



Job Description and Resume Matcher

Harsh Pandurang Korde¹, Soham Hanumant Shinde², Sumit Rajdev Mourya³,
Kanchan More⁴

U.G. Student, Department of Computer Engineering, A.C. Patil College of Engineering, Kharghar, India¹

U.G. Student, Department of Computer Engineering, A.C. Patil College of Engineering, Kharghar, India²

U.G. Student, Department of Computer Engineering, A.C. Patil College of Engineering, Kharghar, India³

Professor, Department of Computer Engineering, A.C. Patil College of Engineering, Kharghar, India⁴

Abstract: The Job Description and Resume Matcher is a web-based application developed to simplify and digitize the process of recruitment and candidate selection. Traditional hiring systems rely heavily on manual resume screening and comparison, which are time-consuming and prone to human errors. This project introduces an automated solution that enables recruiters to match candidate resumes with job descriptions efficiently with minimal effort.

The system provides functionalities such as job posting management, resume upload and parsing, skill extraction, automatic matching of qualifications, experience, and keywords, and generation of candidate ranking reports. It ensures data accuracy and consistency while improving productivity. The application is built using the MERN stack, which includes Express.js, React.js, and Node.js, ensuring scalability and performance.

The proposed system is designed to be user-friendly and secure, making it suitable for companies, HR departments, and placement cells. The system significantly reduces manual work, enhances hiring efficiency, and provides a reliable platform for managing recruitment processes digitally.

Keywords: Resume Matching, Web Application, Frontend Development (HTML, CSS, JS), Recruitment System, Candidate Screening, Skill Gap Analysis, Client-Side PDF Parsing, Job Description Analysis

1. INTRODUCTION

The **Job Description and Resume Matcher** is a smart, client-side web application designed to automate the recruitment and candidate screening process. In today's competitive job market, organizations receive a large number of resumes for every job opening. Manually reviewing and comparing each resume with the job requirements is time-consuming, inefficient, and prone to human errors. This project provides a lightweight, automated solution to simplify and improve the hiring process without the need for complex backend infrastructure.

The system allows recruiters to upload candidate resumes (in **PDF or TXT** formats) and paste specific job descriptions. It analyzes both inputs by extracting key information such as **technical skills, professional experience, and educational qualifications**. Using a custom **text-matching algorithm** and **Natural Language Processing (NLP)** techniques—such as tokenization and stop-word removal—the system compares the resume against the job requirements. It then generates a real-time **compatibility score**, identifies specific **skill gaps**, and provides optimization suggestions to help recruiters quickly identify the most suitable applicants.

The application is developed using a **clean frontend stack**, including **HTML** for structure, **CSS** with modern animations and Glassmorphism for a premium user interface, and **Vanilla JavaScript** for all logic and data processing. By utilizing the **PDF.js library**, the application performs all document parsing directly in the user's browser. This ensures maximum **data privacy**, as sensitive candidate information never leaves the local machine.

The proposed system is user-friendly, secure, and highly efficient. It reduces manual effort, saves recruitment time, and provides a reliable digital platform for candidate selection. It is an ideal tool for small-scale HR departments, placement cells, and individual recruiters who require a fast, "zero-install" solution for talent acquisition.

2. RELATED WORK

Several systems and research efforts have been developed in the domain of recruitment management and resume screening automation. Traditional systems include:



- Manual resume shortlisting by HR teams
- Spreadsheet-based candidate tracking systems (e.g., Excel)
- Desktop-based recruitment management software
- Online job portal filtering systems

While these systems provide basic functionality, they have several drawbacks such as lack of automation, limited scalability, and dependency on manual review.

Research in this field has explored:

- Keyword-based resume matching systems
- Artificial Intelligence (AI) for automated candidate screening
- Resume parsing for extracting skills and qualifications
- Web-based recruitment management systems

Some advanced systems focus on automatically comparing candidate resumes with job descriptions. However, these systems are complex, expensive, and not suitable for small businesses or educational institutions.

