

135-Hour GenAI, GitHub & LinkedIn Automation Curriculum

3-Hour Daily Sessions | Practical, Project-Based Learning | 45 Days

Total Hours: 135

Projects: 45+

GitHub Commits: 45+

LinkedIn Posts: 45+

Day 1: Foundation Setup & GitHub Mastery

Hours: 3

► Session Objectives

Set up complete development environment, Master GitHub basics and workflows, Create professional LinkedIn presence, Build and deploy first project

► Detailed Flow

Hour 1: Environment & GitHub Basics (0:00 - 1:00)

- Introduction to course structure and expectations (15 min)
- Install required tools: VS Code, Git, Python, Node.js (20 min)
- GitHub account creation and profile optimization (15 min)
- Git basics: init, clone, commit, push, pull (10 min)

Hour 2: GitHub Advanced & First Project (1:00 - 2:00)

- Creating repositories and README files (15 min)
- Branching strategies and best practices (15 min)
- GitHub Pages introduction (10 min)
- Start building: Personal Portfolio Website (20 min)

Hour 3: LinkedIn Setup & Deployment (2:00 - 3:00)

- LinkedIn profile optimization for AI/tech (20 min)
- Complete portfolio website with HTML/CSS (30 min)
- Deploy to GitHub Pages (10 min)

✓ Project of the Day

Personal AI Portfolio Website – A responsive portfolio showcasing AI journey, deployed on GitHub Pages

✓ GitHub Push Strategy

Repository: ai-portfolio-website

Commit Message: "Day 1: Created personal AI portfolio with GitHub Pages deployment"

Include: index.html, style.css, README.md

✓ LinkedIn Post Template

🚀 Day 1 of My 135-Hour GenAI Journey!

Today I built and deployed my first project – a personal AI portfolio website using GitHub Pages!

What I learned:

- ✓ Git version control fundamentals
- ✓ GitHub workflow and best practices
- ✓ Web deployment strategies

Live Demo: [Your GitHub Pages URL]

Code: [Your Repo Link]

#GenerativeAI #GitHub #100DaysOfCode #AILearning #WebDevelopment

[Screenshot of your portfolio]

✓ Resources

- 🔗 Git Documentation: <https://git-scm.com/doc>
- 🔗 GitHub Guides: <https://guides.github.com/>
- 🔗 GitHub Pages: <https://pages.github.com/>
- 🔗 Markdown Guide: <https://www.markdownguide.org/>

✓ Instructor Checklist

- All students have GitHub accounts
- First repository created
- LinkedIn profiles created/updated
- Git installed and configured
- Portfolio deployed successfully
- First LinkedIn post published

Day 2: Foundation Setup & GitHub Mastery - Advanced

Hours: 3

► Session Objectives

Deep dive into GitHub collaboration workflows, Master advanced Git commands and branching, Enhance portfolio with advanced features, Build professional development workflow

► Detailed Flow

Hour 1: Advanced Git & Collaboration (0:00 – 1:00)

- Forking and cloning repositories (15 min)
- Creating and managing pull requests (20 min)

Hour 2: Portfolio Enhancement (1:00 – 2:00)

- Adding JavaScript interactivity to portfolio (20 min)
- Implementing responsive design techniques (15 min)

- Code review process and best practices (15 min)
- Resolving merge conflicts (10 min)
- Adding project showcase sections (15 min)
- SEO optimization for portfolio (10 min)

Hour 3: Professional Workflow

Setup (2:00 – 3:00)

- Setting up GitHub Actions for CI/CD (20 min)
- Creating issue templates and project boards (20 min)
- Establishing daily development routine (10 min)
- Review and Q&A session (10 min)

✓ Project of the Day (Continued)

Enhanced Personal AI Portfolio with Advanced Features

✓ GitHub Push Strategy

Repository: ai-portfolio-website

Commit Message: "Day 2: Enhanced portfolio with JavaScript, responsive design, and GitHub Actions"

Include: script.js, responsive.css, github/workflows/deploy.yml

✓ LinkedIn Post Template

🚀 Day 2: Mastering Professional Development Workflows!

Enhanced my AI portfolio and learned advanced GitHub workflows!

Advanced Skills Learned:

- ✓ Pull request creation and code reviews
- ✓ GitHub Actions for automated deployment
- ✓ Responsive web design techniques
- ✓ Professional collaboration workflows

My portfolio now features:

- Interactive project showcase
- Mobile-responsive design
- Automated deployment pipeline
- Professional code organization

Live Demo: [Updated Portfolio URL]

Source Code: [GitHub Repo]

#GitHub #WebDevelopment #CI_CD #ProfessionalDevelopment #Coding



Instructor Checklist

- All students understand pull request workflow
- GitHub Actions pipelines working
- Advanced Git commands mastered
- Portfolios have interactive features
- Professional collaboration practices established
- Daily workflow routine established

Day 3: Introduction to AI & Python Fundamentals

Hours: 3

► Session Objectives

Understand AI, ML, DL relationships, Learn Python essentials for AI, Build first AI-powered application, Establish daily posting habit

► Detailed Flow

Hour 1: AI Fundamentals (0:00 - 1:00)

- What is AI? Types of AI: Narrow, General, Super (20 min)

Hour 2: Python for AI Basics (1:00 - 2:00)

- Python basics: variables, data types, loops (20 min)

- AI vs ML vs DL: Visual explanation with examples (20 min)
- Real-world AI applications across industries (15 min)
- Introduction to Generative AI concept (5 min)

- Functions and libraries (15 min)
- Introduction to key AI libraries: requests, json (10 min)
- Setting up API keys and environment variables (15 min)

Hour 3: Build AI Quote Generator (2:00 – 3:00)

- Project setup and structure (10 min)
- Integrating a free AI API (Hugging Face) (25 min)
- Creating a simple web interface (20 min)
- Testing and debugging (5 min)

✓ Project of the Day

AI-Powered Motivational Quote Generator – Python application using AI API to generate inspirational quotes

✓ GitHub Push Strategy

Repository: ai-quote-generator

Commit Message: "Day 3: Built AI-powered quote generator using Python and Hugging Face API"

Include: main.py, requirements.txt, README.md, .env.example

✓ LinkedIn Post Template

 Day 3: Understanding AI Fundamentals!

Built my first AI-powered application - a Motivational Quote Generator!

Key Learnings:

- 📊 Difference between AI, ML, and Deep Learning
- 🐍 Python fundamentals for AI development
- 🔗 Working with AI APIs (Hugging Face)
- ⚡ Environment setup and API key management

The bot generates unique motivational quotes on demand using transformer models!

GitHub: [Repo Link]

```
#ArtificialIntelligence #Python #MachineLearning #GenerativeAI  
#TechLearning
```

[Screenshot of your quote generator output]

✓ Resources

- 🔗 Python Tutorial: <https://docs.python.org/3/tutorial/>
- 🔗 Hugging Face API: <https://huggingface.co/docs/api-inference/>
- 🔗 AI Basics: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- 🔗 Python for AI: <https://realpython.com/tutorials/machine-learning/>

✓ Instructor Checklist

- AI concepts clearly explained with examples
- API integration working
- Code pushed to GitHub
- Python environment setup complete
- Project runs successfully
- LinkedIn post published with screenshot

Day 4: Advanced Python & API Integration

Hours: 3

► Session Objectives

Master advanced Python concepts for AI, Learn API authentication and security, Build complete web application with AI integration, Implement error handling and logging

► Detailed Flow

Hour 1: Advanced Python for AI (0:00 - 1:00)

- Object-oriented programming for AI applications (20 min)
- Working with virtual environments and dependencies (15 min)
- File handling and data persistence (15 min)
- Exception handling in Python (10 min)

Hour 2: API Security & Best Practices (1:00 - 2:00)

- Secure API key management techniques (20 min)
- Rate limiting and request optimization (15 min)
- Error handling for API failures (15 min)
- Logging and debugging API calls (10 min)

Hour 3: Complete Application Development (2:00 - 3:00)

- Building full web application with Flask/FastAPI (25 min)
- Implementing frontend-backend integration (20 min)
- Adding user authentication (basic) (10 min)
- Deployment preparation (5 min)



Project of the Day (Continued)

Enhanced AI Quote Generator with Web Interface



GitHub Push Strategy

Repository: ai-quote-generator-enhanced

Commit Message: "Day 4: Complete web application with Flask, secure API integration, and error handling"
Include: app.py, templates/, static/, config.py, requirements.txt

✓ LinkedIn Post Template

🚀 Day 4: Building Production-Ready AI Applications!

Enhanced my AI Quote Generator into a complete web application!

Advanced Skills Learned:

- 🔒 Secure API key management
- ⚡ Flask web application development
- ⌚ Error handling and logging
- 🔧 Virtual environment management

Features Added:

- Full web interface with Bootstrap
- Secure API authentication
- Error handling and user feedback
- Production-ready structure

Demo: [Live Application URL]

Source: [GitHub Repo]

#Python #Flask #WebDevelopment #APIIntegration #ProductionReady

✓ Instructor Checklist

- Students master OOP concepts for AI
- Web applications functioning properly
- Production deployment understanding
- Secure API key management implemented
- Error handling and logging working
- Project architecture discussions completed

► Session Objectives

Compare major LLM providers, Understand unique capabilities of each, Build multi-model application, Learn model selection strategies

► Detailed Flow

Hour 1: LLM Landscape Overview (0:00 – 1:00)

- ChatGPT vs Claude vs Gemini vs Llama (20 min)
- Strengths and weaknesses of each (20 min)
- Pricing and API access comparison (10 min)
- Use case matching (10 min)

Hour 2: Hands-on with Each Model (1:00 – 2:00)

- Test Claude's capabilities (15 min)
- Test Gemini's multimodal features (15 min)
- Test Llama (via Groq/Together) (15 min)
- Compare outputs for same task (15 min)

Hour 3: Build Model Comparison Tool (2:00 – 3:00)

- Create unified interface for multiple LLMs (20 min)
- Implement parallel querying (20 min)
- Add response comparison view (15 min)
- Generate comparison reports (5 min)



Project of the Day

LLM Comparison Dashboard – Test same prompt across multiple AI models simultaneously

✓ GitHub Push Strategy

Repository: llm-comparison-dashboard

Commit Message: "Day 5: Built multi-model comparison tool for GPT, Claude, Gemini, Llama"

Include: app.py, models/, comparison.py, reports/, README.md, model_guide.md

✓ LinkedIn Post Template

🤖 Day 5: Exploring GPT Alternatives!

Built an LLM Comparison Dashboard to test GPT, Claude, Gemini, and Llama side-by-side!

Key Findings:

- ⌚ Claude: Best for analysis and safety
- 🌟 Gemini: Excellent multimodal capabilities
- 🦙 Llama: Fast and cost-effective
- 💬 GPT: Most versatile

Tested 20+ prompts across all models - fascinating differences!

Model Selection Guide:

- Code? GPT/Claude
- Analysis? Claude
- Images? Gemini
- Speed/Cost? Llama

Tool: [GitHub Repo]

Results: [Comparison Report]

#LLM #Claude #Gemini #Llama #ChatGPT #AIComparison #ModelSelection

[Side-by-side comparison screenshot]

✓ Resources

🔗 Claude: <https://www.anthropic.com/clause>

🔗 Gemini: <https://deepmind.google/technologies/gemini/>

🔗 Llama: <https://www.llama.com/>

Instructor Checklist

- All models accessible
- Multiple test cases run
- Dashboard functional
- Selection guide created
- Comparison criteria defined
- Differences documented
- Pricing analysis complete

Day 6: Advanced Multi-Model Integration

Hours: 3

► Session Objectives

Implement fallback strategies between models, Build cost optimization algorithms, Create model performance analytics, Develop intelligent model selection system

► Detailed Flow

Hour 1: Intelligent Model Selection (0:00 - 1:00)

- Dynamic model selection based on task type (20 min)
- Cost-performance optimization algorithms (20 min)
- Implementing fallback mechanisms (10 min)
- Quality vs speed trade-offs (10 min)

Hour 2: Performance Analytics (1:00 - 2:00)

- Building model performance tracking (20 min)
- Cost analysis and budgeting features (15 min)
- Response quality metrics implementation (15 min)
- Real-time monitoring dashboard (10 min)

Hour 3: Production Deployment (2:00 - 3:00)

- Containerizing multi-model application (20 min)
- Implementing rate limiting and quotas (15 min)
- Adding user authentication (15 min)
- Deployment to cloud platform (10 min)



Project of the Day (Continued)

Advanced LLM Orchestration System



GitHub Push Strategy

Repository: llm-orchestration-system

Commit Message: "Day 6: Advanced LLM orchestration with intelligent model selection and analytics"

Include: orchestrator.py, analytics/, deployment/, docker-compose.yml



LinkedIn Post Template



Day 6: Building Intelligent LLM Orchestration!

Created an advanced LLM orchestration system with smart model selection!

