**Department of Electrical and Computer Engineering**
**North South University**

# Senior Design Project Report

# AI-powered résumé parsing and ranking

**AMIUR RAHMAN ID#2131466642**

**MD. AZIZ RAIHAN ID#2132681042**

**Faculty Advisor:**

**Dr. Sifat Momen**

**Professor**

**ECE Department**

**Summer, 2025**

# LETTER OF TRANSMITTAL

31st August, 2025

To

Dr. Mohammad Abdul Matin
Chairman,
Department of Electrical and Computer Engineering
North South University, Dhaka

Subject: **Submission of Capstone Project Report on "AI-powered résumé parsing and ranking"**

Dear Sir,

With due respect, we would like to submit our **Capstone Project Report** on **"AI-Powered résumé parsing and ranking"** as a part of our BSc program. The report deals with the development of a complete hiring intelligence platform that automates résumé parsing, scoring, and ranking through a unified Django + ReactJS web application. This project was beneficial to us in gaining experience in the practical field and applying it in real-life recruitment scenarios. We tried to the maximum competence to meet all the dimensions required from this report.

We will be highly obliged if you kindly receive this report and provide your valuable judgment. It would be our immense pleasure if you find this report helpful and informative to have an apparent perspective.

Sincerely Yours,

.........................................................

AMIUR RAHMAN

ECE Department

North South University, Bangladesh

........................................................

MD. AZIZ RAIHAN

ECE Department

North South University, Bangladesh

# APPROVAL

This is to certify that Amiur Rahman (ID #2131466642) and Md. Aziz Raihan (ID #2132681042) of the Department of Electrical and Computer Engineering at North South University have successfully completed the Senior Design Project titled "AI-powered résumé parsing and ranking" under the supervision of Professor Dr. Sifat Momen. This work was carried out in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and has been accepted as satisfactory.

**Supervisor's Signature**

……………………………………….

**Dr. Sifat Momen**

**Professor**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

**Chairman's Signature**

……………………………………….

**Dr. Mohammad Abdul Matin**

**Professor**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

# DECLARATION

It is to declare that this project is our original work. No part of this work has been submitted elsewhere, partially or entirely, for the award of any other degree or diploma. All project-related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been adequately acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students' names & Signatures

**1. AMIUR RAHMAN**

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

**2. MD. AZIZ RAIHAN**

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

# ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Dr. Sifat Momen, Professor, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance, and advice on the experiments, research, and theoretical studies carried out during the course of the current project and also in the preparation of the report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh, for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

# ABSTRACT

This project presents a comprehensive solution for automated resume screening, a process that is often time-consuming and inefficient for recruiters. Existing methods typically rely on basic keyword matching, which can lead to missed opportunities and a lack of nuanced candidate evaluation. To address these limitations, we developed a unified web application that allows recruiters to upload a job listing and a collection of resumes. The system then employs an advanced artificial intelligence model to perform a multi-dimensional analysis, evaluating a candidate's holistic skill set, career progression, and overall fit. Evaluation of the system demonstrated strong performance across key metrics, including high accuracy and a significant proportion of positive user feedback regarding its effectiveness and compatibility. The final product is not just a ranking tool, but a complete hiring intelligence platform that provides transparent and actionable insights, empowering recruiters to make more informed decisions and significantly streamline the entire hiring process.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1 Introduction

## 1.1 Background and Motivation

The modern recruitment landscape has been dramatically reshaped by the proliferation of digital applications. For a single job opening, a company may receive hundreds of resumes, creating an immense challenge for human resource professionals who must manually screen them. This manual review process is not only time-consuming, but it is also highly susceptible to human error and fatigue. Furthermore, the widespread adoption of traditional Applicant Tracking Systems (ATS) has not fully resolved this issue. These systems often rely on a rigid keyword-based filtering approach, which frequently fails to recognize synonyms, contextual relevance, or nuanced skills. As a result, many highly qualified candidates are mistakenly overlooked, leading to prolonged hiring cycles, inconsistent candidate evaluation, and the perpetuation of unconscious biases that can impact the diversity of the talent pool. This critical context highlights a pressing need for a more sophisticated, intelligent, and efficient solution to manage the initial stages of recruitment

This project was motivated by the desire to solve the fundamental problems of modern recruitment, where traditional keyword-based screening often overlooks qualified candidates and introduces bias. Our objective was to develop a system that moves beyond this simplistic filtering, providing a sophisticated analytical framework for comprehensive candidate evaluation. This tool is designed to not only automate the time-consuming task of sorting resumes but also to significantly enhance a recruiter's ability to identify the best-suited candidates. By leveraging an advanced analytical model, our system intelligently assesses skills, experience, and career progression with a nuanced understanding of context, allowing it to surface talent that might otherwise be missed. This innovative approach is designed to mitigate human bias, accelerate the hiring cycle, and ultimately empower recruiters to make more objective and informed decisions.

## 1.2 Purpose and Goal of the Project

In this work, we introduce **AI-Powered Résumé Parsing and Ranking**, an advanced recruitment automation system that integrates multi-layered text processing, semantic analysis, and high-speed batch data handling within a unified Django + ReactJS framework. The system has been designed to address the core inefficiencies of traditional Applicant Tracking Systems (ATS), which are heavily dependent on rigid keyword-based matching and often fail to capture the deeper contextual relationships between job requirements and candidate profiles.

The platform processes structured and unstructured candidate data through a **multi-stage analytical pipeline**. This pipeline incorporates advanced natural language understanding techniques, pattern recognition models, and dynamic feature extraction algorithms to evaluate skill alignment, career trajectory, and contextual fit. The architecture supports **parallelized batch processing**, enabling the simultaneous evaluation of hundreds of résumés without sacrificing precision or speed.

The **AI-Powered Résumé Parsing and Ranking** system is capable of producing not only compatibility scores but also **transparent, data-backed ranking reports** for each candidate. This ensures that recruiters receive actionable intelligence rather than opaque scores, thereby improving decision-making and reducing the risk of bias.

Key contributions of our work include:

- **Parallelized Batch Evaluation:** An optimized processing engine capable of parsing and ranking large volumes of résumés in a single operation.

- **Advanced Text Understanding:** Utilizes multi-phase parsing and semantic similarity measures to detect relevant qualifications beyond simple keyword occurrence.

- **Uniform Requirement-Candidate Processing:** Applies the same analytical rigor to job descriptions and candidate résumés, ensuring precise alignment.

- **Transparent Candidate Reports:** Generates structured outputs that justify rankings using multiple performance dimensions.

- **Scalable, Privacy-First Deployment:** Incorporates secure file handling, optional malware scanning, and modular design to support large-scale enterprise use.

The following sections of this report provide an in-depth explanation of the system's design, the methodology behind its analytical models, and performance metrics derived from extensive testing. By combining **multi-dimensional analysis** with automated ranking, the platform offers a **next-generation recruitment solution** that significantly accelerates candidate shortlisting while maintaining fairness and accuracy.

## 1.3 Organization of the Report

This report is structured into eight chapters that build a complete understanding of the project:

**Chapter 1** introduces the project, outlining its objectives, motivation, and background.
**Chapter 2** reviews related work and literature, highlighting existing approaches in résumé parsing and ranking systems.
**Chapter 3** describes the system design and methodology, including the architecture, backend (Django + DRF), frontend (React + Tailwind), database, and integration of APIs such as Google OAuth and Groq.
**Chapter 4** presents the implementation process, explaining how the different modules were developed, integrated, and tested.
**Chapter 5** discusses the results and evaluation, covering testing outcomes, performance, and validation of the system's effectiveness.
**Chapter 6** focuses on project planning and budgeting, including timeline planning, resource allocation, and the cost breakdown.
**Chapter 7** addresses the complex engineering problems and challenges faced during the project, along with the solutions adopted.

**Chapter 8** concludes the report, summarizing the contributions of the project and suggesting possible directions for future work.

# Chapter 2 Research Literature Review

## 2.1 Existing Research and Limitations

End-to-End Resume Parsing and Candidate Ranking Using BERT

Bhatia et al. (2019) proposed an end-to-end deep learning-based system for resume parsing and candidate ranking using BERT (Bidirectional Encoder Representations from Transformers). The system operated in two stages: first, by extracting key information from resumes, and second, by ranking candidates based on BERT-based phrase pair classification. The model achieved a 72.77% accuracy in classification. However, the study identified a limitation in developing a universal parser for all resume formats, as complex structures led to data loss. To mitigate this, they utilized LinkedIn-style resumes for consistency. Future work suggested incorporating vision-based site segmentation to enhance structural comprehension of resumes[1].

Resume Information Extraction with a Novel Text Block Segmentation Algorithm

Wang & Zu (2019) introduced a deep learning-based resume parsing pipeline using neural network classifiers and distributed embeddings. Their approach combined text block segmentation with resume fact identification, leveraging position-wise line information, integrated word representations, and Named Entity Recognition (NER). The pipeline normalized sections like personal information, work experience, education, and skills. However, reliance on predefined structures restricted adaptability to unconventional resume layouts. They proposed enhancing the system using ontology-based frameworks to better manage varied structures [2].

DeepResume: Enhancing Resume Parsing with Deep Learning

The DeepResume system introduced an advanced deep learning approach using Transformers, CNNs, and RNNs to effectively extract structured data from unstructured resume text. By employing NER and contextual understanding algorithms, DeepResume significantly improved accuracy in extracting critical details like skills, experience, and contact information. This approach successfully overcame limitations in traditional rule-based systems, making candidate screening more scalable and efficient[3].

Automated Resume Parsing Using NLP and Machine Learning Techniques

This paper proposed an NLP and machine learning-based system to automate the extraction and analysis of key information from resumes. Using methods like NER, pattern matching, and semantic analysis, the system efficiently extracted details from formats like PDF and Word, including personal information, work experience, education, and skills. The process also involved ranking candidates based on job relevance. The system's ability to handle multiple formats and languages made it highly adaptable for diverse job markets, enhancing overall recruitment efficiency [4].

Systematic Literature Review of ML and DL in Resume Parsing

This Systematic Literature Review (SLR) explored advancements in resume parsing through Machine Learning (ML) and Deep Learning (DL) techniques. Analyzing 317 research articles, the study detailed how AI-driven strategies have improved recruitment efficiency while reducing human effort. Key insights identified various ML and DL approaches, along with the associated challenges, improvements, and ethical considerations. The structured SLR methodology provided a comprehensive review, aiding the development of scalable and effective recruitment solutions [5].

