100x Applied GenAI Curriculum Overview

This document provides a structured breakdown of the applied GenAI curriculum across four modules. Each module is designed to progressively build applied skills through theory, hands-on labs, and real-world use cases.

Module 1: Diffusion Models

Objective: Build a foundational and applied understanding of diffusion models, including how they work mathematically and how to implement, train, and deploy them for generative tasks.

Week 1–2: Introduction to Diffusion

- Understanding what diffusion models are and how they compare with GANs and VAEs
- Intuition behind the forward (noise addition) and reverse (denoising) processes
- Mathematical formulation of DDPMs (Denoising Diffusion Probabilistic Models)
- Visualizing noise schedules and variance
- Hands-on coding: implementing simple forward and reverse processes in Python

Week 3: Conditional Diffusion

- Conditioning techniques: class labels, embeddings, and prompt-based control
- Understanding classifier-free guidance
- Case studies: Imagen, GLIDE, DALL-E 2
- Applications in text-to-image synthesis

Week 4–5: Implementing Diffusion from Scratch

- Using PyTorch to build a diffusion model from the ground up
- Working with datasets like CelebA, MNIST, and CIFAR

- Customizing the training loop and adding prompt conditioning
- Troubleshooting and stabilizing training

Week 6: Advanced Architectures and Real-World Applications

- U-Net and Transformer integrations for better image fidelity
- Latent Diffusion Models (LDMs) and architecture behind Stable Diffusion
- CLIP and text encoder use for prompt-based generation
- Image inpainting, editing, and control techniques
- Fine-tuning Stable Diffusion on custom datasets

Module 2: Full-Stack GenAI Development

Objective: Equip learners with the skills to develop, deploy, and maintain GenAI-powered applications using full-stack engineering practices.

Week 1–2: Backend Foundations

- Introduction to FastAPI for building AI APIs
- API routing, request/response handling, and middleware setup
- Task queues and asynchronous job management with Celery and Redis
- Deploying REST APIs using Render, Railway, or Vercel
- Best practices for logging, testing, and securing AI endpoints

Week 3: Frontend and Interface Design

- Rapid prototyping using no-code tools like Bubble.io and BuildShip
- Building user-friendly interfaces for interacting with GenAI models
- Connecting frontends to backends using HTTP requests

- Embedding AI functionality in buttons, forms, and UIs
- Use of tools like Superblocks for internal tool building

Week 4–5: Database and Vector Store Integration

- Setting up PostgreSQL databases for stateful AI apps
- Supabase for authentication, database management, and hosting
- Introduction to vector databases: Chroma, Pinecone, Weaviate
- Implementing semantic and similarity-based search using vector embeddings

Week 6: End-to-End Application Build

- Designing and developing a full-stack AI product
- Integration of frontend, backend, database, and vector search
- Live deployment of AI apps
- Debugging tools and optimization techniques

Module 3: Large Language Models (LLMs)

Objective: Master prompt engineering, tool use, fine-tuning, and orchestration of LLMs to solve real-world tasks efficiently and at scale.

Week 1–2: Prompt Engineering and Best Practices

- Techniques: zero-shot, few-shot, and chain-of-thought prompting
- Role prompting and use of system messages
- Handling hallucinations and steering model behavior
- Creating reusable prompt templates for various tasks

Week 3–4: Function Calling and Tool Augmentation

- Using OpenAI's function calling features to trigger external tools
- Building tool-augmented agents using LangChain and LlamaIndex
- Implementing the ReAct framework (Reasoning + Acting)
- Composing workflows with tool routing, tool selection, and response synthesis

Week 5: Fine-Tuning and Evaluation

- Fine-tuning versus instruction tuning: when to use which
- Preparing custom datasets using formats like Alpaca and Dolly
- Techniques: LoRA, QLoRA, and PEFT for parameter-efficient fine-tuning
- Evaluation metrics: BLEU, ROUGE, and GPT-based assessment
- Practical labs: fine-tuning open-source models like Mistral or LLaMA3

Week 6: Retrieval-Augmented Generation (RAG)

- Extending context through retrieval mechanisms
- Chunking and embedding strategies for documents
- RAG pipelines with retrievers and generators
- Using OpenAI Assistants API, LangChain, and LlamaIndex for custom RAG implementations

Module 4: AI Agents

Objective: Learn to build and deploy intelligent agents that can reason, plan, and interact with external systems and APIs in dynamic environments.

Week 1–2: Foundations of Agents

- Understanding agents vs. standard LLM applications
- Basic agent structures: tool use, memory, and autonomy
- Comparison of different agent frameworks: ReAct, AutoGPT, BabyAGI

- Implementing a simple agent with a single tool loop

Week 3: Memory and Planning

- Integrating memory into agents: vector-based memory stores and session memory
- Planning long-term tasks using language models
- Agents that decompose and execute subtasks
- Frameworks: LangGraph, AutoGen, CrewAI

Week 4–5: Multi-Agent Systems

- Designing agents that collaborate and communicate
- Graph-based agent planning and execution with LangGraph DAGs
- Use cases: collaborative research, workflow automation, technical writing
- Challenges: task delegation, data handoff, fallback behavior

Week 6: Deployment and Monitoring

- Adding safeguards and guardrails: content moderation and validation
- Logging, tracing, and performance metrics using LangSmith
- Containerization and scaling using serverless tools
- Deploying agents for real-world applications

Capstone Project

In the final phase, learners select one of the four tracks — Diffusion Models, Full Stack, LLMs, or AI Agents — to build a production-ready project.

- Scoping and ideating a real-world problem
- Architecting the solution using techniques from the chosen module
- Iterative prototyping with feedback from mentors

- Final demo with live deployment and technical documentation