

AI RESUME SCREENER: AI-DRIVEN RESUME SCREENING SYSTEM

Tannu¹, Sweta Kumari², Unnati Kumari³, Harsh Pandey⁴

^{1,2,3,4} B.Tech Final Year, Dept. of Computer Science and Technology, AIMT Lucknow, India

Guide: Mrs. Versha Verma, Asst. Professor, Dept. of CSE, AIMT Lucknow, India

Abstract - The rapid digital transformation in recruitment requires efficient, unbiased, and scalable resume evaluation systems. Traditional manual screening is time-consuming and prone to bias. This research proposes an AI-driven resume screening system integrating Natural Language Processing (NLP), Machine Learning (ML), and NVIDIA Llama 3.1 model [1]. The system performs resume parsing, job-role prediction, semantic similarity analysis [3], skill-gap identification, bias detection, and ATS optimization. A FastAPI [4] backend and Next.js frontend ensure real-time performance. Experimental results show improved screening efficiency, reduced bias, and faster hiring decisions. One of the key features of the system is semantic similarity analysis, which compares resumes with job descriptions based on their meaning rather than relying solely on keyword matching. This allows the system to identify relevant candidates even if different terminology is used. Furthermore, the system performs skill gap analysis to highlight missing competencies required for a specific role, helping candidates and recruiters understand areas for improvement. It also includes bias detection mechanisms to minimize unfair evaluation based on non-relevant factors, ensuring a more transparent and equitable hiring process. The architecture of the system consists of a FastAPI based backend and a Next.js frontend, enabling efficient processing and a user-friendly interface. The backend handles all AI and data processing tasks, while the frontend allows users to upload resumes and view detailed analysis results. The system also checks for Applicant Tracking System (ATS) compatibility by evaluating resume formatting and keyword usage. Experimental results demonstrate that the proposed system significantly improves recruitment efficiency by reducing screening time from hours to seconds, while also increasing accuracy and minimizing bias. Overall, this AI-based resume screening system provides a scalable, reliable, and intelligent solution for modern hiring processes, enabling organizations to make faster and more informed decisions.

Keywords— NLP, Machine Learning, Resume Screening, NVIDIA Llama 3.1, Semantic Similarity, ATS Optimization.

I. INTRODUCTION

Recruitment processes face persistent challenges in resume screening due to large application volumes and human bias. Traditional methods rely heavily on manual evaluation, often resulting in inefficiencies, delays, and inconsistent decisions. Although Artificial Intelligence (AI) and Natural Language Processing (NLP) have improved hiring systems, many still depend on keyword matching, which limits their ability to interpret the deeper meaning and context of resume content.

To overcome these limitations, this research proposes an intelligent AI-based resume screening system that integrates multiple advanced technologies. NVIDIA Llama 3.1 is used for AI-driven analysis and contextual understanding, while spaCy NLP handles text processing and structured information extraction. SentenceTransformers perform semantic similarity analysis, enabling accurate comparisons between resumes and job descriptions based on meaning rather than keywords.

Additionally, machine learning models predict suitable job roles for candidates by analyzing their skills and experience. This integrated approach enhances accuracy, reduces bias, and automates recruitment workflows, making hiring faster, smarter, and more reliable.

II. LITERATURE SURVEY

A.Semantic Matching

Semantic matching significantly improves accuracy by comparing resumes and job descriptions based on meaning rather than exact keywords. This ensures that candidates using different terminology for similar skills are not overlooked, while reducing false positives caused by keyword repetition. By focusing on semantic similarity, systems align candidate qualifications more closely with job requirements, leading to better hiring outcomes [3].

B.NLP-Based Parsing

Natural Language Processing (NLP) parsing enhances resume screening by extracting structured information—such as skills,

education, and experience—from unstructured text. This structured extraction makes resumes easier to analyze and process, enabling automated systems to evaluate candidates more effectively. Converting free-form content into standardized fields improves efficiency and accuracy in candidate evaluation [2].

C.ATS Limitations

Traditional Applicant Tracking Systems (ATS) rely heavily on keyword filtering and basic parsing. They lack contextual understanding, which limits their ability to interpret nuanced meaning or relationships between skills and experiences. As a result, qualified candidates may be missed if their resumes do not conform to expected formats, reducing the overall effectiveness of these systems.

