Continuing Education and Training

with

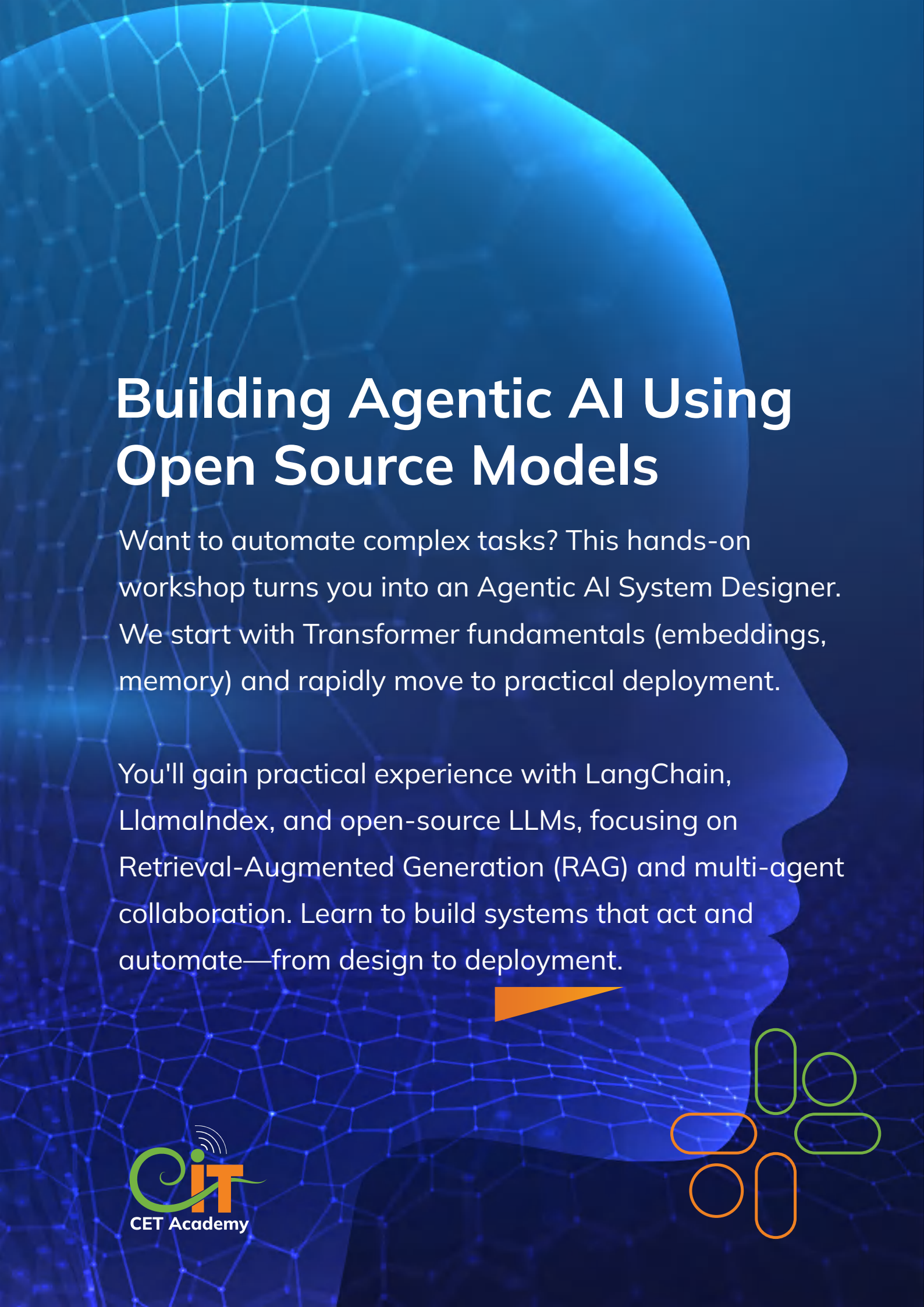
GCIT

Less Training More Application

Visit our website

ceta.gcit.edu.bt

CET Academy



Building Agentic AI Using Open Source Models

Want to automate complex tasks? This hands-on workshop turns you into an Agentic AI System Designer. We start with Transformer fundamentals (embeddings, memory) and rapidly move to practical deployment.

