Generative AI for Beginners: Fundamentals and Practical Application

Curriculum Overview

Field	Description
Course Title	Generative AI for Beginners: Fundamentals and Practical Application
Target Audience	Individuals with basic technical literacy (e.g., scripting/programming knowledge) interested in understanding and utilizing Generative AI models.
Duration	6 Weeks (Approx. 60-80 hours of instruction and project work)
Prerequisites	Familiarity with basic computer science concepts and general comfort using web-based tools.
Focus	Bridging the gap between theoretical AI concepts and practical application via leading Large Language Models (LLMs) and diffusion models.

Learning Outcomes

Upon successful completion of this curriculum, students will be able to:

1. Define and Differentiate: Explain the core concepts of Generative AI, distinguishing it from traditional Machine Learning and Discriminative AI.
2. Master Interaction: Apply advanced Prompt Engineering techniques (e.g., Chain-of-Thought, few-shot prompting) to achieve predictable and high-quality outputs from LLMs.
3. Analyze Architecture: Describe the fundamental components of the Transformer architecture and the function of Large Language Models (LLMs), Embeddings, and tokenization.
4. Utilize Multimodality: Create and manipulate generated content across different modalities, including text, code, and images, using modern tools.

5. Implement Customization: Understand the difference between Fine-Tuning and Retrieval-Augmented Generation (RAG) and determine appropriate strategies for specific business needs.
6. Apply Ethics: Identify and mitigate key ethical risks, including bias, hallucination, and privacy concerns, related to the deployment of Generative AI systems.

Module Structure

Module 1: Generative AI Foundations and Taxonomy

Focus: Establishing the foundational understanding of what Generative AI is, how it works, and its place in the modern technological landscape.

Topic	Key Concepts Covered
1.1 Introduction to AI, ML, and GenAI	Defining Artificial Intelligence (AI), Machine Learning (ML), and specifically Generative AI. Use cases and examples.
1.2 Key Generative Models	Overview of major model types: LLMs (Text), GANs, VAEs, and Diffusion Models (Image). The concept of pre-training.
1.3 Terminology and Components	Understanding essential terms: Tokenization, parameters, dataset size, and the role of foundational models.
1.4 The Generative Workflow	High-level understanding of the input, inference, and output process in a typical generative system.

Module 2: Deep Dive into Large Language Models (LLMs)

Focus: Examining the core technology driving the current Generative AI wave—the Transformer architecture and Large Language Models.

Topic	Key Concepts Covered
2.1 The Transformer Architecture	High-level explanation of the self-attention mechanism, encoders, and decoders (without requiring deep mathematical understanding).

2.2 LLMs in Practice	Differences between common commercial models (e.g., GPT-4, Claude) and open-source alternatives (e.g., Llama).
2.3 Understanding Embeddings	Definition and function of Embeddings (numerical representations of data) and their role in semantic search and context understanding.
2.4 Limitations and Challenges	Analysis of model limitations, including Hallucination, bias leakage, latency, and computational costs.

Module 3: Multimodality and Content Generation

Focus: Moving beyond text to understand how Generative AI creates images, code, and other forms of media.

Topic	**Key Concepts Covered**
3.1 Image Generation Models	Introduction to Diffusion Models (e.g., Stable Diffusion) and their workflow (latent space, denoising).
3.2 Text-to-Image Prompting	Techniques for effective visual prompting, incorporating styles, negative prompts, and aspect ratios.
3.3 Generative AI for Code	Using models (e.g., GitHub Copilot) for code generation, completion, and debugging. Best practices for verification.
3.4 Data Modality Integration	The future of multimodal systems (integrating text, image, and audio input/output).

Module 4: Prompt Engineering Mastery

Focus: The essential skill for every beginner—optimizing input instructions to achieve precise and reliable outputs.

Topic	**Key Concepts Covered**

4.1 Foundations of Effective Prompting	Structure (Persona, Task, Context, Format), Clarity, and Iteration.
4.2 Advanced Prompting Techniques	Zero-shot, Few-shot learning, and Chain-of-Thought (CoT) prompting for complex reasoning tasks.
4.3 Tooling and Automation	Using prompt templates, variables, and basic Python libraries (e.g., OpenAI API calls) to manage prompts programmatically.
4.4 Adversarial Prompting and Guardrails	Understanding prompt injection risks and implementing basic defensive measures (e.g., system prompts, input validation).

