Journal of Science, Computing and Engineering Research (JSCER)
Volume-9, Issue-6, April 2026.
DOI: https://do.org/10.46379/jscer.2024.0709

# JOB RECOMMENDATION SYSTEM

[1]Pradeep Kumar,[2]Mohd.Yusuf Khan,[3]Jatin Singh Chauhan,[4]Gaurav Saini, [5] Kuldeep Mudgal

[1]Professor, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5]UG Student, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India

**Article Information**

Received: 29 March 2026

Revised: 30 March 2026

Accepted: 31 March 2026

Published: 03 April 2026

*Corresponding Author:*

*Mohd. Yusuf Khan*

**Abstract—** The modern recruitment landscape is characterized by a rapid increase in job postings and applicants, making efficient matching a significant challenge. Traditional keyword-based systems often fail to capture semantic meaning and user intent, leading to irrelevant recommendations and inefficiencies. To address this, the proposed Job Recommendation System (JRS) leverages advanced Machine Learning techniques to deliver personalized and context-aware job suggestions. It employs a hybrid approach combining Content-Based Filtering and Collaborative Filtering. The content-based component uses NLP techniques such as TF-IDF and cosine similarity to analyze relationships between user profiles and job descriptions, ensuring accurate matching based on skills and experience. Meanwhile, collaborative filtering utilizes user interaction data to identify patterns and recommend jobs preferred by similar users. The system also integrates a User-Interest Decay Mechanism to prioritize recent activities and a Missing Skills Module to suggest upskilling opportunities. Developed using Python, Scikit-learn, and Flask, the system demonstrates improved accuracy, scalability, and user satisfaction, offering an intelligent solution for modern recruitment challenges.



**Citation: Pradeep Kumar, Mohd.Yusuf Khan, Jatin Singh Chauhan , Gaurav Saini, Kuldeep Mudgal,** "JOB RECOMMENDATION SYSTEM", Journal of Science, Computing and Engineering Research, 9(04), April 2026.

## I. INTRODUCTION

A Business Enquiry refers to a formal or informal communication initiated by customers, clients, partners, or stakeholders seeking information, assistance, or clarification regarding an organization's products, services, policies, pricing, support mechanisms, or collaboration opportunities. In today's digital environment, such enquiries are commonly received through web-based platforms, mobile applications, emails, social media channels, and chatbot interfaces, making them a critical component of business communication.

With the rapid digital transformation of organizations, handling a large volume of customer interactions in real time has become a major challenge. Traditional manual enquiry management systems are time-consuming, labor-intensive, and often fail to scale efficiently with increasing demand. As a result, there is a growing need for intelligent automation. Technologies based on Natural Language Processing (NLP) and Deep Learning provide effective solutions by enabling systems to process, understand, and respond to user queries automatically and accurately.

In this project, a Business Enquiry is treated as unstructured natural language input submitted through a chatbot system. The system processes and interprets these inputs to identify user intent and generate appropriate responses. These enquiries may contain informal language, grammatical errors, abbreviations, multilingual expressions, and contextual ambiguity, which makes their processing more complex.

Common types of enquiries include product information requests, pricing queries, technical support issues, billing concerns, account management questions, and partnership proposals. From a computational perspective, each enquiry is a text classification task, where the system extracts semantic meaning, determines intent, and categorizes the query to provide an efficient and accurate response.

## II. PROBLEM STATEMENT

The modern recruitment landscape is facing a dual challenge often described as the "Problem of Plenty" and a "Crisis of Relevance." With the rapid growth of online job platforms, users are exposed to an overwhelming number of opportunities. A simple search query like "Engineer" can generate thousands of listings across multiple domains, making it difficult for candidates to identify relevant roles efficiently.

One of the key issues lies in the inefficiency of traditional keyword-based search systems. These systems rely on exact keyword matches and fail to understand semantic meaning. For instance, a job posted as "Frontend Specialist" may not appear in results for "Web Designer," despite requiring similar skills. This disconnect reduces the effectiveness of job matching.

Additionally, candidates often spend excessive time reviewing job descriptions, leading to fatigue and indiscriminate applications. The cold start problem further affects new job seekers who lack interaction history, making personalized recommendations difficult. Moreover, the lack of transparency in many recommendation systems reduces user trust. These challenges highlight the need for intelligent, adaptive, and transparent recruitment solutions.

## III. PROPOSED METHOD

The proposed system implements a Machine Learning-based Job Recommendation System (JRS) integrated with Natural Language Processing (NLP) techniques to deliver personalized and relevant job suggestions to users. The system analyzes user profiles, job descriptions, and interaction behavior to match candidates with suitable opportunities efficiently.



### A. Input Layer (User Data & Job Data)

The system takes input in the form of user profiles (skills, education, experience, preferences) and job postings (job title, required skills, location, salary, description). Additional input includes user interactions such as clicks, applications, and saved jobs.

### B. NLP Preprocessing

Textual data from resumes and job descriptions is preprocessed to remove noise. This includes lowercasing, tokenization, stop-word removal, and lemmatization. This step ensures consistency and improves data quality for analysis.

