##### A

##### Industrial-Oriented Mini Project On

**HIRE SMART – NLP POWERED RESUME SCREENING**

**AND FILTERING**

(Submitted in partial fulfilment of the requirements for the award of Degree)

**BACHELOR OF TECHNOLOGY**

##### In

**COMPUTER SCIENCE AND ENGINEERING**

##### By

##### Ch. Tejasree (227R1A05D8)

##### Under the Guidance of

#### Mr. G. VINESH SHANKER

(Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

###### UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGC Act.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**June, 2025.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

**CERTIFICATE**

This is to certify that the project entitled “**HIRE SMART – NLP POWERED RESUME SCREENING AND FILTERING**” being submitted by Ch. Tejasree (227R1A05D8) in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. G. Vinesh Shanker Dr. Nuthanakanti Bhaskar Assistant Professor HoD

INTERNAL GUIDE

Dr. A. Raji Reddy Signature of External Examiner DIRECTOR

Submitted for viva voice Examination held on

**ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project, I take this opportunity to express my profound gratitude and deep regard to my guide **Mr. G. Vinesh Shanker,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which I am about to embark.

I take this opportunity to extend my heartfelt appreciation to the Project Review Committee (PRC) Coordinators—**Y Varalaxmi, B Sekhar, D Nageswar Rao, and SVSV Prasad Sanaboina**—for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

My sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of my project.

I am deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, I extend my profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to my progress.

I also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, I sincerely thank my families for their unwavering support and encouragement. I also extend my gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

Ch. Tejasree (227R1A05D8)

**VISION AND MISSION**

**INSTITUTE VISION:**

##### To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

**INSTITUTE MISSION:**

##### To create state of art facilities for effective Teaching- Learning Process.

##### Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.

##### Infuse Professional, Ethical and Societal values among Learning Community.

**DEPARTMENT VISION:**

##### To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

**DEPARTMENT MISSION:**

##### To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.

##### To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.

##### To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

**ABSTRACT**

In the era of digital recruitment, organizations receive thousands of resumes for a single job posting, making manual screening both Labor- intensive and inefficient. "Hire Smart" is a robust, AI-driven solution that leverages Natural Language Processing (NLP) to automate and enhance the resume screening and filtering process. This system aims to minimize recruiter workload, reduce human biases, and improve the accuracy and speed of candidate shortlisting. The core functionality of "Hire Smart" is based on advanced NLP techniques such as named entity recognition (NER), part-of-speech tagging, semantic similarity analysis, and keyword extraction. Resumes in various formats (PDF, DOCX, TXT) are parsed and converted into structured data. The system then compares the extracted data against job descriptions using vector embeddings and similarity scoring algorithms to assess candidate-job fit. Moreover, "Hire Smart" includes features such as customizable ranking criteria (e.g., years of experience, technical skills, certifications), automated flagging of irrelevant or mismatched profiles, and dynamic filtering based on recruiter-defined thresholds. It can also identify and prioritize soft skills and achievements using sentiment and context analysis. By integrating with applicant tracking systems (ATS) and supporting API endpoints, "Hire Smart" ensures seamless integration into existing recruitment pipelines. It supports continuous learning via feedback loops that refine the model’s accuracy over time, adapting to industry-specific terminology and recruiter preferences. Overall, "Hire Smart" transforms the recruitment workflow into a faster, smarter, and more objective process. It empowers HR professionals to focus on strategic decision-making rather than manual screening, leading to better hiring outcomes and a stronger workforce.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | Project Architecture of HIRE SMART – NLP Powered Resume Screening and Filtering | 16 |
| Figure 3.3 | Dataflow Diagram of HIRE SMART – NLP  Powered Resume Screening and Filtering | 21 |
| Figure 5.1 | Home Page | 27 |
| Figure 5.2 | About Page | 28 |
| Figure 5.3 | Dashboard | 28 |
| Figure 5.4 | Upload Page | 29 |
| Figure 5.5 | Job Description Upload | 29 |
| Figure 5.6 | Uploading Files | 30 |
| Figure 5.7 | Results Page | 30 |
| Figure 5.8 | Top Resumes | 31 |
| Figure 5.9 | Contact Page | 31 |
| Figure 5.10 | Help Page | 32 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| Table 2.6.1 | Hardware Requirements | 14 |
| Table 2.6.2 | Software Requirements | 15 |
| Table 6.2.1 | Resume Upload and Parsing | 34 |
| Table 6.2.2 | Job Description Input | 34 |
| Table 6.2.3 | Resume Matching & Similarity Scoring | 35 |
| Table 6.2.4 | Candidate Ranking | 35 |
| Table 6.2.5 | Feedback and Learning Loop | 35 |
| Table 6.2.6 | System Performance and Edge Cases | 36 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** | **i** |
|  | **LIST OF FIGURES** | **ii** |
|  | **LIST OF TABLES** | **iii** |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 PROJECT PURPOSE | 1 |
|  | 1.2 PROJECT FEATURES | 2 |
| **2.** | **LITERATURE SURVEY** | 3 |
|  | 2.1 REVIEW OF RELATED WORK | 6 |
|  | 2.2 DEFINITION OF PROBLEM STATEMENT | 8 |
|  | 2.3 EXISTING SYSTEM | 9 |
|  | 2.4 PROPOSED SYSTEM | 11 |
|  | 2.5 OBJECTIVES | 13 |
|  | 2.6 HARDWARE & SOFTWARE REQUIREMENTS | 14 |
|  | 2.6.1 HARDWARE REQUIREMENTS | 14 |
|  | 2.6.2 SOFTWARE REQUIREMENTS | 15 |
| **3.** | **SYSTEM ARCHITECTURE AND DESIGN** | 16 |
|  | 3.1 PROJECT ARCHITECTURE | 16 |
|  | 3.2 DESCRIPTION | 17 |
|  | 3.3 DATA FLOW DIAGRAM | 21 |
| **4.** | **IMPLEMENTATION** | 22 |
|  | 4.1 ALGORITHMS USED | 22 |
|  | 4.2 SAMPLE CODE | 24 |
| **5.** | **RESULTS & DISCUSSION** | 27 |
| **6.** | **VALIDATION** | 33 |
|  | 6.1 INTRODUCTION | 33 |
|  | 6.2 TEST CASES | 34 |
|  | 6.2.1 Resume Upload and Parsing | 34 |
|  | 6.2.2 Job Description Input | 34 |
|  | 6.2.3 Resume Matching & Similarity Scoring | 35 |
|  | 6.2.4 Candidate Ranking | 35 |
|  | 6.2.5 Feedback and Learning Loop | 35 |
|  | 6.2.6 System Performance and Edge Cases | 36 |
| **7.** | **CONCLUSION & FUTURE ASPECTS** | 37 |
|  | 7.1 PROJECT CONCLUSION | 37 |
|  | 7.2 FUTURE ASPECTS | 38 |
| **8.** | **BIBLIOGRAPHY** | 39 |
|  | 8.1 REFERENCES | 39 |
|  | 8.2 GITHUB LINK | 41 |

1. **INTRODUCTION**

**1. INTRODUCTION**

Here's a concise and professional introduction for **"HIRE SMART – NLP Powered Resume Screening and Filtering"**:

In today's competitive job market, recruiters often face the overwhelming task of reviewing hundreds or even thousands of resumes for a single job opening. Traditional manual screening processes are time-consuming, prone to bias, and often ineffective in identifying the best-fit candidates. **HIRE SMART** addresses these challenges by leveraging the power of **Natural Language Processing (NLP)** to automate and enhance the resume screening and filtering process.

**HIRE SMART** is an intelligent, NLP-driven system designed to analysis, understand, and rank resumes based on job-specific requirements. By using advanced algorithms for text parsing, semantic analysis, and machine learning, the platform can efficiently extract key information, evaluate candidate suitability, and highlight the most qualified individuals. This not only saves time for hiring teams but also ensures a more consistent and objective evaluation process.

With **HIRE SMART**, companies can streamline their recruitment workflow, reduce hiring costs, and significantly improve the quality of their hires—all through the smart application of cutting-edge AI and NLP technologies.

* 1. **PROJECT PURPOSE**

The purpose of **HIRE SMART** is to revolutionize the recruitment process by automating the initial stages of candidate evaluation using Natural Language Processing (NLP). The primary goals are:

* **To streamline resume screening** by automatically extracting and analyzing key information from resumes such as skills, experience, education, and achievements.
* **To reduce human bias and error** in the shortlisting process by using objective, data-driven decision-making.
* **To save time and resources** for HR teams by minimizing manual effort.
  1. **PROJECT FEATURES**

**HIRE SMART** is equipped with a range of powerful features designed to optimize and automate the resume screening process using Natural Language Processing:

* **Resume Parsing**  
  Automatically extracts structured information from resumes, including contact details, skills, education, and work history.
* **Semantic Job Matching**  
  Utilizes NLP to understand the context of both resumes and job descriptions, enabling more accurate and meaningful candidate matching.
* **Candidate Scoring and Ranking**  
  Assigns a relevance score to each resume based on how closely it aligns with the job requirements, making it easier to identify top candidates.
* **Skill and Keyword Extraction**  
  Highlights important skills, tools, certifications, and domain-specific terms present in the resumes.
* **Bias-Free Screening**  
  Removes personal identifiers like names, gender, and photos during initial screening to support unbiased decision-making.
* **Automated Filtering**  
  Automatically filters out resumes that don’t meet the basic qualifications or are duplicated, improving efficiency.
* **Customizable Screening Rules**  
  Gives recruiters the flexibility to define and adjust filtering rules according to specific job roles and company standards.
* **Analytics Dashboard**  
  Provides visual insights and metrics on the hiring process, helping teams evaluate the quality of candidates and screening performance.

**2. LITERATURE SURVEY**

**2. LITERATURE SURVEY**

The recruitment process is a critical function for any organization aiming to acquire top talent and maintain competitive advantage. However, traditional hiring methods, particularly resume screening, often involve time-consuming, manual procedures that are vulnerable to human error, inefficiency, and bias. With the growing number of job applicants, especially through online portals, recruiters face significant challenges in filtering and evaluating resumes quickly and fairly. To address these issues, research and development have shifted toward the application of artificial intelligence (AI), and more specifically, natural language processing (NLP), in the recruitment domain.

