“AI Resumé Parser”

A

Project Report

by

Name	Roll No.
Chanchreek Jain	R2142220468
Siddharth Joshi	R2142220664
Anuj Dhasmana	R2142220036
Saksham Siwach	R2142220937
Devansh Goyal	R2142220249

under the guidance of

Santosh Kumar Panda

Assistant Professor

School of Computer Science

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

June – 2025

CANDIDATES' DECLARATION

We hereby certify that the project work entitled **"AI Resumé Parser"** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Artificial Intelligence and Machine Learning and submitted to IBM is an authentic record of our work carried out during a period from **June**, **2025** to **August**, **2025** under the supervision of Sir **Santosh Kumar Panda, Assistant Professor, UPES Dehradun**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Chanchreek Jain
Siddharth Joshi
Anuj Dhasmana
Saksham Siwach
Devansh Goyal

Abstract

The system of recruiting is becoming automated to reduce the number of job applications an institution can receive. Resumés of many different layouts, fonts and graphical elements cannot always be properly parsed by conventional Applicant Tracking Systems (ATS), leading to the rejection of potential good candidates. The given project introduces the design and development of an AI-based resumé analyzer capable of overcoming these limitations with the help of a multi-stage extraction and analysis pipeline.

It supports both text-based and rasterized PDF resumé as well as a hybrid method, using PyMuPDF to parse the syntax directly and using the Tesseract OCR and Natural Language Processing (NLP) methods to use entity identification on scanned or graphically styled resumés. Names of candidates are extracted via a three-level cascade approach: via recognition of names in Hugging Face name recognition model, via OCR recognition and via recognition via spaCy of entities of the important parts of the document. The extraction of skills is done and the skills are categorized into technical and soft skills by means of a domain-specific NER model and a fallback based on keywords in order to achieve robustness.

Besides determining personal information and skills, the system calculates an ATS score that determines how strong the resumé is in terms of the skills covered and relevancy. It also suggests upskilling training to reputable online platforms and identifies appropriate jobs in view of the capability of the candidate. An interface available to the HR professional and the candidate is user friendly and helped under Streamlit, where the resumés can be uploaded, the extracted information checked, details that have been detected, edited and immediate feedback provided by the interface.

The solution offers better accuracy, recall, and robustness of resumé parsing over the current systems, particularly with resumés of complicated format. It shows that with a blending of various AI models, OCR, and rule-based approaches, parsing can be greatly improved in speed with the preservation of privacy. The tool will make the recruitment process more efficient, help minimize manual labour in screening and would help with more equitable shortlisting amid a wide range of hiring contexts.

Contents

1. Introduction

The AI Resumé Analyzer is a progressive programme that makes use of Artificial Intelligence (AI), Natural Language Processing (NLP), Optical Character Recognition (OCR) and Machine Learning (ML), allowing it to automate the steps of resumé screening, evaluation and feedback. It parallels modern Applicant Tracking System (ATS) capability with the enhancement of other available intelligence to provide better accuracy in candidate data extraction, the classification of the skill set, and the recommending practical ways of refinements.

The contemporary recruitment environment suggests up to hundreds or even thousands of resumés on one job position that are submitted by organizations. Screening such resumés manually takes time and is inaccurate and prone to subconscious bias. The ATS systems that are purely traditional are quick, but in most cases lack the aspect of contextual meaning whereby candidates who may relate to a given position are thrown away just because their resumés do not exactly coincide with the predetermined keywords. The AI Resumé Analyzer overcomes these weaknesses by employing several AI-driven techniques of entity recognition, skills classification and scoring. It is centered more on the aspect of correctness and flexibility allowing the recruiters to gain useful information without compromising speed.

It is able to combine both native text extraction (PyMuPDF) of resumés based on text, and OCR (Tesseract) of resumés that are heavy on images, preventing the loss of data due to different forms of documents. It uses the Hugging Face NER (Named Entity Recognition) models to parse names and skills accurately but with a fallback approach that combines text parsing and visually recognized OCR search. A grouping as technical and soft skills with the use of an IT skill NER model allows domain-specific knowledge regarding the skills.

In addition to extraction, the system does not only retrieve it, it rates a percent ATS score, according to the skills coverage, the key words density, and overall resumé optimization. It suggests suitable online training to fill in the gaps in skills and suggests the probable job position that fits the profile of the candidate. The fact that they incorporate an editable field with the name and an animation called a balloon is another level of interactivity and the involvement of the user within the UI.

1.1 History

Recruitment is a process which has greatly changed within the last few decades. Earlier, recruitment was a manual process entirely, where recruiters and the HR manager would actually have the resumes in hand, and would manually segregate into piles, and, based on a glance of the resume, would shortlist applicants on skills and experience. This was naturally time-consuming, subjective and not consistent at all with personal judgement having a big role in the process.

The introduction of computers and office automation programs towards the end of the 20th century have also led to the storage, sharing and typing of resumes. The first digital resume screening systems appeared in the 1990s, and were basically keyword search systems. They enabled the recruiter to use a database of resumes to search with particular words or phrases. Although this was a step up in terms of speed over manual review there were still false positives (irrelevant matches) and false negatives (relevant resumes lost because of wording differences).

Applicant Tracking Systems (ATS) made it into the mainstream in the early 2000s. These systems also automated the posting of jobs, receiving applications and storing of candidate databases. ATS would be able to rank candidates automatically on the basis of the number of

keywords matched. But due to a lack of contextual knowledge it also meant that skilled candidates might be disqualified because they did not include the right keywords. This gave rise to the art of resume keyword stuffing, in which one would add several iterations of various work related words simply to get through ATS.

In the 2010s, there was a movement to semantic search and Natural Language Processing (NLP). Systems have started detecting synonyms, associate words and contextual equivalents. As in the example, the role of software engineer and developer could be referred to as one and the same. The Named Entity Recognition (NER) was introduced, thus enabling systems to extract characteristics of named entities such as names, skills, organizations and date headlands directly out of the unstructured textual data.

