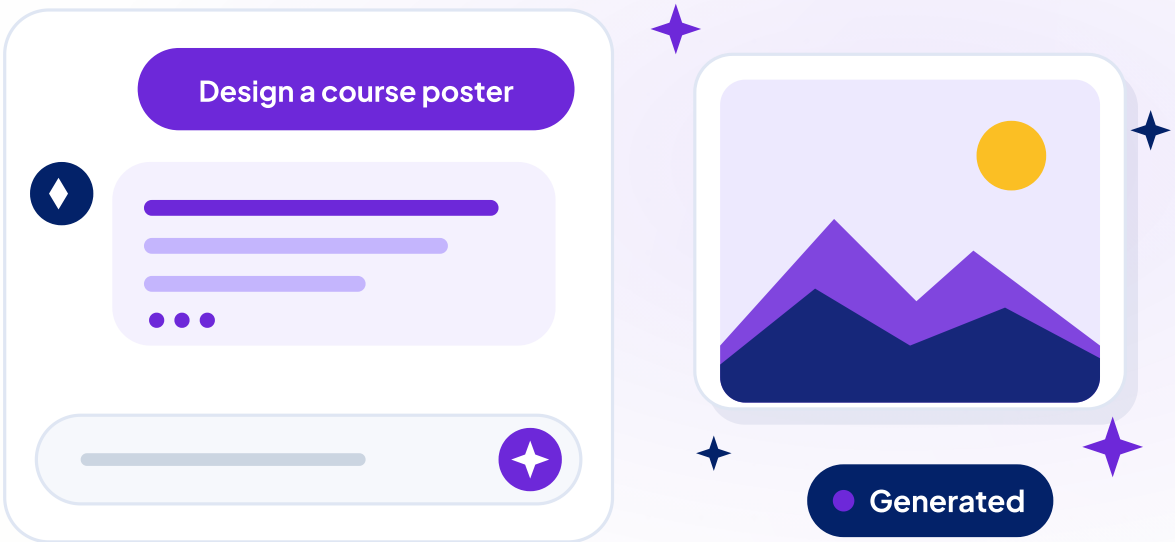
TECH DRONA
DEBUG YOUR CAREER. DEPLOY YOUR DREAMS.

Coming soon · 2026

AI COURSE

Generative AI

Build, evaluate and ship applications powered by large language models.



TECHNOLOGIES COVERED

Python · OpenAI · Gemini · Hugging Face · LangChain · Vector databases · Streamlit

50 hours	10	Labs + capstone
Duration	Modules	Projects

01 ABOUT THE COURSE

Course overview

A 50-hour course that goes from how large language models work to shipping real applications with them. Learners practise prompt engineering with hosted and open models, build apps with LangChain and Streamlit, and create retrieval-augmented (RAG) systems over their own documents. They finish by evaluating quality, knowing when to fine-tune and applying responsible-AI practices.

DURATION
50 hours
Training hours, including guided labs

MODULES
10
Each module ends with hands-on practice

PROJECTS
Labs + capstone
Built, reviewed and demonstrated

LEVEL
Beginner to intermediate

What you will learn

LLM foundations
Tokens · Transformers · Context windows · Model choice

Prompt engineering
Few-shot · Chain of thought · Structured output

LLM applications
LangChain · Output parsers · Memory · Streamlit

RAG and evaluation
Embeddings · Vector databases · RAG · RAGAS · LoRA

02 WHO IT IS FOR

Goals & outcomes

Who it is for

Students, fresh graduates, developers and analysts who want to build AI features into products.

Prerequisites

- Working Python: functions, lists, dictionaries and pip
- No machine-learning background is required

Learning objectives

- Explain how LLMs work, including tokens, transformers, context windows and hallucination.
- Write reliable prompts and get structured output from hosted and open models.
- Build LLM applications with LangChain and give them a Streamlit interface.
- Build, evaluate and improve a retrieval-augmented generation (RAG) system.
- Decide between prompting, RAG and fine-tuning, and apply responsible-AI practices.

Key highlights

- No prior AI background needed; working Python is enough.
- Hands-on with OpenAI, Gemini and open models on Hugging Face.
- Every module ends in a small working application.
- Builds the foundation for the Agentic AI course.

Expected outcomes

Build LLM apps
Design and build applications on hosted and open models.

Ground answers in data
Build RAG systems that cite their sources.

Measure quality
Evaluate outputs and improve them with evidence.

Use AI responsibly
Control cost, protect personal data and add guardrails.

