Project 84: Resume Screener with CSV Upload

Description:
This tool helps HR teams and recruiters automatically screen resumes stored in a CSV file. It summarizes candidates based on criteria like skills, experience, and keywords—and returns a shortlist of the best-fit profiles.

resume_screener_csv.py

```
import os
import openai
import gradio as gr
import pandas as pd

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

# Function to screen resumes based on job criteria
def screen_resumes(csv_file, job_role, must_have_skills):
    try:
        # Load resumes
        df = pd.read_csv(csv_file.name)

        # Ensure resume text column exists
        if "resume" not in df.columns:
            return "Error: The CSV must have a 'resume' column containing resu

        shortlisted = []

        # Loop through each resume and evaluate with GPT
        for index, row in df.iterrows():
            resume_text = row["resume"]

            prompt = (
                f"You are a smart HR assistant. Based on the job role '{job_rc
                f"{must_have_skills}, decide if this resume is a good match.\r
                f"Reply with 'MATCH' or 'NO MATCH' and a short reason why."
            )
            response = openai.ChatCompletion.create(
                model="gpt-3.5-turbo",
                messages=[{"role": "user", "content": prompt}]
            )

            result = response["choices"][0]["message"]["content"].strip()
            if "MATCH" in result:
                shortlisted.append({
                    "Index": index,
                    "Result": result,
                    "Resume Snippet": resume_text[:300] + "..."
                })

        if not shortlisted:
            return "No suitable candidates found."

        # Return results
        return "\n\n".join(
            [f"Candidate {c['Index']}:\n{c['Result']}\n{c['Resume Snippet']}"
        )

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

# Gradio interface
iface = gr.Interface(
    fn=screen_resumes,
    inputs=[
        gr.File(label="Upload CSV with 'resume' Column", file_types=[".csv"]),
        gr.Textbox(label="Job Role (e.g. Data Scientist, Sales Executive)"),
        gr.Textbox(label="Must-Have Skills (comma separated)")
    ],
    outputs="text",
    title="📄 Resume Screener (CSV)",
    description="Upload a CSV with resumes and I'll return the top candidates
)

# Launch the screener
iface.launch()
```