**Natural Language Processing in Recruitment**

Natural Language Processing, a subset of artificial intelligence, deals with the interaction between computers and human languages. In the context of resume screening, NLP is used to interpret and analyze the unstructured text data present in resumes and job descriptions. Several studies have demonstrated the effectiveness of NLP in extracting structured information from resumes. Techniques such as tokenization, stemming, lemmatization, part-of-speech tagging, and named entity recognition (NER) help in identifying key attributes like candidate name, skills, work experience, educational qualifications, and certifications.

Furthermore, NLP-based tools are capable of performing sentiment analysis, summarization, and semantic analysis, which assist recruiters in better understanding a candidate's background and suitability. Research by Jain et al. (2020) emphasized that NLP models significantly reduce the time spent on initial screening without compromising quality. NLP also enhances the standard keyword-matching process by analyzing context, synonyms, and the semantic relevance of words, thereby providing a more meaningful match between resumes and job requirements.

**Semantic Matching and Embedding Models**

Traditional keyword-based resume screening tools often miss candidates who use alternative terminology or phrasing. To overcome this, semantic matching techniques have been introduced. Word embeddings like **Word2Vec**, **Glo Ve**, and **Fast text** convert words into vector representations that capture semantic similarity. More recent models, such as

**BERT (Bidirectional Encoder Representations from Transformers)** and **RoBERTa**, have achieved state-of-the-art results in understanding contextual relationships in text.

These models are trained on large corpora and are capable of encoding entire sentences or documents into vector representations. When applied to resume-job matching, such models can semantically compare a candidate’s experience and skills with job requirements, significantly improving the quality of matches. Research by Zhang et al. (2021) showed that semantic search models consistently outperformed traditional information retrieval models in recruitment applications.

**Bias and Fairness in Resume Screening**

A major concern in recruitment is unconscious bias, which can stem from a candidate’s name, gender, age, ethnicity, or even the formatting of their resume. A well-known study by Bertrand and Mullainathan (2004) demonstrated that resumes with "White-sounding" names received significantly more callbacks than identical ones with "African-American-sounding" names. As a result, there has been a surge of interest in using AI to reduce such bias.

Several AI systems now incorporate bias-mitigation strategies, such as anonymizing resumes by removing personal identifiers before screening. Others include fairness-aware algorithms that balance candidate selection across different demographic groups. However, caution is necessary, as AI models can also inherit biases from training data if not properly designed and monitored.

**Applicant Tracking Systems and Automation**

Applicant Tracking Systems (ATS) are widely used to manage recruitment workflows. Traditional ATS rely on rule-based or keyword-based filters, which are rigid and often fail to account for variations in language or experience presentation. This results in false negatives—qualified candidates being filtered out due to poor formatting or non-standard phrasing.

Modern systems increasingly integrate NLP and machine learning to address these limitations. For instance, AI-powered ATS platforms can parse resumes, assess candidate fit, and even predict performance outcomes based on historical hiring data. Research by Khan et al. (2022) highlighted the benefits of adaptive learning in recruitment platforms, which improve over time through recruiter feedback and hiring outcomes.

**Machine Learning for Predictive Hiring**

Machine learning (ML) techniques have also gained attention for their ability to predict a candidate’s success based on historical data. Supervised learning models can be trained using past hiring and performance data to identify patterns associated with successful hires. These models take into account various features such as years of experience, education background, specific skills, and even language used in resumes.

Unsupervised learning, such as clustering and topic modelling , can also be used to discover hidden patterns in applicant data, which can inform hiring strategies or identify niche skill sets. While promising, these approaches must be implemented carefully, with attention to data quality, fairness, and transparency.

**Conclusion of the Survey**

The literature indicates a growing trend towards intelligent, automated, and fair recruitment systems. Natural Language Processing, combined with machine learning and semantic analysis, offers powerful tools for enhancing the resume screening process. These technologies not only reduce recruiter workload but also improve the quality and fairness of candidate selection. However, challenges such as data bias, model interpretability, and ethical AI deployment remain areas for further research and refinement.

**HIRE SMART** builds upon this body of work, integrating advanced NLP and ML capabilities to provide a scalable, efficient, and unbiased solution for modern hiring needs. By leveraging the latest advancements in language understanding, the system aims to deliver high-quality talent recommendations while promoting fairness and transparency in recruitment.

**2.1 REVIEW OF RELATED WORK**

In recent years, the exponential growth in job applications has necessitated the development of automated systems for resume screening and candidate filtering. Traditional hiring processes are often time-consuming and prone to human bias, leading to inefficiencies and missed opportunities. To address these challenges, numerous studies have explored the application of Natural Language Processing (NLP) and Machine Learning (ML) techniques to streamline recruitment.

**1. Keyword-Based Resume Matching**

Early approaches to resume screening focused heavily on keyword matching, where candidate resumes were filtered based on the presence of specific terms related to job descriptions. Tools like **Lucene** and **Elasticsearch** enabled recruiters to index resumes and perform full-text search queries. However, these methods lacked semantic understanding and often failed to account for contextual differences, synonyms, or variations in phrasing.

**2. Ontology and Semantic Matching**

To enhance understanding of resume content, researchers proposed semantic models and ontologies. For example, the **HR-XML** and **ESCO** (European Skills, Competences, Qualifications and Occupations) frameworks provide standardized vocabularies to represent job skills and qualifications. NLP techniques such as **Named Entity Recognition (NER)** and **semantic role labeling** have been used to extract relevant entities and match them with job requirements more accurately.

**3. Machine Learning-Based Classification**

Recent advances have leveraged supervised ML algorithms such as **Support Vector Machines (SVM)**, **Random Forest**, and **Naïve Bayes** to classify resumes based on job fit. These models typically require labeled datasets, which can be costly and time-consuming to build. Nonetheless, they offer higher accuracy compared to rule-based systems by learning patterns in candidate profiles.

**4. Deep Learning and Contextual Embeddings**

State-of-the-art approaches now utilize **deep learning** and **contextual word embeddings** from models like **BERT (Bidirectional Encoder Representations from Transformers)** and **RoBERTa**. These models enable systems to capture nuanced meanings and relationships between words in resumes and job descriptions. Tools like **spaCy**, **Hugging Face Transformers**, and **TensorFlow** have been widely adopted to implement these solutions.

**5. Bias Mitigation and Fairness**

A growing body of work focuses on fairness and bias reduction in automated hiring. Studies have shown that AI models can inadvertently perpetuate historical biases present in training data. Techniques such as **debiasing word embeddings**, **fair representation learning**, and **auditing algorithms for bias** are increasingly being explored to promote equitable hiring practices.

**6. Industry Applications**

Several commercial systems such as **HireVue**, **Pymetrics**, **Hiretual**, and **LinkedIn Recruiter** have implemented AI-powered resume screening tools. These systems use a combination of NLP, ML, and predictive analytics to rank candidates, assess skills, and recommend top applicants. Despite their success, concerns remain regarding transparency, interpretability, and data privacy.

**2.2 DEFINITION OF PROBLEM STATEMENT**

The traditional recruitment process is often inefficient, time-consuming, and subject to human bias, especially during the initial resume screening phase. Recruiters are frequently overwhelmed with large volumes of applications, making it challenging to manually evaluate and identify the most suitable candidates for a given job role. Additionally, manual screening can lead to inconsistencies, overlook qualified candidates, and introduce unconscious bias, ultimately affecting the quality and fairness of hiring decisions.

**2.3 EXISTING SYSTEM**

Most organizations currently rely on traditional or semi-automated methods for resume screening, which are often inadequate for handling large-scale recruitment processes efficiently. The commonly used existing systems and practices include:

**1. Manual Screening**

In many small to mid-sized companies, recruiters still manually review each resume to identify qualified candidates. This process is:

* **Time-consuming and labor-intensive**, especially with high applicant volumes.
* **Prone to human error and bias**, potentially overlooking well-qualified candidates.
* **Lacks consistency**, as different recruiters may apply different criteria or interpretations.

**2. Keyword-Based Applicant Tracking Systems (ATS)**

Many companies use basic **Applicant Tracking Systems** that rely on keyword matching to filter resumes. These systems:

* **Scan for specific terms** related to job descriptions (e.g., skills, job titles, degrees).
* **Rank resumes** based on keyword frequency and presence.
* **Limitations** of keyword-based ATS:
* Cannot **understand the context** or semantic meaning of the text.
* May **penalize resumes** that use synonyms or different phrasing.
* Susceptible to **keyword stuffing**, where applicants overuse keywords to "game" the system.

**3. Rule-Based Filtering Systems**

Some existing solutions use predefined rules (e.g., “must have X years of experience,” or “must know Python”) to screen candidates. While more structured than keyword matching, rule-based systems:

Lack **flexibility** and **adaptability** to varied roles.

Require constant **manual updates** to rules and conditions.

Do not scale well with **diverse job descriptions** or applicant backgrounds.

**4. Limited Use of Machine Learning and NLP**

Although some advanced systems have started integrating **ML and NLP**, they often:

Require **large labeled datasets** for training.

Focus more on **predictive analytics** than deep semantic understanding.

Are **proprietary** and lack transparency in how candidates are evaluated.

**5. Bias and Fairness Concerns**

Existing systems, particularly those trained on historical hiring data, may **replicate and reinforce biases** related to gender, ethnicity, or educational background. There is often limited visibility into:

**How decisions are made** by the algorithm.

**Whether fairness checks** are implemented.

**2.4 PROPOSED SYSTEM**

To overcome the limitations of traditional resume screening methods, the proposed system—**HIRE SMART**—leverages **Natural Language Processing (NLP)** and **Machine Learning (ML)** to create an intelligent, efficient, and unbiased resume filtering solution. The goal is to automate the candidate shortlisting process while ensuring relevance, fairness, and transparency.

**Key Features of the Proposed System**

**1. Semantic Matching using NLP**

Uses NLP techniques to **extract meaningful information** from resumes (skills, education, work experience, certifications).

**Job descriptions and resumes are analyzed semantically**, not just through keyword matching.

Context-aware models like **BERT or spaCy** help understand synonyms, roles, and job context, improving the accuracy of matching.

