SmartResume Analyzer

A modern, AI-driven platform to optimize your job search by matching resumes with job descriptions, highlighting strengths and gaps, and offering actionable improvement tips through an interactive web interface.

🔎Features

- **Resume & Job Description Upload**: Drag-and-drop or browse to upload PDF/DOCX files (up to 200 MB each).
- **Overall Match Score**: Calculates a single percentage reflecting resume-to-job alignment.
- **Section-Level Breakdown**: Detailed scores for Skills, Experience, and Education.
- **Keyword Extraction & Highlighting**: Identifies key terms in the JD, highlights matches in green and gaps in red within your resume text.
- **Interactive Resume Editor**: Apply AI-generated improvement tips inline with undo/redo history.
- **Visual Analytics**: Bar charts for section scores, dual word clouds for matched vs. missing keywords, and a timeline slider to filter experiences by year.
- **PDF Report Export**: Generate and download a comprehensive PDF summary of your analysis.

🗜️Quick Start

1. **Clone repository**

```
git clone https://github.com/<your-username>/SmartResumeAnalyzer.git
cd SmartResumeAnalyzer
```

2. **Setup virtual environment**
3. macOS/Linux:

```
python3 -m venv .venv
source .venv/bin/activate
```

4. Windows (PowerShell):

```
python -m venv .venv
Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned
.\.venv\Scripts\Activate.ps1
```

5. **Install dependencies**

```
pip install -r requirements.txt
```

6. **Configure API keys** Create a `.env` file or export in shell:

```
OPENAI_API_KEY=sk-...
PINECONE_API_KEY=...
PINECONE_ENV=us-west1-gcp
```

7. **Run backend**

```
uvicorn main:app --reload
```

Opens FastAPI at `http://localhost:8000`.

8. **Run frontend**

```
streamlit run app.py
```

Opens UI at `http://localhost:8501`.

⚙️Configuration

- **OPENAI_API_KEY**: Your OpenAI secret key for embeddings.
- **PINECONE_API_KEY & PINECONE_ENV**: Your Pinecone credentials and region.
- **ENV Ports**: Backend (`8000`), Frontend (`8501`) can be customized via environment variables.

📂Project Structure

```
├── app.py            # Streamlit frontend UI
├── main.py           # FastAPI backend entrypoint
├── scorer.py         # AI & vector-search logic
├── resume_parser.py  # PDF/DOCX parsing utility
├── tips_generator.py # Improvement-tip engine (optional)
├── requirements.txt  # Python dependencies
├── settings.json     # UI theming and layout config
└── .venv/            # Local Python environment (ignored)
```

Made with 🖱️ by [Your Name]