System Features:

- 🤖 Intelligent model routing based on task
- 💰 Dynamic cost optimization
- 📊 Real-time performance analytics
- ⚡ Automatic fallback mechanisms

Key Innovations:

- 40% cost reduction through smart routing
- 95% success rate with automatic fallbacks
- Real-time cost tracking and alerts
- Production-ready deployment

Demo: [Live System URL]

Architecture: [GitHub Repo]

Instructor Checklist

- Intelligent model selection working
- Performance analytics implemented
- Production deployment successful
- Cost optimization algorithms functional
- Fallback mechanisms reliable
- System architecture documented

Day 7: Large Language Models & First Chatbot Hours: 3

► Session Objectives

Understand LLM architecture and concepts, Learn about transformers and attention mechanisms, Build a custom chatbot using OpenAI/free alternatives, Practice prompt engineering basics

► Detailed Flow

Hour 1: Understanding LLMs (0:00 - 1:00)

- What are Large Language Models? (15 min)
- Transformer architecture basics (attention is all you need) (20 min)
- Pre-training vs Fine-tuning explained (15 min)
- Popular LLMs: GPT, BERT, LLAMA overview (10 min)

Hour 2: Prompt Engineering Fundamentals (1:00 - 2:00)

- What is prompt engineering? (10 min)
- Elements of effective prompts: context, instruction, examples (20 min)
- Prompt patterns: zero-shot, few-shot, chain-of-thought (20 min)
- Hands-on: Testing different prompts (10 min)

Hour 3: Build Custom Chatbot

(2:00 – 3:00)

- Setting up LLM API (OpenAI or free alternative) (15 min)
- Building a specialized chatbot (e.g., Python tutor) (30 min)
- Adding conversation memory (10 min)
- Testing and refinement (5 min)



Project of the Day

AI Study Buddy Chatbot – An educational chatbot specialized in explaining programming concepts



GitHub Push Strategy

Repository: ai-study-buddy

Commit Message: "Day 7: Created AI Study Buddy chatbot with LLM integration and prompt engineering"

Include: chatbot.py, prompts.txt, requirements.txt, README.md, conversation_history.json



LinkedIn Post Template



Day 7: Diving Deep into Large Language Models!

Created an AI Study Buddy – a personalized chatbot that helps explain programming concepts!

Today's Discoveries:

- 🧠 How transformer models revolutionized NLP
- 💡 The art of prompt engineering
- ⌚ Pre-training vs Fine-tuning in LLMs
- 💬 Building conversational AI with memory

This chatbot can explain any programming concept in simple terms and even provide code examples!

Try it: [Demo Link]

Code: [GitHub Link]

#LLM #Transformers #Chatbots #PromptEngineering #OpenAI #NLP

[Screenshot of chatbot conversation]

✓ Resources

- 🔗 Attention Is All You Need: <https://arxiv.org/abs/1706.03762>
- 🔗 Prompt Engineering Guide: <https://www.promptingguide.ai/>
- 🔗 OpenAI API Docs: <https://platform.openai.com/docs/>
- 🔗 LangChain Documentation: <https://python.langchain.com/docs/>
- 🔗 Free LLM APIs: <https://huggingface.co/models>

✓ Instructor Checklist

- Transformer concepts explained visually
- API keys configured properly
- Conversation memory working
- LinkedIn post includes demo
- Students understand prompt engineering basics
- Chatbot responds appropriately
- GitHub repository well-documented

Day 8: Advanced Chatbot Development

Hours: 3

► Session Objectives

Implement advanced conversation management, Build context-aware chatbots, Add file upload and processing capabilities, Create chatbot analytics and monitoring

► Detailed Flow

Hour 1: Advanced Conversation Management (0:00 - 1:00)

- Implementing conversation context tracking (20 min)
- Handling multi-turn dialogues effectively (15 min)
- Managing conversation state and history (15 min)
- Implementing conversation timeouts (10 min)

Hour 2: File Processing & Context Enrichment (1:00 - 2:00)

- Adding file upload capabilities to chatbot (20 min)
- Processing PDF, DOCX, and TXT files (15 min)
- Extracting and using file content in conversations (15 min)
- Implementing document summarization (10 min)

Hour 3: Analytics & Production Features (2:00 - 3:00)

- Building chatbot usage analytics (20 min)
- Implementing user feedback collection (15 min)
- Adding performance monitoring (15 min)
- Preparing for production deployment (10 min)



Project of the Day (Continued)

Enhanced AI Study Buddy with Advanced Features



GitHub Push Strategy

Repository: ai-study-buddy-advanced

Commit Message: "Day 8: Enhanced chatbot with file processing, advanced conversation management, and analytics"

Include: enhanced_chatbot.py, file_processor/, analytics/, monitoring/



LinkedIn Post Template

Enhanced my AI Study Buddy with enterprise features!

Advanced Features Added:

-  File upload & processing (PDF, DOCX, TXT)
-  Advanced conversation context management
-  Usage analytics and monitoring
-  Production deployment configuration

Performance Metrics:

- 95% conversation continuity
- 30% faster response times
- Real-time usage analytics
- File processing in < 5 seconds

Demo: [Enhanced Chatbot URL]

Source: [GitHub Repo]

#ChatbotDevelopment #NLP #AIEngineering #ProductionAI
#ConversationalAI



Instructor Checklist

- Advanced conversation management implemented
- Context enrichment functioning properly
- Production deployment preparation complete
- File processing capabilities working
- Analytics system collecting data
- Performance monitoring established

Day 9: Generative Adversarial Networks (GANs)

Introduction

Hours: 3

► Session Objectives

Understand GAN architecture (Generator + Discriminator), Learn about latent space and image generation, Build a simple GAN or use pre-trained models, Generate first AI images

► Detailed Flow

Hour 1: GAN Theory (0:00 – 1:00)

- Introduction to Generative Models (10 min)
- GAN Architecture: Generator vs Discriminator (20 min)
- Training process and adversarial learning (15 min)
- Applications: deepfakes, art, data augmentation (15 min)

Hour 2: Hands-on with Pre-trained GANs (1:00 – 2:00)

- Setting up PyTorch/TensorFlow environment (15 min)
- Loading pre-trained GAN models (10 min)
- Understanding latent space exploration (15 min)
- Generating images with different seeds (20 min)

Hour 3: Build GAN Image Generator (2:00 – 3:00)

- Create a simple web interface for GAN (20 min)
- Implement random face/art generation (25 min)
- Save and display results (10 min)
- Create image gallery (5 min)



Project of the Day

AI Art Generator using StyleGAN - Generate unique artwork using pre-trained GAN models



GitHub Push Strategy

Repository: stylegan-art-generator

Commit Message: "Day 9: Built AI art generator using StyleGAN with web interface"

Include: generate.py, app.py, models/, outputs/, requirements.txt, README.md

✓ LinkedIn Post Template

⌚ Day 9: Exploring Generative Adversarial Networks!

Built an AI Art Generator that creates unique artwork using StyleGAN!

What I Learned:

- ⌚ How GANs work: Generator vs Discriminator
- ⌚ Exploring latent space in neural networks
- ⌚ Image generation from random noise
- ⌚ Working with pre-trained models

Generated 50+ unique AI artworks today!

Gallery: [\[GitHub Pages Link\]](#)

Source Code: [\[GitHub Repo\]](#)

#GAN #AIArt #DeepLearning #GenerativeAI #StyleGAN #ComputerVision

[Grid of 4 generated images]

✓ Resources

- ⌚ GAN Paper: <https://arxiv.org/abs/1406.2661>
- ⌚ StyleGAN: <https://github.com/NVlabs/stylegan2>
- ⌚ PyTorch GANs:
https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
- ⌚ This Person Does Not Exist: <https://thispersondoesnotexist.com/>
- ⌚ GAN Lab Visualization: <https://poloclub.github.io/ganlab/>

✓ Instructor Checklist

- GAN concepts explained with diagrams
- Environment setup complete (PyTorch/TF)

20

- Pre-trained models downloaded
- Gallery interface created
- LinkedIn post has visual examples
- Image generation working
- GitHub includes sample outputs

Day 10: Advanced GAN Training & Customization

Hours: 3

► Session Objectives

Train custom GAN models, Implement GAN training monitoring, Create style transfer with GANs, Build GAN-based image editing tools

► Detailed Flow

Hour 1: Custom GAN Training (0:00 - 1:00)

- Preparing custom datasets for GAN training (20 min)
- Implementing DCGAN architecture from scratch (15 min)
- Monitoring GAN training: loss curves, generated samples (15 min)
- Addressing common GAN training issues (10 min)

Hour 2: Style Transfer & Manipulation (1:00 - 2:00)

- Implementing style mixing with StyleGAN (20 min)
- Creating interpolation between generated images (15 min)
- Building image attribute manipulation tools (15 min)
- Generating image variations (10 min)

Hour 3: Production GAN Application (2:00 - 3:00)

- Building batch image generation pipeline (20 min)
- Implementing quality filtering for generated images (15 min)

- Creating API for GAN image generation (15 min)
- Deployment considerations for GAN models (10 min)



Project of the Day (Continued)

Advanced GAN Training System with Custom Models



GitHub Push Strategy

Repository: advanced-gan-training

Commit Message: "Day 10: Custom GAN training system with monitoring and style transfer"

Include: train_gan.py, style_transfer.py, monitoring/, custom_models/



LinkedIn Post Template



Day 10: Mastering GAN Training & Customization!

Built an advanced GAN training system with custom models and style manipulation!

Advanced Features:

- 🔧 Custom GAN training from scratch
- 🎨 Style transfer and mixing capabilities
- 📊 Real-time training monitoring
- ⚡ Batch generation pipeline

Achievements:

- Trained custom GAN on personal dataset
- Implemented style interpolation
- Built production-ready generation API
- Achieved 85% realistic image generation

This opens up possibilities for custom AI art generation for any domain!

Demo: [Live Generation API]

Source: [GitHub Repo]

Instructor Checklist

- Custom GAN training implemented
- Style transfer functioning
- Production API operational
- Training monitoring working
- Image quality filtering effective
- Students understand training challenges

Day 11: Variational Autoencoders (VAEs) & Image Manipulation

Hours: 3

► Session Objectives

Understand VAE architecture and latent space, Learn image encoding and decoding, Build image manipulation tools, Compare GANs vs VAEs

► Detailed Flow

Hour 1: VAE Fundamentals (0:00 - 1:00)

- Autoencoders basics: Encoder-Decoder (15 min)
- Variational Autoencoders: probabilistic approach (20 min)
- Latent space representation and sampling (15 min)
- VAE vs GAN: when to use what (10 min)

Hour 2: Practical VAE Implementation (1:00 - 2:00)

- Load pre-trained VAE models (15 min)
- Image encoding to latent vectors (15 min)
- Latent space interpolation (15 min)
- Decoding and reconstruction (15 min)

Hour 3: Build Image Morphing Tool (2:00 - 3:00)

- Create image upload interface (15 min)
- Implement image-to-image morphing (25 min)
- Generate smooth transitions between images (15 min)
- Export video/GIF of morphing process (5 min)

✓ Project of the Day

AI Image Morpher - Tool that smoothly transitions between two images through latent space

✓ GitHub Push Strategy

Repository: vae-image-morpher

Commit Message: "Day 11: Built VAE-based image morphing tool with latent space interpolation"

Include: morpher.py, encoder.py, decoder.py, web_app.py, examples/, README.md

✓ LinkedIn Post Template

⌚ Day 11: Mastering Variational Autoencoders !

Created an AI Image Morpher that seamlessly transitions between any two images!

Today's Insights:

- 🌐 VAE architecture: Encoding & Decoding
- 🌐 Latent space as a compressed representation
- 🎬 Creating smooth interpolations between images
- ⚖️ GANs vs VAEs: Understanding the differences

Watch my AI morph a cat into a dog in 10 smooth steps !

Demo: [Video/GIF Link]

Code: [GitHub Repo]

#VAE #Autoencoders #ComputerVision #DeepLearning #AIArt
#ImageProcessing

[GIF of image morphing]

✓ Resources

- 🔗 VAE Tutorial: <https://arxiv.org/abs/1312.6114>
- 🔗 Keras VAE Example: <https://keras.io/examples/generative/vae/>
- 🔗 Understanding VAEs:
<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>
- 🔗 Latent Space Visualization: <https://projector.tensorflow.org/>

✓ Instructor Checklist

- VAE theory explained clearly
- Latent space interpolation working
- GIF/video export functioning
- LinkedIn post includes morphing demo
- Encoding/decoding demonstrated
- Morphing produces smooth results
- Comparison chart (GAN vs VAE) created

Day 12: Advanced VAE Applications & Hybrid Models

Hours: 3

► Session Objectives

► Detailed Flow

Hour 1: Conditional VAEs (0:00 - 1:00)

- Understanding conditional generation (20 min)
- Implementing CVAE for labeled data (15 min)
- Controlling generated image attributes (15 min)
- Building attribute-based image search (10 min)

Hour 2: VAE-GAN Hybrid Models (1:00 - 2:00)

- Combining VAE and GAN strengths (20 min)
- Implementing VAE-GAN architecture (15 min)
- Training hybrid models effectively (15 min)
- Comparing results with pure models (10 min)

Hour 3: Production Applications (2:00 - 3:00)

- Building anomaly detection with VAEs (20 min)
- Implementing recommendation systems (15 min)
- Creating data augmentation pipelines (15 min)
- Performance optimization techniques (10 min)



Project of the Day (Continued)

Advanced VAE Applications System



GitHub Push Strategy

Repository: advanced-vae-applications

Commit Message: "Day 12: Advanced VAE applications including CVAE, VAE-GAN, and anomaly detection"

✓ LinkedIn Post Template

🚀 Day 12: Advanced VAE Applications & Hybrid Models!

Built a comprehensive VAE application system with multiple advanced use cases!

