# GenAI Roadmap for Beginners: Overview

1. What is Generative AI?

Generative AI as: **"a type of AI that creates new content such as text, images, music, or even code by learning patterns from existing data, mimicking human creativity."**

This has ability to "mimic human creativity" is GenAI's **"biggest superpower"**, fundamentally changing the long-held belief that AI could not replace human creativity. While earlier AI techniques like Symbolic AI, Fuzzy Logic, Evolutionary Algorithms, NLP, and Computer Vision contributed, **Machine Learning** emerged as the most impactful. However, traditional Machine Learning focused on predictive tasks (regression, classification, ranking) and **"कभी भी मशीन लर्निंग को इस तरह के प्रॉब्लम स्टेटमेंट्स में यूज नहीं किया जाता था जहां पर ह्यूमन क्रिएटिविटी की जरूरत होती थी"** (Machine Learning was never used for problem statements requiring human creativity). GenAI has bridged this gap.

2. Generative AI's Place in the AI Landscape

Hierarchical view:

- **AI** (outer circle)
- **Machine Learning**
- **Deep Learning**
- **Generative AI** (innermost circle)

This lineage indicates GenAI's emergence from advancements within Deep Learning, particularly with the advent of the Transformer architecture.

3. Why Learn Generative AI? (Impact Areas)

Kanhiya argues for GenAI's importance by highlighting its significant impact across various sectors, stating it has become **"हमारे करंट दुनिया की सबसे पावरफुल टेक्नोलॉजी"** (the most powerful technology in our current world) within two years.

Four key impact areas are identified:

- **Customer Support:** GenAI has revolutionized customer handling for large companies by providing a **"first level of chatbot"** that reduces the need for human executives, leading to significant cost savings and improved efficiency.
- **Content Creation:** GenAI tools are deeply penetrating this industry, helping create text-based content (blogs, websites), video content, and even music. The quality is so high that **"आज की डेट में अगर मैं मीडियम पे कोई आर्टिकल पढ़ता हूं तो मैं 100% श्योर हो के नहीं बोल सकता कि वह किसी इंसान ने लिखा है या फिर एआई ने लिखा है"** (today, if I read an article on Medium, I cannot be 100% sure if a human or AI wrote it).
- **Education:** Tools like ChatGPT have made learning and exploring new topics **"काफी इजी"** (quite easy), acting as a "personal tutor." This is forcing schools to rethink educational methodologies.
- **Software Development:** GenAI models are highly proficient at **"writing code"**, including "production-ready code." This could reduce the number of programmers needed for tasks, with tools making programming "kind of easy."

4. Is Generative AI a Successful Technology?

Kanhiya uses a "series of questions" to evaluate GenAI's success, comparing it to the Internet (highly successful) and Blockchain/Crypto (yet to reach full potential).

The questions and GenAI's answers:

- **Does it solve real-world problems? YES.** Examples include customer support challenges and providing personal tutors in education.
- **Is it useful on a daily basis? YES.** GenAI tools are regularly used for daily tasks.
- **Is it impacting the world economy? ABSOLUTELY YES.** Kanhiya cites a recent news story about a new Chinese AI model causing a "$1 trillion wipe out" in US tech company stock shares.
- **Is it creating new jobs? YES.** A completely new job role, "AI Engineer," has emerged in the technical field, with demand "increasing every day."
- **Is it accessible? YES.** GenAI tools are easy to use, even for non-technical users, as they don't require coding and can be operated using natural language (English or Hindi).

Conclusion: Since all questions yield a "YES," Kanhiya believes GenAI is **"सक्सेसफुल टेक्नोलॉजी बनने के रास्ते पर है इंटरनेट वाले रास्ते पे है"** (on the path to becoming a successful technology, on the path of the Internet). He advises investing time and effort into learning GenAI because it is still nascent and will greatly benefit early adopters.

5. Reasons for Delay in Creating GenAI Content

Kanhiya explains his delay:

- **Doubt about the Technology:** Initial skepticism about whether GenAI was a truly powerful technology or just a "bubble." This doubt has now been resolved.
- **Time Commitment Issues:** Personal life commitments limited his ability to allocate time. This issue has also been resolved.
- **Fear/Overwhelm due to Rapid Pace:** The **"हर दिन कुछ नया आ जाता था कोई नया मॉडल आ जाता था कोई नया टूल आ जाता था कोई नया रिसर्च पेपर आ जाता था कोई नया टर्मिनोलॉजी आ जाता था"** (every day something new comes, a new model, tool, research paper, terminology) made it difficult to decide what and where to study. This remains a challenge, addressed by his new mental model.