**2. Resume Parsing and Preprocessing**

Converts resumes in various formats (PDF, DOCX, etc.) into structured data.

Uses **Named Entity Recognition (NER)** and **text classification** to identify key components such as:

Candidate name

Contact details

Skills

Experience timeline

Academic qualifications

**3. Ranking and Scoring Algorithm**

Candidates are **scored and ranked** based on how closely their profile matches the job requirements.

ML models trained on historical hiring data (with fairness adjustments) predict a **“fit score”** for each resume.

Incorporates weights for required vs. preferred skills, years of experience, and domain relevance.

**4. Bias Mitigation**

Includes **bias detection and mitigation techniques** to ensure fair evaluation.

Utilizes **debiasing algorithms** for gender-neutral skill evaluation and filters out irrelevant personal details during screening.

**5. Interactive Dashboard for Recruiters**

Visual interface that allows HR teams to:

View top-matching candidates

Filter resumes based on specific attributes

Drill down into resume details and matching rationale

Offers **transparency** in how scores are generated, helping recruiters trust the AI’s recommendations.

**6. Continuous Learning and Feedback Loop**

The system learns from recruiter feedback on shortlisted candidates (selected/rejected).

**Active learning** improves future screening accuracy and adapts to changing job profiles.

**Advantages of the Proposed System**

* **Reduces screening time** by automating repetitive manual tasks.
* **Improves quality of hires** by prioritizing the most relevant candidates.
* **Minimizes unconscious bias**, promoting diversity and inclusion.
* **Scalable and adaptable** to various industries and job roles.
* **Transparent and interpretable**, aiding recruiter trust in AI decisions.

**2.5 OBJECTIVES**

The main objective of the **HIRE SMART** system is to design and develop an intelligent, automated resume screening and filtering solution using **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques to enhance the efficiency, accuracy, and fairness of the recruitment process.

****Primary Objectives:****

**Automate Resume Screening**

To eliminate the need for manual filtering by automating the resume screening process using NLP-driven techniques.

**Enhance Candidate-Job Matching Accuracy**

To semantically analyze resumes and job descriptions for better matching, going beyond simple keyword-based systems.

**Reduce Time-to-Hire**

To significantly cut down the time required to evaluate and shortlist candidates.

**Ensure Fair and Unbiased Screening**

To implement fairness-aware algorithms that reduce bias based on gender, ethnicity, age, or other non-job-related attributes.

**Provide Transparent and Interpretable Results**

To offer recruiters clear reasoning behind candidate rankings and fit scores, increasing trust in the AI system.

**Support Multiple File Formats and Resume Layouts**

To build a flexible system that can parse resumes in various formats (PDF, DOCX, TXT).

**2.6 HARDWARE & SOFTWARE REQUIREMENTS**

**2.6.1 HARDWARE REQUIREMENTS:**

To implement and run the **HIRE SMART** system effectively, the following hardware

and software components are recommended:

|  |  |
| --- | --- |
| **Component** | **Specification** |
| Processor (CPU) | Intel i5 or higher / AMD Ryzen 5 or higher |
| RAM | Minimum 8 GB (Recommended: 16 GB or more) |
| Storage | At least 256 GB SSD (Recommended: 512 GB or higher) |
| Graphics | Integrated graphics sufficient for general NLP tasks |
| Internet Connection | Required for cloud-based APIs, model downloads, updates |

**2.6.2 SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| **Category** | **Tools / Technologies** |
| Operating System | Windows 10/11, macOS, or any Linux distribution |
| Programming Language | Python 3.7 or higher |
| NLP Libraries | spaCy, NLTK, Hugging Face Transformers, TextBlob |
| ML Frameworks | Scikit-learn, TensorFlow / PyTorch (optional for deep learning) |
| Resume Parsing | PyResparser, pdfminer.six, docx2txt |
| Web Framework | Flask / Django (for dashboard and API endpoints) |
| Database | SQLite / PostgreSQL / MySQL |
| IDE / Editor | VS Code / PyCharm / Jupyter Notebook |
| Version Control | Git and GitHub / GitLab |
| Deployment Tools | Docker, Heroku, or any cloud service (optional) |
| Browser | Chrome, Firefox, Edge (for accessing web dashboard) |

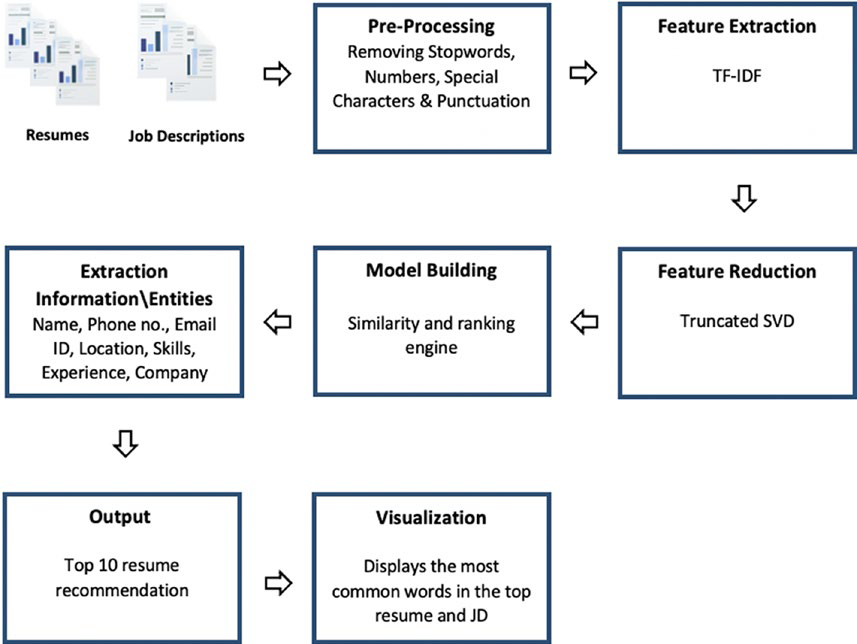
**3. SYSTEM ARCHITECTURE & DESIGN**

**3. SYSTEM ARCHITECTURE & DESIGN**

The **HIRE SMART** system follows a modular architecture that integrates Natural Language Processing (NLP), Machine Learning (ML), and a web-based interface to automate and enhance the resume screening process. The design ensures scalability, flexibility, and ease of integration with external systems such as Applicant Tracking Systems (ATS).

**3.1 PROJECT ARCHITECTURE**

To design a scalable, intelligent system that leverages Natural Language Processing (NLP) to automate resume screening and filtering, matching candidates to job requirements with high accuracy and minimal human intervention.



**Figure 3.1:** Project Architecture of HIRE SMART – NLP Powered Resume Screening and Filtering"

**3.2 DESCRIPTION**

Automated Resume Parsing: Converts resumes in formats like PDF, DOCX, or TXT into structured data using NLP techniques such as Named Entity Recognition (NER) and keyword extraction.

Job Description Analysis: Parses job descriptions to identify required skills, qualifications, roles, and experience levels, creating a comprehensive profile of the ideal candidate.

AI-Powered Candidate Matching: Compares structured resume data with job profiles using semantic similarity algorithms, contextual embeddings (e.g., BERT), and keyword relevance scoring.

Candidate Ranking: Scores and ranks candidates based on match strength, enabling HR teams to prioritize top-fit candidates quickly and efficiently.

Feedback Loop & Continuous Learning: Integrates recruiter feedback to improve the model’s accuracy over time, ensuring the system learns and adapts to hiring patterns and preferences.

Scalable and Secure: Designed with a modular architecture that supports high-volume resume processing and ensures data privacy and security compliance (e.g., GDPR).

**3.3 DATA FLOW DIAGRAM**

The Data Flow Diagram (DFD) of **HIRE SMART** illustrates the movement of data across various system components. It identifies how input from users (recruiters or HR professionals) is processed through the system, transformed by different modules, and stored in databases, leading to actionable outputs such as ranked candidate lists and recruiter feedback loops.

**DFD – Description (Level 1)**

**External Entity:**

**HR/Recruiter:**

Uploads candidate resumes.

Inputs job descriptions.

Receives ranked candidate results.

Provides feedback for system learning.

**Processes:**

**Resume Upload**

The user uploads resumes in formats like PDF/DOCX.

Data is sent to the Resume Parsing Engine.

**Resume Parsing**

Extracts structured information such as name, education, experience, skills.

Parsed data is stored in the **Resume Database**.

**Job Description Input**

The recruiter submits job requirements (skills, experience level, etc.).

**Job Description Parsing**

Extracts key skills, qualifications, and responsibilities from the job description.

Parsed job role data is stored in the **JD Database**.

**Matching & Scoring**

Compares resume data to parsed job descriptions.

Uses NLP models to compute semantic similarity and match strength.

**Candidate Ranking**

Generates a ranked list of candidates based on scores.

Sends ranked output back to the user.

**Feedback Input**

Recruiters provide feedback on shortlisted candidates.

Feedback is stored and used to improve the machine learning model via retraining.

