

S.No	Title	Overview
1	Course Kickoff & Fundamentals of Python	- Provide a high-level roadmap of topics: Python, ML, LLMs, Fine-tuning, RAG, Deployment, and Agents. - Explain how Python is the lingua franca of AI due to its simplicity, extensive libraries (NumPy, Pandas, Scikit-learn, PyTorch), and community support. - Cover Python basics: data types, loops, functions, OOP—ensuring everyone can write and read AI-related Python code.
2	Fundamentals of Machine Learning	- Introduce core ML concepts: supervised/unsupervised learning, bias-variance, overfitting, and generalization. - Cover statistics (mean, variance, distributions), probability, and the importance of experimentation in AI. - If time permits, introduce core Python packages: NumPy, Pandas, Matplotlib, Seaborn for EDA and ML workflows.
3	Building a Simple ML Model with Scikit-learn	- Walk through building a pipeline: data loading, preprocessing, training, evaluation. - Use models like logistic regression, decision trees to make ML less abstract. - Discuss how classical ML still solves many real-world AI problems in domains like healthcare, finance, etc.
4	Building a Deep Learning Model with Pytorch	- Explain neural networks, activation functions, layers,

		backpropagation. - Implement a simple neural net with Pytorch to reinforce abstraction with practice. - Deep learning is foundational for modern AI systems like vision models and LLMs.
5	NLP & Transformers: Historical to Modern View	- Cover NLP evolution: bag-of-words → word embeddings → transformers. - Introduce the transformer architecture: self-attention, positional encoding. - NLP is central to AI's understanding of human language; transformers are the backbone of models like GPT and BERT.
6	LLMs in Practice with Ollama	- Demonstrate how to run LLMs locally using Ollama for privacy and control. - Explore prompting, limitations, and useful applications like summarization or Q&A. - Real-world LLM usage enables building AI-driven assistants, chatbots, or copilots.
7	Retrieval-Augmented Generation (RAG)	- Explain RAG concept: combining retrieval with generation for more accurate, grounded responses. - Cover cosine similarity, embeddings, and vector stores (like ChromaDB, FAISS). - RAG is vital for domain-specific AI—e.g., medical chatbots referencing latest research.
8	Fine-tuning an LLM for Your Use Case	- Show how to fine-tune using domain-specific data

		<p>for better performance. - Explain the difference between fine-tuning and prompt-engineering. - Fine-tuning helps models specialize in areas like legal, healthcare, or enterprise documents.</p>
9	Deployment of LLMs	<p>- Discuss model deployment strategies: REST APIs, Streamlit- Cover real-world concerns: latency, cost, scaling, monitoring. - Deployment bridges the gap between research and usable AI products.</p>
10	Real Agentic AI Use Case Implementation	<p>- Define Agentic AI: systems that can plan, reason, and act autonomously. - Implement a simple use case using an open-source agent framework.</p>