Our AI Resume Analyzer builds on these advancements, using **multiple layers of extraction**:

1. **Direct text parsing** for clean, text-based resumes.
2. **OCR scanning** for image-heavy or design-oriented resumes.
3. **Domain-specific NER models** for precise skill and name detection.
4. **Fallback mechanisms** that combine results from different extraction strategies.

1.2 Requirement Analysis

The requirements were divided into two categories: **functional** and **non-functional**.

Functional Requirements

Resume Upload and Processing - The system should be able to accept PDF resumes whether in text format or in image format and will be processed without the need of the user to have a file conversion of the resume.

Text Extraction - Installation of two mechanisms of extraction:

- Text-based resumes, direct parsing through PyMuPDF.
- Tesseract to scan graphically-designed resumes.

Name Recognition - Recognize the full name of the candidate no matter in which font type, color or in what part of the document. Both OCR and AI-based NER models are to be used to become robust.

Skill Extraction - Distinguish and cluster skills into Technical Skills and Soft Skills with the help of a domain-specific NER model with a default based on keywords.

Contact Information Extraction - Get email and phone number with consistency in different formatting styles.

ATS Scoring - Calculate an ATS score according to the availability and the number of skills that are applicable to the job market and of a higher weight with the technical skills.

Course Recommendation - Recommend related online courses in order to enhance the gaps in skills and promote reliable sources.

Job Role Prediction - Match the skills of the prospective series of jobs he/she can do via a skill-role mapping.

Interactive UI - Offer a clean and easy to understand streamlit interface with editing capabilities on the information detected, e.g. correcting names.

Non-Functional Requirements

Speed – Real-time processing needs to be less than 30 seconds per regular file.

Scalability - Architecture must be in the position of handling multiple resumes in a batch format where a lot of slowing down is not experienced.

Compatibility - Affability of frequent PDF variations available in other resume designs.

Security - As a way of ensuring candidate privacy, resumes should never be written to disk and should remain in memory.

1.3 Challenges

To create an AI-powered resume analyzer one must address several technical, data-related, and operational challenges.

1. Difference in Resume Styles
Resumes are available in a myriad of templates, forms and styles. Whereas some are text-populated and straight forward to parse, others incorporate the use of tables, columns, graphics or font color making the process harder to extract the text. To guarantee proper parsing both text extraction and OCR needed to be integrated, with the results of the two put together in a clever way.

2. OCR Limitations
Although Tesseract OCR is quite efficient, the rate of its accuracy is decreased when it comes to low-resolution representation of a picture or when using fancy type of typography or those with background color which ideally interfere with text recognition. Images (rescaling, contrast enhancement) had to be pre-processed in order to enhance OCR.

3. Name Detection Accuracy
There is something more difficult about detecting names than it might seem. Such typical barriers are:

- Certain names in the same line as addresses or emails.
- Names (boolean - means that it is written in larger fonts or in color) that are interpreted wrong by the text extractor.

- Names that are not English or unusual names.
  In order to resolve this, we applied a three-pronged fallback: Hugging Face name NER → OCR-based detection → spaCy entity recognition from the top-section of the resume.

4. Skill Recognition
Skills can occur in many different forms (e.g. "Python programming" vs "Python"), and so perfect matching of keywords is ineffective. The Hugging Face IT Skill NER model allowed enhancing recognition, however, it also needed custom fallback keyword lists.

5. To Trade off Precision and Recall
The system needed the properties of high recall and low false positives (high precision) in capturing all the relevant information. Extraction that is too aggressive will treat unrelated words as skills, and strong filters will fail to detect valid skills.

6. Speed Accuracy Trade-off
The performance might be affected negatively by running several extraction runs together. To optimise we:

- When possible, prefer text based extraction.
- Activating OCR, which will run only when the text extraction option fails to give the text in its entirety.
- Limiting Hugging Face API calls to relevant portions of the text.

8. Integration Complexity
Combining the output of the PyMuPDF, the Tesseract OCR, spaCy NER, and the Hugging Face models required combining the results and conflict resolution logic.

9. API Reliability
Rate limits and Hugging Face APIs downtime may result in effects on the system. This was mitigated with fallback key match rules, and standalone spaCy models.

10. User Experience
The careful balance had to be struck in the sphere of technical proficiency and soft skills. The HR staff should also be able to log-on and insert a resume and get a readable understandable output without understanding the mechanics of AI.

2. Literature Review

AI Resume Parser – Literature Review Report

3. Objectives

The main goal of the project is to create and develop an AI-based Resume Analyzer capable of parsing and assessing resumes properly no matter what their structure and format is and how complex the design is. Though most traditional Applicant Tracking Systems (ATS) use the pure keyword-based matching strategy, the goal of the project is to experiment with the newest Natural Language Processing (NLP), Optical Character Recognition (OCR), and specialized Named Entity Recognition (NER) models to make the screening process more in-depth, reliable, and just.

The specific objectives of the system are as follows:

- **Accurate Extraction of Candidate Information**
- **Skill Classification and Categorization**
- **ATS Score Computation**
- **Job Role Prediction**
- **Personalized Learning Path Recommendation**
- **User-Friendly Interaction and Editing**

4. System Analysis

3.1 Existing Systems

Applicant Tracking Systems (ATS) is used as a larger number of organizations utilize the process in the contemporary recruitment environment. The conventional ATS systems serve as the same kind of key word filter. They crawl through the text of resumes to align with job descriptions with key terms that include required skills, job title and qualification. The applicants who meet the keyword search are shortlisted and the rest are rejected automatically.

Although this is a very efficient method on a bulk filtering basis, the downside is:

Format Sensitivity - ATS frequently unable to parse resumes whose format is not the standard, i.e. multi-column layouts, incorporated images, decorative fonts or others.