System Features:

- ⌚ Conditional VAE for attribute-controlled generation
- 🤝 VAE-GAN hybrid models for improved quality
- 🔍 Anomaly detection in complex datasets
- 📈 Recommendation system using latent representations

Practical Applications:

- Generate images with specific attributes
- Detect anomalies in manufacturing processes
- Recommend similar products using latent space
- Create hybrid models combining VAE & GAN benefits

Demo: [Live Applications Dashboard]

Source: [GitHub Repo]

#VAE #DeepLearning #AnomalyDetection #RecommendationSystems
#HybridModels

✓ Instructor Checklist

- Conditional VAE implementation working
- Anomaly detection effective
- Performance optimization applied
- VAE-GAN hybrid models functional
- Recommendation systems operational
- Students understand practical applications

► Session Objectives

Understand BERT architecture and bidirectional context, Learn about tokenization and embeddings, Build sentiment analysis tool, Implement text classification

► Detailed Flow

Hour 1: BERT Architecture (0:00 - 1:00)

- Transformer encoders and BERT (15 min)
- Bidirectional context understanding (15 min)
- Pre-training tasks: MLM and NSP (15 min)
- BERT variants: RoBERTa, ALBERT, DistilBERT (15 min)

Hour 2: Working with BERT (1:00 - 2:00)

- Hugging Face Transformers library (15 min)
- Loading pre-trained BERT models (10 min)
- Tokenization and input preparation (15 min)
- Fine-tuning basics (concept overview) (20 min)

Hour 3: Build Sentiment Analyzer (2:00 - 3:00)

- Create sentiment analysis pipeline (20 min)
- Build web interface for text input (15 min)
- Add emotion detection (multi-class) (15 min)
- Visualize results with charts (10 min)



Project of the Day

AI Sentiment & Emotion Analyzer - Analyze text sentiment and detect multiple emotions using BERT

✓ GitHub Push Strategy

Repository: bert-sentiment-analyzer

Commit Message: "Day 13: Built BERT-powered sentiment and emotion analyzer with visualization"

Include: analyzer.py, app.py, models/, requirements.txt, README.md, static/

✓ LinkedIn Post Template

📊 Day 13: Exploring BERT & Text Understanding!

Built an AI-powered Sentiment & Emotion Analyzer using BERT!

Key Takeaways:

- 🔤 Bidirectional transformers for context understanding
- 🎭 Multi-class emotion detection (joy, anger, fear, etc.)
- 📈 Real-time sentiment scoring
- 🤖 Working with Hugging Face Transformers

Analyzed 100+ text samples with 95%+ accuracy!

Try it: [Demo Link]

GitHub: [Repo Link]

#BERT #NLP #SentimentAnalysis #Transformers #HuggingFace
#TextAnalytics

[Screenshot showing sentiment analysis results]

✓ Resources

- 🔗 BERT Paper: <https://arxiv.org/abs/1810.04805>
- 🔗 Hugging Face Course: <https://huggingface.co/course/chapter1>
- 🔗 BERT Explained: <https://jalammar.github.io/illustrated-bert/>
- 🔗 Transformers Library: <https://huggingface.co/docs/transformers/>

✓ Instructor Checklist

- BERT architecture explained with diagrams
- Transformers library installed

- Model loading successful
- Emotion detection working
- Demo ready for LinkedIn
- Sentiment analysis accurate
- Visualization clear and informative

Day 14: Advanced BERT Applications & Fine-tuning

Hours: 3

► Session Objectives

Implement BERT fine-tuning for custom tasks, Build multi-label classification systems, Create text similarity and search applications, Develop BERT-based question answering

► Detailed Flow

Hour 1: BERT Fine-tuning Techniques (0:00 - 1:00)

- Preparing custom datasets for fine-tuning (20 min)
- Implementing different fine-tuning strategies (15 min)
- Monitoring fine-tuning progress (15 min)
- Evaluating fine-tuned models (10 min)

Hour 2: Advanced Text Applications (1:00 - 2:00)

- Building multi-label classification system (20 min)
- Implementing semantic text similarity (15 min)
- Creating text search and retrieval (15 min)
- Developing question answering systems (10 min)

Hour 3: Production Deployment (2:00 - 3:00)

- Optimizing BERT models for production (20 min)
- Implementing batch processing for efficiency (15 min)

- Building REST API for text analysis (15 min)
- Monitoring model performance in production (10 min)



Project of the Day (Continued)

Advanced BERT Application Suite



GitHub Push Strategy

Repository: advanced-bert-applications

Commit Message: "Day 14: Advanced BERT applications including fine-tuning, QA, and similarity search"

Include: fine_tuning.py, qa_system.py, similarity_search/, production_api/



LinkedIn Post Template



Day 14: Advanced BERT Applications & Fine-tuning!

Built a comprehensive BERT application suite with production-ready features!

Advanced Applications:

- 🎯 Fine-tuned BERT for custom domain tasks
- 🔍 Question answering system with context
- 📊 Multi-label classification for complex texts
- 🔗 Semantic similarity and search

Performance Metrics:

- 92% accuracy on custom classification task
- <100ms response time for QA
- 95% similarity matching accuracy
- Production API handling 1000+ requests/min

This demonstrates the real power of transformer models beyond basic sentiment analysis!

Demo: [Live API Endpoints]

Source: [GitHub Repo]

Instructor Checklist

- BERT fine-tuning implemented
- Question answering system functional
- Production optimization applied
- Multi-label classification working
- Semantic similarity effective
- Performance monitoring established

Day 15: Advanced Prompt Engineering & ChatGPT Deep Dive

Hours: 3

► Session Objectives

Master advanced prompt techniques, Learn few-shot and chain-of-thought prompting, Build prompt template library, Create GPT-powered applications

► Detailed Flow

Hour 1: Advanced Prompting Techniques (0:00 - 1:00)

- Zero-shot vs Few-shot prompting (15 min)
- Chain-of-thought reasoning (15 min)
- Role prompting and persona creation (15 min)
- Prompt parameters: temperature, top_p, etc. (15 min)

Hour 2: Prompt Design Patterns (1:00 - 2:00)

- Structured output prompts (JSON, XML) (15 min)
- Instruction following optimization (15 min)
- Context window management (15 min)
- Handling prompt injection and security (15 min)

min)

Hour 3: Build Prompt Library Tool (2:00 – 3:00)

- Design prompt template system (15 min)
- Create categorized prompt library (20 min)
- Build testing interface (15 min)
- Implement version control for prompts (10 min)

✓ Project of the Day

AI Prompt Library & Tester - Comprehensive tool to store, test, and version prompts

✓ GitHub Push Strategy

Repository: ai-prompt-library

Commit Message: "Day 15: Created comprehensive prompt engineering library with testing interface"

Include: prompts.json, tester.py, app.py, examples/, README.md, docs/

✓ LinkedIn Post Template

⌚ Day 15: Mastering the Art of Prompt Engineering!

Built a comprehensive AI Prompt Library with 50+ tested templates!

What I Discovered:

- 💡 Chain-of-thought prompting for complex reasoning
- 🎭 Role-based prompts for specialized outputs
- 📊 Structured outputs (JSON/XML) from LLMs
- 🔒 Prompt security and injection prevention

Categories included: Code generation, Data analysis, Creative writing, Problem-solving, and more!

Access the library: [GitHub Pages Link]

Source: [GitHub Repo]

#PromptEngineering #ChatGPT #LLM #AITools #GPT4 #PromptDesign

[Screenshot of prompt library interface]

✓ Resources

- 🔗 Prompt Engineering Guide: <https://www.promptingguide.ai/>
- 🔗 OpenAI Best Practices:
<https://platform.openai.com/docs/guides/prompt-engineering>
- 🔗 Prompt Patterns: <https://arxiv.org/abs/2302.11382>
- 🔗 Learn Prompting: <https://learnprompting.org/>

✓ Instructor Checklist

- Advanced techniques demonstrated live
- Testing interface functional
- Documentation comprehensive
- LinkedIn post includes examples
- 50+ prompt templates created
- Security considerations discussed
- GitHub repository organized

Day 16: Enterprise Prompt Engineering & Optimization

Hours: 3

► Session Objectives

Build enterprise-grade prompt management systems, Implement A/B testing for prompts, Create automated prompt optimization, Develop monitoring and analytics for prompt performance

► Detailed Flow

Hour 1: Enterprise Prompt Systems (0:00 - 1:00)

- Designing scalable prompt management architecture (20 min)
- Implementing team collaboration features (15 min)
- Building version control and rollback systems (15 min)
- Creating approval workflows for prompts (10 min)

Hour 2: Optimization & Testing (1:00 - 2:00)

- Implementing A/B testing framework for prompts (20 min)
- Building automated prompt optimization algorithms (15 min)
- Creating performance benchmarking (15 min)
- Implementing cost optimization strategies (10 min)

Hour 3: Analytics & Monitoring (2:00 - 3:00)

- Building real-time prompt performance dashboards (20 min)
- Implementing alerting for prompt degradation (15 min)
- Creating usage analytics and reporting (15 min)
- Integrating with existing enterprise systems (10 min)



Project of the Day (Continued)

Enterprise Prompt Management Platform



GitHub Push Strategy

Repository: enterprise-prompt-platform

Commit Message: "Day 16: Enterprise-grade prompt management with A/B testing, optimization, and analytics"

Include: platform/, ab_testing/, optimization/, analytics/, monitoring/

✓ LinkedIn Post Template



Day 16: Building Enterprise Prompt Engineering Systems!

Created a production-ready prompt management platform for large-scale AI applications!

Enterprise Features:

- 🏢 Team collaboration & version control
- 📊 A/B testing with statistical significance
- ⚡ Automated prompt optimization
- 📈 Real-time performance analytics
- 🔔 Alerting for prompt degradation

Business Impact:

- 35% improvement in prompt effectiveness
- 50% reduction in manual prompt tuning
- Real-time monitoring of AI costs
- Scalable to 1000+ prompts across teams

This is how large organizations scale their AI prompt engineering efforts!

Demo: [Platform Dashboard]

Source: [GitHub Repo]

#PromptEngineering #EnterpriseAI #AIOptimization #MLOps
#AIGovernance



Instructor Checklist

- Enterprise architecture designed
- Optimization algorithms working
- Alerting system implemented
- A/B testing framework functional
- Analytics dashboard operational
- Scalability considerations addressed

► Session Objectives

Compare major chatbot platforms, Build multi-turn conversation system, Implement context management, Create specialized domain chatbot

► Detailed Flow

Hour 1: Chatbot Platforms

Overview (0:00 – 1:00)

- ChatGPT, Claude, Bard/Gemini comparison (20 min)
- Platform capabilities and limitations (15 min)
- API access and pricing models (10 min)
- When to use which platform (15 min)

Hour 2: Conversational AI Design

(1:00 – 2:00)

- Conversation flow design principles (15 min)
- Context window and memory management (20 min)
- Multi-turn dialogue handling (15 min)
- Error handling and fallback strategies (10 min)

Hour 3: Build Domain-Specific

Chatbot (2:00 – 3:00)

- Create medical/legal/education assistant (20 min)
- Implement RAG (basic concept) (20 min)
- Add personality and tone control (10 min)
- Testing and refinement (10 min)



Project of the Day

Healthcare Assistant Chatbot – Specialized AI assistant for health information (disclaimer included)

✓ GitHub Push Strategy

Repository: healthcare-ai-assistant

Commit Message: "Day 17: Built specialized healthcare chatbot with context management and RAG"

Include: chatbot.py, knowledge_base/, context_manager.py, app.py, README.md

✓ LinkedIn Post Template



Day 17: Building Intelligent Conversational AI!

Created a Healthcare Assistant Chatbot with domain-specific knowledge!

Implementation Highlights:

- 💬 Multi-turn conversation with context memory
- 📚 RAG (Retrieval Augmented Generation) basics
- 🎯 Domain specialization in healthcare
- ⚖️ Ethical considerations and disclaimers

The bot can answer health questions, provide wellness tips, and guide users - all while maintaining conversation context!

Try it: [\[Demo Link\]](#)

Code: [\[GitHub Repo\]](#)

*Note: For educational purposes only, not medical advice

#ConversationalAI #Chatbots #HealthTech #RAG #AIEthics #NLP

[Screenshot of conversation with chatbot]

✓ Resources

🔗 **Conversational AI Design:**

<https://www.nngroup.com/articles/chatbot-design/>

🔗 **RAG Tutorial:** <https://www.pinecone.io/learn/retrieval-augmented-generation/>

🔗 **LangChain for Chatbots:**

https://python.langchain.com/docs/use_cases/chatbots/

✓ Instructor Checklist

- Platform comparison chart created
- Multi-turn conversations functional
- Ethical disclaimers included
- Demo video recorded
- Context management working
- Domain knowledge integrated
- Error handling robust

Day 18: Enterprise Chatbot Development & Deployment

Hours: 3

▶ Session Objectives

Build production-ready chatbot systems, Implement advanced RAG for chatbots, Create chatbot analytics and monitoring, Deploy chatbots at scale

▶ Detailed Flow

Hour 1: Production Chatbot Architecture (0:00 - 1:00)

- Designing scalable chatbot infrastructure (20 min)
- Implementing load balancing and failover (15 min)
- Building session management for thousands of users (15 min)
- Security considerations for chatbot deployment (10 min)

Hour 2: Advanced RAG Implementation (1:00 - 2:00)

- Building sophisticated document retrieval systems (20 min)
- Implementing context-aware answer generation (15 min)
- Adding citation and source verification (15 min)

- Optimizing RAG for real-time responses (10 min)

Hour 3: Monitoring & Analytics (2:00 – 3:00)

- Building chatbot performance dashboards (20 min)
- Implementing conversation quality metrics (15 min)
- Creating user satisfaction tracking (15 min)
- Setting up automated testing pipelines (10 min)

✓ Project of the Day (Continued)

Enterprise Healthcare Chatbot Platform

✓ GitHub Push Strategy

Repository: enterprise-healthcare-chatbot

Commit Message: "Day 18: Production-ready healthcare chatbot with advanced RAG and monitoring"

Include: production_chatbot.py, advanced_rag/, monitoring/, deployment/

✓ LinkedIn Post Template

🚀 Day 18: Building Enterprise-Grade Chatbot Systems!

