**JobGenie: An AI-Powered Career Assistant for Job Seekers**  
A cloud-native, API-driven application that uses NLP and Speech Recognition to help users improve their resumes, understand job descriptions, practice mock interviews, and receive AI-driven feedback.

**Planned Sub-Tasks & Models**

| **#** | **Sub-Task** | **Category** | **Model/API** |
| --- | --- | --- | --- |
| 1 | Extract key skills from Job Description | NLP – NER / Keyphrase Extraction | Hugging Face (spaCy, KeyBERT) |
| 2 | Resume rewriting & optimization | NLP – Text Generation | OpenAI GPT / Llama2 |
| 3 | Voice-based mock interview Q&A | Speech Recognition + NLP QA | Whisper (ASR) + OpenAI GPT |
| 4 | Feedback on interview answers | NLP – Sentiment / Coherence | GPT + Hugging Face classifiers |
| 5 | Job Fit Score (Resume vs JD) | NLP – Similarity Matching | Sentence Transformers (SBERT) |

**1. Environment Setup**

**1.1 Create a new virtual environment:**

python -m venv jobgenie-env

jobgenie-env\Scripts\activate

**1.2 Dependencies**

pip install openai transformers torch gradio sentence-transformers spacy keybert speechrecognition pyaudio git+https://github.com/openai/whisper.git datasets scikit-learn

pip install pipwin

pipwin install pyaudio

python -m pip install --upgrade pip

Note: If pipwin failed to install pyaudio, Use sounddevice Instead of pyaudio

Before that give a try: pip uninstall pyaudio

pip install pyaudio

**NLP sub-task 1: Resume Section Rewriting**

We’ll use OpenAI’s GPT to:

* Analyze a resume section
* Suggest improvements
* Reword the text professionally

**Securely Store OpenAI API Key**

pip install python-dotenv

Refer: <https://www.youtube.com/watch?v=OB99E7Y1cMA&ab_channel=AndersJensen> (How to get API key of the Open API)

Once API key setup is complete,

Run -> python app.py

Open: <http://127.0.0.1:7860/>

Sample section: Worked on cloud computing. Handled few AWS services. Created EC2 instances and S3 buckets. Helped team in deploying applications. Also wrote some documentation and fixed small bugs.

Or

Developed web pages. Wrote code in React. Handled API calls. Worked in a team and attended daily meetings.

Note: If using OpenAPI exceeds your budget then switch to hugging face (Use 1st implementation in resume\_Optimization)

Progress so far:

**What We’ve Accomplished:**

* Set up a clean Python project with venv
* Installed all necessary packages (including transformers, gradio)
* Created .env for secure API key storage (optional for later)
* Built resume\_optimizer.py using **Hugging Face model**
* Created a working **Gradio interface**
* Tested it with real inputs — and it works

**Module 2: job\_description\_analyzer.py**

**Goal:**

Analyze a given Job Description (JD) to extract key skills, responsibilities, and match them with resume sections. Then, calculate a Job Fit Score.

| **Feature** | **Type** | **Tools/Models** |
| --- | --- | --- |
| Extract Skills from JD | NLP – NER / Keyphrase Extraction | KeyBERT |
| Compare Resume & JD | Text Similarity | SentenceTransformers |
| Job Fit Score | Cosine Similarity | SentenceTransformers |

**Step 1: Install Required Packages**

pip install keybert sentence-transformers

Update the job\_description\_analyzer.py and update app.py  
Run app.py

**Example to Try -**

Job Description:

“Looking for a Backend Developer with strong knowledge in Python, FastAPI, Docker, PostgreSQL, and RESTful APIs. Experience with AWS or GCP is a plus.”

Resume Section:

“Developed microservices using FastAPI and Docker. Managed PostgreSQL databases and deployed apps to AWS EC2.”

**Module 3: Voice-based Interview Assistant**

Feature that:

* Asks the user a mock interview question (text)
* Lets the user respond via voice
* Transcribes their voice to text (using Whisper or speech\_recognition)
* Optionally scores or gives feedback on the response

| **Step** | **Task** |
| --- | --- |
| 1️ | Setup voice recording input (with sounddevice or gr.Audio) |
| 2️ | Transcribe voice using whisper or openai-whisper |
| 3️ | Analyze response with a small LLM (e.g., flan-t5 or GPT) |
| 4️ | Display the transcript and feedback on screen |

**Step 1: Install Whisper** also need ffmpeg installed.

pip install git+https://github.com/openai/whisper.git

Download ffmpeg (precompiled)

Go to the official Windows build site:

https://www.gyan.dev/ffmpeg/builds/

* Scroll to "Release builds"
* Click on: ffmpeg-release-essentials.zip (under “Windows EXE files”)
* Extract the zip (e.g., to C:\ffmpeg)

Add ffmpeg to System PATH

You can run ffmpeg from anywhere in terminal:

1. Open Start → Search "Environment Variables"
2. Click "Edit the system environment variables"
3. Click "Environment Variables…"
4. Under System variables, find and select Path, then click Edit…
5. Click New, then paste: C:\ffmpeg\ffmpeg-<version>\bin
6. Verify Installation: ffmpeg -version

**Module 4: Voice Feedback Tracker Dashboard**

* Shows your past transcripts
* Displays AI feedback side-by-side
* Filters by date

Step 1: Install Streamlit and Plotly

pip install streamlit plotly pandas

>>>dashboard.py

**streamlit run dashboard.py**