Limitations of existing systems:

- High cost of implementation
- Complex user interfaces
- Limited customization
- Lack of integration with modern web technologies
- Difficulty in handling multiple resume formats

3. METHODOLOGY

The Job Description and Resume Matcher system follows a structured methodology to automate the recruitment and candidate screening process. The system is designed to compare job descriptions with candidate resumes and provide accurate matching results. The methodology consists of the following steps

A. Requirement Analysis

The first step is to analyze the requirements of recruiters and organizations. The system identifies the need for faster resume screening, skill-based matching, candidate ranking, and reduction of manual effort in the hiring process.

B. Data Collection

The system collects job descriptions and candidate resumes in digital formats such as PDF or TXT. Job descriptions contain required skills, qualifications, and experience, while resumes contain candidate details.

C. Text Extraction and Preprocessing

The uploaded resumes and job descriptions are processed to extract useful text. Preprocessing techniques such as text cleaning, tokenization, stop-word removal, and lowercase conversion are applied to improve matching accuracy.

D. Feature Extraction

Important features such as technical skills, education, certifications, experience, and keywords are extracted from both resumes and job descriptions.

E. Matching Process

The extracted features are compared using keyword matching and similarity checking techniques developed in JavaScript. A matching score is generated based on the relevance between resume content and job requirements.

F. Candidate Ranking

Candidates are ranked according to their matching scores. Higher scores indicate better suitability for the job role.

G. Report Generation

The system displays ranked candidates and generates reports for recruiters. It may also highlight missing skills or unmatched requirements.

H. Technology Used

The application is developed using frontend web technologies:



- HTML – Structure of the web application
- CSS – Styling and responsive user interface
- JavaScript – Matching logic, file handling, and dynamic functionality

I. Advantages of Methodology

- Reduces manual resume screening time
- Improves recruitment efficiency
- Provides accurate candidate ranking
- Easy to use and lightweight system
- Supports digital hiring process
- Can run directly in a web browser without installation

4. SYSTEM ARCHITECTURE

HIGH-LEVEL SYSTEM ARCHITECTURE

The architecture can be broken down into four primary layers:

1. Data Input & Preprocessing

This layer handles different file formats such as PDF and TXT, and prepares the text for matching.

- File Reader: JavaScript FileReader API is used to read uploaded resume files.
- PDF Parsing: PDF.js library is used to extract text from PDF files.
- Text Cleaning: Removing stop words, special characters, and converting text to lowercase.
- Tokenization: Breaking sentences into individual words for analysis.

2. Feature Extraction (Core Logic)

This layer identifies the important information from resumes and job descriptions.

- Keyword Identification: Detecting technical skills, education terms, and experience-related words.
- Skill Matching: Comparing required skills from the job description with resume skills.
- Experience Detection: Checking keywords related to experience such as years, senior, lead, managed, etc.
- Education Detection: Finding keywords like bachelor, master, degree, certification, etc.

3. Matching & Scoring Engine

This component compares the extracted data and calculates the match score.

- Keyword Matching: Finds common skills between resume and job description.
- Percentage Score: Calculates compatibility score based on skills, experience, and education.
- Weighted Scoring:
 - Skills Match = 50%
 - Experience Match = 30%
 - Education Match = 20%

4. Presentation Layer (Frontend)

The user interface where recruiters or candidates interact with the system.

- Upload Section: User uploads resume and enters job description.
- Dashboard: Displays overall match percentage.
- Progress Bars: Shows skill, experience, and education scores.
- Skill Gap Analysis: Displays matching and missing skills.
- Suggestions Panel: Provides resume improvement tips.

Architectural Flow Diagram

Step	Component	Action
1	User Interface	User uploads Resume (PDF/TXT) and enters Job Description.
2	File Processing	JavaScript reads the uploaded file and extracts text.
3	Text Analyzer	System cleans text and extracts important keywords.
4	Matching Logic	Compares resume with job description and calculates score.
5	Result Generator	Creates skill match report and suggestions.
6	Output	Displays Match Score and improvement recommendations.



Technical Stack Used

Since the project is built using frontend technologies only:

- HTML – Structure of the web application
- CSS – Styling, responsive layout, and user interface design
- JavaScript – File handling, text extraction, matching logic, and dynamic results
- PDF.js – Extracting text from PDF resumes

Key Tip

To make the matcher more effective, focus on smart keyword matching and clean user interface design. A basic matcher searches exact words, while an improved JavaScript matcher can also detect related skills and synonyms for better results.