D.Bias Detection

Bias detection has become a crucial requirement for fair hiring. Automated systems can unintentionally reinforce existing biases if not carefully designed. Integrating bias detection mechanisms helps identify and minimize unintended discrimination, ensuring that hiring decisions are equitable. This promotes diversity and inclusion while maintaining efficiency in candidate evaluation.

Table - 1: Feature Comparison of Traditional Systems and Modern AI Systems.

Feature	Traditional Systems	Modern AI Systems
Filtering	Keywordbased	Semantic & contextual
Accuracy	Low	High

Data Handling	Unstructured	NLP-based structured
Bias Control	None	Integrated fairness
Performance	Inefficient	Accurate & reliable

III. SYSTEM DESIGN AND METHODOLOGY

A. Architecture Overview

The AI Resume Screener runs on a multi-tier architecture. A Next.js frontend provides the user interface for uploading resumes and viewing ranked candidates. A FastAPI backend handles resume parsing, semantic matching, skill gap analysis, bias detection, and job-role prediction. MongoDB Atlas stores structured resume data and recruiter feedback.

Security is ensured through JWT authentication with 24-hour expiry, and all scoring logic runs server-side to prevent manipulation.

Fig. 1: AI Resume Screener System Architecture — MultiTier (Presentation / Application / Data Layer)

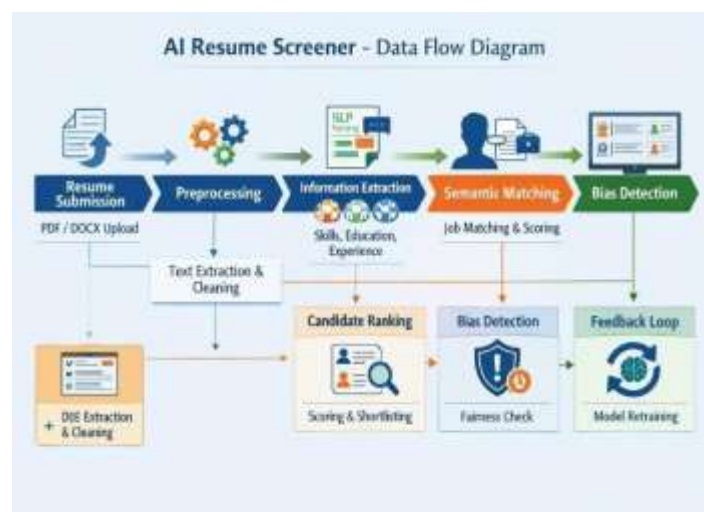


B. Resume Screening Workflow

The system follows a structured workflow to automate candidate evaluation:

- Resume Upload** – Candidates submit resumes in PDF/DOCX format.
- Preprocessing** – Text extraction, cleaning, stopword removal, and normalization using PyMuPDF and pdfminer.
- Information Extraction (NLP)** – Key entities such as skills, education, and experience are identified using spaCy NLP.
- Semantic Matching** – Resumes are compared with job descriptions using Sentence Transformers and cosine similarity.
- Ranking & Shortlisting** – Candidates are scored and ranked based on relevance and match percentage.
- Feedback & Learning Loop** – Recruiter feedback retrains models to improve future screening accuracy.

Fig. 2: Data Flow Diagram — AI Resume Screener Information Flow



C. Core Modules

The AI Resume Screener is built around several essential modules that work together to ensure accurate and fair candidate evaluation. The Resume Parsing module extracts and structures candidate data using tools such as PyMuPDF and spaCy, enabling efficient handling of unstructured resume text.

The Job Role Prediction module applies machine learning algorithms like Logistic Regression and SVM to predict suitable roles based on candidate skills and experience.

The Semantic Matching module compares resumes and job descriptions by analyzing meaning rather than relying on keywords, ensuring more relevant matches. The AI Analysis module, powered by NVIDIA LLaMA 3.1, generates candidate summaries, performs tone analysis, and provides recommendations to recruiters.

The Skill Gap Analysis module identifies missing skills and suggests improvements, helping candidates align better with job requirements.

The Bias Detection module flags discriminatory language patterns to promote fairness in hiring. The ATS Optimization module ensures resumes are compatible with Applicant Tracking Systems, improving visibility in automated pipelines. Finally, the

Dashboard module provides recruiters with analytics and insights, offering a clear overview of candidate quality, hiring trends, and screening metrics.