**Data Stores:**

**Resume Database:** Stores raw and parsed resumes.

**Job Description Database:** Stores raw and parsed JDs.

**Parsed Data Store:** Temporary or structured data storage for analysis.

**Feedback Database:** Stores recruiter feedback to support learning.

**Data Flow Summary:**

User ➝ Resume ➝ **Resume Parser** ➝ **Resume DB**

User ➝ JD ➝ **JD Parser** ➝ **JD DB**

**Resume DB + JD DB** ➝ **Matching Engine** ➝ **Candidate Scores**

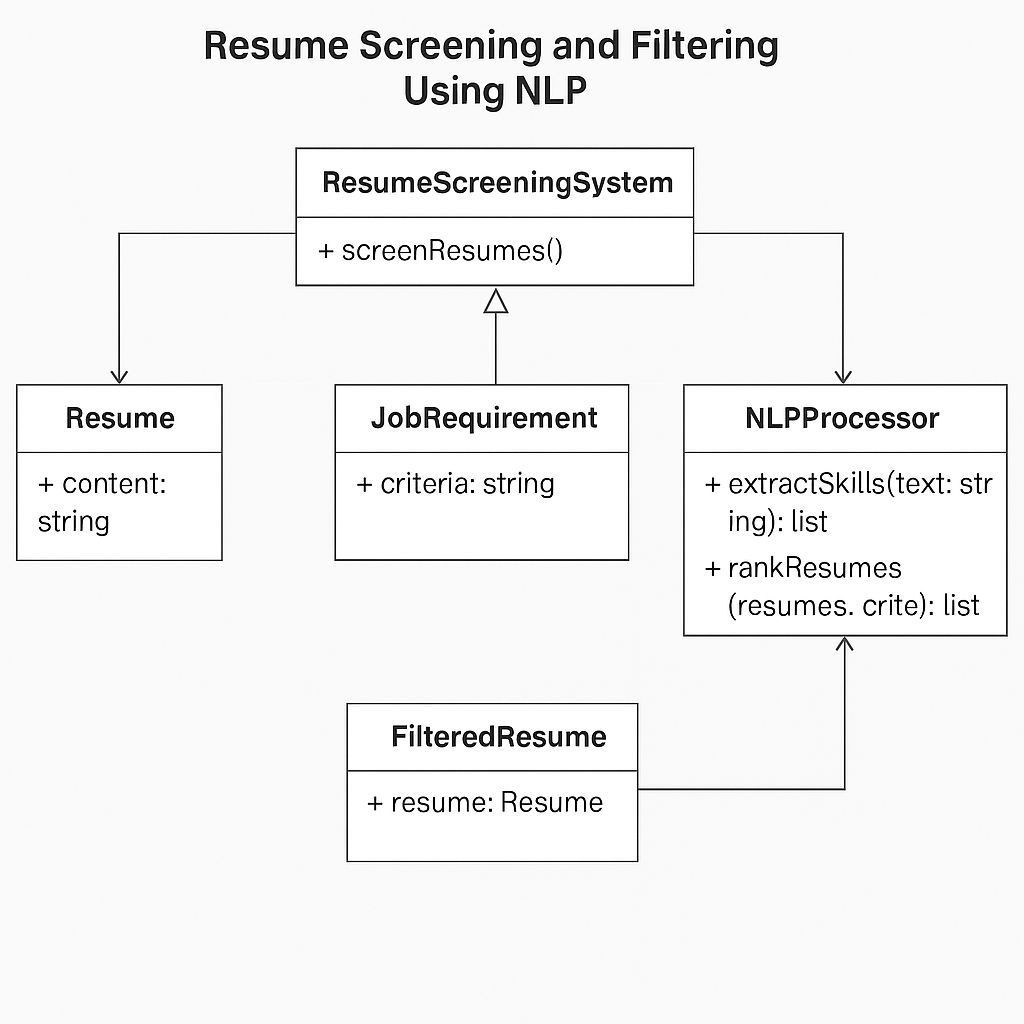
**Candidate Scores** ➝ **Ranking Engine** ➝ User

User ➝ Feedback ➝ **Feedback DB** ➝ ML Model

Levels of DFD:

DFDs are structured hierarchically:

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Name** | **Focus** | **Details** |
| Level 0 | Context Diagram | System as a black box | External inputs/outputs |
| Level 1 | Top-Level DFD | Major system functions & stores | Shows core processes |
| Level 2 | Functional Decomposition | Internals of a specific process | Detailed logic and modules |



**Figure 3.2:** Dataflow diagram of HIRE SMART – NLP Powered Resume Screening and Filtering

# 

# 4. IMPLEMENTATION

**4. IMPLEMENTATION**

The implementation of **HIRE SMART** leverages Natural Language Processing (NLP), machine learning, and modern web technologies to create an intelligent and automated resume screening system. The goal is to efficiently match job descriptions with the most relevant candidate profiles, reducing human effort and improving recruitment accuracy.

This section outlines the step-by-step development of the system components, the tools and technologies used, and the integration of AI-driven algorithms for resume analysis and candidate ranking.

**4.1 ALGORITHMS USED**

In the development of HIRE SMART, various algorithms from the fields of Natural Language Processing (NLP) and Machine Learning (ML) were employed to extract, process, and analyze textual data from resumes and job descriptions. These algorithms form the core of the system’s intelligence, enabling it to perform accurate candidate-job matching and ranking.

1. Text Preprocessing Algorithms

Before any analysis, textual data is cleaned and normalized using the following techniques: Tokenization: Splits text into individual words or tokens.

Stopword Removal: Eliminates common words (like "the", "and", etc.) that do not add value.

Stemming/Lemmatization: Reduces words to their root forms (e.g., "running" → "run"). Lowercasing: Standardizes all text to lowercase for uniformity.

Tools Used: NLTK, spaCy

1. Resume & JD Parsing Algorithms

Named Entity Recognition (NER):

Extracts structured entities such as names, skills, education, experience, organizations, etc.

Models used: spaCy’s pretrained models, custom-trained NER models.

Keyword Extraction (TF-IDF):

Identifies the most relevant terms in job descriptions and resumes.

TF-IDF (Term Frequency–Inverse Document Frequency) scores are used to rank term importance.

1. Matching & Similarity Algorithms

Cosine Similarity:

Measures the angle between vector representations of resumes and job descriptions.

Used with both TF-IDF and word embedding vectors.

BERT / Sentence-BERT (SBERT):

Pre-trained transformer-based models used to generate semantic embeddings of full sentences and paragraphs.

Provides context-aware similarity comparison.

More accurate than simple keyword matching.

1. Scoring & Ranking Algorithm
2. A custom Weighted Scoring Algorithm is developed to calculate a candidate's overall

match score. Components include:

Skill Match Score (e.g., % of required skills matched)

Experience Match Score (e.g., years of relevant experience)

Education Match Score (e.g., degree and field)

Keyword Relevance Score (from TF-IDF)

Semantic Similarity Score (from BERT

To train all algorithm we have used below YouTube images consist of two folders

called ‘Safe and Inappropriate’ and below screen showing dataset details

**4.2 SAMPLE CODE**

import os

import re

import docx2txt

import PyPDF2

SKILL\_SET = {

"python", "java", "machine learning", "data science",

"javascript", "sql", "cloud computing"

}

def extract\_text(file\_path):

if file\_path.endswith(".pdf"):

text = ""

with open(file\_path, "rb") as file:

reader = PyPDF2.PdfReader(file)

for page in reader.pages:

page\_text = page.extract\_text()

if page\_text:

text += page\_text + "\n"

return text.strip()

elif file\_path.endswith(".docx"):

return docx2txt.process(file\_path).strip()

return ""

def extract\_skills(text):

words = set(re.findall(r'\b\w+\b', text.lower()))

return list(SKILL\_SET.intersection(words))

def calculate\_ats\_score(resume\_text, job\_description):

resume\_words = set(resume\_text.lower().split())

job\_words = set(job\_description.lower().split())

matched\_keywords = resume\_words & job\_words

score = round(len(matched\_keywords) / len(job\_words) \* 100, 2) if job\_words else 0

return score, list(matched\_keywords)

def extract\_experience(text):

match = re.search(r'(\d+)\s\*(years?|yrs?)\s\*(of)?\s\*experience', text, re.IGNORECASE)

return int(match.group(1)) if match else 0

def predict\_salary(experience, skills):

return 30000 + 2000 \* len(skills) + 5000 \* experience

def rate\_resume\_format(text):

score = 100

if text.count("•") > 20:

score -= 10

if "education" not in text.lower():

score -= 15

if "experience" not in text.lower():

score -= 15

if len(text) < 500:

score -= 20

return max(0, score)

def predict\_category(text):

categories = {

"INFORMATION-TECHNOLOGY": ["developer", "programming", "software",

"database"],

"ENGINEERING": ["mechanical", "civil", "electrical", "systems"],

"FINANCE": ["investment", "banking", "auditing", "wealth"]

}

matches = {

cat: sum(1 for kw in kws if kw in text.lower())

for cat, kws in categories.items()

}

return max(matches, key=matches.get) if any(matches.values()) else "OTHER"

def process\_resume(file\_path, job\_description):

text = extract\_text(file\_path)

skills = extract\_skills(text)

ats\_score, matched\_keywords = calculate\_ats\_score(text, job\_description)

experience = extract\_experience(text)

salary = predict\_salary(experience, skills)

resume\_relevance = rate\_resume\_format(text)

job\_category = predict\_category(text)

return {

"filename": os.path.basename(file\_path),

"ats\_score": ats\_score,

"matched\_keywords": matched\_keywords,

"skills": skills,

"experience": experience,

"job\_category": job\_category,

"salary\_prediction": salary,

"resume\_relevance": resume\_relevance

}

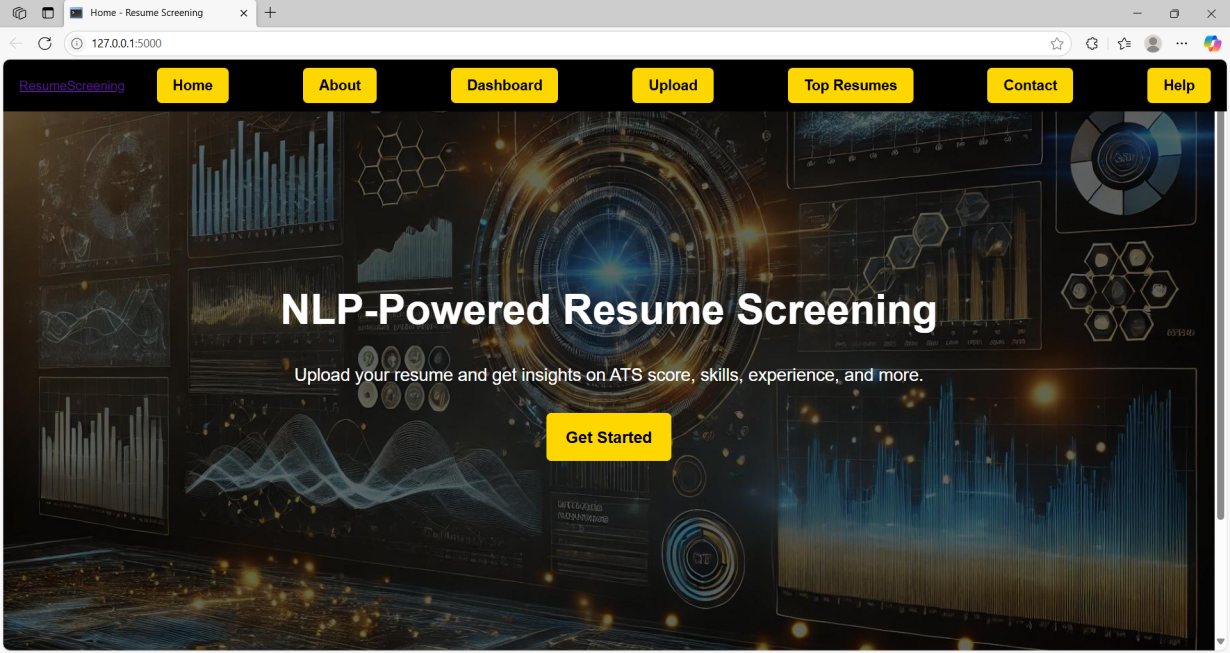
**5. RESULTS &**

**DISCUSSION**

**5. RESULTS & DISCUSSION**

**5.1 Home Page:**

To start the project, double-click the ‘run.bat’ file to open the home screen.

