

# ZERO TO HERO IN RETRIEVAL AUGMENTED GENERATION

Yogesh Haribhau Kulkarni



# Program at a Glance

## Zero to Hero in RAG - 15 Day Training



# From Zero to Hero in RAG

**Learning Path:** Python → DSA → System Design → NLP → RAG → Prod

## Program Overview

- ▶ Duration: 15 working days (3 weeks)
- ▶ Daily: 8 hours (4 learning + 4 coding)
- ▶ Target: Software engineers with basic programming knowledge

## What You'll Build

- ▶ 15 progressive projects
- ▶ 1 production capstone
- ▶ Complete end-to-end RAG

## Portfolio Highlights:

- ▶ Week 1: 5 Python projects + System design
- ▶ Week 2: NLP pipeline + ML model + Embeddings
- ▶ Week 3: Production RAG system with UI

# Program Architecture

## Phase 1: Foundations (Days 1-5)

- ▶ Python Fundas
- ▶ Advanced Python
- ▶ DSA Algorithms
- ▶ System Design

## Phase 2: NLP & ML (Days 6-10)

- ▶ spaCy NLP
- ▶ Text Processing
- ▶ Sentiment Analysis
- ▶ Word Embeddings

## Phase 3: RAG (Days 11-15)

- ▶ RAG from Scratch
- ▶ Docling Parsing
- ▶ LangChain RAG
- ▶ Streamlit UI

### Daily Structure:

- ▶ 09:00-13:00: Learning (concept + guided examples)
- ▶ 14:00-18:00: Coding (project implementation)
- ▶ Thursday 16:00-17:00: Weekly QnA Session

# Day 1: Python Fundamentals & OOP

## Morning Session (4 hrs)

- ▶ Variables & data types
- ▶ Control flow (if, loops)
- ▶ Functions & scope
- ▶ Object-oriented programming
- ▶ Inheritance & polymorphism

## Afternoon: Pick 1 Project

- ▶ Shopping Cart System
- ▶ Employment Hierarchy
- ▶ Library Management
- ▶ Game tic-tac-toe
- ▶ Banking System

## Resources (Free):

- ▶ Google's Python Class (<https://developers.google.com/edu/python>)
- ▶ Python Essentials (<https://pythoninstitute.org/>)
- ▶ Real Python OOP ([realpython.com](https://realpython.com))

**Difficulty:** Medium | **Time:** 4 hours | **Deliverable:** Working code

## Day 2: Advanced Python & File Handling

### Morning Session (4 hrs)

- ▶ Decorators
- ▶ Generators & iterators
- ▶ Context managers
- ▶ Exception handling
- ▶ File I/O, JSON, CSV

### Afternoon: Pick 1 Project

- ▶ Logging decorators
- ▶ CSV data processor
- ▶ File backup system
- ▶ Config file manager
- ▶ Validation decorators

### Resources (Free):

- ▶ Real Python Decorators ([realpython.com/decorators](https://realpython.com/decorators))
- ▶ W3Schools File Handling ([w3schools.com/python/file](https://w3schools.com/python/file))
- ▶ DataCamp Data Processing (freemium)

**Difficulty:** Medium | **Time:** 4 hours | **Deliverable:** Working code

## Day 3: Python Advanced Deep Dive

### Morning Session (4 hrs)

- ▶ Metaclasses
- ▶ Descriptors
- ▶ Comprehensions
- ▶ Async/await
- ▶ Memory management
- ▶ Performance optimization

### Resources (Free):

- ▶ Real Python Concurrency ([realpython.com/concurrency](https://realpython.com/concurrency))
- ▶ Fluent Python concepts ([github.com/fluentpython](https://github.com/fluentpython))
- ▶ Memory Management ([realpython.com/memory](https://realpython.com/memory))

### Afternoon: Pick 1 Project

- ▶ Async web scraper
- ▶ Lazy-loading properties
- ▶ Concurrent file processor
- ▶ Generator ETL pipeline
- ▶ Thread-safe cache

**Difficulty:** Medium | **Time:** 4 hours | **Deliverable:** Working code

## Day 4: Data Structures & Algorithms

### Morning Session (4 hrs)

- ▶ Arrays, linked lists
- ▶ Stacks, queues
- ▶ Sorting algorithms
- ▶ Searching algorithms
- ▶ Big O complexity
- ▶ Time/space analysis