NLP-Based Approaches

Information Extraction from Free-Form CV Documents in Multiple Languages

Vukadin et al. (2021) developed a dual-model NLP system to extract information from unstructured, multilingual CVs. Leveraging transformer architectures and the encoder component of BERT, the model extracted data at both the section and item levels. They introduced unique tokens like [NEW LINE] and [SKILL] to enhance parsing accuracy. Although successful, challenges arose in handling diverse linguistic structures. Future work suggested incorporating more languages to enhance adaptability [6].

Automatic Extraction of Usable Information from Unstructured Resumes

Kopparapu (2015) designed a two-pass NLP system for extracting structured information from unstructured resumes. The first pass grouped resume data into labeled blocks, while the second applied heuristics and pattern-matching algorithms to extract detailed information. The system achieved 91% accuracy and 88% recall across various formats. However, it faced challenges in identifying precise data for high-skilled roles. The study emphasized enhancing precision in candidate selection for specialized roles [7].

Conditional Random Fields (CRF)-Based Approaches

Information Extraction from Resume Documents in PDF Formats

Chen et al. (2016) addressed the complexity of extracting data from PDF resumes using a hierarchical extraction technique. The process involved breaking pages into blocks, categorizing each block using a Conditional Random Field (CRF) model, and employing sequence labeling for detailed information extraction. Layout-based features were crucial, improving F1-scores by over 20%. However, their approach was tailored for PDF formats, limiting its adaptability to other document types. Future recommendations included refining page segmentation techniques [8].

Study of Information Extraction in Resume Parsing

Nguyen et al. (2018) proposed a deep learning approach combining CNN, BiLSTM, and CRF for entity recognition and text normalization across multiple resume formats. The model achieved an F1-score of over 81%, effectively identifying entities like names and skills. However, limitations included insufficient training data and the need for advanced computational resources. The authors suggested improving hardware configurations and expanding datasets to enhance performance [9].

Ontology based approach

This paper discusses an ontology-driven information extraction system designed to convert unstructured résumés into a structured format, enabling easier expert search via the Semantic Web. The system uses ontologies such as Education, Location, Occupation, and Résumé Ontologies to extract and parse relevant information. It involves converting resumes into plain text, segmenting the content, and applying ontology-based parsing techniques. The results show that this system enhances the efficiency of parsing and structuring résumés. However, there are research gaps in improving system efficiency, expanding language support, and better handling complex or non-standard resume formats.[10]

This study introduces the Ontology-based Résumé Parser (ORP), developed in collaboration between Kariyer.net and TUBITAK, to convert structured résumés into an ontological model. The system, designed to improve expert search for companies, follows a Semantic Web approach and is tested on Turkish and English résumés. While promising for efficient expert finding, the research is limited in language diversity and scalability.[11]

This study presents an ontology-driven information parsing system designed to convert free-form résumés into a structured format for expert finding. The system extracts key information such as experience, education, and business-related details from résumés within human resources databases. Using a Semantic Web approach, it aims to process millions of résumés efficiently. However, limited prior research in this area and challenges related to scalability remain key concerns. [12]

Machine Learning based approach

This paper reviews ML and NLP-based resume screening techniques, focusing on semantic search for context-based parsing. It compares existing resume parsers and discusses ML models that analyze unstructured text like a human. The study highlights research gaps in handling diverse writing styles, word choices, and syntax variations, emphasizing future improvements in resume parsing accuracy. [13]

With the rise of online recruitment, job seekers submit resumes in varied formats, making filtering difficult. This study proposes an NLP-based resume parser using Named Entity Recognition (NER) from Stanford CoreNLP to extract candidate details and categorize resumes into domains (e.g., Computer Science, Business Development). The system converts unstructured resumes into a structured format, improving recruitment efficiency. It achieves 91.47% accuracy in resume classification. [14]

Finding the right match between job seekers and employers is a major challenge in today's job market. This study proposes an ML and NLP-based system with two key components: a Resume Parser and a Job Recommender. The Resume Parser, implemented in Python using spaCy and

PDFMiner, extracts essential details from resumes. The Job Recommender analyzes job titles using TF-IDF and cosine similarity to suggest relevant job openings. This approach streamlines recruitment by automating skill extraction and job matching.[15]

AI-Based Resume Parser and Job Matcher

Alsharef et al proposed an AI-based system for automated resume parsing and candidate matching. Their work demonstrated the foundational, yet limited, nature of early keyword-based approaches, which often failed to handle diverse resume formats and could lead to the unfair rejection of qualified candidates. This paper serves as an essential baseline, highlighting the problems our project aims to solve. [16]

Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening

Singh et al. introduced a Large Language Model (LLM) based agent framework for resume screening. This study demonstrates the potential of LLMs for efficient summarization and candidate grading. However, it also points to the "black box" nature of these models, which can obscure the reasoning behind their decisions. Our project aims to build upon this work by creating a more transparent and explainable agent that provides clear insights to recruiters. [17]

Automated Resume Screening System Using AI Model

This paper presents a complete, end-to-end system for automated resume screening. The authors detail a system that uses an AI model, specifically prompted as an Applicant Tracking System (ATS), to analyze resumes and compare them against job descriptions. A key contribution is the discussion of the structured workflow, which includes job posting, application submission, and an AI-driven model that not only screens but also provides feedback to candidates. [18]

# Chapter 3 Methodology

## 3.1 System Design

The system architecture for the proposed **AI-Powered Résumé Parsing and Ranking** platform, encapsulates the end-to-end operational workflow, spanning from secure file ingestion to ranked candidate presentation. The implementation follows a modular, service-oriented paradigm, unifying a **Django REST Framework (DRF)** backend with a **React.js** frontend, orchestrated by asynchronous task execution pipelines and underpinned by rigorous data security protocols.

#### 1. Input Acquisition Layer

The recruitment workflow commences with the recruiter initiating two distinct uploads via the **React.js** dashboard:

1. A **Job Description (JD) PDF** encapsulating the formal role requirements.
2. A **ZIP archive** containing multiple candidate résumés, strictly in PDF format.

File transmission occurs over secure HTTPS channels, with client-side authentication enforced through **JWT tokens** or **Google OAuth** integration to guarantee request legitimacy.

### 2. Validation and Security Enforcement Layer

Upon reception, the DRF-powered **API Gateway** conducts a series of validation checks, encompassing:

- **MIME type and extension verification** to ensure conformance with allowed formats.
- **File size thresholds** to preclude resource exhaustion attacks.
- **Optional antivirus scanning** using **ClamAV** via clamd bindings (file_secuirity.py), which performs signature-based threat detection and halts processing upon identifying malicious payloads.

Validated files are persisted into a controlled-access **media/ directory**, ensuring that unverified files never enter the processing pipeline.

## 3. Asynchronous Processing and Orchestration Layer

To decouple user interactions from computationally expensive résumé evaluation tasks, the system employs **Celery** with **Redis** as the message broker and **django-celery-results** for task result persistence. This architecture enables horizontal scalability, allowing additional Celery workers to be provisioned as workload demands escalate.

The processing workflow is bifurcated into two principal Celery tasks:

### Stage 1 — Job Description Processing

- **Text Extraction**: The JD PDF is processed using fitz (**PyMuPDF**) to extract unstructured text content.
- **Title Identification**: The extracted corpus is passed to the AI parsing logic, which applies natural language analysis to isolate the primary job title.
- **Persistence**: The JD text and associated title are committed to the JobRequirement model in the relational database.

### Stage 2 — Résumé Parsing and Ranking

- **Archive Expansion**: The ZIP archive is decompressed within a temporary filesystem sandbox, isolating valid PDF files for subsequent parsing.
- **Text Acquisition**: Each résumé undergoes text extraction via fitz, producing a raw text representation.
- **AI Evaluation**: The résumé text, JD text, and job title are jointly submitted to the AI evaluation logic. The AI returns a structured JSON document comprising:
    - extracted_info: Candidate metadata (Name, Email, Phone, Skills, Education, Experience, etc.).
    - ranking_analysis: A numerical CompatibilityScore (0–100) and a curated list of key strengths.

- **Database Insertion**: Each processed résumé is recorded in the RankedResume model with denormalized top-level fields (compatibility_score, candidate_name, candidate_email) for query optimization.

### 4. Data Persistence and Auditability Layer

During development, the system leverages **SQLite** as its primary datastore, with schema design accommodating migration to production-grade RDBMS solutions. Every state-altering action is logged in the AuditEvent model, capturing metadata such as user identity, timestamps, and contextual parameters — enabling full operational traceability and compliance alignment.

### 5. Presentation and Retrieval Layer

The **Results API** exposes candidate rankings for a given résumé batch, sorted in descending order of compatibility score, returning the top ten matches. The **React.js** frontend consumes this API to render:

- Candidate identification details.
- Extracted skillsets and qualifications.
- AI-generated compatibility scores and strengths.

UI components are styled using **Tailwind CSS** and animated with **Framer Motion**, delivering a responsive and dynamic user experience. Secure HTTP requests are facilitated by **Axios**, with automatic token injection for authenticated sessions.

### 6. Architectural Principles

The architecture adheres to the principles of **security**, **scalability**, and **observability**:

- **Security**: Multi-layered access control, antivirus scanning, and encrypted communications ensure the confidentiality and integrity of sensitive PII.
- **Scalability**: Asynchronous processing enables high-throughput batch handling without degrading interactive responsiveness.
- **Observability**: Structured audit logging and database-level traceability facilitate post-event analysis and compliance reporting.