You'll gain practical experience with LangChain, LlamaIndex, and open-source LLMs, focusing on Retrieval-Augmented Generation (RAG) and multi-agent collaboration. Learn to build systems that act and automate—from design to deployment.

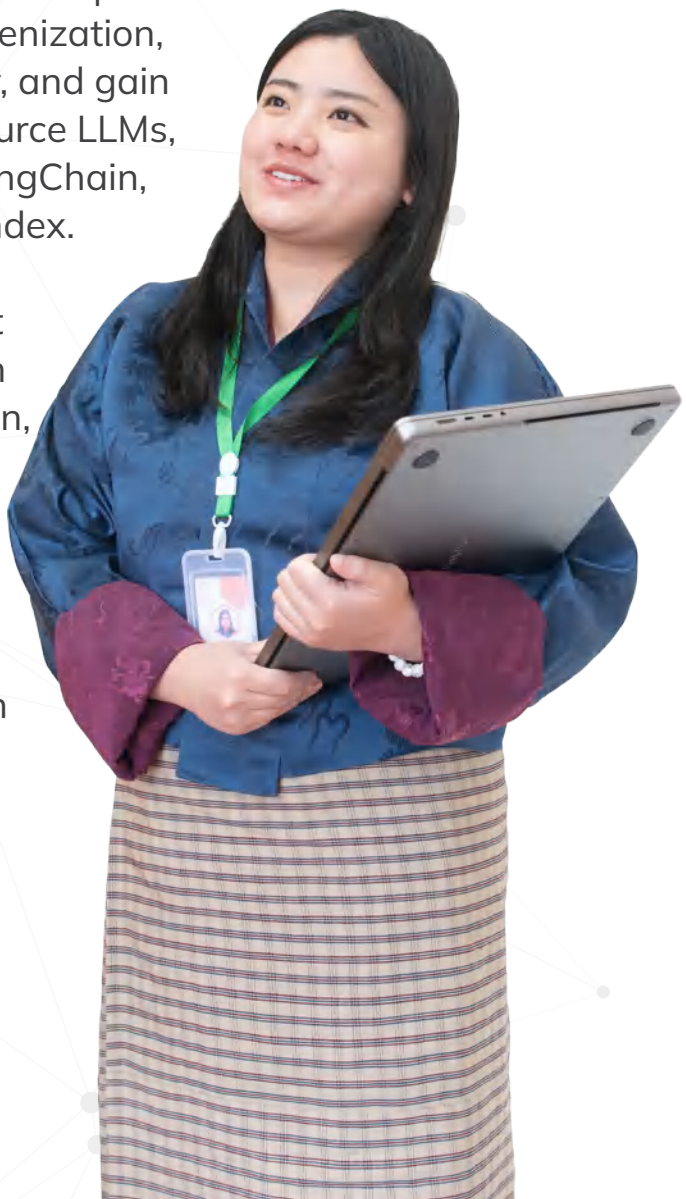
CET Academy

Workshop
Overview

Building Agentic AI Using Open Source Models

This workshop provides a hands-on introduction to Agentic AI, guiding participants to design, build, and deploy autonomous, goal-driven AI systems using open-source large language models (LLMs) and generative AI frameworks. Participants will explore Transformer fundamentals, including tokenization, embeddings, self-attention, and memory, and gain practical experience with major open-source LLMs, as well as agent frameworks such as LangChain, AutoGPT-style frameworks, and LlamaIndex.