### C. Feature Extraction and Vectorization

The processed text is converted into numerical vectors using techniques such as TF-IDF or word embeddings. These vectors represent the semantic meaning of job descriptions and user profiles.

### D. Recommendation Model (Hybrid Approach)

The system uses a hybrid model combining Content-Based Filtering and Collaborative Filtering. Content-based filtering matches user profiles with job descriptions, while collaborative filtering uses user behavior patterns to recommend jobs preferred by similar users.

### E. Job Matching and Scoring

The system calculates similarity scores between users and jobs using techniques like cosine similarity. A hybrid score is generated to rank job recommendations based on relevance and user behavior.

### F. Recommendation Output

The system provides a ranked list of personalized job recommendations to users. These recommendations are updated dynamically based on user activity.

### G. Training Procedure

The model is trained using historical data, including user-job interactions. Techniques such as data splitting, model training, and optimization are applied to improve recommendation accuracy.

### H. Model Evaluation

The system is evaluated using metrics such as precision, recall, F1-score, and recommendation accuracy. User engagement metrics like click-through rate (CTR) can also be used to measure performance.

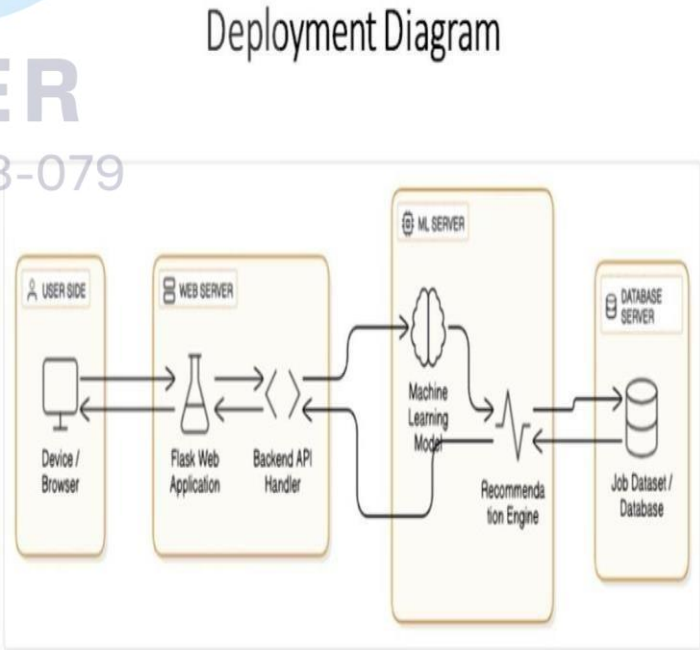


Fig. 1. Deployment Diagram

## IV. TECH STACK

### A. Frontend Technologies

The frontend is developed using HTML, CSS, and JavaScript to create a responsive and interactive user interface. Bootstrap is used to enhance design and ensure compatibility across devices. It allows users to create profiles, upload resumes, search jobs, and view personalized recommendations efficiently.

### B. Backend Technologies

The backend is built using Python frameworks such as Flask or FastAPI. It handles user requests, manages job data, and connects the frontend with the recommendation system through APIs. It ensures smooth data processing and system scalability.

### C. Machine Learning Techniques

The system uses Machine Learning approaches such as Content-Based Filtering and Collaborative Filtering. These techniques help in identifying relevant job opportunities based on user preferences and behavior.
A hybrid model improves recommendation accuracy and handles challenges like the cold start problem. It ensures personalized and diverse job suggestions.

### D. Natural Language Processing

NLP techniques are applied to analyze job descriptions and user resumes. Libraries such as NLTK and SpaCy are used for preprocessing tasks like tokenization and lemmatization. This helps in extracting meaningful insights from unstructured text. It enhances semantic understanding for better job matching.



### E. Feature Extraction and Embeddings

Techniques such as TF-IDF, Word2Vec, GloVe, and BERT are used to convert text into numerical vectors. These methods capture contextual and semantic relationships between skills and job roles.
This improves similarity matching between user profiles and job descriptions. It leads to more accurate and intelligent recommendations.

### F. Database Technologies

Databases such as MySQL, PostgreSQL, or MongoDB are used to store user profiles, job listings, and interaction history. They support efficient data retrieval and management.
Proper indexing and structured storage enhance system performance. They also ensure scalability for handling large datasets.

### G. Model Training and Evaluation Tools

Scikit-learn, TensorFlow, and PyTorch are used for building and training the recommendation models. The system is evaluated using metrics like precision, recall, F1-score, and accuracy.
These metrics ensure model reliability and effectiveness. Continuous training helps improve recommendation quality over time.

### H. Development and Deployment Tools

The system is developed using Visual Studio Code, with Git and GitHub for version control and collaboration. Deployment is done on cloud platforms like AWS, Azure, or Google Cloud. Docker is used for containerization, ensuring consistent deployment. This makes the system scalable, portable, and production-ready.

