## **Generative AI System Design – Architect-Level Syllabus**

(FAANG-grade, Production + Interview Ready)

### **PHASE 1 – Foundations of GenAI System Design**

### **(Lay the base, align with architecture thinking)**

1. Evolution of System Design in AI  
   * From traditional distributed systems → ML systems → LLM-powered architectures
   * Why GenAI needs new patterns (dynamic orchestration, multi-modal, tool-augmented)
2. Core Principles of System Design  
   * Scalability, reliability, availability, consistency, latency trade-offs
   * How these principles shift with AI workloads
3. GenAI-Specific Constraints  
   * Model latency & throughput
   * Token limits & context window design
   * Data privacy & compliance (GDPR, HIPAA, DPDP)
4. HLD vs LLD in GenAI  
   * Mapping each to AI lifecycle
   * Role of architecture diagrams, data flow maps, sequence diagrams

### **PHASE 2 – High-Level Design (HLD) for GenAI**

1. Component Layering in GenAI Systems  
   * Presentation, Orchestration, LLM, Data, Infra
2. Core GenAI Architectural Patterns  
   * RAG (basic, advanced, hybrid)
   * Agentic orchestration (LangChain, LangGraph, AutoGen, CrewAI)
   * Multi-modal pipelines
3. Integration Patterns  
   * API gateway, microservices, serverless patterns
   * Multi-LLM routing & fallback
4. Security, Compliance, and Guardrails  
   * Data encryption, access control
   * Hallucination prevention
   * Moderation layers

### **PHASE 3 – Low-Level Design (LLD) for GenAI**

1. Data Ingestion & Preprocessing  
   * Loaders, chunking, cleaning, enrichment
2. Embedding & Vector Store Layer  
   * Embedding selection, dimensionality
   * Pinecone, Qdrant, Milvus, Weaviate
   * Hybrid search (vector + BM25 + metadata filtering)
3. Prompt Orchestration Layer  
   * Prompt templates, chains, LCEL
   * Dynamic prompt assembly
4. Agent Patterns  
   * ReAct, Plan-and-Execute, Toolformer, Multi-agent negotiation
   * State management (LangGraph AgentState)
5. Validation & Structured Outputs  
   * Pydantic v2, JSON Schema enforcement
   * Field validators, before/after transforms
6. Evaluation Layer  
   * RAGAS, TruLens, human feedback loops
7. Observability Layer  
   * Logging, tracing, monitoring (LangSmith, OpenTelemetry)
   * Token/cost tracking

### **PHASE 4 – Production-Grade GenAI Architectures**

1. FAANG-Grade RAG Pipelines  
   * Modular, service-oriented RAG
   * Domain-specific optimizations
2. Enterprise-Ready Agent Systems  
   * Tool registry, execution sandboxing, HITL integration
3. Scaling & Deployment  
   * GPU inference endpoints (vLLM, TensorRT-LLM)
   * AWS/GCP/Azure patterns
   * CI/CD for LLMOps
4. Caching & Performance  
   * Prompt caching, embedding caching
   * Latency optimization strategies

### **PHASE 5 – Specialized GenAI System Designs**

1. Multi-Modal System Design  
   * Vision-Language models, speech pipelines, document parsing
2. Domain-Specific GenAI  
   * Legal, medical, financial, e-commerce architectures
3. Compliance-Aware Architectures  
   * LexiGuard-style governance layers
   * Multi-jurisdiction compliance mapping

### **PHASE 6 – Future-Readiness & Advanced Patterns**

1. Self-Improving Systems  
   * Auto-evaluation → auto-finetuning loops
   * Continuous prompt optimization (DSPy, PromptLayer)
2. On-Device & Edge GenAI  
   * Running LLMs locally (GGUF, Ollama, MLX)
3. Hybrid Human+AI Systems  
   * Human-in-the-loop at design, inference, and feedback stages
4. GenAI System Design Anti-Patterns  
   * What to avoid in architecture and why

### **Deliverables for Each Module**

For every module above, you’ll prepare:

* Theory Notes: 10–30 pages each, FAANG-grade
* Interview QnA: 15–30 scenario-based questions
* Diagrams: HLD & LLD visualizations
* Mini-Project / Pattern Demo: Minimal but functional code artifact
* Best Practices & Pitfalls: Production lessons learned

### **Why This Works for You**

* Covers breadth & depth — aligns with your Principal Architect target level.
* Blends interview prep + real-world readiness — every topic has both design theory & code pattern.
* Matches your projects — You can cross-link with your LexiGuard, Travel Planner, and Document Portal builds for portfolio-ready case studies.
* Future-proof — Includes emerging patterns (multi-modal, self-improving, compliance-focused).