

Project 90: Chat with Terms & Conditions File

Description:

This bot lets users upload a Terms & Conditions document (PDF or TXT), then ask natural questions like “Can I cancel anytime?” or “How is my data used?” The assistant reads and responds in plain English using only the content from the legal file.

chat_with_terms.py

```
import os
import openai
import gradio as gr
import PyPDF2

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

# Function to extract text from a T&C file
def extract_terms_text(file):
    try:
        if file.name.endswith(".pdf"):
            reader = PyPDF2.PdfReader(file)
            return "\n".join([page.extract_text() for page in reader.pages])
        else:
            return file.read().decode("utf-8")
    except Exception as e:
        return f"Error extracting text: {str(e)}"

# Function to answer question using legal content
def chat_with_terms(file, question):
    try:
        terms_text = extract_terms_text(file)

        prompt = (
            "You are a legal assistant. Use only the following Terms and Condi\n"
            "in clear, everyday language. Avoid guessing or inventing informat\n"
            f"{terms_text[:4000]}\n\n"
            f"Question: {question}\nAnswer:"
        )
```

```

    )

    # Query OpenAI
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}]
    )

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

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

# Gradio UI
iface = gr.Interface(
    fn=chat_with_terms,
    inputs=[
        gr.File(label="Upload Terms & Conditions (.txt or .pdf)", file_types=[
            gr.Textbox(label="Ask a Question (e.g. Can I get a refund?)")
        ],
    outputs="text",
    title="📖 Chat with Terms & Conditions",
    description="Upload your T&C file and ask questions about your rights, ref
)

# Launch it!
iface.launch()

```