

# Project 68: Email Draft Generator

---

## Description:

This email assistant crafts professional, polite, or casual emails based on your intent and tone. Whether it's following up, requesting something, declining politely, or introducing yourself—it creates a clear draft ready to send!

---

## email\_draft\_generator.py

```
import openai
import os
import gradio as gr

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

# Function to generate an email draft
def generate_email_draft(purpose, recipient_description, tone):
    # Prompt the AI to behave like a communication assistant
    messages = [
        {
            "role": "system",
            "content": (
                f"You are a professional email writing assistant. Based on the"
                f"draft a clear and well-structured email. Make sure the messa
            )
        },
        {
            "role": "user",
            "content": (
                f"Purpose: {purpose}\n"
                f"Recipient Description: {recipient_description}\n"
                f"Tone: {tone}"
            )
        }
    ]

    try:
```

```

# Request email draft from OpenAI
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo", # GPT-4 for extra polish
    messages=messages
)

# Return the email draft
return response["choices"][0]["message"]["content"].strip()

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

# Gradio UI for generating email drafts
iface = gr.Interface(
    fn=generate_email_draft,
    inputs=[
        gr.Textbox(label="What's the Purpose of the Email?"),
        gr.Textbox(label="Describe the Recipient (e.g. my manager, a potential client, a friend, etc.)"),
        gr.Radio(["Formal", "Professional", "Friendly", "Casual"], label="Select Tone"),
    ],
    outputs="text",
    title="✉ Email Draft Generator",
    description="Describe your purpose and audience, and get a well-crafted email draft."
)

# Launch the email assistant
iface.launch()

```