

Project 56: Coding Concept Explainer

Description:

This tool explains programming concepts in plain, beginner-friendly language. Whether you want to understand recursion, APIs, variables, or OOP—it provides examples, analogies, and even simple code snippets to help you truly get it.

`coding_concept_explainer.py`

```
import openai
import os
import gradio as gr

# Load the OpenAI API Key
openai.api_key = os.getenv("OPENAI_API_KEY")

# Function to explain coding concepts
def explain_concept(concept, level):
    # Prompt the assistant to act like a coding mentor
    messages = [
        {
            "role": "system",
            "content": (
                f"You are a friendly coding teacher. Explain the concept '{concept}' "
                "Include a definition, a simple analogy, a code example (in Py
            )
        },
        {
            "role": "user",
            "content": concept
        }
    ]

    try:
        # Get concept explanation from OpenAI
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=messages
```

```

    )

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

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

# Gradio interface
iface = gr.Interface(
    fn=explain_concept,
    inputs=[
        gr.Textbox(label="Enter a Coding Concept (e.g. recursion, list comprehensions)"),
        gr.Radio(["beginner", "intermediate", "advanced"], label="Explanation Level"),
    ],
    outputs="text",
    title="💡 Coding Concept Explainer",
    description="Type a coding term and choose your level. Get a clear explanation."
)

# Launch the explainer
iface.launch()

```