**SUMMER INTERNSHIP REPORT**

**Computer Science and Engineering**

**JIIT Noida**



**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY**

B TECH – Ist Year

(2-Credit Internship)

TITLE: JOBfit

Internship Duration: 9 June, 2025 – 5 July, 2025

GROUP-8

Submitted by: 1) DHRUVI SHARMA (992401030276)

2) SOUMYA SONEJA (2401020061)

3) AARUSH BHAGUNA (2401200026)

4) ANSHIKA SRIVASTAV (2405170095)

Submitted to: Ms. NEETU SINGH

**DECLARATION**

I hereby declare that the Summer Internship Project Report titled **"JOBfit: Skill Gap Analysis using ML and IR"** submitted to **Jaypee Institute of Information Technology, Noida** is a record of original work carried out by me during the summer internship as a part of our academic curriculum.

This project was undertaken as a **group project** by the GROUP 8 students. The work presented in this report is the result of our own efforts and has been conducted under the guidance of Ms. Neetu Singh, and has not been submitted elsewhere for any other academic purpose.

Soumya Soneja

2401020061

5 July 2025

**CERTIFICATE**

**This is to certify that Soumya Soneja, Roll No. 2401020061, has successfully completed the Summer Internship Project titled “JOBfit” during the period June 9 – July 5, 2025, under my supervision.**

**Ms. Neetu Singh**

**Department of CSE**

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to everyone who supported and guided me throughout the successful completion of my summer internship project titled **"JobFit: Skill Gap Analysis using ML and IR."**

First and foremost, I am deeply thankful to **Jaypee Institute of Information Technology** for providing me with the opportunity to undertake this project as a part of my academic curriculum.

I express my heartfelt appreciation to my faculty mentor, **Ms. Neetu Singh**, for her constant support, expert guidance, and valuable feedback at every stage of the project. Her mentorship played a crucial role in shaping my ideas and refining my approach.

I am also grateful to the **industry mentors and professionals** who shared their practical insights and helped me understand the real-world applications of Machine Learning and Information Retrieval in skill gap analysis.

A special note of thanks goes to my **teammates** for their continuous encouragement and support throughout the project duration.

Last but not least, I extend my gratitude to **all contributors—direct or indirect—who played a role in helping me bring this project to completion.**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **Topic** | **Page Number** |
| Introduction | 5 |
| Objectives of the Internship | 6-7 |
| Problem Overview | 8 |
| Tools and Technologies Used | 9 |
| Weekly Progress Summary | 10-11 |
| System Design/Workflow/Architecture | 12-14 |
| Implementation Details | 15-16 |
| Output Samples | 17-19 |
| Challenges Faced and Solutions | 20-21 |
| Key learnings | 22 |
| Conclusion | 23 |
| References | 24 |

**INTRODUCTION**

The project titled “JOBfit” focuses on the integration of Machine Learning (ML) and Information Retrieval (IR) techniques to build an efficient job recommendation and matching system. In the current digital era, finding suitable job opportunities that align with an individual’s skillset and preferences has become a challenging task. Many job seekers struggle with identifying the right jobs that match their qualifications, interests, and goals.

The primary objective of this project is to simplify this process by creating an intelligent system that analyzes job descriptions and candidate profiles to provide accurate job recommendations. By using ML algorithms and IR methods, the system can retrieve relevant job postings and suggest the best-fit positions to users.

This topic was selected for the internship as it addresses a highly relevant and practical problem faced by millions of job seekers worldwide. Additionally, it provided an opportunity to apply theoretical knowledge of ML and IR in a real-world scenario, enhancing both technical and analytical skills.

**OBJECTIVES**

The main objectives of this internship project, titled “JOBfit”, are as follows:

1. To develop an intelligent system that recommends suitable job opportunities based on user profiles, skills, and preferences.

2. To apply Machine Learning (ML) algorithms and Information Retrieval (IR) techniques for job matching and ranking.

3. To automate the process of filtering and retrieving relevant job postings from large datasets.

4. To enhance the accuracy and efficiency of job recommendations through continuous model improvement.

5. To gain hands-on experience in building real-world ML & IR-based applications.

6. To strengthen practical knowledge of data preprocessing, model building, evaluation, and deployment.

7. To solve a real-life problem faced by job seekers and recruiters using technical solutions.

**PROBLEM OVERVIEW**

In today’s competitive job market, job seekers face difficulties in finding job opportunities that align with their skills, interests, and career goals. Similarly, companies struggle to identify the right candidates from a large pool of applicants.

