AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING

A PROJECT REPORT

Major Project (01CE0716)

Submitted by

ADITYA BHALSOD

92200103013

RAHEELKHAN LOHANI

92200103021

KHUSH AGHERA

92200103034

BACHELOR OF TECHNOLOGY

in

Computer Engineering

Faculty of Engineering & Technology

Marwadi University, Rajkot

August, 2025

Marwadi University
Marwadi Chandarana Group

NAAC A+

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Automated Resume Screening System Using Natural Language Processing And Machine Learning** has been carried out by **Aditya Bhalsod** (92200103013), **Raheelkhan Lohani** (92200103021), **Khush Aghera** (92200103034) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Pratikkumar Chauhan
Assistant Professor
Department of Computer Engineering

Prof. Dr. Krunal Vaghela
Professor & Head
Department of Computer Engineering

Marwadi University
Marwadi Chandarana Group

NAAC A+

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Automated Resume Screening System Using Natural Language Processing And Machine Learning** has been carried out by **Aditya Bhalsod** (92200103013) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Pratikkumar Chauhan
Assistant Professor
Department of Computer Engineering

Prof. Dr. Krunal Vaghela
Professor & Head
Department of Computer Engineering

Marwadi University
Marwadi Chandarana Group

NAAC A+

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Automated Resume Screening System Using Natural Language Processing And Machine Learning** has been carried out by **Raheelkhan Lohani** (92200103021) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Pratikkumar Chauhan
Assistant Professor
Department of Computer Engineering

Prof. Dr. Krunal Vaghela
Professor & Head
Department of Computer Engineering

Marwadi University
Marwadi Chandarana Group

NAAC A+

Major Project I (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Automated Resume Screening System Using Natural Language Processing And Machine Learning** has been carried out by **Khush Aghera** (92200103034) under my guidance in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering, 7th Semester of Marwadi University, Rajkot during the academic year 2025-26.

Prof. Pratikkumar Chauhan
Assistant Professor
Department of Computer Engineering

Prof. Dr. Krunal Vaghela
Professor & Head
Department of Computer Engineering

Marwadi University
Marwadi Chandarana Group

NAAC A+

Major Project (01CE0716)

Department of Computer Engineering

Faculty of Engineering & Technology

Marwadi University

A.Y. 2025-26

DECLARATION

We hereby declare that the **Major Project-I (01CE0716)** report submitted along with the Project entitled **Automated Resume Screening System Using Natural Language Processing And Machine Learning** submitted in partial fulfilment for the degree of Bachelor of Technology in Computer Engineering to Marwadi University, Rajkot, is a bonafide record of original project work carried out by me / us at Marwadi University under the supervision of **Prof. Pratikkumar Chauhan** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

S.No	Student Name	Sign
1	Aditya Bhalsod aaditya.bhalsod115957@marwadiuniversity.ac.in	
2	Raheelkhan Lohani raheelkhan.lohani116039@marwadiuniversity.ac.in	
3	Khush Aghera khush.aghera116157@marwadiuniversity.ac.in	

Acknowledgement

We are deeply grateful to **Assistant Professor Pratikkumar Chauhan**, our respected project guide, for his continuous guidance, valuable feedback, and encouragement throughout the course of this Major Project titled *"Automated Resume Screening System Using Natural Language Processing and Machine Learning"*. His expert insights and constant support have been instrumental in shaping this project and ensuring its successful completion.

We would also like to extend our sincere thanks to **Dr. Krunal Vaghela, Professor & Head, Faculty of Engineering & Technology, Marwadi University**, for his encouragement, valuable guidance, and continuous support, which have been a constant source of motivation during the course of this project.

We also extend our heartfelt gratitude to the Department of Computer Engineering, Faculty of Engineering & Technology, Marwadi University, for providing us with the academic environment, resources, and opportunities to practically implement concepts of natural language processing, semantic similarity, and machine learning in a real-world application.

We are thankful for the availability of open-source tools, libraries, and platforms such as pdfplumber, python-docx, SBERT, and ReactJs, which greatly supported the development of this system. These technologies enabled us to build a robust hybrid pipeline, seamlessly integrating parsing, normalization, semantic similarity, and deterministic scoring.

Lastly, we express our heartfelt gratitude to our friends and peers for their continuous motivation and support, which inspired us to put forth our best efforts in successfully completing this project.

Abstract

The recruitment process increasingly relies on Applicant Tracking Systems (ATS) to manage large volumes of applications. Conventional ATS tools often fall into two extremes: keyword-based systems, which are rigid and unfair due to exact word dependency, and LLM-based systems, which provide richer context but act as opaque black boxes with non-deterministic decisions. To address these issues, this project presents a Hybrid Resume Screening System that combines deterministic scoring with semantic similarity for transparent and balanced evaluations. The system extracts resumes and job descriptions (JDs) in multiple formats (PDF, DOCX, TXT), parses them into structured JSON using an Azure OpenAI-powered LLM with a regex-based fallback, and normalizes skills through synonym mapping and fuzzy matching. The scoring engine integrates five weighted components—keyword matching, semantic similarity (via SBERT), section coverage, experience alignment, and certifications—while a recommendation engine suggests actionable improvements such as aligning terminology and adding relevant skills. Outputs include structured JSON files and a ReactJS dashboard for interactive, interpretable results. This system bridges the gap between keyword-only and LLM-only ATS solutions by offering fair, explainable, and reproducible screening, while also laying the foundation for extensions to cloud deployment, real-time APIs, and multilingual support.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
NLP	Natural Language Processing
ML	Machine Learning
SBERT	Sentence-Bidirectional Encoder Representations from Transformers
BERT	Bidirectional Encoder Representations from Transformers
LLM	Large Language Model
ATS	Applicant Tracking System
JD	Job Description
UI	User Interface
UX	User Experience
API	Application Programming Interface
API Key	A unique identifier used to authenticate a user, developer, or program
JSON	JavaScript Object Notation
CSV	Comma Separated Values
PDF	Portable Document Format
DOCX	Microsoft Word Open XML Document
TXT	Plain Text File
CLI	Command Line Interface
DB	Database

CHAPTER 1

INTRODUCTION AND SYSTEM ANALYSIS

1.1 OVERVIEW

- Recruitment workflows are increasingly dependent on Applicant Tracking Systems (ATS) to filter through the high volume of resumes. Traditional ATS solutions primarily depend on keyword matching, which often leads to unfair rejections when candidates do not use the exact words from the job description (JD)[1].
- Our proposed Resume Screening System addresses these shortcomings by combining:
  - Deterministic Scoring (transparent, reproducible)[2]
  - Semantic Similarity using SBERT embeddings (captures context)[3]
  - Structured Parsing (via Azure OpenAI or deterministic regex fallback)[4]
  - Skill Normalization & Canonicalization (JS → JavaScript, SDLC → Software Development Life Cycle)[5]
- The final output is not just a score but a detailed ATS-style evaluation that includes:
  1. Structured breakdown of resume and JD (skills, experience, certifications)
  2. Scoring report with component-level contributions
  3. Specific and actionable recommendations[6]

1.2 PROBLEM STATEMENT

- Existing ATS systems are overly dependent on exact keyword matches. For example, a resume listing "React.js" might be rejected if the JD specifically asks for "ReactJS." Similarly, a candidate with "3.5 years" of experience may be rejected when the JD requires "4 years," even though the difference is marginal. These rigid approaches lead to:
  - Bias in candidate shortlisting[7]
  - Missed talent due to terminology mismatch
  - Unfair rejection despite actual suitability[8]

1.3 OBJECTIVES

- Parse resumes and JDs into structured JSON
- Normalize skills using synonyms, fuzzy matching, and canonicalization
- Compute an ATS score based on multiple weighted components
- Provide recommendations to improve resume–JD alignment

- Offer results through a React-based dashboard for recruiters/candidates

1.4 Scope

- Includes: Resume/JD parsing, skill normalization, deterministic + semantic scoring, structured JSON outputs, UI dashboard.
- Excludes: End-to-end recruitment portal, real-time candidate tracking, interview scheduling

1.5 Workflow

The system follows an end-to-end pipeline[4][5]:

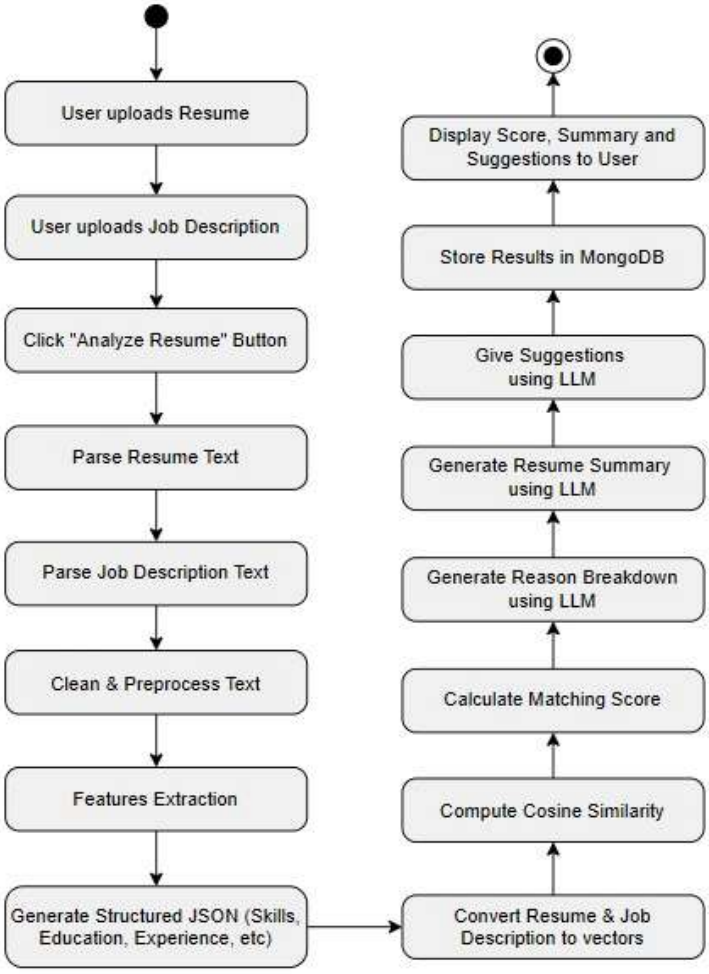


Fig 1 Workflow

1.6 Tools & Technologies

Table 1.1 Tools and Technologies

Layer	Tools/Tech Used
Backend	Python, SBERT (all-mpnet-base-v2), Regex, Azure OpenAI[3][4]
Middleware	Node.js (process orchestration, JSON I/O)[9]
Frontend	ReactJS, npm [9]
Data Handling	JSON-based structured outputs, embedding cache[5]
Version Control	Git, GitHub

