AI Resume Intelligence System - Project Progress Summary

---

1. Project Goal

- Build a serious AI-powered Resume Intelligence System using Flask + PostgreSQL.
- Implement real NLP and ML (not rule-based shortcuts).
- Follow clean architecture and professional Git workflow.

2. Backend Foundation

- Built Flask backend with psycopg2 PostgreSQL integration.
- Implemented full CRUD operations.
- Created resumes table and added resume_text column for NLP.
- Ensured raw resume text storage for future ML flexibility.

3. Resume Text Handling

- Initially planned PDF/DOC upload with text extraction (pdfplumber, python-docx).
- Temporarily switched to JSON-based resume_text for ML development due to tool limitations.
- Stored diverse resume_text entries for meaningful corpus creation.

4. TF-IDF Vectorization

- Extracted all resume_text from database.
- Built TfidfVectorizer from real DB corpus.
- Vocabulary size achieved: 119 features.
- Understood TF, IDF, and vector space representation.

5. Cosine Similarity Engine

- Converted resumes and job descriptions into TF-IDF vectors.
- Computed cosine similarity between job description and all resumes.
- Ranked resumes by similarity score.
- Validated ranking with backend and frontend job descriptions.

6. Understanding Labels & ML Transition

- Explained concept of supervised learning and labels (0/1).
- Discussed weak supervision using similarity threshold.
- Analyzed risks of threshold selection and class imbalance.
- Prepared for Logistic Regression training phase.

7. Git & Project Structure Hygiene

- Separated main and ml-training-phase branches.
- Cleaned repository structure.
- Removed generated artifacts (vectorizer.pkl, uploads, venv) from tracking.
- Ensured professional repository layout.

8. Current System Capabilities

- Resume storage with full text.
- TF-IDF feature extraction from real database.
- Cosine similarity-based ranking.
- Ready for supervised Logistic Regression training.

9. Key Concepts Mastered

- TF-IDF (Term Frequency - Inverse Document Frequency).
- Vector space representation of text.
- Cosine similarity.
- Weak supervision.
- Logistic Regression basics.

- Class imbalance awareness.
- Professional Git workflow.