The project “JOBfit” addresses this challenge by developing a smart job recommendation system that matches job seekers with suitable job roles. The system analyzes both job descriptions and user profiles using advanced Machine Learning and Information Retrieval techniques to provide personalized job suggestions.

This project simulates a real-world solution that could be used by recruitment firms, job portals, and professional networking platforms.

**TOOLS AND TECHNOLOGIES USED**

The following tools, technologies, and programming languages were used in the development of the JOBfit system:

Python (Programming Language)

Scikit-learn (Machine Learning Library)

Pandas (Data Manipulation)

NumPy (Numerical Computing)

NLTK (Natural Language Processing, for text preprocessing)

TF-IDF Vectorizer (For text feature extraction)

Cosine Similarity (For matching and ranking)

Jupyter Notebook (Development Environment)

Matplotlib (Data visualization)

Ranking Metrices (Precision, Recall, F1, NDCG, MAP)

Ranking Model (LambdaMART)

**WEEKLY PROGRESS SUMMARY**

Week 1:

Understanding the project objectives and defining the scope of the JOBfit system.

Research on existing job recommendation systems and their working methodologies.

Familiarization with Machine Learning and Information Retrieval concepts relevant to the project.

Week 2:

Data collection and preparation of sample datasets for job descriptions and candidate profiles.

Preprocessing of text data, including tokenization, stop word removal, and stemming/lemmatization.

Implementation of TF-IDF vectorization to convert text into numerical features.

Week 3:

Building the core recommendation model using cosine similarity to match jobs with profiles.

Testing and evaluation of the system on sample queries and datasets.

Fine-tuning of preprocessing and feature extraction steps to improve results.

Week 4:

Addition of advanced filtering techniques to improve relevance of recommendations.

Preparing project documentation, including system architecture, flow diagrams, and results.

Final review and completion of the internship report.

**SYSTEM DESIGN/WORKFLOW/ARCHITECTURE**

**WORKFLOW:**

The overall workflow of the JOBfit system involves the following steps:

1. Data Collection:

Job Descriptions Dataset

Candidate Profiles Dataset

2. Data Preprocessing:

Text Cleaning (removing special characters, punctuation, etc.)

Tokenization

Stop Word Removal

Stemming or Lemmatization

3. Feature Extraction:

TF-IDF Vectorization to convert text into feature vectors.

4. Similarity Computation:

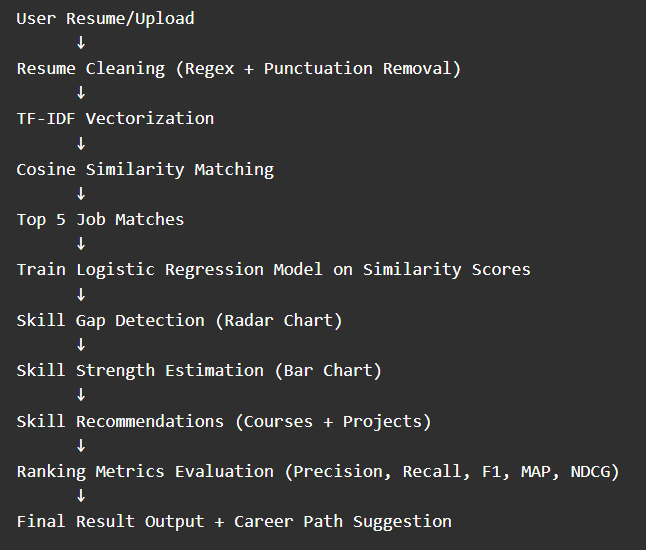
Cosine Similarity to calculate the similarity between job descriptions and user profiles.

5. Ranking and Recommendation:

Ranking job listings based on similarity scores.

Presenting the top job recommendations to the user.

**ARCHITECTURE**



**IMPLEMENTATION DETAILS**

The implementation of the JOBfit system involved the following key steps:

Step 1: Data Collection

Sample job descriptions and candidate profiles were collected for testing the system. The dataset included text-based job requirements and skill sets of users.

Step 2: Data Preprocessing

Text Cleaning: Removal of special characters, extra spaces, and punctuations.

Tokenization: Splitting text into individual words/tokens.

Stop Word Removal: Removing common words that do not contribute much to meaning (e.g., "the", "is", "in").

Stemming/Lemmatization: Reducing words to their root forms to improve matching.

