

Project 85: Chat with Markdown Notes

Description:

This project allows users to upload a .md file (Markdown format) and ask questions about its contents. Whether it's project notes, documentation, or study guides, the assistant can extract insights and answer contextually from the file.

chat_with_markdown.py

```
import os
import openai
import gradio as gr

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

# Function to chat with Markdown file content
def chat_with_markdown(markdown_file, user_question):
    try:
        # Read and decode Markdown file
        content = markdown_file.read().decode("utf-8")

        # Prompt to use markdown content as context
        prompt = (
            f"You are an assistant that answers questions based on the user's\n"
            f"Here are the notes:\n{content[:4000]}\n\n" # Trim for token limit
            f"Question: {user_question}\nAnswer:"
        )

        # Generate response from 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 reading markdown file: {str(e)}"

# Gradio interface
iface = gr.Interface(
    fn=chat_with_markdown,
    inputs=[
        gr.File(label="Upload .md File", file_types=[".md"]),
        gr.Textbox(label="Ask a Question About the Notes")
    ],
    outputs="text",
    title="📝 Chat with Markdown Notes",
    description="Upload your markdown file and ask any question about its content"
)

# Launch the chatbot
iface.launch()

```