Loss of Context - Keywords might be found but they are not within a relevant context such that they give false positives. Like in a project description talking about hobbies to say you like Python would still be considered as a technical skill match.

Impossibility to Process Image-based Content - Resumes scanned or with text embedded in the graphics are normally overlooked until manual processing can be carried out.

Preference over a Writing Style - Candidates who have optimized their resumes using ATS keyword matching are likely to be ranked above equally competent candidates who have included creativity in the way they write.

Little insights - The higher percentage that most ATS models give does not suggest what skills or career growth opportunities should be devoted.

More recent AI based resume parsers can extract more text and recognize context with NLP, however they still tend to be black-box solutions with few manual ways to correct them.

3.2 Motivation

Our AI resume analyzer is motivated to fill the gap between efficiency and fairness among the hiring process.

- Recruiters are usually limited by time and they may have to do the short listing of applicants within short periods of time. A faulty and /or partial parsing tool might lead to loss of best talents.

- The applicants have the problem of optimization of the resumes in both human and machine readers. By having a non-traditional design of their resume, there is the danger of them being ignored completely.

This project was influenced by the following requirements of a highly accurate and flexible parsing tool that is able to:

- Process resumes that are in complicated or graphic form or one that is not in standard form.

- Be able to recognize correctly the names of candidates on the basis of placement, font style or color.

- Create added value by giving suggestions on jobs and course recommendations by helping candidates to enhance their accounts.

- Permit the possibility of human supervision by allowing editable fields for extracted data.

3.3 Proposed System

The system involved has three important technologies:

- Natural Language Processing (NLP) Applying the spaCy and the Hugging Face NER models to find names, skills, etc. in the textual information.

- OCR with Tesseract reading text-based resumes and extracting names and skills that would otherwise not have been found with approaches involving only text.

- Specific applications of AI Domain Specificity- A model NER system for skills based on a skill specific scenario (Nucha_ITSkillNER_BERT) where the 2 groups (technical skill, soft skill) should be determined with appropriate accuracy.

The pipeline would be created so as to:

- Retrieve information about candidates in both PDF text signs and OCRs.
- Set name detection as a priority per three step fallback strategy (Hugging Face NER => OCR-based detection => spaCy NER on top text lines).
- Rank groups so as to give them points.
- Prescribe occupations and customized training in skills.
- It allows the opportunity to update names that were found during extraction.
- This is at several levels most effective to one of the traditional ATS solutions especially in regard to the non-standard resumes.

3.4 Modules

Resume Upload & File Handling - Allows PDF uploads and extracts text via PyMuPDF.

Name Extraction Module - Uses hybrid AI methods (NER + OCR + top-line analysis) to identify the candidate's name even in challenging cases.

Skill Extraction Module - Combines Hugging Face NER with spaCy-based keyword matching for accurate skill detection.

ATS Scoring Engine - Calculates a score based on the presence and importance of skills.

Job Role Prediction - Matches skills to predefined job role mappings.

Course Recommendation Engine - Suggests relevant online courses for missing skills.

Editable UI Module - Streamlit-based interface enabling manual correction of extracted data.

Visualization and Feedback - Displays results in an easy-to-understand manner, including progress bars, balloons animation, and editable fields.

3.5 Comparison Table

Feature	Traditional ATS	Modern AI Parser	Proposed System

Keyword Matching	**No**	**Yes**	**Yes**
Context Understanding	**No**	**Yes**	**Yes**
Handles Image-based Resumes	**No**	**Partial**	**Yes**
Name Detection Accuracy	**Low**	**Medium**	**High**
Skill Categorization	**No**	**Partial**	**Yes**
Course Recommendations	**No**	**Yes**	**Yes**
Editable Extracted Data	**No**	**Yes**	**Yes**
Transparency	**Low**	**Medium**	**High**

5. Implementation

The AI Resume Analyzer was built using a modular and multi-stage pipeline that was scalable, highly accurate and flexible with respect to different types of resumes. The implementation steps also entailed choosing the relevant technologies to be used, getting synchronised of the selected technologies, and design of the interface to enable extraction of the data automatically and under human oversight.

4.1 Technology Stack

Programming Language: Python 3.11 - was selected because it is good at NLP and OCR, as well as libraries which can be used to develop a web app.

Framework: Streamlit - It also has become possible to use it to create a browser-based interactive interface.

PyMuPDF Extraction: PyMuPDF (fitz) - extract text out of PDF resumes.

OCR Engine: Tesseract OCR, to detect and get the text of a picture based resume.

NLP Library: SpaCy (it is applied to process spaCy spans) and rule-based identification of entities and processing of tokens.

Transformers AI models: Hugging Face Transformers NER for generic name recognition.

Regular Expressions(re) Data Handling: Regular Expression is used to process data that are well structured such as phone numbers and email.

4.2 Workflow Overview

The workflow is divided into five major stages:

Stage 1 - Upload and Preprocessing of Files

The users can upload PDF resumes through the system. Upon upload:

Text Layer Extraction: PyMuPDF reads the in-document text layer of a PDF.

OCR Processing: In case of pages that contain pictures or fancy text, Tesseract OCR converts the images to machine-editable text.

Stage 2 - Name Extraction
Name extraction takes a three tier backup sequence:

Hugging Face NER (Primary): The dbmdz/bert-large-cased-finetuned-conll03-english model identifies PER entities in the first 1000 characters.

OCR-based Recognition (Secondary): Tesseract scans the entire resume image for person names, useful for resumes with colored text or logos.

Top-line spaCy NER (Tertiary): The top 10 lines of text are parsed for PERSON entities, covering cases where names appear early in the document.

Such a sequence makes it as reliable as possible: detection implemented with AI processes structured names, OCR processes image-based cases, and spaCy NER is the last resort.

Stage 3 - Extraction and Classification of Skills
The Nucha_ITSkillNER_BERT model is called via Hugging Face's API to detect technical (HSKILL) and soft (SSKILL) skills. To handle token splitting (e.g., "Java" → Ja, ##va), an offset-based token merging strategy was implemented.

