Generative AI (GenAI) Overview

- Generative AI refers to AI systems capable of creating new content such as text, images, code, or audio by learning patterns from large datasets.
- Built on **foundational models**, which are pre-trained for general purposes and adaptable to multiple downstream tasks.
- Foundational models can be approached from two perspectives:
    - **Builder Perspective** → focuses on designing, training, optimizing, and deploying models.
    - **User Perspective** → focuses on using and integrating models into applications.

Impact Areas of Generative AI

1. **Customer Support**
    - Conversational AI for chatbots and virtual assistants.
    - Automating query resolution and improving customer experience.
2. **Content Creation**
    - Text, blog, and report generation.
    - Creative media: music, images, and video generation.
3. **Education**
    - Personalized learning assistants.
    - Automated grading and tutoring.
4. **Software Development**
    - Code generation and debugging assistants.
    - Documentation and test-case creation.

Builder's Perspective

Focuses on how foundational models are created and optimized:

- **Transformer Architecture**
    - Core design enabling attention mechanisms.
- **Types of Transformers**
    - Encoder-only (e.g., BERT).
    - Decoder-only (e.g., GPT).
    - Encoder–decoder (e.g., T5).
- **Pretraining**
    - Training objectives, tokenization, strategies, handling challenges.
- **Fine-tuning**
    - Task-specific tuning.
    - Instruction tuning.
    - Continual pretraining.
    - RLHF (Reinforcement Learning with Human Feedback).
    - PEFT (Parameter Efficient Fine-Tuning).
- **Optimization**
    - Training optimization.
    - Model compression.
    - Inference optimization.
- **Evaluation**
    - Benchmarking and model validation.
- **Deployment**
    - Scaling models for real-world usage.

User's Perspective

Focuses on applying and extending foundational models:

- **Building Basic LLM Applications**

    - Open source vs. closed source models.
    - Using APIs (e.g., OpenAI, Anthropic).
    - Tools/frameworks: LangChain, HuggingFace, Ollama.
- **Prompt Engineering**
    - Crafting effective prompts to guide model outputs.
- **Retrieval-Augmented Generation (RAG)**
    - Combining LLMs with external knowledge bases.
- **Fine-tuning**
    - Customizing pre-trained models for domain-specific tasks.
- **Agents**
    - Autonomous multi-step systems powered by LLMs.
- **LLMOps**
    - Operationalization, monitoring, and maintenance of LLM-powered apps.