### 🎯 Objective:

Apply your GenAI knowledge in a real-world, production-grade project tailored to your domain of interest. Learn how to integrate GenAI models into full-stack systems, explore ongoing research trends, and publish meaningful, portfolio-worthy work.

## 🔍 Curriculum Breakdown

### 1. Domain-Specific Applications

#### 📌 1.1 Healthcare

* Medical document summarization & Q&A
* Chest X-ray or pathology image classification using CNNs or Vision Transformers
* Conversational agents for clinical guidance (with RAG + safety layers)

#### 📌 1.2 Finance

* Market summarization using LLMs
* Fraud detection using anomaly detection or ML classification
* Risk assessment using probabilistic modeling and embeddings

#### 📌 1.3 Education

* Personalized tutors powered by GenAI (context-aware Q&A + curriculum adaptation)
* AI-powered content creators for quizzes, flashcards, and feedback

#### 📌 1.4 Entertainment

* Story, plot, or script generation using fine-tuned LLMs
* AI music generation (AudioCraft, MusicGen)
* Game design assistants or NPC behavior generators

📌 Exercise: Select one vertical and ideate 2–3 project concepts using GenAI

### 2. Full-Stack GenAI Systems

#### 📌 2.1 System Design

* Combine UI (Streamlit, React) + API (FastAPI, Flask) + model backend (transformer, RAG, agent)
* Consider user interaction, context memory, personalization

#### 📌 2.2 Real-Time Interfaces

* Use Gradio or Streamlit for fast prototyping
* Create chatbots, co-pilots, editors with live inputs/outputs

#### 📌 2.3 Tool Integration

* Use LangChain/LlamaIndex for document querying
* Add plug-ins: calculators, file uploaders, speech-to-text

📌 Mini Project: Design a wireframe + architecture diagram for your GenAI system

### 3. MLOps in Production

#### 📌 3.1 Building Pipelines

* Automate data ingestion → preprocessing → model prediction → logging
* Use MLflow for versioning and model registry

#### 📌 3.2 Lifecycle Management

* Add logging, monitoring (W&B, Prometheus)
* Retraining triggers via data drift or feedback loops

#### 📌 3.3 Deployment Options

* Deploy via:
  + Cloud (AWS/GCP/Vertex)
  + Hugging Face Spaces
  + Local on-prem (Docker/Kubernetes)

📌 Activity: Write a YAML config for your model + deploy a test version using Docker or Streamlit Cloud

### 4. Research & Innovation

#### 📌 4.1 Staying Current

* Follow state-of-the-art models: Gemini, GPT-4V, Claude, LLaMA 3
* Track changes via model cards, benchmarks, and releases

#### 📌 4.2 Paper Exploration & Prototyping

* Use Papers with Code, arXiv, Semantic Scholar to review breakthroughs
* Reproduce key components of a published model (or reimplement with open weights)

📌 Challenge: Choose a recent paper and replicate a simplified version using open-source resources

### 5. Capstone Project Development

🎓 *Goal:* Deliver a fully functional, documented GenAI application or toolkit

#### 📌 Steps:

* Identify a real-world problem or domain-specific use case
* Perform domain research and define success criteria
* Design the app architecture (frontend, backend, model)
* Choose between a hosted LLM (e.g., OpenAI) or fine-tuned local model
* Build RAG, tool-use, or prompt chains as needed
* Deploy and test in a user-simulated or live environment
* Collect feedback and iterate

📌 Deliverables:

* GitHub repo with full codebase and README
* Live app/demo (HF Spaces, Streamlit, Docker URL)
* Optional: project video, technical blog, or presentation deck

### 6. Tools & Resources

#### ✅ Interfaces & Apps

* LangChain, LlamaIndex for document-based chat and tools
* Gradio, Streamlit for UIs

#### ✅ Deployment & Infra

* Docker, Kubernetes, MLflow, Hugging Face Spaces

#### ✅ Model Access

* Hugging Face, OpenAI, Cohere, Anthropic, Gemini APIs

#### ✅ Research Sources

* Papers with Code, arXiv, Semantic Scholar
* Newsletters: *The Batch (DeepLearning.ai)*, *Import AI*, *Substack ML papers*