**Course title:**

**Hands-on Generative AI Engineering with Large Language Models**

**Introduction to the Course**

* Course Objectives
* Course Structure
* Learning Paths

**Part 1: Software Prerequisites for Python Projects**

* IDE
  + VS Code
  + PyCharm
* Terminal
  + Windows: PowerShell, …
  + Mac OS: iTerm2, …
  + Linux: Bash, …
* Python Installation
  + Python installer
  + Anaconda distributions
* Python Environment
  + `venv`
  + `conda`
* Python Package Installation
  + Pypi, pip
  + Anaconda, conda
* Pieces of Software Used in This Course

**Part 2: Introduction to Transformer**

* Introduction to NLP before and after Transformer’s arrival
* Mastering Transformer’s block-by-block
* Transformer’s training process
* Transformer’s inference process

**Part 3: Implementing Transformer from Scratch with PyTorch**

* Introduction to training process implementation
* Implementing transformer as a Python package
* Calling the train and inference processes
* Experimenting with notebooks

**Part 4: Generative AI with the Hugging Face Ecosystem**

* Introduction to Hugging Face
* Hugging Face Hubs
  + Models
  + Datasets
  + Spaces
* Hugging Face Libraries
  + Transformers
  + Datasets,
  + Evaluate, …
* Practical guides with Hugging Face
  + Fine-tune a Pre-trained Language Model with Hugging Face
  + Hugging Face Notebook Guides

**Part 5: Components to Build LLM-based Web Applications**

* LangChain / LlamaIndex
* Open source / Private LLM
* Vector Embedding
* Vector database
* Prompt Engineering
* Streamlit / Gradio

**Part 6: Building LLM-based Web Applications**

* Task-specific AI Assistants
  + Culinary AI Assistant
  + Marketing AI Assistant
  + Customer AI Assistant
  + SQL-querying AI Assistant
  + Travel AI assistant
  + Summarization AI Assistant
  + Interview AI Assistant
* Simple AI Chatbot
* RAG (Retrieval Augmented Generation)-based AI Chatbot
* Agent-based AI Chatbot
  + AI Chatbot with Math Problems
  + AI Chatbot with Search Problems

**Part 7: Serving LLM-based Web Applications**

* Create the Frontend and Backend as two separate services
* Communicate between frontend and backend using a REST API
* Serve the application with Docker
  + Install, run and enable communication between Frontend and Backend in a single Docker container
* Use-case
  + An LLM-based song recommendation app

**What We Have Learned**

**Beyond the Course**

**Thank you**

**Short table of content**

**Hands-on Generative AI Engineering with Large Language Models**

**Introduction to the Course**

**Part 1: Software Prerequisites for Python Projects**

**Part 2: Introduction to Transformer – Attention Is All You Need**

**Part 3: Implementing Transformer from Scratch with PyTorch**

**Part 4: Generative AI with the Hugging Face Ecosystem**

**Part 5: Components to Build LLM-based Web Applications**

**Part 6: Building LLM-based Web Applications**

**Part 7: Serving LLM-based Web Applications**

**What We Have Learned**

**Beyond the Course**

**Thank you**