GenAI Engineer Learning Roadmap

# Phase 1: AI & Deep Learning Foundations (Weeks 1–4)

## Week 1–2: Python + ML Basics

- What to Learn:  
 • Python (functions, classes, NumPy, Pandas)  
 • Scikit-learn (linear/logistic regression, decision trees, clustering)

- Why: Core ML logic is essential to understanding LLMs.

- Tools: Jupyter, Google Colab, Scikit-learn

- Project: Predict house prices / customer churn

## Week 3–4: Deep Learning Essentials

- What to Learn:  
 • Neural Networks, CNNs, RNNs, LSTM  
 • Overfitting, dropout, optimizers

- Why: All GenAI models are deep learning-based.

- Tools: TensorFlow/Keras or PyTorch

- Project: MNIST image classification, sentiment analysis on tweets

# Phase 2: NLP & Transformers (Weeks 5–8)

## Week 5–6: NLP Fundamentals

- What to Learn:  
 • Tokenization, lemmatization, POS, NER  
 • Word embeddings (Word2Vec, GloVe, FastText)

- Why: Understanding text as data is crucial for GenAI.

- Tools: SpaCy, NLTK, gensim

- Project: NER tagger for medical terms

## Week 7–8: Transformers & LLMs

- What to Learn:  
 • Transformer architecture (self-attention, positional encoding)  
 • BERT, GPT-2/3, T5 models  
 • Hugging Face Transformers basics

- Why: All LLMs use this architecture.

- Tools: Hugging Face, 🤗 Datasets, Tokenizers

- Project: Fine-tune BERT for sentiment analysis or T5 for summarization

# Phase 3: GenAI Specialization (Weeks 9–14)

## Week 9–10: Prompt Engineering + Embeddings

- What to Learn:  
 • Zero-shot, few-shot, chain-of-thought prompts  
 • Text embeddings (OpenAI, BERT-based, SentenceTransformers)

- Why: Better prompts = better LLM performance; embeddings = search & RAG

- Tools: OpenAI API, Hugging Face, sentence-transformers

- Project: Prompt-tuned chatbot; semantic search engine for product reviews

## Week 11–12: Retrieval-Augmented Generation (RAG)

- What to Learn:  
 • What is RAG: retriever + generator  
 • Chunking, vector search, context injection  
 • LangChain and LlamaIndex pipelines

- Why: RAG makes LLMs domain-aware and scalable

- Tools: FAISS, Pinecone, LangChain, LlamaIndex

- Project: Q&A bot for your course PDFs or GitHub README files

## Week 13–14: Fine-tuning LLMs

- What to Learn:  
 • LoRA, PEFT, full fine-tuning  
 • Dataset formats (JSONL, DPO, SFT)

- Why: Custom behavior, privacy, and performance

- Tools: Hugging Face Trainer, PEFT library, Transformers

- Project: Fine-tune LLaMA-2/T5 on domain-specific data (e.g., medical questions)

# Phase 4: Agents, MLOps & Deployment (Weeks 15–20)

## Week 15–16: LLM Agents + Tools

- What to Learn:  
 • LangChain agents (ReAct, tool use)  
 • Memory and state management

- Why: Agents make LLMs interactive and goal-driven

- Tools: LangChain, OpenAI tools

- Project: GPT agent that searches Google + answers with citations

## Week 17–18: GenAI App Development

- What to Learn:  
 • Streamlit/Gradio UI  
 • FastAPI + LangChain backend

- Why: Let others interact with your GenAI models

- Tools: Streamlit, Gradio, FastAPI

- Project: YouTube channel chatbot (RAG pipeline + Gradio)

## Week 19–20: Deployment & Monitoring

- What to Learn:  
 • Docker, GitHub Actions, Inference Endpoints  
 • Model monitoring (latency, usage, feedback loop)

- Why: Take your GenAI solution to production

- Tools: Docker, Hugging Face Hub, AWS/GCP, Weights & Biases

- Project: Deploy your GenAI app on Hugging Face Spaces or AWS