agentic-research-assistant/

│

├── README.md # Présentation projet (but, archi, usage, roadmap)

├── LICENSE # Licence open-source (MIT ou Apache 2.0 conseillé)

├── requirements.txt # Dépendances Python

├── docker-compose.yml # Infra locale (Postgres, pgvector, Redis)

├── .gitignore

│

├── src/ # Code source

│ ├── \_\_init\_\_.py

│ ├── main.py # Point d'entrée FastAPI

│ ├── agents/

│ │ ├── planner.py # Agent chef (ReAct / Reflexion)

│ │ ├── collector\_papers.py

│ │ ├── collector\_news.py

│ │ ├── collector\_patents.py

│ │ ├── writer.py # Génération rapport (templates Jinja)

│ │ └── critic.py # Vérification factualité

│ ├── core/

│ │ ├── rag.py # RAG + embeddings

│ │ ├── ranking.py # Ranking qualité

│ │ ├── db.py # Connexion Postgres + pgvector

│ │ ├── logger.py # Logs structurés (OpenTelemetry)

│ │ └── config.py # Variables d’env, constantes

│ └── front/

│ └── api.py # Routes FastAPI (REST/GraphQL)

│

├── notebooks/ # Expérimentations & éval académique

│ ├── evaluation.ipynb # Métriques (NDCG, factualité, etc.)

│ ├── ablations.ipynb # Tests sans critic, sans ranking, etc.

│ └── demo\_report.ipynb # Exemple génération d’un rapport

│

├── data/

│ ├── raw/ # Sources brutes (JSON, RSS, exports API)

│ ├── processed/ # Chunks, embeddings, scores

│ └── eval/ # Dataset d’évaluation (requêtes + jugements)

│

├── reports/

│ ├── samples/ # Exemples PDF/Markdown générés

│ └── dashboards/ # Dashboards observabilité (Grafana JSON)

│

└── docs/

├── architecture.png # Schéma archi (mermaid ou draw.io)

├── roadmap.md # Plan de dev détaillé

└── api\_docs.md # Spécifications API