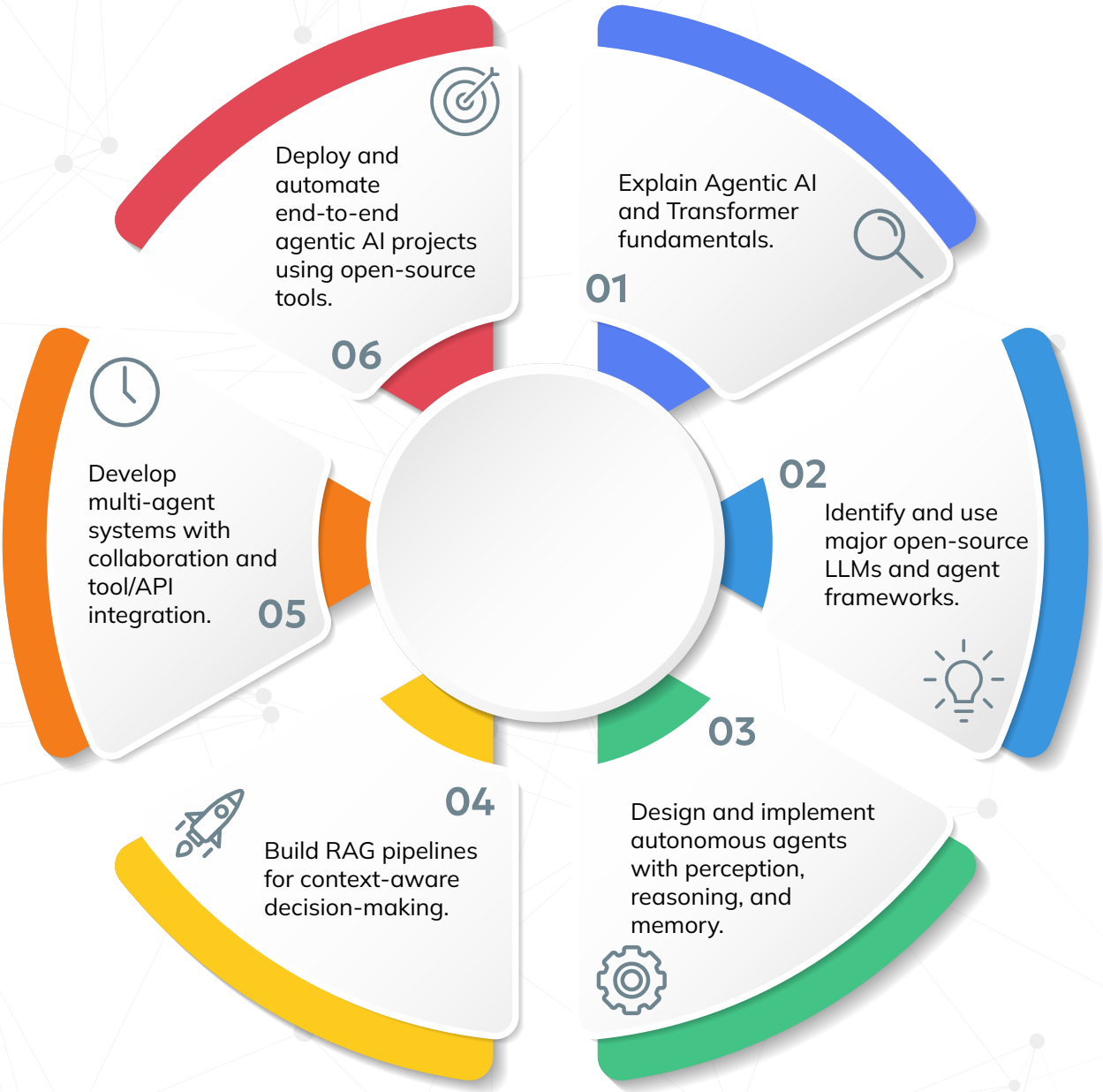
The workshop covers autonomous agent design, Retrieval-Augmented Generation (RAG) pipelines, multi-agent collaboration, and the integration of tools and APIs. Through lectures, hands-on exercises, and mini-projects, participants will learn to build end-to-end, deployable agentic AI systems that can perceive, reason, plan, act, and automate complex tasks in real-world or simulated environments.



Learning
OUTCOME

Building Agentic AI Using Open Source Models

By the end of the workshop, you will be able to:



Workshop
SCHEDULE

Module 01
Introduction to Agentic AI and Transformer Foundations

- What is Agentic AI?
- Transformer Architecture Deep Dive:
    - Self-attention, embeddings, tokenisation, etc.
    - How Transformers enable reasoning and memory.
- **Overview of open-source LLMs:** LLaMA, MPT, Falcon, Dolly, Vicuna.
- **Agent frameworks:** LangChain, AutoGPT-style frameworks, LlamaIndex.
- Introduction to GEN AI frameworks and use cases.
- Setting up a Python environment with open-source libraries.
- **Hands-On Exercises:**
    - Run an open-source LLM locally or in Colab.
    - Generate outputs and test basic prompt-response cycles.