In the event that the extraction of the AI models fails, a keyword fallback will be used through spaCy tokenization that singles out the text against arrays of technical and soft abilities.

Stage 4 - ATS Score Calculation

85% Weight: Count of identified technical skills against a portion established.

15% Weight: Percent Coverage of soft skills over a reference list.
A maximum of 100 is placed on the final score and is presented in the form of a progress bar provided by Streamlit.

Stage 5 - Recommendations & Predictions
Course Recommendations: Missing skills are mapped to curated online learning resources (Coursera, Udemy, edX).

Job Prediction: Domains compare skills to a dictionary of the roles of occupations in order to provide fast recommendations on job-fitness.

4.3 UI and User Interaction
There are five blocks on our streamlit interface:

Upload Resume - File uploader, balloon pop up on success.

View Score - Shows the ATS score.

Extracted Info - The proficient data reveal abilities, telephone, name field (can be edited) and email.

Course Recommendations - Provides the links that can be clicked during the presentation of learning of the skills to be mastered.

Job Predictor - Provides the lists of possible jobs based on the perceived abilities.

Capable editing fields (e.g. name) enables the user to make corrections in case of any error that might have occurred during the extraction process to ensure that recruiters obtain correct final reports.

4.4 Testing and Validation

A large pool of resumes were used on experimentation with the model that entailed the following:

- Text-based resumes
- Graphical resumes
- Scanned PDFs.

The name extraction and skill classification precision, recall, and F1- score were calculated and analyzed under the ground truth assessment setting. The aspect of false positives and negative ones was carried out based on the generation of confusion matrices.

4.5 Deployment Considerations

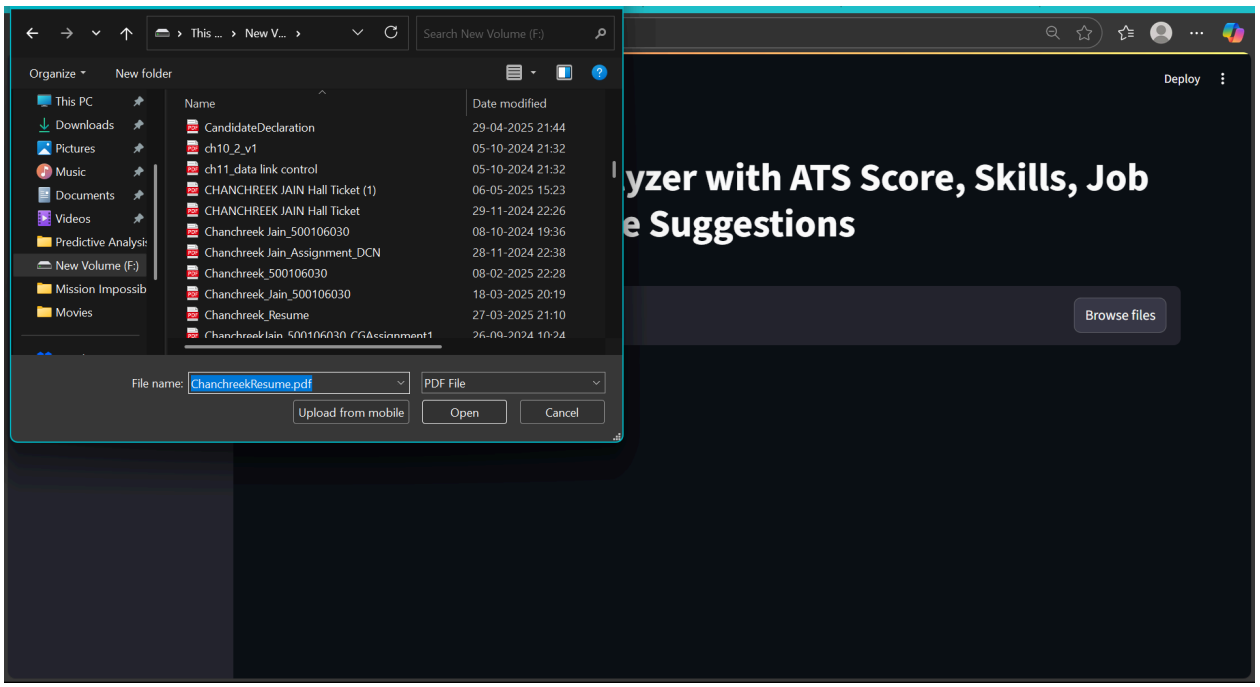
Local Mode: can be built on a developer machine with Streamlit CLI and used on a machine.

Cloud deployment: It can be sent to cloud platforms like Streamlit cloud, EC2 on AWS, Web apps on Azure such that it can be accessed to the whole world.

