

PYTHON WITH AI

★ MODULE 1: Introduction to Modern AI (LLMs)

1.1 AI, ML, Deep Learning Overview

- Traditional ML vs Deep Learning
- Why LLMs replaced most ML workflows

1.2 LLM Concepts

- Tokens & embeddings
- Context window
- Prompt engineering fundamentals

1.3 Leading AI Models

- GPT-5.1
 - Gemini 2.0
 - Claude 4.2
 - Llama 3.1, Mistral, DeepSeek (open source)
 - Multimodal models
-

★ MODULE 2: Python Libraries for AI

2.1 Data & Scientific Computing

- NumPy
- Pandas (DataFrames, preprocessing)
- Matplotlib / Seaborn (visualization)

2.2 Working with Text

- Regular Expressions
 - Text cleaning / tokenization
-

★ MODULE 3: Using LLM APIs with Python

3.1 API Integration Basics

- REST API
- API keys & security
- Rate limiting

3.2 Python SDKs for AI

- OpenAI Python SDK
- Google Gemini Python API
- Anthropic Python SDK
- HuggingFace Inference API

3.3 Building AI Features

- Chatbots
- Summarizers
- Translators
- Rewriters
- Information extraction
- JSON-mode structured outputs
- Streaming responses

3.4 Hands-on Projects

- AI conversational chatbot
- PDF summarizer

- AI text cleaner
-

★ MODULE 4: RAG (Retrieval-Augmented Generation)

4.1 Understanding RAG

- What is retrieval?
- Chunking strategies
- Embeddings theory

4.2 Generating Embeddings in Python

- OpenAI embeddings
- Sentence Transformers (HuggingFace)
- Local embedding models

4.3 Vector Databases

- Pinecone
- ChromaDB
- Qdrant
- FAISS

4.4 Building End-to-End RAG Systems

- Convert documents → chunks → embeddings
- Store & retrieve vectors
- Send context to LLM
- Build Q&A bots

4.5 RAG Projects

- PDF Knowledge Base
 - AI Student Notes Generator
 - FAQ Automation System
-

★ MODULE 5: Building AI Agents with Python

5.1 What Are AI Agents?

- LLM reasoning
- Tools & actions
- Memory
- Multi-step planning

5.2 Agent Frameworks

- LangChain
- LangGraph
- OpenAI Assistants API
- CrewAI (team of agents)

5.3 Tools & Function Calling

- Creating Python functions as tools
- File search
- Calculator
- Database lookups
- Email sender
- Browser automation

5.4 Multi-Agent Systems

- Role-based agents

- Supervisor agents
- Collaborative tasks

5.5 Agent Projects

- AI automation assistant
- Email & report generator
- Invoice reader + analyzer agent

★ MODULE 6: Local LLMs

6.1 Installing & Running Local Models

- Ollama
- LM Studio
- Running Llama 3.1 / Mistral / DeepSeek locally

6.2 Python Integration

- REST API usage
- Embeddings
- Offline chatbots

6.3 Projects

- Offline AI Chat Assistant
- Secure internal AI bot (no cloud needed)

★ MODULE 7: Deep Learning & NLP (Optional)

7.1 Neural Network Basics

- Tensors
- Activation functions
- Loss & optimization

7.2 NLP Techniques

- Tokenization
- Named Entity Recognition
- Sequence modelling

7.3 Python Libraries

- TensorFlow
- PyTorch
- Transformers

7.4 Projects

- Sentiment analysis
- Text classification

★ MODULE 8: Databases for AI Applications

8.1 SQL Databases

- SQLite
- PostgreSQL
- Storing chat history & metadata

8.2 NoSQL Databases

- MongoDB
- Redis

8.3 Hybrid Search

- Keyword + vector search combo

★ MODULE 9: AI Projects (Capstone)

1. AI Chatbot using Python + OpenAI/Gemini
2. RAG-based Document Search Engine
3. Resume Analyzer & Job Match System
4. Multi-Agent Automation System
5. Local AI Assistant (offline using Ollama)
6. AI Code Reviewer
7. AI Note Generator for Students

★ MODULE 10: Deployment & DevOps (Optional)

10.1 Deploying AI Apps

- FastAPI / Flask
- Docker + containers
- Render / Railway
- AWS EC2 / Lambda

10.2 Best Practices

- Token security
- Error handling
- Monitoring & logs

★ MODULE 11: Interview Preparation

- Python interview questions
- AI + LLM interview questions
- GitHub portfolio building
- Resume building

🎯 Final Outcome

- ✓ Python Developer (AI/ML)
- ✓ AI Engineer
- ✓ RAG Engineer
- ✓ AI Automation Engineer
- ✓ Multi-Agent System Developer
- ✓ LLM Integration Engineer
- ✓ Backend AI Developer (FastAPI/Flask)