

**UNIVERSITY OF TOMORROW, DEHRADUN**

**“Resume screening system”**

*Prepared by Team No: 45*

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | Students Name | Roll Number | Sap Id |
| 1. | Shivam Singh | R2142220819 | 500106045 |
| 2. | Akanksha Bahuguna | R2142221005 | 500107767 |
| 3. | Dhruv yadav | R2142221012 | 500107790 |
| 4. | Shikhar Bhakuni | R2142220756 | 500107660 |

BACHELOR OF TECHNOLOGY, COMPUTER SCIENCE ENGINEERING

With specialization in AIML

*Under the guidance of*

***Dr. Upasana***

Division of Computer Science (SOCS), AIML Cluster, UPES

Bidholi Campus, Energy Acres, Dehradun – 248007

**Problem Statement**

Recruitment teams today face the challenge of reviewing hundreds or even thousands of resumes for a single job opening. This manual process is not only time‑consuming but also inconsistent, as human judgment can vary from person to person and day to day. Small companies without dedicated HR tools often struggle even more to keep up. There is a clear need for an automated system that can read resumes, score them fairly against job requirements (using an ATS score), and suggest the job fields the candidate appears most suited for.

Despite this need, building such a system isn’t straightforward. One of the major hurdles is finding reliable, annotated resume datasets for different job roles. Most resumes contain unstructured information, making automated analysis challenging. In addition, existing tools tend to be expensive, limited to certain industries, or too complex for small teams. Our project aims to create a practical, lightweight solution to bridge this gap.

### ****Objectives****

* **Automate resume analysis:** Develop a system that calculates an ATS score to show how well a resume matches role requirements.
* **Field recommendation:** Automatically suggest which job roles (currently Software Engineer, Data Analyst, and Consultant) the candidate’s profile aligns with, based on keywords and contextual information.
* **Ease of use:** Build a simple, interactive user interface so that non‑technical HR staff or small organizations can use it easily.
* **Technology stack:** Design the backend entirely in **Python**, leveraging its powerful text processing libraries, and launch the frontend through **Streamlit** to quickly deploy a web‑based application.
* **Scalability and future growth:** Make the system modular so we can add more job fields, improve scoring logic, and integrate machine learning models later as data availability improves.
* **Support better decision making:** Help hiring teams shortlist candidates faster and more fairly, reducing time‑to‑hire and minimizing bias.

### ****Methodology****

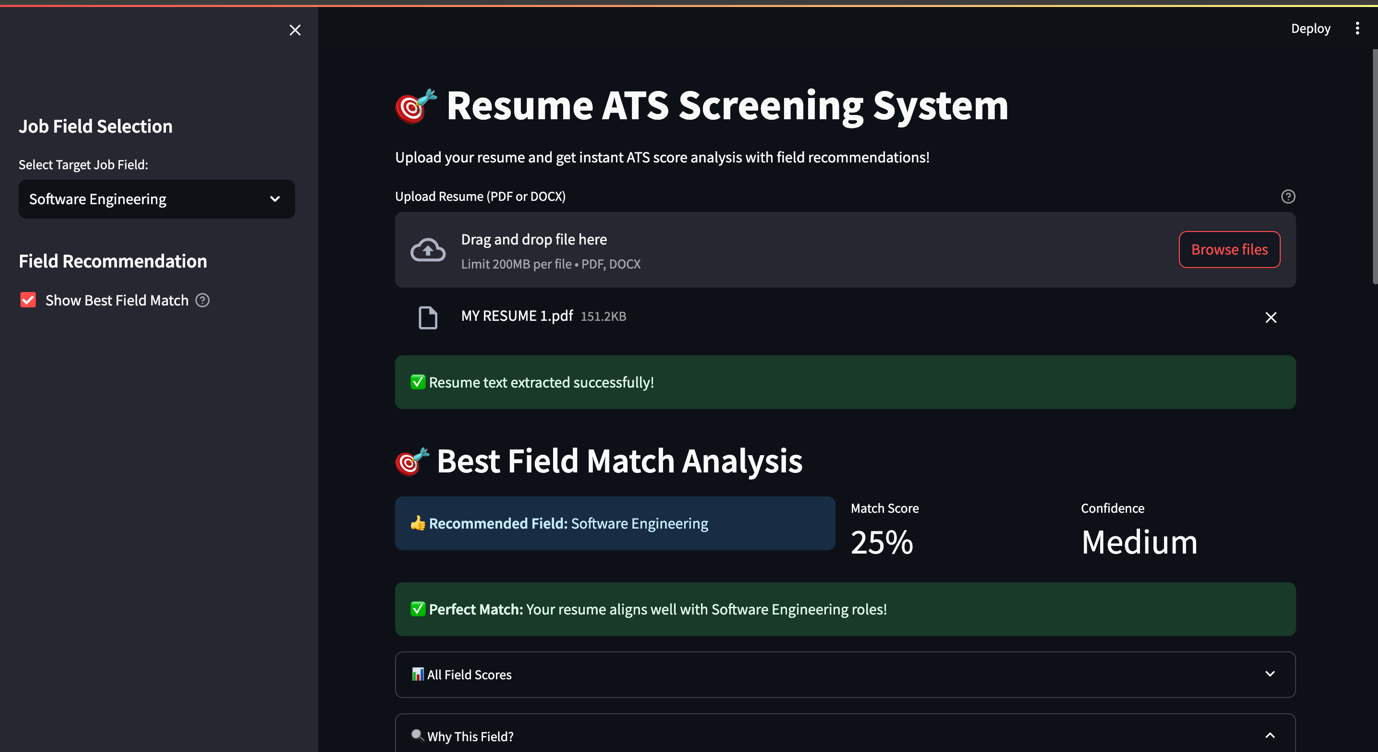
The project combines rule‑based logic and data‑driven techniques, implemented fully in **Python**, and delivered through a **Streamlit** app:

