**Open Resume Auditor for ATS - MVP Document**

### 🚀 Project Title:

**Open Resume Auditor**

### 🔎 Vision Statement:

Empower job seekers with a free, AI-powered resume auditing tool that ensures compliance with ATS standards and increases their chances of landing interviews.

### 🏒 MVP Goal:

Deliver a working prototype that: - Accepts PDF/DOCX resume uploads - Parses and analyzes resumes for ATS compliance - Compares resume against job description - Provides AI-powered improvement suggestions

### 🌐 Tech Stack:

* **Frontend:** Next.js (TypeScript), Tailwind CSS
* **Backend:** Flask (Python)
* **Database:** SQLite
* **AI/NLP Models:** spaCy, sentence-transformers, custom fine-tuned LLM (later)
* **Storage:** Local or Firebase for uploads (optional)

### 🔄 MVP Features:

#### 1. Resume Upload & Parsing

* Drag-and-drop file upload (PDF/DOCX)
* Flask backend parses resume using:
  + pdfplumber (PDF)
  + python-docx (DOCX)
  + spaCy (NER, sections)
* Extracted raw text saved in SQLite

#### 2. ATS Compliance Audit

* Check for:
  + Contact Info
  + Work Experience
  + Education Section
  + Skill Section
  + No tables/images
  + Bullet points presence
* Return an ATS score (0-100%)

#### 3. Job Description Matching

* Input job description (text area)
* NLP to extract required keywords
* Compare against resume
* Show missing/matched keywords

#### 4. AI-Powered Suggestions

* Provide tips on:
  + Formatting
  + Keyword additions
  + Section improvements
* Phase 1: Use GPT or OpenAI API
* Phase 2: Transition to custom fine-tuned LLM

#### 5. Resume Rewriting (Phase 2+)

* Generate rewritten resume snippets
* Option to download improved version

### 🤖 Future LLM Training Plan:

**Phase 1: Dataset Building** - Parse and label uploaded resumes - Store sections, audit scores, and suggestions - Use Hugging Face datasets format

**Phase 2: Model Training** - Fine-tune bert-base-uncased or flan-t5-small - Train on tasks like: - Resume classification - Section detection - Feedback generation

**Phase 3: Integration** - Expose LLM endpoint via Flask - Use locally or on GPU cloud (e.g., Replicate, RunPod)

### 📊 Database Schema (SQLite)

**resumes** - id (int, PK) - file\_name (text) - parsed\_text (text) - ats\_score (int) - uploaded\_at (timestamp)

**job\_descriptions** - id (int, PK) - resume\_id (FK) - text (text)

**improvements** - id (int, PK) - resume\_id (FK) - section (text) - suggestion (text) - rating (optional user feedback)

### 🌐 Frontend Pages (Next.js)

* / Upload resume + paste job description
* /results Show audit score, feedback, missing keywords
* /improve (optional) AI-enhanced resume suggestions

### 🏋️ Project Outcomes

* Increase interview call-backs for job seekers
* Help optimize resumes for real-world job markets
* Build training data for a custom open-source LLM

### 🔢 License:

MIT (Open Source)

### ✨ Maintainer:

Gilbert Kibet Korir GitHub: [github.com/Kibet-g](https://github.com/Kibet-g)