Created a production-ready healthcare chatbot platform with enterprise features!

Advanced Features:

- 💻 Scalable architecture for 10k+ concurrent users
- 🔍 Advanced RAG with citation and verification
- 📊 Real-time conversation analytics
- 🔒 HIPAA-compliant security measures

40

Performance Metrics:

- 95% answer accuracy with citations
- <2 second response time
- 99.9% uptime SLA
- Real-time quality monitoring

This demonstrates how to take chatbots from prototypes to production systems!

Demo: [Enterprise Dashboard]

Source: [GitHub Repo]

#EnterpriseAI #Chatbots #HealthcareAI #RAG #ProductionDeployment
#AIScalability



Instructor Checklist

- Production architecture designed
- Performance monitoring working
- Scalability testing conducted
- Advanced RAG implemented
- Security measures implemented
- Quality metrics established

Day 19: Introduction to n8n & Workflow Automation

Hours: 3

► Session Objectives

Understand workflow automation concepts, Set up n8n environment, Create first automated workflows, Integrate AI with automation

► Detailed Flow

Hour 1: n8n Setup & Basics (0:00 - 1:00)

Hour 2: Building First Workflows (1:00 - 2:00)

41

- What is n8n? Workflow automation intro (15 min)
 - Installation: Cloud vs Self-hosted (15 min)
 - n8n interface walkthrough (15 min)
 - Understanding nodes and connections (15 min)
- Webhook triggers and HTTP requests (20 min)
 - Data transformation and manipulation (20 min)
 - Conditional logic and branching (10 min)
 - Error handling in workflows (10 min)

Hour 3: AI-Powered Automation (2:00 – 3:00)

- Integrating OpenAI/AI APIs with n8n (20 min)
- Build: Auto email responder with AI (25 min)
- Testing and debugging workflows (10 min)
- Deployment and monitoring (5 min)

✓ Project of the Day

AI Email Auto-Responder – n8n workflow that auto-generates email responses using AI

✓ GitHub Push Strategy

Repository: n8n-ai-workflows

Commit Message: "Day 19: Created n8n workflow for AI-powered email auto-responder"

Include: workflows/, email-responder.json, README.md, setup-guide.md, screenshots/

✓ LinkedIn Post Template

⚙️ Day 19: Automating with AI & n8n!

Built an AI-powered Email Auto-Responder using n8n workflow automation!

What I Automated:

-  Email monitoring and parsing
-  AI-generated contextual responses
-  Smart reply suggestions
-  Response analytics

This workflow reads emails, understands context using AI, and generates professional responses automatically!

Workflow: [n8n Cloud Share Link]

GitHub: [Repo Link]

#n8n #Automation #WorkflowAutomation #NoCode #AIAutomation
#ProductivityTools

[Screenshot of n8n workflow diagram]

✓ Resources

-  n8n Documentation: <https://docs.n8n.io/>
-  n8n Community: <https://community.n8n.io/>
-  Workflow Templates: <https://n8n.io/workflows/>
-  n8n Academy: <https://docs.n8n.io/courses/>

✓ Instructor Checklist

- n8n installed/accessible successfully
 - Interface navigation comfortable
- First workflow created
 - AI integration working
- Email automation functional
 - Workflow exported and documented
- Screenshots for LinkedIn prepared

Day 20: Advanced n8n Workflows & Enterprise Automation

Hours: 3

▶ Session Objectives

Build complex multi-step workflows, Implement error handling and retry logic, Create workflow orchestration systems, Deploy production automation at scale

▶ Detailed Flow

Hour 1: Advanced Workflow

Design (0:00 – 1:00)

- Building complex decision trees in n8n (20 min)
- Implementing parallel processing workflows (15 min)
- Creating workflow templates for reuse (15 min)
- Best practices for workflow organization (10 min)

Hour 2: Production Automation

Systems (1:00 – 2:00)

- Building enterprise automation pipelines (20 min)
- Implementing robust error handling and retries (15 min)
- Creating monitoring and alerting systems (15 min)
- Performance optimization for large workflows (10 min)

Hour 3: Integration & Deployment

(2:00 – 3:00)

- Integrating n8n with external systems (20 min)
- Building API-driven automation workflows (15 min)
- Deploying n8n in production environments (15 min)
- Security and access control implementation (10 min)

✓ Project of the Day (Continued)

Enterprise Automation Platform with n8n

✓ GitHub Push Strategy

Repository: enterprise-n8n-automation

Commit Message: "Day 20: Enterprise-grade automation platform with complex workflows and monitoring"

Include: complex_workflows/, monitoring_system/, deployment/, security_config/

✓ LinkedIn Post Template



Day 20: Building Enterprise Automation Systems with n8n!

Created a production-ready automation platform handling complex business processes!

Advanced Features:

- 🏢 Multi-department workflow orchestration
- 🔄 Advanced error handling with retry logic
- 📊 Real-time workflow monitoring dashboard
- 🔒 Enterprise-grade security and access control

Automation Impact:

- Reduced manual processing time by 80%
- Automated 15+ business processes
- 99.5% workflow success rate
- Real-time SLA monitoring

This shows how n8n can power enterprise automation at scale!

Demo: [Workflow Dashboard]

Source: [GitHub Repo]

#WorkflowAutomation #n8n #EnterpriseAutomation #BusinessProcess
#DigitalTransformation

✓ Instructor Checklist

- Complex workflows designed
- Error handling implemented

45

- Monitoring system functional
- Security measures in place
- Production deployment configured
- Performance optimization applied

Day 21: LinkedIn Automation with n8n

Hours: 3

► Session Objectives

Automate LinkedIn posting, Create content calendars, Build engagement automation, Design AI content generator for LinkedIn

► Detailed Flow

Hour 1: LinkedIn API & n8n Integration (0:00 - 1:00)

- LinkedIn API overview and permissions (15 min)
- Connecting LinkedIn to n8n (20 min)
- Understanding LinkedIn posting limits (10 min)
- Content formatting for LinkedIn (15 min)

Hour 2: Content Automation Workflow (1:00 - 2:00)

- Build content scheduling system (20 min)
- Create hashtag generator with AI (15 min)
- Implement image/media handling (15 min)
- Set up posting calendar (10 min)

Hour 3: Advanced LinkedIn Automation (2:00 - 3:00)

- Auto-generate posts from RSS/news (20 min)
- Create engagement tracking (15 min)
- Build content variation system (15 min)
- Analytics and reporting (10 min)



Project of the Day

LinkedIn Content Automation System - Complete workflow for automated, AI-generated LinkedIn posts



GitHub Push Strategy

Repository: linkedin-automation-n8n

Commit Message: "Day 21: Built complete LinkedIn automation system with AI content generation"

Include: workflows/, linkedin-poster.json, content-templates/, README.md, analytics/



LinkedIn Post Template

Built a complete LinkedIn Content Automation System!

Features Implemented:

-  Scheduled posting system
-  AI-generated post content
-  Smart hashtag generation
-  Engagement analytics
-  Media handling and formatting

This system can generate and post AI-created content based on trending topics, all automatically!

Check out the workflow: [\[GitHub Repo\]](#)

What would you automate on LinkedIn?

```
#LinkedInAutomation #n8n #ContentAutomation #AIContent  
#SocialMediaMarketing
```

[Screenshot of n8n LinkedIn workflow]

Resources

-  LinkedIn API: <https://learn.microsoft.com/en-us/linkedin/>
-  n8n LinkedIn Node: <https://docs.n8n.io/integrations/builtin/app-nodes/n8n-nodes-base.linkedin/>
-  Social Media Automation Best Practices:
<https://buffer.com/resources/social-media-automation/>
-  LinkedIn Algorithm Guide:
<https://www.linkedin.com/business/marketing/blog/linkedin-ads/understanding-the-linkedin-algorithm>

Instructor Checklist

- LinkedIn API credentials configured
- Posting workflow functional
- n8n LinkedIn integration working
- AI content generation quality checked

- Scheduling system operational
- Ethics and best practices discussed
- Analytics dashboard created

Day 22: Advanced Social Media Automation & Analytics

Hours: 3

► Session Objectives

Build multi-platform social automation, Implement sentiment analysis for engagement, Create social listening workflows, Develop comprehensive social analytics

► Detailed Flow

Hour 1: Multi-Platform Automation (0:00 – 1:00)

- Building cross-platform posting workflows (20 min)
- Implementing content adaptation for different platforms (15 min)
- Creating unified content calendar (15 min)
- Handling platform-specific limitations (10 min)

Hour 2: Engagement & Listening (1:00 – 2:00)

- Building sentiment analysis for comments (20 min)
- Implementing automated engagement responses (15 min)
- Creating social listening workflows (15 min)
- Building brand mention tracking (10 min)

Hour 3: Analytics & Optimization (2:00 – 3:00)

- Building comprehensive social analytics dashboard (20 min)

- Implementing A/B testing for content (15 min)
- Creating ROI tracking for social activities (15 min)
- Building optimization recommendations (10 min)



Project of the Day (Continued)

Enterprise Social Media Management Platform



GitHub Push Strategy

Repository: enterprise-social-automation

Commit Message: "Day 22: Enterprise social media platform with multi-platform automation and analytics"

Include: multi_platform/, engagement_system/, analytics_dashboard/, optimization/



LinkedIn Post Template



Day 22: Building Enterprise Social Media Automation!

Created a comprehensive social media management platform for businesses!

Platform Features:

- 🌐 Multi-platform automation (LinkedIn, Twitter, Facebook)
- ⌚ AI-powered engagement and sentiment analysis
- 📊 Comprehensive analytics with ROI tracking
- 🔄 Automated optimization recommendations

Business Impact:

- 3x increase in content consistency
- 40% improvement in engagement rates
- Real-time brand sentiment monitoring
- Automated optimization saving 15 hours/week

This is how modern businesses manage their social presence at scale!

50

Demo: [Analytics Dashboard]

Source: [GitHub Repo]

#SocialMediaAutomation #DigitalMarketing #AIinMarketing
#SocialAnalytics #MarketingAutomation

Instructor Checklist

- Multi-platform automation working
- Sentiment analysis implemented
- Social listening functional
- Analytics dashboard operational
- A/B testing framework created
- ROI tracking implemented

Day 23: Image Generation – Stable Diffusion & Midjourney

Hours: 3

► Session Objectives

Understand diffusion models, Master prompt writing for image generation, Compare different image AI tools, Generate professional images

► Detailed Flow

Hour 1: Diffusion Models Theory (0:00 – 1:00)

- How diffusion models work (forward & reverse) (20 min)
- Stable Diffusion architecture overview (15 min)
- Text-to-image process explained (15 min)

Hour 2: Hands-on Image Generation (1:00 – 2:00)

- Setting up Stable Diffusion (local/API) (15 min)
- Midjourney basics and Discord setup (15 min)
- Prompt engineering for images (20 min)

- Midjourney vs DALL-E vs Stable Diffusion (10 min)

- Parameters: steps, CFG scale, sampling methods (10 min)

Hour 3: Build Image Generator App (2:00 – 3:00)

- Create web interface for Stable Diffusion (20 min)
- Implement prompt templates (15 min)
- Add style presets (realistic, anime, art, etc.) (15 min)
- Gallery and download functionality (10 min)

✓ Project of the Day

AI Image Studio – Web app for generating images with Stable Diffusion + prompt library

✓ GitHub Push Strategy

Repository: ai-image-studio

Commit Message: "Day 23: Built AI Image Studio with Stable Diffusion and prompt templates"

Include: app.py, generate.py, prompts.json, gallery/, requirements.txt, README.md

✓ LinkedIn Post Template

🎨 Day 23: Mastering AI Image Generation!

Created an AI Image Studio using Stable Diffusion!

What I Learned:

- 🌊 How diffusion models denoise images
- 👉 Prompt engineering for stunning visuals
- 🎯 CFG scale, sampling methods, and steps
- 🖼️ Comparing Midjourney, DALL-E, Stable Diffusion

Generated 50+ professional images today across various styles:
photorealistic, anime, oil painting, and more!

Gallery: [GitHub Pages Link]

Source: [GitHub Repo]

Which style is your favorite? Comment below!

