AI Resume Analyzer - Complete Project Report & Viva Guide

Section 1: Detailed Project Report

Problem Statement

Students often struggle to create industry-ready resumes that meet the standards of recruiters and Applicant Tracking Systems (ATS). Many resumes are rejected automatically because they lack the right keywords, proper formatting, or relevant information. This reduces the chances of students being shortlisted for placements or interviews.

Proposed Solution

The AI Resume Analyzer is a smart tool that helps students and job seekers improve their resumes. It automatically parses uploaded resumes, analyzes the content using Natural Language Processing (NLP), scores them based on multiple parameters, and provides improvement suggestions. This ensures resumes are ATS-friendly and aligned with industry standards.

Workflow of the System

1. Resume Upload – Users upload their resumes in PDF or DOC format.
2. Text Extraction – Resume content is extracted using parsing libraries.
3. NLP Processing – Extracted text is processed to identify skills, education, projects, and experience.
4. Analysis – AI/ML models compare extracted content with industry/job requirements.
5. Scoring – Resume is scored based on formatting, keyword match, and relevance.
6. Suggestions – The system provides recommendations for improvement.

Technologies Used

- Programming Language: Python
- Backend Frameworks: Flask / Django
- Libraries: NLTK, SpaCy, PyPDF2, pdfminer, docx2txt, scikit-learn
- Database: MySQL / MongoDB
- Frontend: HTML, CSS, JavaScript
- Deployment: AWS / Heroku

Features

- Resume parsing from PDF/DOC
- Keyword matching against job descriptions
- Resume scoring based on multiple factors
- AI-driven improvement suggestions
- Export of updated resume

Advantages

- Helps students create ATS-friendly resumes
- Saves recruiters’ time in shortlisting
- Increases chances of selection
- Provides structured feedback for improvement

Limitations

- Accuracy depends on dataset and model training
- Cannot fully replace human evaluation
- May not handle resumes with complex layouts, images, or tables

Future Scope

- AI-powered auto-resume builder
- Advanced deep learning-based analysis
- Integration with LinkedIn and placement portals
- Real-time resume vs. job description matching
- Ensuring data privacy and security of resumes

Section 2: Viva Questions with Answers

Q: What problem does your project solve?

A: It solves the problem of non-ATS-friendly resumes. Our system analyzes resumes, scores them, and suggests improvements, helping students create industry-ready resumes.

Q: What is ATS and why is it important?

A: ATS stands for Applicant Tracking System. It is used by companies to automatically filter resumes. If a resume is not ATS-friendly, it may be rejected before reaching HR.

Q: Which libraries did you use for text extraction?

A: We used PyPDF2, pdfminer, and docx2txt to extract text from PDF and Word resumes.

Q: What is NLP and how do you use it?

A: NLP means Natural Language Processing. We use it to process resume text, extract key sections like skills and education, and compare them with job requirements.

Q: How do you score resumes?

A: We assign scores based on keyword matching, formatting quality, grammar, and relevance to the job description.

Q: What are stop words and tokenization?

A: Stop words are common words like 'is', 'the', 'and' that do not add value. Tokenization is splitting text into smaller parts (words/sentences) for analysis.

Q: How do you match resume with job description?

A: We extract keywords from both the resume and the job description using NLP and then compare them. The percentage of match determines the score.

Q: What are the limitations of your system?

A: The system depends on dataset accuracy, struggles with resumes having images/tables, and cannot fully replace human recruiters.

Q: How can your system be improved in the future?

A: We can improve it by using deep learning for better accuracy, integrating LinkedIn APIs, enabling auto-resume generation, and ensuring strong data privacy.

Q: Can it be integrated with placement portals?

A: Yes, our system can be integrated with college placement portals so students get automatic feedback before applying.

Section 3: Beginner-Friendly Explanation of Terms & Concepts

Introduction in Simple Words

A resume is a document that shows your skills, education, projects, and experience. Companies use resumes to decide whether to call you for an interview. But most big companies use ATS (Applicant Tracking Systems) before HR reads them. If your resume does not have the right keywords, ATS will reject it. Our AI Resume Analyzer checks resumes, gives a score, and suggests improvements so resumes become ATS-friendly.

Meaning of Key Terms

1. ATS (Applicant Tracking System): Software that filters resumes automatically.
2. Parsing: Extracting text from documents like PDF/Word.
3. NLP (Natural Language Processing): AI that helps computers understand human language.
4. Tokenization: Breaking text into smaller parts like words or sentences.
5. Stop Words: Common words like 'is', 'the', 'and'.
6. Keyword Matching: Comparing words in the resume with job description.
7. Machine Learning: AI technique where computers learn from data.
8. Scoring: Giving a numerical value (like 80/100) to show resume strength.

Project in Simple Steps

1. Upload Resume
2. Extract Text
3. NLP Analysis
4. Compare with Job Description
5. Score
6. Suggestions

Why It Is Useful

- For Students: Helps improve resumes for placements.
- For Recruiters: Saves time by filtering resumes.
- For Colleges: Increases placement success rate.

Example

Job requires: Python, SQL, Machine Learning
Resume has: Python, Java
→ Score: 50/100 with suggestions to add SQL and Machine Learning.

Analogy

Think of ATS like a metal detector at the airport: only people without metal pass. Similarly, only resumes with correct keywords pass ATS. Our analyzer is like a coach that trains you to pass the ATS check.

Section 4: Technologies & Languages Used

Python (Programming Language)

The main language of the project. Easy to learn and widely used in AI/ML. Used for NLP, parsing resumes, and backend logic.

Flask / Django (Backend Frameworks)

Frameworks to build the web app and connect frontend with backend. Example: Flask handles resume upload requests.

NLP Libraries

NLTK and spaCy help process text, tokenize, remove stop words, and extract keywords.

Parsing Libraries

PyPDF2, pdfminer, docx2txt extract text from PDF/Word resumes.

Machine Learning Libraries

scikit-learn for similarity matching and scoring. Pandas & NumPy for data handling.

Database (MySQL / MongoDB)

MySQL for structured data, MongoDB for flexible NoSQL data storage. Stores resumes and scores.

Frontend

HTML, CSS, JavaScript build the web interface for uploading resumes and displaying results.

Deployment Platforms

AWS / Heroku used to host the project online, making it accessible anywhere.