

Project 89: Custom FAQ Bot (from File)

Description:

This chatbot reads your FAQ document (text, Markdown, or PDF), understands the questions and answers, and lets users ask new questions. It retrieves the most relevant FAQs and answers like a helpful support agent.

custom_faq_bot.py

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

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

# Function to extract text from FAQ file
def extract_faq_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 FAQ: {str(e)}"

# Main Q&A function
def faq_chat(faq_file, question):
    try:
        faq_text = extract_faq_text(faq_file)

        prompt = (
            "You are a smart FAQ assistant. Answer the user's question using c"
            f"{faq_text[:4000]}\n\n" # Truncate for token safety
            f"User's Question: {question}\nAnswer:"
        )
```

```

# OpenAI call
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 answering question: {str(e)}"

# Gradio interface
iface = gr.Interface(
    fn=faq_chat,
    inputs=[
        gr.File(label="Upload FAQ File (.txt, .md, .pdf)", file_types=[".txt",
        gr.Textbox(label="Ask a Question")
    ],
    outputs="text",
    title="📄 Custom FAQ Bot",
    description="Upload your FAQ doc and turn it into a smart chatbot that ans
)

# Launch the bot
iface.launch()

```