#StableDiffusion #AIArt #Midjourney #ImageGeneration
#DiffusionModels #GenerativeAI

[4-image grid showing different styles]

✓ Resources

- 🔗 Stable Diffusion Paper: <https://arxiv.org/abs/2112.10752>
- 🔗 Stable Diffusion WebUI: <https://github.com/AUTOMATIC1111/stable-diffusion-webui>
- 🔗 Midjourney Documentation: <https://docs.midjourney.com/>
- 🔗 Prompt Guide: <https://prompthero.com/stable-diffusion-prompt-guide>
- 🔗 Lexica (Prompt Search): <https://lexica.art/>

✓ Instructor Checklist

- Diffusion process explained clearly
- 20+ quality images generated
- Style presets working
- Best images selected for LinkedIn
- Access to image generation tools setup
- Prompt library created
- Gallery interface functional

► Session Objectives

Build commercial image generation platforms, Implement batch processing for images, Create image editing and refinement workflows, Develop copyright-safe image generation

► Detailed Flow

Hour 1: Commercial Image Generation (0:00 – 1:00)

- Building scalable image generation APIs (20 min)
- Implementing batch processing pipelines (15 min)
- Creating commercial licensing systems (15 min)
- Ensuring copyright compliance (10 min)

Hour 2: Image Refinement Workflows (1:00 – 2:00)

- Building automated image enhancement workflows (20 min)
- Implementing style consistency across images (15 min)
- Creating image upscaling and refinement (15 min)
- Building quality control systems (10 min)

Hour 3: Production Deployment (2:00 – 3:00)

- Deploying image generation at scale (20 min)
- Implementing cost optimization for generation (15 min)
- Building monitoring for generation quality (15 min)
- Creating user management and billing (10 min)



Project of the Day (Continued)

✓ GitHub Push Strategy

Repository: commercial-image-platform

Commit Message: "Day 24: Commercial image generation platform with batch processing and licensing"

Include: api_platform/, batch_processing/, licensing/, quality_control/

✓ LinkedIn Post Template

🚀 Day 24: Building Commercial Image Generation Platforms!

Created a production-ready image generation platform for businesses!

Commercial Features:

- 💰 Batch processing for 1000+ images/day
- ⚡ Scalable API with rate limiting
- 📝 Commercial licensing system
- 🔍 Quality control and copyright compliance

Business Applications:

- Marketing asset generation
- Product image creation
- Social media content automation
- E-commerce image generation

This demonstrates how to monetize AI image generation at scale!

Demo: [Platform Interface]

Source: [GitHub Repo]

#AIArt #CommercialAI #ImageGeneration #SaaS #TechBusiness
#GenerativeAI

✓ Instructor Checklist

- Commercial platform architecture designed
- Licensing system functional
- Batch processing implemented
- Quality control working

- Scalability testing conducted
- Cost optimization applied

Day 25: Advanced Image Generation – ControlNet & Inpainting

Hours: 3

► Session Objectives

Learn ControlNet for precise control, Master inpainting and outpainting, Implement image-to-image generation, Build advanced image editing tools

► Detailed Flow

Hour 1: ControlNet Introduction (0:00 - 1:00)

- What is ControlNet? Control methods (20 min)
- Canny edge, depth maps, pose detection (20 min)
- Use cases for controlled generation (10 min)
- Setup and installation (10 min)

Hour 2: Inpainting & Outpainting (1:00 - 2:00)

- Inpainting theory and applications (15 min)
- Mask creation and editing (15 min)
- Outpainting to extend images (15 min)
- Image-to-image transformations (15 min)

Hour 3: Build Advanced Image Editor (2:00 - 3:00)

- Create canvas-based editing interface (20 min)
- Implement inpainting tool (20 min)
- Add ControlNet integration (15 min)
- Export and save functionality (5 min)

✓ Project of the Day

AI Image Editor Pro – Advanced editor with inpainting, outpainting, and ControlNet

✓ GitHub Push Strategy

Repository: ai-image-editor-pro

Commit Message: "Day 25: Built advanced AI image editor with ControlNet and inpainting capabilities"

Include: editor.py, controlnet_models/, masks/, app.py, examples/, README.md

✓ LinkedIn Post Template

✍️ Day 25: Advanced AI Image Manipulation!

Built an AI Image Editor Pro with ControlNet and Inpainting!

Advanced Features:

- 🎯 ControlNet for precise control (pose, depth, canny)
- ✨ Smart inpainting to fix/modify images
- 🖼️ Outpainting to expand image boundaries
- 🔄 Image-to-image transformations

Transformed 30+ images today - removing objects, changing backgrounds, extending scenes, and more!

Before/After: [Comparison Images]

Try it: [Demo Link]

Code: [GitHub Repo]

#ControlNet #ImageEditing #AITools #ComputerVision #Inpainting
#PhotoEditing

[Before/After comparison grid]

✓ Resources

- 🔗 ControlNet Paper: <https://arxiv.org/abs/2302.05543>
- 🔗 ControlNet Guide: <https://stable-diffusion-art.com/controlnet/>
- 🔗 Inpainting Tutorial: <https://huggingface.co/docs/diffusers/using-diffusers/inpaint>
- 🔗 Image Editing with AI: <https://www.creativebloq.com/features/ai-image-editing>



Instructor Checklist

- ControlNet models downloaded
- Inpainting producing quality results
- Canvas interface intuitive
- Tutorial documentation complete
- Edge/depth/pose detection working
- Outpainting functional
- Before/after examples compelling

Day 26: Professional Image Editing Platform

Hours: 3

▶ Session Objectives

Build professional image editing suite, Implement batch image processing, Create collaborative editing features, Develop commercial image editing workflows

▶ Detailed Flow

Hour 1: Professional Editing Suite (0:00 - 1:00)

- Building comprehensive editing tools interface (20 min)
- Implementing layer-based editing (15 min)

Hour 2: Batch & Collaborative Features (1:00 - 2:00)

- Implementing batch processing workflows (20 min)
- Building collaborative editing features (15 min)

- Creating preset and template systems (15 min)
- Building export optimization features (10 min)

- Creating version control for edits (15 min)
- Implementing team workflows (10 min)

Hour 3: Commercial Workflows (2:00 – 3:00)

- Building commercial licensing into editor (20 min)
- Creating client delivery systems (15 min)
- Implementing watermarking and protection (15 min)
- Building analytics for editor usage (10 min)

✓ Project of the Day (Continued)

Professional AI Image Editing Platform

✓ GitHub Push Strategy

Repository: professional-image-editor

Commit Message: "Day 26: Professional image editing platform with collaboration and commercial features"

Include: editor_suite/, batch_processing/, collaboration/, commercial_features/

✓ LinkedIn Post Template



Day 26: Building Professional Image Editing Platforms!

Created a commercial-grade AI image editing platform for creative professionals!

Professional Features:

- 🎨 Layer-based editing with advanced tools
- 🤝 Collaborative editing for teams

59

- 📦 Batch processing for efficiency
- 💰 Commercial licensing and client management

Target Users:

- Marketing agencies
- E-commerce businesses
- Content creators
- Design studios

This platform brings professional-grade AI editing to businesses of all sizes!

Demo: [Editing Platform]

Source: [GitHub Repo]

#ImageEditing #CreativeTools #SaaS #DesignTools #AIEditioning
#ProfessionalServices

✓ Instructor Checklist

- Professional editing suite complete
- Collaboration features working
- Client management system built
- Batch processing functional
- Commercial licensing implemented
- Analytics tracking operational

Day 27: AI Video Generation – Introduction

Hours: 3

▶ Session Objectives

Understand text-to-video generation, Explore video AI platforms, Create animated videos from text, Build video automation workflows

▶ Detailed Flow

60

Hour 1: Video Generation Landscape (0:00 - 1:00)

- Text-to-video vs image-to-video (15 min)
- Platforms: Runway, Pika, Synthesia overview (20 min)
- Video AI capabilities and limitations (15 min)
- Use cases: marketing, education, entertainment (10 min)

Hour 2: Hands-on Video Generation (1:00 - 2:00)

- Using Runway Gen-2 or free alternatives (20 min)
- Prompt engineering for videos (15 min)
- Camera movements and transitions (15 min)
- Exporting and post-processing (10 min)

Hour 3: Build Video Automation Tool (2:00 - 3:00)

- Script-to-video concept (15 min)
- Integrate AI for script writing (15 min)
- Automate video generation pipeline (20 min)
- Add music and effects (10 min)



Project of the Day

AI Video Story Generator - Tool that creates short videos from text stories



GitHub Push Strategy

Repository: ai-video-generator

Commit Message: "Day 27: Built AI video story generator from text to video pipeline"

Include: generator.py, scripts/, videos/, requirements.txt, README.md, examples/



LinkedIn Post Template

61

Created an AI Video Story Generator - text to cinematic video!

Today's Breakthroughs:

-  Text-to-video generation
-  Camera movements and transitions
-  Style control and consistency
-  Automated video production pipeline

Generated 10+ short video clips from story prompts! The future of content creation is here.

Watch samples: [YouTube/Drive Link]

Source Code: [GitHub Repo]

#AIVideo #Runway #VideoGeneration #ContentCreation #GenerativeAI
#VideoProduction

[Video thumbnail or GIF preview]

Resources

-  Runway ML: <https://runwayml.com/>
-  Pika Labs: <https://pika.art/>
-  Video AI Guide: <https://www.synthesia.io/post/ai-video-generator>
-  ModelScope Text-to-Video: <https://huggingface.co/damo-vilab/text-to-video-ms-1.7b>

Instructor Checklist

- Access to video AI platforms
- Prompt techniques demonstrated
- Automation pipeline working
- Upload to video platform complete
- 10+ videos generated
- Quality assessment criteria defined
- Video samples compiled

Day 28: Professional Video Production Platform

Hours: 3

► Session Objectives

Build commercial video generation platform, Implement batch video processing, Create video editing and post-production workflows, Develop monetization strategies for video AI

► Detailed Flow

Hour 1: Commercial Video Platform Architecture (0:00 – 1:00)

- Designing scalable video generation infrastructure (20 min)
- Implementing queue management for video rendering (15 min)
- Building rendering farm optimization (15 min)
- Cost analysis for commercial video generation (10 min)

Hour 2: Advanced Video Editing Workflows (1:00 – 2:00)

- Building automated video editing pipelines (20 min)
- Implementing AI-powered video enhancement (15 min)
- Creating template systems for consistent branding (15 min)
- Adding automated subtitles and captions (10 min)

Hour 3: Monetization & Distribution (2:00 – 3:00)

- Implementing pricing models for video generation (20 min)
- Building distribution to social platforms (15 min)
- Creating analytics for video performance (15 min)
- Developing API for third-party integration (10 min)

✓ Project of the Day (Continued)

Commercial Video Generation Platform

✓ GitHub Push Strategy

Repository: commercial-video-platform

Commit Message: "Day 28: Commercial video generation platform with batch processing and monetization"

Include: video_platform/, batch_rendering/, monetization/, distribution/

✓ LinkedIn Post Template



Day 28: Building Commercial Video Generation Platforms!

Created a production-ready video generation platform for businesses!

Commercial Features:

- 💰 Batch processing for 100+ videos/day
- 🎬 Automated editing and enhancement
- 📊 Performance analytics dashboard
- 🌐 Multi-platform distribution

Business Applications:

- Social media content creation
- Marketing video production
- Educational content generation

- E-commerce product videos

This platform enables businesses to create professional videos at scale!

Demo: [Video Platform Interface]

Source: [GitHub Repo]

#VideoGeneration #CommercialAI #ContentCreation #VideoMarketing

#AIProduction

Instructor Checklist

- Commercial platform architecture designed
- Video enhancement workflows functional
- Distribution pipeline working
- Batch rendering implemented
- Monetization system built
- Analytics dashboard operational

Day 29: Synthesia & AI Avatars

Hours: 3

► Session Objectives

Master Synthesia platform, Create AI avatar videos, Build educational content with avatars, Integrate with automation workflows

► Detailed Flow

Hour 1: Synthesia Deep Dive (0:00 - 1:00)

- Synthesia platform overview (15 min)
- AI avatars and voices explained (15 min)

Hour 2: Creating Avatar Videos (1:00 - 2:00)

- Select and customize avatars (15 min)
- Write scripts for different use cases (20 min)

- Script writing for avatar videos (15 min)
- Customization options (backgrounds, gestures) (15 min)
- Generate multiple videos (15 min)
- Review and optimize (10 min)

Hour 3: Build Training Video Generator (2:00 - 3:00)

- Design course content template (15 min)
- Automate video generation from outline (20 min)
- Create series of training videos (20 min)
- Export and organize content (5 min)

✓ Project of the Day

AI Training Course Generator – Automated system to create training videos with AI avatars

✓ GitHub Push Strategy

Repository: ai-training-course-generator

Commit Message: "Day 29: Built automated training course generator using Synthesia AI avatars"

Include: course_generator.py, scripts/, templates/, videos/, README.md, api_integration.py

✓ LinkedIn Post Template

 Day 29: Creating AI Avatar Training Videos!

Built an AI Training Course Generator using Synthesia!

What I Created:

66

- 🎭 Customized AI avatar presenters
- 📚 5-video training course on Python basics
- 🌐 Multi-language support
- ⚡ Automated generation from course outlines

Created professional training content in 1 hour that would normally take days!