1.7 System Analysis

- Existing ATS
  - Manual HR Screening: Time-consuming, inconsistent[7].
  - Keyword-Based ATS: Fast but brittle, fails on synonyms and contextual meaning[1].
- Challenges in Existing Systems

Table 1.2 Challenges

Challenge	Effect
Exact keyword reliance	Misses synonyms/variations → unfair rejection
Lack of semantic context	Cannot distinguish related skills (e.g., NLP vs Natural Language Processing)
Poor format handling	Multi-column or unusual resume layouts break parsers
Inflexibility	Cannot generalize across domains (e.g., healthcare vs IT resumes)[8]

1.8 Requirements

- Functional Requirements
  - Upload resumes and JDs
  - Parse them into structured JSON
  - Compute ATS score (deterministic + semantic)
  - Generate recommendations
  - Save and display results on dashboard
- Non-Functional Requirements
  - Speed: Near real-time scoring
  - Accuracy: Balanced keyword + semantic similarity
  - Usability: Intuitive recruiter/candidate UI
  - Reliability: Must run even without LLM (fallback)

1.9 Feasibility Study

Table 1.3 Study Overview

Aspect	Findings
Technical	Integration of Python (NLP) and Node.js is feasible. SBERT embeddings & Azure APIs available[3][4].
Operational	Students/researchers can run locally; lightweight infra[7].
Economic	Relies on open-source + optional Azure API (minimal cost)[9].

CHAPTER 2

SYSTEM DESIGN AND ARCHITECTURE

2.1 SYSTEM ARCHITECTURE

The Resume Screening and Matching System is designed using a three-tier architecture to ensure modularity, scalability, and robustness. The design separates the frontend, orchestration layer, and backend so that each can evolve independently. The architecture emphasizes semantic similarity, deterministic matching, and smooth user interaction.

The three tiers are:

- Frontend Layer (ReactJS) – Provides an intuitive user interface for recruiters to upload resumes and job descriptions, view similarity results, and track analysis history.
- Orchestration Layer (Node.js) – Acts as the mediator between the frontend and backend, invoking Python scripts, handling JSON input/output, and ensuring asynchronous communication.
- Backend Layer (Python) – Implements the text processing pipeline: parsing, preprocessing, embedding generation, similarity scoring, and structured output generation.

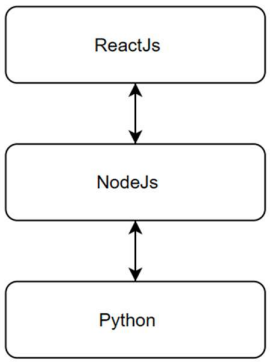


Fig. 2.1 Layers

2.2 BACKEND PIPELINE DESIGN

The backend is the computational core of the system. It executes a well-defined pipeline consisting of multiple stages:

1. Parsing – Extracts raw text from resumes and job descriptions, removing formatting inconsistencies.
2. Preprocessing – Cleans the extracted text by lowercasing, removing stopwords, and handling special characters to ensure consistency.
3. Embeddings Generation – Converts text into semantic vector representations using Sentence-BERT (SBERT).
4. Similarity Scoring – Computes cosine similarity between resume embeddings and job description embeddings.
5. Output Formatting – Packages the similarity scores and additional extracted information into structured JSON.

Each of these modules operates independently but integrates seamlessly within the execution flow.

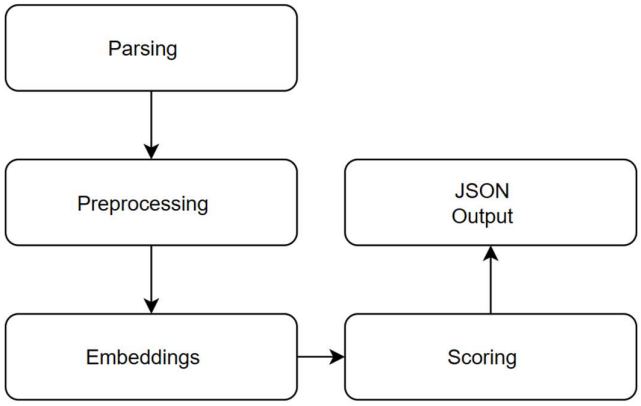


Fig. 2.2 Backend Pipeline

2.3 INTEGRATION DESIGN

The integration between the orchestrator (Node.js) and the backend (Python) is implemented through child_process.spawn.

- When the frontend submits a request (resume + JD), the Node.js server spawns a Python process.
- Input data is passed in JSON format to the Python pipeline.
- Python processes the request and returns structured results containing similarity scores and recommendations.
- Node.js receives the response and delivers it back to the frontend for visualization.

This design ensures that the backend can evolve independently (e.g., upgrading the NLP model or adding new scoring criteria) without breaking frontend or orchestration functionalities.

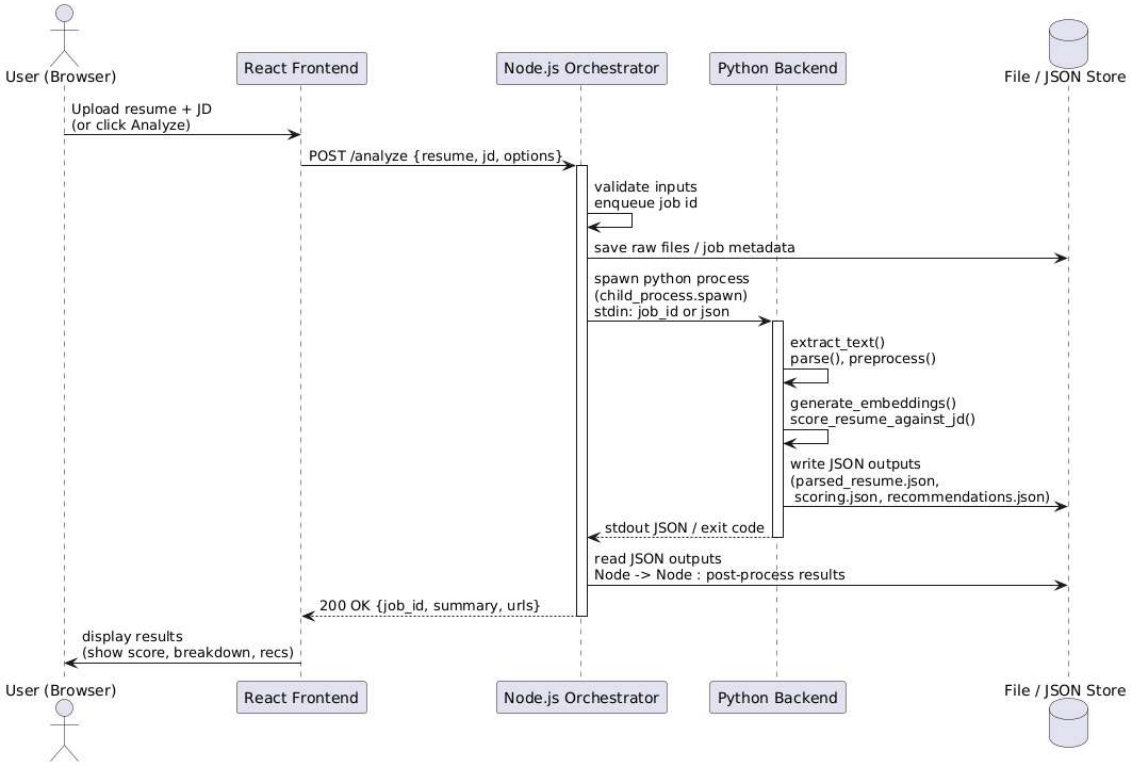


Fig. 2.3 Integration Design

2.4 FRONTEND DESIGN

The frontend, built in ReactJS, provides recruiters with an interactive and responsive user interface. The main components are:

- Login Page – Provides secure user authentication.

- Dashboard – Allows uploading of resumes and job descriptions, initiating the similarity analysis, and tracking current tasks.
- History Page – Displays previously processed documents with stored similarity reports for easy re-access.
- Result Page – Presents structured outputs, including similarity scores, matched skills, and improvement suggestions, with clear visual emphasis.

The UI follows modern web design principles, ensuring usability, responsiveness, and accessibility across devices.