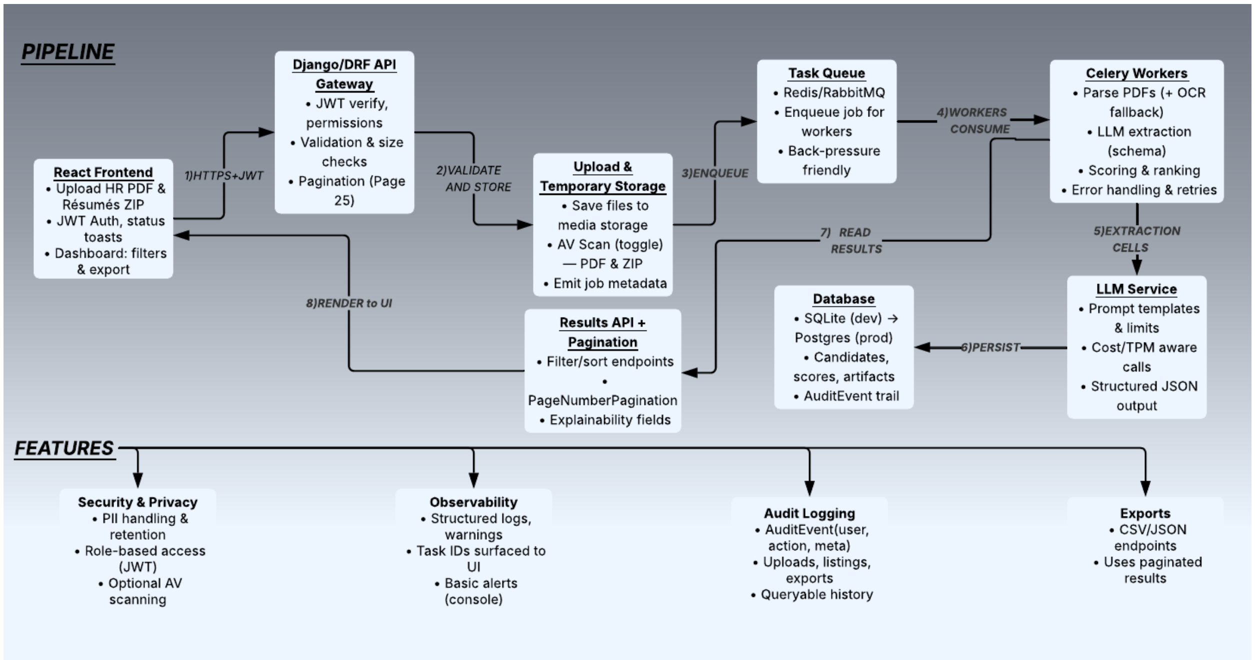


Figure no 3.1.1: System Design

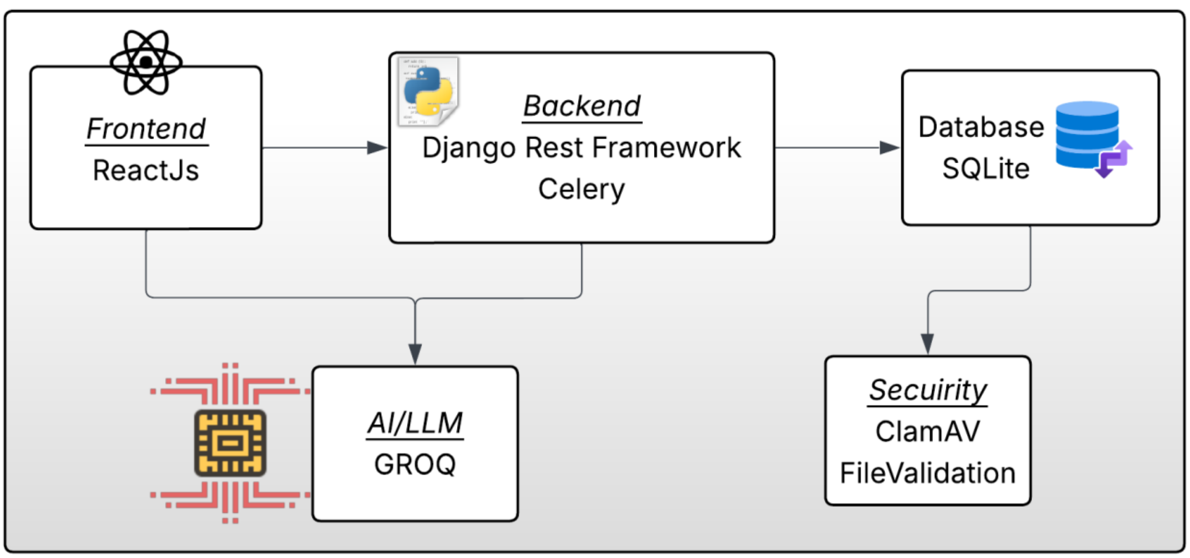


Figure no 3.1.2: System Architecture

## 3.2 Hardware and/or Software Components

The AI-Powered Résumé Parsing and Ranking platform is a fully software-based solution that combines a Django REST Framework backend, a React.js frontend, asynchronous background processing, and AI-powered document parsing. The system runs on standard computing hardware or a cloud-based virtual server.

The following table lists the primary tools and technologies used in the project.

Table 3.1.1: List of Software/Hardware Tools

| Tool | Functions | Other Similar Tools (if any) | Why Selected This Tool |
|---|---|---|---|
| **Django** | Backend web framework for API endpoints, database operations, and authentication. | Flask, FastAPI | Robust framework with built-in ORM, authentication, and admin interface; integrates well with Django REST Framework. |
| **Django REST Framework (DRF)** | Serializers, request validation, and API response formatting. | Flask-RESTful, FastAPI | Easy integration with Django; powerful serializer and authentication support. |
| **Celery** | Asynchronous background processing for batch résumé parsing/ranking. | RQ, Huey | Well-documented, scalable task queue with strong Django integration. |
| **SQLite** | Lightweight relational database for storing user data, job descriptions, parsed résumés, and scores. | PostgreSQL, MySQL | Easy to set up for development; no separate server required. |
| **JWT (djangorestframework-simplejwt)** | Token-based authentication for secure API calls. | OAuth2, Session Auth | Simpler and stateless authentication for frontend-backend communication. |
| **Google OAuth (django-allauth)** | Social login via Google account. | Auth0, Firebase Auth | Widely used, integrates easily with Django, secure and user-friendly login method. |

| | | | |
|---|---|---|---|
| **Groq LLM API** | Parses résumé content and job descriptions, returning structured JSON. | OpenAI GPT, Anthropic Claude | High-speed LLM service with low latency, supports structured JSON output for automation. |
| **ClamAV via clamd** | Antivirus scanning of uploaded files. | Sophos, Avast CLI | Open-source, integrates with Python, reliable malware detection for uploads. |
| **fitz (PyMuPDF)** | PDF parsing for extracting job description text. | pdfminer.six, PyPDF2 | High performance and accuracy in extracting text from PDFs. |
| **python-docx** | DOCX résumé parsing. | docx2txt, Mammoth | Reliable and easy DOCX text extraction in Python. |
| **React.js** | Frontend framework for interactive UI. | Angular, Vue.js | Component-based, fast rendering, and widely adopted. |
| **Tailwind CSS** | Utility-first CSS framework for responsive design. | Bootstrap, Bulma | Flexible styling, highly customizable, modern design approach. |
| **Framer Motion** | Frontend animations and transitions. | GSAP, Anime.js | Easy to use with React, smooth animation effects. |
| **Axios** | HTTP client for API calls from frontend. | Fetch API, SuperAgent | Promise-based, easier syntax for request/response handling in React. |

## 3.3 Hardware and/or Software Implementation

The system is implemented entirely in software and follows a modular architecture that separates frontend, backend, and background processing layers. The implementation steps are as follows:

### 1. User Authentication and Access Control

- **JWT Authentication**: Implemented using djangorestframework-simplejwt to secure all API endpoints.
- **Google OAuth**: Integrated via django-allauth to allow users to log in with their Google accounts.
- The React frontend stores and injects JWT tokens into all outgoing API requests via Axios interceptors.

### 2. File Upload and Validation

- **Job Description Upload**: The recruiter uploads a PDF containing job requirements through the JobRequirementUploadView endpoint.
- **Résumé Batch Upload**: A ZIP archive containing multiple résumé PDFs is uploaded through the ResumeBatchUploadView endpoint.
- **File Validation and Security**:
    - antivirus scanning using ClamAV (file_secuirity.py) blocks malicious files.
    - ZIP extraction only processes files ending in .pdf.

### 3. Asynchronous Processing (Celery + Redis)

- **Stage 1 – Job Description Processing**:
    - Extracts text from the uploaded PDF using fitz (PyMuPDF).
    - Sends extracted text to the Groq LLM API to identify the job title.
    - Saves extracted text and title in the JobRequirement model.

- **Stage 2 – Résumé Parsing and Ranking**:
    - Extracts each résumé from the ZIP archive into a temporary directory.

    - Parses résumé text with fitz.
    - Sends résumé text, job description, and job title to the Groq LLM API via llm_utils.py.
    - LLM returns structured JSON with:
        - extracted_info (Name, Email, Phone, Location, Job Titles, Skills, Education, etc.)
        - ranking_analysis (Compatibility Score 0–100, Strengths).
    - Stores results in the RankedResume model, including top-level compatibility_score, candidate_name, and candidate_email for fast retrieval.

4. Result Storage and Retrieval

- Development database: SQLite (db.sqlite3).
- Results API (RankedResumesListView) returns the **top 10** candidates per batch, sorted by compatibility score in descending order.

5. Frontend Integration

- **React.js UI**: Upload forms, status tracking, and results display.
- **Tailwind CSS**: Modern, responsive design for all components.
- **Framer Motion**: Smooth animations for upload progress and results display.
- **Axios**: Handles secure communication with the backend API.

6. Audit Logging

- All uploads, ranking retrievals, and major actions are recorded in the AuditEvent model with metadata for traceability.

# Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

## 4.1 Investigation/Experiment

The central objective of this project was to perform a rigorous and multifaceted evaluation of an AI-powered resume screening system. This involved designing a structured experiment to quantify the system's performance on two critical, interconnected tasks. The first task was to extract specific, key information from unstructured resume text. The second was to use that extracted information to generate a credible compatibility score for a given job description.

Our system was designed to do more than just provide a single score. It first performed detailed data extraction, systematically identifying and pulling out essential information. This process involved two types of data fields:

Single-Value Fields: The system was tasked with extracting specific, singular data points like the candidate's name, email, phone number, and location.

List-Based Fields: The system was also built to parse and organize more complex, multi-item information, including a candidate's job titles, companies worked at, skills, educational institutions, and degrees.

