Gen AI with POC – Advanced AI Solutions & Development

45-Session Comprehensive Syllabus

CodeYAA Network *Empowering AI Excellence*

Course Overview

Duration: 45 Sessions (1-3 hours each, Weekends only)
Prerequisites: Basic Python & AI/ML concepts
Focus: Advanced Generative AI Development & Production-Grade Solutions
Deliverables: 15 Hands-on POC Projects

Course Structure

Phase 1: AI Foundations & GenAI Fundamentals (Sessions 1-10)

Phase 2: LLMs, Prompt Engineering & Fine-tuning (Sessions 11-20)

Phase 3: RAG Systems & Vector Databases (Sessions 21-30)

Phase 4: Advanced AI Agents & Production Systems (Sessions 31-40)

Phase 5: Deployment, Scaling & Capstone Projects (Sessions 41-45)

Detailed Session Breakdown

PHASE 1: AI FOUNDATIONS & GENAI FUNDAMENTALS

Session 1: GenAI Landscape & Development Environment Setup

Topics Covered:

- Current GenAI ecosystem (2025 trends)
- Open-source vs Proprietary models comparison
- Development environment setup (Python, Jupyter, Colab)
- Hugging Face ecosystem overview

Hands-on Coding/Tools:

- Google Colab setup
- Hugging Face Transformers installation
- Git/GitHub for AI projects
- First API calls to open-source models

Mini-project:

- Environment validation & first model interaction

Session 2: Transformer Architecture Deep Dive

Topics Covered:

- Transformer architecture internals
- Attention mechanisms & self-attention
- Encoder-decoder vs decoder-only models
- Tokenization and embedding concepts

Hands-on Coding/Tools:

- Hugging Face Transformers
- Tokenizer exploration
- Attention visualization tools

Mini-project:

- Build custom tokenizer for domain-specific text

Session 3: Introduction to Large Language Models

Topics Covered:

- LLM categories (GPT, BERT, T5, etc.)
- Open-source alternatives (Llama, Mistral, Falcon)
- Model size vs performance trade-offs
- Computational requirements

Hands-on Coding/Tools:

- Hugging Face model hub exploration
- Loading different model sizes
- Performance benchmarking

Mini-project:

- Compare 3 different open-source LLMs on text generation tasks

---

Session 4: Text Generation & Sampling Strategies

Topics Covered:

- Generation strategies (greedy, beam search, sampling)
- Temperature, top-k, top-p parameters
- Controlling generation quality and creativity
- Bias and fairness in text generation

Hands-on Coding/Tools:

- Hugging Face generation utils
- Parameter tuning experiments
- Custom generation pipelines

POC 1: Creative Writing Assistant

- Build a creative writing tool with multiple generation modes
- Implement story continuation, poetry generation, and style transfer

---

Session 5: Google Gemini API Integration

Topics Covered:

- Gemini API setup and authentication
- Gemini Pro vs Ultra capabilities
- Multimodal features (text, image, code)
- Cost optimization strategies

Hands-on Coding/Tools:

- Google AI Studio
- Gemini Python SDK
- API key management
- Rate limiting and error handling

Mini-project:

- Build Gemini-powered Q&A system

Session 6: Prompt Engineering Fundamentals

Topics Covered:

- Prompt design principles
- Zero-shot, few-shot, and chain-of-thought prompting
- Prompt templates and variables
- Common prompt engineering pitfalls

Hands-on Coding/Tools:

- LangChain PromptTemplate
- Prompt optimization tools
- A/B testing prompts

POC 2: Intelligent Email Assistant

- Email classification, summarization, and response generation
- Multiple prompt strategies for different email types

Session 7: LangChain Framework Introduction

Topics Covered:

- LangChain architecture and components
- Chains, prompts, and memory concepts
- LLM integrations and abstractions
- Community ecosystem

Hands-on Coding/Tools:

- LangChain installation and setup
- Basic chain creation
- LLM wrapper implementations

Mini-project:

- Build sequential processing chain for data analysis

Session 8: Memory Systems in LangChain

Topics Covered:

