SnA Hackathon Project - DealRoom Copilot (MVP)

Setup & Run Guide (macOS + Windows)

What this app does

DealRoom Copilot is a local, offline-first demo that lets you create a deal, upload a data room (PDF/DOCX/PPTX/XLSX/CSV/ZIP), index it into a vector database (RAG), run two agent pipelines (Legal and Financial), and chat with citations grounded in the uploaded documents.

What you need

This guide assumes you have basic command-line access (Terminal on macOS, PowerShell on Windows).

System requirements

- macOS (Apple Silicon or Intel) **or** Windows 10/11.
- Python 3.11+.
- Ollama installed and running (local LLM runtime).
- Disk space for local models (e.g., qwen2.5:7b is several GB).
- Optional (recommended for scanned PDFs): Tesseract + Poppler (OCR).

Important note about model storage

Ollama stores downloaded models in its own system folder (not inside this project). That means your Git repository stays small and will not include model files.

Project folder structure (expected)

From the project root you should see these folders/files:

```
SnA_Hackathon_Project/
  app/
  src/
  storage/
  requirements.txt
  .env.example
```

Step-by-step setup (macOS - first time)

1) Install Homebrew (if you do not have it)

Check:

```
brew --version
```

If it says command not found, install:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/H
```

2) Install Python 3 (if needed)

Check:

```
python3 --version
```

If missing, install via Homebrew:

```
brew install python
```

3) Install Ollama (local LLM runtime)

Install the Ollama app for macOS, then confirm it works in Terminal:

```
ollama --version
```

4) Optional OCR dependencies (recommended)

If you want scanned PDFs to work well, install:

```
brew install tesseract poppler
```

5) Get the project code

If using Git:

```
git clone <YOUR_REPO_URL>
cd SnA_Hackathon_Project
```

If using a ZIP, unzip it and then:

```
cd /path/to/SnA_Hackathon_Project
```

6) Create and activate a virtual environment

```
python3 -m venv .venv
source .venv/bin/activate
python3 -m pip install --upgrade pip
```

7) Install Python dependencies

Install requirements:

```
python3 -m pip install -r requirements.txt
```

If your project uses Office and OCR extractors, also install:

```
python3 -m pip install -U python-docx python-pptx pandas openpyxl pypdf pytesse
```

8) Create .env from .env.example

Files starting with a dot (like .env) are hidden on macOS. That is normal. You can still create and edit them.

```
cp .env.example .env
```

If you want to see hidden files in Finder or VS Code, press Command + Shift + .

9) Start Ollama and pull models

Verify Ollama is running:

```
curl http://localhost:11434/api/tags
```

If it fails, open the Ollama app (Applications -> Ollama) and retry.

Download a chat model and an embedding model:

```
ollama pull qwen2.5:7b
ollama pull nomic-embed-text
```

Check installed models:

```
ollama list
```

10) Run the Streamlit app

```
source .venv/bin/activate
streamlit run app/Home.py
```

The app opens in your browser (usually at http://localhost:8501).

Step-by-step setup (Windows 10/11 - first time)

1) Install Python 3.11+

Install Python from python.org and make sure you tick **Add python.exe to PATH** during setup.

Verify in PowerShell:

```
python --version
python -m pip --version
```

2) Install Git (optional but recommended)

If you plan to clone the repo, install Git for Windows. Otherwise, download the project ZIP.

3) Install Ollama (Windows)

Install the Ollama Windows app, then verify in PowerShell:

```
ollama --version
```

If the command is not recognized, restart PowerShell (or reboot once) and try again.

4) Optional OCR dependencies (recommended for scanned PDFs)

If you need OCR, install **Tesseract** and **Poppler** on Windows. After installation, ensure both are on PATH (or configure their paths in your environment).

- Tesseract: install an official/maintained Windows build and add its install folder to PATH.
- Poppler: download a Windows build (poppler-utils) and add its *bin* folder to PATH.

5) Get the project code

If using Git (PowerShell):

