TOC : Agentic ROADMAP

Chapter 1: Foundations of Large Language Models

Goal: Build strong conceptual clarity before coding.

Topics

- What are Large Language Models (LLMs)
- Evolution of NLP and why transformers changed everything
- Tokens, embeddings, attention, and context windows
- High-level transformer architecture (intuitive, not math-heavy)
- Training lifecycle: pretraining, fine-tuning, alignment
- Open-source vs closed-source LLM ecosystems
- Strengths, limitations, and real-world constraints of LLMs

Practical

- Model comparison walkthrough
- Understanding outputs vs training data behavior

Chapter 2: Prompting Fundamentals

Goal: Learn how to *control* LLM behavior.

Topics

- Why prompting works
- Prompt anatomy: instruction, context, examples, constraints
- Determinism vs creativity
- Output formatting and reliability
- Prompt failure modes & hallucinations

Practical

- Writing prompts for:
    - Summarization
    - Structured JSON outputs
    - Domain-specific answers

Chapter 3: Advanced Prompt Engineering

Goal: Extract consistent, high-quality outputs.

Topics

- Zero-shot, one-shot, few-shot prompting
- Chain-of-thought & reasoning prompts
- Role-based prompting
- Constraint-based prompting
- Prompt optimization patterns
- Prompt versioning strategies

Practical

- Prompt experiments & evaluations
- Building reusable prompt libraries

Chapter 4: Working with Open-Source LLMs

Goal: Gain independence from paid APIs.

Topics

- Open-source LLM landscape
- Model selection strategies
- Tokenizers & inference pipelines
- Local vs hosted inference
- Performance and cost trade-offs

Practical

- Run inference using open-source LLMs
- Build a simple text-generation service
- Compare open-source vs API-based outputs

Chapter 5: LangChain & Backend Foundations

Goal: Move from scripts to systems.

Topics

- Why orchestration frameworks are needed
- LangChain core abstractions
- Prompt templates & dynamic prompts
- Chains and sequential execution
- Introduction to FastAPI
- Designing LLM-powered APIs

Practical

- Create your first LLM-backed FastAPI service
- Dynamic prompt injection via API inputs

Chapter 6: LLM API Integrations

Goal: Work confidently with hosted LLMs.

Topics

- Sending prompts & receiving responses
- Parameter tuning (temperature, top-k, top-p)
- Streaming responses
- Error handling & retries
- Rate limits & cost optimization

Practical

- Integrate Gemini-like APIs
- Build a chat-style backend
- Add logging & observability

Chapter 7: Retrieval Augmented Generation (RAG) – Foundations

Goal: Solve hallucinations & knowledge limitations.

Topics

- Why RAG is required
- RAG architecture overview
- Embeddings & vector databases
- Chunking strategies
- Query → retrieval → generation flow

Practical

- Build a basic RAG pipeline
- Connect documents to LLMs
- Ask context-aware questions

Chapter 8: RAG Variants & System Design

Goal: Design scalable RAG architectures.

Topics

- Naive RAG vs advanced RAG
- Single-hop vs multi-hop retrieval
- Hybrid search (vector + keyword)
- Metadata filtering
- Context window optimization

Practical

- Improve retrieval relevance
- Design RAG schemas for large datasets

Chapter 9: Advanced RAG & Evaluation

Goal: Make RAG systems production-grade.

Topics

- Query enhancement & query rewriting
- Re-ranking techniques

- LLMs as judges
- RAG evaluation metrics
- Offline vs online evaluation
- Reducing hallucinations systematically

Practical

- Add re-ranking to RAG
- Evaluate RAG quality
- Build a **solid RAG project**

Chapter 10: Introduction to Agents

Goal: Understand autonomy in AI systems.

Topics

- What are agents
- LLMs vs agents
- Tools & function calling
- Planning, reasoning, acting loops
- Agent failure modes

Practical

- Build a basic tool-using agent
- Compare static vs agentic workflows

Chapter 11: Agent Architectures & LangGraph

Goal: Design structured agent systems.

Topics

- Famous agent architectures (theory)
- Reactive vs planning agents
- Workflow systems vs agents
- LangGraph fundamentals
- State, memory & context handling

Practical

- Convert RAG into an agentic system
- Add memory to conversations

Chapter 12: Multi-Agent Systems

Goal: Scale intelligence via collaboration.

Topics

- Multi-agent architecture concepts
- When to use multi-agent systems
- Communication patterns
- Coordination strategies
- Pros, cons & limitations

Practical

- Design a multi-agent workflow
- Debug agent interactions

Chapter 13: Capstone Project – Multi-Agent AI System

Goal: Demonstrate real-world readiness.

Project Scope

- Multi-agent system with:
    - Planner agent
    - Retriever agent
    - Reasoning agent
    - Validator / Judge agent
- Tool usage & memory
- RAG + agents combined

Deliverables

- Architecture diagram

- Working codebase
- API endpoints
- Evaluation report
- Deployment-ready structure