📘 Lecture 02: How LLM Works :

🔍 What is an LLM?

- **LLM = Large Language Model**
- यह एक ऐसा model है जो बहुत बड़े text data पर train किया जाता है।
- यह **calculation नहीं करता**, बल्कि **prediction** करता है।

🧠 LLM = Prediction, Not Computation

- ❌ LLM actual **calculation नहीं करता**।
- ✅ यह **trained examples** के आधार पर **prediction करता है**।

📌 Example:

```
2 + 2 = ?
LLM → Predicts: 4
```

- यह answer **calculate नहीं करता**, बल्कि **2+2 = 4** को बहुत बार देख चुका है — इसलिए सही जवाब predict कर देता है।

🧠 But: **LLM doesn't "know" math. It just imitates patterns.**

🍓 Strawberry Example: Counting Characters

- Question: *How many 'r's are in "strawberry"?*
- LLM may answer correctly: 2
- लेकिन ये गिन नहीं रहा, बस अनुमान लगा रहा है।

🛠️ Accuracy के लिए LLM external tool (जैसे Python code) की मदद लेता है।

🧮 LLM vs. Code Execution

Task	Can LLM Do It?
2+2 Prediction	✅ Yes (by pattern)
Actual code execution	❌ No
Current temperature lookup	❌ No
Using tools (e.g., Python)	✅ With help

- LLM के पास **live data या internet access नहीं होता** (by default).
- वो सिर्फ training data तक सीमित होता है।

📌 **Example:**

```
Q: What is the temperature in Varanasi?

LLM: Varanasi ka temperature 32°C hai.
```

👀 Did it calculate it?
❌ No — यह सिर्फ एक अनुमान या past data से trained info है।

🔧 How LLM Uses Tools (Like Python)

- Complex tasks के लिए LLM खुद code लिख सकता है।
- फिर external tool उस code को execute करता है।

🧪 **Example:**

```
# Count number of 'r' in "strawberry"
word = "strawberry"
print(word.count('r'))  # Output: 2
```

🧠 LLM can generate such code, but **cannot run it** itself.

💻 LLM Doesn't Have Live Knowledge

- LLM को किसी specific date या live web data की knowledge नहीं होती।

- वह **online trained** नहीं रहता।
- Example: *वह नहीं बता सकता कि अभी Varanasi का तापमान कितना है।*

🗓️ Updated LLM (like ChatGPT) might respond:

"As of my last update on 15 May 2023..."

🧵 Short-Term Memory = Context Window

- ChatGPT एक conversation में पिछले messages को **context window** में रखता है।
- Example:

```
User: Hi, I'm Harshal
Bot: Hi Harshal, nice to meet you!

User: What's my name?
Bot: Your name is Harshal
```

