

Project 11: TL;DR Summarizer

Description:

This tool takes long text and summarizes it into a short, digestible version—like a TL;DR. It's great for summarizing articles, essays, notes, or documents into concise highlights. The user pastes in a wall of text, and the bot returns the core message in just a few lines.

tldr_summarizer.py

```
import openai
import os
import gradio as gr

# Load your OpenAI API key from environment or set it directly
openai.api_key = os.getenv("OPENAI_API_KEY")

# Function that summarizes long input text
def summarize_text(long_text):
    # Define system prompt to instruct the assistant to be a summarizer
    messages = [
        {
            "role": "system",
            "content": (
                "You are a professional summarizer. Take long passages of text\n"
                "Focus on the main idea and key takeaways. Output should be 3-"
            )
        },
        {
            "role": "user",
            "content": f"Summarize this:\n\n{long_text}" # Input from user
        }
    ]

    try:
        # Call the OpenAI API to generate the summary
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo", # Use GPT-4 if preferred
            messages=messages
```

```

    )

    # Extract and return summary content
    return response['choices'][0]['message']['content'].strip()

except Exception as e:
    # Return error if something fails (e.g., missing key)
    return f"Error: {str(e)}"

# Gradio interface for easy interaction
iface = gr.Interface(
    fn=summarize_text,                # The function to run
    inputs="textbox",                # Multiline input for long text
    outputs="text",                  # Output summary
    title="📝 TL;DR Summarizer",      # Title of the app
    description=(
        "Paste in any long text and get a short TL;DR-style summary. "
        "Try: news articles, essays, blogs, or meeting notes."
    )
)

# Launch the app locally in your browser
iface.launch()

```