Step 3: Feature Extraction using TF-IDF

TF-IDF Vectorizer was used to convert preprocessed text into numerical feature vectors.

This step enabled the system to capture the importance of words in both job descriptions and user profiles.

Step 4: Similarity Computation

Cosine Similarity was applied to compare the TF-IDF vectors of job descriptions and candidate profiles.

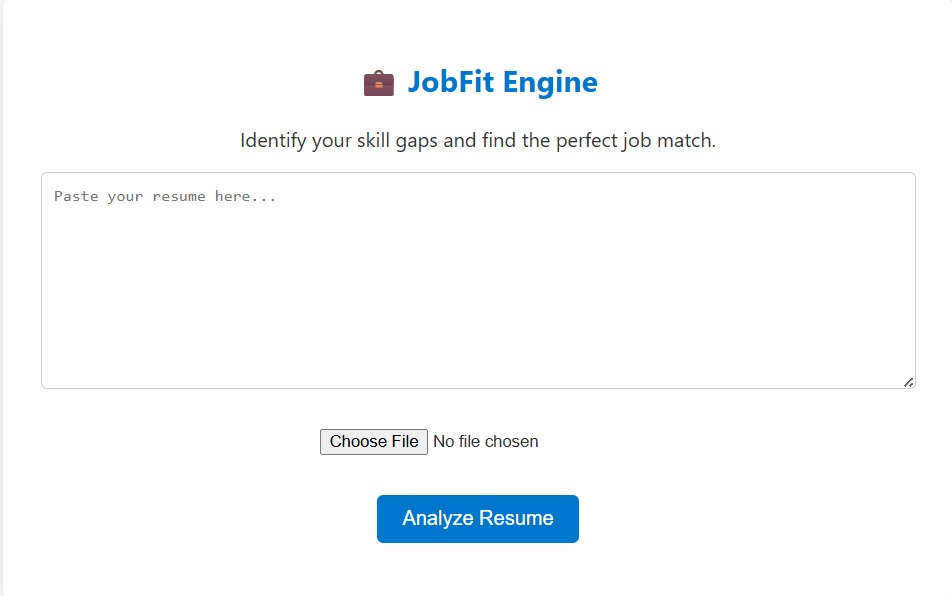
The similarity score indicated how well a particular job matched the user's profile.

Step 5: Job Recommendation

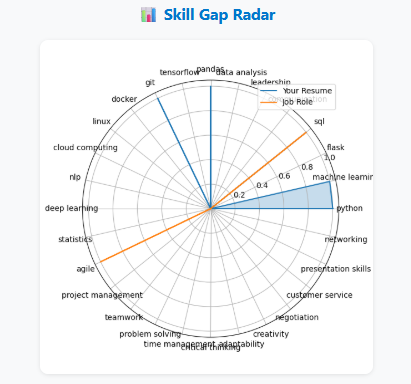
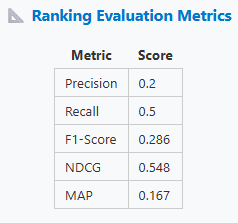
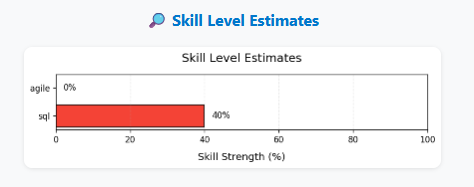
Jobs were ranked based on similarity scores.

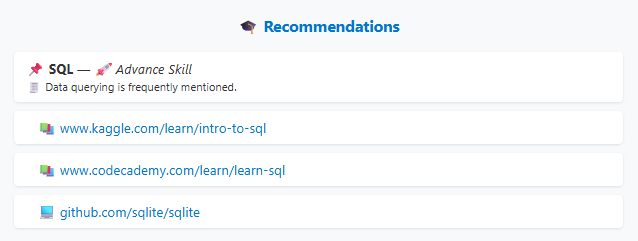
The top matching job recommendations were displayed as output.

**OUTPUT SAMPLES**







**CHALLENGES FACED AND SOLUTIONS**

**1. Inconsistent Resume and Job Description Formats**

**Challenge:**  
Resumes and job postings vary a lot in structure, grammar, and vocabulary. Extracting meaningful skills from unstructured text was difficult.

**Solution:**

* Used **regex** and **text cleaning techniques** (HTML tag removal, lowercasing, punctuation removal).
* Applied **TF-IDF vectorization** to normalize and extract features from both resumes and job descriptions.

