Project Roadmap: Intelligent AML Assistant

PHASE 1: Data Collection & Setup

- Define Use Cases & Scenarios
- Collect Sample Data (transactions, KYC, regulations, sanctions)
- Prepare config.yaml (thresholds, API keys, model paths)
- Install dependencies via requirements.txt

PHASE 2: ML-Based Transaction Analysis

- Explore & clean data (01_data_exploration.ipynb)
- Build anomaly detection model (Isolation Forest/Autoencoder)
- Train risk classifier (e.g., XGBoost)
- Add explainability (SHAP/LIME)

PHASE 3: LLM-Powered SAR Generation

- Prepare case data format (JSON)
- Design and test LLM prompts (sar_generator.py)
- Generate SAR narratives with OpenAI/DeepSeek LLMs

PHASE 4: KYC/EDD Automation

- Parse and clean KYC docs (preprocess_kyc.py)
- Use LLM to validate KYC (kyc_validator.py)
- Summarize EDD and extract risk signals

PHASE 5: RAG Assistant for AML Research

- Ingest AML regulation corpus
- Create vector index (FAISS/Chroma)
- Build RAG pipeline (rag_research_assistant.py)
- Query AML policies via RAG

PHASE 6: Real-Time Monitoring (Optional Advanced)

Project Roadmap: Intelligent AML Assistant

- Simulate real-time transaction stream
- Apply real-time scoring logic
- Push suspicious alerts to queue/dashboard

PHASE 7: Dashboard & Review System

- Build Streamlit dashboard (dashboard.py)
- Create SAR submission/review UI
- Optional: expose FastAPI endpoints

PHASE 8: Testing, Docs, and Demo

- Add unit tests for key components
- Include sample inputs/outputs (demo/)
- Generate final report and visuals (reports/)
- Optional: record screencast video

OPTIONAL ENHANCEMENTS

- Fine-tune LLMs for SAR tasks
- Use Neo4j for entity graphs
- Integrate adverse media APIs
- Add active learning retraining loop