SKILLS YOU BUILD

Prompt engineering · LLM APIs · LangChain · Embeddings · RAG · Streamlit · Evaluation

03

TEN MODULES, ONE EVALUATED CAPSTONE

Course curriculum

01 MODULE 1 — AI and Generative AI foundations

3 hrs

1.1 **The landscape**
AI vs machine learning vs deep learning · what makes AI generative · real-world use cases

1.2 **Model families**
foundation models · open-source vs closed-source LLMs · text, image, audio and multimodal models

02 MODULE 2 — How large language models work

4 hrs

2.1 **Inside an LLM**
tokens and tokenisation · embeddings · transformers and attention

2.2 **Training and behaviour**
pre-training vs instruction tuning · context windows · temperature and top-p · why models hallucinate

03 MODULE 3 — Working with model APIs

4 hrs

3.1 **Hosted models**
OpenAI · Google Gemini · Anthropic Claude · API keys and setup

3.2 **Open models**
Hugging Face · running models locally with Ollama

3.3 **Cost and limits**
token usage and monitoring · rate limits · streaming responses

04 MODULE 4 — Prompt engineering

5 hrs

4.1 **Core techniques**
instruction-based prompting · zero-, one- and few-shot · role and system prompts

4.2 **Reasoning techniques**
chain prompting · chain of thought · self-consistency · tree of thought

4.3 **Reliable output**
verifying and controlling output · structured JSON output · prompt-injection basics

05 MODULE 5 — LangChain essentials

6 hrs

5.1 **Framework basics**
what LangChain is · key components and modules

5.2 **Prompts and chains**
prompt templates · chains and their uses · output parsers

5.3 **Conversations and safety**
memory · moderating responses · handling hallucinations

06 MODULE 6 — Building LLM applications

5 hrs

6.1 **Text applications**
sentiment analysis · text classification · summarisation of long documents

6.2 **Information extraction**
extracting structured data from documents · PDF and OCR input

6.3 **Apps people can use**
chatbots · customer-service assistants · Streamlit front ends

07

MODULE 7

Embeddings and vector databases

4 hrs

7.1 **Embeddings**
embedding models · similarity search · chunking strategies

7.2 **Vector stores**
Chroma, FAISS and pgvector · metadata filtering · hybrid search and re-ranking

08

MODULE 8

Retrieval-augmented generation (RAG)

6 hrs

8.1 **Why RAG**
in-context vs external knowledge · RAG pipeline architecture

8.2 **Building the pipeline**
document loaders · text splitters · embeddings and vector storage · retrievers

8.3 **Better answers**
citations · conversation memory · evaluating retrieval quality

09

MODULE 9

Evaluation, fine-tuning and responsible AI

4 hrs

9.1 **Evaluating LLM apps**
test sets · faithfulness and relevance · LLM-as-judge · RAGAS

9.2 **Fine-tuning**
when to fine-tune vs prompt vs RAG · LoRA on a small open model

9.3 **Responsible AI**
guardrails · handling personal data · bias and safety

10

MODULE 10

Capstone project

9 hrs

10.1	**Plan the solution** use case · data sources · evaluation plan	
10.2	**Build and evaluate** LLM pipeline · Streamlit or web front end · evaluation report	
10.3	**Demo and review** live demonstration · review feedback	PROJECT

Total duration · 50 hours

Training hours across ten modules, including guided labs and the capstone project.

04 LEARN BY BUILDING

Practical learning

Hands-on learning

- A prompt library that turns messy text into clean JSON.
- Sentiment, classification and summarisation apps with LangChain output parsers.
- A Streamlit chatbot with memory and moderated responses.
- A "chat with your documents" RAG app with citations, scored with RAGAS.

Tools & technologies

Models

OpenAI · Google Gemini · Anthropic Claude · Hugging Face · Ollama

Frameworks

LangChain · Streamlit

Retrieval

Chroma · FAISS · pgvector

Workflow

Python · Jupyter · VS Code · RAGAS · Git and GitHub

Projects

Mini project · Vendor quotation scorer

Extract prices, taxes and terms from vendor quotations, then score and compare suppliers.

Major project · Resume onboarding assistant

Parse uploaded resumes into structured profiles, score them and suggest improvements.

Capstone · choose one

A bank-statement analyser that classifies transactions, a college-policy assistant using RAG, or a project of your own approved by the trainer.

TECH DRONA

DEBUG YOUR CAREER. DEPLOY YOUR DREAMS.

One technology partner. From learning to building to scaling.

Software engineering, AI and cloud, practical training and internships, recruitment and technology procurement, from one accountable team in Vijayawada.

NEXT STEP

Coming soon: join the waitlist

techdrona.in/contact

WEBSITE
techdrona.in

EMAIL
info@techdrona.com

LOCATION
Vijayawada, Andhra Pradesh, India

MSME registered

AICTE partner

Course details, schedules and fees are confirmed on enquiry and may change.