Sample Videos: [Video Links]

Generator Tool: [GitHub Repo]

Pros: Fast, multilingual, consistent

Cons: Limited customization, uncanny valley

#Synthesia #AIAvatars #EdTech #VideoLearning #AITools
#OnlineLearning

[Thumbnail of avatar from video]

✓ Resources

- 🔗 Synthesia: <https://www.synthesia.io/>
- 🔗 AI Avatar Guide: <https://www.synthesia.io/post/ai-avatar>
- 🔗 Video Script Writing: <https://blog.hubspot.com/marketing/video-script-template>
- 🔗 Synthesia API: <https://docs.synthesia.io/>

✓ Instructor Checklist

- Synthesia account setup (trial/educational)
- 5+ videos generated
- Multi-language tested
- Pros/cons analysis included
- Multiple avatars tested
- Scripts optimized for avatars
- Course series created

▶ Session Objectives

Build enterprise-grade avatar video platform, Implement multi-avatar video systems, Create interactive avatar experiences, Develop analytics for avatar video performance

▶ Detailed Flow

Hour 1: Enterprise Video Platform (0:00 – 1:00)

- Building scalable avatar video infrastructure (20 min)
- Implementing multi-avatar synchronization (15 min)
- Creating brand customization systems (15 min)
- Building approval workflows for corporate content (10 min)

Hour 2: Interactive Avatar Experiences (1:00 – 2:00)

- Implementing interactive Q&A avatars (20 min)
- Building personalized avatar messaging (15 min)
- Creating avatar-based training simulations (15 min)
- Developing real-time avatar interactions (10 min)

Hour 3: Analytics & Optimization (2:00 – 3:00)

- Building viewer engagement analytics (20 min)
- Implementing A/B testing for avatar performance (15 min)
- Creating content effectiveness metrics (15 min)
- Developing optimization recommendations (10 min)



Project of the Day (Continued)

✓ GitHub Push Strategy

Repository: enterprise-avatar-platform

Commit Message: "Day 30: Enterprise avatar video platform with interactivity and analytics"

Include: platform/, interactive_features/, analytics/, enterprise_features/

✓ LinkedIn Post Template

🚀 Day 30: Building Enterprise Avatar Video Solutions!

Created a comprehensive avatar video platform for corporate use!

Enterprise Features:

- 🏢 Multi-avatar corporate branding
- 🤝 Interactive training simulations
- 📊 Advanced engagement analytics
- 🔒 Enterprise security and compliance

Business Applications:

- Corporate training at scale
- Interactive customer service
- Personalized marketing messages
- Multi-language global communications

This platform brings professional avatar video production to enterprise level!

Demo: [Enterprise Dashboard]

Source: [GitHub Repo]

#AvatarAI #EnterpriseVideo #CorporateTraining #EdTech #BusinessAI

✓ Instructor Checklist

- Enterprise platform architecture designed
- Multi-avatar systems working
- Interactive features implemented
- Analytics dashboard functional

- Corporate compliance features added
- Scalability testing completed

Day 31: GitHub Copilot - AI Pair Programming

Hours: 3

► Session Objectives

Set up and configure GitHub Copilot, Master Copilot features and shortcuts, Write code faster with AI assistance, Understand limitations and best practices

► Detailed Flow

Hour 1: Copilot Setup & Basics (0:00 – 1:00)

- GitHub Copilot installation in VS Code (15 min)
- Understanding how Copilot works (15 min)
- Basic code suggestions and completions (15 min)
- Accepting, rejecting, and cycling suggestions (15 min)

Hour 2: Advanced Copilot Features (1:00 – 2:00)

- Comment-to-code generation (20 min)
- Auto-filling repetitive code patterns (15 min)
- Copilot Chat for explanations (15 min)
- Testing with Copilot (10 min)

Hour 3: Build Project with Copilot (2:00 – 3:00)

- Create a web scraper with Copilot assistance (25 min)
- Document code with AI-generated docstrings (15 min)
- Debug and optimize with Copilot (15 min)
- Compare: with vs without Copilot (5 min)

✓ Project of the Day

Web Scraper Tool – Built entirely with GitHub Copilot assistance, documented with AI

✓ GitHub Push Strategy

Repository: copilot-web-scraper

Commit Message: "Day 31: Built web scraper with GitHub Copilot – 3x faster development!"

Include: scraper.py, tests/, README.md, copilot_insights.md, requirements.txt

✓ LinkedIn Post Template

💡 Day 31: Coding with an AI Pair Programmer!

Built a complete web scraper with GitHub Copilot – development speed increased by 3x!

Copilot Superpowers:

- 💬 Comment-to-code generation
- 🔄 Auto-completing repetitive patterns
- 📝 AI-generated documentation
- 🐛 Smart debugging suggestions
- ✅ Test case generation

What took me 3 hours now takes 1 hour – game changer!

Project: [GitHub Repo]

Time Comparison: Solo coding vs Copilot-assisted

#GitHubCopilot #AIPairProgramming #CodeGeneration #DeveloperTools
#ProductivityHack

[Screenshot of Copilot in action]

✓ Resources

🔗 GitHub Copilot: <https://github.com/features/copilot>

🔗 Copilot Docs: <https://docs.github.com/en/copilot>

🔗 Best Practices: <https://github.blog/2023-06-20/how-to-write-better-prompts-for-github-copilot/>

🔗 Copilot Patterns: <https://www.youtube.com/watch?v=dhfTasGYQ4o>

Instructor Checklist

- All students have Copilot access
- Basic features demonstrated
- Complete project built
- Best practices document created
- IDE extensions installed
- Advanced features practiced
- Productivity metrics tracked

Day 32: Advanced GitHub Copilot & Code Review

Hours: 3

► Session Objectives

Navigate unfamiliar code with Copilot, Use Copilot for code review, Test generation and debugging, Build complex application with AI

► Detailed Flow

Hour 1: Exploring Unfamiliar Territory (0:00 - 1:00)

- Using Copilot in new codebases (20 min)
- Explaining complex code with AI (15 min)
- Refactoring suggestions from Copilot (15 min)
- Learning new frameworks faster (10 min)

Hour 2: Testing & Quality Assurance (1:00 - 2:00)

- Generating unit tests with Copilot (20 min)
- Creating test data and fixtures (15 min)
- Code review with AI assistance (15 min)
- Security vulnerability detection (10 min)

Hour 3: Build Full-Stack App (2:00 - 3:00)

- Design a todo app (backend + frontend) (10 min)
- Build backend API with Copilot (20 min)
- Create frontend with AI assistance (20 min)
- Test and deploy (10 min)



Project of the Day

AI-Assisted Full-Stack Todo App - Complete CRUD application built with Copilot



GitHub Push Strategy

Repository: copilot-fullstack-todo

Commit Message: "Day 32: Built full-stack todo app with GitHub Copilot - tests included!"

Include: backend/, frontend/, tests/, README.md, deployment.md, copilot_lessons.md



LinkedIn Post Template



Day 32: Building Full-Stack Apps with AI!

Created a complete full-stack Todo application with GitHub Copilot as my pair programmer!

What Copilot Helped With:

- ⚙️ Backend API (FastAPI/Flask)
- 🎨 Frontend UI (React/HTML+JS)
- ✅ Unit test generation (90% coverage!)
- 🐛 Bug detection and fixes
- 📦 Deployment configuration

Built in 3 hours what normally takes a full day!

Live Demo: [Deployment Link]

Source: [GitHub Repo]

The future of coding is collaborative AI!

#FullStackDevelopment #GitHubCopilot #WebDev #AITools #CRUD
#TestDrivenDevelopment

[Screenshot of the todo app]

✓ Resources

- 🔗 Testing with Copilot: <https://github.blog/2023-08-02/how-github-copilot-is-getting-better-at-understanding-your-code/>
- 🔗 Code Review Guide: <https://github.com/features/copilot/code-review>
- 🔗 FastAPI: <https://fastapi.tiangolo.com/>
- 🔗 React Quick Start: <https://react.dev/learn>

✓ Instructor Checklist

- Test generation working properly
- Backend API functional
- Tests passing
- Documentation comprehensive
- Code review features demonstrated
- Frontend connected to backend
- App deployed successfully

Day 33: Enterprise Copilot Integration

Hours: 3

► Session Objectives

Implement Copilot in team environments, Build custom Copilot extensions, Create Copilot analytics and monitoring, Develop Copilot training programs

► Detailed Flow

Hour 1: Team Integration Strategies (0:00 - 1:00)

- Implementing Copilot across development teams (20 min)
- Creating team coding standards with AI assistance (15 min)
- Building shared prompt libraries for teams (15 min)
- Managing code consistency with Copilot (10 min)

Hour 2: Custom Extensions & Integrations (1:00 - 2:00)

- Building custom Copilot extensions (20 min)
- Integrating Copilot with existing development tools (15 min)
- Creating domain-specific Copilot training (15 min)
- Implementing security scanning with Copilot (10 min)

Hour 3: Analytics & Optimization (2:00 - 3:00)

- Building Copilot usage analytics (20 min)
- Implementing productivity tracking (15 min)
- Creating quality improvement metrics (15 min)
- Developing optimization recommendations (10 min)



Project of the Day (Continued)

Enterprise Copilot Integration Platform



GitHub Push Strategy

Repository: enterprise-copilot-platform

Commit Message: "Day 33: Enterprise Copilot integration with team management and analytics"

Include: team_integration/, extensions/, analytics/, training_programs/



LinkedIn Post Template



Day 33: Scaling AI Pair Programming for Enterprises!

Built an enterprise Copilot integration platform for development teams!

Enterprise Features:

- 💻 Team-wide Copilot configuration management
- 🔧 Custom extensions for domain-specific code
- 📊 Productivity and quality analytics
- 👉 AI-assisted developer training programs

Business Impact:

- 40% faster onboarding for new developers
- 25% reduction in code review time
- Consistent coding standards across teams
- Real-time productivity monitoring

This is how large organizations scale AI-assisted development!

Demo: [Team Dashboard]

Source: [GitHub Repo]

#EnterpriseDev #GitHubCopilot #DeveloperProductivity
#TeamCollaboration #AIinDevelopment



Instructor Checklist

- Team integration strategies implemented
- Analytics system operational
- Security integration working
- Custom extensions functional
- Training programs developed
- Productivity metrics established

Day 34: Retrieval Augmented Generation (RAG) - Part 1

76

► Session Objectives

Understand RAG architecture, Learn vector databases, Implement document embedding, Build basic RAG system

► Detailed Flow

Hour 1: RAG Fundamentals (0:00 - 1:00)

- What is RAG? Why do we need it? (15 min)
- RAG architecture: Retrieval + Generation (20 min)
- Vector embeddings explained (15 min)
- Semantic search vs keyword search (10 min)

Hour 2: Vector Databases & Embeddings (1:00 - 2:00)

- Introduction to ChromaDB/Pinecone/FAISS (15 min)
- Creating embeddings with sentence transformers (20 min)
- Storing and indexing documents (15 min)
- Similarity search implementation (10 min)

Hour 3: Build Document Q&A System (2:00 - 3:00)

- Load and chunk documents (15 min)
- Create vector embeddings (10 min)
- Implement retrieval function (15 min)
- Generate answers with LLM (15 min)
- Testing with sample documents (5 min)



✓ GitHub Push Strategy

Repository: document-qa-rag

Commit Message: "Day 34: Built RAG-based document Q&A system with vector embeddings"

Include: rag_system.py, embeddings/, documents/, vector_db/, app.py, README.md

✓ LinkedIn Post Template

🔍 Day 34: Building RAG Systems - Chat with Documents!

Created a Document Q&A system using Retrieval Augmented Generation!

RAG Magic:

- 📚 Upload any document (PDF, TXT, etc.)
- 📅 Convert to vector embeddings
- 🔍 Semantic search for relevant chunks
- 🤖 Generate accurate answers with LLM

No more scrolling through 100-page PDFs - just ask questions!

Tested on: Research papers, legal docs, manuals

Try it: [Demo Link]

Code: [GitHub Repo]

#RAG #VectorDatabase #SemanticSearch #LLM #DocumentAI #NLP

[Screenshot of Q&A in action]

✓ Resources

🔗 RAG Paper: <https://arxiv.org/abs/2005.11401>

🔗 LangChain RAG:

https://python.langchain.com/docs/use_cases/question_answering/

🔗 ChromaDB: <https://www.trychroma.com/>

🔗 Sentence Transformers: <https://www.sbert.net/>

✓ Instructor Checklist

- RAG concept clearly explained
- Embeddings generating correctly
- LLM integration functional
- Multiple document types tested
- Vector DB setup complete
- Retrieval working accurately
- Q&A providing relevant answers

Day 35: RAG – Part 2 Advanced & Optimization

Hours: 3

▶ Session Objectives

Optimize RAG performance, Implement advanced retrieval strategies, Build multi-document RAG, Add conversation memory to RAG

▶ Detailed Flow

Hour 1: RAG Optimization (0:00 - 1:00)

- Chunking strategies and overlap (20 min)
- Embedding model selection (15 min)
- Reranking retrieved documents (15 min)
- Caching for performance (10 min)

Hour 2: Advanced RAG Techniques (1:00 - 2:00)

- Multi-query retrieval (15 min)
- Hybrid search (keyword + semantic) (20 min)
- Metadata filtering (15 min)
- Source attribution and citations (10 min)

Hour 3: Build Knowledge Base

Chatbot (2:00 – 3:00)

- Multi-document ingestion system (15 min)
- Conversational memory integration (15 min)
- Build web interface (20 min)
- Testing and refinement (10 min)



Project of the Day

Enterprise Knowledge Base Chatbot – Multi-document RAG with conversation history



GitHub Push Strategy

Repository: enterprise-knowledge-chatbot

Commit Message: "Day 35: Built advanced RAG chatbot with multi-document support and memory"

Include: chatbot.py, knowledge_base/, retriever.py, app.py, tests/, README.md



LinkedIn Post Template



Day 35: Advanced RAG – Enterprise Knowledge Systems!

