Software Requirements Specification (SRS)

for AI Resume Analyzer (Full-Stack + ML)

Version: 1.1 (Updated with Adversarial Defense)
Date: October 26, 2023
Status: Draft

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the **AI Resume Analyzer**. This system is designed to automate the initial screening process of recruitment by parsing resumes, extracting key information, and scoring candidates against specific job descriptions using Machine Learning (ML) and Natural Language Processing (NLP).

1.2 Scope

The AI Resume Analyzer is a web-based application that serves two primary user groups: Candidates and Recruiters.

- **For Candidates:** The system analyzes their resume against a target job description, identifying missing keywords and providing suggestions.
- **For Recruiters:** The system allows batch uploading, automatic parsing, and ranking.
- **Security Scope:** The system includes specific defenses against "ATS hacking" techniques and adversarial attacks aimed at manipulating the scoring engine.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **JD:** Job Description
- **NLP:** Natural Language Processing
- **NER:** Named Entity Recognition
- **ATS:** Applicant Tracking System
- **Adversarial Attack:** Malicious inputs designed to confuse or trick the AI model (e.g., invisible text, prompt injection).
- **RBAC:** Role-Based Access Control

2. Overall Description

2.1 Product Functions

- User account management and role separation.
- Resume file uploading with malware/exploit scanning.
- Semantic matching analysis between Resume and JD.
- **Adversarial Defense Layer:** Detecting keyword stuffing and manipulation attempts.
- Report generation.

2.2 User Classes

- **Candidate:** General users looking for resume feedback.
- **Recruiter:** HR professionals requiring batch processing and reliability.
- **Admin:** System maintainers.

2.3 Operating Environment

- **Client:** Modern web browsers (Chrome, Firefox, Safari, Edge).
- **Server:** Dockerized Microservices on Cloud (AWS/GCP).
- **Database:** PostgreSQL + Redis + S3.

3. Specific Requirements

3.1 External Interface Requirements

- **UI:** Next.js (React) with TailwindCSS.
- **API:** RESTful API (NestJS).
- **ML Service:** Python (FastAPI).

3.2 Functional Requirements (FR)

3.2.1 User Authentication & Profile

- **FR-01:** Register/Login via Email/Password and OAuth (Google, LinkedIn).
- **FR-02:** RBAC for Candidates, Recruiters, and Admins.
- **FR-03:** Dashboard for history and saved configurations.

3.2.2 Resume Upload & Parsing

- **FR-04:** Support .pdf, .docx, .txt. Max size 5MB.
- **FR-05:** Extract text preserving logical section order.
- **FR-06 (NER):** Extract Contact Info, Education, Skills, Experience.

3.2.3 AI Analysis & Scoring

- **FR-07:** Calculate Semantic Similarity Score (0-100%) using Transformer models.
- **FR-08:** Keyword Gap Analysis (Hard Match vs. Missing Critical).
- **FR-09:** Qualitative feedback (metrics check, action verb check).

3.2.4 Reporting

- **FR-10:** Interactive Scorecard (Donut charts, graphs).
- **FR-11:** PDF Export of the analysis.

- **FR-12:** Ranked Leaderboard (Recruiter view).

3.2.5 Adversarial Defense & Input Validation (NEW)

- **FR-SEC-01 (Invisible Text Detection):** The parsing module shall detect and flag text rendered in the same color as the background (e.g., white text on white paper) intended to "stuff" keywords.
- **FR-SEC-02 (Zero-Width Character Filtering):** The system shall sanitize input to remove zero-width spaces and homoglyphs often used to bypass blacklists or confuse NLP models.
- **FR-SEC-03 (PDF Structure Analysis):** The system shall validate the internal structure of PDF files to prevent buffer overflow attacks or malicious script execution embedded in document metadata.
- **FR-SEC-04 (Prompt Injection Defense):** If LLMs are used for textual feedback, the system shall treat all resume text as untrusted data, wrapping it in delimiters to prevent the model from interpreting resume content as system instructions.
- **FR-SEC-05 (Metadata Cross-Reference):** The system shall compare the visible text layer against the underlying metadata to detect significant discrepancies (e.g., text layer contains "Senior Developer" but metadata contains "Entry Level").

3.3 Non-Functional Requirements (NFR)

3.3.1 Performance

- **NFR-01:** Parsing/Analysis pipeline < 5 seconds.
- **NFR-02:** Support 50+ concurrent uploads.

3.3.2 Reliability

- **NFR-03:** Parsing accuracy >90%.
- **NFR-04:** 99.9% Uptime.

3.3.3 Security & Robustness (UPDATED)

- **NFR-SEC-01 (Data Encryption):** AES-256 at rest, TLS 1.2+ in transit.
- **NFR-SEC-02 (PII Masking):** "Blind Hiring" mode to mask PII.
- **NFR-SEC-03 (Rate Limiting):** The system shall enforce strict rate limits on analysis requests (e.g., 10 per hour per user) to prevent "Model Inversion" attacks where an attacker aggressively queries the model to reverse-engineer the scoring algorithm.
- **NFR-SEC-04 (Anomaly Detection):** The system shall flag resumes that achieve a 100% or near-perfect match score as "Suspicious" for manual review, as this indicates likely copy-pasting of the JD into the resume (Overfitting).
- **NFR-SEC-05 (Dependency Scanning):** The ML pipeline container images must be scanned for vulnerabilities in Python libraries (e.g., pickle deserialization vulnerabilities).

3.3.4 Usability

- **NFR-05:** Mobile-responsive design.

- **NFR-06:** WCAG 2.1 AA Accessibility compliance.

4. Technical Constraints

- **Backend:** Python (FastAPI) for ML, NestJS for API.
- **Infrastructure:** Docker containers.
- **Sanitization:** All file parsing must occur in an isolated (sandboxed) environment or ephemeral container to prevent server-side compromise from malicious files.

5. Appendices

- **Appendix A:** Threat Model Diagram (To be added)