Tokenizer (GenAI) – Short Hinglish Explanation 👇

Tokenizer GenAI ka **translator** hota hai.
Ye **text ko chhote-chhote parts (tokens)** me tod deta hai taaki model usse samajh sake.

🔹 **Token kya hota hai?**
Word, sub-word ya character ka piece.

Example:
👉 `"I love ChatGPT"`
➡ Tokens: `["I", "love", "Chat", "GPT"]`

🔹 **Kyun zaroori hai?**
AI directly text nahi samajhta,
wo **numbers (tokens IDs)** samajhta hai.

🔹 **Simple line me:**
👉 *Tokenizer text → tokens → numbers me convert karta hai,*
jisse GenAI predict & generate kar pata hai.

User Question

> *"I am going to India, my budget is 10k, I want to save 5k. Tell me 3 best budget-friendly locations."*

🧠 Step 1: Tokenization (Question todna)

LLM pehle sentence ko **tokens** me todta hai:

```
I | am | going | to | India | budget | 10k | save | 5k | 3 | best | location | budget | friendly
```

👉 Isse LLM ko samajh aata hai:

- **Place** → India
- **Total budget** → ₹10,000
- **Constraint** → ₹5,000 save karna
- **Output** → 3 locations

🧠 Step 2: Intent Understanding (User kya chahta hai)

LLM infer karta hai:

✔ Travel recommendation
✔ Budget constraint problem
✔ Optimization (best + cheap)
✔ List format expected

👉 Ye ek **planning + reasoning task** ban jata hai.

🧠 Step 3: Knowledge Recall (Training Data se)

LLM apni training se yaad karta hai:

- India me **cheap travel places**
- Backpacker cities
- Hostels, street food, cheap transport wale areas

💡 Example knowledge chunks:

- Rishikesh = cheap stay + free attractions
- Pushkar = small town, low cost
- McLeod Ganj = backpacker culture

⚠️ But: **Exact prices ya live cost nahi pata**

🧠 Step 4: Constraint Matching (Budget logic)

LLM internally sochta hai:

- ₹10k total
- ₹5k save karna →
  👉 Max spending allowed = **₹5k**

Toh wo:
❌ Goa / Shimla jaise expensive places avoid karega
✅ Small towns / spiritual / backpacker locations choose karega

---

🧠 Step 5: Answer Generation (Next-token prediction)

Ab LLM **word by word predict karta hai**:

> “Based on your budget, some affordable destinations in India are…”

👉 Ye fact check nahi karta
👉 Sirf **most probable helpful answer generate karta hai**

✨ Final Answer ka Structure

LLM usually aise format me deta hai:

1️⃣ Location name
2️⃣ Short reason (cheap stay, food, travel)
3️⃣ General affordability statement

📌 One-Line Technical Summary (Interview Ready)

👉 *LLM tokenizes the query, identifies intent and constraints, recalls similar patterns from training data, and generates a plausible answer using next-token prediction — without verifying real-time costs.*

🔹 *LMM = Large Multimodal Model*

👉 ***LLM** sirf **text** samajhta hai*
👉 ***LMM** = **text + image + audio + video** sab samajh sakta hai*

🧠 *Difference samjho (Example se)*

✅ *LLM (Text only)*

User:

> *“Is image me kya hai?”*

LLM:
❌ *Image dekh hi nahi sakta*

✅ *LMM (Multimodal)*

User:

> *“Is image me kya hai?” (image upload ki)*

LMM:
✔ *“Image me ek ladka laptop pe kaam kar raha hai…”*

🔍 *LMM kya-kya samajhta hai?*

- 📝 *Text (chat, documents)*
- 🖼️ *Image (photos, diagrams)*
- 🎙️ *Audio (speech)*
- 🎥 *Video (frames + audio)*

🧩 *Example (Real-Life)*

👉 *Tum flight ticket ka screenshot upload karo*
👉 *Pucho: “Isme date aur price batao”*
➡️ ***LMM image read + text extract karke answer de dega***

🛠️ *Internally LMM kaise kaam karta hai?*

1️⃣ *Image / audio ko **tokens** me convert karta hai*
2️⃣ *Text tokens ke saath **combine** karta hai*
3️⃣ *Phir reasoning + generation karta hai*

User

↓

Zomato App (UI)

↓

Chatbot API

↓

Intent Detection (LLM/NLP)

↓

Internal DB + Recommendation Engine

↓

Response Generation (LLM / Template)

↓ *User*

🔁 *Simple Request–Response Flow (Frontend ↔ Server ↔ LLM)*