- Conversation memory types
- Buffer, summary, and vector store memory
- Memory persistence strategies
- Context window management

Hands-on Coding/Tools:

- LangChain memory modules
- SQLite for conversation storage
- Memory optimization techniques

POC 3: Conversational Knowledge Assistant

- Multi-turn conversations with persistent memory
- Context-aware responses and conversation summarization

Session 9: Document Processing & Text Analytics

Topics Covered:

- Document loaders (PDF, DOCX, HTML, etc.)
- Text splitting strategies
- Metadata extraction and processing
- OCR integration for scanned documents

Hands-on Coding/Tools:

- LangChain document loaders
- PyPDF2, python-docx
- Text splitters and chunking strategies

Mini-project:

- Build document processing pipeline for multiple formats

Session 10: Embeddings & Semantic Search

Topics Covered:

- Word embeddings vs sentence embeddings
- Popular embedding models (Sentence-BERT, OpenAI embeddings)
- Similarity search algorithms
- Embedding fine-tuning concepts

Hands-on Coding/Tools:

- Sentence-Transformers library
- HuggingFace embedding models
- Cosine similarity implementations

POC 4: Semantic Document Search Engine

- Index large document collections
- Semantic search with ranking and filtering

PHASE 2: LLMS, PROMPT ENGINEERING & FINE-TUNING

Session 11: Advanced Prompt Engineering Techniques

Topics Covered:

- Chain-of-thought and tree-of-thought prompting
- Role-based prompting and persona development
- Prompt injection prevention

- Multi-step reasoning prompts

Hands-on Coding/Tools:

- Advanced LangChain prompt templates
- Prompt security testing tools
- Reasoning chain implementations

Mini-project:

- Build complex reasoning system for mathematical problem solving

---

Session 12: Fine-tuning Strategies & Techniques

Topics Covered:

- Full fine-tuning vs parameter-efficient methods
- LoRA (Low-Rank Adaptation) principles
- PEFT (Parameter Efficient Fine-Tuning)
- Dataset preparation for fine-tuning

Hands-on Coding/Tools:

- Hugging Face PEFT library
- LoRA implementation
- Training data preparation scripts

POC 5: Domain-Specific Chatbot with Fine-tuning

- Fine-tune open-source model for specific domain (legal, medical, etc.)
- Compare base model vs fine-tuned performance

Session 13: Model Quantization & Optimization

Topics Covered:

- Quantization techniques (INT8, INT4)
- Model pruning and distillation
- ONNX conversion and optimization

- Hardware-specific optimizations

Hands-on Coding/Tools:

- Hugging Face Optimum
- ONNX Runtime
- Quantization libraries

Mini-project:

- Optimize model for edge deployment with performance benchmarking

---

Session 14: Custom Training Loops & Datasets

Topics Covered:

- PyTorch training loops for transformers
- Custom dataset creation and validation
- Training monitoring and visualization
- Hyperparameter optimization

Hands-on Coding/Tools:

- PyTorch Lightning
- Weights & Biases for monitoring
- Custom dataset classes

Mini-project:

- Build custom training pipeline with comprehensive logging

Session 15: Instruction Tuning & RLHF Concepts

Topics Covered:

- Instruction following vs completion models
- Supervised fine-tuning (SFT)
- Reinforcement Learning from Human Feedback (RLHF)
- Direct Preference Optimization (DPO)

Hands-on Coding/Tools:

- Instruction dataset creation
- TRL (Transformer Reinforcement Learning)
- Preference data annotation tools

POC 6: Instruction-Tuned Assistant

- Create instruction-following model for specific task domain
- Implement feedback collection system

Session 16: Multimodal AI with Vision-Language Models

Topics Covered:

- Vision-Language model architectures
- Image captioning and VQA (Visual Question Answering)
- CLIP and similar models
- Multimodal prompt engineering

Hands-on Coding/Tools:

- Hugging Face vision models
- PIL/OpenCV for image processing
- Gradio for multimodal interfaces

Mini-project:

- Build image analysis and description system

Session 17: Code Generation & Programming Assistance

Topics Covered:

- Code generation models (CodeT5, StarCoder)
- Programming language understanding
- Code completion and debugging
- Security considerations in code generation