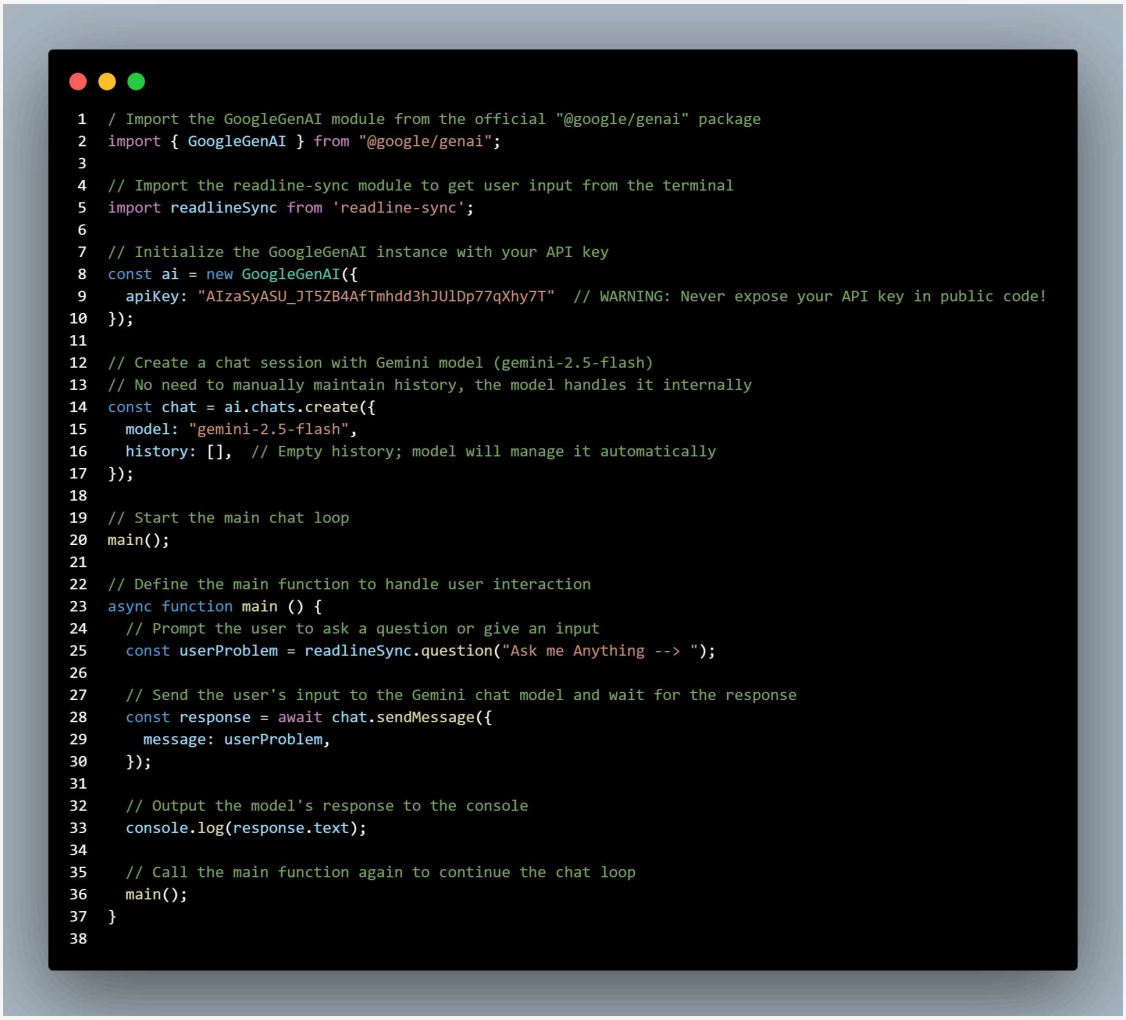
💡 इसका reason: ChatGPT को **short-term context** भेजा जाता है।
Long-term memory by default नहीं होती (unless explicitly enabled).

🧠 How History Works Internally (Behind the Scenes)

```
history = [
  { role: "user", part: ["Hi, I am Harshal"] },
  { role: "assistant", part: ["Hi Harshal, nice to meet you!"] }
]
```

- Context is dynamically updated via such conversation arrays.
- LLM doesn't "remember" — it **relies on the visible context.**

🧰 Tool Integration: Example – Using Gemini via JavaScript :



```
/ Import the GoogleGenAI module from the official "@google/genai" package
import { GoogleGenAI } from "@google/genai";

// Import the readline-sync module to get user input from the terminal
import readlineSync from 'readline-sync';

// Initialize the GoogleGenAI instance with your API key
const ai = new GoogleGenAI({
  apiKey: "AIzaSyASU_JT5ZB4AfTmhdd3hJUlDp77qXhy7T"  // WARNING: Never expose your API key in public code!
});

// Create a chat session with Gemini model (gemini-2.5-flash)
// No need to manually maintain history, the model handles it internally
const chat = ai.chats.create({
  model: "gemini-2.5-flash",
  history: [],  // Empty history; model will manage it automatically
});

// Start the main chat loop
main();

// Define the main function to handle user interaction
async function main () {
  // Prompt the user to ask a question or give an input
  const userProblem = readlineSync.question("Ask me Anything --> ");

  // Send the user's input to the Gemini chat model and wait for the response
  const response = await chat.sendMessage({
    message: userProblem,
  });

  // Output the model's response to the console
  console.log(response.text);

  // Call the main function again to continue the chat loop
  main();
}
```

⚠️ **Warning:** Never expose API keys publicly — they can be misused.

📚 Summary: Key Takeaways

✅ **LLM is a prediction model**, not a calculator or live-data fetcher.
✅ It **predicts answers** based on trained data and patterns.
✅ It **can't run code**, but can write code (e.g., Python, JS).
✅ For calculations or real-time data, LLM uses **external tools.**
✅ LLM has **no real memory** — it relies on **context window.**
✅ ChatGPT-style models hold **short-term conversation memory** using message history arrays.
✅ LLMs can integrate with tools like **Python, Gemini, APIs** for enhanced capabilities.

🧠 **Pro Tip:**
When you ask LLM:

> "What's the temperature in Varanasi?"

It doesn't "know" it — it either:

- Predicts from training data, or
- Uses an API/tool (if integrated) to fetch the answer.