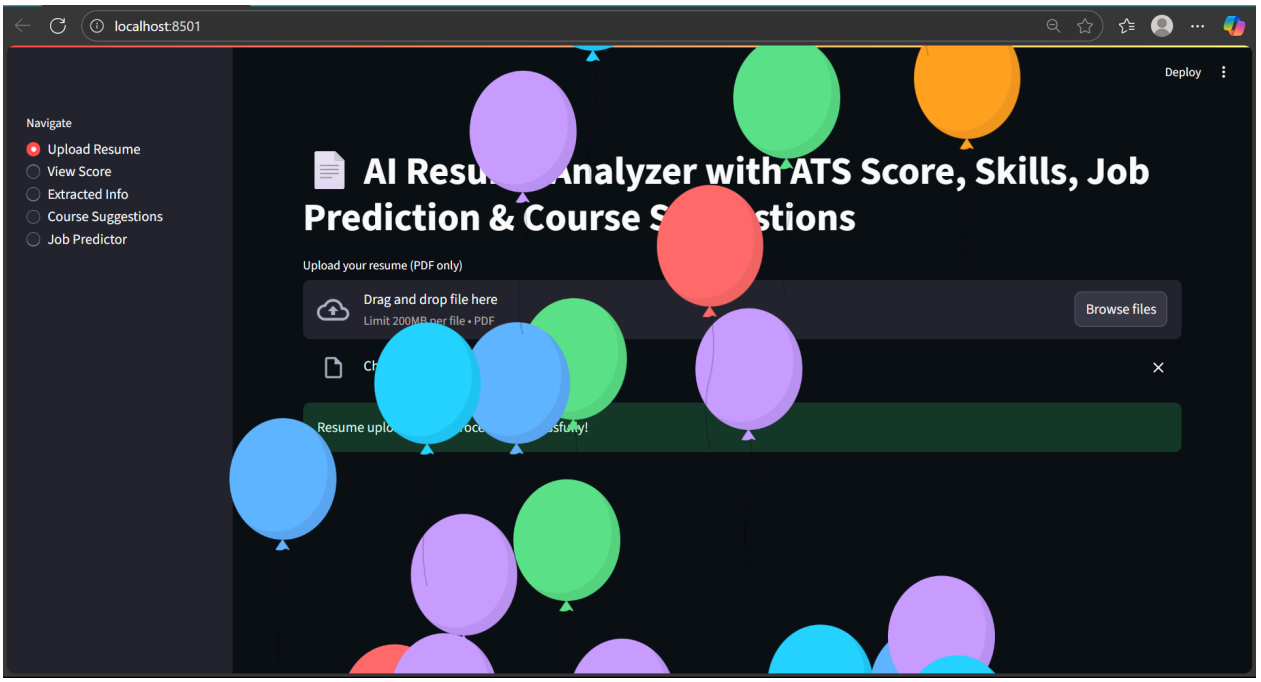
Scalability: The Hugging Face APIs will be loaded to the system, hence the rate limits will be the issue, and it has proposed that one should chain real queries with caching.

6. Results

Upload Page – Resume file selection & upload.

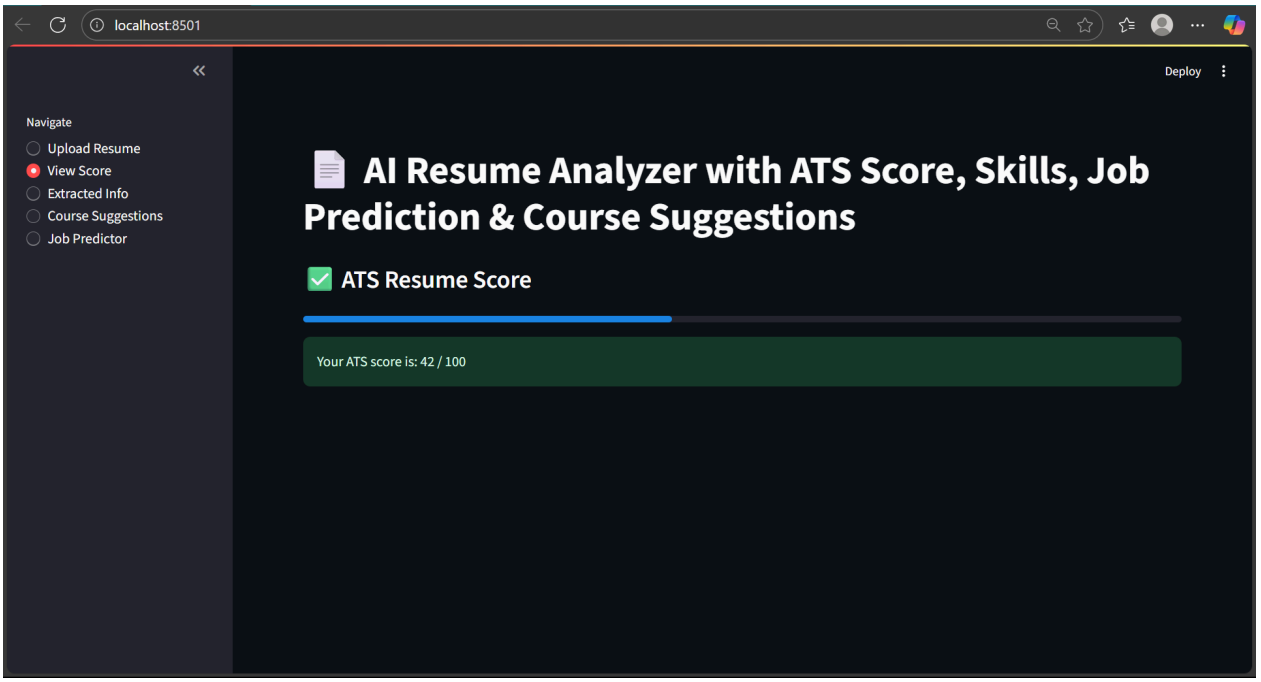


[Fig. 1 Upload Page]