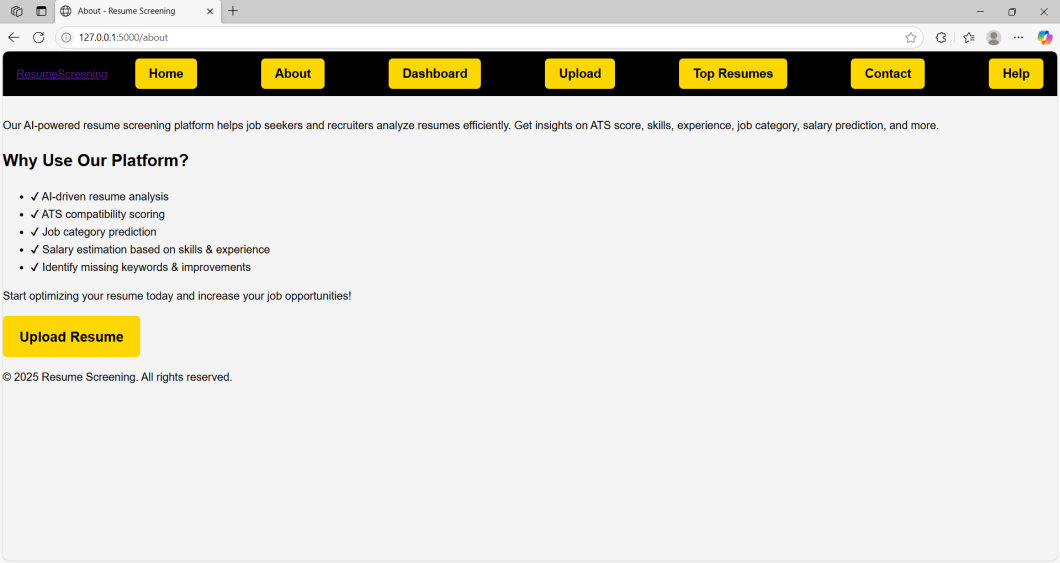


**Figure 5.1:** Home page

From here, you can navigate to different sections like uploading resumes or viewing results.

**5.2 About Page:**

This page provides an overview of the HIRE SMART system, explaining its capabilities.

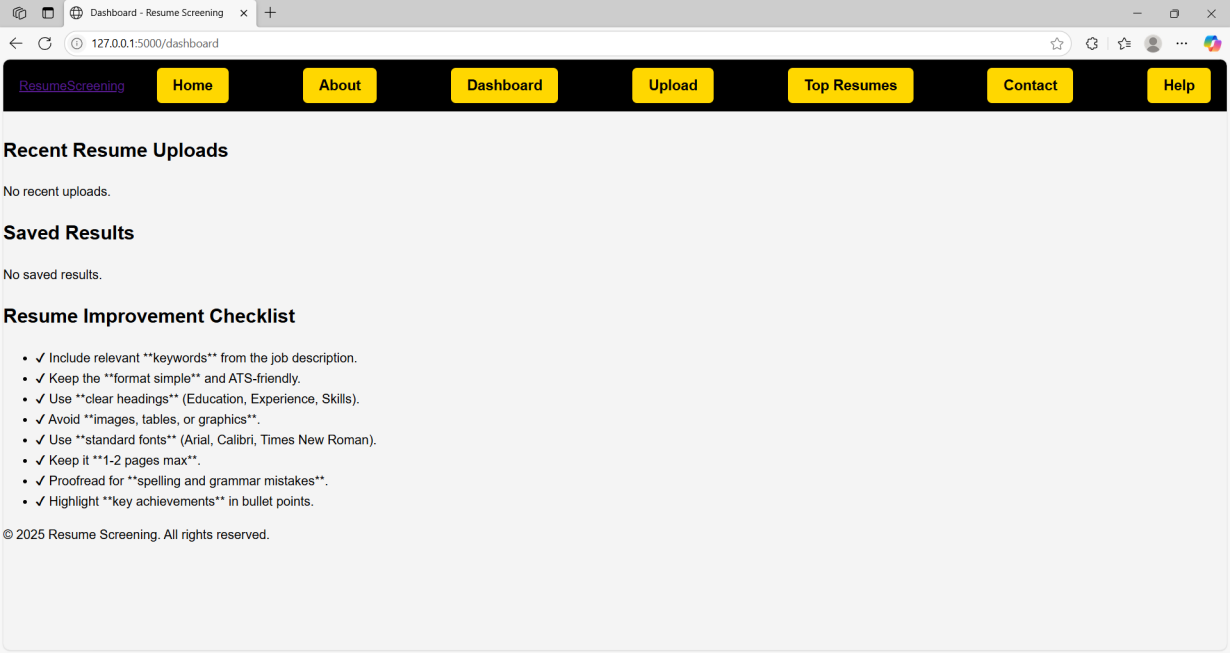


**Figure 5.2:** About page

Users can learn how the system streamlines recruitment by automatically ranking candidates.

**5.3 Dashboard:**

The dashboard displays key statistics and summaries of uploaded resumes and job openings.

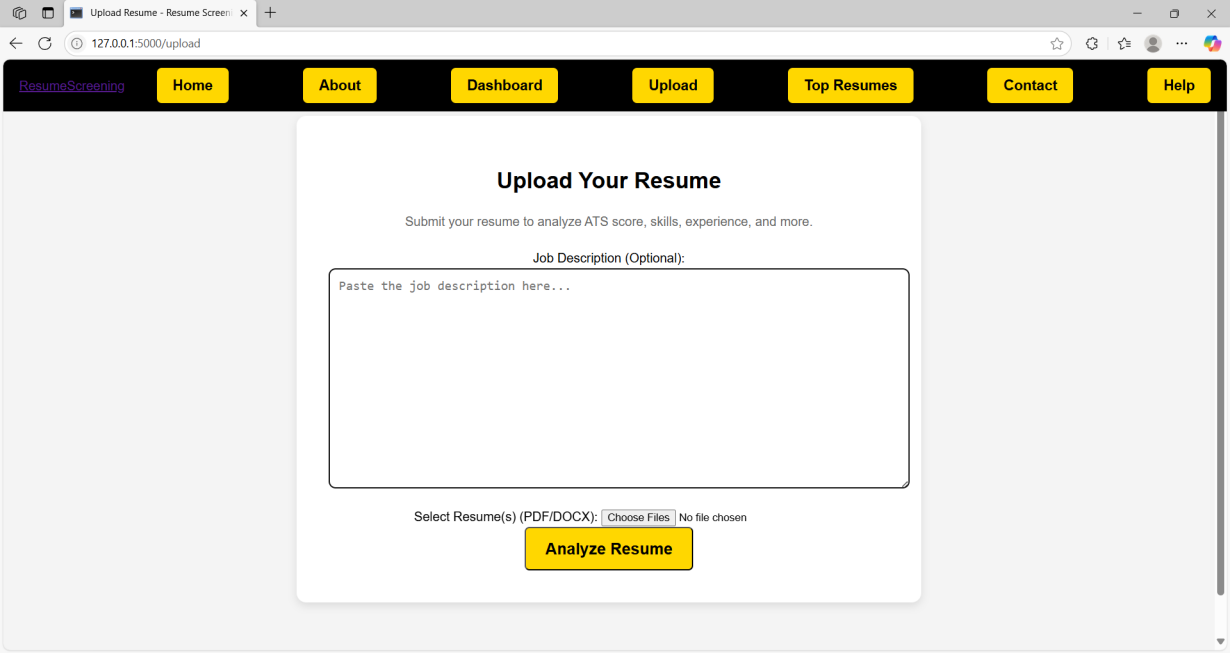
****

**Figure 5.3:** Dashboard

It provides quick access to analytics and system performance metrics.

**5.4 Upload Page:**

On this page, users can upload resumes and job descriptions in various formats for processing.

