

AI Estate Assistant - Developer Guide

AI Estate Assistant - Developer Guide

This guide covers everything you need to know to develop, run, and maintain your local AI Estate Assistant using Chainlit + LangChain + Ollama.

Overview

Tech Stack:

- **Ollama** (LLM backend, local models)
- **LangChain** (Document handling + logic)
- **Chainlit** (Web chat UI)
- **OneDrive (optional)** (Model storage + cross-device sync)

Goal:

Build a chat assistant capable of answering questions about documents (e.g., financial records), with long-term potential for monetization.

Ollama Setup

Running Ollama

Recommended: Use the Ollama App

- Runs in background automatically
- Does not tie up your terminal
- Starts on boot (optional)

Alternative: `ollama serve` (manual)

```
```bash
ollama serve
```
```

- Locks terminal while running

- Use `Ctrl + C` to stop when finished

****Best Practice Use App for day-to-day development.****

Ollama Model Paths

****Preferred model location (with OneDrive):****

C:/Users/messi/OneDrive/AI/Models/OllamaModels

****Two ways to set model path:****

1. ****config.toml**** (`C:/Users/messi/.ollama/config.toml`)

```
```toml
[paths]
models = "C:/Users/messi/OneDrive/AI/Models/OllamaModels"
```
```

2. ****Environment Variable (priority override)****

```
```bash
OLLAMA_MODELS=C:/Users/messi/OneDrive/AI/Models/OllamaModels
```
```

****Important:**** Restart VS Code or terminal after changing environment variables.

****Verify with:****

```
```bash
ollama list
```
```

Permissions + OneDrive Notes

****Access Denied Errors?****

- Happens if models are in strict OneDrive folders

- Solution: Move to a less restricted folder like `AI/Models`
- Ensure write permissions are granted

Chainlit Setup

Running Chainlit

```
```bash
chainlit run chat.py
```
```

- Opens browser chat UI
- Connects to Ollama automatically
- Supports multi-document input and chat memory

****Keep Ollama running before starting Chainlit!****

Chainlit + LangChain Integration

- Uses LangChain to load documents (DOCX, PDF, etc.)
- Splits documents and stores vectors in FAISS
- Queries are passed to Ollama + results streamed to Chainlit UI

Chainlit Translation Notice

...

Translation file for en-CA not found. Using default translation en-US.

...

This is normal Chainlit does not have Canadian English localization fallback is US English. No action required.

Development Workflow

...

[Start Ollama (App preferred)]

[Run Chainlit chat.py]

[Browser chat available ask questions]

...

****Tip:**** Use a separate terminal window or tab for Chainlit if running `ollama serve` manually.

Troubleshooting Guide

****Ollama shows wrong model path ****

- Check environment variable `OLLAMA_MODELS`
- Check `config.toml`
- Restart terminal after changes

****Chainlit stuck in terminal ****

- Running `ollama serve` locks terminal use new terminal or `Ctrl + C` to stop

****Access Denied in OneDrive ****

- Adjust folder permissions
- Move models to cleaner OneDrive path

****en-CA translation message ****

- Normal no problem

****Ollama list shows no models ****

- Check `OLLAMA_MODELS` path
- Ensure models are in correct folder

Summary

Ollama running in background (App or serve)

Chainlit connects and chats

LangChain loads + indexes documents

Chat memory tracks conversations

Fully integrated local AI assistant READY
