Agent-OS + LangSmith Demo Project

This document contains everything needed to start the LangSmith Demo project with Agent-OS and Curso

---
1. Cursor Agent Prompt

ROLE
You are a Senior AI Engineer + Solutions Architect. Build a production-grade, audit-friendly LangSmi

CONTEXT
We're starting a fresh repo called `langsmith-demo`. We will:
- Implement a minimal but enterprise-ready RAG API with FastAPI.
- Integrate LangSmith for tracing & evaluations (offline + online hooks).
- Add datasets & evaluators to gate changes (CI).
- Provide clean project structure, docs, Makefile, and GitHub Actions.

GOALS
1) Ship a working FastAPI service with /health, /v1/answer (RAG), /v1/evaluate/offline (batch eval).
2) LangSmith tracing on every critical step (retrieval, LLM call, synth).
3) Offline eval pipeline (dataset + evaluators: groundedness, correctness/helpfulness).
4) Quality gates: type-check, lint, tests, and a minimum eval score threshold (e.g., groundedness ≥ 0
5) Clear docs & ADRs.

NON-GOALS
- Fancy UI. No frontend.
- Cloud infra (Terraform/K8s) for now; keep Dockerfile ready.

TECH STACK & RULES
- Python 3.11, FastAPI, Uvicorn.
- LangChain, LangSmith.
- Retrieval: local Markdown corpus, FAISS or Chroma.
- Testing: pytest + httpx, coverage ≥ 80%.
- Lint: ruff; Typing: mypy --strict.
- Docs: README.md, ARCHITECTURE.md, OPERATIONS.md, SECURITY.md.
- ADRs for technical decisions.
- CI: GitHub Actions.
- Security: .env via secrets, PII redaction.

PROJECT SKELETON (see repo structure inside spec section).

---
2. Standards (~/.agent-os/standards/)

tech-stack.md
- Python 3.11
- FastAPI + Uvicorn
- LangChain + LangSmith
- Vectorstore: FAISS or Chroma
- Tests: pytest, httpx, coverage ≥ 80%
- Lint: ruff; Typing: mypy --strict
- CI: GitHub Actions with eval gates
- Security: .env via secrets, PII redaction

best-practices.md
- TDD first; ADR for decisions
- Logs JSON with request_id
- Compliance: versioned eval datasets
- CI gate: groundedness ≥ 0.75, correctness ≥ 0.70

---
3. Product (.agent-os/product/)

mission.md
LangSmith Demo: FastAPI microservice with RAG and tracing.
Goal: correct and grounded answers (groundedness ≥ 0.75).

stack.md
FastAPI 0.110+, Python 3.11
LangChain + LangSmith
Vectorstore: FAISS
Tests: pytest + httpx
Infra: Dockerfile, GitHub Actions

roadmap.md
Backlog:
- RAG pipeline

```
- LangSmith tracing
- Offline eval pipeline

In-progress:
(none)

Done:
(none)

decisions.md
2025-09-24: Chosen FastAPI + LangSmith + FAISS as initial stack.

------------------------------------------------------------
## 4. First Spec (.agent-os/specs/2025-09-24-tracing-evals/)

srd.md
Feature: Add LangSmith tracing to /v1/answer and offline eval pipeline.
Acceptance:
- /v1/answer returns {answer, sources, trace_url}
- Offline eval runs dataset, generates report
- CI fails if metrics below thresholds

tech-spec.md
Architecture:
- FastAPI app with /health and /v1/answer
- Vectorstore FAISS loading data/knowledge/*.md
- LangSmith client for tracing
- Offline eval with JSONL datasets + criteria groundedness/correctness

tasks.md
- Create FastAPI structure + health route
- Implement FAISS vectorstore
- Implement rag_pipeline with LangSmith tracing
- Create /v1/answer
- Add dataset JSONL in evals/datasets
- Implement eval_offline script
- Write tests
- Add Makefile, CI workflow, Dockerfile

------------------------------------------------------------
## 5. Cursor Commands

/create-spec "New feature X"
/execute-tasks
/refactor-spec "Improve retrieval"
/review-adr

------------------------------------------------------------
## 6. Daily Workflow

1. Create or adjust spec
2. Execute tasks with /execute-tasks
3. Run locally:
   make lint
   make type
   make test
   make eval-offline
4. Check report in evals/reports/DATE_run/
5. Commit + push → CI runs gates

------------------------------------------------------------
## 7. Dataset Example (evals/datasets/rag_iso_eval_v1.jsonl)

{"q": "What is ISO 42001?", "reference": "ISO 42001 is the AI management system standard."}
{"q": "What are mandatory policies?", "reference": "Risk management, monitoring, CAPA logs."}
{"q": "How does LangSmith help compliance?", "reference": "Provides audit-ready traces and eval repo

------------------------------------------------------------
END OF DOCUMENT
```