

## Chủ đề chính

- Tiếp tục với **Transformers** và ứng dụng các mô hình tiên tiến nhất.
- Sử dụng **API** của OpenAI, Anthropic (Claude) và Google để tích hợp AI vào các ứng dụng thực tế.

## Chuyển sang chủ đề mới: Giới thiệu về Gradio

Tại sao Gradio đặc biệt và hữu ích?

- Gradio giúp **tạo giao diện người dùng nhanh chóng** mà không cần kiến thức sâu về lập trình frontend.
- Hỗ trợ **triển khai nguyên mẫu AI**, giúp **data scientists** và **engineers** dễ dàng kiểm thử mô hình và chia sẻ với người dùng cuối.
- Đơn giản hơn nhiều so với các công cụ như **React**, giúp giảm đáng kể lượng **boilerplate code**.
- Là một phần của **Hugging Face**, hỗ trợ mạnh mẽ trong việc xây dựng ứng dụng **machine learning**.

## Cách sử dụng Gradio

Chỉ cần một dòng mã là có thể khởi tạo giao diện từ một hàm đơn giản!

Ví dụ:

```
import gradio as gr

def greet(name):
    return f"Xin chào, {name}!"

demo = gr.Interface(fn=greet, inputs="text", outputs="text")
```

`demo.launch()` **Kế hoạch triển khai**

- Xây dựng giao diện **UI cho các API GPT, Claude, Gemini**.
  - Phát triển **UI cho brochure** đã thiết kế trong các bài trước.
  - **Thêm tính năng streaming và markdown** để cải thiện trải nghiệm người dùng.
- => **Mục tiêu:** Tạo một hệ thống linh hoạt, dễ sử dụng để thử nghiệm và triển khai mô hình AI!