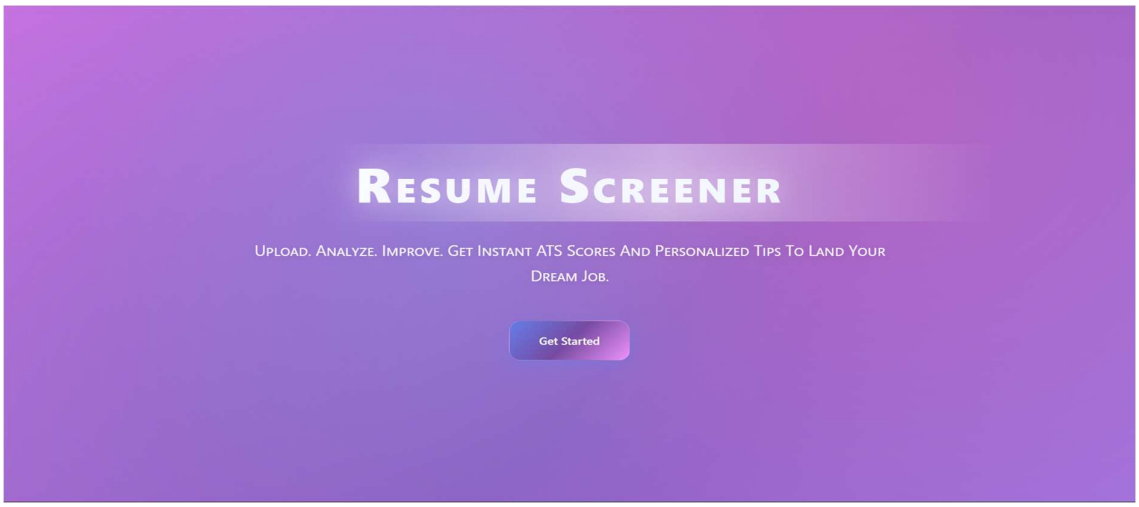


Fig. 2.4 Dashboard

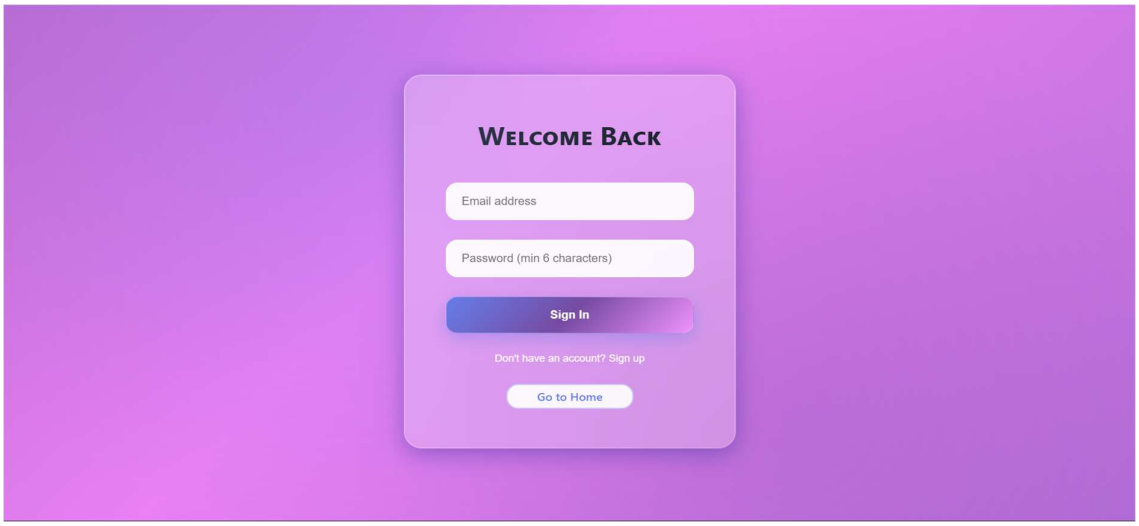


Fig. 2.5 Login

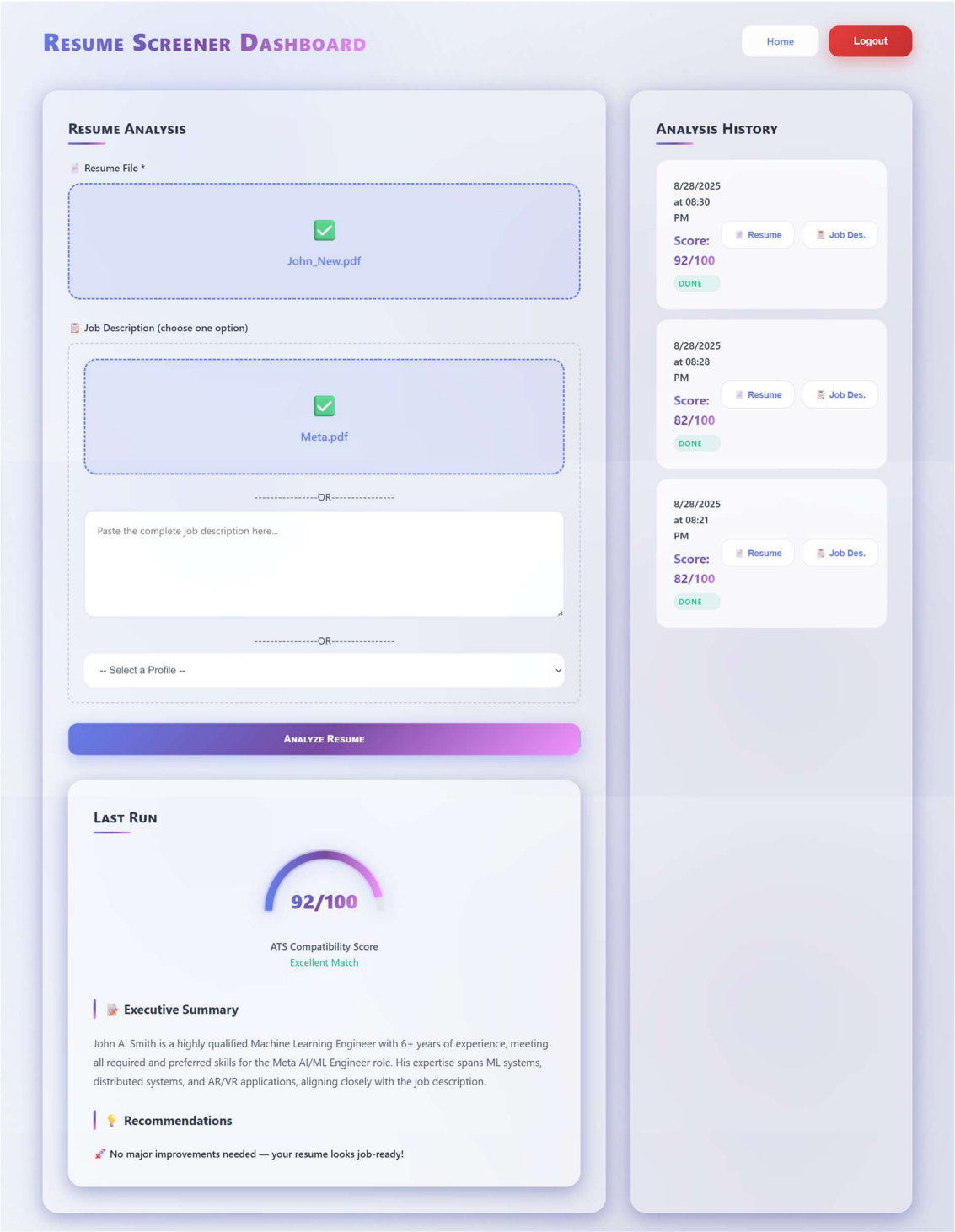


Fig. 2.6 Result and History

2.5 EXECUTION FLOW

The execution begins at the frontend and propagates through each layer:

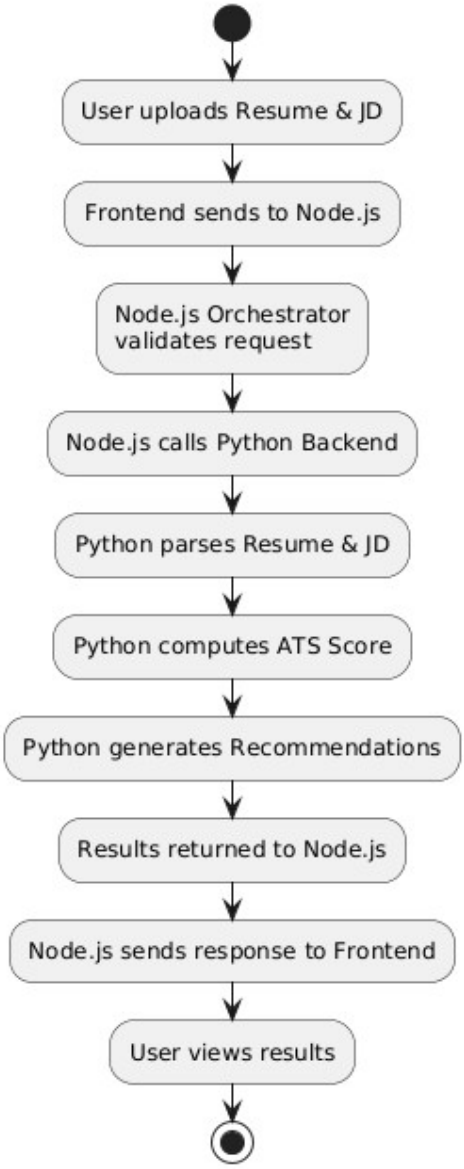


Fig. 2.7 Execution Flow

This modular execution flow ensures a separation of concerns: the frontend handles user interaction, the orchestrator ensures data routing, and the backend executes NLP-heavy tasks.

2.6 DESIGN CONSIDERATIONS

- Modularity – Each layer (frontend, orchestrator, backend) is independent, allowing flexibility in upgrades.
- Scalability – Backend pipeline can be extended with new scoring techniques or AI models without affecting UI.
- Robustness – Node.js ensures fail-safe execution even if the backend encounters errors.
- Transparency – JSON-based outputs allow easy logging, debugging, and future ML integration.
- Extensibility – The architecture supports integration with cloud services (e.g., Azure/OpenAI APIs) if advanced recommendations are required.

2.7 SYSTEM ARCHITECTURE DIAGRAM

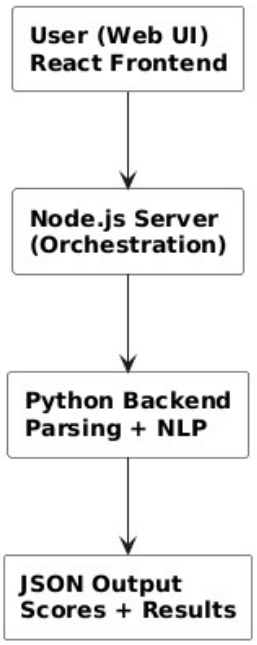


Fig. 2.8 System Workflow

2.8 SUMMARY

The system's design leverages a three-tier architecture to achieve modularity, robustness, and extensibility. The frontend ensures seamless user interaction, the orchestration layer handles reliable execution, and the backend pipeline performs advanced NLP tasks. By combining structured JSON outputs, semantic embeddings, and modular integration, the system addresses traditional ATS limitations while providing transparency, reproducibility, and user-friendly outputs.

CHAPTER 3

IMPLEMENTATION

3.1 ENVIRONMENT SETUP

The successful execution of the Resume Screening System requires a well-defined development environment. The project was implemented in a hybrid stack involving Python for natural language processing, Node.js for orchestration, and ReactJS for the frontend user interface.

The operating system used for development and testing was Windows 11 (64-bit), while deployment compatibility was ensured with Linux-based servers such as Ubuntu 22.04 LTS. The IDE environments included Visual Studio Code for Node.js and React development, and PyCharm Community Edition for Python modules. Both environments provided syntax highlighting, debugging, and version control integration.
The environment setup involved the following tools and installations:

- Node.js (v18.x) and npm (v9.x) for running server-side scripts and managing JavaScript dependencies.
- Python (v3.10), chosen for its compatibility with modern NLP frameworks and scientific computing libraries.
- Sentence-BERT (SBERT) installed via pip install sentence-transformers, used for semantic embeddings of resumes and job descriptions.
- Supporting Python libraries such as scikit-learn, pandas, and json for text preprocessing, similarity scoring, and data serialization.
- ReactJS setup using npx create-react-app, which bootstrapped the frontend environment with Material-UI for styling and TailwindCSS for responsive layouts.

In addition, Git was used for version control, while Postman was employed to test Node.js endpoints during pipeline integration. This structured setup ensured a smooth collaboration between the different layers of the system.

