AI-Powered Virtual Interviewer — Updated Blueprint with UI

Overview

This updated project blueprint builds upon the previous version by adding a **user-friendly web interface** using Gradio. Participants can now select a role, enter their name, record answers via a mic button, and receive per-question feedback in a clean browser interface. The final screen provides overall ratings and a downloadable report.

Updated Requirements

```
scikit-learn
nltk
spacy
pandas
numpy
matplotlib
pdfplumber
opencv-python
SpeechRecognition
pyaudio
gradio
```

New File: src/ui_gradio.py

```
import os, json, numpy as np
import gradio as gr
import speech_recognition as sr

from utils import load_questions, append_session_log, now, REPORTS
from scoring import score_answer
from report import save_report
from behavior import audio_proxies_from_pcm, behavior_rating, capture_webcam_metrics

def transcribe_from_gradio(numpy_audio, sample_rate):
    if numpy_audio is None:
        return "", b""
    arr = np.asarray(numpy_audio, dtype=np.float32)
    if arr.ndim == 2:
        arr = arr.mean(axis=1)
    target_sr = 16000
    if sample_rate != target_sr:
        import numpy as _np, math
        ratio = target_sr / float(sample_rate)
        new_len = int(math.ceil(len(arr) * ratio))
        x_old = _np.linspace(0, 1, len(arr), endpoint=False)
        x_new = _np.linspace(0, 1, new_len, endpoint=False)
        arr = _np.interp(x_new, x_old, arr).astype(np.float32)
    arr = np.clip(arr, -1.0, 1.0)
    int16 = (arr * 32767.0).astype(np.int16)
    pcm_bytes = int16.tobytes()

    r = sr.Recognizer()
    audio = sr.AudioData(pcm_bytes, target_sr, 2)
    try:
        text = r.recognize_google(audio)
    except Exception:
        text = ""
    return text, pcm_bytes
# (UI code continues as shown in the assistant response)
```

Update to src/interviewer.py

```
def score_one(item, record_seconds=8, use_webcam=False):
    webcam = {"face_visible_ratio":0,"centered_ratio":0,"avg_brightness":0,"motion_jitter":0,"eye_contac
    if use_webcam:
        webcam = capture_webcam_metrics(seconds=record_seconds)
    text, pcm = mic_to_text_and_audio(timeout=record_seconds+4)
    ans_score = score_answer(text, item)
    aud_feats = audio_proxies_from_pcm(pcm)
    beh = behavior_rating(webcam, aud_feats)
    return {
        "answer_text": text,
        "answer_score": ans_score,
        "behavior_raw": {"webcam": webcam, "audio": aud_feats},
        "behavior": beh
    }
```

How to Run

```
# Console version (CLI flow):
python -m src.interviewer

# Web interface version (Gradio UI):
python -m src.ui_gradio

# Then open http://127.0.0.1:7860 in your browser.
```