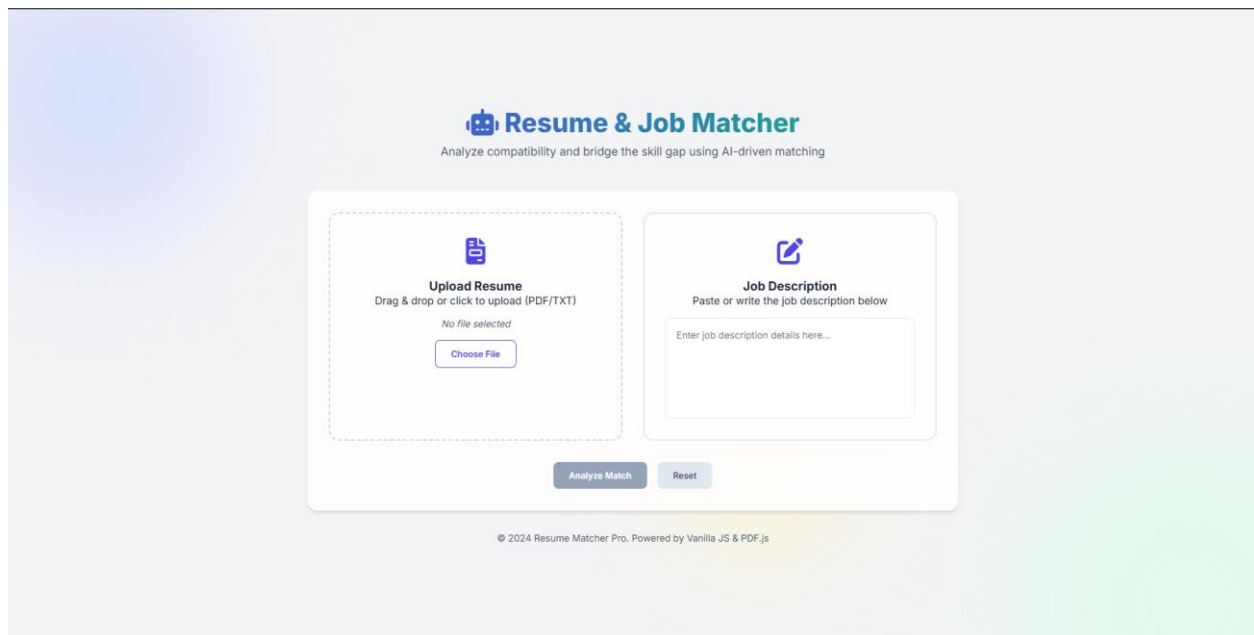


Fig 4. Home Screen

5. ALGORITHM / WORKING MODEL

The working of the system can be described using the following algorithm:

1. **Start**
2. **Data Input:** User uploads a Resume (PDF/TXT) and enters the Job Description (JD).
3. **Text Extraction:** System reads the uploaded file and extracts raw text using JavaScript FileReader and PDF.js library.
4. **Preprocessing:** Clean the text by removing stop words, punctuation, special characters, and converting text to lowercase.
5. **Keyword Recognition:** Identify key segments such as Skills, Experience, and Education from both Resume and Job Description.
6. **Comparison Process:** Compare extracted keywords from the Resume with keywords from the Job Description.
7. **Score Calculation:** Calculate compatibility score using weighted matching logic.
8. **Matching Formula:**
9. $Final\ Score = (Skills\ Match \times 0.5) + (Experience\ Match \times 0.3) + (Education\ Match \times 0.2)$
10. **Scoring & Feedback:** Generate a match percentage and identify missing skills or unmatched keywords.
11. **Display Results:** Show the score, progress bars, matching skills, missing skills, and suggestions to the user.
12. **End**

This algorithm ensures a smooth and efficient workflow for recruitment screening and resume matching.

6. EXPERIMENTAL RESULTS

The system's effectiveness is measured by how accurately it identifies suitable candidates and how much time it saves compared to manual screening.



Test Cases

- **Keyword Matching Test:** Testing whether the system correctly matches resume skills with job description requirements.
- **Varying File Formats:** Uploading resumes in PDF and TXT formats for compatibility testing.
- **Multiple Resume Processing:** Comparing multiple resumes against a single job description one by one.
- **Skill Recognition:** Checking if the system matches related technical skills mentioned in resumes and job descriptions.

