

PROJECT: PERSONALIZED LANGUAGE TUTOR



You will leverage OpenAI's API to build a dynamic, AI-powered language learning assistant. This project focuses on implementing key features that enhance the user's language skills.

The project will provide sample audio data (`data/sample.wav`) and utilize OpenAI's speech-to-text model to transcribe the audio into text. Furthermore, using OpenAI's `gpt-4o-mini` model will enable the translation of the text into target languages, facilitating seamless cross-lingual communication. The Grammar Check feature will detect and suggest corrections for grammatical errors in user-generated text, helping users refine their writing skills through AI-driven feedback. Finally, Pronunciation Feedback will evaluate users' spoken language and provide personalized improvement suggestions, assisting them in mastering proper pronunciation.

By integrating these features using OpenAI's API, the Personalized Language Tutor will offer comprehensive, AI-enhanced features to support language learners. Your challenge is implementing these tools seamlessly, creating a user-friendly experience that harnesses AI's potential to transform language learning.

The data

There is one audio file named `sample.wav`. This audio file contains Harvard sentences. These sentences are phonetically balanced, meaning they use specific phonemes at the same frequency they appear in English, and they are designed to cover a wide range of sounds and contexts. An example of a Harvard sentence is: "The stale smell of old beer lingers."

```
import os
import openai
from openai import OpenAI
```

```
# Continue coding here
# Use as many cells as you like
import os
import openai
from openai import OpenAI
client = OpenAI()

audio_path = "data/sample.wav"
target_language="French"
with open(audio_path, "rb") as audio_file:
    transcription_text = client.audio.transcriptions.create(
        file=audio_file,
        model="whisper-1",
        response_format="text",
        language="en"
    )

response=client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role":"user", "content": f"translate to {target_language} : {transcription_text}" }]
)
translated_text=response.choices[0].message.content

response1=client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[ {"role":"system", "content": "You are a grammar expert. Correct grammatical errors and return the corrected text only." },
              {"role":"user", "content": f"correct this {translated_text}" }]
)
grammar_feedback=response1.choices[0].message.content

target_text=grammar_feedback

response2=client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[ {"role":"system", "content": (
                    "You are a pronunciation expert. The corrected text represents the ideal "
                    "pronunciation. Compare it with the user's spoken transcription and provide "
                    "specific suggestions for improving pronunciation, stress, and clarity."
                )},
              {"role":"user", "content": (
                    f"User spoken transcription:\n{transcription_text}\n\n"
                    f"Ideal (corrected) text:\n{target_text}\n\n"
                    "Provide pronunciation improvement feedback."
                )}
            ]
)
```

)

```
pronunciation_feedback=response2.choices[0].message.content
```