6. The Mental Model: Foundation Models at the Core

To overcome the information overload and "noise," Kanhiya developed a mental model centered around **Foundation Models**.

- **Definition of Foundation Models:** These are **"बहुत बड़े स्केल के एआई मॉडल्स"** (very large-scale AI models) trained on "huge amount of data" (like the entire internet) and requiring "a lot of hardware" (GPUs) and "crores of rupees" to train.
- **Key Characteristic:** Unlike task-specific Machine Learning models, Foundation Models are **"जनरलाइज्ड मॉडल्स"** (generalized models). They can solve "more than one task" because of their large architecture, numerous parameters, and extensive training data.
- **Example:** Large Language Models (LLMs) are excellent examples, forming the "backbone of Generative AI." LLMs can perform text generation, sentiment analysis, summarization, and question-answering. Large Multimodal Models (LMMs) also exist, handling images, videos, and sound.

7. The Two Perspectives of Generative AI

Based on the mental model, GenAI is divided into two core parts, both revolving around Foundation Models:

1. **Using Foundation Models (User Perspective):** This involves utilizing existing, pre-built Foundation Models for various applications.

2. **Building Foundation Models (Builder Perspective):** This involves creating and deploying new Foundation Models, typically done by large companies.

Any new GenAI term or tool will fall into one of these two categories, simplifying the learning process. Kanhiya provides examples:

Term Perspective Explanation Prompt Engineering User Refining inputs to existing LLMs for better responses. RLHF Builder Modifying LLM behaviour for safety during the building process. RAG User Using existing LLMs to answer questions on private data. Pre-training Builder Training a foundation model on vast datasets using large hardware. Quantization Builder Optimizing model size for deployment in various environments. AI Agents User Building software that uses LLMs to perform actions beyond just Q&A (e.g., booking tickets). Vector Databases User Used for RAG implementation. Fine-tuning Both Can be done during model creation (Builder) or to enhance an existing model for specific tasks (User).

8. Curriculum Breakdown

Kanhiya presents detailed curricula for both perspectives:

A. Builder Side Curriculum (Prerequisites: ML and Deep Learning Fundamentals, TensorFlow/PyTorch)

This side is more technical and relevant for research scientists and machine learning engineers.

1. **Transformer Architecture:** Understanding encoders, decoders, embeddings, self-attention, layer normalization, and language modeling.
2. **Types of Transformers:** Encoder-only (e.g., BERT), Decoder-only (e.g., GPT), and Encoder-Decoder based.
3. **Pre-training:** Training objectives, tokenization strategies, training strategies (local, cloud, distributed), challenges and solutions, and evaluation.
4. **Optimization:** Techniques to make large models runnable on normal hardware (e.g., quantization, knowledge distillation, inference time reduction).
5. **Fine-tuning:** Task-specific tuning, instruction tuning, continual pre-training, RLHF, and PEFT (Parameter-Efficient Fine-Tuning).
6. **Evaluation:** Thorough techniques and metrics to assess model performance (e.g., LLM leaderboards).
7. **Deployment:** Making the models accessible for public use.

B. User Side Curriculum

This side is less technical and easier, suitable for anyone with some software development knowledge, focusing on using existing models to build applications.

1. **Building Basic LLM Apps:** Learning about available LLM types (closed-source, open-source), using APIs, Hugging Face, Ollama, and frameworks like LangChain.

- **Improving LLM Response:Prompt Engineering:** The "art and science of writing prompts" to refine LLM outputs.
- **RAG (Retrieval Augmented Generation):** Enabling LLMs to answer questions using private data.
- **Fine-tuning:** Shallow-level fine-tuning of existing models for specific tasks.

1. **AI Agents:** Building applications where LLMs can not only converse but also perform actions by accessing tools (e.g., booking a hotel).
2. **LLM Ops:** The umbrella term for all activities involved in building, evaluating, improving, and deploying LLM-based applications for customers.
3. **Miscellaneous (Multimodal Models):** Working with models that handle non-text inputs/outputs (audio, video) and understanding concepts like Stable Diffusion.

9. The AI Engineer Role

Kanhiya states that while a software developer can easily build AI applications (User-side focus), an **AI Engineer** is a new, in-demand role that possesses knowledge of **both** the User and Builder sides. This dual understanding allows for better operation on the user side and higher salary potential. He recommends learning both but emphasizes that focus can vary based on career goals (research scientist for builder, general software developer for user).