Hands-on Coding/Tools:

- Hugging Face Code models
- Code execution sandboxes
- AST (Abstract Syntax Tree) parsing

POC 7: AI Code Review Assistant

- Automated code review and improvement suggestions
- Multiple programming language support

Session 18: Audio Processing & Speech AI

Topics Covered:

- Speech-to-text and text-to-speech
- Audio embeddings and analysis
- Whisper model integration
- Voice cloning considerations

Hands-on Coding/Tools:

- OpenAI Whisper
- Audio processing libraries (librosa)
- Real-time audio streaming

Mini-project:

- Build voice-controlled AI assistant with transcription

Session 19: Model Evaluation & Benchmarking

Topics Covered:

- Evaluation metrics for generative models
- Human evaluation vs automated metrics
- Benchmark datasets and leaderboards
- A/B testing for AI systems

Hands-on Coding/Tools:

- BLEU, ROUGE, BERTScore metrics
- Custom evaluation frameworks
- Statistical significance testing

Mini-project:

- Comprehensive evaluation suite for text generation models

Session 20: AI Safety & Responsible Development

Topics Covered:

- Bias detection and mitigation
- Harmful content filtering
- AI governance frameworks
- Privacy-preserving techniques

Hands-on Coding/Tools:

- Bias detection toolkits
- Content moderation APIs
- Differential privacy libraries

POC 8: Ethical AI Content Moderator

- Build content moderation system with bias detection
- Implement fairness metrics and monitoring

PHASE 3: RAG SYSTEMS & VECTOR DATABASES

Session 21: Vector Databases Deep Dive

Topics Covered:

- Vector database architectures
- FAISS, ChromaDB, Pinecone, Weaviate comparison
- Indexing strategies and performance optimization

- Distributed vector search

Hands-on Coding/Tools:

- FAISS library setup
- ChromaDB integration
- Vector index optimization

Mini-project:

- Performance comparison of different vector databases

Session 22: Building RAG Systems - Architecture

Topics Covered:

- RAG system components and architecture
- Retrieval strategies and ranking
- Context length management
- RAG vs fine-tuning trade-offs

Hands-on Coding/Tools:

- LangChain RAG chains
- Custom retrieval implementations
- Context compression techniques

POC 9: Enterprise Document Q&A System

- Build RAG system for company knowledge base
- Support multiple document types and sources

Session 23: Advanced Retrieval Strategies

Topics Covered:

- Hybrid search (dense + sparse)
- Multi-vector retrieval
- Hierarchical retrieval systems

- Query expansion and rewriting

Hands-on Coding/Tools:

- BM25 + dense retrieval combination
- Multi-stage retrieval pipelines
- Query preprocessing tools

Mini-project:

- Implement hybrid search system with performance analysis

Session 24: RAG with Structured Data

Topics Covered:

- Text-to-SQL with RAG
- Knowledge graphs integration
- Structured data embeddings
- Multi-modal RAG systems

Hands-on Coding/Tools:

- SQLAlchemy for database connections
- Neo4j for knowledge graphs
- Structured data vectorization

POC 10: Business Intelligence RAG Assistant

- Query structured databases using natural language
- Generate insights from business data

Session 25: RAG Optimization & Performance Tuning

Topics Covered:

- Retrieval quality metrics
- Chunk size optimization
- Re-ranking strategies

- Caching and performance optimization

Hands-on Coding/Tools:

- RAG evaluation frameworks
- Re-ranking models
- Redis for caching

Mini-project:

- Optimize RAG system performance with comprehensive metrics

Session 26: Real-time RAG Systems

Topics Covered:

- Streaming RAG implementations
- Real-time data ingestion
- Incremental index updates
- Low-latency optimization

Hands-on Coding/Tools:

- FastAPI for real-time APIs
- Kafka for data streaming
- Async programming patterns

Mini-project:

- Build real-time news analysis RAG system

Session 27: Multi-Agent RAG Systems

Topics Covered:

- Agent-based RAG architectures
- Collaborative information retrieval
- Agent communication protocols
- Specialized agent roles