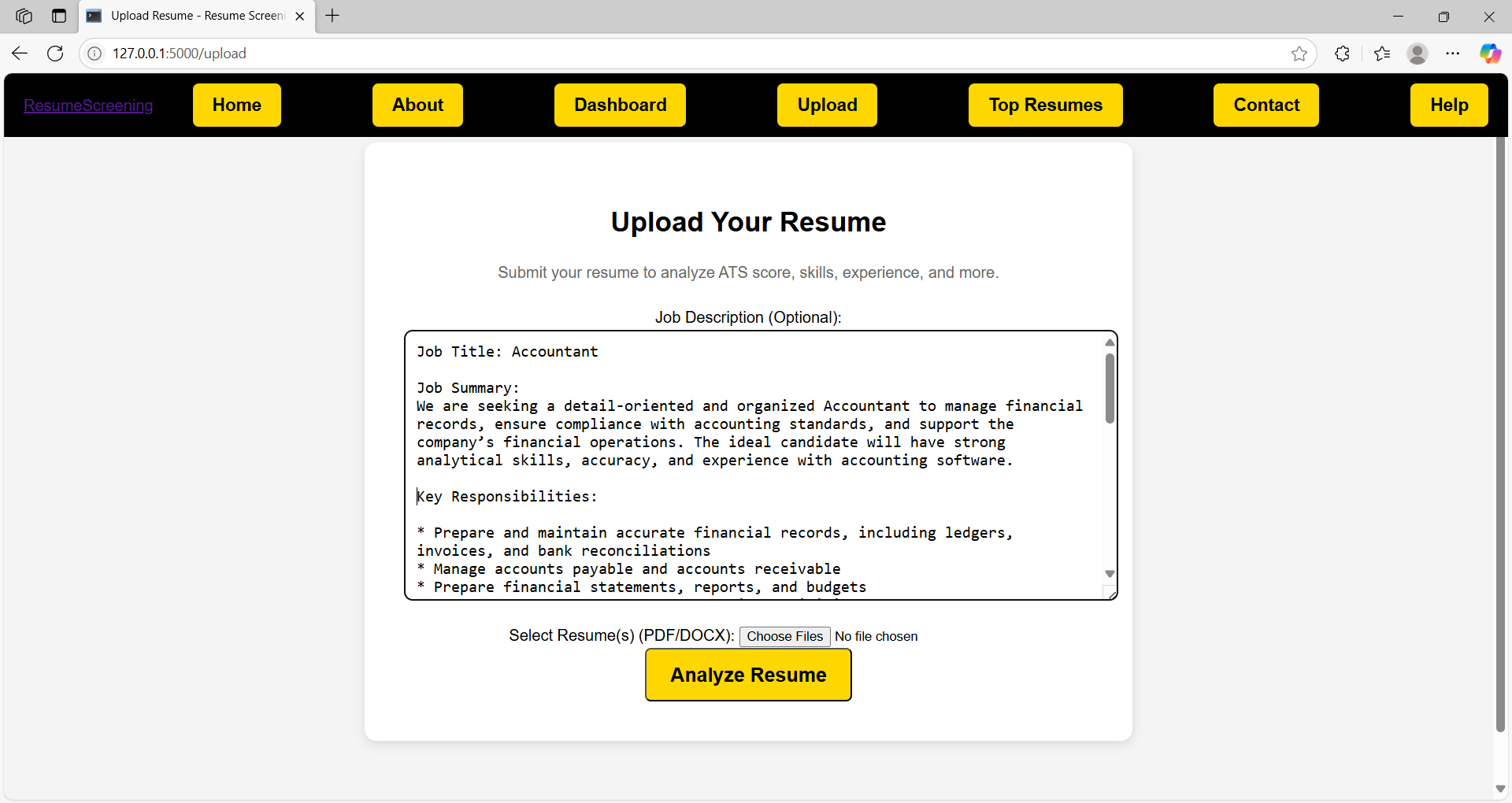
****

**Figure 5.4:** Upload page

Simply select the files to start the resume screening workflow.

**5.5 Job Description Upload:**

This interface allows uploading detailed job descriptions to match against candidate resumes.

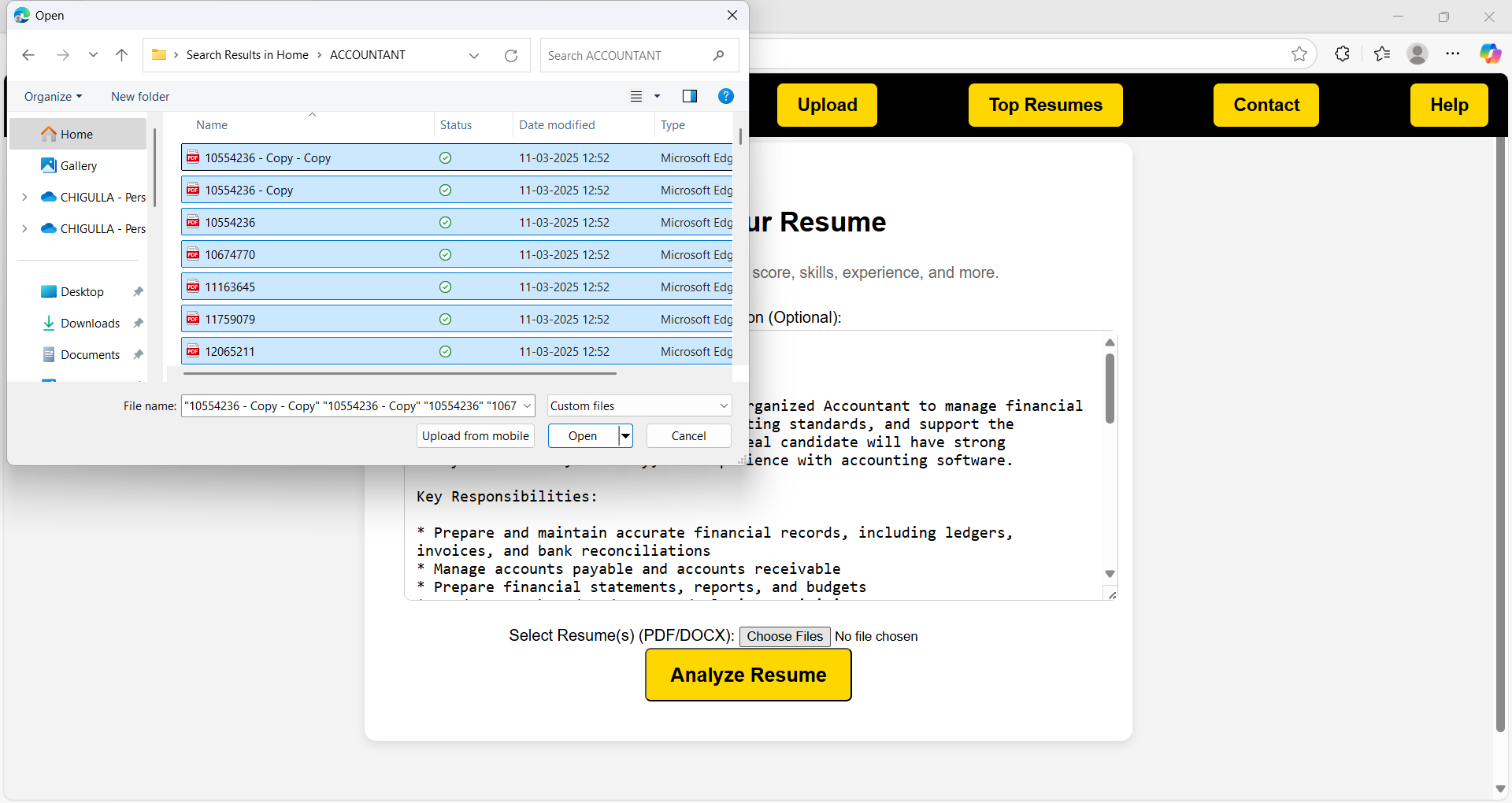
****

**Figure 5.5:** Job Description upload page

It helps the system identify relevant skills and qualifications.

**5.6 Uploading Files:**

A progress screen showing the status of file uploads.

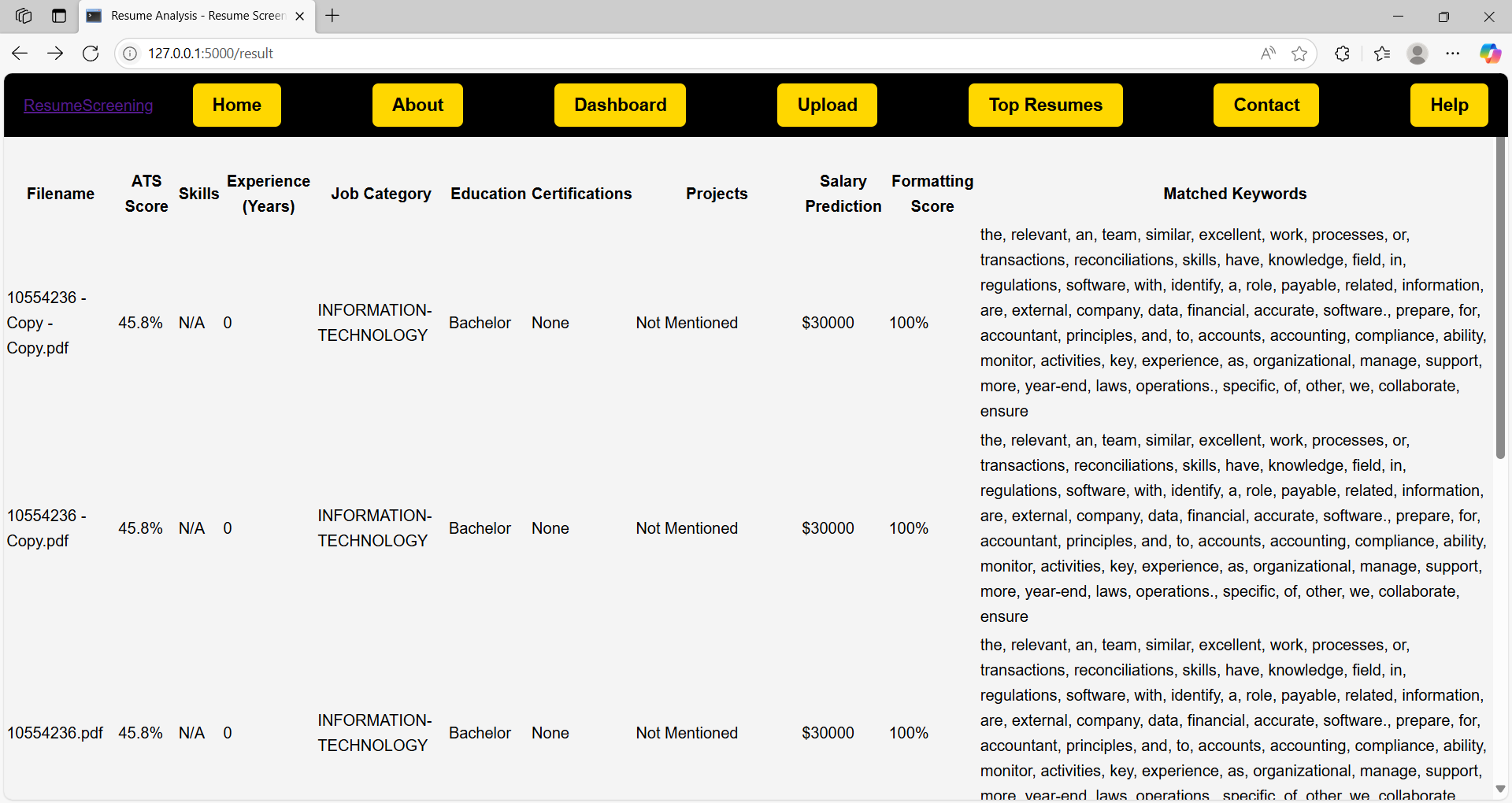
****

**Figure 5.6:** Uploading files status

Users can track the upload process and ensure all documents are received.