```
git clone <YOUR_REPO_URL>
cd SnA_Hackathon_Project
```

If using a ZIP, unzip it and then:

```
cd C:\path\to\SnA_Hackathon_Project
```

6) Create and activate a virtual environment

```
python -m venv .venv
.\.venv\Scripts\activate
python -m pip install --upgrade pip
```

7) Install Python dependencies

```
python -m pip install -r requirements.txt
```

If your project uses Office and OCR extractors, also install:

```
python -m pip install -U python-docx python-pptx pandas openpyxl pypdf pytesser
```

8) Create .env from .env.example

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

9) Start Ollama and pull models

Verify Ollama is running:

```
curl http://localhost:11434/api/tags
```

Download a chat model and an embedding model:

```
ollama pull qwen2.5:7b
ollama pull nomic-embed-text
ollama list
```

10) Run the Streamlit app

```
.\.venv\Scripts\activate
streamlit run app\Home.py
```

The app opens in your browser (usually at http://localhost:8501).

How to use the app (demo flow)

A) Create / select a deal (Screen 1)

On Home, create a deal or pick one from the dropdown. All uploads and results are scoped per deal_id.

B) Upload documents (Documents tab)

Upload any of: PDF, DOCX, PPTX, XLSX, CSV, ZIP.

- ZIP uploads are unpacked and supported files are queued automatically.

C) Ingest into RAG (Qdrant) - indexing step

Click the button to ingest queued docs into RAG. This performs extraction, chunking, embeddings, and writes vectors + metadata into Qdrant.

D) Run Router + Agents (CrewAI) - two pipelines

Click the button to run the agent pipelines.

- Router agent classifies doc_type (financial / contract / other).
- Legal agent extracts risk level, red flags, evidence snippets.
- Financial agent computes metrics/ratios and flags anomalies (rule-based MVP).

E) Explore the tabs

- Deal Overview: executive summary cards + generated headlines.
- Financial: anomalies + metrics (if financial docs exist).
- Legal & Contracts: risks sorted High/Medium/Low with evidence snippets.
- Documents: doc_type + analyzed status per file.
- Copilot Chat: ask questions; answers include citations and a Sources expander.

Troubleshooting (common issues)

Ollama not found

If Terminal/PowerShell says 'ollama: command not found' or it is not recognized:

- Make sure the Ollama app is installed.
- Restart Terminal/PowerShell (on Windows, a reboot may help).
- Run: `ollama --version`

Cannot connect to localhost:11434

If curl to Ollama fails:

```
curl http://localhost:11434/api/tags
```

- Open the Ollama app and retry.

Qdrant storage folder already accessed

This happens if more than one Streamlit/Python process is holding the local Qdrant folder.

macOS fix:

```
pkill -f streamlit || true
find . -name "__pycache__" -type d -prune -exec rm -rf {} +
source .venv/bin/activate
streamlit run app/Home.py
```

Windows fix (PowerShell):

```
Get-Process python,streamlit -ErrorAction SilentlyContinue | Stop-Process -Forc
python -m pip cache purge
.\.venv\Scripts\activate
streamlit run app\Home.py
```

If still stuck, reset the local vector store (you will re-ingest docs):

```
rm -rf storage/qdrant
mkdir -p storage/qdrant
```

Scanned PDFs show no text

Install OCR dependencies and re-ingest:

```
brew install tesseract poppler
python3 -m pip install -U pytesseract pdf2image
```

On Windows, ensure Tesseract + Poppler are installed and on PATH, then reinstall:

```
python -m pip install -U pytesseract pdf2image
```

ModuleNotFoundError: src ...

Run Streamlit from the project root (where app/ and src/ exist). Check:

```
pwd
ls
```

Quick run (after you have set it up once)

macOS:

```
cd SnA_Hackathon_Project
source .venv/bin/activate
streamlit run app/Home.py
```

Windows:

```
cd SnA_Hackathon_Project
.\.venv\Scripts\activate
streamlit run app\Home.py
```

End of document.