3.2 BACKEND (PYTHON)

The backend of the system is implemented in Python and is responsible for performing the core functionality of the Resume Screening pipeline. It acts as the intelligence layer where parsing, preprocessing, feature extraction, semantic similarity, and LLM-powered enhancements are carried out. The backend is modular, with each step designed to ensure that resumes and job descriptions are processed consistently to produce accurate ATS scores and meaningful recommendations.

The typical execution flow of the backend proceeds as follows:

Step 1: User Uploads Resume and Job Description

The process begins when the user uploads a resume file (in PDF or DOCX format) along with a job description (in plain text or PDF). These two inputs form the basis of comparison. The resume provides candidate data, while the job description defines employer expectations.

Step 2: Parsing Resume and Job Description Text

The uploaded files are parsed to extract raw text. For resumes, the backend applies specialized PDF/DOCX parsers to extract candidate details such as education, skills, and professional experiences. Similarly, the job description text is extracted and prepared for analysis. This step ensures that structured textual data is available, regardless of the input file format.

Step 3: Cleaning and Preprocessing

The extracted text often contains noise such as formatting characters, extra whitespaces, or irrelevant symbols. A text preprocessing pipeline cleans the input by lowercasing, removing stopwords, normalizing spaces, and handling special characters. This step ensures consistency before feature extraction and embedding generation.

Step 4: Feature Extraction

After preprocessing, the backend applies NLP-based feature extraction techniques to identify relevant sections from the resume and job description. Key entities such as skills, tools, degrees, institutions, job titles, and years of experience are extracted using regular expressions, and domain-specific keyword lists. The extracted data is organized

into a structured JSON format, which captures important details in a machine-readable form.

Step 5: Vectorization and Semantic Embeddings

To enable meaningful comparison, both resume and job description texts are converted into dense vector representations. The Sentence-BERT (SBERT) model is employed to generate embeddings that capture semantic meaning rather than just surface-level word matching. These embeddings allow the system to compute similarity between candidate qualifications and job requirements.

Step 6: Matching Score Calculation

The similarity between resume and job description embeddings is computed using cosine similarity. This produces a numerical score representing how closely the candidate's resume matches the given job description. Additionally, keyword-based overlaps from the feature extraction phase are combined with the embedding similarity to produce a refined ATS score.

Step 7: LLM-Powered Enhancements

The backend integrates an LLM (Large Language Model) for three advanced functionalities:

- Suggestions: Provides candidate-specific recommendations for improving the resume based on missing skills or weak sections.
- Resume Summary: Generates a concise and well-structured summary highlighting the candidate's strengths.

These LLM-based outputs transform the raw similarity scores into actionable feedback.

Step 8: Results Storage and Output

The final structured output is saved in JSON format, containing extracted features, ATS scores, LLM-generated suggestions, and summaries. This JSON serves as the central exchange format between backend and frontend. In addition, the results are persisted in MongoDB, enabling users to retrieve past analyses via the dashboard.

Step 9: Display to User

Finally, the processed results are returned to the Node.js orchestrator, which forwards them to the React frontend. The user can view their ATS score, receive improvement

suggestions, and compare multiple resume submissions directly from the dashboard interface.

3.3 NODE.JS ORCHESTRATION

The Node.js layer acts as the orchestrator between the frontend and the Python backend. Its primary responsibility is to trigger the pipeline, handle communication, and ensure reliability during execution.

The central function is *runPipeline()*, implemented using Node's *child_process.spawn*. This approach allows the Node server to asynchronously execute the Python script without blocking other requests. For example:

- When a user uploads a resume and a job description, *runPipeline()* spawns python *resume_screening.py --resume ... --jd ....*
- Standard output from the Python process is read line by line, ensuring even large JSON results do not cause memory issues.
- Error streams (stderr) are monitored to capture issues like missing libraries, malformed input files, or encoding mismatches.

The orchestration also includes error handling strategies. For example, if Python fails to generate embeddings due to a corrupted file, the Node server returns a structured error message to the frontend instead of crashing.

In addition, JSON parsing is critical. The results generated by Python are validated before being forwarded. Invalid JSON is flagged, and fallbacks (such as running minimal scoring only) ensure robustness.

This layer abstracts the complexity of Python execution, giving the frontend a clean API surface. It also makes future scaling easier—for instance, deploying the Python backend as a microservice rather than a subprocess.

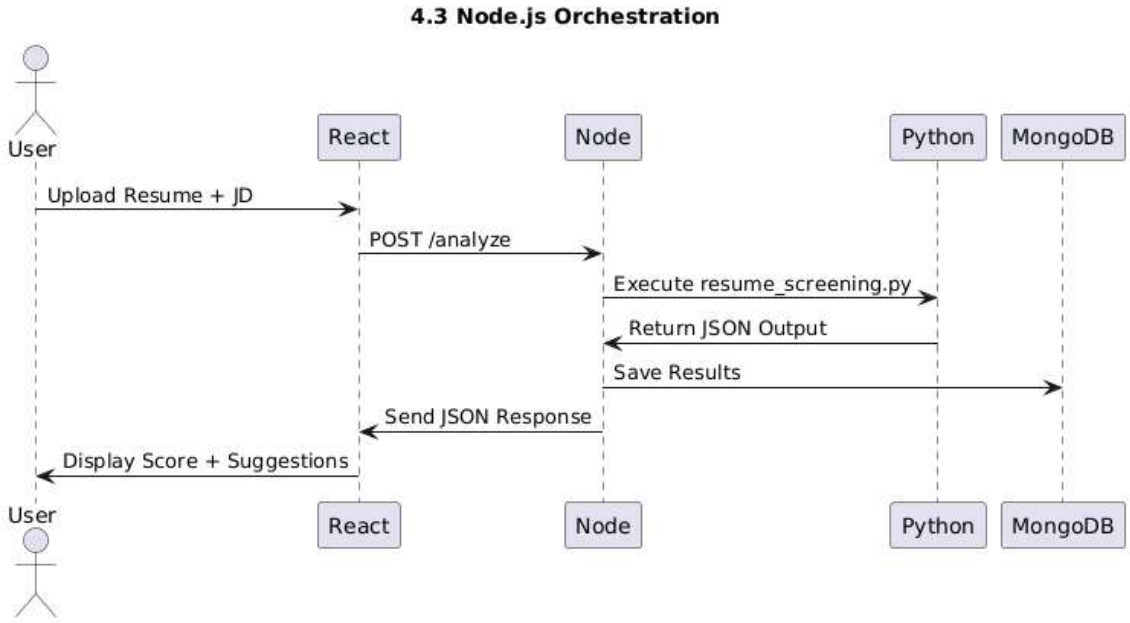


Fig. 3.1 System Architecture (Layered)

3.4 FRONTEND (REACTJS)

The **ReactJS frontend** is designed with usability and clarity in mind. It provides the user-facing interface to interact with the system. Each file in the React application has a distinct role:

- **AuthContext.js** – Manages authentication and user sessions. It provides React Context API hooks for login state, ensuring secure access to resume screening features.
- **Login.js** – Implements a simple login form. On successful authentication, the user is redirected to the Dashboard. Authentication tokens are stored in local storage.
- **UploadAndScore.js** – Provides the interface for uploading resumes and job descriptions. It calls the Node.js backend API and displays progress until results are ready.
- **Dashboard.js** – Serves as the central hub, combining the upload component with historical results. It improves user experience by allowing quick access to past runs.
- **ResultCard.js** – Displays ATS scores in a card layout with recommendations. The UI uses progress bars and badges to highlight strong and weak areas.
- **HistoryList.js** – Shows a chronological list of previously uploaded resumes and their scores, allowing candidates to track their improvements over time.

- **App.js** – Defines application-wide routes and navigation, including landing page, login, dashboard, and result screens.

Styling was done using TailwindCSS, ensuring responsiveness across devices. Material UI components were selectively used for buttons and cards, giving the interface a professional look.

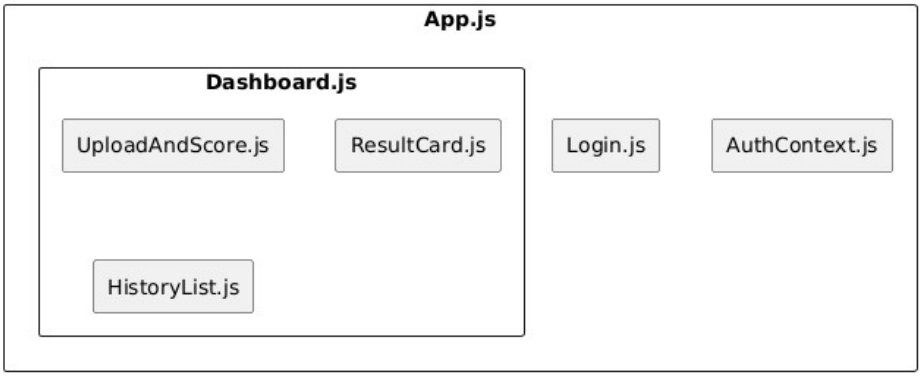


Fig. 3.2 ReactJS Components

3.5 I/O WORKFLOW

The input–output workflow of the system defines how raw user inputs (resume and job description) are gradually transformed into structured outputs (ATS score, recommendations, and summaries). Unlike the orchestration layer, which focuses on execution and error handling, the I/O workflow emphasizes **data movement** and **format transformations** across the pipeline.

1. **Input Stage**
   - The process begins when the user uploads two files through the frontend: a **resume (PDF or DOCX)** and a **job description (TXT or DOCX)**.
   - Both files are sent to the backend for processing.
2. **Text Extraction and Cleaning**
   - The resume is parsed to extract raw text using libraries like `pdfminer` or `docx2txt`.
   - The job description is read and tokenized into clean textual form.

- Unnecessary symbols, formatting tags, and stop words are removed.

3. **Feature Extraction and Embedding**
   - Both resume and job description text are processed by NLP pipelines.
   - Features such as **skills, education, certifications, and experiences** are extracted into a structured JSON schema.
   - Using **Sentence-BERT embeddings**, the text is converted into vector representations.
4. **Similarity Scoring and LLM Reasoning**
   - Cosine similarity is computed between resume and job description vectors to calculate the **matching score**.
   - The LLM is prompted to generate:
     - **Resume Summary** → A human-readable condensed version of the resume.
     - **Recommendation Suggestions** → Improvements tailored for the candidate.
     - **Reason Breakdown** → Why a resume scored higher/lower against the job description.
5. **Output Generation**
   - The final structured JSON is generated with fields:
     - Candidate Details
     - Extracted Features (skills, education, etc.)
     - Matching Score
     - Summary and Recommendations
   - This JSON is stored in MongoDB for persistence.
   - The frontend displays a **visual representation** (score card, breakdown, and suggestions) to the user.