Hands-on Coding/Tools:

- LangChain agents framework
- Multi-agent orchestration
- Custom agent implementations

POC 11: Research Assistant Agent Network

- Multiple specialized agents for different research tasks
- Collaborative document analysis and synthesis

Session 28: RAG Security & Privacy

Topics Covered:

- Access control in RAG systems
- Data encryption and secure retrieval
- Privacy-preserving embeddings
- Audit trails and compliance

Hands-on Coding/Tools:

- JWT authentication
- Encryption libraries
- Access control implementations

Mini-project:

- Secure RAG system with role-based access control

Session 29: Cross-lingual & Multilingual RAG

Topics Covered:

- Multilingual embedding models
- Cross-lingual information retrieval
- Translation integration
- Cultural context considerations

Hands-on Coding/Tools:

- Multilingual BERT models
- Translation APIs
- Language detection tools

Mini-project:

- Build multilingual knowledge base with cross-lingual search

---

Session 30: RAG System Monitoring & Observability

Topics Covered:

- RAG system monitoring strategies
- Quality metrics tracking
- Performance dashboards
- Alerting and incident response

Hands-on Coding/Tools:

- Prometheus for metrics
- Grafana for dashboards
- Custom logging frameworks

POC 12: Production RAG System with Full Observability

- Complete RAG system with monitoring, alerting, and analytics
- Performance optimization based on real usage data

PHASE 4: ADVANCED AI AGENTS & PRODUCTION SYSTEMS

Session 31: AI Agents Architecture & Design Patterns

Topics Covered:

- Agent architectures (ReAct, Plan-and-Execute)
- Tool integration and function calling
- Agent memory and state management

- Multi-agent coordination patterns

Hands-on Coding/Tools:

- LangChain agents
- Custom tool implementations
- Agent state management systems

Mini-project:

- Build research agent with multiple tool integrations

Session 32: Function Calling & Tool Integration

Topics Covered:

- Function calling with open-source models
- Custom tool development
- API integration patterns
- Error handling in tool usage

Hands-on Coding/Tools:

- OpenAI function calling format
- Custom function schemas
- API wrapper development

Mini-project:

- Create agent with database query and web search capabilities

---

Session 33: Workflow Orchestration & Automation

Topics Covered:

- Workflow design patterns
- Task scheduling and queuing
- Error recovery and retry mechanisms
- Workflow monitoring and logging

Hands-on Coding/Tools:

- Celery for task queuing
- Apache Airflow basics
- Custom workflow engines

POC 13: Automated Business Process Agent

- Multi-step business process automation
- Integration with external systems and APIs

Session 34: Testing AI Systems

Topics Covered:

- Unit testing for AI components
- Integration testing strategies
- Property-based testing
- Regression testing for model updates

Hands-on Coding/Tools:

- pytest for AI systems
- Hypothesis library
- Custom test harnesses

Mini-project:

- Comprehensive test suite for AI application

Session 35: AI System Scaling & Performance

Topics Covered:

- Horizontal vs vertical scaling
- Load balancing strategies
- Caching layers and optimization
- Resource monitoring and auto-scaling

Hands-on Coding/Tools:

- Docker containerization
- Load balancers (nginx)
- Monitoring tools

Mini-project:

- Scale AI application for high-throughput scenarios

Session 36: Model Serving & Inference Optimization

Topics Covered:

- Model serving architectures
- Batch vs online inference
- Model versioning and A/B testing
- GPU optimization and batching

Hands-on Coding/Tools:

- TorchServe
- TensorRT optimization
- Custom serving solutions

Mini-project:

- Build high-performance model serving system

Session 37: CI/CD for AI Systems

Topics Covered:

- MLOps pipeline design
- Automated testing and validation
- Model versioning and rollback
- Deployment automation

Hands-on Coding/Tools:

- GitHub Actions
- MLflow for model tracking
- Docker for containerization

POC 14: Complete MLOps Pipeline

- End-to-end CI/CD for AI application
- Automated testing, validation, and deployment

Session 38: Edge AI & Mobile Deployment

Topics Covered:

