

Generative AI:-

“AI that can *create* things — like text, images, or ideas — based on what you ask.”

What is Generative AI? (Simple Explanation)

Generative AI refers to artificial intelligence systems that can **create new content** such as:

- Text
- Images
- Code
- Audio / Music
- Video
- 3D Models
- Designs

These models don't just retrieve information — they **generate** something new based on the patterns they learned from huge datasets.

☐ **Example:**

Prompt: “*Write a poem about the moon.*”
Generative AI creates a brand-new poem.

Prompt: “*Generate an image of a cat riding a bicycle.*”
AI produces an original image.

These Are Top Generative AI Systems

1 📱 ChatGPT (OpenAI) – Text + Code + Image + Reasoning

- One of the most advanced generative AI chat systems
- Generates text, explanations, stories, code, essays, summaries
- Can also generate **images**, **analyze files**, **solve problems**, and more

✓ Example of: **LLM + Multimodal Generative AI**

2 📱 Grok (xAI) – Conversational AI + Live Internet Access

- Created by Elon Musk's company xAI
- Designed to be witty, fast, and always up-to-date

- Generates text, analyzes data, writes code
- Uses the **Grok-1 model**, a generative LLM

✓ Example of: **LLM-based conversational generative AI**

3 **Gemini (Google) – Multimodal AI**

- Can handle **text, images, code, audio, and video**
- Trained on large multimodal datasets
- Used in Google apps: Gmail, Search, Maps, Android

✓ Example of: **Multimodal generative AI system**

4 **Microsoft Copilot – AI Assistant for Productivity**

- Integrated into Windows, Office (Word, Excel, PowerPoint), Edge
- Writes emails, summarizes documents, creates presentations
- Generates code through GitHub Copilot

✓ Example of:

- **Generative AI for productivity**
 - **Code generation (GitHub Copilot)**
-

Why These Are Great Examples of Generative AI

Because they can:

- ✓ **Generate new text (answers, essays, stories, reports)**
- ✓ **Generate or help generate code**
- ✓ **Generate images (ChatGPT, Gemini)**
- ✓ **Understand and process images/files**
- ✓ **Assist creatively (emails, presentations, writing)**

These tools use Large Language Models (LLMs), which are the core of modern **Generative AI**.

In simple words:

Yes — **ChatGPT, Grok, Gemini, and Copilot** are the top practical examples of **Generative AI tools used by millions of people**.

They show how generative AI can work in:

- Chatbots
- Coding assistants
- Writing assistants
- Productivity tools
- Knowledge search
- Creative tools

How Generative AI Works (Basics)

Generative AI uses special models called **Large Generative Models** that learn patterns from millions or billions of examples.

There are different types:

1. LLMs (Large Language Models)

These generate **text, code, chatbot conversations**, etc.
Examples: Llama, GPT, Mistral, Qwen, Gemma.

2. Diffusion Models

These generate **images, art, videos**.
Examples: Stable Diffusion, SDXL, Stable Video Diffusion.

3. Audio Generators

Generate speech and music.
Examples: MusicGen, Bark, OpenVoice.

4. Multimodal Models

Handle multiple types (text + image + audio).
Examples: Llama 3.2, Qwen2-VL, Pixtral, Kosmos-2.

LLM (Large Language Model):-

“A computer program trained to read and produce language. It predicts the next words similar to how texting apps predict your next word, but far more advanced.”

? What is an LLM (Large Language Model)

- An LLM is a type of AI model trained on *massive* amounts of text data — billions of words, from books, websites, articles, code, etc. — so that it learns patterns of human language (grammar, meaning, context).
- When you give it a prompt (some text input), it “predicts” what comes next — word by word — to generate human-like text: answers, stories, summaries, translations, code, and more.
- Because they have very large numbers of internal parameters (the “knobs” the model adjusted during training), LLMs can produce surprisingly nuanced, coherent, context-aware outputs.

- LLMs are the backbone behind “generative AI” tools: chatbots, writing assistants, translators, summarizers, etc.

In short: **LLMs are powerful AI** “text engines” that — given a prompt — can generate or transform language in many useful ways.

For example:-

<https://mistral.ai/>

? Sample Open-Source / Free LLMs (2025)

Model / Name	Parameters / Size / Notes	Strengths / Best Use Cases
GPT-J	~ 6 B parameters Wikipedia+1	Light enough for small hardware — good for writing, simple content generation, experimenting without big compute.
GPT-NeoX	Up to ~ 20 B params GeeksforGeeks+1	Better quality than smaller models; suitable for longer text, more context, content + code generation.
BLOOM	Up to ~ 176 B params newnex.io+1	Multilingual support (many languages + programming languages); good for global use, translation + generation.
MPT	Offers ~ 7 B and ~ 30 B param versions newnex.io	Balanced option — efficient and usable even without high-end hardware; decent for many tasks.
Dolly 2.0	Derived from smaller/open models newnex.io	Useful if you want a lighter model with fine-tuning possibilities, simpler tasks or prototypes.

Model / Name	Parameters / Size / Notes	Strengths / Best Use Cases
XGen-7B	7 B params (open-source) newnex.io	Good for lighter setups — generates text efficiently without heavy compute needs.
Mixtral (8x7B)	Sparse / Mixture-of-Experts — efficient inference & good results per compute arXiv+1	Great if you want high performance without enormous GPU resources — good for code generation, multilingual tasks, long context.
DBRX	132 B params (Mixture-of-Experts; 36 B active per token) Wikipedia	High performance — strong in reasoning, code, large-scale text — useful if you have decent hardware or cloud GPU.
Gemma	Released variants up to ~ 27 B (2025) by DeepMind Wikipedia	Lightweight to mid-range; good for multilingual, multimodal (text+some extended functionality), easier to run than huge models.
Code Llama	Multiple variants: 7B / 13B / 34B / 70B for code-specialized tasks arXiv	Excellent for coding, code generation, code completion, dev-oriented tasks — better than general models at programming.
Mistral 7B	7 B params — efficient & accessible Simplilearn.com+1	Lightweight, cost-efficient, good for chatbots, quick text generation, prototyping — ideal if you don't have powerful GPU.
Falcon	Various versions (small to larger) under Apache-2 license Simplilearn.com+1	Flexible — you choose based on hardware, useful for many tasks (text, code, multilingual).
TinyLlama	~ 1.1 B params (very lightweight) arXiv	Great for experiments, very low-resource environments, educational purposes, quick prototyping.
Qwen 3	Released in 2025 — includes dense and sparse variants, supports large context windows (~128K tokens) Wikipedia+1	Powerful and modern: works well for long documents, multilingual text, and complex tasks (reasoning, writing, summarization).
DeepSeek-R1	High-performance open-source LLM (2025) Titan AI Explore+1	Built for reasoning, advanced language generation & agentic tasks — good if you need deeper capabilities (code, reasoning, large context).

🔗 What is BLOOM?

BLOOM is a fully **open-source LLM** trained by the BigScience project. It comes in sizes from **560M** → **176B parameters**.

- License: **Responsible AI License (RAIL)**
 - Truly open (weights + code + data are public)
 - Good for text generation, multilingual tasks, and experimentation
-

1. Use BLOOM with Ollama (Easiest Way)

Ollama supports BLOOM and BLOOMZ (instruction-tuned).

❑ Pull the model

```
ollama pull bloom
```

❑ RUN

```
ollama run bloom
```

For BLOOMZ (better instructions)

```
ollama pull bloomz  
ollama run bloomz
```

For 7B or other sizes:

```
ollama pull bloomz:7b  
ollama run bloomz:7b
```