Enterprise Agent Architecture Notes

Day 7 – Tool Registry + Routing Engine

- Implemented BaseTool contract to standardize all tools.
- Created ToolRegistry to manage tools dynamically by name.
- Built IntelligentRouter to control tool execution flow.
- Added confidence-based fallback (RAG → Web Search).
- Integrated RetryPolicy and TimeoutExecutor for reliability.
- Created ReliableExecutor to combine retry + timeout handling.
- Added structured logging for request tracking and debugging.
- Separated planner logic from execution logic.

Key Concepts to Master from Day 7:

- Registry Pattern – allows dynamic tool management.
- Abstraction via Base Classes – ensures contract consistency.
- Separation of Concerns – planner, router, executor are independent layers.
- Deterministic Execution – no random tool calls.
- Confidence Threshold Logic – fallback when similarity is low.
- Reliability Engineering – retries and timeouts prevent system failure.
- Structured Responses – all tools return status, data, metadata.

Day 12 – FastAPI Deployment Layer

- Created FastAPI application for API deployment.
- Added /health endpoint for service monitoring.
- Implemented API key authentication middleware.
- Configured CORS for frontend compatibility.
- Added request size validation.
- Implemented structured response schema using Pydantic.
- Centralized environment configuration (.env + settings layer).
- Connected backend agent system to API endpoints.

Key Concepts to Master from Day 12:

- API Layer vs Core Logic Separation.
- Dependency Injection pattern.
- Authentication via headers (x-api-key).
- Environment-based configuration for security.
- Structured request/response contracts.
- Health checks for DevOps monitoring.
- Using Postman for API testing workflows.

What You Have Built So Far

- Hybrid AI system (RAG + Web fallback).
- Enterprise execution control system.
- Reliable and fault-tolerant tool execution.
- Production-ready API backend.
- Secure and testable deployment layer.
- Scalable architecture foundation.