### Resources (Free):

- ▶ W3Schools DSA ([w3schools.com/dsa](https://www.w3schools.com/dsa))
- ▶ GeeksforGeeks ([geeksforgeeks.org/dsa](https://www.geeksforgeeks.org/dsa))
- ▶ LeetCode Explore ([leetcode.com/explore](https://leetcode.com/explore))

### Afternoon: Pick 1 Project

- ▶ Linked list (LeetCode 206)
- ▶ 5 LeetCode problems
- ▶ Sorting algorithms
- ▶ Expression evaluation
- ▶ Hash map problems

**Difficulty:** Medium | **Time:** 4 hours | **Deliverable:** Working code



# Day 5: System Design Fundamentals

## Morning Session (4 hrs)

- ▶ Requirements clarification
- ▶ Scalability patterns
- ▶ Caching strategies
- ▶ Load balancing
- ▶ Database design
- ▶ CAP theorem

## Resources (Free):

- ▶ System Design Handbook ([algorithms.wtf](https://algorithms.wtf))
- ▶ DesignGurus Guide ([designgurus.io](https://designgurus.io))
- ▶ InterviewBit ([interviewbit.com](https://interviewbit.com))

## Afternoon: Pick 1 Project

- ▶ URL shortener (TinyURL)
- ▶ Real-time chat system
- ▶ API rate limiter
- ▶ Real-time leaderboard
- ▶ Notification system

**Difficulty:** Medium | **Time:** 4 hours | **Deliverable:** Working code

# Week 1 Milestone Assessment

## Learning Outcomes

- ▶ Python OOP mastery
- ▶ Advanced patterns
- ▶ DSA competency
- ▶ System thinking

## Success Criteria

- ▶ 5 projects completed
- ▶ Core concepts understood
- ▶ Code quality  $\geq 80\%$
- ▶ Ready for NLP phase

## Assessment Activities:

- ▶ Code review of all 5 projects
- ▶ Technical Q&A on concepts
- ▶ System design explanation
- ▶ Readiness verification

## Assessment Rubric:

- ▶ Functionality (50 pts) - All features working
- ▶ Code Quality (20 pts) - Clean, documented
- ▶ Design Principles (20 pts) - Architecture
- ▶ Presentation (10 pts) - Documentation quality

## Day 6: NLP Basics with spaCy

### Morning Session (4 hrs)

- ▶ NLP fundamentals
- ▶ spaCy pipeline
- ▶ Tokenization
- ▶ POS tagging
- ▶ Named Entity Recognition

### Resources (Free):

- ▶ spaCy Advanced Course ([course.spacy.io](https://course.spacy.io))
- ▶ Real Python spaCy ([realpython.com/spacy](https://realpython.com/spacy))
- ▶ GeeksforGeeks NLP ([geeksforgeeks.org/nlp](https://www.geeksforgeeks.org/nlp))

### Afternoon: Pick 1 Project

- ▶ Text tokenizer
- ▶ NER system
- ▶ Text similarity
- ▶ Dependency parser
- ▶ Entity linker

# Day 7: NLP Processing & Text Analysis

## Morning Session (4 hrs)

- ▶ Text preprocessing
- ▶ Lemmatization
- ▶ Stop word removal
- ▶ Text normalization
- ▶ TF-IDF vectorization
- ▶ Feature extraction

## Resources (Free):

- ▶ Real Python Text Processing ([realpython.com](https://realpython.com))
- ▶ NLTK Documentation ([nltk.org](https://www.nltk.org))
- ▶ DataCamp Text Analytics (freemium)

## Afternoon: Pick 1 Project

- ▶ Preprocessing pipeline
- ▶ TF-IDF vectorizer
- ▶ Advanced text cleaner
- ▶ Keyword extractor
- ▶ Document summarizer

## Day 8: NLP Advanced Topics

### Morning Session (4 hrs)

- ▶ Dependency parsing
- ▶ Syntax trees
- ▶ Coreference resolution
- ▶ Topic modeling
- ▶ Information extraction
- ▶ Semantic role labeling