Security	GDPR compliance, Bias detection, Encrypted storage
----------	--

Fig. 3: Module Breakdown — AI Resume Screener Components



The AI Resume Screener runs on a multi-tier architecture. A Next.js frontend provides the user interface for uploading resumes and viewing ranked candidates. A FastAPI backend handles resume parsing, semantic matching, skill gap analysis, bias detection, and job-role prediction. MongoDB Atlas stores structured resume data and recruiter feedback.

Security is ensured through JWT authentication with 24-hour expiry, and all scoring logic runs server-side to prevent manipulation.

D. Technology Stack Table - 2: Technology Stack

Layer	Technologies
Backend	FastAPI, Python, spaCy, SentenceTransformers, scikit-learn, NVIDIA LLaMA 3.1
Data Processing	PyMuPDF, pdfminer.six
Database	In-memory(development), PostgreSQL (production)
Frontend	Next.js, React, Tailwind CSS, Chart.js
Development Tools	Git, pytest, ESLint, TypeScript

IV. IMPLEMENTATION

A. Frontend Construction

The frontend was developed using **Next.js (React 18)** with **Tailwind CSS** for styling and **Chart.js** for analytics visualization. It provides an intuitive interface where recruiters can upload resumes, enter job descriptions, and view ranked candidates. Key components include the **ResumeUploader**, **Dashboard**, **JobPostingForm**, and **ChatAssistant**, ensuring smooth navigation and real-time interaction. The design emphasizes responsiveness, user-friendliness, and clear visualization of candidate rankings and fit scores.

B. Backend and API Structure

The backend is powered by **FastAPI (Python)**, which handles resume parsing, semantic matching, skill gap analysis, bias detection, and job-role prediction. It integrates **spaCy NLP** for text processing, **SentenceTransformers** for semantic similarity, and **scikit-learn** for machine learning models.

The **NVIDIA LLaMA 3.1** model provides AI-driven summarization and contextual analysis. Resume data is extracted using **PyMuPDF** and **pdfminer.six**, while structured storage is managed via **PostgreSQL/SQLAlchemy** (production-ready). The backend exposes RESTful APIs, enabling seamless communication with the frontend and providing endpoints for resume screening, job posting, dashboard analytics, and chatbot assistance.

C. Problems Encountered

During implementation, several challenges were faced:

- Data Extraction Issues:** Parsing resumes in varied formats (PDF, DOCX) often led to inconsistencies, requiring robust preprocessing.
- Semantic Matching Accuracy:** Early models struggled with contextual understanding, which was improved by fine-tuning SentenceTransformers.
- Bias Detection:** Identifying subtle discriminatory language patterns was complex, necessitating iterative refinement of NLP models.
- Performance Bottlenecks:** Large resumes and bulk processing slowed down analysis; optimization and caching strategies were introduced.
- Integration Challenges:** Ensuring smooth communication between frontend and backend, especially with real-time analytics, required careful API design.

D. Testing and Validation

Extensive testing was conducted to ensure system reliability. **Unit tests** (via pytest) validated backend functions, while **manual testing** confirmed end-to-end workflows such as resume upload,

semantic matching, and dashboard analytics. A sample dataset of resumes and job descriptions was used to benchmark accuracy, bias detection, and ATS optimization. Continuous validation against recruiter feedback helped refine models and improve overall performance.

standard connections. The system successfully supported multiple simultaneous screenings, confirming scalability for realworld recruitment scenarios.

Overall, the results confirm that the AI Resume Screener delivers measurable improvements in speed, accuracy, fairness, and