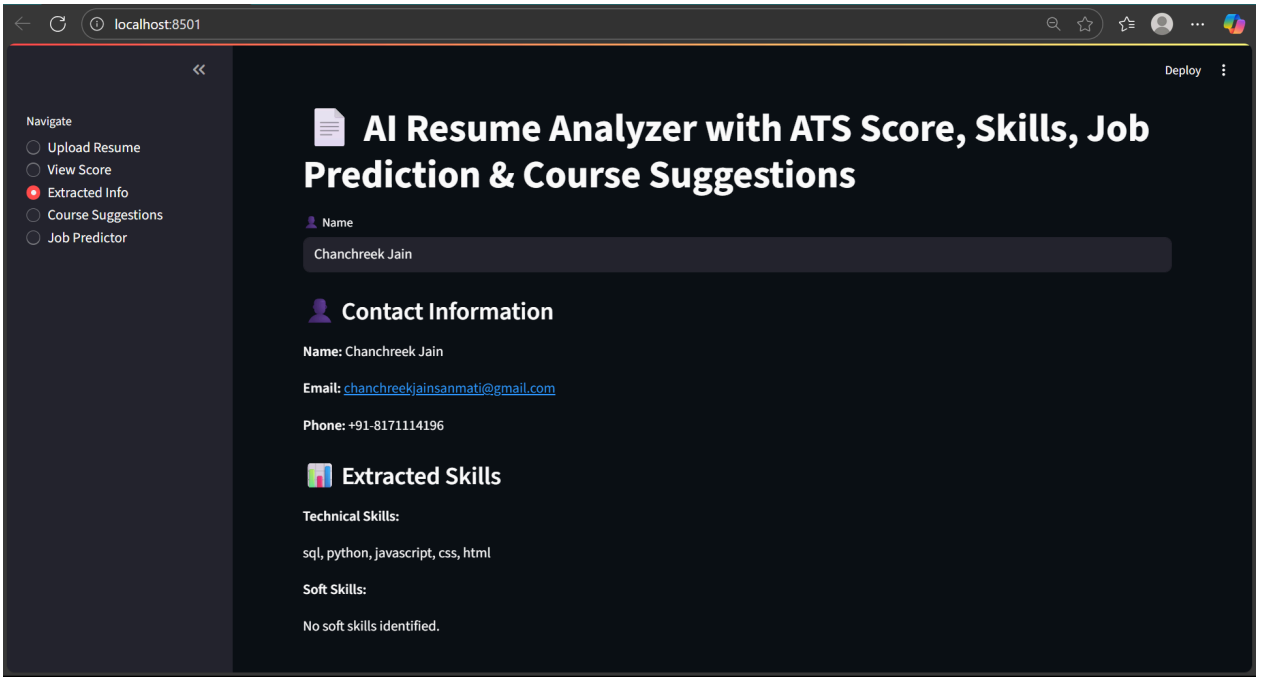
[Fig. 2 Successful Upload Animation]

ATS Score Page – Visual ATS score bar and percentage.



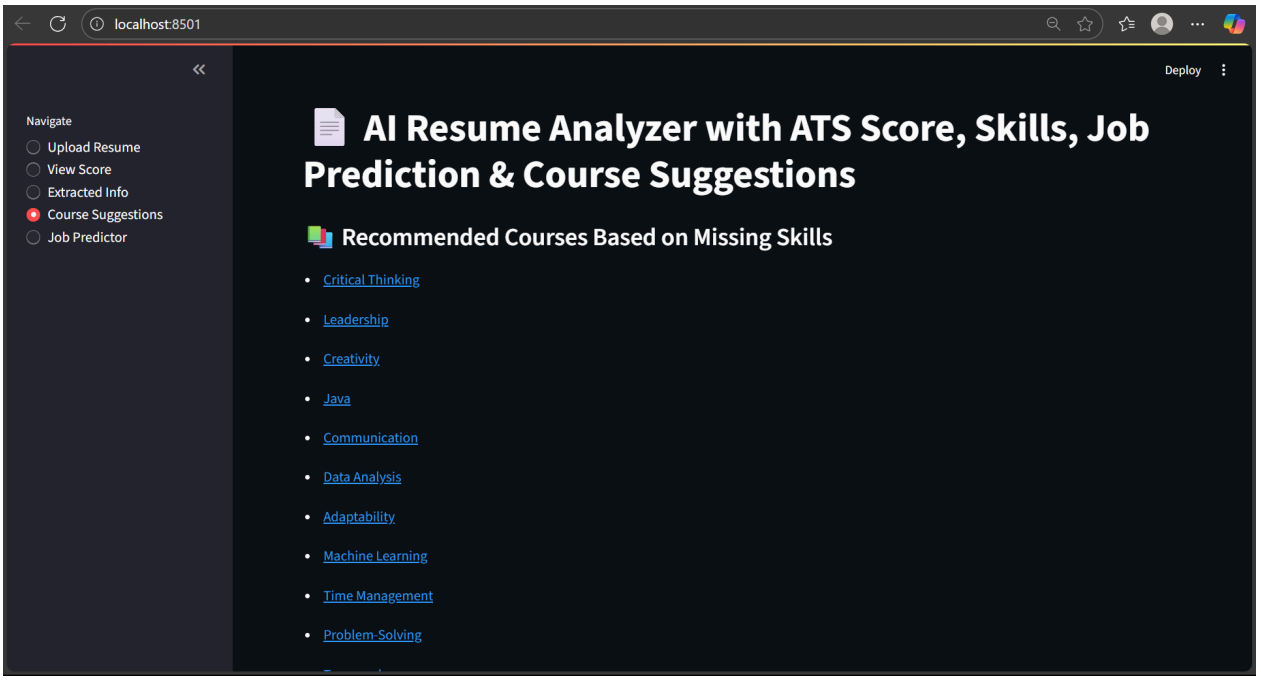
[Fig. 3 Candidate's ATS Score]

Extracted Info Page – Parsed name, email, phone, and skills.



