PROJECT ABSTRACT

Resume ATS Compatibility Checker

A web tool that scores a resume against a job description the way applicant tracking systems parse them — keyword coverage, section checks and formatting risks — built-to-order with a working demo, report, PPT and viva kit.

01. Title

Resume ATS Compatibility Checker

02. Introduction / Background

Most large recruiters filter applications through Applicant Tracking Systems before any human reads them. These systems parse resume text, extract skills and sections, and rank candidates against the job description's keyword set — and an estimated majority of resumes are rejected at this machine stage, often for fixable reasons: missing keywords, unparseable formatting, or absent sections. Students typically discover this only after silent rejections. An ATS compatibility checker demystifies the process: paste the resume and the job description, and the tool reports a compatibility score with the exact matched/missing keywords, section coverage and formatting risks — the same dimensions a parser evaluates.

03. Problem Statement

Students write resumes for humans and submit them to machines. Without feedback, they cannot know which keywords the posting expects, whether their layout survives parsing, or which sections the system looks for — so strong candidates get filtered out by formatting. This project builds a transparent, rule-based ATS simulator: deterministic keyword extraction from the job description, coverage scoring against the resume, section-presence checks (contact, education, experience/projects, skills), and formatting-risk flags. It teaches how parsing works while giving immediately actionable fixes, with no black-box claims.

04. Objectives

- Extract a skill-keyword dictionary from the pasted job description deterministically.
- Score keyword coverage of the resume against the posting's keyword set.
- Check section presence: contact block, education, experience/projects, skills.
- Flag formatting risks that break parsers (tables, graphics, non-standard headings).
- Produce prioritized, posting-specific improvement tips.
- Deliver the complete student kit: working web app, source code, report, PPT and viva Q&A.;

05. Proposed Methodology

The checker is a client-side rule-based analyzer — deliberately transparent, not a black-box model. (1) Tokenization: both texts are normalized (lowercased, punctuation stripped) and tokenized. (2) Keyword extraction: the job description is scanned against a curated 21-skill dictionary covering languages, frameworks, tools and practices. (3) Coverage: each posting keyword is classified hit/miss against the resume tokens. (4) Sections: regex/structure checks detect contact patterns, education markers, experience/project mentions and a skills block. (5) Scoring: weighted composite — skills 55%, sections 25%, format 20% — rendered as an animated score ring with a verdict band.

06. System Architecture / Working

A single-page app with three views. The input view holds two text panes (resume, job description) pre-seeded with a realistic fresher resume and posting. Analysis runs as a staged pipeline with progress feedback: parse, extract, format-check, score. The results view shows the compatibility score ring, hit/miss keyword chips, formatting checks and section coverage table. The tips view lists prioritized fixes derived from the actual misses. All computation is local in the browser — no data leaves the machine, which is itself a talking point for the viva.

07. Hardware / Software Requirements

- Any modern web browser (Chrome, Edge, Firefox)
- Single HTML/CSS/JS file — no server, no build step, runs offline
- No external libraries or network calls in the demo
- Report/PPT tooling: any office suite; report delivered as PDF

08. Expected Outcomes

- A working analyzer that scores any pasted resume/JD pair in seconds.
- Deterministic, explainable scoring — every point traceable to a keyword or check.
- Seeded demo data showing a realistic moderate-match analysis end-to-end.
- Prioritized improvement tips generated from the actual keyword gaps.
- Project report, presentation and viva Q&A; matching the implemented features.

09. Applications

- Placement cells running resume-screening workshops
- Students tailoring resumes per job posting
- Career-guidance counsellors demonstrating how ATS parsing works
- Classroom demo of tokenization and rule-based text analysis

10. Future Scope

- PDF/DOCX upload with real text extraction
- Larger, domain-specific keyword dictionaries (data, embedded, design)
- Synonym handling (e.g. JS = JavaScript) via a curated map
- Side-by-side before/after scoring as the user edits

- Exportable one-page ATS report PDF

11. Declaration

This abstract describes the built-to-order deliverable. The score is a deterministic rule-based estimate for learning purposes — it is not a claim about any commercial ATS product, and no accuracy or test-result figures are stated.