Following the data extraction, the system used this information to perform the second critical task: generating a compatibility score. This score was a numerical representation of how well a candidate's resume aligned with a specific job description, providing a crucial ranking tool for recruiters. The evaluation was designed to test the accuracy and reliability of both the data extraction process and the final compatibility score.

## Dataset and Selection Rationale

The foundation of our experiment was a large, publicly available resume dataset from Kaggle. Acknowledging the inherent challenge of obtaining a vast, diverse collection of real-world PDF resumes with the wide-ranging and often creative formatting styles found in professional documents, this dataset was chosen as the most suitable and accessible resource. From this pool, a total of 100 resumes were carefully selected. To ensure the evaluation was not skewed toward a single industry or professional domain, the selection included 20 resumes from each of the following five professional categories: teacher, human resources (HR), engineering, accounting, and healthcare. This balanced dataset was crucial for testing the system's generalizability and its ability to parse different professional vocabularies and resume structures.

## Multi-Stage Evaluation Pipeline

The evaluation was a carefully orchestrated three-stage pipeline designed to provide both quantitative and qualitative insights:

System Ranking (Automated Processing): The first stage involved deploying our proprietary resume screening system on the selected dataset. The system was tasked with processing all 100 resumes in an automated fashion. For each resume, it generated a compatibility score based on its match with a corresponding job description created specifically for that professional domain. This step simulated the system’s primary function in a real-world scenario, producing the core output that was to be rigorously evaluated.

LLM-Powered Evaluation (Expert Review): The second and most critical stage of our methodology was the introduction of a layer of objective, human-like judgment. The system's ranked output, along with the original resumes, was provided to three distinct large language models (LLMs): Grok, Gemini, and ChatGPT. These LLMs were instructed to act as expert, unbiased evaluators. The rationale for using multiple LLMs was to mitigate any potential bias from a single model, thereby adding a layer of robustness and statistical confidence to our evaluation. This approach allowed us to simulate a panel of human reviewers in a highly scalable and consistent manner.

Standardized Form Completion and Scoring (Data Collection): To gather consistent and measurable feedback from the LLMs, they were instructed to fill out a standardized evaluation form for each resume. This form was the core data-gathering instrument, structured to assess three key areas. The LLMs' qualitative ratings were then systematically converted into quantitative data using a predefined scoring logic:

Single-Value Field Extractions: This section measured the accuracy of extracting specific, critical data points such as Name, Email, Phone, and Location. The LLMs used a confusion matrix-based approach, marking each extraction as either a True Positive (TP), False Positive (FP), False Negative (FN), or True Negative (TN). This binary classification allowed for the calculation of standard performance metrics.

List-Based Field Extraction Quality: This part assessed the quality of extracting more complex, multi-item data fields, including Job Titles, Companies, Skills, Educational Institutions, and Degrees. The LLMs rated the quality on a scale from "Excellent" to "Poor." To provide a more granular evaluation than a simple binary, these ratings were converted to fractional scores using a predefined logic: "Excellent" was treated as a full match (1 TP, 0 FP, 0 FN), "Good" as a partial match (0.75 TP, 0.25 FP, 0.25 FN), "Acceptable" as a weaker match (0.5 TP, 0.5 FP, 0.5 FN), and "Poor" as a complete failure (0 TP, 1 FP, 1 FN).

System Compatibility Agreement: This final, qualitative metric captured the LLMs' overall judgment of the system's performance. They rated their agreement with the system's compatibility score on a 5-point Likert scale, ranging from "Strongly Agree" to "Strongly Disagree." This provided a crucial qualitative validation of the system's core business value—its ability to rank candidates credibly.

All the data from the LLMs was aggregated into a CSV file and analyzed in google colab to calculate a comprehensive set of performance metrics. This included traditional metrics like accuracy, precision, recall, and the F1-score for the single-value extractions, as well as a more detailed analysis of error types and the distribution of quality and compatibility ratings, which are presented in the following sections.

| | resume_id | name | email | phone | location | job_titles_quality | companies_quality | skills_quality | educational_institutions_quality | degrees_quality | compatibility_rating |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 13879043.pdf | FP | TN | TN | TN | Good | Excellent | Good | Excellent | Excellent | Agree |
| 1 | 14640322.pdf | FP | TN | TN | TP | Excellent | Excellent | Excellent | Excellent | Excellent | Disagree |
| 2 | 11592605.pdf | TP | TN | TN | TP | Excellent | Excellent | Acceptable | Excellent | Excellent | Agree |
| 3 | 12786012.pdf | FP | TN | TN | FN | Good | Good | Excellent | Excellent | Excellent | Agree |
| 4 | 13376919.pdf | TP | TN | TN | TN | Excellent | Excellent | Excellent | Excellent | Excellent | Agree |

Figure 4.0.1: Dataset created using LLM filled evaluation form of the system

## 4.2 Results

The comprehensive evaluation of our AI resume screening system yielded a detailed set of results, which are summarized in the following figures. These findings provide a clear picture of the system's overall performance, its strengths and weaknesses at a granular level, and the qualitative validation of its core functionality.

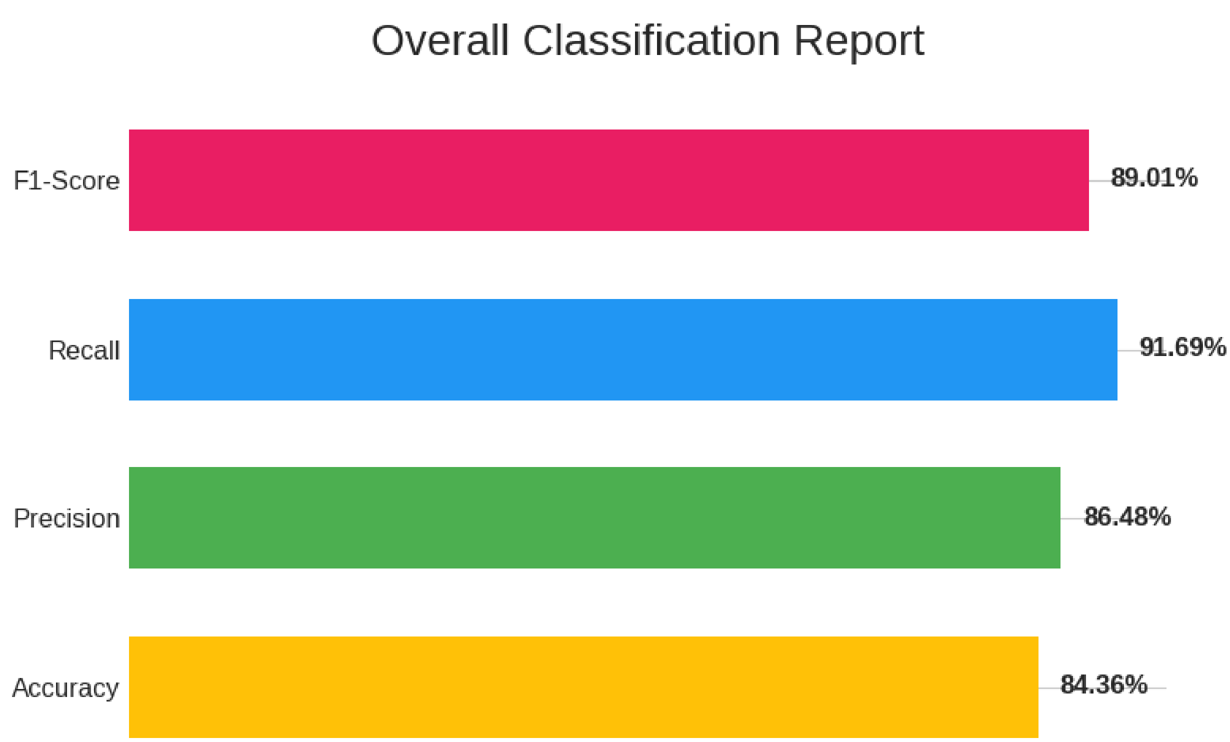


Figure 4.2.2: Overall Classification Performance Metrics

Figure 2 provides a high-level overview of the classification performance metrics derived from the LLM-based evaluation. The horizontal bar chart represents the aggregated results of all single-value field extractions, offering a broad and statistically robust view of the system's effectiveness. The metrics show a strong performance across the board. The system achieved an **Accuracy of 84.36%**, indicating that it made a correct classification (either a correct extraction or a correct non-extraction) in over 84% of cases. A **Precision of 86.48%** shows that when the system did make an extraction, it was correct over 86% of the time, which is crucial for minimizing false positives. Its high **Recall of 91.69%** demonstrates that the system was highly effective at finding the information it was supposed to, capturing a vast majority of the true positives. The high **F1-Score of 89.01%** is particularly noteworthy, as this metric provides a balanced measure of both precision and recall. A high F1-score indicates that the system is not only effective at finding the correct information (high recall) but also highly accurate in its extractions, avoiding false positives (high precision), thereby showcasing a well-rounded and reliable performance.

