Project 33: Blog Post Generator

Description:

This tool generates complete blog posts based on a topic, audience, and desired tone. Whether it's technical, motivational, informative, or casual—it produces well-structured blog content that's ready to post or personalize.

`blog_post_generator.py`

```
import openai
import os
import gradio as gr

# Load OpenAI API Key
openai.api_key = os.getenv("OPENAI_API_KEY")

# Function to generate a blog post
def generate_blog(topic, audience, tone):
    # System message to guide assistant as a blog writer
    messages = [
        {
            "role": "system",
            "content": (
                f"You are a skilled blog writer. Write a complete blog post ab
                "Include an engaging introduction, 2-3 key sections, and a cor
            )
        },
        {
            "role": "user",
            "content": f"Blog topic: {topic}"
        }
    ]

    try:
        # Send request to OpenAI API
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=messages
```

```
        )

        # Return blog content
        return response["choices"][0]["message"]["content"].strip()

    except Exception as e:
        # Return error
        return f"Error: {str(e)}"

# Gradio interface for blog generation
iface = gr.Interface(
    fn=generate_blog,                                    # Blog generator f
    inputs=[
        gr.Textbox(label="Topic (e.g., AI in Education)"),
        gr.Textbox(label="Target Audience (e.g., high school students, startup
        gr.Radio(["informative", "motivational", "casual", "technical"], label
    ],
    outputs="text",                                      # Blog post result
    title="📝 Blog Post Generator",
    description="Enter a blog topic, your audience, and tone—and I'll generate
)

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