Module 5: Customization and Retrieval Systems

Focus: Learning how to adapt foundational models to specific domain knowledge using modern architectural patterns.

Topic	**Key Concepts Covered**
5.1 RAG vs. Fine-Tuning	A detailed comparison of when to use Retrieval-Augmented Generation (RAG) for external knowledge vs. Fine-Tuning for behavioral change.
5.2 Introduction to RAG	Components of a RAG system: data ingestion, chunking, vector databases, and retrieval (high level).
5.3 Principles of Fine-Tuning	Understanding transfer learning, creating high-quality training datasets, and evaluating a fine-tuned model (e.g., supervised fine-tuning).
5.4 Basic Orchestration Frameworks	Introduction to the purpose of frameworks like LangChain or LlamaIndex for managing complex multi-step generative workflows.

Module 6: Deployment, Ethics, and Industry Applications

Focus: Reviewing real-world applications, ethical guidelines, and pathways to integrating Generative AI into existing systems.

Topic	Key Concepts Covered
6.1 Generative AI in the Industry	Case studies demonstrating value across sectors: Content creation, customer service (chatbots), pharmaceutical research, and finance.
6.2 Responsible AI and Governance	Deep dive into ethical considerations: data ownership, intellectual property rights, bias mitigation, and transparency.
6.3 Model Integration	Using REST APIs (e.g., accessing an LLM service via Azure or AWS) for application integration and managing latency.
6.4 Future Trends	Discussion of emerging topics: small language models (SLMs), synthetic data generation, and autonomous agents.

Projects

Students will complete two structured projects to solidify their practical skills.

Project 1: Advanced Prompt Engineering Portfolio (Modules 3 & 4)

Objective: Demonstrate mastery of advanced prompting techniques across different content modalities.

- Task A (Text Generation): Develop and test three distinct prompts (Zero-shot, Few-shot, and Chain-of-Thought) to solve a complex logical reasoning problem and analyze output performance.
- Task B (Multimodal Generation): Utilize a public image generation tool (or API) to create a series of 5 images based on a detailed prompt template. Document the prompt structure, negative prompts, and model settings used to achieve the desired outcome.
- Deliverable: A technical report documenting the prompts, system settings, resulting output quality, and a reflection on prompt optimization strategies.

Project 2: Knowledge Retrieval Sandbox (Module 5)

Objective: Implement a simplified version of a Retrieval-Augmented Generation (RAG) system using open-source tools to answer questions based *only* on a specific, non-public document.

- Task: Select a small PDF or text document (e.g., a company policy manual or a research paper). Implement a script (using simple Python libraries for vectorization) that converts the document content into embeddings, and then uses a local LLM or API to answer questions about the document.
- Deliverable: Functioning Python script demonstrating document ingestion and Q&A capability, along with a README explaining the architecture choices (chunking size, embedding model used).

Capstone Project: Domain-Specific AI Assistant

Objective: Design, build, and evaluate a minimum viable product (MVP) Generative AI application that solves a specific real-world problem or automates an industry workflow, leveraging skills from all modules.

Requirements:

1. Problem Definition: Clearly articulate the target user and the specific business problem the assistant solves (e.g., summarizing medical records, drafting legal briefs, triaging IT tickets).
2. Architecture: The solution must incorporate a modular design, likely utilizing RAG or fine-tuning concepts to provide domain-specific knowledge.
3. Interaction Layer: Develop a simple user interface (e.g., a basic web app via Streamlit or Gradio, or a command-line interface) for user interaction.
4. Ethical Analysis: Include a section detailing the potential risks of the solution (bias, misuse, hallucination) and the guardrails implemented to mitigate them.

Deliverable:

- A publicly accessible code repository (GitHub).
- A presentation/demo showcasing the application's functionality.
- A final technical documentation report detailing the LLM choice, prompt templates, customization method, and ethical review.