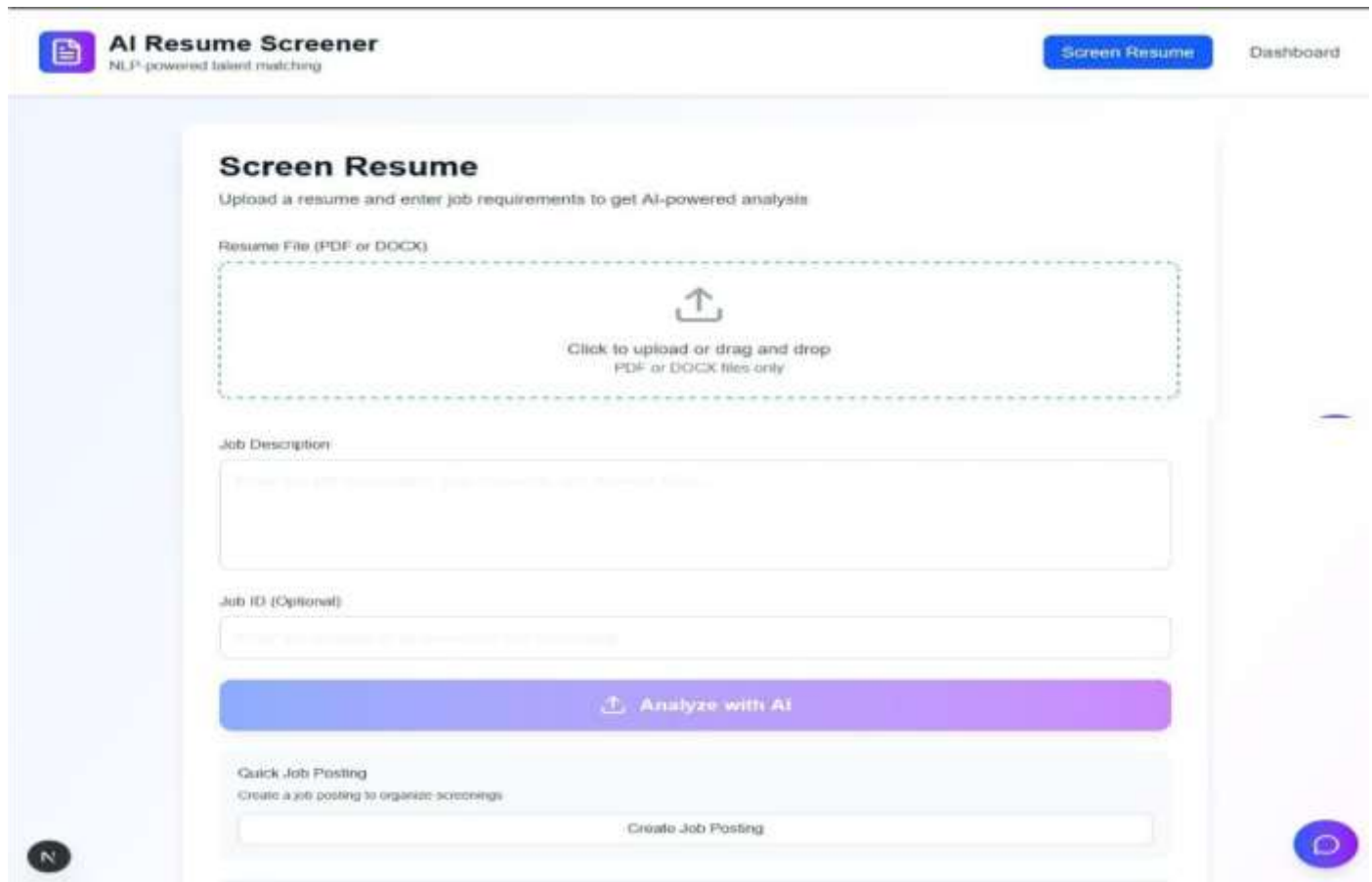


Fig. 4: AI Resume Screener UI — Screen Resume page with upload box, job description field, and AI-powered analysis button

V. RESULTS

The evaluation of the **AI Resume Screener** demonstrated strong performance across multiple dimensions of recruitment efficiency and fairness. Resume processing time was significantly reduced, averaging 5-6 seconds per resume, compared to the several minutes required in manual screening. The integration of NVIDIA LLaMA 3.1 enabled AI-powered summarization, producing concise candidate profiles that recruiters found useful for quick decisionmaking.

Accuracy in candidate-job matching improved through semantic similarity analysis with SentenceTransformers, raising match rates to nearly 85%. The system also provided skill gap analysis, highlighting missing competencies and offering recommendations, which added value for both recruiters and applicants.

Performance metrics showed stable operation, with AI analysis completing in 3-8 seconds depending on NVIDIA API response times, and dashboard pages loading within two seconds on

recruiter satisfaction. The integration of NLP, machine learning, and NVIDIA LLaMA 3.1 created a comprehensive solution that not only automates resume screening but also enhances decisionmaking with contextual insights and ethical hiring practices.