Built an Enterprise Knowledge Base Chatbot with advanced RAG!

Advanced Features:

- 📄 Multi-document ingestion (PDFs, DOCS, web pages)
- 🧠 Conversation memory across sessions
- 🎯 Hybrid search (semantic + keyword)
- 📊 Source attribution with citations
- ⚡ Optimized chunking and reranking

Loaded 50+ documents – answers questions with 95% accuracy and provides sources!

Demo: [Video Link]

Code: [GitHub Repo]

Perfect for: Customer support, internal docs, research

#RAG #EnterpriseAI #KnowledgeBase #Chatbot #DocumentAI #AISearch

[Screenshot showing source citations]

✓ Resources

- 🔗 Advanced RAG: <https://www.anthropic.com/index/contextual-retrieval>
- 🔗 LlamaIndex: <https://docs.llamaindex.ai/en/stable/>
- 🔗 Reranking Models:
<https://www.sbert.net/examples/applications/cross-encoder/README.html>
- 🔗 RAG Evaluation: <https://docs.ragas.io/en/latest/>

✓ Instructor Checklist

- Chunking optimized for better retrieval
- Reranking improving results
- Conversation memory working
- Performance metrics tracked
- Hybrid search implemented
- Citations showing correct sources
- Multi-document handling robust

Day 36: Production RAG Systems

Hours: 3

► Session Objectives

Build production-ready RAG systems, Implement RAG monitoring and analytics, Create RAG optimization pipelines, Develop enterprise RAG deployment strategies

▶ Detailed Flow

Hour 1: Production Architecture (0:00 - 1:00)

- Designing scalable RAG infrastructure (20 min)
- Implementing caching and load balancing (15 min)
- Building document versioning systems (15 min)
- Creating backup and recovery strategies (10 min)

Hour 2: Monitoring & Optimization (1:00 - 2:00)

- Building RAG performance monitoring (20 min)
- Implementing automated optimization pipelines (15 min)
- Creating quality assurance workflows (15 min)
- Developing cost optimization strategies (10 min)

Hour 3: Enterprise Deployment (2:00 - 3:00)

- Deploying RAG systems at scale (20 min)
- Implementing security and access controls (15 min)
- Building integration with existing systems (15 min)
- Creating maintenance and update procedures (10 min)



Project of the Day (Continued)

Production RAG Platform



GitHub Push Strategy

Repository: production-rag-platform

Commit Message: "Day 36: Production RAG platform with monitoring, optimization, and scaling"

Include: production_platform/, monitoring_system/, optimization/, enterprise_deployment/



LinkedIn Post Template



Day 36: Building Production RAG Systems at Scale!

Created an enterprise-grade RAG platform ready for production deployment!

Production Features:

- 🏢 Scalable architecture for millions of documents
- 📊 Real-time performance monitoring
- ⚡ Automated optimization pipelines
- 🔒 Enterprise security and compliance

Performance Metrics:

- 99.9% uptime SLA
- <1 second response time
- 95% answer accuracy
- Automatic scaling based on load

This platform demonstrates how to take RAG from prototype to production!

Demo: [Production Dashboard]

Source: [GitHub Repo]

#RAG #ProductionAI #EnterpriseAI #KnowledgeManagement #AISystems



Instructor Checklist

- Production architecture designed
- Optimization pipelines working
- Scalability testing completed
- Monitoring system operational
- Security measures implemented
- Deployment procedures documented

Day 37: Model Context Protocol (MCP) - Introduction

Hours: 3

83

► Session Objectives

Understand MCP architecture, Learn server and client concepts, Build first MCP server, Connect AI models to external tools

► Detailed Flow

Hour 1: MCP Fundamentals (0:00 - 1:00)

- What is Model Context Protocol? (15 min)
- MCP architecture: servers, clients, transports (20 min)
- Use cases: extending AI capabilities (15 min)
- Comparison with function calling/plugins (10 min)

Hour 2: Building MCP Server (1:00 - 2:00)

- MCP SDK installation and setup (15 min)
- Creating a simple MCP server (20 min)
- Defining tools and resources (15 min)
- Testing with MCP inspector (10 min)

Hour 3: Connect AI to MCP Server (2:00 - 3:00)

- Integrate MCP server with Claude/GPT (20 min)
- Build: Calculator + File System MCP (25 min)
- Testing tool execution (10 min)
- Error handling and debugging (5 min)



Project of the Day

File Management MCP Server – AI can read, write, and manage files through MCP



GitHub Push Strategy

Repository: mcp-file-manager

Commit Message: "Day 37: Built MCP server for file management with AI integration"

Include: server.py, client.py, tools/, README.md, examples/, tests/

✓ LinkedIn Post Template



Day 37: Extending AI with Model Context Protocol!

Built an MCP server that lets AI manage files on my computer!

MCP Capabilities:

- 🔧 Custom tools for AI models
- 📁 File system access (read/write/list)
- 🧮 Calculator operations
- 🔗 Connect AI to any external system
- 🔒 Secure and controlled access

AI can now perform real actions in the real world!

Demo: [Video of AI managing files]

Code: [GitHub Repo]

#MCP #AI #Claude #ToolUse #AIIntegration #Automation

[Screenshot of MCP server tools]

✓ Resources

- 🔗 MCP Documentation: <https://modelcontextprotocol.io/>
- 🔗 MCP Specification: <https://spec.modelcontextprotocol.io/>
- 🔗 MCP Python SDK: <https://github.com/anthropics/mcp-python-sdk>
- 🔗 Example Servers: <https://github.com/modelcontextprotocol/servers>

✓ Instructor Checklist

- MCP concepts explained clearly
- First MCP server running
- AI client connected
- SDK installed successfully
- Tools registered correctly
- Tool execution working

- Security considerations discussed

Day 38: MCP Advanced – Building Production Servers

Hours: 3

► Session Objectives

Build complex MCP servers, Implement authentication and security, Create multi-tool MCP systems, Deploy MCP servers

► Detailed Flow

Hour 1: Advanced MCP Patterns (0:00 – 1:00)

- Complex tool implementations (20 min)
- Resource management in MCP (15 min)
- Prompt templates in MCP (15 min)
- State management and context (10 min)

Hour 2: Security & Production Readiness (1:00 – 2:00)

- Authentication and authorization (20 min)
- Rate limiting and error handling (15 min)
- Logging and monitoring (15 min)
- Deployment strategies (10 min)

Hour 3: Build Database MCP Server (2:00 – 3:00)

- Design database query MCP (15 min)
- Implement CRUD operations as tools (25 min)
- Add data visualization tools (15 min)
- Testing and security audit (5 min)



Project of the Day

Database Query MCP Server – AI can query and analyze databases securely



GitHub Push Strategy

Repository: mcp-database-server

Commit Message: "Day 38: Built production-ready MCP database server with security"

Include: server.py, auth/, db_tools/, monitoring/, README.md, deployment.md



LinkedIn Post Template

87

Day 38: Production-Ready MCP Servers!

Built a secure Database MCP Server - AI can now query and analyze data!

Production Features:

- 🔒 Authentication & authorization
- 📊 SQL query execution
- 📈 Data visualization tools
- ⚠️ Rate limiting & error handling
- 📝 Comprehensive logging
- 🚀 Deployment-ready

AI can now answer questions like: "Show me top 10 customers by revenue" and get real data!

Architecture: [Diagram Link]

Code: [GitHub Repo]

#MCP #Database #AITools #ProductionAI #DataAnalysis #Security

[Screenshot of AI querying database]

✓ Resources

- 🔗 MCP Best Practices: <https://modelcontextprotocol.io/docs/best-practices>
- 🔗 Security Guide: <https://modelcontextprotocol.io/docs/security>
- 🔗 MCP Deployment: <https://modelcontextprotocol.io/docs/deployment>
- 🔗 Database Connectors: <https://www.sqlalchemy.org/>

✓ Instructor Checklist

- Advanced patterns demonstrated
- Rate limiting working
- Database queries executing safely
- Deployment documentation complete
- Authentication implemented
- Error handling comprehensive
- Logs being captured

Day 39: n8n + AI Agents - Building Intelligent Workflows

Hours: 3

► Session Objectives

Create AI agents in n8n, Build autonomous decision-making workflows, Implement agent loops and reasoning, Connect multiple AI services

► Detailed Flow

Hour 1: AI Agents Concepts (0:00 - 1:00)

- What are AI agents? (15 min)
- Agent frameworks: ReAct, Plan-and-Execute (20 min)
- Building agents in n8n (15 min)
- Use cases for autonomous agents (10 min)

Hour 2: Building Agent Workflows (1:00 - 2:00)

- Create decision-making logic (20 min)
- Implement agent tools in n8n (20 min)
- Add reasoning and memory (15 min)
- Error recovery and fallbacks (5 min)

Hour 3: Build Research Agent (2:00 - 3:00)

- Design autonomous research workflow (15 min)
- Implement: search → analyze → summarize (25 min)
- Add output formatting (10 min)
- Test with complex queries (10 min)

✓ Project of the Day

AI Research Agent - Autonomous agent that researches topics and creates reports

✓ GitHub Push Strategy

Repository: n8n-research-agent

Commit Message: "Day 39: Built autonomous research agent with n8n and AI"

Include: workflows/, agent_config.json, tools/, reports/, README.md

✓ LinkedIn Post Template

🤖 Day 39: Building Autonomous AI Agents!

Created a Research Agent that autonomously researches any topic and creates comprehensive reports!

Agent Capabilities:

- 🔍 Web search for information
- 📊 Data analysis and synthesis
- 🖨 Report generation
- 🔄 Iterative refinement
- 🧠 Decision-making logic

Gave it "Research latest AI trends" - it searched 20+ sources, analyzed patterns, and created a 5-page report. All automatically!

Watch it work: [Demo Video]

Workflow: [GitHub Repo]

#AIAgents #n8n #Automation #AutonomousAI #ResearchAutomation

[Screenshot of workflow diagram]

✓ Resources

🔗 AI Agents Guide: <https://www.anthropic.com/research/building-effective-agents>

🔗 n8n AI Agent: <https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/>

🔗 LangChain Agents:

<https://python.langchain.com/docs/modules/agents/>

🔗 ReAct Paper: <https://arxiv.org/abs/2210.03629>

✓ Instructor Checklist

- Agent concepts explained with examples
- Tools properly configured
- Research quality high
- Error handling robust
- Decision-making logic clear
- Agent loops functioning
- Reports well-formatted

Day 40: Multi-Agent Systems & Orchestration

Hours: 3

▶ Session Objectives

Design multi-agent architectures, Implement agent-to-agent communication, Build specialized agent teams, Create complex autonomous systems

▶ Detailed Flow

Hour 1: Multi-Agent Architecture (0:00 – 1:00)

- Single vs multi-agent systems (15 min)
- Agent roles and specialization (20 min)
- Communication patterns (15 min)
- Orchestration strategies (10 min)

Hour 2: Building Agent Teams (1:00 – 2:00)

- Define agent responsibilities (15 min)
- Implement inter-agent messaging (20 min)
- Create supervisor agent (15 min)
- Test agent collaboration (10 min)

Hour 3: Build Content Creation

Team (2:00 – 3:00)

- Design: Researcher + Writer + Editor agents (15 min)
- Implement workflow orchestration (25 min)
- Create blog post generation pipeline (15 min)
- Quality review and output (5 min)



Project of the Day

AI Content Creation Team – Multi-agent system: researcher, writer, editor working together



GitHub Push Strategy

Repository: multi-agent-content-team

Commit Message: "Day 40: Built multi-agent system with researcher, writer, and editor"

Include: agents/, orchestrator.py, workflows/, outputs/, README.md, architecture.md



LinkedIn Post Template



Day 40: Multi-Agent Systems – AI Teamwork!

Built a Content Creation Team with 3 specialized AI agents working together!

The Team:

📝 Researcher Agent: Gathers information

✍️ Writer Agent: Creates draft content

✅ Editor Agent: Reviews and refines

Gave them a topic → they researched, wrote, and edited a complete blog post autonomously!

The future: Specialized AI teams, not single models

Article: [Generated Content Link]

System: [GitHub Repo]

#MultiAgent #AITeams #ContentCreation #Orchestration #AutonomousAI

[Diagram of agent collaboration]

✓ Resources

- 🔗 Multi-Agent Systems: <https://arxiv.org/abs/2308.08155>
- 🔗 AutoGen: <https://microsoft.github.io/autogen/>
- 🔗 CrewAI: <https://docs.crewai.com/>
- 🔗 Agent Communication:
<https://www.anthropic.com/research/building-effective-agents>

✓ Instructor Checklist

- Multi-agent architecture designed
 - Agent roles clearly defined
- Communication working properly
 - Orchestration logic functional
- Output quality high
 - Collaboration visible in logs
- Architecture documented

Day 41: Bias, Fairness & AI Ethics

Hours: 3

► Session Objectives

Understand bias in AI models, Learn fairness evaluation techniques, Implement bias detection tools, Build responsible AI systems

► Detailed Flow

93

Hour 1: AI Bias & Ethics (0:00 - 1:00)

- Types of bias: data, algorithmic, societal (20 min)
- Real-world bias examples and impacts (20 min)
- Fairness definitions and metrics (15 min)
- Ethical AI principles (5 min)

Hour 2: Bias Detection & Mitigation (1:00 - 2:00)

- Detecting bias in datasets (20 min)
- Testing model fairness (20 min)
- Mitigation strategies (15 min)
- Transparency and explainability (5 min)

Hour 3: Build Bias Detector Tool (2:00 - 3:00)

- Create bias testing framework (20 min)
- Implement fairness metrics (20 min)
- Build dashboard for results (15 min)
- Generate bias report (5 min)



Project of the Day

AI Bias Detector – Tool to analyze and report bias in AI models and datasets



GitHub Push Strategy

Repository: ai-bias-detector

Commit Message: "Day 41: Built bias detection tool with fairness metrics and reporting"

Include: detector.py, metrics/, tests/, reports/, README.md, ethics_guide.md



LinkedIn Post Template

94

Created a Bias Detector tool to evaluate fairness in AI systems!

