

Project Documentation: AI Resume Builder & ATS Optimizer

Project by: Satvik Narendra Trivedi

+91 8291647102 / trivedisatvik2@gmail.com

1. Executive Summary

The **AI Resume Forge** is a robust, full-stack application designed to optimize candidate resumes for Applicant Tracking Systems (ATS). By leveraging Google's advanced Gemini AI models, the system performs deep semantic analysis of resumes against specific job descriptions (JDs). It identifies skill gaps, provides a compatibility score, and autonomously rewrites content to maximize hiring potential.

Unlike standard keyword counters, this tool uses Large Language Models (LLMs) to understand context, nuance, and tone. It features a unique "**Failover Architecture**" that ensures 99.9% uptime by automatically switching between AI models if quota limits are reached.

2. Technical Architecture & Innovation

A. The "Auto-Switch" Failover Engine

The core innovation of this platform is its resilience. Relying on public API tiers often leads to "Rate Limit" (429) errors. We implemented a smart routing algorithm to mitigate this:

1. **Primary Route:** The system first attempts to use the user's preferred high-performance model (e.g., **Gemini 2.5 Flash**).
2. **Automatic Detection:** If the API returns a 429 Resource Exhausted error, the system traps the exception instantly.
3. **Seamless Rerouting:** It automatically redirects the payload to the backup model (**Gemini 2.0 Flash**).
4. **Safety Net:** If the backup fails, it defaults to the high-capacity **Gemini 1.5 Flash**, ensuring the user never sees a crash or error screen.

B. "Crash-Proof" HTTP Integration

To avoid stability issues often associated with client SDKs, this project utilizes **Direct HTTP Requests**. This decouples the application from library-specific dependencies, preventing "Illegal Header" errors and ensuring lightweight, fast execution.

C. Input Sanitization & Safety Valve

PDFs often contain hidden metadata that can bloat file size. The system includes a **Safety Valve** that:

- Extracts raw text from complex PDF/DOCX layouts.
- Truncates payload data exceeding 10,000 characters to prevent token overflow.
- Sanitizes API keys to remove invisible whitespace that causes authentication failures.

3. Detailed Step-by-Step Workflow

Step 1: Data Ingestion & Parsing

- **Function:** Accepts resumes in PDF or DOCX formats, or via manual text entry.
- **Process:** The PyPDF2 and python-docx libraries strip formatting to extract clean, machine-readable text.
- **Screenshot:-**

The image consists of two side-by-side screenshots of a software interface. The left screenshot shows the 'Settings' page with a dark background. It features a text input field for 'Enter Gemini API Key' with redacted content, and a dropdown menu for 'Select Model' containing options: 'gemini-2.5-flash', 'gemini-2.0-flash', and 'gemini-exp-1206'. The right screenshot shows the '1. Input Data' page. At the top, there are input options: 'Upload' (selected) and 'Manual'. Below that is a 'Job Description (Required)' field with the text 'cybersecurity'. Underneath is a 'PDF/DOCX' section with a 'Drag and drop file here' area, which contains a file named 'Satvik_resume.pdf' (107.3KB). A green progress bar at the bottom indicates the file has been 'Parsed (3512 chars)'. There are also 'Browse files' and 'x' buttons.

Step 2: Semantic ATS Analysis

- **Function:** Acts as a strict Technical Recruiter.
- **Process:** The AI analyzes the Resume and Job Description side-by-side. It calculates a **Match Percentage** based on semantic relevance, not just keyword matching.
- **Output:**
 - **Score:** 0-100% compatibility rating.
 - **Gap Analysis:** Specifically lists missing hard/soft skills.
 - **Reasoning:** Explains *why* the score was given.

The screenshot shows a dark-themed user interface for resume analysis. At the top, it says "2. ATS Analysis". Below that is a "Check Score" button. Underneath the button, the word "Score" is followed by "75%". A detailed analysis follows:
The resume presents a strong, direct match for an AI/ML Intern position through the 'AI/ML Internship at Quillio'. This experience explicitly mentions developing and implementing machine learning algorithms, collecting and preprocessing large datasets, training models, utilizing deep learning models, and working with Digital Signal Processing (DSP), which aligns very well with the JD. The 'Technical Skills' section further supports this with 'Python' and specifically 'Pytorch', a key deep learning framework. However, the overall resume narrative is heavily skewed towards cybersecurity, with the summary, other internships, projects, and certifications primarily focusing on this domain. This dilutes the AI/ML focus and may give the impression that AI/ML is a secondary interest. While key concepts are present, the breadth of common AI/ML libraries and tools (e.g., TensorFlow, Scikit-learn, NumPy, Pandas) is not fully represented, which might be expected for a comprehensive AI/ML profile.
Missing: TensorFlow, Scikit-learn, NumPy, Pandas, Machine Learning frameworks (general, beyond PyTorch)