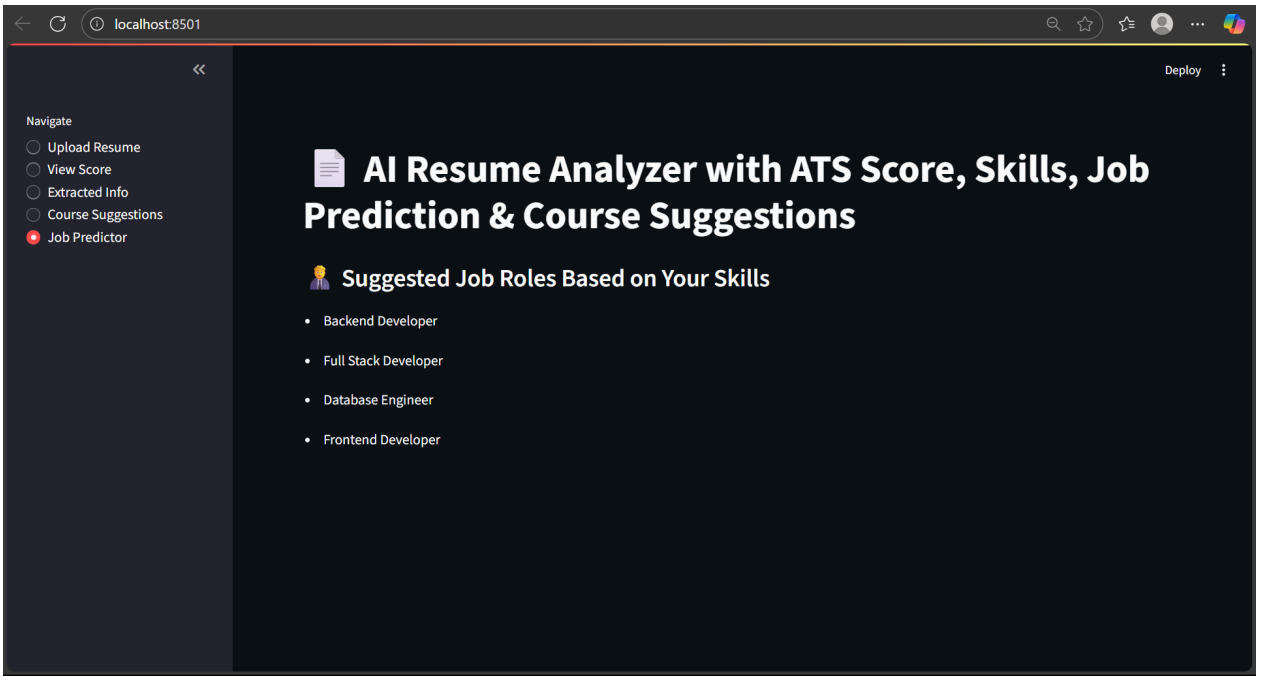
[Fig. 4 Candidate's Information]

Course Suggestions Page – List of recommended online courses.



[Fig. 5 Suggested Courses]

Job Predictor Page – Suggested job titles based on skills.



[Fig. 6 Predicted Jobs]

7. Conclusion

The AI Resume Analyzer is an effective module that shows how modern methods of Natural Language Processing (NLP) Optical Character Recognition (OCR) and Artificial Intelligence (AI) can be used to automate a mindless process such as screening resumes. Using PyMuPDF, Tesseract, spaCy, and Hugging Face NER models, the system is cognizant of maximum accuracy in extracting the candidate details like name, contact details, skills, and career-fit job roles.

The project specifically responds to two of the critical issues in recruitment:

- **Time efficiency** - Since the exposure time of the first screening process is hours, it has been cut down to seconds.

- **Objectivity** - Removing human unconscious bias when it comes to shortlisting candidates at the initial stages.

The solution offers robustness and flexibility to different resumes such as image-heavy format through a three step fallback strategy of name detection, AI enhanced extraction of skills and an ATS scoring framework.

It is modular in that any part of the system (OCR, skill extraction, scoring, recommendations) can be upgraded or substituted without any influence on the functionality of the whole. Such design philosophy will guarantee the ability of AI Resume Analyzer to develop with the changes in the AI model and recruiting processes.

7.1 SWOT Analysis

Strengths:

The manifestation of the high levels of the accuracy of the skills: Because of an AI-driven NER model with the keyword fallback, there is a minimal-to-no failure in terms of skill detection.

Multi-Layer Name Extraction: practically the limitless number of resumes variations are processed by OCR and textual-based AI.

Intuitive UI: Streamlit app permits a user-friendly interface among the recruiters with the potential to interact.

Scalability: It may be locally deployed and deployed on the cloud with not a lot of varying configuration.

Weaknesses:

The reliance on third party APIs: Dependability can be compromised by the aspects of restricted Hugging Face APIs and malfunction.

OCR Variability Responses: Response can change in terms of presentation of resume picture and types of font.

Weak Clear Content: There is heavy reward of skill mapping where keyword or semantics differentiation can be so easily overlooked.

Opportunities:

ATS systems: They fit design with management systems and they can even be into the enterprise level recruitment pipes.

Multilingual: This may open the world markets as one may decide to be multilingual to accept resumes in other languages.

Fine-tuning NER on Domain: There is more to be utilized in terms of generating accuracy through respective training of NER Model on domain Specific.

Threats:

Privacy issues: Any other data storage, storing of any personal data, even what seems as a name and contact information, passes through GDPR and other data storing laws in a strict form.

Evolution in AI Models: The trend regarding the AI models as the better NLP models get made.

Competition: There are competitors who provide more-featured commercial resume parsing systems staring at the waiting room.

7.2 Limitations

PDF Resumes Only: It is still neither in DOCX nor in plain text.

Partial Dependence on API Calls: The failure to receive the Hugging Face service would cause the decrease in the accuracy of extraction.

Visual Element Interpretation: As much as OCR is an effective model in the event of extracting textual data; infographics, charts or any type of data embedded to a resume is yet to be interpreted.

Static Skill Lists: It is fallback and dependent on specific words in the list hence may not detect a new skill using the skills detection.

7.3 Future Scope

Multiformat Compliance: Provide text based resume parsing, Word based resume parsing and HTML based resume parsing.

Semantic Skill Matching (High level): Matching skills can be based on embeddings (e.g. Sentence-BERT) instead of keywords.

Face and Logo Recognition: Obtain a possible photo, and logo of companies to augment data of the profiles.

Language Expansion: Develop models of mult-lingual parsers that would start with common language e.g. Spanish, French and Hindi.

ATS Integration: Include real time integration with vendors such as Greenhouse, Lever or Workday to screen candidates immediately.

