Google-GEMINI GenAI Engineer Roadmap

PHASE 1: THE DATA FOUNDATION (Day 1–12)

Fast-track the math and logic behind the models.

- **Day 1:** AI Ecosystem 2026: Reasoning Models (o1) vs. Legacy LLMs. Setup: Dev Containers & Poetry.
- **Day 2:** NumPy: Matrix math for Transformers (Dot products & Softmax intuition).
- **Day 3:** Pandas: Handling unstructured data (JSONL, Parquet) for LLM training.
- **Day 4:** EDA for GenAI: Detecting "Data Poisoning" and bias in training sets.
- **Day 5:** Probability for AI: Why LLMs are "Stochastic Parrots" (Temperature & Top-P).
- **Day 6:** ML Overview: When to use XGBoost vs. when to use an LLM (Cost vs. Accuracy).
- **Day 7:** Scikit-learn: Building a "Router" to classify intent before calling an API.
- **Day 8:** Feature Engineering: Text-to-Features (Advanced Tokenization).
- **Day 9:** Evaluation: Precision/Recall vs. Perplexity.
- **Day 10:** Hyperparameter Tuning: Optimization for small models.
- **Day 11:** Explainability: Why did the model hallucinate? (Attention Maps).
- **Day 12 – MINI PROJECT**
  - **Project 1: Intelligent Lead Scraper & Classifier**
  - **Resume Line:** Engineered a data pipeline that uses ML to classify 10k+ leads with 92% accuracy before LLM processing.

🔹 PHASE 2: NLP TO AGENTIC THINKING (Day 13–25)

Moving from "Text In" to "Intelligence Out."

- **Day 13:** The Tokenization Crisis: Costs, limits, and Tiktoken.
- **Day 14:** Modern Embeddings: Late Interaction models (ColBERT) vs. Dense vectors.
- **Day 15:** Sentiment Analysis 2.0: Detecting nuance and sarcasm.
- **Day 16:** Vector Theory: Cosine Similarity vs. Euclidean Distance.
- **Day 17:** The Transformer: Self-Attention vs. Cross-Attention.
- **Day 18:** PyTorch: Tensors and GPU memory management (VRAM 101).

- **Day 19:** Building a Neural Network for text classification from scratch.
- **Day 20:** Attention is All You Need: Deep dive into the 2017 paper.
- **Day 21:** The Rise of SLMs: Why Llama 3/Phi-4 are beating GPT-4 in niche tasks.
- **Day 22:** Hugging Face: Using the Model Hub and Inference Endpoints.
- **Day 23:** Quantization: Running 70B models on consumer hardware (GGUF/EXL2).
- **Day 24:** Model Merging: Combining specialized models.
- **Day 25 – MINI PROJECT**
  - **Project 2: Local Privacy-First Support Bot**
  - **Resume Line:** Deployed a quantized SLM (Llama 3.2) locally, reducing API costs by 100% for sensitive data.

◆ PHASE 3: GENAI ARCHITECTURE & RAG (Day 26–55)

The meat of the plan. Focus on "Agentic RAG."

- **Day 26:** Reasoning Models: Chain-of-Thought (CoT) and "Hidden" scratchpads.
- **Day 27:** Prompt Engineering: Few-shot, System Role optimization.
- **Day 28:** Advanced Patterns: Skeleton-of-Thought and Tree-of-Thoughts.
- **Day 29:** LLM APIs: Rate limits, Batching, and Asynchronous calls.
- **Day 30:** Chatbot Memory: Window buffer vs. Summary memory.
- **Day 31:** LangChain vs. LlamaIndex: Choosing the right tool.
- **Day 32:** LangGraph: Building stateful, multi-turn AI workflows.
- **Day 33:** Vector DBs: Pinecone, Weaviate, and pgvector (SQL + Vectors).
- **Day 34:** Hybrid Search: Combining Keyword (BM25) with Semantic search.
- **Day 35:** RAG Architecture: The "Naïve RAG" vs. "Advanced RAG" gap.
- **Day 36:** Query Transformation: Re-writing user questions for better retrieval.
- **Day 37:** Chunking 2.0: Semantic chunking and Recursive Character splitting.
- **Day 38:** RAG Evaluation: The RAGAS Framework (Faithfulness, Answer Relevance).
- **Day 39:** Reducing Hallucinations: Self-Correction loops.
- **Day 40 – MAJOR PROJECT**
  - **Project 3: "Agentic" Enterprise Knowledge Base**
  - **Resume Line:** Architected a RAG system with a self-correction loop, reducing hallucinations by 40% using RAGAS.

- **Day 41:** Fine-Tuning: PEFT and LoRA (When to actually do it).
- **Day 42:** Tool Calling: Giving the LLM a calculator and a browser.
- **Day 43:** AI Agents: The ReAct framework.
- **Day 44:** Multi-Agent Orchestration: CrewAI and Autogen.
- **Day 45:** Build a "Researcher & Writer" Agent team.
- **Day 46:** Long-term Memory: Using Mem0 or Zep for agents.
- **Day 47:** API Integration: Letting agents talk to Slack/Jira/GitHub.
- **Day 48:** Guardrails: NeMo Guardrails (Preventing prompt injections).
- **Day 49:** LLM-as-a-Judge: Automated grading of AI responses.
- **Day 50:** Observability: LangSmith and Arize Phoenix.
- **Day 51–55 – MAJOR PROJECT**
  - **Project 4: Autonomous "DevOps" Agent**
  - **Resume Line:** Built a multi-agent system that monitors GitHub issues and automatically drafts PRs with 70% accuracy.

◆ PHASE 4: PRODUCTION & SCALE (Day 56–90)

Turning a script into a business-grade product.

- **Day 56:** FastAPI: High-performance streaming responses (SSE).
- **Day 57:** Scalable Backends: Redis for caching LLM responses.
- **Day 58:** Security: Handling API Keys and PII Redaction.
- **Day 59:** Dockerizing AI: Handling massive CUDA dependencies.
- **Day 60:** Cloud Deployment: AWS Bedrock vs. GCP Vertex AI.
- **Day 61:** Cost Optimization: Token counting and tiered routing.
- **Day 62:** Scaling: Load balancing multiple LLM providers.
- **Day 63:** Monitoring: Alerting on "Drift" and high latency.
- **Day 64:** Compliance: GDPR and AI Act basics.
- **Day 65 – CAPSTONE**
  - **Project 5: AI "Customer Engineer" Agent**
  - **Features:** Real-time tool use, RAG, Human-in-the-loop triggers, and full observability dashboard.
- **Day 66–90:** Job Hunt Mode (Portfolio, System Design interviews, and Networking).