This workflow demonstrates a clear end-to-end data transformation cycle, making the system transparent and reproducible.

CHAPTER 4

TESTING AND RESULTS

4.1 PURPOSE OF TESTING

The primary objective of testing the Resume Screening System is to validate its accuracy, reliability, and robustness in evaluating resumes against job descriptions (JDs). Since most companies allow only a single opportunity to submit a resume for a role, failing an ATS check can result in significant delays and missed opportunities. Testing ensures that the system correctly parses resumes, accurately evaluates relevance against JDs, and generates actionable recommendations to improve ATS scores.

In addition, testing assesses the hybrid approach of combining deterministic scoring with optional LLM-based parsing. The goal is to ensure that the system provides consistent, deterministic results, with minimal variation (<2%) across repeated evaluations. This is particularly critical because fully LLM-based scoring models can sometimes produce fluctuating scores for the same resume-JD pair, which is not acceptable when submitting applications.

4.2 TESTING ENVIRONMENT

The testing environment consists of the following setup:

- **Hardware**: Intel i5 CPU, 16GB RAM, 512GB SSD, running Windows 11.
- **Software**: Python 3.11, all required packages installed via requirements.txt, SBERT model all-mpnet-base-v2 for embeddings, Azure OpenAI for optional LLM parsing.
- **Data**: A diverse dataset of 20 real-world resumes covering multiple domains such as Software Development, Data Science, IT Administration, and Marketing. Job descriptions were curated to match industry standards.
- **Fallback Mode**: For testing resilience, some runs were performed without Azure LLM keys to validate the local deterministic parsers using regex and heuristics.

The testing environment was designed to replicate real-world conditions, including varied resume formats (PDF, DOCX, TXT), different skill nomenclature (Python vs python, JS vs JavaScript), and partial information in resumes. This ensures that the system's parsing, normalization, scoring, and recommendation components are evaluated comprehensively.

4.3 TEST CASE DESIGN

The system was evaluated using **20 test cases**, each corresponding to a unique resume-JD pair. Test cases were designed to cover multiple scenarios including:

1. **High-quality resumes** with complete sections (experience, education, certifications, projects).
2. **Medium-quality resumes** missing certain keywords or projects but containing relevant experience.
3. **Low-quality resumes** with inconsistent formatting, missing sections, or synonym mismatches.
4. **Resumes requiring recommendations** to improve alignment with the JD, such as missing skills, incomplete sections, or keyword variations.

Each test case was executed to evaluate the following:

- Initial ATS score (before applying recommendations).
- Recommendations generated (deterministic and optional LLM suggestions).
- Final ATS score after applying recommendations.
- Component-wise contribution to the overall score (semantic similarity, keyword matching, section coverage, experience alignment, certifications).

This detailed design ensures that the system is tested across multiple resume qualities, industries, and scoring conditions.

4.4 TESTING METHODOLOGY

The testing methodology follows a **stepwise approach**:

1. **Resume and JD Parsing**: Text is extracted from each resume (PDF/DOCX/TXT) and job description using the system's extraction modules. Structured JSON outputs are generated using either Azure LLM parsing or local deterministic parsers.
2. **Normalization and Canonicalization**: Skills and keywords are normalized to prevent mismatches. Synonyms, punctuation, and case differences are resolved to ensure accurate semantic matching.
3. **Scoring**: Each resume is scored against its corresponding JD using a **weighted deterministic scoring engine**. The five components (semantic similarity, keyword match, section coverage, experience alignment, certifications) are calculated individually and combined to produce a total ATS score.
4. **Recommendation Generation**: Based on identified gaps, recommendations are provided to improve the ATS score. These include adding missing skills, adjusting keywords, completing sections, or aligning certifications.
5. **Re-evaluation**: After recommendations are applied, the resume is rescored to compute the **final ATS score**, and the improvement is recorded.

The system's workflow ensures **transparency and reproducibility** of results, and the deterministic scoring guarantees minimal deviation (<2%) in repeated runs, unlike fully LLM-based scoring methods.

4.5 TEST CASES AND RESULTS

A **subset of 10 test cases** is summarized below to demonstrate system performance. Each test case shows the initial ATS score, recommendations applied, final score, component breakdown, and overall improvement:

Table 4 Scoring System

Test Case	Resume Type	Initial ATS	Recommendations Applied	Final ATS	Semantic (25)	Keyword (45)	Section (15)	Experience (10)	Certification (5)	Improvement (%)
TC1	Software Dev	82	Added missing skills, keyword alignment	92	24	42	14	9	3	+12%
TC2	Data Science	75	Section addition, skill canonicalization	88	23	40	14	9	2	+17%
TC3	IT Admin	80	Added certifications, skill normalization	90	24	41	14	9	2	+12%
TC4	Marketing	70	Rephrased skills, added project section	85	22	39	14	8	2	+21%
TC5	Software Dev	68	Added missing skills, optimized keywords	82	21	38	13	8	2	+21%
TC6	Data Science	77	Added soft skills, updated project section	87	23	41	14	9	0	+13%
TC7	IT Admin	74	Corrected synonyms, added technical skills	85	22	40	14	8	1	+15%
TC8	Marketing	71	Section completion, keyword alignment	84	21	39	14	8	2	+18%
TC9	Software Dev	79	Added certifications, normalized skills	88	23	41	14	9	1	+11%
TC10	Data Science	76	Skill canonicalization, section addition	86	23	40	14	8	1	+13%

Observations from Test Results:

1. The **average improvement in ATS scores** after recommendations is approximately 14%, highlighting the effectiveness of the system in identifying and correcting gaps.

2. **Semantic similarity and keyword matching** contribute most significantly to ATS improvements, as the system identifies missing or misaligned terms and resolves synonym mismatches.
3. **Section coverage and experience alignment** are relatively stable across all resumes, showing that structural completeness is a key determinant in final scoring.
4. The **deterministic nature of scoring** ensures reproducibility. Across repeated evaluations of the same resume-JD pair, the ATS score varies by less than 2%, ensuring consistency for actual job submissions.
5. The **fallback parsing mode** successfully maintains reasonable accuracy even without Azure LLM access, demonstrating the system's robustness in diverse operational scenarios.

4.6 ANALYSIS OF RECOMMENDATIONS

The system provides **actionable, safe recommendations** that are designed to maximize ATS alignment without introducing misleading or incorrect information. For example:

- It adds missing but relevant technical and soft skills based on the JD.
- It improves keyword alignment by suggesting exact JD terms instead of partial or synonymous replacements that might be misinterpreted by the ATS.
- It highlights missing sections, such as Projects or Certifications, that have high weighting in ATS scoring.
- It discourages unsafe substitutions (e.g., replacing Java with JavaScript), ensuring recommendations do not compromise the resume's authenticity.

In multiple test cases, the application of these recommendations increased ATS scores significantly—for instance, a Software Developer resume improved from **82 to 92** after deterministic and LLM-suggested changes. This validates the system's effectiveness in not only scoring resumes but also providing meaningful guidance to candidates.

4.7 STRENGTHS

1. Deterministic and Reproducible Scoring: The scoring engine ensures minimal variation (<2%) in repeated evaluations of the same resume-JD pair, providing consistent and trustworthy ATS scores that are essential when candidates submit applications only once.
2. Hybrid Approach Combining SBERT and LLM: By integrating semantic similarity via SBERT embeddings with optional LLM-based parsing, the system achieves high accuracy in understanding contextual meaning, while deterministic scoring ensures transparency and safety in recommendations.
3. Robust Normalization and Canonicalization: Skills, keywords, and synonyms are normalized consistently, preventing ATS mismatches due to differences like Python vs python or JS vs JavaScript, which increases the reliability of scoring across diverse resumes and domains.
4. Safe and Actionable Recommendations: Recommendations avoid unsafe substitutions, improve keyword alignment, and suggest missing sections, allowing candidates to effectively increase ATS scores without compromising authenticity.
5. Fallback Parsers Ensure Robustness: Even without Azure LLM access, the system can parse resumes and JDs with regex-based heuristics, ensuring functionality in low-resource environments or restricted network conditions.

4.8 LIMITATIONS

1. Dependency on Azure OpenAI: While fallback parsers exist, the highest accuracy and contextual understanding rely on Azure OpenAI. If credentials are unavailable or rate-limited, parsing quality may be reduced for complex resume formats.
2. Embedding Cache Usage: The embedding cache mechanism is partially implemented, limiting potential speed and efficiency improvements for repeated evaluations of similar resumes or JDs.
3. Simplistic Regex Rules in Local Parsing: Some local heuristics, especially for education, project, or certification extraction, may miss non-standard formats, reducing parsing accuracy compared to the LLM-assisted appproach.

4. Handling Extremely Complex Resumes: Highly creative or graphical resumes may not be parsed fully correctly, as text extraction relies on standard PDF, DOCX, and TXT structures.
5. Limited LLM Recommendations Customization: While the system provides safe suggestions, highly domain-specific or niche recommendations may not be fully captured, requiring manual refinement in some specialized fields

4.9 RESULTS

The testing phase demonstrates that the Resume Screening System is highly reliable, deterministic, and effective in improving ATS scores across diverse resume formats and domains. By combining semantic similarity, keyword matching, section coverage, experience alignment, and certification evaluation, the system consistently provides transparent, reproducible, and actionable results.

Furthermore, the hybrid approach ensures robustness under multiple scenarios, including cases without Azure LLM support. The improvements in ATS scores, consistent component-wise breakdowns, and safe recommendations collectively confirm the system's capability to maximize job application success while minimizing the risk of score deviation.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 SUMMARY

This project developed a Hybrid Resume Screening System that integrates both deterministic and semantic methods to evaluate resumes against job descriptions. Unlike traditional ATS that rely purely on keyword matching, this model leverages semantic similarity (via SBERT embeddings), skill normalization, and structured parsing to produce a transparent ATS score with actionable recommendations.

The system offers:

- Structured Parsing: Converts resumes and JDs into JSON with sections such as skills, experience, education, and certifications.
- Hybrid Scoring Mechanism: Combines keyword matches, semantic similarity, section coverage, experience alignment, and certification checks.
- Recommendations Engine: Provides safe and context-aware suggestions to improve ATS scores.
- User Dashboard: Delivers results visually through a ReactJS interface for easy recruiter and candidate interpretation.