What It Analyzes:

-  Dataset bias (gender, race, age, etc.)
-  Model performance across groups
-  Fairness metrics (demographic parity, equal opportunity)
-  Comprehensive bias reports
-  Mitigation recommendations

Tested on 3 datasets - found significant bias in 2 of them!

Responsible AI isn't optional, it's essential.

Tool: [\[GitHub Repo\]](#)

Ethics Guide: [\[Documentation\]](#)

#AIEthics #ResponsibleAI #FairnessInAI #BiasDetection #EthicalAI
#AIGovernance

[Chart showing bias metrics]

Resources

-  AI Ethics: <https://www.microsoft.com/en-us/ai/responsible-ai>
-  Fairness Indicators:
https://www.tensorflow.org/responsible_ai/fairness_indicators/guide
-  AI Bias Research: <https://arxiv.org/abs/1908.09635>
-  Ethics Guidelines: <https://www.anthropic.com/index/core-views-on-ai-safety>

Instructor Checklist

- Bias types explained with examples
- Detection methods working
- Ethical discussions held
- Real-world cases analyzed
- Fairness metrics understood
- Reports comprehensive
- Mitigation strategies clear

Day 42: Model Evaluation & Performance Metrics

Hours: 3

▶ Session Objectives

Learn model evaluation techniques, Understand LLM-specific metrics, Implement automated testing, Build evaluation pipelines

▶ Detailed Flow

Hour 1: Evaluation Fundamentals (0:00 - 1:00)

- Traditional ML metrics (accuracy, F1, etc.) (15 min)
- LLM evaluation challenges (20 min)
- Metrics: perplexity, BLEU, ROUGE, BERTScore (15 min)
- Human evaluation vs automated (10 min)

Hour 2: LLM Testing Strategies (1:00 - 2:00)

- Creating test datasets (20 min)
- Prompt testing and A/B comparison (20 min)
- Regression testing for LLMs (15 min)
- Cost and latency monitoring (5 min)

Hour 3: Build Evaluation Suite (2:00 - 3:00)

- Create automated test runner (20 min)
- Implement multiple evaluation metrics (20 min)
- Build comparison dashboard (15 min)
- Generate evaluation reports (5 min)

✓ Project of the Day

LLM Evaluation Suite - Automated testing and evaluation framework for AI models

✓ GitHub Push Strategy

Repository: llm-evaluation-suite

Commit Message: "Day 42: Built comprehensive LLM evaluation suite with multiple metrics"

Include: evaluator.py, tests/, metrics/, dashboard.py, reports/, README.md

✓ LinkedIn Post Template

📊 Day 42: Evaluating AI Model Performance!

Built an LLM Evaluation Suite to test AI models scientifically!

Evaluation Features:

- ✓ 10+ automated metrics (BLEU, ROUGE, BERTScore, etc.)
- 📈 A/B comparison of prompts and models
- ⚡ Performance monitoring (latency, cost)
- ⌚ Regression testing
- 📊 Visual dashboards and reports

Tested 5 different prompts - found 40% performance variation!

Measuring > Guessing

Framework: [GitHub Repo]

Results: [Evaluation Report]

#LLMEvaluation #AITesting #ModelPerformance #MLOps #AIMetrics
#DataScience

[Comparison chart of different approaches]

✓ Resources

🔗 LLM Evaluation: <https://huggingface.co/spaces/evaluate-metric/evaluate-metric>

🔗 BLEU Score: <https://www.nltk.org/api/nltk.translate.html>

🔗 BERTScore: https://github.com/Tiiiger/bert_score

🔗 Evaluation Metrics: <https://arxiv.org/abs/2004.14820>

✓ Instructor Checklist

- Metrics explained clearly
- Test datasets created
- Automated testing working
- Multiple metrics implemented
- Dashboard functional
- Reports informative
- Best practices documented

Day 43: Fine-Tuning Introduction & Concepts

Hours: 3

▶ Session Objectives

Understand pre-training vs fine-tuning, Learn when to fine-tune vs prompt engineer, Prepare datasets for fine-tuning, Understand fine-tuning costs and benefits

▶ Detailed Flow

Hour 1: Fine-Tuning Fundamentals (0:00 - 1:00)

- Pre-training vs fine-tuning explained (20 min)
- When to fine-tune: use cases (15 min)
- Fine-tuning vs RAG vs prompt engineering (15 min)
- Types: full fine-tuning, LoRA, QLoRA (10 min)

Hour 2: Dataset Preparation (1:00 - 2:00)

- Data collection and cleaning (20 min)
- Creating training/validation splits (15 min)
- Formatting for fine-tuning (JSON, JSONL) (15 min)
- Data quality assessment (10 min)

Hour 3: Fine-Tuning Demo (2:00 - 3:00)

- Using OpenAI fine-tuning API (or Hugging Face) (25 min)
- Monitoring training progress (15 min)
- Evaluating fine-tuned model (15 min)
- Cost analysis (5 min)



Project of the Day

Fine-Tuned Customer Support Bot – Model fine-tuned on customer support conversations



GitHub Push Strategy

Repository: finetuned-support-bot

Commit Message: "Day 43: Fine-tuned model for customer support with curated dataset"

Include: dataset/, training_script.py, evaluation.py, model_info.md, README.md



LinkedIn Post Template

🎯 Day 43: Fine-Tuning AI Models!

Fine-tuned a GPT model for customer support – specialized AI performance!

Fine-Tuning Journey:

- 📚 Curated 500+ support conversations
- 🔧 Formatted training data
- ⚙️ Trained custom model
- 📊 Evaluated: 30% improvement over base model
- 💰 Cost analysis: \$15 for training

When to fine-tune vs prompt engineer? Now I know!

Results: [Comparison Data]

Guide: [GitHub Repo]

#FineTuning #LLM #MachineLearning #CustomAI #ModelTraining
#AIOptimization

[Before/After comparison chart]

✓ Resources

- 🔗 Fine-Tuning Guide: <https://platform.openai.com/docs/guides/fine-tuning>
- 🔗 LoRA Paper: <https://arxiv.org/abs/2106.09685>
- 🔗 Hugging Face Training: <https://huggingface.co/docs/transformers/training>
- 🔗 Dataset Best Practices: <https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset>

✓ Instructor Checklist

- Fine-tuning concepts clear
- Decision framework provided
- Dataset prepared properly
- Training initiated successfully
- Evaluation completed
- Cost breakdown provided
- Comparison with base model
- done

Day 44: Building Production AI Applications

Hours: 3

► Session Objectives

Learn production best practices, Implement error handling and monitoring, Add rate limiting and caching, Deploy scalable AI applications

▶ Detailed Flow

Hour 1: Production Considerations (0:00 - 1:00)

- Development vs production AI (15 min)
- Error handling strategies (20 min)
- Rate limiting and quotas (15 min)
- Monitoring and logging (10 min)

Hour 2: Optimization Techniques (1:00 - 2:00)

- Caching strategies for AI responses (20 min)
- Async processing and queues (20 min)
- Cost optimization techniques (15 min)
- Fallback mechanisms (5 min)

Hour 3: Deploy Production App (2:00 - 3:00)

- Containerize with Docker (15 min)
- Add monitoring and logging (15 min)
- Deploy to cloud (Render/Railway/etc.) (20 min)
- Load testing and optimization (10 min)



Project of the Day

Production-Ready AI API - Scalable, monitored, cached AI service



GitHub Push Strategy

Repository: production-ai-api

Commit Message: "Day 44: Built production-ready AI API with monitoring, caching, and deployment"

Include: api/, Dockerfile, monitoring/, tests/, deployment/, README.md, architecture.md



LinkedIn Post Template

101



Deployed a production-ready AI API with enterprise features!

Production Features:

- Error handling & retry logic
- Response caching (10x faster)
- Monitoring & logging (Prometheus)
- Rate limiting & authentication
- Dockerized deployment
- Auto-scaling ready

Went from dev prototype to production-grade in one session!

Live API: [Deployment URL]

Code: [GitHub Repo]

Docs: [API Documentation]

#ProductionAI #APIDevelopment #Docker #CloudDeployment #MLOps
#DevOps

[Architecture diagram]

✓ Resources

- Docker Tutorial: <https://docs.docker.com/get-started/>
- FastAPI Production: <https://fastapi.tiangolo.com/deployment/>
- Monitoring: <https://prometheus.io/docs/introduction/overview/>
- Railway Deployment: <https://docs.railway.app/>

✓ Instructor Checklist

- Production patterns explained
- Caching working
- Docker container built
- Load testing completed
- Error handling implemented
- Monitoring setup
- Deployment successful

▶ Session Objectives

Review all projects built so far, Optimize GitHub profiles, Create comprehensive portfolio, Prepare project presentations

▶ Detailed Flow

Hour 1: GitHub Portfolio

Optimization (0:00 - 1:00)

- Profile README enhancement (20 min)
- Repository organization and cleanup (20 min)
- Adding badges, demos, screenshots (15 min)
- Creating project showcase (5 min)

Hour 2: LinkedIn Presence Building

(1:00 - 2:00)

- Create featured projects section (20 min)
- Write comprehensive skill endorsements (20 min)
- Build recommendation strategy (15 min)
- Network expansion techniques (5 min)

Hour 3: Build Portfolio Website v2

(2:00 - 3:00)

- Design enhanced portfolio (15 min)
- Add all 30 projects with demos (25 min)
- Implement search and filtering (15 min)
- Deploy and optimize (5 min)



Project of the Day

Enhanced AI Portfolio Hub - Comprehensive showcase of all projects

✓ GitHub Push Strategy

Repository: ai-portfolio-hub

Commit Message: "Day 45: Launched comprehensive AI portfolio with 30+ projects"

Include: index.html, projects/, assets/, styles/, README.md, about.md

✓ LinkedIn Post Template



Day 45: Portfolio Milestone!

Completed 30 AI projects in 45 days! Built my comprehensive AI portfolio.

Projects Include:

- 🤖 10+ AI applications
- 📊 5+ data analysis tools
- 🎨 Image & video generation systems
- ⚙️ Automation workflows
- 🔧 Developer tools

Every project:

- ✅ Deployed and live
- 📝 Fully documented
- 🔗 Open source
- 💡 Real-world applicable

Check out my portfolio: [Portfolio URL]

GitHub: [Profile Link]

The journey continues beyond the curriculum!

What should I build next?

#100DaysOfAI #Portfolio #OpenSource #AIProjects #GenAI

[Portfolio screenshot]

✓ Resources

- 🔗 GitHub Profile Guide: <https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github/profile>

104

- 🔗 Portfolio Inspiration: <https://github.com/emmabostian/developer-portfolios>
- 🔗 LinkedIn Optimization:
<https://www.linkedin.com/business/talent/blog/product-tips/linkedin-profile-summaries-that-we-love-and-how-to-boost-your-own>

✓ Instructor Checklist

- All repositories organized
- Portfolio website impressive
- Demo links working
- Presentations prepared
- READMEs comprehensive
- LinkedIn profiles optimized
- Screenshots compelling

COMPLETED 45-DAY CURRICULUM SUMMARY

Curriculum Structure Successfully Implemented

45 Days Total – Each with 3-hour sessions

Original Content Preserved – All provided topics covered

Enhanced Learning – Each topic gets dedicated attention

Professional Development – Daily LinkedIn posting and GitHub commits

Instructor Support – Clear checklists for each session

Key Features Delivered

- ✓ Day-by-day structure with clear objectives
- ✓ Hourly breakdowns for precise scheduling
- ✓ Project-based learning with daily GitHub commits
- ✓ Professional branding with LinkedIn templates
- ✓ Instructor support with comprehensive checklists
- ✓ Resource guides for further learning

-  Production-ready skills across all AI domains
-

Total Learning Outcomes

- 45 AI Projects completed and documented
 - 45 GitHub Repositories with professional code
 - 45 LinkedIn Posts building professional brand
 - Comprehensive Portfolio showcasing diverse AI skills
 - Production-ready experience across multiple AI domains
 - Industry-relevant skills in high-demand areas
-

Final Day 45 Deliverables

- Complete portfolio website with all projects
 - Optimized GitHub profile showcasing 45 projects
 - Enhanced LinkedIn presence with featured projects
 - Presentation-ready project documentation
 - Career-ready AI development skills
-

The 45-day curriculum is now complete with all sessions structured according to your specifications. Each day maintains the 3-hour format with detailed breakdowns, ensuring instructors have clear guidance while preserving all original content and expanding learning opportunities.

135-Hour GenAI, GitHub & LinkedIn Automation Curriculum | 45 Days Complete