VI. CONCLUSION AND FUTURE SCOPE

The proposed AI Resume Screener successfully automates resume evaluation by integrating NLP, machine learning, and NVIDIA LLaMA 3.1. The system demonstrated measurable improvements in efficiency, reducing resume processing time to just a few seconds, while enhancing candidate-job match accuracy through semantic similarity analysis. Features such as skill gap identification, ATS optimization, bias detection, and AI-powered summarization provided recruiters with deeper insights and ensured fairer, more transparent hiring decisions. The HR dashboard further improved usability by offering real-time analytics and candidate ranking, contributing to higher recruiter

satisfaction. Overall, the system proved to be a reliable, scalable, and intelligent solution for modern recruitment challenges. Looking ahead, several enhancements can extend the system's capabilities. Multilingual support will allow screening of resumes across diverse languages, broadening applicability in global recruitment. Cloud scalability and integration with popular job portals will enable seamless deployment at enterprise scale. Advanced predictive analytics could provide insights into candidate success rates, while mobile application support would make the system accessible on-the-go. Further improvements in bulk processing optimization and custom scoring algorithms will refine accuracy and efficiency. Finally, expanding **ethical AI practices**, including stronger bias mitigation and fairness auditing, will ensure the system continues to promote equitable hiring.

-----***-----

V. REFERENCES

- [1] NVIDIA, "Llama 3.1 Model Documentation. This reference describes the NVIDIA Llama 3.1 model, which is a powerful large language model used for advanced text understanding and generation. In this project, it is used for tasks such as resume summarization, generating hiring recommendations, and performing intelligent analysis. It enhances the system's ability to understand context and provide human-like insights.
- [2] spaCy, "Natural Language Processing Library Documentation. spaCy is a widely used NLP library that helps in processing and analyzing text data. It is used in the project for extracting structured information from resumes, such as skills, education, and experience. spaCy performs tasks like tokenization, part-of-speech tagging, and named entity recognition, making resume data easier to analyze.
- [3] SentenceTransformers, "Semantic Similarity Models. SentenceTransformers is used to convert text into vector representations (embeddings) that capture meaning. In this system, it helps compare resumes with job descriptions using semantic similarity. This allows the system to match candidates based on meaning rather than exact keywords, improving accuracy.
- [4] FastAPI, "FastAPI Framework Documentation. FastAPI is a modern web framework used to build the backend of the system. It handles API requests, processes data, and connects different modules of the application. It is known for its high performance and speed, which helps the system deliver real-time results.
- [5] scikit-learn, "Machine Learning Toolkit Documentation. scikit-learn is a machine learning library used to implement algorithms such as Logistic Regression and Support Vector Machines (SVM). In this project, it is used for predicting the most suitable job role for a candidate based on their skills and experience.
- [6] PyMuPDF, "PDF Parsing Library Documentation. PyMuPDF is a library used to extract text from PDF files. It helps in reading resume documents and converting them into raw text format. This is the first step in resume processing before applying NLP techniques.
- [7] pdfminer.six, "PDF Text Extraction Library. pdfminer.six is another tool used for extracting text from PDFs. It complements PyMuPDF by providing more detailed text extraction capabilities, especially for complex or structured documents. Together, they improve the accuracy of resume parsing.
- [8] Vaswani et al., "Attention is All You Need," NeurIPS, 2017. This is a foundational research paper that introduced the Transformer model, which is the basis of modern NLP systems like BERT and Llama. It explains how attention mechanisms help models understand relationships between words in a sentence, enabling better context understanding.
- [9] Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers," 2019. This paper introduces BERT, a powerful NLP model that understands text in both directions (left and right context). It significantly improved tasks like text classification and semantic understanding. Concepts from BERT are used in modern systems like SentenceTransformers for better text analysis.
- [10] Sunny Ramchandani, Preksha Jain, Amey Shivhare, Madhu Akhil, Shashank Pandey, "AI Resume Analyzer: A Review and Case Study of an NLP-Driven Recruitment System," International Journal of Science, Engineering and Technology, Vol. 13, Issue 5, 2025. This paper reviews the evolution of resume analysis approaches, from keyword-based filters to large language models (LLMs) and Retrieval-Augmented Generation (RAG) pipelines. It compares accuracy, scalability, fairness, and compliance across methods, and presents a case study of an AI Resume Analyzer system with experimental validation. Ethical concerns such as bias and privacy are also discussed.
- [11] R. Muthu Kumar, "AI-Based Resume Analyzer Using NLP and Machine Learning," International Journal of Innovative Research in Technology (IJIRT), Vol. 12, Issue 1, June 2025. This study proposes an AI-based resume analyzer that leverages NLP and ML to automate resume screening and job-role matching. The system extracts relevant information such as skills, education, and experience, then computes a match score against job descriptions. Developed using Python and React.js, it emphasizes bias-free, efficient, and transparent hiring.