### Resources (Free):

- ▶ Advanced spaCy Course ([course.spacy.io](https://course.spacy.io))
- ▶ Real Python NLP ([realpython.com](https://realpython.com))
- ▶ GeeksforGeeks Advanced ([geeksforgeeks.org/nlp](https://geeksforgeeks.org/nlp))

### Afternoon: Pick 1 Project

- ▶ Syntax parser
- ▶ Coreference resolution
- ▶ Relation extraction
- ▶ Semantic role labeler
- ▶ QA system (template-based)

# Day 9: Machine Learning + Sentiment Analysis

## Morning Session (4 hrs)

- ▶ Sentiment analysis approaches
- ▶ Feature engineering
- ▶ ML classifiers
- ▶ Model evaluation
- ▶ Imbalanced datasets
- ▶ Ensemble methods

## Resources (Free):

- ▶ DataCamp NLTK ([datacamp.com/tutorial/nltk](https://datacamp.com/tutorial/nltk))
- ▶ Real Python Sentiment ([realpython.com/sentiment](https://realpython.com/sentiment))
- ▶ Towards Data Science ([towardsdatascience.com](https://towardsdatascience.com))

## Afternoon: Pick 1 Project

- ▶ Lexicon-based analyzer
- ▶ ML classifier
- ▶ Ensemble model
- ▶ Real-time analyzer
- ▶ Aspect-based sentiment

# Day 10: Word Embeddings

## Morning Session (4 hrs)

- ▶ Word2Vec (CBOW, Skip-Gram)
- ▶ GloVe embeddings
- ▶ FastText
- ▶ Embedding visualization
- ▶ Pre-trained models
- ▶ Transfer learning

## Resources (Free):

- ▶ GeeksforGeeks Word2Vec ([geeksforgeeks.org](https://www.geeksforgeeks.org/))
- ▶ Milvus Embeddings ([milvus.io](https://milvus.io/))
- ▶ TensorFlow Word2Vec ([tensorflow.org](https://www.tensorflow.org/))

## Afternoon: Pick 1 Project

- ▶ Word2Vec training
- ▶ Analogy solver
- ▶ Document similarity
- ▶ t-SNE visualization
- ▶ Semantic search engine

# Week 2 Milestone Assessment

## Learning Outcomes

- ▶ NLP pipelines
- ▶ ML classification
- ▶ Embeddings mastery
- ▶ Semantic search

## Assessment Activities:

- ▶ NLP pipeline demo
- ▶ Sentiment model evaluation
- ▶ Embedding visualization
- ▶ Component integration check

## Success Criteria

- ▶ NLP pipeline functional
- ▶ ML model  $\geq 80\%$  accuracy
- ▶ Embedding system works
- ▶ Ready for RAG phase



# Day 11: RAG from Scratch

## Morning Session (4 hrs)

- ▶ RAG architecture
- ▶ Vector stores
- ▶ Chunking strategies
- ▶ Retrieval mechanisms
- ▶ Re-ranking
- ▶ Integration patterns

## Resources (Free):

- ▶ RAG from Scratch ([linkedin.com/pulse](https://www.linkedin.com/pulse/rag-from-scratch-yuhang-hu))
- ▶ HuggingFace RAG ([huggingface.co/blog](https://huggingface.co/blog))
- ▶ Glean RAG Guide ([glean.com/blog](https://glean.com/blog))

## Why RAG? Combines retrieval with generation for:

- ▶ Up-to-date information beyond training data
- ▶ Grounded, factual responses with citations
- ▶ Domain-specific knowledge integration

**Key Components:** Document Loading → Chunking → Embedding → Vector Store → Retrieval → LLM → Response

## Afternoon: Pick 1 Project

- ▶ Basic RAG system
- ▶ Document chunking
- ▶ Re-ranking retriever
- ▶ Full RAG pipeline
- ▶ Multi-doc RAG

## Day 12: Docling Document Parsing

### Morning Session (4 hrs)

- ▶ Docling architecture
- ▶ PDF/DOCX/PPTX parsing
- ▶ Layout analysis
- ▶ OCR capabilities
- ▶ Table extraction
- ▶ Image extraction

### Resources (Free):

- ▶ DataCamp Docling ([datacamp.com/tutorial/docling](https://datacamp.com/tutorial/docling))
- ▶ IBM Docling ([github.com/docling-project](https://github.com/docling-project))
- ▶ Geek Avenue ([youtube.com](https://youtube.com) - Docling Tutorial)