Module 02
Foundations of Autonomous Agent Design and GEN AI Tool-Calling

- Core agent components: **Perception, Reasoning, Action, Memory.**
- Task decomposition and planning.
- Getting Started with LangChain and LlamaIndex
- **Tool-Calling with GEN AI:** Connecting agents to APIs, calculators, or external systems.
- Best practices for agent orchestration and safety measures.
- Ethical AI design using open-source tools.
- **Hands-On Exercises:**
    - Build a rule-based agent in Python.
    - Implement a multi-step task planner using LangChain.
    - Create a simple agent that uses tool-calling (e.g., search or calculator API).

Workshop
SCHEDULE

Module 03 — Integrating Open-Source LLMs with RAG Pipelines

- LLMs as reasoning engines: grounding, context, and decision-making.
- Prompt engineering for agentic behaviour.

  RAG Pipeline Design:
  - **Retrievers:** FAISS, Chroma, Milvus, Pinecone
  - **Generators:** LLaMA 3, Mistral, Falcon, Dolly (via Ollama), or MPT
- **Vector Databases & Knowledge Management:** indexing and embedding best practices.
- Memory management and contextual persistence (LangChain Memory, LlamaIndex).
- **RAG Use Cases:** question answering, summarisation, and task-aware agents.
- **Hands-On Exercises:**
  - Implement a RAG pipeline using Pinecone + Ollama embeddings
  - Build an Agentic RAG System (combine retrieval + generation)
  - Use Ollama for embedding and/or generation

Module 04 — Multi-Agent Collaboration and End-to-End Development

- Single vs. Multi-Agent Systems: communication and coordination.
- **Simulation Environments**: Gym, PettingZoo, MiniGrid.
- Agent Collaboration & Negotiation: task allocation, message passing, and role definition.
- Integrating RAG and tool-calling within multi-agent contexts.
- Integrating RAG, REST APIs, and Tool-Calling within Multi-Agent Contexts:
  - Agents calling external tools and REST APIs to get live data or perform actions
- **Mini-Project Kickoff:** Design an End-to-End Agentic AI Workflow (from Planning to Deployment).
- **Hands-On Exercises:**
  - Integrate at least one REST API call in an agent workflow
  - Implement two or more collaborating agents in a simulation.
  - Begin building a small-scale project with your assigned group.

Workshop
SCHEDULE

Module 05

Project & Automation Tools

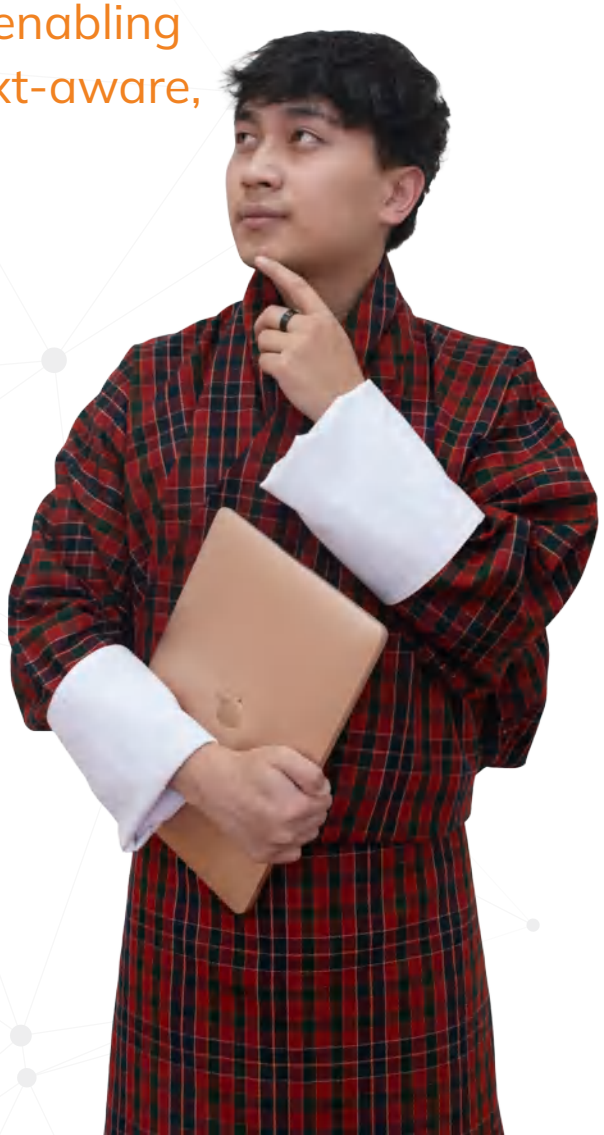
- **End-to-End System Integration:** perception, reasoning, RAG, and tool-calling.
- **Automation Tools for Agentic AI:** Workflow orchestration, AutoGPT-style task loops.
- Evaluation metrics: success rate, efficiency, adaptability.
- Deployment options (Local, Streamlit, or Gradio).
- **Hands-On Exercises:**
    - Build and deploy a complete agentic AI project using open-source models.
    - Automate tasks through workflow tools.
    - Present or demo the capstone project for evaluation.

