

AI Agent for Structured Job Information Extraction and Resume Generation

Project Overview

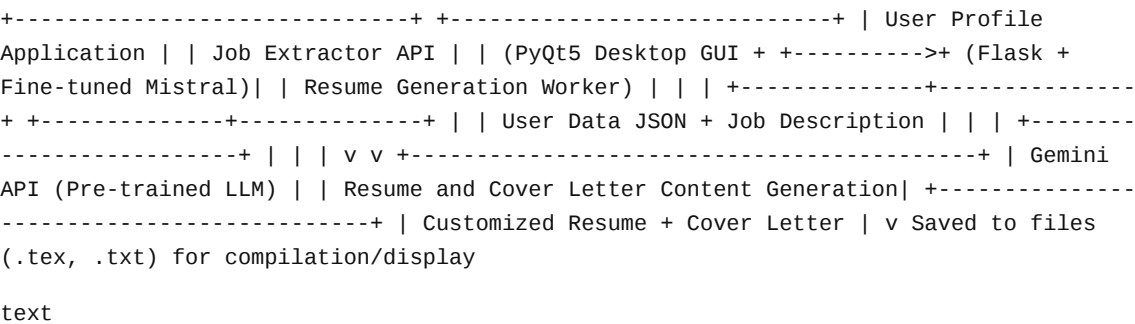
This project is an AI-powered desktop application that automates the generation of tailored resumes and cover letters for job applications. It extracts structured job information from unstructured job descriptions and leverages that to customize application documents based on user profile data.

Manually tailoring resumes and cover letters for each job can be time-consuming and error-prone, especially with unstructured free-text job postings. Our solution streamlines this process with an integrated AI pipeline.

Features

- Extracts key structured fields from any job description:
 - Core Responsibilities
 - Required Skills
 - Educational Requirements
 - Experience Level
 - Preferred Qualifications
 - Compensation and Benefits
- Generates LaTeX-formatted resumes and plain text cover letters customized to the job role
- User-friendly PyQt5 desktop GUI for personal data and job input management
- Asynchronous resume generation with progress tracking
- Saves outputs as `.tex` and `.txt` files for easy compilation and use

Architecture



Components

- **UserProfileApp** (PyQt5 Desktop Application):
 - Collects user and job data
 - Saves data to local JSON files
 - Triggers asynchronous AI resume generation
 - Displays generation progress and results

- **ResumeGenerationWorker** (QThread):
 - Extracts required skills from job descriptions by calling the Job Extractor API
 - Combines user profile and job info, then sends prompt to Gemini API
 - Saves generated resume and cover letter to files
- **Job Extractor API** (Flask Server with fine-tuned Mistral model):
 - Extracts structured job info JSON from unstructured descriptions
 - Serves requests at `/extract` endpoint
- **Gemini API** (Google’s generative AI platform):
 - Generates expertly formatted resumes and cover letters from structured data

Data Design

- **User Profile JSON:**
 - Personal info, education, work experience, projects, skills, leadership, achievements
- **Job Application JSON:**
 - Job title, company, job description text
- **Extracted Job Info JSON:**

```
{ "Core Responsibilities": "...", "Required Skills": "...", "Educational Requirements": "...", "Experience Level": "...", "Preferred Qualifications": "...", "Compensation and Benefits": "..." }
```

text

Technologies Used

Component	Technology	Reason
Desktop GUI	PyQt5	Robust GUI with threading support
Background Worker	QThread	Keeps UI responsive during AI calls
Model Serving API	Flask	Lightweight REST API for inference
LLM Model (Extractor)	Mistral-7B + LoRA	Small VRAM, low hallucination, instruction following
Model Optimization	BitsAndBytes 4-bit Quant	Efficient GPU memory use
Fine-tuning Method	PEFT LoRA	Resource-efficient adapter tuning

Resume Gen API	Google Gemini API	Powerful text generation and formatting
HTTP Client	requests	Reliable REST interaction
Config Management	dotenv	Secure API key management

Design Decisions

- **Separation of Concerns:** Modular design for extraction vs generation allows independent improvements.
 - **Efficiency:** LoRA adapters and quantization support consumer GPU use.
 - **User-Centric:** Intuitive desktop interface with progress bars and error handling.
 - **Security:** API keys stored in environment variables; model runs locally.
 - **Scalability:** API endpoints designed to scale separately from the desktop app.
-

Getting Started

1. Clone the repository.
 2. Install dependencies for Python, PyQt5, and APIs.
 3. Configure environment variables (`GEMINI_API_KEY`).
 4. Run the Flask API server to host the fine-tuned model.
 5. Launch the desktop app, enter profile and job info.
 6. Click **Generate AI Resume** to start the pipeline.
 7. Generated resume (LaTeX) and cover letter (text) saved locally.
-