — without affecting the macOS system Python.
Key Reasons
1. **Isolation** – Prevents dependency conflicts between projects.
2. **Dependency Control** – Keeps installations local and reproducible using `pip freeze > requirements.txt`.
3. **System Safety** – Protects your global Python environment.
4. **IDE Integration** – Tells VS Code to use the venv for your Copilot project.
5. **Reproducibility** – Anyone can recreate the same setup using `pip install -r requirements.txt`.
In the Context of Financial Analyst Copilot
This project uses:
- LangChain (RAG orchestration)
- FAISS (vector search)
- SentenceTransformers (text embeddings)
- Streamlit (frontend UI)
Keeping these inside a venv ensures stability and avoids version conflicts.
Setup Steps Summary
1. Create environment:
```bash
python3 -m venv venv
```
2. Activate:
```bash
source venv/bin/activate
```
3. Verify:
```bash
python --version # Should show Python 3.14.0
```
4. Install dependencies:
```bash
pip install langchain faiss-cpu sentence-transformers streamlit requests beautifulsoup4 pandas
```
5. Freeze requirements:
```bash
pip freeze > requirements.txt
```
■ In short: Using venv keeps your Financial Analyst Copilot isolated, reproducible, and future-proof while maintaining a clean system setup.