# INSTALL REQUIRED LIBRARIES

!pip install openai tavily-python

# IMPORT LIBRARIES

from datetime import datetime

from tavily import TavilyClient

from openai import OpenAI

# ENTER YOUR API KEYS

OPENAI\_API\_KEY = "sk-proj-k6X6d6beIcmrsCDOnadgpWyCY\_awOV4KqzzhYpF\_jObHwf4fYgOHwwq5JWdjOwSw2ddCgXHBF3T3BlbkFJH0mv4Ef9grvoU8hBd3VNIu9ehWGi60Bx6wuUy7HeALhvDCGbLqgF7q7KFlNkPTBw7sAm-9FpgA"     # Replace with your OpenAI API key

TAVILY\_API\_KEY = "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"   # 👉 Replace with your Tavily API key

# ⚙ INITIALIZE CLIENTS

client = OpenAI(api\_key=OPENAI\_API\_KEY)

tavily = TavilyClient(api\_key=TAVILY\_API\_KEY)

# 🗓 FUNCTION TO GET CURRENT DATE

def get\_current\_date():

    """Return today's date in YYYY-MM-DD format."""

    return datetime.now().strftime("%Y-%m-%d")

# 🌐 FUNCTION TO PERFORM WEB SEARCH

def search\_web(query):

    """Perform live search using Tavily API."""

    try:

        results = tavily.search(query)

        if "results" in results and len(results["results"]) > 0:

            return results["results"][0]["content"]

        else:

            return "No relevant results found."

    except Exception as e:

        return f"Error: {e}"

# 🤖 FUNCTION TO ASK LLM (OpenAI GPT)

def ask\_llm(user\_query, web\_info, current\_date):

    """Ask the LLM to generate a smart answer using context."""

    try:

        response = client.chat.completions.create(

            model="gpt-4o-mini",

            messages=[

                {"role": "system", "content": f"Today's date is {current\_date}. Use it for reasoning."},

                {"role": "user", "content": f"User question: {user\_query}\nWeb info: {web\_info}"}

            ]

        )

        return response.choices[0].message.content

    except Exception as e:

        return f"Error generating answer: {e}"

# 🔁 MAIN FUNCTION TO RUN THE APP

def run\_llm\_app():

    print("🤖 Welcome to the Infosys Springboard LLM App!")

    print("You can ask any question (Example: 'Who won the IPL 2024?')\n")

    user\_query = input("💬 Enter your question: ")

    if not user\_query.strip():

        print("⚠ Please enter a valid question.")

        return

    current\_date = get\_current\_date()

    print(f"\n📅 Current Date: {current\_date}")

    print(f"🔍 Searching for: {user\_query}")

    search\_query = f"{user\_query} (as of {current\_date})"

    web\_info = search\_web(search\_query)

    print("\n🌐 Web Search Result (Summary):")

    print(web\_info[:500], "...")

    final\_answer = ask\_llm(user\_query, web\_info, current\_date)

    print("\n💡 Final Answer:")

    print(final\_answer)

# 🚀 RUN THE APP

run\_llm\_app()