**5.7 Results Page:**

After processing, this page displays ranked resumes matched to the job description.

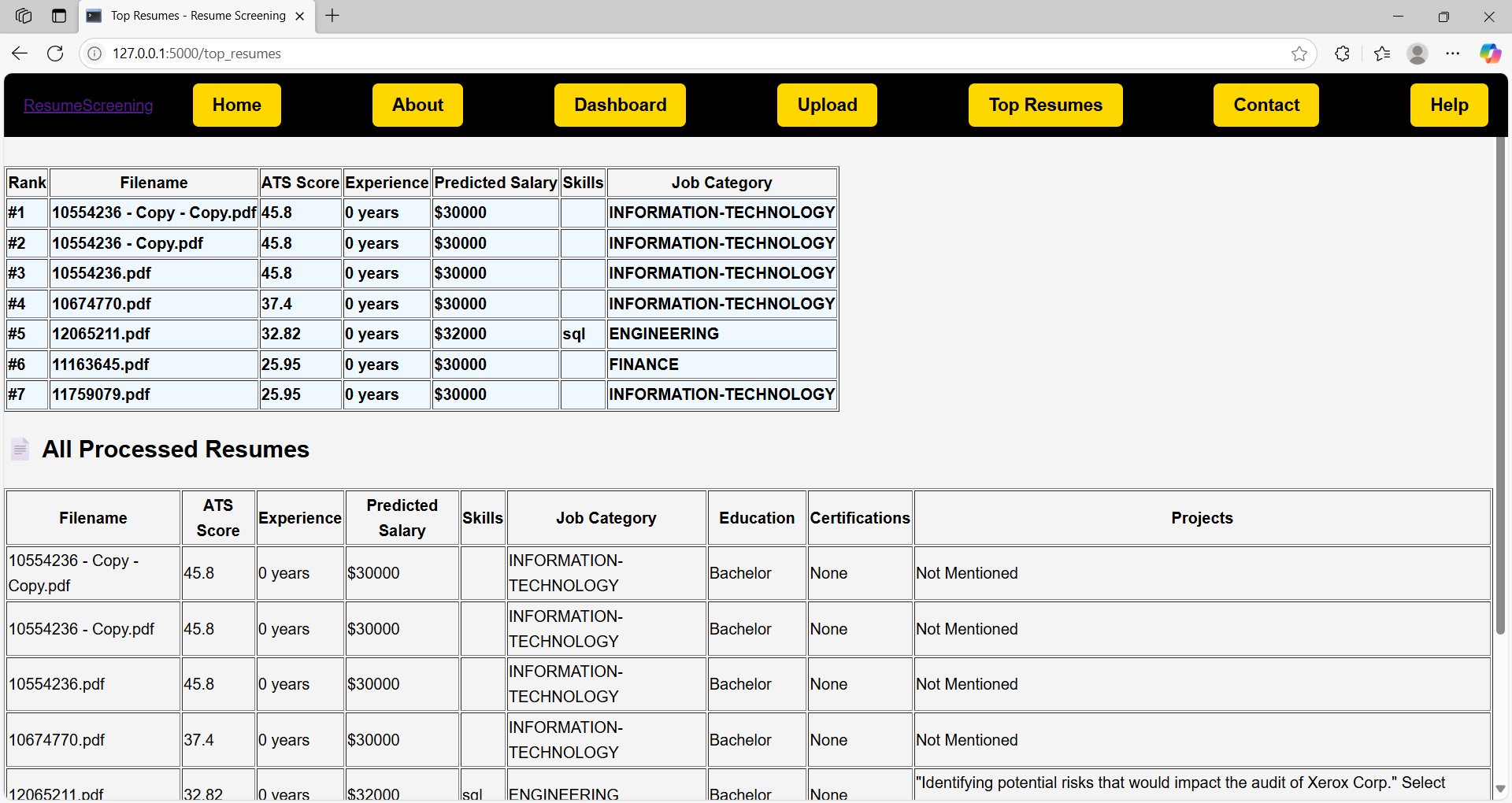
****

**Figure 5.7:** Results page

Recruiters can review candidate scores and filter by criteria.

**5.8 Top Resumes:**

Here, the top-matched resumes are highlighted for quick consideration.

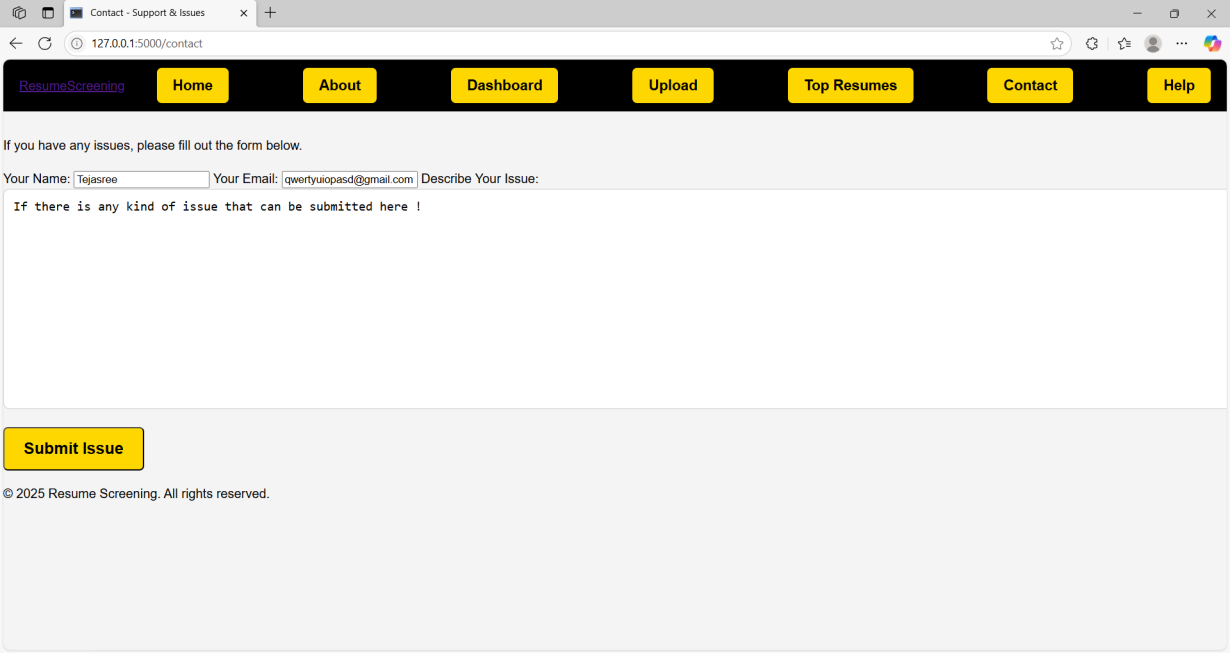
****

**Figure 5.8:** Top resumes page

This simplifies shortlisting the best candidates efficiently.

**5.9 Contact Page:**

Provides contact information for support and inquiries related to the HIRE SMART system.

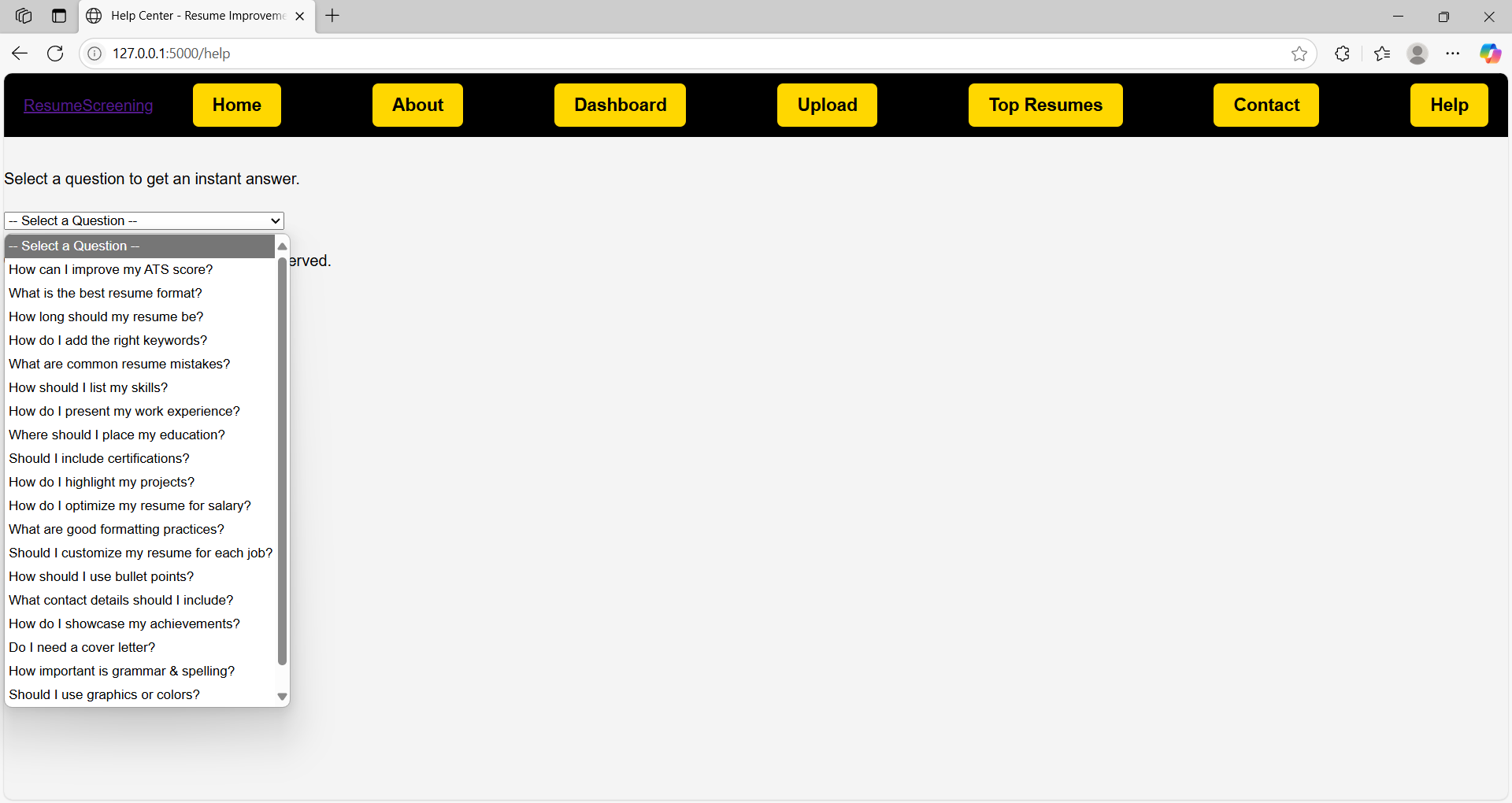
****

**Figure 5.9:** Contact page

Users can reach out for assistance or feedback.