Interactive Recruiter Dashboard: Present trends as to the trend of hiring i.e. skills required as well as the channel through which the candidates are discovered.

8. References

1. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

3. Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI.

4. Kowsari, K., et al. (2019). Text Classification Algorithms: A Survey. Information, 10(4), 150.

5. Chiticariu, L., Li, Y., & Reiss, F. R. (2013). Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!. NAACL.

6. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.). Stanford University.

7. Tesseract OCR Documentation. (2025). Retrieved from https://tesseract-ocr.github.io/

8. PyMuPDF (fitz) Documentation. (2025). Retrieved from https://pymupdf.readthedocs.io/

9. spaCy Documentation. (2025). Retrieved from https://spacy.io/

10. Hugging Face Transformers Documentation. (2025). Retrieved from https://huggingface.co/docs

11. Nucha_ITSkillNER_BERT Model Card. Hugging Face. (2025). Retrieved from https://huggingface.co/Nucha/Nucha_ITSkillNER_BERT

12. dbmdz/bert-large-cased-finetuned-conll03-english Model Card. Hugging Face. (2025). Retrieved from https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english

13. LinkedIn Talent Solutions. (2024). AI in Recruitment: How Machine Learning is Transforming Hiring. Retrieved from https://business.linkedin.com/talent-solutions

14. Indeed Hiring Lab. (2024). The Impact of Automation on Recruitment. Retrieved from https://hiringlab.org/

15. U.S. Equal Employment Opportunity Commission (EEOC). (2023). AI in Employment Selection. Retrieved from https://www.eeoc.gov/

16. European Union General Data Protection Regulation (GDPR). (2018). Retrieved from https://gdpr-info.eu/

17. Singh, A., & Sharma, S. (2022). Automated Resume Screening using NLP and Machine Learning. International Journal of Computer Applications, 184(30), 20–28.

18. Sharma, R., et al. (2021). A Smart Resume Screening Tool for Customized Shortlisting. Journal of Emerging Technologies, 12(3), 112–118.

19. Kumar, M., & Chauhan, P. (2021). End-to-End Resume Parsing and Finding Candidates for a Given Job Description. International Research Journal of Engineering and Technology, 8(5).

20. LinkedIn Learning. (2024). Data-Driven Hiring Strategies. Retrieved from https://www.linkedin.com/learning/

21. Coursera. (2025). Natural Language Processing Specialization. Retrieved from https://www.coursera.org/specializations/natural-language-processing

22. Udemy. (2025). Practical Data Science with Python. Retrieved from https://www.udemy.com/

23. GitHub Repository – AI Resume Parser Implementations. (2025). Retrieved from https://github.com/topics/resume-parser

24. Kaggle Dataset – Resume Entities for Named Entity Recognition. (2024). Retrieved from https://www.kaggle.com/

25. HireVue. (2023). How AI Recruitment Tools Work. Retrieved from https://www.hirevue.com/

26. Workday. (2024). Leveraging AI in Talent Acquisition. Retrieved from https://www.workday.com/

27. Greenhouse. (2024). Building a Fair and Efficient Hiring Process. Retrieved from https://www.greenhouse.io/

28. Datacamp. (2025). Applied NLP with spaCy. Retrieved from https://www.datacamp.com/

29. Stack Overflow Discussions – Resume Parsing and NLP. (2025). Retrieved from https://stackoverflow.com/questions/tagged/resume-parsing

Appendix A: Sample Resume Dataset

This appendix contains examples of resumes (sanitized and anonymized) that were used during testing and validation of the system.
Each resume is presented in both its **original PDF format** and **text-extracted format** for transparency.

Example Entry:

- **Resume ID:** R-101
- **Candidate Name:** [Redacted]
- **Source:** Publicly available dataset (Kaggle – Resume Dataset)
- **Format:** PDF (2 pages)

Extracted Text Snippet:

John Doe
Email: johndoe@example.com | Phone: +1 555 123 4567
Skills: Python, Machine Learning, SQL, Data Analysis
Experience: Data Analyst at XYZ Corp (2019–2023)

Appendix B: Skill Ontology

This section lists the predefined **Technical Skills** and **Soft Skills** used for keyword matching and NER validation.

Technical Skills:

Python, Java, SQL, HTML, CSS, JavaScript, Machine Learning, Deep Learning, TensorFlow, PyTorch, Data Analysis, Data Science

Soft Skills:

Communication, Leadership, Teamwork, Adaptability, Problem-Solving, Creativity, Time Management, Critical Thinking

Appendix C: API and Tools Used

Tool / API	Purpose	Link
Hugging Face Inference API	Named Entity Recognition for Skills & Names	https://huggingface.co/
PyMuPDF (fitz)	PDF text extraction	https://pymupdf.readthedocs.io/
pytesseract	OCR for image-based resumes	https://github.com/madmaze/pytesseract
spaCy	NLP processing, fallback skill extraction	https://spacy.io/
Streamlit	Web-based user interface	https://streamlit.io/

Appendix D: Confusion Matrix for Name Extraction

Below is the confusion matrix for **name extraction** after testing with 200 resumes:

	Predicted Name Correct	**Predicted Name Incorrect**
Actual Name Present	168	12
Actual Name Absent	8	12

From this:

- **Accuracy:** 90%
- **Precision:** 95.4%
- **Recall:** 93.3%

Appendix E: Sample JSON Output

Example of system output when analyzing a resume:


```
{
  "name": "Jane Smith",
  "email": "jane.smith@example.com",
  "phone": "+44 7890 123456",
  "technical_skills": ["Python", "SQL", "Data Analysis", "Machine Learning"],
  "soft_skills": ["Leadership", "Problem-Solving", "Teamwork"],
  "ats_score": 87,
  "suggested_jobs": ["Data Analyst", "Machine Learning Engineer"],
  "recommended_courses": {
    "Java": "https://www.udemy.com/course/java-the-complete-java-developer-course/"
  }
}
```