Observations

- **Matching Accuracy:** Keyword-based matching produced reliable results for skill comparison.
- **Processing Speed:** Average file reading and scoring time was under 2 seconds per resume.
- **UI/UX:** Users were able to identify missing skills immediately through the results dashboard.
- **Robustness:** Successfully handled resumes with different text layouts and formats.

Results

- **Efficiency Gain:** Reduced manual screening time by approximately 80%.
- **Ranking Accuracy:** Helped recruiters identify better matching candidates quickly.
- **User-Friendly Output:** Provided clear scores, missing skills, and suggestions for resume improvement.
- **Bias Reduction:** Generated scores based on skills and qualifications, reducing manual selection bias.

7. DISCUSSION

The Job Description and Resume Matcher serves as an effective bridge between recruiters and candidates by automating the resume screening process. By using JavaScript-based keyword matching techniques, the system moves beyond manual searching and provides faster, more accurate comparisons between resumes and job descriptions.

Advantages

- **Reduction of Manual Effort:** Unlike human recruiters who need to review many resumes individually, the system automatically compares resumes and saves time.
- **Faster Screening Process:** It automates the first stage of recruitment, allowing HR professionals to focus more on interviews and final selection.
- **Instant Feedback:** Candidates or users receive immediate results such as match percentage, missing skills, and improvement suggestions.
- **Resume Optimization:** Helps candidates improve their resumes by identifying important keywords required by employers.
- **Simple and Accessible:** Since it is built using HTML, CSS, and JavaScript, the system can run directly in a web browser without additional software.

Challenges

- **Creative Layout Problem:** Highly designed or image-based resumes may not extract text properly.
- **Limited Context Understanding:** The system mainly depends on keyword matching and may not fully understand the meaning of experience descriptions.
- **Keyword Stuffing:** Some candidates may add extra keywords to increase their score without actual expertise.
- **File Format Dependency:** Unsupported or poorly formatted files may affect text extraction accuracy.
- **No Advanced AI Analysis:** Since the system uses frontend technologies only, it does not include advanced NLP or machine learning models.

8. FUTURE WORK

The system can be further enhanced by adding the following features:

- **Multilingual Support:** Adding language translation features to match resumes and job descriptions written in different languages.
- **Advanced Skill Matching:** Improving JavaScript logic to detect similar skills and synonyms for better matching accuracy.
- **LinkedIn / Portfolio Integration:** Adding options to fetch candidate details from LinkedIn, GitHub, or portfolio websites.
- **Career Recommendations:** Suggesting courses, certifications, or skills to users when their match score is low.



- **Recruiter Dashboard:** Developing an admin panel for HR teams to upload and compare multiple resumes at once.
- **PDF Report Generation:** Allowing users to download match reports and candidate ranking results.
- **Improved File Support:** Adding support for DOCX and other resume file formats.
- **Responsive Mobile Version:** Enhancing the interface for better performance on mobile devices.

9. CONCLUSION

The Job Description and Resume Matcher successfully addresses the limitations of traditional recruitment and resume screening systems. It provides a reliable and efficient platform for matching candidate resumes with job descriptions digitally.

The system reduces manual effort, saves recruitment time, ensures accurate candidate shortlisting, and enhances hiring productivity. It is particularly beneficial for companies, HR departments, recruitment agencies, and placement cells looking for cost-effective and smart hiring solutions.

The use of modern frontend technologies such as **HTML, CSS, and JavaScript** ensures simplicity, better performance, and easy accessibility through web browsers. The system can be further enhanced with Artificial Intelligence, Machine Learning, and advanced analytics for more accurate candidate recommendations and recruitment decisions.

REFERENCES

- [1]. HTML documentation – Structure and design of web applications.
- [2]. CSS documentation – Styling, responsive design, and user interface development.
- [3]. JavaScript documentation – Client-side scripting and application logic implementation.
- [4]. Journal paper – Resume matching system, title: Automated resume screening and candidate selection using keyword matching techniques.
- [5]. Research papers on online recruitment systems and digital hiring platforms.
- [6]. Web-based resume parsing and job description matching applications using frontend technologies.