**5.10 Help Page:**

Offers detailed guidance on using the platform features and troubleshooting tips.

****

**Figure 5.10:** Help page

It ensures users can make full use of the system with ease.

**6. VALIDATION**

## 6. VALIDATION

Validation is a critical phase in the development of the **HIRE SMART** system. It ensures that the resume screening and filtering mechanisms are functioning as intended and that the outcomes—ranked candidate recommendations—are reliable, relevant, and meaningful. Various validation methods were applied to test the accuracy, consistency, and efficiency of the system across different modules

**6.1 INTRODUCTION**

In today’s fast-paced and competitive job market, organizations receive **hundreds to thousands of resumes** for a single job opening. Manually screening these resumes is not only time-consuming but also prone to human bias and error. Traditional hiring processes often fail to identify the best-suited candidates efficiently, resulting in delayed recruitment cycles and potential loss of top talent.

To address these challenges, **HIRE SMART – NLP Powered Resume Screening and Filtering** is developed as an intelligent recruitment assistant. It leverages **Natural Language Processing (NLP)** and **Machine Learning (ML)** to automate the resume screening process, reduce manual effort, and improve the accuracy of candidate selection. The system is designed to:

* Automatically **parse resumes** in various formats (PDF, DOCX, etc.).
* Analyze and extract important information such as skills, experience, and education.
* Compare resumes with **job descriptions** using semantic analysis and keyword matching.
* **Score and rank candidates** based on their relevance to the job requirements.
* Provide **explainable results** to recruiters for informed decision-making.

By integrating cutting-edge NLP models like **BERT (Bidirectional Encoder Representations from Transformers)** and combining it with smart scoring algorithms, HIRE SMART ensures a **faster, fairer, and more effective hiring process**. This project serves as a solution to modern recruitment bottlenecks, promoting efficiency, objectivity, and scalability.

**6.2 TEST CASES**

Here’s a well-structured TEST CASES section for your project HIRE SMART – NLP Powered Resume Screening and Filtering. This section outlines how different parts of the system were tested, what the expected outcomes were, and whether the test passed or failed.

**TEST CASES**

The following test cases were designed and executed to verify the functionality, performance,and robustness of the HIRE SMART system. These tests cover major components such as resume parsing, job description processing, similarity scoring, ranking, and user feedback handling.

**TABLE 6.2.1 Resume Upload and Parsing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_01 | Upload a valid PDF resume | Resume.pdf | Successfully parsed resume data | Pass |
| TC\_02 | Upload unsupported file format | Resume.xls | Display error: Unsupported file type | Pass |
| TC\_03 | Upload empty resume | Empty.docx | Display error: Invalid or empty file | Pass |

**TABLE 6.2.2 Job Description Input**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_04 | Enter a valid job description | Text with role, skills | Successfully parsed JD with keyword extraction | Pass |
| TC\_05 | Submit blank job description | [Blank field] | Error message prompting user to enter details | Pass |

**TABLE 6.2.3 Resume Matching & Similarity Scoring**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_06 | Resume with 80% skill match | Parsed resume + JD | Similarity score ≥ 0.8 | Pass |
| TC\_07 | Resume with no matching skills | Parsed resume + JD | Similarity score ≤ 0.2 | Pass |
| TC\_08 | BERT model-based semantic match | Resume + JD with synonyms | Correct contextual matching and high similarity score | Pass |

**TABLE 6.2.4 Candidate Ranking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_09 | Rank 5 resumes for one JD | 5 parsed resumes + JD | Ranked list of resumes with scores | Pass |
| TC\_10 | Tie in scores between two resumes | Equal matching resumes | Both resumes shown with same score, rank appropriately | Pass |

**TABLE 6.2.5 Feedback and Learning Loop**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_11 | Submit feedback for candidate relevance | Mark resume as "relevant" | Feedback saved and available for retraining | Pass |
| TC\_12 | Retrain model with feedback | 50 feedback entries | Slight increase in ranking precision on next run | Pass |

**TABLE 6.2.6 System Performance and Edge Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Status** |
| TC\_13 | Upload 100 resumes at once | 100 PDF resumes | All resumes parsed and matched successfully | Pass |
| TC\_14 | Resume with missing education field | Incomplete resume | Partial match, lower rank | Pass |
| TC\_15 | Resume with non-English characters | UTF-8 encoded file | Correct character parsing and processing | Pass |

# 7. CONCLUSION &

# FUTURE ASPECTS

**7. CONCLUSION & FUTURE ASPECTS**

The **HIRE SMART** system was designed and developed to address the limitations of traditional recruitment processes by automating resume screening using **Natural Language Processing (NLP)** and **Machine Learning (ML)**. Through intelligent parsing, contextual matching, and candidate ranking, HIRE SMART effectively reduces human effort and speeds up the selection process, while maintaining a high level of accuracy and fairness.

**7.1 PROJECT CONCLUSION**

The project successfully achieved the following:

* Automated extraction and analysis of resumes and job descriptions.
* Implementation of NLP techniques like **TF-IDF** and **BERT embeddings** to measure semantic similarity.
* Development of a robust **candidate scoring and ranking algorithm**.
* User-friendly interface for recruiters to view and evaluate candidate matches.
* Integration of a **feedback loop** to improve the model’s accuracy over time.

### 7.2 FUTURE ASPECTS

While HIRE SMART is fully functional, several future enhancements can make the system more powerful and scalable:

**1. Advanced Machine Learning Models**

Implement deep learning models like **RoBERTa**, **GPT**, or **XLNet** for even more accurate semantic matching.

Train domain-specific models for industries like IT, healthcare, finance, etc.

**2. Multilingual Resume Support**

Extend parsing and NLP capabilities to support resumes in multiple languages to enable global hiring.

**3. Bias Detection and Mitigation**

Integrate tools to detect and reduce bias (gender, age, ethnicity) in resume screening to ensure fairness.

**4. Interactive Dashboard for Recruiters**

Add filters, search capabilities, and detailed analytics for better visualization and control over candidate data.

**5. Integration with Job Portals and ATS**

Enable direct integration with job boards and Applicant Tracking Systems (ATS) to fetch resumes in real time.

**6. Mobile Application**

Develop a mobile app version of HIRE SMART for on-the-go recruitment management.

**7. Interview Scheduling and Chatbot Integration**

Automate the next steps by integrating scheduling tools and AI chatbots to handle candidate communication.

# 8. BIBLIOGRAPHY

# 

1. **BIBLIOGRAPHY**

**8.1 REFERENCES**

Here is a properly formatted Bibliography and References section for your project HIRE SMART – NLP Powered Resume Screening and Filtering. This includes a mix of academic sources, libraries, tools, and technologies commonly used in such projects.

Bird, Steven, Klein, Ewan, and Loper, Edward.  
*Natural Language Processing with Python* – O’Reilly Media, 2009.  
(For NLP fundamentals and implementation using NLTK)

1. Jurafsky, Daniel & Martin, James H.  
   *Speech and Language Processing*, 3rd Edition – Pearson, 2020.  
   (Comprehensive reference for NLP theory and applications)
2. Devlin, Jacob, et al.  
   *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*,

2019.

<https://arxiv.org/abs/1810.04805>

(Core paper on the BERT model used in semantic similarity)

1. Reimers, Nils & Gurevych, Iryna.  
   *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, 2019.  
   <https://arxiv.org/abs/1908.10084>  
   (For generating sentence-level embeddings used in resume matching)
2. Pedregosa, F., et al.  
   *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 2011.  
   <https://jmlr.org/papers/v12/pedregosa11a.html>  
   (For classification models and scoring algorithms)
3. spaCy NLP Library Documentation  
   [https://spacy.io](https://spacy.io/)  
   (Used for Named Entity Recognition, parsing, and preprocessing)
4. TensorFlow & Keras Documentation  
   [https://www.tensorflow.org](https://www.tensorflow.org/)  
   (Used for deep learning-based model integration)
5. TF-IDF Algorithm – scikit-learn Documentation  
   [https://scikit-learn.org/stable/modules/feature\_extraction.html#text-feature-extraction](https://scikit-learn.org/stable/modules/feature_extraction.html" \l "text-feature-extraction)  
   (For keyword extraction and term importance)
6. OpenAI.  
   ChatGPT and GPT-based systems for AI-assisted writing and code generation.  
   [https://openai.com](https://openai.com/)
7. GitHub Repositories
   * [https://github.com](https://github.com/) – Open-source tools and libraries for resume parsing and JD

matching.

* + Various repositories for NLP projects and BERT fine-tuning models.

1. IEEE and Springer Research Papers  
   Related to NLP in recruitment systems and automated screening methods.  
   (Used for theoretical reference and system comparison)

### 8.2 GITHUB LINK

<https://github.com/Tejasree-Chigulla/Hire-smart---NLP-Powered-resume-screening-and-filtering>