# Local AI Agent

This project runs a local AI assistant with Docker. The assistant can correct text, summarize text, read files from `./files`, save generated files back to `./files`, search the web with HTTPS sources, answer from indexed books with RAG, and keep short chat memory between turns.

## Technologies

This project uses Python for the agent logic, Docker and Docker Compose for runtime orchestration, and Ollama for local language model inference. Web research is handled through local web tooling with optional FastMCP integration, and image OCR uses the OCR.Space API. Document question answering uses a local RAG index stored in `./rag` with BM25-style retrieval implemented in the project code. If you want the original inspiration/source repos, see AIAgent-MCP and FastMCP.

## Quick Start

Install Docker (Engine + Compose). Docker Desktop is not required. Then copy `.env.example` to `.env`:

```
copy .env.example .env
```

Open `.env` and set at least:

```
OLLAMA_MODEL=gemma3:1b
OCR_SPACE_API_KEY=your_key_here
```

You can keep these defaults or change them if you want. `OLLAMA_MODEL` is just an example and can be any local Ollama model tag, including but not limited to `gemma3:4b`. `OCR_SPACE_API_KEY` is only required if you want OCR on image files. If your model tag contains a size like `8b`, make sure `OLLAMA_MAX_B` is high enough (or set to `0` to disable that size check).

For image OCR, the agent chooses OCR language from the prompt (for example French prompts use `fre`, English prompts use `eng`) and shows the current OCR language in phase output. If language cannot be inferred, it falls back to `OCR_SPACE_LANGUAGE` from `.env` (default `eng`).

`OLLAMA_MAX_B` is a safety limit for model size. It helps prevent loading a model that is too large for your machine. For example, with `OLLAMA_MAX_B=4`, a model like `8b` is blocked. If you want to allow any size, set `OLLAMA_MAX_B=0`.

Start the assistant with:

```
docker compose run --rm --build agent
```

The first run downloads the model and can take time.

To restart de agent once the inital setup done.

```
docker compose run --rm agent
```

To stop chat, press `Ctrl+C`. If you want to shut down everything and remove all volumes, run:

```
docker compose down -v
```

## How to Use

Put your input files in `./files` and then ask naturally in chat. You can write prompts like `Correct this: i has a apple.`, `Summarize this file: test.docx`, `What is the latest AI news today?`, or `Summarize https://example.com/page`. The assistant automatically chooses the best mode between web, book, chat, correct, and summarize.

## Book RAG

If you want Q&A on a document, place the book in `./files` and build an index:

```
docker compose run --rm agent python agent.py index --file book.pdf --name
mybook
```

Then start chat and load the index:

```
docker compose run --rm agent
```

Inside chat, use `/book mybook` and ask your questions. By default, chat auto-loads the most recent index on startup. You can disable it with `AGENT_AUTO_BOOK=off` in `.env`, or force a specific one with `AGENT_DEFAULT_BOOK=<index_name>`.

For PDF indexing, all pages are read by default. If you want to cap pages for speed, set `AGENT_MAX_PDF_PAGES` in `.env` to a positive number.

## Chat Commands

You can type `/help` to see commands. You can force web mode with `/research <query>`. You can control verification with `/quality on` or `/quality off`. You can check and clear memory with `/memory` and `/memory clear`. You can switch book mode with `/book <name>` and `/book off`.

## Progress Visibility

During long tasks, the assistant shows phase updates such as searching, fetching URLs, writing, and verifying. By default it also echoes visible `[Phase] ...` lines so you can see progress even when spinner rendering is unreliable in some terminals. You can control this with `AGENT_PHASE_ECHO`, and tune status behavior with `AGENT_PHASE_STYLE`, `AGENT_STATUS_REPEAT_S`, and `AGENT_STATUS_CLEAR_S`.

When model generation takes time, it also prints periodic thinking updates with elapsed seconds. You can tune that interval with `AGENT_THINK_HEARTBEAT_S`.

## Files and Output

Generated files are saved in `./files`. If a filename already exists, the assistant automatically creates a new name with a `_note`, `_note2`, or `_note3` suffix.

## Troubleshooting

If you changed code or settings, run a clean build:

```
docker compose run --rm --build agent
```

To stop services:

```
docker compose down
```

To stop services and remove volumes for a full reset:

```
docker compose down -v
```

If file reading fails, confirm the file is really inside `./files` and that the format is valid.

## Report Mistakes or Bugs

If you find a mistake, please open an issue and include the prompt, full terminal output, file name and type used, date and time, and your OS.

## Disclaimer

This project is still under active development. Behavior, commands, and outputs can change at any time. Always review important results before using them in production or in legal, medical, or financial contexts.

## License

This project is licensed under Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). See `LICENSE` for details and the full legal code link.