Why This **PROGRAM?**

This hands-on workshop equips participants with the practical skills to design, build, and deploy agentic AI systems using open-source large language models (LLMs) and generative AI frameworks. Learners will gain a strong foundation in Transformers, autonomous agent design, retrieval-augmented generation (RAG), multi-agent collaboration, and end-to-end system integration, enabling them to create intelligent, context-aware, and deployable AI solutions.

Target Audience

- Diploma, undergraduate, or graduate students in computer science, software engineering, or related fields.
- AI/ML developers or engineers seeking to transition into Agentic AI and autonomous systems.
- Researchers or professionals interested in LLMs, RAG pipelines, and multi-agent systems.
- Enthusiasts and innovators looking to build practical AI applications using open-source tools.



Things To **KNOW**

Prerequisite:

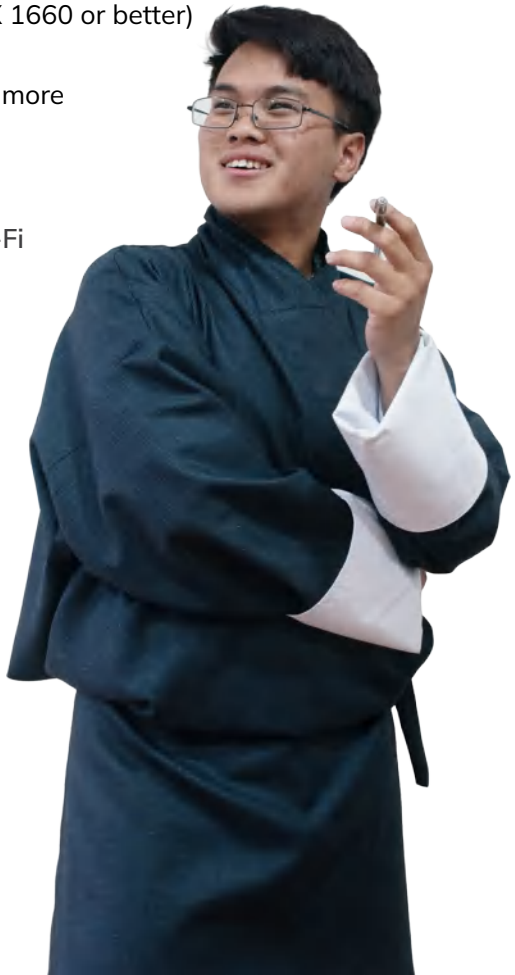
- Basic understanding of Python programming
- Familiarity with machine learning concepts (models, training).
- Basic knowledge of neural networks and transformer models is recommended.
- Comfort with using command-line interfaces (CLI) and virtual environments

Minimum Computer Specifications:

- Windows 10 / 11 Operating System or later version **(MAC OS is also supported)**
- CPU of a Quad-core processor (Intel Core i5/AMD Ryzen 5 or newer)
- GPU: Dedicated GPU with 6 GB VRAM (e.g., NVIDIA GTX 1660 or better)
- RAM of 16 GB or more
- Connection to Internet with Wi-Fi access

Workshop Details:

- Duration: 1 week (5 days), 8 hours per day (total 40 hours)
- Delivery Mode: In-Person



Capable. Elevated. Thriving.

CET Academy

ceta.gcit.edu.bt

info.ceta@gcit.edu.bt

+975 77 132 432

Thimphu, Bhutan