- Model optimization for edge devices
- Mobile deployment strategies
- On-device inference considerations
- Privacy and security for edge AI

Hands-on Coding/Tools:

- ONNX Runtime Mobile
- Core ML for iOS
- TensorFlow Lite

Mini-project:

- Deploy AI model to mobile/edge environment

Session 39: AI Governance & Compliance

Topics Covered:

- AI governance frameworks
- Compliance requirements (GDPR, etc.)
- Model documentation and lineage
- Risk assessment and management

Hands-on Coding/Tools:

- Model documentation tools
- Compliance checking frameworks
- Risk assessment methodologies

Mini-project:

- Create comprehensive AI governance documentation

Session 40: Advanced Deployment Strategies

Topics Covered:

- Multi-cloud deployment
- Kubernetes for AI workloads
- Serverless AI functions
- Cost optimization strategies

Hands-on Coding/Tools:

- Kubernetes basics
- AWS Lambda/Google Cloud Functions
- Cost monitoring tools

Mini-project:

- Deploy AI system across multiple cloud platforms

PHASE 5: DEPLOYMENT, SCALING & CAPSTONE PROJECTS

Session 41: Production Architecture Design

Topics Covered:

- Enterprise AI architecture patterns
- Microservices for AI systems
- Data pipeline design
- Security architecture considerations

Hands-on Coding/Tools:

- Architecture diagram tools
- Microservice frameworks
- Security assessment tools

Mini-project:

- Design complete enterprise AI architecture

Session 42: Real-world Case Studies Analysis

Topics Covered:

- Finance: Fraud detection and risk assessment
- Healthcare: Clinical decision support systems
- Education: Personalized learning platforms
- Retail: Recommendation and pricing systems

Hands-on Coding/Tools:

- Case study analysis frameworks
- Industry-specific datasets
- Domain adaptation techniques

Mini-project:

- Analyze and propose improvements for real-world AI system

Session 43: Capstone Project Planning & Architecture

Topics Covered:

- Project scope definition
- Technical architecture design
- Team collaboration strategies
- Timeline and milestone planning

Hands-on Coding/Tools:

- Project planning tools

- Architecture documentation
- Collaboration platforms

POC 15 - Part 1: Capstone Project Initiation

- Define and architect comprehensive AI solution
- Set up development environment and initial codebase

Session 44: Capstone Project Implementation

Topics Covered:

- Full-stack AI application development
- Integration of multiple AI components
- Testing and validation strategies
- Performance optimization

Hands-on Coding/Tools:

- Full development stack
- Integration testing
- Performance profiling

POC 15 - Part 2: Capstone Project Development

- Implement core functionality
- Integrate AI components with production-ready code

Session 45: Capstone Project Deployment & Presentation

Topics Covered:

- Production deployment
- Monitoring and observability setup
- Documentation and handover
- Project presentation and demo

Hands-on Coding/Tools:

- Production deployment tools
- Presentation frameworks
- Documentation generation

POC 15 - Final: Capstone Project Completion

- Deploy to production environment
- Present comprehensive AI solution with full documentation

Course Outcomes

By the end of this course, learners will have:

1. **15 Working POC Projects** demonstrating various AI capabilities
2. **Production-Grade Skills** in AI system development and deployment
3. **Industry Best Practices** knowledge for enterprise AI development
4. **Hands-on Experience** with latest AI tools and frameworks
5. **Portfolio** of deployable AI applications
6. **Confidence** to architect and build complex AI solutions

---

Key Technologies & Tools Covered

- **Models**: Hugging Face Transformers, Google Gemini, Open-source LLMs
- **Frameworks**: LangChain, PyTorch, TensorFlow
- **Vector DBs**: FAISS, ChromaDB, Pinecone, Weaviate
- **Development**: Python, Jupyter, Google Colab, VS Code
- **Deployment**: Docker, Kubernetes, FastAPI, Streamlit
- **Cloud**: AWS, Google Cloud, Azure (free tiers)
- **Monitoring**: Prometheus, Grafana, MLflow
- **Testing**: pytest, custom AI testing frameworks

CodeYAA Network - Empowering the next generation of AI developers