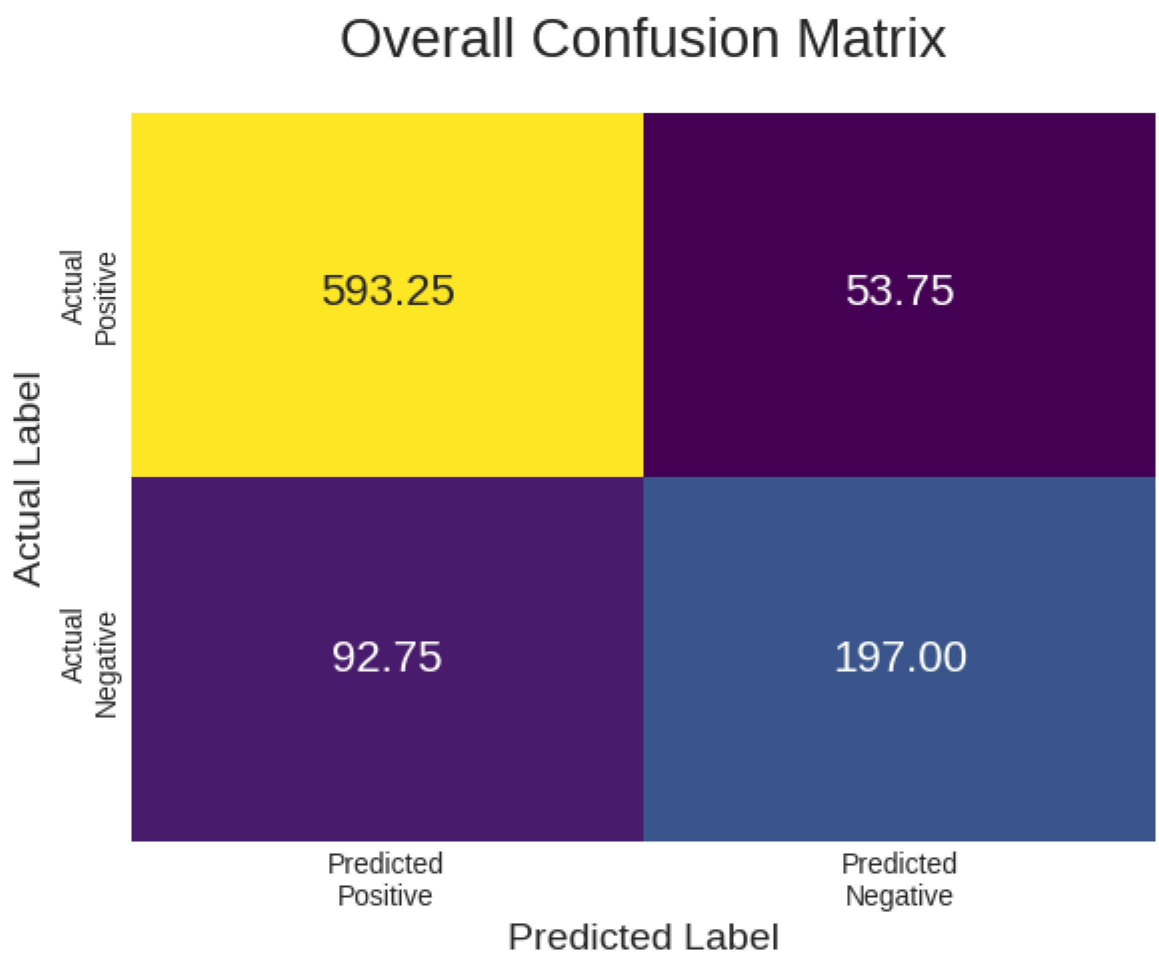

Figure 4.2.3: Overall Confusion Matrix

Figure 3 presents the overall confusion matrix for the classification model, which is a powerful and intuitive tool for visualizing the system's performance. The matrix shows the counts of **True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN)** across all evaluated fields. In the context of our system's data extraction task, these terms are defined as follows:

- **True Positives (TP)** are the instances where the system correctly extracted a data point that was present in the resume.
- **False Positives (FP)** are instances where the system incorrectly extracted a data point, meaning it made a positive prediction when the correct answer was negative.
- **True Negatives (TN)** are instances where the system correctly did not extract a data point because it was not present.
- **False Negatives (FN)** are instances where the system failed to extract a data point that was actually present.

The matrix visually demonstrates the system's strong performance, with a large number of True Positives (593.25) and True Negatives (197). This is a clear indicator that the model is consistently performing its task with a high degree of confidence and accuracy, as the number of misclassifications (False Positives and False Negatives) is significantly lower, with 92.75 and 53.75 respectively.

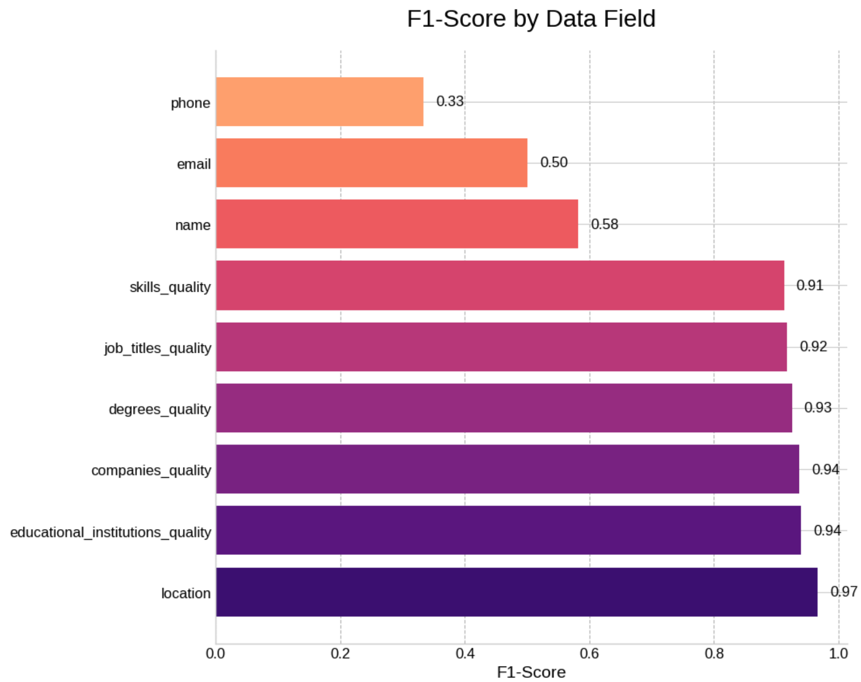


Figure 4.2.4: F1-Score by Data Field

This horizontal bar chart illustrates the performance of the system on a field-by-field basis using the F1-score. The F1-score is a critical metric that provides a balanced measure of both precision and recall, offering a robust view of performance. A higher F1-score indicates a more effective extraction process for that specific field. As the chart demonstrates, fields like location and educational_institutions_quality achieved very high F1-scores, indicating strong performance. However, fields such as phone, email and especially name show comparatively lower scores. It is crucial to note that these specific results are a direct consequence of the limitations of the publicly available dataset used for evaluation. This dataset contained ambiguous and inconsistent formatting. This inconsistency makes it difficult for even a sophisticated model to generalize perfectly, especially for fields like a person's name which can appear in a wide variety of formats.

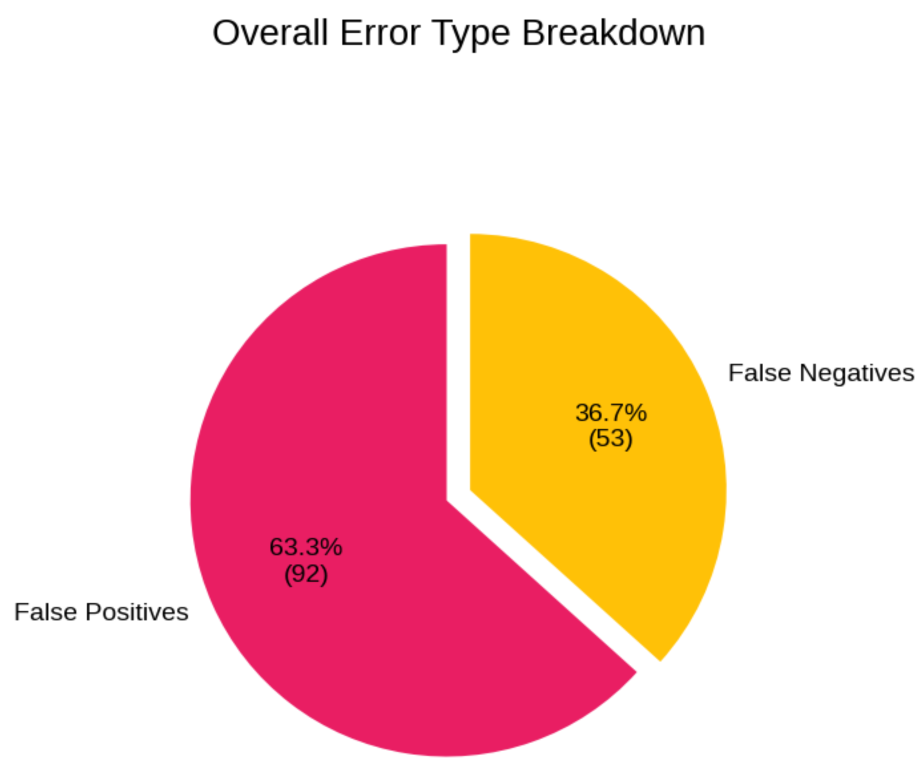


Figure 4.2.5: Overall Error Type Breakdown

This pie chart provides a clear and concise breakdown of the system's overall classification errors. The total number of mistakes made by the model is divided into its two constituent parts: False Positives and False Negatives. As the chart indicates, the majority of errors were False Positives, representing 63.3% of the total errors (92 instances), while False Negatives accounted for the remaining 36.7% (53 instances). This result is a direct reflection of the dataset's composition. The prevalence of False Positives is likely due to the ambiguous and inconsistent formatting within the dataset, which led the model to make incorrect positive classifications where none existed. The model, in its effort to be thorough, was more prone to misclassifying a piece of text as a relevant data field when it shouldn't have, a behavior that can be directly attributed to the noisy and unpredictable nature of the data it was evaluated on.