### Afternoon: Pick 1 Project

- ▶ Multi-format parser
- ▶ Table extractor
- ▶ Image extractor
- ▶ OCR processor
- ▶ Hierarchy extractor

# Day 13: LangChain RAG Implementation

## Morning Session (4 hrs)

- ▶ LangChain framework
- ▶ Document loaders
- ▶ Text splitters
- ▶ Embeddings integration
- ▶ Vector stores
- ▶ LCEL expressions

## Resources (Free):

- ▶ LangChain RAG ([python.langchain.com](https://python.langchain.com))
- ▶ Kody Simpson YouTube ([youtube.com](https://youtube.com))
- ▶ DataCamp RAG ([datacamp.com/courses](https://datacamp.com/courses))

## Afternoon: Pick 1 Project

- ▶ PDF Q&A system
- ▶ Web scraping RAG
- ▶ Multi-index RAG
- ▶ Conversation RAG
- ▶ Advanced RAG

# Day 14: Streamlit UI Development

## Morning Session (4 hrs)

- ▶ Streamlit basics
- ▶ Layout components
- ▶ Interactive widgets
- ▶ Data visualization
- ▶ File uploads
- ▶ State management

## Resources (Free):

- ▶ Streamlit Docs ([docs.streamlit.io](https://docs.streamlit.io))
- ▶ GeeksforGeeks Streamlit ([geeksforgeeks.org](https://www.geeksforgeeks.org))
- ▶ DataQuest Chatbot ([dataquest.io](https://dataquest.io))

## Afternoon: Pick 1 Project

- ▶ Interactive dashboard
- ▶ Chatbot interface
- ▶ Document uploader
- ▶ RAG interface
- ▶ Multi-page app

# Day 15: End-to-End Capstone Project

## Morning Session (4 hrs)

- ▶ Architecture planning
- ▶ Component integration
- ▶ Testing strategies
- ▶ Performance optimization
- ▶ Error handling
- ▶ Deployment prep

### Integration Requirements:

- ▶ Docling: Multi-format document parsing
- ▶ LangChain: RAG orchestration & chains
- ▶ Streamlit: Interactive web interface
- ▶ All components working together

## Pick 1 Project (Full Stack)

- ▶ Document Intelligence
- ▶ Advanced Multi-Doc RAG
- ▶ Enterprise Knowledge Base
- ▶ Real-Time Chat Analyzer
- ▶ AI Research Tool

**Capstone Evaluation (100 pts):** Functionality (40) | Code Quality (30)  
| Architecture (20) | Documentation (10) **Minimum Score:** 70/100 to pass

## Week 3 Milestone & Final Assessment

### Learning Outcomes

- ▶ Complete RAG systems
- ▶ Document processing
- ▶ LLM orchestration
- ▶ Production applications

### **Production Readiness Checklist:**

- ▶ Error handling implemented
- ▶ Performance optimized
- ▶ Documentation complete
- ▶ Deployment configuration ready

### **Final Assessment:**

- ▶ Capstone project demonstration
- ▶ Architecture explanation
- ▶ Code walkthrough
- ▶ Future enhancements discussion

### Success Criteria

- ▶ Capstone fully functional
- ▶ Code well-documented
- ▶ Architecture sound
- ▶ Production-ready

# Congratulations! Next Steps

## Technical Skills

- ▶ Advanced Python
- ▶ NLP & ML systems
- ▶ RAG architecture
- ▶ Full-stack development

## Career Pathways

- ▶ AI/ML Engineer
- ▶ RAG Specialist
- ▶ LLM Infrastructure
- ▶ Product Engineer (AI)

## Next Certifications:

- ▶ Azure AI Fundamentals (AI-900)
- ▶ LangChain Academy Certification

## Continuous Learning Resources:

- ▶ LangChain Academy ([academy.langchain.com](https://academy.langchain.com))
- ▶ DeepLearning.AI Courses ([deeplearning.ai](https://deeplearning.ai))
- ▶ Hugging Face Course ([huggingface.co/course](https://huggingface.co/course))
- ▶ GitHub: Explore open source RAG projects

**Keep Building, Keep Learning!**

