

Project 88: Product Manual Assistant

Description:

This assistant helps users understand and troubleshoot any product by letting them upload a manual (PDF or text). It parses the content and allows users to ask questions like “How do I reset the device?” or “What does the LED blink mean?”

product_manual_assistant.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 PDF or TXT
def extract_manual_text(file):
    try:
        if file.name.endswith(".pdf"):
            pdf_reader = PyPDF2.PdfReader(file)
            return "\n".join([page.extract_text() for page in pdf_reader.pages])
        else:
            return file.read().decode("utf-8")
    except Exception as e:
        return f"Error reading file: {str(e)}"

# Chat with the product manual
def ask_product_manual(file, user_question):
    try:
        manual_text = extract_manual_text(file)

        # Create a structured prompt
        prompt = (
            f"You are a helpful assistant that answers questions based on this\n\n"
            f"{manual_text[:4000]}\n\n" # Truncate to fit model input
            f"Question: {user_question}\nAnswer:"
        )
```

```

    )

    # Ask 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: {str(e)}"

# Gradio interface
iface = gr.Interface(
    fn=ask_product_manual,
    inputs=[
        gr.File(label="Upload Product Manual (.pdf or .txt)", file_types=[".pdf", ".txt"]),
        gr.Textbox(label="Ask a Question (e.g. How do I reset it?)"),
    ],
    outputs="text",
    title="📖 Product Manual Assistant",
    description="Upload a product manual and ask questions like setup instructions"
)

# Launch the app
iface.launch()

```