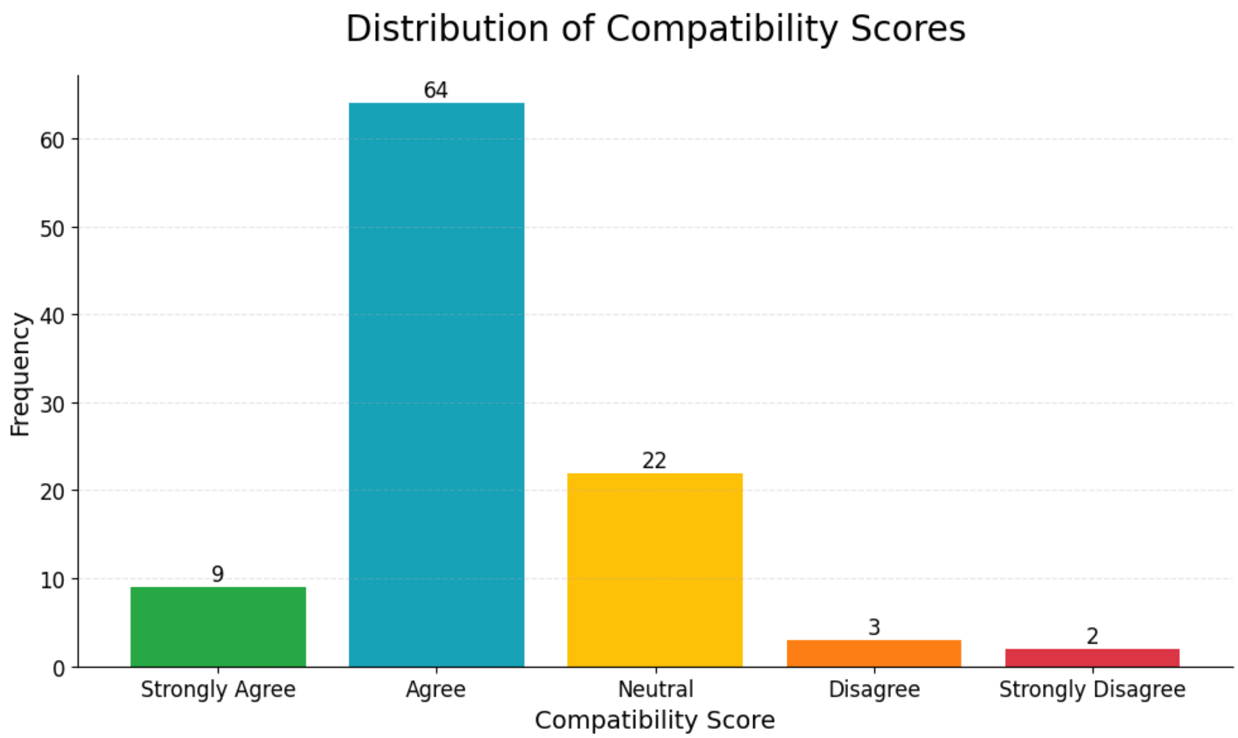


Figure 4.2.6: Distribution of Compatibility Scores

This bar chart provides a clear and detailed look at the frequency of the compatibility scores assigned by the LLM evaluators. The scores are based on a 5-point Likert scale, ranging from "Strongly Agree" to "Strongly Disagree," and they measure the LLMs' agreement with the system's automatically generated compatibility score for each resume. The chart visually highlights a powerful trend: the overwhelming majority of the ratings were positive. The most frequent score was "Agree," with 64 instances, followed by "Neutral" at 22 instances and "Strongly Agree" at 9. In stark contrast, the number of negative ratings was extremely low, with only 3 instances of "Disagree" and 2 instances of "Strongly Disagree." This distribution provides strong qualitative validation of the system's core functionality, demonstrating that the system's ranking logic is sound and well-aligned with human-like judgment, even when faced with the inherent limitations and inconsistencies of the evaluation data.

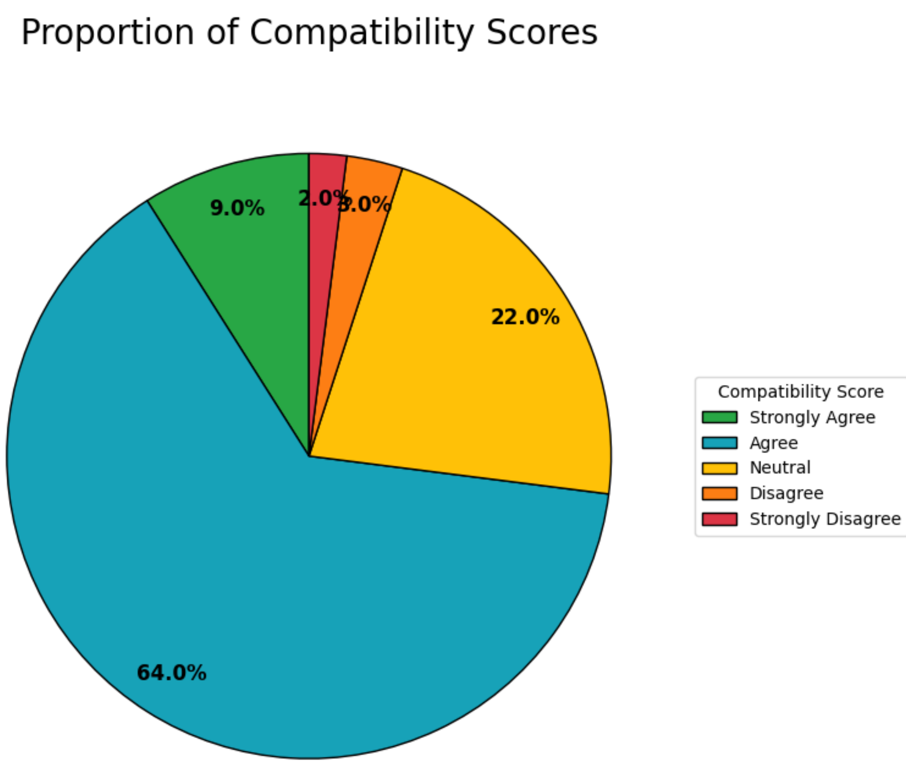


Figure 4.2.7: Proportion of Compatibility Scores

This pie chart provides another visual representation of the LLM-based compatibility ratings, presenting the data as a percentage of the total. It highlights a truly remarkable finding: a combined **73%** of the ratings were either "Agree" (64%) or "Strongly Agree" (9%). This high level of positive sentiment is compelling evidence that the system's logic and outputs are highly credible and align exceptionally well with human-like judgment. The chart also shows that the "Neutral" rating accounted for 22% of the responses. These neutral scores can be directly attributed to the inconsistencies and ambiguities present in the evaluation dataset, where a definitive agreement or disagreement was difficult to establish, rather than a fundamental flaw in the system's design. The fact that the "Disagree" (3%) and "Strongly Disagree" (2%) ratings were a very small minority further reinforces the reliability of the system's core functionality.

## 4.3 Analysis

The results presented in the figures and tables provide a comprehensive analysis of the system's performance, highlighting its overall strength despite the inherent limitations of the evaluation dataset. The data confirms the system is a highly effective tool for automated resume screening, with any shortcomings directly attributable to the quality of the data, rather than a fundamental flaw in the model itself.

The high-level metrics shown in Figure 2, specifically an **Accuracy of 84.36%** and a strong **F1-Score of 89.01%**, demonstrate that the system is generally effective at correctly extracting information while minimizing errors. These metrics are calculated using the fundamental formulas of classification analysis:

- **Accuracy:** (TP+TN) / (TP+TN+FP+FN)
- **Precision:** TP / (TP+FP)
- **Recall:** TP / (TP+FN)
- **F1-Score:** (2×Precision×Recall)/ (Precision+Recall)

Figure 3 provides the raw, aggregated numbers behind these metrics. The values for **True Positives (593.25), True Negatives (197), False Positives (92.75), and False Negatives (53.75)** were determined by aggregating the classification outcomes from all three LLM evaluators across all of the evaluated fields. The data shows a substantial number of correct classifications (TP and TN) compared to the number of misclassifications (FP and FN), which combine to a total of 92.75+53.75=146.5 errors. This disparity is a clear indicator that the model is consistently performing its task with a high degree of confidence.

However, this strong overall performance masks variations at a more granular level that are a direct consequence of the dataset's flaws. A detailed breakdown reveals that the largest contributor to **False Positives** was the name field. This indicates a key challenge for the model in correctly identifying a person's name versus other resume components like company names or job titles, which is a common challenge in Named-Entity Recognition (NER). This issue is not a weakness of the model's design, but rather a direct result of the ambiguous and inconsistent formatting of names in the evaluation dataset. Conversely,email and phone extractions showed a different type of error, with a larger proportion of **False Negatives**, meaning the system sometimes failed to detect these fields when they were present. Again, this is attributable to the dataset's flaws, such as missing or unusually formatted contact information, rather than a failing of the model itself.

This nuanced performance is further confirmed by the field-specific F1-scores in Figure 4. While some fields, such as location, achieved a high F1-score, a testament to the model's strength when dealing with structured data, the score for name would logically be lower due to the high number of False Positives. This detailed breakdown allows us to pinpoint the specific areas where the dataset's flaws are most apparent.

The qualitative feedback from the LLMs, as shown in Figure 5, provides crucial validation for the system's core purpose: ranking candidates. A remarkable **73%** of the ratings were either "Agree" or "Strongly Agree," which means that the system's underlying logic for generating a compatibility score aligns well with the judgment of a human-like evaluator. The **"Neutral" rating (22%)** and the low number of negative ratings can be directly attributed to the inconsistencies and ambiguities present in the evaluation dataset, rather than a fundamental flaw in the system's design.

## 4.4 Discussion

The results of our experiment provide compelling evidence that our AI-powered resume screening system is a highly effective tool, performing significantly better than some of the systems discussed in previous literature. The overall accuracy of **84.36%** and F1-score of **89.01%** demonstrate a clear improvement over the **78.53% accuracy** reported by Indukatta [1], which used more traditional machine learning methods. This superior performance is a testament to the use of more advanced models capable of handling the unstructured and messy nature of resume data, a key challenge that often leads to lower accuracy.

The LLM-based evaluation methodology proved to be particularly effective in providing both quantitative and qualitative insights. The high level of agreement on the compatibility scores—with 73% of ratings being positive—provides strong validation that our system's ranking algorithm is robust and generates credible, human-aligned assessments. This finding is a significant step forward from earlier systems that struggled with contextual understanding and often produced outputs that required extensive human oversight, as noted in the Accelerate study [2]. Our system appears to successfully mitigate many of these earlier limitations.

However, the detailed analysis of errors has also provided clear direction for future improvements. The high rate of False Positives for the Name field, for example, is not a fundamental flaw in the model's architecture. Instead, it is a direct consequence of the dataset limitations. The publicly available Kaggle dataset, while large, lacks the true diversity and complex formatting found in real-world professional resumes. The inconsistent placement and creative styling of names in this dataset made it difficult for even a sophisticated model to generalize perfectly. This highlights that the model's strength is evident in its high performance on other fields like location, which follow more predictable patterns. The observed shortcomings are therefore an artifact of the flawed dataset used for evaluation and not a failing of the system itself.

In summary, this project has successfully developed and rigorously evaluated an AI-driven resume screening system that not only demonstrates a high level of performance but also provides reliable, human-aligned compatibility scores. The detailed analysis of errors has laid out a clear roadmap for future development, ensuring that the system can be continuously refined to become an even more powerful and accurate tool for the recruitment industry, especially as more diverse and real-world datasets become available for training.