**2. Identifying Relevant Skills in Text**

**Challenge:**  
Mapping natural language to specific skills (like leadership, Python, Docker) is non-trivial, especially without labeled datasets.

**Solution:**

* Created a **predefined skill list** and performed **keyword-based matching**.
* Could be improved in future with **Named Entity Recognition (NER)** or **BERT-based embeddings**.

**3. Generating Visual Feedback (Charts)**

**Challenge:**  
Creating effective visualizations (radar and bar charts) with many skills led to cluttered visuals or misrepresentation.

**Solution:**

* Dynamically adjusted the skill set based on overlap between resume and job.
* Used **Matplotlib** with polar plots and bar charts, and saved images server-side for rendering in Flask.

**4. Model Accuracy Without Labeled Dataset**

**Challenge:**  
No labeled data to train a model on what constitutes a good or bad match.

**Solution:**

* Created a **pseudo-labeling scheme** using cosine similarity thresholding (0.3 as a cutoff).
* Trained a **Logistic Regression model** on these generated labels for basic ranking.

**5. Performance Bottlenecks with Large Dataset**

**Challenge:**  
Similarity calculation and chart generation could slow down with a large number of resumes/jobs.

**Solution:**

* Limited job match output to **Top 5 results**.
* Used **vectorized operations with NumPy and SciKit-learn** for performance.

**6. Noisy Data / Non-Informative Resumes**

**Challenge:**  
Some resumes were either empty, improperly formatted, or too short to produce meaningful analysis.

**Solution:**

* Added checks to **skip empty or invalid rows** during training.
* Used try/except blocks to make the model robust and avoid crashing on edge cases.

**7. Evaluation of Recommendations**

**Challenge:**  
It was hard to measure if recommendations were truly helpful or accurate.

**Solution:**

* Introduced standard **ranking metrics** (Precision, Recall, F1, NDCG, MAP) to simulate evaluation.
* This allows rough performance estimation without labeled ground truth.

**KEY LEARNINGS**

**1. Real-World Data Handling**

* Learned how to **clean and preprocess unstructured text data** (HTML tags, punctuation, casing).
* Understood the challenges of **messy or missing values** in real datasets.

**2. Natural Language Processing (NLP)**

* Gained practical experience with **TF-IDF Vectorization** for converting text into numerical form.
* Understood **cosine similarity** for matching resumes with job descriptions.

**3. Machine Learning Workflow**

* Implemented a full ML pipeline: **data processing → feature extraction → model training → evaluation**.
* Used **Logistic Regression** for classification and ranking purposes.
* Learned how to handle **pseudo-labeling** when actual labels are unavailable.

**4. Evaluation Metrics for Ranking Systems**

* Understood and applied **ranking metrics**:  
  Precision, Recall, F1-Score, NDCG, and MAP.
* Learned how to interpret and use these metrics to evaluate resume-job match quality.

**5. Flask Web Development**

* Built a **user-friendly Flask app** that takes resume input, processes it, and returns visual + textual output.
* Learned how to **integrate ML models into a web backend**.

**6. Data Visualization**

* Created **radar and bar charts** to represent skill gaps visually.
* Understood how visual feedback helps users quickly assess strengths and weaknesses.

**7. Team Collaboration and Version Control**

* Practiced **task delegation** and role division (frontend/backend, model, design).
* Used tools like **Git** for version control and team coordination.

**8. Software Robustness**

* Learned how to **write resilient code** with exception handling, validations, and fallback mechanisms.
* Built a **modular and scalable structure** for future enhancements (e.g., PDF parsing, GPT integration).

**CONCLUSIONS**

The JOBfit project successfully demonstrated the use of Machine Learning and Information Retrieval for solving a real-world problem—job recommendations and matching. Through this internship, the core objective of developing an intelligent, automated job recommendation system was achieved.

The project helped in gaining valuable technical skills, such as text preprocessing, vectorization, and similarity-based ranking, as well as non-technical skills like project planning, documentation, and problem-solving.

This internship experience has strengthened both theoretical and practical understanding of key concepts in ML and IR, and has laid a solid foundation for future projects in this domain.

**REFERENCES**

1. Scikit-learn Documentation — https://scikit-learn.org/

(Used for TF-IDF and similarity computation.)

2. NLTK Documentation — https://www.nltk.org/

(Used for text preprocessing tasks.)

3. Python Official Documentation — https://docs.python.org/3/

(Used for basic Python functionalities.)