Screenshot of a demo resume analysis

Step 3: Generative Enhancement

- **Function:** Rewrites the resume to target the specific job.
- **Process:** The AI uses the gaps identified in Step 2 to rephrase the professional summary, experience bullet points, and skills section. It adopts the terminology and tone of the Job Description.
- **Output:** Structured JSON data containing the optimized resume content.

```
3. Enhancement
Enhance Resume

{
  "name": "Satvik Trivedi",
  "email": "trivedisatvik@gmail.com",
  "phone": "+91 8291647102",
  "summary": [
    "Highly motivated B.Tech student with practical experience in developing and implementing machine learning algorithms and deep learning models for complex simulations and optimization tasks, particularly in Digital Signal Processing. Possessing a foundational understanding of data preprocessing and analysis, eager to apply and expand AI/ML skills in an intern role."
  ],
  "skills": [
    0: "Python",
    1: "Pytorch",
    2: "Machine Learning",
    3: "Deep Learning",
    4: "Data Preprocessing",
    5: "Algorithm Development",
    6: "Digital Signal Processing (DSP)",
    7: "Node.js",
    8: "Angular.js",
    9: "Django",
    10: "React"
  ],
  "experience": [
    {
      "points": [
        0: "Investigated real cybercrime cases involving online financial fraud, demonstrating strong analytical and problem-solving skills."
        1: "Monitored security alerts and logs using SIEM tools to identify potential cyber threats and anomalies, involving large-scale data analysis and pattern recognition."
        2: "Analyzed network traffic, firewall logs, and endpoint alerts to detect suspicious activity, enhancing data interpretation and threat intelligence capabilities."
      ]
    }
  ],
  "education": [
    0: {
      "degree": "Bachelor's in Cyber Security",
      "school": "Shah & Anchor Kutchhi Engineering College, Mumbai University",
      "year": "2026"
    },
    1: {
      "degree": "HSC (PCM)",
      "school": "Mahesh Tutorials, Mumbai University",
      "year": "2022"
    }
  ]
}
```

Step 4: Multi-Format Export Engine

The system offers three distinct export pipelines to suit different industry standards:

1. Direct PDF Generation:

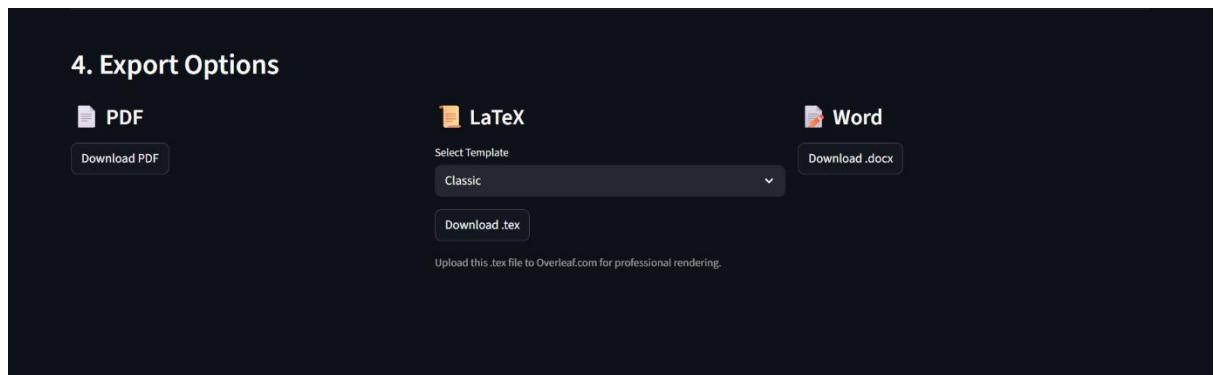
- *Technology*: Uses FPDF to draw a clean, standardized resume layout pixel-by-pixel.
- *Use Case*: Quick applications where a locked format is required.

2. Word Document (.docx):

- *Technology*: Uses python-docx.
- *Use Case*: Users who wish to make further manual edits to the AI-generated content.

3. LaTeX Source Code (.tex):

- *Technology*: String template injection.
- *Templates*: Includes "Classic," "Modern," and "Technical" styles.
- *Use Case*: Academic, Research, or Engineering roles requiring high-typesetting quality (e.g., via Overleaf).



Screenshot of Download options.