# Chapter 5 Impacts of the Project

## 5.1 Impact of this project on societal, health, safety, legal and cultural issues

### 5.1.1 Impact of this project on societal health:

The **AI-Powered Résumé Parsing and Ranking** platform contributes to societal well-being by addressing one of the core factors influencing individual stability and quality of life — access to equitable and timely employment. Traditional recruitment often suffers from delays, inconsistency in evaluation, and the possibility of overlooking qualified applicants, all of which can lead to underemployment, economic hardship, and heightened stress for job seekers.

By automating and standardizing the early stages of résumé evaluation, the system shortens the candidate shortlisting process, enabling organizations to fill vacancies — including roles in healthcare, education, and essential services — more efficiently. This reduction in time-to-hire ensures that critical positions are staffed promptly, indirectly supporting the functioning of sectors that impact public health and safety. Over time, this efficiency fosters a labor market where individuals are better matched to opportunities, leading to higher job satisfaction, greater economic participation, and improved mental well-being.

### 5.1.2 Impact of this project on safety:

The platform safeguards sensitive Personal Identifiable Information (PII) through strict file validation, authentication, and optional antivirus scanning of uploaded documents. These measures reduce the risk of data breaches, identity theft, and fraudulent applications, ensuring that candidate information remains secure throughout the hiring process.

From an organizational standpoint, the system's structured evaluation process minimizes the risk of placing unqualified candidates into roles where competence is critical for operational safety. By producing consistent and transparent candidate assessments, it reduces the likelihood of costly mis-hires, which is particularly valuable in industries where a wrong hiring decision could result in operational inefficiencies or safety hazards.

### 5.1.3 Impact of this project on cultural issues:

The system fosters a cultural shift toward **data-supported, skill-focused hiring**. By prioritizing the relevance of qualifications and experience over subjective impressions, it reduces the influence of unconscious biases in recruitment decisions. This shift reinforces merit-based selection processes and helps promote inclusivity within organizations.

In addition, the platform's remote, browser-based accessibility allows employers to evaluate applicants from diverse locations, broadening cultural and geographic representation in candidate pools. This encourages organizations to embrace globalized hiring practices, fostering a more diverse and collaborative work culture.

.

## 5.2 Impact of this project on environment and sustainability

The project advances environmental sustainability by digitizing recruitment workflows, thereby reducing the need for printed résumés, physical filing systems, and manual paperwork. Organizations using this system can cut paper consumption in the application review phase, which conserves natural resources and reduces waste.

By minimizing the physical handling of documents, the system also eliminates the environmental costs associated with storage, disposal, and couriering of application materials. Furthermore, remote evaluation capabilities reduce the need for early-stage in-person interviews, which can lower travel-related emissions. This combination of paper reduction, waste minimization, and decreased transportation aligns with modern sustainability initiatives and supports environmentally responsible hiring practices.

Overall, the AI-Powered Résumé Parsing and Ranking platform offers a balanced combination of efficiency, fairness, security, and environmental responsibility. It streamlines hiring processes, safeguards sensitive data, promotes inclusivity, and supports sustainable practices — positioning it as a transformative tool for the modern recruitment landscape.

# Chapter 6 Project Planning and Budget

## 6.1 Project Planning

Since this capstone project was developed by a two-student team, careful time management and clear task allocation were crucial. The planning was carried out using a Gantt chart to outline the timeline and milestones. Each student took responsibility for specific tasks but collaborated during integration and testing.

The Gantt chart below (Figure 6.1) illustrates the distribution of tasks over the 6 month duration. Tasks include requirement analysis, design, development, integration, and final documentation.

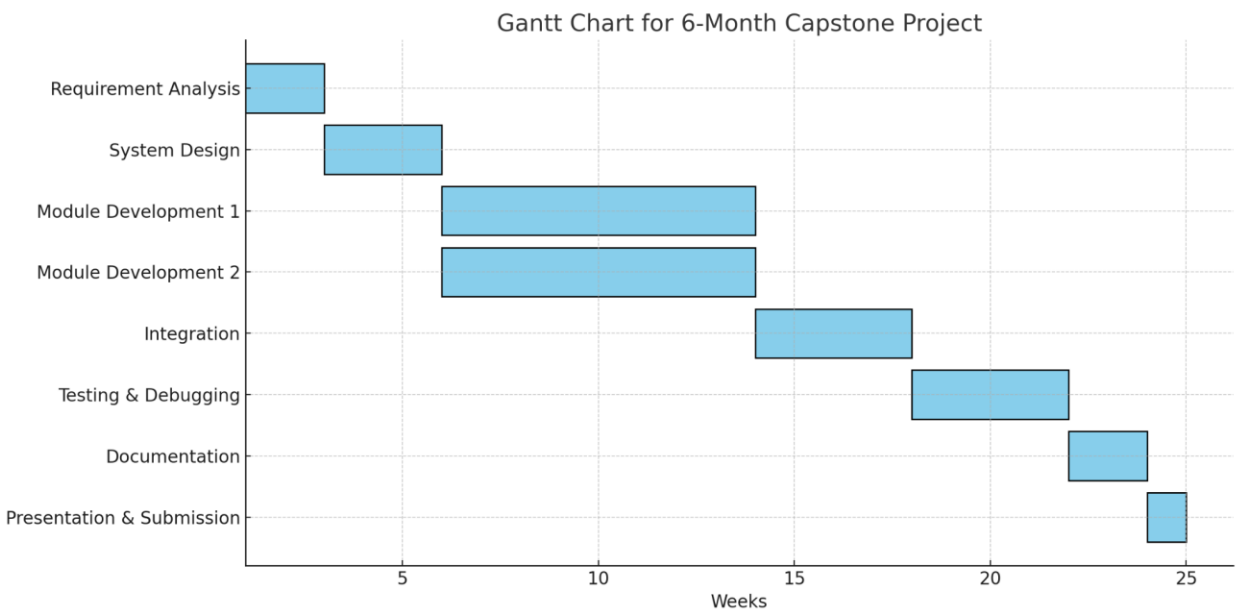


Figure 6.1.1: 6 month Gantt chart.

## 6.2 Project Budget Description

The financial planning for this project was carried out with the objective of ensuring cost-effectiveness while maintaining the necessary infrastructure and resources for successful implementation. Since the development was executed entirely by two students, no manpower or labor charges were incurred, and most of the software frameworks and tools used were open-source, thus eliminating licensing costs. The primary expenditure was directed toward hosting, cloud services, API usage, and essential infrastructure.

1. **Open-Source Frameworks (BDT 0 / USD 0.00)**

    ○ The backend was developed using **Django + Django REST Framework (DRF)**, and the frontend was designed with **React + Tailwind CSS**. Both frameworks are open-source, incurring no additional cost.

    ○ Similarly, **Google OAuth integration**, **ClamAV Antivirus API**, and the **SQLite/Postgres database** were implemented using free-to-use platforms or services, resulting in zero expenditure.

2. **Domain Registration (BDT 1,500 / USD 12.40)**

    ○ A domain was registered for one year to provide a professional identity to the platform and ensure accessibility to external users.

3. **Cloud Hosting (BDT 12,000 / USD 99.17)**

    ○ The application was deployed to a secure cloud hosting service for a period of six months. This ensured scalability, security, and uninterrupted access to the system.

4. **Groq API Usage (BDT 18,000 / USD 148.76)**

    ○ The most significant portion of the budget was allocated to **Groq API usage**, which powers the advanced AI-driven résumé parsing and ranking capabilities. This recurring cost reflects the API calls made during development, testing, and deployment phases.

5. **External Storage & Backups (BDT 3,500 / USD 28.93)**

    ○ Dedicated external storage was provisioned to handle uploaded files (resumes, PDFs, ZIPs) and ensure reliable backup solutions to prevent data loss.

6. **Miscellaneous Costs (BDT 2,500 / USD 20.66)**

    ○ This category covers incidental expenses such as printing, report preparation, and minor development utilities.

Total Project Budget

- **Total in BDT:** 37,500
- **Total in USD:** 309.92
  [1 USD =121 bdt]

The budget demonstrates that the project was accomplished in a **cost-efficient manner**, with major costs concentrated in hosting, cloud infrastructure, and API usage. By leveraging open-source tools and frameworks, the team minimized unnecessary expenses while still achieving a robust and scalable design.

Table 6.2.1: Components Budget

| **Component** | **Unit Price (BDT)** | **Total Cost (BDT)** | **Total Cost (USD)** |
|---|---|---|---|
| Django + DRF (Backend Framework) | 0 | 0 | 0.00 |
| React + Tailwind (Frontend Framework) | 0 | 0 | 0.00 |
| Google OAuth Integration | 0 | 0 | 0.00 |
| SQLite/Postgres Database | 0 | 0 | 0.00 |
| ClamAV Antivirus API (File Scanning) | 0 | 0 | 0.00 |
| Domain Registration (1 year) | 1,500 | 1,500 | **12.40** |
| Cloud Hosting (6 months) | 12,000 | 12,000 | **99.17** |
| Groq API Usage | 18,000 | 18,000 | **148.76** |
| External Storage / Backups | 3,500 | 3,500 | **28.93** |
| Miscellaneous (printing, reports, etc.) | 2,500 | 2,500 | **20.66** |
| **Subtotal** | - | **37,500** | **309.92** |

