

Project 75: Meeting Agenda Generator

Description:

This assistant helps you craft a well-structured meeting agenda based on the purpose, attendees, and time constraints. It includes topics, time blocks, and action goals—perfect for keeping meetings on track and outcome-driven.

meeting_agenda_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 meeting agenda
def generate_meeting_agenda(meeting_type, attendees, duration):
    # Prompt AI to act like a meeting planning assistant
    messages = [
        {
            "role": "system",
            "content": (
                "You are a meeting planning assistant. Based on the meeting ty\n"
                "create a clear agenda. Include topic headings, estimated time\n"
                "Keep it organized and professional."
            )
        },
        {
            "role": "user",
            "content": (
                f"Meeting Type: {meeting_type}\n"
                f"Attendees: {attendees}\n"
                f"Total Duration: {duration} minutes"
            )
        }
    ]
```

```

try:
    # Generate the agenda with OpenAI
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=messages
    )
    return response["choices"][0]["message"]["content"].strip()

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

# Gradio interface
iface = gr.Interface(
    fn=generate_meeting_agenda,
    inputs=[
        gr.Textbox(label="Type of Meeting (e.g. Sprint Review, Product Brainst
        gr.Textbox(label="Who's Attending? (e.g. Devs, Designers, PMs)"),
        gr.Slider(15, 120, step=5, label="Total Duration (minutes)")
    ],
    outputs="text",
    title="📅 Meeting Agenda Generator",
    description="Describe the meeting and attendees, and I'll give you a struc
)

# Launch the agenda planner
iface.launch()

```