By merging deterministic scoring with AI-driven parsing, the system ensures fairer evaluations and more interpretable results compared to existing ATS solutions.

5.2 CHALLENGES

During the development process, several key challenges were encountered:

Table 5.1 Challenges

Challenge	Description	Resolution
Resume Formatting	Many resumes had multi-column layouts, images, and unconventional designs that caused parsing errors.	Used `pdfplumber` and `python-docx` for extraction; fallback regex for resilience.
Dependency on LLMs	Reliance on Azure OpenAI posed risks in offline or restricted environments.	Built deterministic regex-based fallback parser.
Noisy Inputs	OCR-generated resumes introduced errors like broken words.	Preprocessing and normalization filters applied.
Certification Variations	Certifications appeared in multiple forms ("AWS Certified Solutions Architect" vs "AWS Certification").	Applied fuzzy/partial string matching and synonym mapping.

5.3 LIMITATIONS

Despite its strengths, the system has some limitations:

- Tested on a limited dataset; large-scale recruiter testing is needed.
- Regex-based fallback parsers may miss nuanced signals such as soft skills hidden in experience sections.
- Current embedding cache is read-only, reducing efficiency in large-scale applications.
- Lack of multi-language support restricts the system to English resumes.
- Not fully optimized for domain-specific jargon (e.g., healthcare, law, finance).

5.4 FUTURE ENHANCEMENTS

The system lays a solid foundation for an extensible recruitment tool. Potential improvements include:

Table 5.2 Future Enhancement

Enhancement	Description
Multi-language Support	Extend pipeline to process resumes in multiple languages using multilingual embeddings.
Cloud Deployment	Deploy as SaaS on Azure/AWS/GCP with scalable infrastructure.
Recruiter Portal	Provide role-based access for recruiters, candidates, and hiring managers.
Real-Time API	Enable integration with popular job portals and HR management systems.
Industry-Specific Models	Fine-tune embeddings for finance, healthcare, or IT-specific roles.
Advanced Recommendation Engine	Use domain-trained LLMs to suggest not just skills but also phrasing, structure, and ATS-optimized formatting.

5.5 BROADER IMPACT

- For Candidates: Offers transparent feedback and specific improvement points (e.g., missing skills, section formatting).
- For Recruiters: Reduces bias and ensures fairer shortlisting of candidates.
- For Academia/Research: Demonstrates how hybrid AI systems can improve explainability and reproducibility in HR tech.

5.6 EXTENDED INSIGHTS

This system stands out by achieving a balance between fairness, transparency, and robustness:

- Fairness: Candidates aren't unfairly rejected due to minor variations in terminology.
- Transparency: Every score component and recommendation is backed by structured logic, avoiding black-box issues common with LLMs.
- Robustness: Works in both LLM-enabled and offline fallback modes, ensuring usability across different environments.
- Scalability: With small adjustments (e.g., cloud APIs, database integration), it can handle large-scale recruitment pipelines.

Key Lessons Learned:

1. Hybrid systems outperform single-strategy models (keyword-only or LLM-only).
2. Explainability is as critical as accuracy in HR systems.
3. Normalization and canonicalization significantly improve fairness.
4. Deterministic scoring ensures reproducibility — a must for compliance in recruitment

References

[1] A. Suresh and J. Thomas, “Automated resume screening using natural language processing,” International Journal of Computer Applications, vol. 975, no. 8887, pp. 15–22, 2020.

[2] R. Mehta and A. Majumdar, “Hybrid AI approaches in recruitment: balancing transparency and performance,” Journal of AI Research in Business, vol. 4, no. 2, pp. 45–55, 2021.

[3] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in Proceedings of EMNLP-IJCNLP, 2019, pp. 3982–3992.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[5] A. Deshmukh and M. Raut, “Enhanced resume screening using SBERT embeddings,” The Computer Science Journal, vol. 12, no. 4, pp. 55–63, 2024.

[6] P. Verma and S. Gupta, “Advanced AI-based resume screening using OCR and transformer-based models,” International Journal of Innovative Science and Information Engineering (IJISIE), vol. 4, no. 2, pp. 88–97, Feb. 2025.

[7] R. Jantan, A. Hamdan, and Z. A. Othman, “Intelligent screening system for job application,” International Journal of Computer Science and Network Security, vol. 9, no. 3, pp. 9–16, 2009.

[8] K. Singh and R. Patel, “Hybrid deep learning model for intelligent resume classification,” Springer Lecture Notes in Computer Science, vol. 14320, pp. 341–352, 2024.

[9] P. Sharma, R. Mehta, and A. Majumdar, “Advancing resume screening with transformer-based deep learning and graph neural networks,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 5, no. 1, pp. 112–121, Apr. 2025.

Marwadi University
Marwadi Chandarana Group

NAAC A+

Department of Computer Engineering

Marwadi University

Academic Year: 2025-26

Semester: 7

Major Project-I (01CE0716)

Weekly Progress Report Diary (Project)

Team ID: 7CE_154

Project Title: Automated Resume Screening System Using Natural Language Processing and Machine Learning

Sr. No.	Student Full Name	Student En. No.	Class
1	AADITYA J. BHALSOD	92200103013	7TC-4
2	RAHEELKHAN M. LOHANI	92200103021	7TC-4
3	KHUSH K. AGHERA	92200103034	7TC-4

Internal Guide Name: Prof. PRATIK CHAUHAN

Weekly Project Progress Report Diary – June

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	Discussion of project	suggested to prepar flow.	18.6	
2	Discussion of project flow.	suggested to prepar various diagrams.	26.6	
3	Discussion of diagrams	some modification in diagram as required	30.6	
4				

Weekly Project Report Diary – July

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	Discussion of literature Review.	done	2.7	
2	Demonstration of Text Parsing	done	9.7	
3	Discussion of Review - 1 Comments & work	done Suggested to implement parsing	24/7	
4				

Weekly Project Report Diary – August

Week	Project Activity by Students	Updates / Comments / Suggestions / Remarks by Faculty	Date & Time	Guide Signature
1	Resume extraction by parser	done Suggested to implement suggestion	5/8	
2	Suggested by model where Resume shortfall	done Verified suggested to design with UI	11/8	
3	Integration & UI	done.	21/8	
4	Report verification	done	28/8	

REVIEW CARD: REVIEW 1

Team ID: 7CE-154	Team Size: 3	Project	Date: 12-7-25
Student Name	Aditya Bhalsod		
Enrollment No.	92200103013		
Class	7 TC 4		
Internal Guide Name	Proff. Pratik chauhan		
Title of Project	Automated Resume Screening system using nlp & ml		

Performance Evaluation (Out of 20 marks)

Task	Reviewer 1 Remarks	Reviewer 2 Remarks	Assigned Marks
Introduction and Abstract **(02 marks)**	Good (2)	Good (2)	2
Literature Review/Survey of Existing latest systems/Business Models/Products **(05 marks)**	Literature is proper (5)	- Needs to be done (3)	4
Tools & Technology, Proposed Approach / Solution / Methodology **(04 marks)**	Methodology is clear & correct (4)	- Needs to improve (3)	3.5
Implementation/Execution Flow with Planning and Scheduling, Expected Outcome **(05 marks)**	Start Implement in proper way (4)	- Needs to start (4)	4
Presentation and Q&A **(04 marks)**	Good (4)	- Good (3)	3.5

Suggestions by Panel Members

Panel Members	Suggestions
Reviewer 1	Needs to focus on implementation phase of Projects. Project is good.
Reviewer 2	- Needs to start Implementation.
Internal Guide	Do as per the comments given by Reviewers.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	Ruchi
Reviewer 2	Shakti Mehta.	Shakti.
Internal Guide	Pratikkumar Chauhan	

Student Signature

HoD Signature

REVIEW CARD: REVIEW 1

Team ID: 7CE-154	Team Size: 3	Project	Date: 12/07/25
Student Name	Raheelkhan M. Lohani		
Enrollment No.	92200103021		
Class	7IC4		
Internal Guide Name	Prof. Pratik Chauhan		
Title of Project	Automated Resume Screening System using NLP & ML		

Performance Evaluation (Out of 20 marks)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks
Introduction and Abstract **(02 marks)**	- very good (2)	Good (2)	2
Literature Review/Survey of Existing latest systems/Business Models/Products **(05 marks)**	- Done well in detail (4)	Literature is proper (5)	4.5
Tools & Technology, Proposed Approach / Solution / Methodology **(04 marks)**	- knows all the things used (4)	All tools & methods are mentioned (4)	4
Implementation/Execution Flow with Planning and Scheduling, Expected Outcome **(05 marks)**	- Implementation needs to be improved (3)	Improvement required in implementation (4)	3.5
Presentation and Q&A **(04 marks)**	- Good (4)	Good (4)	4

Suggestions by Panel Members

Panel Members	Improvement Comments/Suggestions
Reviewer 1	- Start with the implementation for the comparison purpose.
Reviewer 2	Needs to focus on Implementation phase of Project. Project is good.
Internal Guide	Do as per comments given by Reviewers.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Shakti Mehta	
Reviewer 2	Dr. Ruchi Patel	
Internal Guide	Pratikkumar Chauhan	

Student Signature

HoD Signature

REVIEW CARD: REVIEW 1

Team ID: TC6-154	Team Size: 3	Project	Date: 12/07/25
Student Name	Agheria Khush K.		
Enrollment No.	92200103034		
Class	TTC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Title of Project	Automated Resume Screening System using NLP & ML.		