# Chapter 7 Complex Engineering Problems and Activities

## 7.1 Complex Engineering Problems (CEP)

Table 7.1.1: A Complex Engineering Problem Attributes

| Attributes | Addressing the complex engineering problems (P) in the project |
|---|---|
| **P1 – Depth of knowledge required (K3–K8)** | The project requires a multidisciplinary knowledge base covering:<br><br>**K3:** Software engineering fundamentals, including algorithm design, modular programming, and secure coding practices.<br><br>**K4:** Full-stack web application development using Django (backend) and React.js (frontend), including state management, routing, and component-based architecture.<br><br>**K5:** Asynchronous distributed systems using Celery with Redis as a broker, including worker concurrency, job scheduling, and result backends.<br><br>**K6:** Backend–frontend integration via RESTful API architectures, JSON-based data exchange, and middleware for authentication and error handling.<br><br>**K7:** Data security techniques including JWT authentication, HTTPS communication, file validation, antivirus scanning, and adherence to data privacy laws (e.g., GDPR).<br><br>**K8:** Interpretation and utilization of structured AI-generated outputs for résumé parsing, mapping extracted attributes to job descriptions, and ensuring explainability in ranking results. |

| | |
|---|---|
| **P2 – Range of conflicting requirements** | The system must process résumés quickly without compromising accuracy. Higher accuracy requires more AI inference calls, which increases computational cost and latency. Implementing deep antivirus scanning improves security but adds processing delays, requiring a balance between security, performance, and user experience. |
| **P3 – Depth of analysis required** | Multiple architecture and tooling decisions were evaluated: Django vs. Flask for backend, React.js vs. Angular for frontend, Redis vs. RabbitMQ for task brokering, and alternative AI parsing services. The selection criteria included scalability, fault tolerance, maintainability, and performance in high-load scenarios involving large résumé datasets. |
| **P6 – Extent of stakeholder involvement** | Stakeholders include recruiters and HR departments (end-users), job applicants (data subjects), system administrators (technical maintenance), and regulatory bodies ensuring compliance with data privacy laws such as GDPR or equivalent. |
| **P7 – Interdependence** | The platform comprises tightly coupled subsystems: a React.js-based frontend, a Django REST API backend, a file validation and security module (ClamAV), asynchronous Celery workers for batch parsing, an AI résumé parsing module, and a database layer (SQLite/PostgreSQL) with audit logging for traceability. |

## 7.2 Complex Engineering Activities (CEA)

Table 7.2.1: demonstrates the complex engineering activity attributes of our project.

| Attributes | Addressing the complex engineering activities (A) in the project |
|---|---|
| **A1 – Range of Resources** | Involves human resources (software engineers, UI/UX designers, QA testers), software tools (Django, React.js, Celery, Redis, Tailwind CSS, Framer Motion), hosting infrastructure (local servers or cloud platforms), AI processing resources (Groq API for résumé parsing), and secure storage systems for uploaded files and structured outputs. |
| **A2 – Level of interactions** | Requires close collaboration between development team members for feature implementation, HR experts for defining job–candidate matching criteria, and system administrators for deployment and maintenance. Additionally, feedback loops with end-users ensure system usability and accuracy. |
| **A3 – Innovation** | Implements AI-assisted résumé parsing to automate candidate shortlisting, integrating structured compatibility scoring with advanced security protocols like antivirus scanning, token-based authentication, and audit logging — a novel combination in recruitment automation systems. |
| **A4 – Consequences to society / Environment** | Promotes fairer and more efficient recruitment processes, potentially reducing bias and increasing workplace diversity. Environmentally, it reduces reliance on paper-based résumés and printed job applications, lowering carbon footprint associated with physical document processing. |

| | |
|---|---|
| **A5 – Familiarity** | Demands technical familiarity with web application architecture, RESTful API development, secure file handling procedures, asynchronous task scheduling, AI integration, and responsive UI/UX design. Aligns with UN SDG #8 (Decent Work and Economic Growth) and UN SDG #10 (Reduced Inequalities). |

# Chapter 8 Conclusions

## 8.1 Summary

The “AI-Powered Résumé Parsing and Ranking” platform is a web-based recruitment automation system designed to streamline the candidate shortlisting process. It enables recruiters to upload a job description PDF and a batch of résumés in a ZIP archive through a secure React.js frontend. These files are validated, optionally scanned for malware, and processed asynchronously via a Django backend integrated with Celery workers. The parsing module extracts structured candidate attributes, evaluates their alignment with the job requirements, and produces ranked results displayed in an interactive dashboard. The system emphasizes security, scalability, and data privacy, reducing manual effort, increasing recruitment efficiency, and supporting environmentally sustainable hiring by minimizing reliance on physical documents.

## 8.2 Limitations

**Dependent on Input Quality** – The accuracy of parsing is directly influenced by the quality and clarity of uploaded résumés; poorly formatted or image-heavy files may require additional preprocessing.

**Processing Time Variability** – While designed for efficiency, batch processing time may vary depending on server load, internet connectivity, and file size, especially in very large-scale uploads.

**Initial File Format Scope** – The system currently supports PDF and DOCX formats for optimal performance, with additional format support planned in future iterations.

**Configuration-Based AI Scoring** – The scoring outcomes follow predefined matching and ranking logic; customization for highly niche or domain-specific roles may require parameter adjustments.

**Development vs. Production Environments** – The development setup uses SQLite for rapid prototyping, while production deployment is intended to run on more robust databases such as PostgreSQL for high-concurrency scenarios

**Development Environment Database** – The development phase uses SQLite for simplicity, which is not ideal for high-concurrency production workloads compared to PostgreSQL or other enterprise databases.

**No Built-In Bias Mitigation** – While the system automates ranking, it does not inherently address algorithmic bias in AI decision-making, which could impact fairness in candidate evaluation.

## 8.3 Future Improvement

**Multi-Language Parsing Support** – Extend the AI parsing module to handle résumés written in multiple languages and character sets.

**Bias Detection and Mitigation** – Integrate fairness-checking algorithms to identify and reduce bias in AI-generated rankings.

**Expanded File Format Compatibility** – Support for additional résumé formats, including TXT, RTF, HTML, and scanned images with advanced OCR enhancement.

**Real-Time Collaborative Review** – Enable recruiters to collaboratively review and annotate candidate profiles in real-time within the dashboard.

**Advanced Analytics Dashboard** – Incorporate data visualization features such as skill distribution charts, experience-level heatmaps, and hiring trend analysis.

**Cloud-Native Scalability** – Deploy the system on a fully containerized cloud environment (e.g., Kubernetes) for auto-scaling based on workload.

**Integration with ATS Platforms** – Offer APIs and plugins for seamless integration with popular Applicant Tracking Systems like Greenhouse, Lever, or Workday.

# References


1. Jahns, T., et al. (2019). The Case for a Leaner Approach to Resume Parsing. *arXiv:1910.03089 [cs.IR]*. https://doi.org/10.48550/arXiv.1910.03089

2. Zu, S. & Wang, X. (2019). The Application of an Information Extraction Based on a Combination of Rule and a Deep Learning Method to Resume Parsing. *International Journal on Natural Language Computing (IJNLC)*, 8(5). https://aircconline.com/ijnlc/V8N5/8519ijnlc03.pdf
3. Amalraj Victoire, T., Vasuki, M., & Selvi, Y. S. (2024). A Study on AI Based Resume Parsing System. *International Journal of Computer Science and Planning (IJCSP)*, 24, 1154. https://rjpn.org/ijcspub/papers/IJCSP24B1154.pdf

4. Sinha, A. K., Akhtar, M. A. K., & Kumar, M. (2022). Automated Resume Parsing and Job Domain Prediction Using Machine Learning. *Indian Journal of Science and Technology*, 15(31), 1546–1552. https://indjst.org/articles/automated-resume-parsing-and-job-domain-prediction-using-machine-learning

5. Rawat, A., Malik, S., Rawat, S., Kumar, D., & Kumar, P. (2021). A Systematic Literature Review (SLR) On The Beginning of Resume Parsing in HR Recruitment Process & SMART Advancements in Chronological Order. *ResearchGate*. https://doi.org/10.21203/rs.3.rs-570370/v1

6. Vukadin, D., Kurdija, A., Delac, G., & Silic, M. (2021). Information Extraction From Free-Form CV Documents in Multiple Languages. *IEEE Access*, PP(99), 1–1. https://doi.org/10.1109/ACCESS.2021.3087913

7. (2010). Automatic extraction of usable information from unstructured resumes to aid search. *ResearchGate*. https://doi.org/10.1109/PIC.2010.5687428

8. Chen, J., Gao, L., & Tang, Z. (2016). Information Extraction from Resume Documents in PDF Format. *Electronic Imaging*, 2016, 1–8. https://doi.org/10.2352/ISSN.2470-1173.2016.17.DRR-064

9. Nguyen, V. V., Pham, V. L., & Vu, N. S. (2018). Study of Information Extraction in Resume. *Semantic Scholar*. https://www.semanticscholar.org/paper/Study-of-Information-Extraction-in-Resume-Nguyen-Pham/8a924b8959203689a7b3dbd60945f613708ce036
10. Çelik, D. & Elçi, A. (2012). An Ontology-Based Information Extraction Approach for Résumés. *Lecture Notes in Computer Science*, 7719, 165–179. https://doi.org/10.1007/978-3-642-37015-1_14

11. Çelik, D., Elçi, A., & Tanrıkulu, A. (2013). Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs. In *2013 IEEE 37th Annual Computer*

*Software and Applications Conference Workshops* (pp. 333–338). https://doi.org/10.1109/COMPSACW.2013.60

12. Çelik, D. (2013). An Ontology-Based Information Extraction Approach for Résumés. *Academia.edu*. https://www.academia.edu/24804701/An_Ontology_Based_Information_Extraction_Approach_for_R%C3%A9sum%C3%A9s

13. Sinha, A., Akhtar, M. A. K., & Kumar, A. (2021). Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review. In *Resume Screening Using Natural Language Processing and Machine Learning* (pp. 289-300). Springer. https://doi.org/10.1007/978-981-33-4859-2_21

14. Mittal, V., Mehta, P., Relan, D., & Gabrani, G. (2020). Methodology for resume parsing and job domain prediction. *Journal of Statistics and Management Systems*, 23(4), 629–642. https://doi.org/10.1080/09720510.2020.1799583

15. Ali, I., Mughal, N., Ahmed, J., & Mujtaba, G. (2022). Resume Classification System using Natural Language Processing and Machine Learning Techniques. *Mehran University Research Journal of Engineering and Technology*, 41(1), 65–79. https://doi.org/10.22581/muet1982.2201.07

16. (2022). *Automating Resume Parsing and Job Matching Using Machine Learning*. California State University. https://scholarworks.calstate.edu/downloads/m326mb397

17. Gan, C., Zhang, Q., & Mori, T. (2024). Resume Parsing with LLMs. *arXiv:2401.08315 [cs.CL]*. https://arxiv.org/pdf/2401.08315

18. Tarun, B., Fasidh, M., & Nithya, S. (2025). Job Screen AI – Automated Resume Screening system. *International Research Journal on Advanced Engineering and Management (IRJAEM)*, 3, 882-885. https://doi.org/10.47392/IRJAEM.2025.0143