import tkinter as tk

from tkinter import messagebox

from PIL import Image, ImageTk

from tkinter import ttk # Import ttk for the progress bar

import speech\_recognition as sr # Import speech recognition

import requests

from io import BytesIO

class SoftSkillAI:

def \_init\_(self):

self.window = tk.Tk() # Create a new Tkinter window

self.window.title("Soft Skill Enhancement AI")

# Initialize score tracking

self.score = 0

# Load the main image

try:

image\_url = "https://images.app.goo.gl/aD5YZsP4A9r13HPeA"

response = requests.get(image\_url)

image = Image.open(BytesIO(response.content))

photo = ImageTk.PhotoImage(image)

except Exception as e:

print(f"Error loading main image: {e}")

photo = None

# Display the main image

if photo:

label = tk.Label(self.window, image=photo)

label.image = photo # Keep a reference to prevent garbage collection

label.pack()

# Create buttons for each soft skill

self.communication\_skills\_button = tk.Button(self.window, text="Communication Skills", command=self.communication\_skills)

self.communication\_skills\_button.pack()

self.teamwork\_collaboration\_button = tk.Button(self.window, text="Teamwork and Collaboration", command=self.teamwork\_collaboration)

self.teamwork\_collaboration\_button.pack()

# Add score display

self.score\_label = tk.Label(self.window, text=f"Score: {self.score}")

self.score\_label.pack()

# Add a progress bar

self.progress = ttk.Progressbar(self.window, length=200, mode='determinate')

self.progress.pack()

def update\_score(self, correct):

if correct:

self.score += 1

self.score\_label.config(text=f"Score: {self.score}")

self.update\_progress()

def update\_progress(self):

self.progress['value'] += 50 # Assuming each exercise adds 50% to the progress bar

def communication\_skills(self):

# Display content for communication skills

content\_window = tk.Toplevel(self.window)

content\_window.title("Communication Skills")

# Display description

description\_label = tk.Label(content\_window, text="Learn how to communicate effectively with active listening.")

description\_label.pack()

# Display image

try:

image\_url = "https://images.app.goo.gl/qijYMq78q8tfPGKA8"

response = requests.get(image\_url)

image = Image.open(BytesIO(response.content))

photo = ImageTk.PhotoImage(image)

except Exception as e:

print(f"Error loading communication skills image: {e}")

photo = None

if photo:

label = tk.Label(content\_window, image=photo)

label.image = photo

label.pack()

# Display exercise/quiz

exercise\_label = tk.Label(content\_window, text="Exercise: What is active listening?")

exercise\_label.pack()

options = ["Giving advice", "Asking questions", "Paying attention to the speaker"]

variable = tk.StringVar(content\_window)

variable.set(options[0])

option\_menu = tk.OptionMenu(content\_window, variable, \*options)

option\_menu.pack()

# Add voice input button

voice\_button = tk.Button(content\_window, text="Answer with Voice", command=lambda: self.speech\_recognition(content\_window, "Paying attention to the speaker"))

voice\_button.pack()

submit\_button = tk.Button(content\_window, text="Submit", command=lambda: self.check\_answer(variable.get(), content\_window, "Paying attention to the speaker"))

submit\_button.pack()

def teamwork\_collaboration(self):

# Display content for teamwork and collaboration

content\_window = tk.Toplevel(self.window)

content\_window.title("Teamwork and Collaboration")

# Display description

description\_label = tk.Label(content\_window, text="Understand the importance of working together as a team.")

description\_label.pack()

# Display image

try:

image\_url = "https://images.app.goo.gl/qijYMq78q8tfPGKA8"

response = requests.get(image\_url)

image = Image.open(BytesIO(response.content))

photo = ImageTk.PhotoImage(image)

except Exception as e:

print(f"Error loading teamwork collaboration image: {e}")

photo = None

if photo:

label = tk.Label(content\_window, image=photo)

label.image = photo

label.pack()

# Display exercise/quiz

exercise\_label = tk.Label(content\_window, text="Exercise: What is the importance of teamwork?")

exercise\_label.pack()

options = ["Increased productivity", "Improved communication", "Enhanced creativity"]

variable = tk.StringVar(content\_window)

variable.set(options[0])

option\_menu = tk.OptionMenu(content\_window, variable, \*options)

option\_menu.pack()

# Add voice input button

voice\_button = tk.Button(content\_window, text="Answer with Voice", command=lambda: self.speech\_recognition(content\_window, "Increased productivity"))

voice\_button.pack()

submit\_button = tk.Button(content\_window, text="Submit", command=lambda: self.check\_answer(variable.get(), content\_window, "Increased productivity"))

submit\_button.pack()

def speech\_recognition(self, window, correct\_answer):

recognizer = sr.Recognizer()

with sr.Microphone() as source:

try:

print("Listening for your answer...")

audio = recognizer.listen(source)

recognized\_text = recognizer.recognize\_google(audio)

print(f"You said: {recognized\_text}")

self.check\_answer(recognized\_text, window, correct\_answer)

except sr.UnknownValueError:

messagebox.showerror("Error", "Sorry, I did not understand that. Please try again.")

except sr.RequestError as e:

messagebox.showerror("Error", f"Could not request results; {e}")

except Exception as e:

messagebox.showerror("Error", f"An unexpected error occurred: {e}")

def check\_answer(self, answer, window, correct\_answer):

if answer.lower() == correct\_answer.lower():

messagebox.showinfo("Correct!", "Your answer is correct!")

self.update\_score(True)

else:

messagebox.showinfo("Incorrect", "Your answer is incorrect. Try again!")

self.update\_score(False)

window.destroy()

def run(self):

self.window.mainloop()

if \_name\_ == "\_main\_":

app = SoftSkillAI()

app.run()