* **Data collection:** We searched extensively on GitHub and other public repositories to gather resumes tagged with job categories, since high‑quality resume datasets are rare.
* **Text preprocessing:** Using Python libraries like re, NLTK, or spaCy, we clean and tokenize the text, removing unnecessary parts and extracting meaningful keywords.
* **Feature extraction:** Identify core features such as skills, certifications, degrees, and experience. For instance, detecting “Python,” “data analysis,” or “consulting projects” as indicators of certain roles.
* **Rule‑based ATS scoring:** We built initial scoring rules, e.g., adding points if the resume mentions required tools or degrees, and reducing points if key skills are missing.
* **Recommended fields:** Using a set of domain‑specific keyword lists and context checks, the system suggests which of the supported job fields the resume is best suited for.
* **Frontend with Streamlit:** We created an interactive web app where users upload resumes (PDF or text). The app instantly displays ATS scores and recommended fields in a clean dashboard.
* **Feedback loop:** We manually reviewed sample outputs to refine scoring rules and recommendation logic.

### ****Work Done So Far****

* Collected and organized publicly available resume datasets with basic labeling.
* Implemented text cleaning, tokenization, and keyword extraction in Python.
* Built a rule‑based scoring engine that calculates ATS scores based on predefined feature weights.
* Developed an initial recommendation system to identify if a candidate fits best as a Software Engineer, Data Analyst, or Consultant.
* Designed and launched a basic yet functional **Streamlit** frontend that lets users upload resumes and see results immediately.
* Tested the system on sample resumes to check usability and refine rule thresholds.
* Documented the code and planned the architecture to allow integration of machine learning models in the future.

**FRONTEND**

****

**DATASET LINK**

[**https://drive.google.com/file/d/11PPfLuhdsL\_ktNwy\_sJ8o9NjOFAdXuj9/view?usp=drive\_link**](https://drive.google.com/file/d/11PPfLuhdsL_ktNwy_sJ8o9NjOFAdXuj9/view?usp=drive_link)

### ****Work To Be Done****

* **Make it usable for everyone:** Build the system as a **simple ATS tool** that can be used by any organization, HR team, or even individuals who want to quickly analyze resumes.
* **Easy deployment:** Make sure the system can be easily deployed and shared — for example, using **Streamlit Cloud**, so users don’t need to install anything complex.
* **UI improvements:** Enhance the user interface to look cleaner and more professional, making it easier to upload resumes and read the results.
* **Advanced NLP:** Try to add more advanced natural language processing techniques (like named entity recognition or embeddings) to understand resume content better instead of only checking for keywords.
* **Better scoring logic:** Refine the ATS scoring to be more accurate and balanced, considering the context and frequency of keywords, rather than just their presence.
* **Extract and show text:** Add functionality to **extract text from uploaded resumes (PDF or DOC)** and clearly display it on the screen, so users can see exactly what the system has read.
* **Show analysis results:** After processing, display the recommended fields, ATS score, and important extracted features directly in the app for the user to review.
* **Enhance documentation:** Write clear instructions and comments so others can understand, use, or improve the system in the future.

### ****Conclusion****

This project is an important step toward making the hiring process faster, fairer, and more data‑driven. By building the system entirely in **Python** and using **Streamlit** for an intuitive web interface, we’ve ensured it’s both powerful and easy to use. Although we currently rely on rule‑based models and a limited dataset, the system already demonstrates how resumes can be scored objectively and matched to relevant fields. Going forward, adding machine learning models, expanding datasets, and polishing the interface will make it suitable for wider adoption by organizations of all sizes. Ultimately, the system aims to help recruiters spend less time screening resumes and more time interviewing the right candidates.