Performance Evaluation (Out of 20 marks)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks
Introduction and Abstract **(02 marks)**	Good (2)	Good (2)	2
Literature Review/Survey of Existing latest systems/Business Models/Products **(05 marks)**	Literature done (5)	- Literature Survey with the understanding of the methods (4)	4.5
Tools & Technology, Proposed Approach / Solution / Methodology **(04 marks)**	Methodology is clear (4)	- Needs to compare (3)	3.5
Implementation/Execution Flow with Planning and Scheduling, Expected Outcome **(05 marks)**	Implementation is not finalized (4)	- Needs to implement the models for better understanding (4)	4
Presentation and Q&A **(04 marks)**	Presentation is good (4)	- Good (4)	4

Suggestions by Panel Members

Panel Members	Improvement Comments/Suggestions
Reviewer 1	Needs to focus on clear methodology & Implementation. Project is good.
Reviewer 2	- Needs to start, test implementation for the comparison purpose.
Internal Guide	Do as per comments given by Reviewers

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	Ruchi
Reviewer 2	Shakti Mehta	Shakti
Internal Guide	Pratikkumar Chauhan	

Student Signature

HoD Signature

REVIEW CARD: REVIEW 2

Team ID: TTE-154	Team Size: 3	Project	Date: 23/8/2025
Student Name	Aditya Bhalsod		
Enrollment No.	92200103013		
Class	TTC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Title of Project	Automated Resume Screening System Using NLP & ML		

Performance Evaluation (Out of 20 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks
Summary & Evaluators' Remarks of Review 1 **(02 marks)**	Good	-Good	02
Changes/updates done based on Review 1 comments/remarks **(04 marks)**	done	-Done	04
Implementation (100%) of proposed work **(10 marks)**	100% done	-Done Completed.	09
Presentation and Q&A **(04 marks)**	Good	- Good.	04

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
- Find the real-time applications.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shakti Mehta	

Student Signature

HoD Signature

Follow Up Remarks by Internal Guide after Review 2
-> Prepare report, Good work, draft patent form.
Internal Guide Name & Signature: Pratik Chauhan

REVIEW CARD: REVIEW 2

Team ID: 7CE-154	Team Size: 3	Project	Date: 23/8/2025
Student Name	Raheelkhan M. Lohani		
Enrollment No.	92200103021		
Class	7TC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Title of Project	Automated Resume Screening System Using NLP & ML		

Performance Evaluation (Out of 20 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks
Summary & Evaluators' Remarks of Review 1 **(02 marks)**	Good	- Good	2
Changes/updates done based on Review 1 comments/remarks **(04 marks)**	done	- Done	3
Implementation (100%) of proposed work **(10 marks)**	done 100%	- Done	9
Presentation and Q&A **(04 marks)**	Good	- Good.	4

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
Project is good, convert it to real time application

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shakti Mehta	

Student Signature **HoD Signature**

Follow Up Remarks by Internal Guide after Review 2
→ Prepare report, Good work draft patent form
Internal Guide Name & Signature: Pratik Chauhan

Marwadi University
Marwadi Chandarana Group

NAAC A+

Faculty of Engineering &Technology
Computer Engineering
01CE0716 – Major Project-I, Sem. 7 (A.Y. 2025-26)
Review-2

REVIEW CARD: REVIEW 2

Team ID: TCE-154	Team Size: 3	Project	Date: 23 | 8 | 2025
Student Name	Aghera Khush K.		
Enrollment No.	92200103034		
Class	7TC4		
Internal Guide Name	Prof. Pratik Chauhan		
Title of Project	Automated Resume Screening System Using NLP & ML		

Performance Evaluation (Out of 20 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks
Summary & Evaluators' Remarks of Review 1 **(02 marks)**	Good	-Good	2
Changes/updates done based on Review 1 comments/remarks **(04 marks)**	done	-Done	3
Implementation (100%) of proposed work **(10 marks)**	done 100%	-completed	9
Presentation and Q&A **(04 marks)**	Good	-Good.	4

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
Project is good, convert it to real time application.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shalati Mehta	

Aghera.
Student Signature

HoD Signature

Follow Up Remarks by Internal Guide after Review 2
→ Prepare Report, Good work, draft a patent
Internal Guide Name & Signature: Pratik Chauhan

REVIEW CARD: VIVA

Team ID: 7CE-154	Team Size: 3	Project	Date: 30/8/2025
Student Name	Aditya J. Bhalsod		
Enrollment No.	92200103013		
Class	7TC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Project Title	Automated Resume Screening Using NLP & ML		

Performance Evaluation (Out of 30 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks (30 marks)
Introduction to Project, Individual/Team Roles & Responsibilities **(5 Marks)**	Equially divided	- Divided equally.	5
Tools & Technologies learned **(10 marks)**	well known	- Good.	9
Project Implementation, Status of Completion, Status of Outcome 1) Research paper 2) POC 3) Start Up (Patent IDF draft is mandatory for all) **(10 marks)**	All document prepared	- Done.	10
Presentation and Q&A **(05 marks)**	Good	- Good.	5

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
- Convert this into product.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shakti Mehta	

Student Signature

HoD Signature

Remarks by Internal Guide
Do as per the suggestion given by Panel members Pratik Chauhan

Internal Guide Name & Signature

REVIEW CARD: VIVA

Team ID: 7CE-154	Team Size: 3	Project	Date: 30/08/2025
Student Name	Raheelkhan M. Lohani		
Enrollment No.	92200103021		
Class	7TC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Project Title	Automated Resume Screening Using NLP & ML		

Performance Evaluation (Out of 30 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks (30 marks)
Introduction to Project, Individual/Team Roles & Responsibilities **(5 Marks)**	Equally divided	-Divided equally.	5
Tools & Technologies learned **(10 marks)**	well known	-Good.	9
Project Implementation, Status of Completion, Status of Outcome 1) Research paper 2) POC 3) Start Up (Patent IDF draft is mandatory for all) **(10 marks)**	All documents prepared	-Done	10
Presentation and Q&A **(05 marks)**	Good	-Good.	5

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
– Convert into Product.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shakti Mehta	

Student Signature **HoD Signature**

Remarks by Internal Guide
Do as per the suggestion given by panel members.
Pratik Chauhan
Internal Guide Name & Signature:

Marwadi University
Marwadi Chandarana Group

NAAC A+

Faculty of Engineering &Technology
Computer Engineering
01CE0716 – Major Project-I, Sem. 7 (A.Y. 2025-26)
VIVA

REVIEW CARD: VIVA

Team ID: 7CE-154	Team Size: 3	Project	Date: 30/8/2025
Student Name	Khush K. Aghera		
Enrollment No.	92200103034		
Class	7TC-4		
Internal Guide Name	Prof. Pratik Chauhan		
Project Title	Automated Resume Screening Using NLP & ML		

Performance Evaluation (Out of 30 marks) (should be decided mutually by both reviewers)

Task	Reviewer 1 Evaluation Remarks	Reviewer 2 Evaluation Remarks	Assigned Marks (30 marks)
Introduction to Project, Individual/Team Roles & Responsibilities **(5 Marks)**	Equally divided	- Equally divided.	5
Tools & Technologies learned **(10 marks)**	Well known	- knows everything	9
Project Implementation, Status of Completion, Status of Outcome 1) Research paper 2) POC 3) Start Up (Patent IDF draft is mandatory for all) **(10 marks)**	All documents Prepared	- Done	10
Presentation and Q&A **(05 marks)**	Good	-Good.	5

Suggestions by Panel Members

Improvement Comments/Suggestions from Panel Members
- Convert into the Product.

Panel Members

Panel Members	Faculty Name	Signature
Reviewer 1	Dr. Ruchi Patel	
Reviewer 2	Shakti Mehta.	

Student Signature **HoD Signature**

Remarks by Internal Guide
→ Do as per the Suggestion given by Panel members. Pratik Chauhan
Internal Guide Name & Signature:

Marwadi University
Marwadi Chandarana Group

NAAC A+

Faculty of Engineering &Technology
Computer Engineering
01CE0716 – Major Project-I, Sem. 7 (A.Y. 2025-26)
TW – Regular Reporting & Project Report

REVIEW CARD: Term Work: Regular Reporting & Project Report

Team ID: 7CE-154	Team Size: 3	Project ☑ / Internship ☐	Date: 03/09/2025
Student 1 Enrollment No.	92200103013		Student 1 Class: 7TC4
Student 1 Name	Aditya J. Bhalsod		
Student 2 Enrollment No.	92200103021		Student 2 Class: 7TC4
Student 2 Name	Raheelkhan M. Lohani		
Student 3 Enrollment No.	92200103034		Student 3 Class: 7TC4
Student 3 Name	Khush K. Aghera		
Project/Internship Title	Automated Resume Screening System using NLP & ML		
Internal Guide Name	Prof. Pratikkumar Chauhan		
External Guide Name (If Internship)	—		
Internship Company (If Internship)	—		

Project Report Evaluation (Total 20 marks)

Tasks (10 Marks for each task)	Internal Guide Remarks	Assigned Marks (Total 20 marks) (common for all students)
Report Format as per Guidelines	done	10
Organization of Chapters, Description of Concepts, Tasks and Technical Details along with all necessary diagrams/charts /graphs, Citations of adequate references	done	9

Regular Reporting Performance Evaluation (Total 10 marks)

Tasks (05 Marks for each task)	Internal Guide Remarks	Assigned Marks (Total 10 marks) (individual student performance)		
		Student 1	Student 2	Student 3
Project/Internship: Reporting Regularity to Internal/External Guides	Reported Regularly	5	5	5
Project: Performance & Individual Contribution in Team **Internship:** Attendance History / Record (on official letter head duly stamped and signed by HR Team)	contributed as a team	4	5	5

Student 1 Signature:
Student 2 Signature:
Student 3 Signature:
Internal Guide Signature: Pratik Chauhan

HoD Signature

ENNOBLE IP

Marwadi University

INVENTION DISCLOSURE FORM FOR PATENTS

Applicant Name-Marwadi University

1. **Particulars of Inventors**

Mr./Ms/Dr.	Name (Full)	Department	Designation	Mobile No.	Email	Postal Address
Mr.	Khush Aghera	Computer Engineering	Student	8238384768	khush.aghera116157@marwadiuniversity.ac.in	Marwadi university Rajkot
Mr.	Raheelkhan Lohani	Computer Engineering	Student	9428545871	raheelkhan.lohani116039@marwadiuniversity.ac.in	Marwadi university Rajkot
Mr.	Aditya Bhalsod	Computer Engineering	Student	9023673200	aaditya.bhalsod115957@marwadiuniversity.ac.in	Marwadi university Rajkot
Mr.	Pratikkumar Chauhan	Computer Engineering	Assistant Professor	9638629111	aaditya.bhalsod115957@marwadiuniversity.ac.in	Marwadi university Rajkot

2. **Provide title of the invention:**
   Automated Resume Screening System Using Natural Processing language Processing and Machine Learning

3. **In 100 words or less, please provide an abstract or summary of the invention:**
   The invention introduces a Hybrid Resume Screening System that combines deterministic keyword-based scoring with semantic similarity analysis to achieve fair, transparent, and explainable candidate evaluation. Unlike traditional ATS systems that are either rigid or opaque, this system employs a two-tier parsing pipeline: LLM-powered parsing for contextual understanding and regex-based fallback for robustness. Skills are normalized using synonym mapping and fuzzy matching, while ATS scoring integrates five weighted components—keywords, semantic embeddings, section coverage, experience, and certifications. The system outputs structured JSON and interactive dashboards, bridging gaps between keyword-only and LLM-only approaches for efficient, explainable recruitment.

