Project 22: Book Recommender

Description:
This chatbot suggests books tailored to your interests, mood, or favorite authors. Whether you're craving thrillers, looking for feel-good reads, or exploring self-help, this AI-powered recommender curates a list just for you—with mini summaries to boot.

---

book_recommender.py

```
import openai
import os
import gradio as gr

# Load OpenAI API key securely from environment variable
openai.api_key = os.getenv("OPENAI_API_KEY")

# Function to provide personalized book suggestions
def recommend_books(user_input):
    # System prompt sets up the AI as a book-loving assistant
    messages = [
        {
            "role": "system",
            "content": (
                "You are a helpful and passionate book recommender. Based on a
                "suggest a few great books. For each book, give the title, aut
                "Be warm and encouraging in your tone."
            )
        },
        {
            "role": "user",
            "content": f"I'm looking for: {user_input}"
        }
    ]

    try:
        # Send the request to OpenAI's API
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",  # Use GPT-4 for even more nuance
            messages=messages
        )

        # Return the curated book list
        return response['choices'][0]['message']['content'].strip()

    except Exception as e:
        # Display error if something goes wrong
        return f"Error: {str(e)}"

# Gradio interface for book suggestions
iface = gr.Interface(
    fn=recommend_books,                            # Core function
    inputs=gr.Textbox(lines=2, placeholder="e.g. I like mystery novels with pl
    outputs="text",                                # Display output
    title="📚 Book Recommender Bot",                # App title
    description=(
        "Tell me what kind of books you're in the mood for and I'll recommend
        "Try things like: 'Books like The Alchemist', 'Dark fantasy adventures
    )
)

# Launch the web app
iface.launch()
```