

AI-Powered Resume Shortlisting System

Goal

To build a web-based system that automatically reads, analyzes, and ranks resumes according to a company's specific job requirements.

The system extracts details from resumes, cover letters, and LinkedIn profiles, compares them against a job description, and produces a ranked shortlist of the best-fit candidates.

It helps recruiters save time, increase accuracy, and make data-driven hiring decisions.

Real-World Use Case

Recruiters and HR teams often receive hundreds of resumes for a single job opening. Manually reading and shortlisting them is time-consuming, inconsistent, and biased.

This system solves that by automatically:

- Reading resumes, cover letters, or LinkedIn profiles
- Matching key skills, experience, and education to the job requirement
- Scoring each candidate
- Displaying the results on a recruiter dashboard

Example

Job Role: Frontend Developer

Requirements: React.js, JavaScript, HTML, CSS, 2+ years of experience

Candidate	Match Score	Result
Aditi Sharma	70%	 Shortlisted
Rahul Verma	60%	 Consider
Neha Patel	50%	 Rejected

Core Features

- 1. Job Requirement Setup: Recruiters can create job listings by entering job title, department, required & preferred skills, experience, education level, keywords, and description.
- 2. Resume Upload & Extraction: Upload resumes (PDF/DOCX) and optional LinkedIn links. Extracts name, skills, experience, education, achievements, and projects.
- 3. Skill Matching & Candidate Scoring: Uses weighted scoring

1. Resume Format & Presentation – 15 Points

Sub-category	Points
Clean layout & structure	5
Proper headings (Education, Experience, Skills, etc.)	3
Consistent fonts, spacing, alignment	3
Professional formatting (no clutter, balanced white space)	4
<input checked="" type="checkbox"/> Total: 15/100	

2. Contact & Basic Information – 5 Points

Sub-category	Points
Full name, phone, email	3
LinkedIn / Portfolio / GitHub link	2
<input checked="" type="checkbox"/> Total: 5/100	

3. Resume Summary / Objective – 10 Points

Sub-category	Points
Clear objective or professional summary	5
Relevant to job role	3
Short & meaningful	2
<input checked="" type="checkbox"/> Total: 10/100	

4. Skills Section – 15 Points

Sub-category	Points
Relevant technical skills	7
Relevant soft skills	3
Categorized (Tech / Tools / Soft skills)	3

Sub-category	Points
No irrelevant or spam skills	2
<input checked="" type="checkbox"/> Total: 15/100	

5. Experience Section – 20 Points

Sub-category	Points
Job titles + company names	5
Proper timeline (dates must be correct)	3
Clear descriptions of responsibilities	5
Achievements using numbers (e.g., 30% improvement)	5
Relevance to job	2
<input checked="" type="checkbox"/> Total: 20/100	

6. Projects Section – 15 Points

Sub-category	Points
At least 2 technical or relevant projects	5
Project description is clear	3
Responsibilities + tools used mentioned	5
Live link / GitHub link provided	2
<input checked="" type="checkbox"/> Total: 15/100	

7. Education Section – 10 Points

Sub-category	Points
Degree + college name	4
Passing year	2

Sub-category	Points
Score/CGPA	1
Relevant coursework (if any)	3
<input checked="" type="checkbox"/> Total: 10/100	

8. Certifications / Trainings – 5 Points

Sub-category	Points
Relevant certifications	3
Verified certificate links (optional)	2
<input checked="" type="checkbox"/> Total: 5/100	

9. Additional Sections – 3 Points

Sub-category	Points
Awards / Achievements	1
Volunteering / Extra activities	1
Languages or Interests section	1
<input checked="" type="checkbox"/> Total: 3/100	

10. Grammar, Spelling & Professional Tone – 2 Points

Sub-category	Points
No spelling mistakes	1
Professional writing tone	1
<input checked="" type="checkbox"/> Total: 2/100	

TOTAL RESUME SCORE = 100 POINTS

- 4. Dashboard & Candidate View: Displays candidate name, score, skills, experience, and includes filters for score range, education, and years of experience.
- 5. Download & Export: Download shortlisted candidates as Excel(CSV) or PDF, share or copy details, and export to HR tools.

Goals & Success Criteria

Metric	Target
Resume Parsing Accuracy	≥ 85%
Match Scoring Accuracy	≥ 80%
Processing Speed	< 10 sec per resume
Export Functionality	Fully working CSV/PDF downloads

Python Tech Stack (Without MERN Stack)

Frontend: HTML, CSS, JavaScript (Vanilla or with Bootstrap / Tailwind CSS)

Backend: Python (Flask or FastAPI) Also with JAVs

Database: SQLite or PostgreSQL

Resume Parsing: Python Libraries (spaCy, PyPDF2, pdfminer, NLTK)

Storage: Local storage or Firebase Storage (optional)

Authentication (Optional): JWT or Firebase Auth

Deployment: Render, Railway, or PythonAnywhere

Java Tech Stack

Frontend

- HTML, CSS, JavaScript (Vanilla / Bootstrap / Tailwind) — unchanged

Backend (Java)

- **Spring Boot** (recommended) or **Quarkus / Micronaut**
- **Spring Web (REST), Spring Validation, Spring Security (JWT)**

Database

- **PostgreSQL or MySQL (prod), H2**

Resume Parsing (Java libs)

- PDFs: **Apache PDFBox, Apache Tika** (MIME detect + text extract)
- DOC/DOCX: **Apache POI**
- NLP: **Apache OpenNLP, Stanford CoreNLP** (advanced), **LingPipe** (alt)
- Keywords/Scoring helpers: simple TF-IDF (Lucene), RAKE implementations (Java), or custom regex

Auth (Optional)

- **JWT** with Spring Security & **jjwt** or **java-jwt**

System Flow

1. 1. Recruiter creates a new job posting
- 1.1 That has sections of uploading JD and Resume (CV'S)
2. Resumes uploaded in multiples files to process one by one.
3. Backend extracts text and identifies skills.
4. Scoring algorithm compares data with job requirements.
5. Dashboard displays ranked candidates.
6. Recruiter downloads or exports shortlisted results.

Download & Export Options

Users can export:

-  Top 10 candidates (PDF) – formatted for HR review
-  All shortlisted candidates (CSV) – includes detailed scoring and contact info

Final Output Example (Demo)

HR uploads: 20 resumes (PDFs) ONLY at a time

Job Role: Frontend Developer and etc.

Required Skills: React, JavaScript, CSS, HTML