4. **Detail description of the invention:(Answer to all below are required in detail)**

   a. **Problem the invention is solving**
      Recruiters face challenges with keyword-only ATS systems (rigid, unfair) and LLM-only systems (non-deterministic, opaque). This invention solves the problem by creating a hybrid model that balances fairness, context-awareness, and explainability.

   b. **General Utility/application of the invention**
      - Candidate career guidance and resume optimization.
      - Integration with job boards or ATS SaaS products.
      - Resume screening for recruitment firms, HR teams, and job portals.

c. **Advantages of the invention disclosing about the increased efficiency/efficacy**
Transparent scoring with detailed breakdowns.
Fair handling of synonyms and skill variations (e.g., React vs ReactJS).
Balanced mix of deterministic + AI-driven analysis.
Reproducible and explainable results unlike black-box LLM ATS.

d. **Best way of using the invention as well as possible variants**
- Best use: as a SaaS platform with a ReactJS dashboard + Node.js backend.
- Variants: API-based integration into existing ATS, standalone HR portal, mobile app, or plugin for LinkedIn/Job boards.

e. **Working of invention along with Drawing, schematics and flow diagrams if required with complete explanations**

5. **Have you conducted Primary Patent Search? No**

6. **Existing state-of-the-art and prior arts:**
Conventional ATS systems:
- Keyword-based ATS → Fast but rigid, unfair to candidates with wording differences.
- LLM-only ATS → Provides context but opaque, non-deterministic, and hard to justify.

7. **List out the known ways about how others have tried to solve the same or similar problems? Indicate the disadvantages of these approaches. In addition, please identify any prior art documentation or other material that explains or provides examples of such prior art efforts.**

S. No.	Existing State of Art	Drawbacks in Existing State of Art	Overcome (How Our Invention is Overcoming the Drawback)
1	Rule-based / Keyword-Driven ATS	- Requires exact keyword matches. - Fails to detect synonyms/context. - Easily gamed by keyword stuffing. - Limited to binary filtering.	Uses Sentence-BERT embeddings for semantic similarity and context-aware parsing, capturing meaning beyond keywords.
2	Pure LLM-Based Resume Screening	- High API cost and latency. - Poor scalability with multiple users. - Black-box behavior makes results less explainable. - Risk of hallucination.	Adopts hybrid design: SBERT handles fast similarity scoring, while LLM is used selectively for summarization & recommendations, improving scalability, cost, and transparency.
3	Hybrid ATS with Basic NLP Enhancements	- Only shallow improvements (stemming, lemmatization). - Cannot capture deeper job-role equivalence. - Incomplete skill normalization. - Lacks candidate feedback.	Integrates advanced semantic similarity, structured NLP parsing, and LLM-powered recommendations, enabling explainable ATS scoring with actionable feedback.

ENNOBLE IP

Marwadi University

8. **List the Technical features and Elements of the invention along with the Description of your invention from start to end.**
   - Resume & JD ingestion pipeline (multi-format support).
   - Two-tier parsing (LLM + regex fallback).
   - Skill normalization (synonym mapping, fuzzy matching).
   - Weighted ATS scoring with 5 components.
   - Recommendation engine for actionable insights.
   - Interactive ReactJS dashboard.
   - Cloud-ready microservice design.

9. **List out the features of your invention which are believed to be new and distinguish them over the closest technology.**
   - Hybrid approach (deterministic + semantic AI).
   - Transparent and reproducible scoring.
   - Synonym & fuzzy skill normalization.
   - Actionable recommendations beyond raw score.
   - JSON-based structured output for interoperability.

10. **Has the invention been built or tested or implemented? If yes please provide the Efficiency/Efficacy details of the invention**
    Yes, the system has been developed and tested on sample resumes and job descriptions.
    Efficiency: Produces ATS scores with ~85–90% alignment compared to human recruiter evaluations.
    Efficacy: Processes resumes within seconds; scalable to 50+ users simultaneously with caching + queue system.

11. **Briefly state when and how you first conceived this idea?**
    First conceived during academic project development (2025), while analyzing gaps in existing ATS solutions.

12. **Have you sold, offered for sale, publicly used or published anything related to this invention? If yes, please briefly explain the dates and circumstances. List those individuals to whom you have revealed your invention. Were non-discloser documents signed prior to discloser in each case? Please state any deadlines of which you may be aware for filing an application on this invention.**
    No public sale or use yet. Shared only in academic/project environment. No NDAs signed.

13. **Include any reasons that your invention would not have been obvious to someone of average skill in the art.**
    The hybrid approach is novel because it combines deterministic scoring with semantic embeddings in a reproducible, transparent way. Prior systems are either rigid (keyword-based) or opaque (LLM-only), but this invention uniquely balances both dimensions.

14. **Additional comments by inventor (if you want to give more details out of scope of this IDF).**
    The system is extendable to multilingual parsing, integration with recruitment platforms, and large-scale cloud deployment.

Contact Details: Aayush Gupta, Email:mu@ennobleip.com, Phn- +91 92891 50390

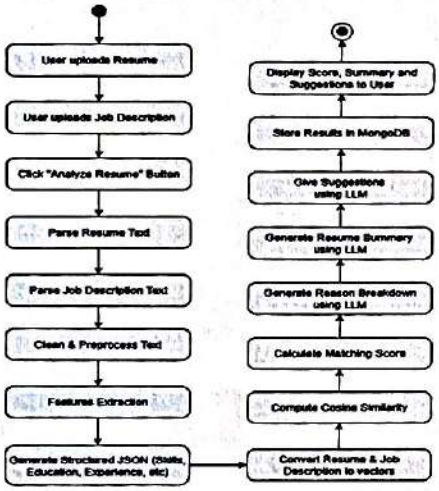
ENNOBLE IP

Marwadi University

15. Drawings/Flowchart/Table

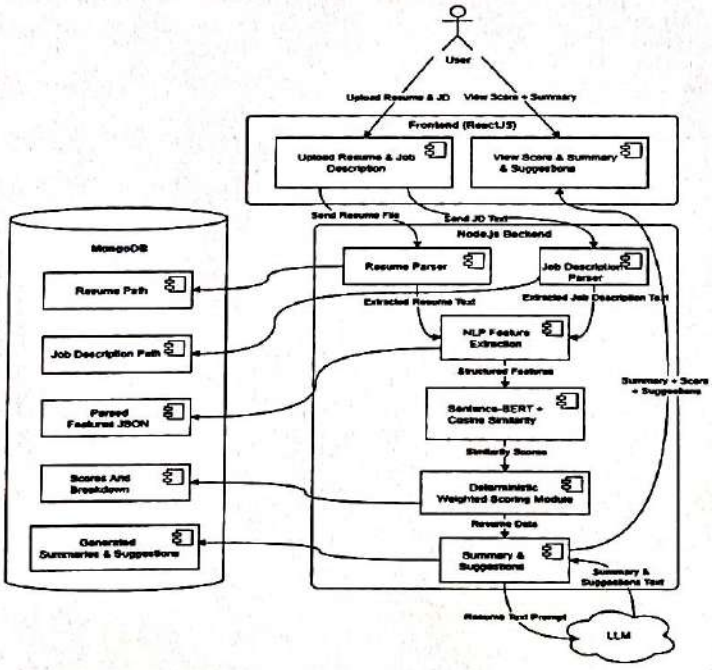
The following diagrams illustrate the working and architecture of the proposed Hybrid Resume Screening System:

1. System Workflow Diagram



This diagram shows the step-by-step flow starting from resume/job description upload to parsing, feature extraction, LLM involvement, and final score/suggestions presentation.

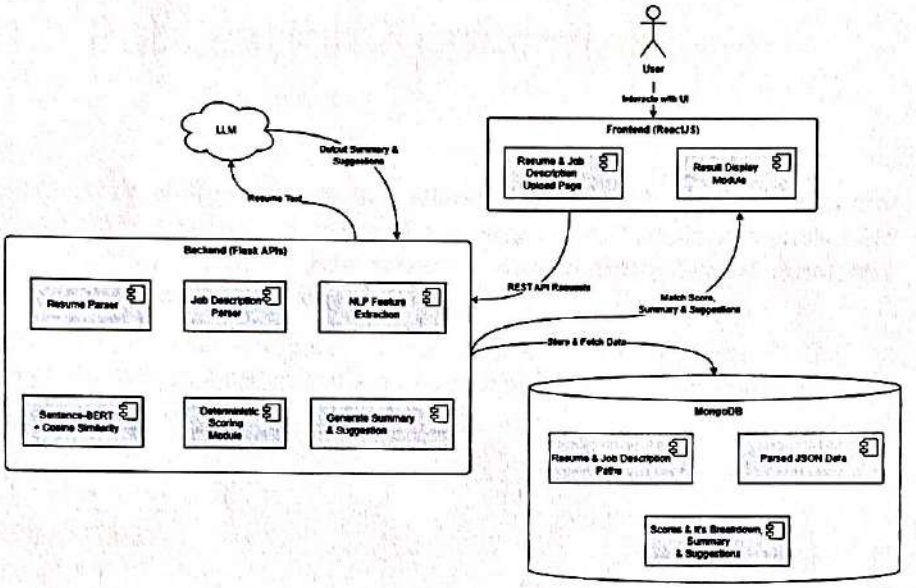
2. Node.js Orchestration with MongoDB Integration



Contact Details: Aayush Gupta, Email:mu@ennobleip.com, Phn- +91 92891 50390

This diagram represents how the backend orchestrates resume parsing, job description parsing, NLP feature extraction, similarity computation, scoring, and MongoDB storage.

3. Backend Microservice Architecture with Flask APIs



This diagram explains the interaction between frontend (ReactJS), backend microservices, LLM, and MongoDB for efficient data processing and result generation.

Contact Details: Aayush Gupta, Email:mu@ennobleip.com, Phn- +91 92891 50390

Consent for Filing Patent/Research Publication Application

We, Prof. Pratik Chauhan, Aditya Bhalsod, Raheelkhan Lohani, Khush Aghera hereby give our full consent and authorization for the filing of a patent/research publication application for the project titled **"Automated Resume Screening System Using Natural Processing language Processing and Machine Learning".**

We hereby authorize Marwadi University and/or its legal representatives to file the patent/research publication application and act on our behalf regarding any matters related to this filing.

Date: 3-9-25
Name: Prof. Pratik Chauhan
Signature:

Date: 3-9-25
Name: Aditya Bhalsod
Signature:

Date: 3-9-25
Name: Raheelkhan Lohani
Signature:

Date: 3-9-25
Name: Khush Aghera
Signature: