import google.generativeai as genai

import speech\_recognition as sr

import pyttsx3

import time

# Initialize speech engine

engine = pyttsx3.init()

engine.setProperty('rate', 150) # Speed of speech

engine.setProperty('volume', 1.0) # Volume (0.0 to 1.0)

# Initialize speech recognizer

recognizer = sr.Recognizer()

def speak(text):

print("\nConvoBot:", text)

engine.say(text)

engine.runAndWait()

def listen():

with sr.Microphone() as source:

print("\nListening...")

recognizer.adjust\_for\_ambient\_noise(source, duration=0.5)

try:

audio = recognizer.listen(source, timeout=5)

text = recognizer.recognize\_google(audio)

print("You:", text)

return text

except sr.WaitTimeoutError:

speak("I couldn't hear anything. Please try again.")

return None

except sr.UnknownValueError:

speak("I couldn't understand that. Please try again.")

return None

except Exception as e:

print(f"Error: {str(e)}")

return None

genai.configure(api\_key="AIzaSyCS\_RiG\_ENKDiOet8SYxCwENpIIocKIHs0")

generation\_config = {

"temperature": 1,

"top\_p": 0.95,

"top\_k": 40,

"max\_output\_tokens": 8192,

"response\_mime\_type": "text/plain",

}

model = genai.GenerativeModel(

model\_name="gemini-2.0-flash",

generation\_config=generation\_config,

)

# Initialize chat history

chat\_history = []

initial\_prompts = [

{

"input": "Who are you?",

"output": "Hello! I'm ConvoBot, your AI English tutor from CONVO COACH. I'm designed to help you master English through natural conversations. I can assist with grammar, pronunciation, vocabulary, and real-world communication scenarios. How would you like to improve your English today?"

},

{

"input": "What can you teach me?",

"output": "As your CONVO COACH tutor, I can help you with: 1) Grammar correction and explanations 2) Pronunciation practice 3) Vocabulary building 4) Conversation practice 5) Business English 6) Idioms and expressions. What would you like to focus on?"

},

{

"input": "I want to practice speaking",

"output": "Great choice! At CONVO COACH, we believe in learning through real conversations. Let's start with a role-play scenario. Would you prefer: 1) Job interview practice 2) Restaurant ordering 3) Business meeting, or 4) Casual conversation?"

},

{

"input": "My grammar is not good",

"output": "Don't worry! CONVO COACH specializes in making grammar learning natural and fun. Let's start by you telling me about your day, and I'll help correct any mistakes while explaining the rules in a simple way. How does that sound?"

}

]

# Modified chat loop

speak("Welcome to CONVO COACH English Tutor! Say 'exit' to end the conversation.")

# Initialize chat with personality context

chat\_history = []

for prompt in initial\_prompts:

chat\_history.extend([

f"input: {prompt['input']}",

f"output: {prompt['output']}"

])

while True:

user\_input = listen()

if user\_input is None:

continue

if user\_input.lower() == 'exit':

speak("Thank you for learning with CONVO COACH! Have a great day!")

break

chat\_history.extend([

f"input: {user\_input}",

"output: As your CONVO COACH tutor, "

])

try:

response = model.generate\_content(chat\_history[-10:]) # Keep last 5 exchanges for context

speak(response.text)

chat\_history[-1] = f"output: {response.text}"

except Exception as e:

speak("I apologize, but I couldn't process that. Could you rephrase your question?")

chat\_history = chat\_history[:-2]