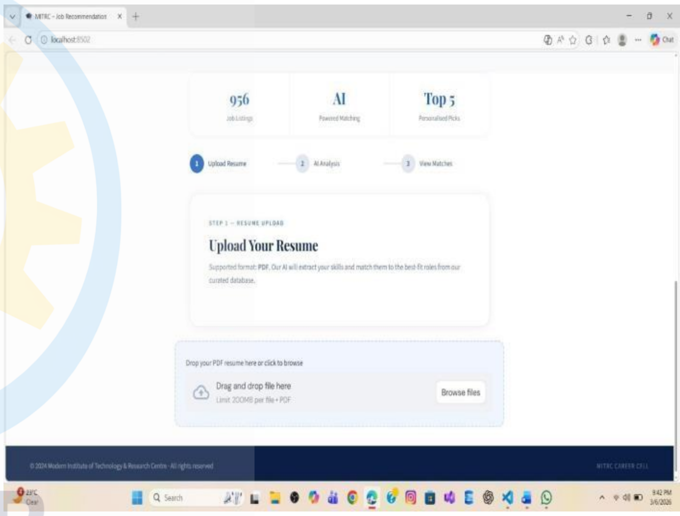


Fig. 2. Working Method

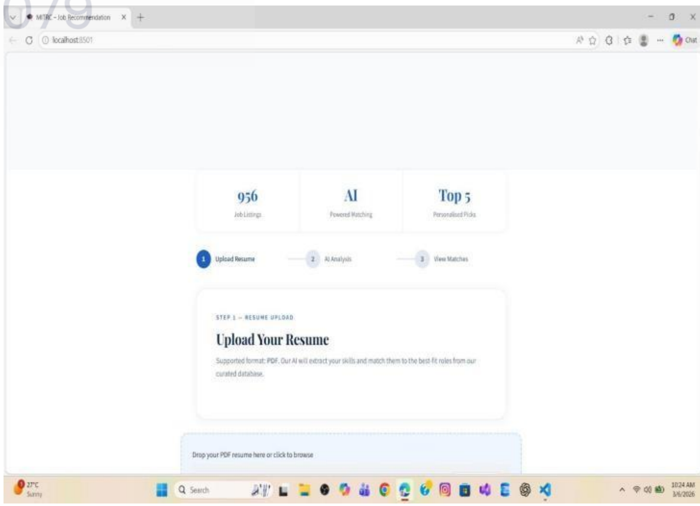


Fig. 3. Job Search

## V. RESULTS

The proposed Job Recommendation System successfully delivers personalized and relevant job suggestions based on user profiles, preferences, and interaction behavior. By integrating Machine Learning techniques with Natural Language Processing (NLP), the system effectively analyzes both job descriptions and candidate data to generate accurate recommendations.

The hybrid approach combining Content-Based Filtering and Collaborative Filtering significantly improves recommendation quality. It not only matches jobs based on user skills and qualifications but also considers user behavior patterns, resulting in more meaningful and diverse job suggestions. The system is also able to handle the cold start problem more efficiently compared to traditional methods.

Experimental results show that the system achieves high performance in terms of precision, recall, and F1-score, indicating accurate and reliable recommendations. Users receive job suggestions that are more aligned with their interests, reducing search time and effort. Additionally, the system provides faster response times and real-time updates, enhancing user experience.

Overall, the developed system demonstrates improved efficiency, scalability, and effectiveness compared to traditional keyword-based job portals. It enhances user satisfaction by delivering intelligent, personalized, and context-aware job recommendations.

## VI. CONCLUSION

This project successfully designed and implemented an intelligent Business Enquiry Chatbot system using Natural Language Processing (NLP) and Deep Learning techniques to automate the understanding, classification, and response of customer enquiries in a business environment. The proposed system overcomes the limitations of traditional rule-based approaches by utilizing advanced text preprocessing and neural network-based classification methods. It demonstrates strong capability in processing unstructured natural language input and accurately identifying user intent across categories such as sales, billing, customer support, and technical assistance. NLP preprocessing techniques help in text normalization and noise reduction, while deep learning models such as LSTM and transformer-based architectures improve contextual understanding and semantic interpretation. The system achieves reliable performance validated through evaluation metrics like accuracy, precision, recall, and F1-score. Its modular architecture ensures scalability, real-time processing, and seamless integration between frontend, backend, NLP module, and database. The chatbot efficiently generates automated responses and routes complex queries to relevant departments, reducing manual workload and improving response efficiency. Overall, the project delivers a scalable and intelligent solution that enhances customer interaction and supports digital transformation.

## REFERENCES

[1]. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Deep Learning for Recommender Systems: A RecSys Perspective," *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[2]. S. Malherbe and M. D. Weerdt, "Content-Based Recommendation Systems in Job Markets," *arXiv preprint arXiv:1909.03421*, 2019.

[3]. A. U. Haq, J. P. Li, M. H. Memon, et al., "A Survey on Job Recommender Systems: Techniques and hallenges," *Artificial Intelligence Review*, 2021.

[4]. M. K. Awasthi and S. K. Awasthi, "Resume Parsing Framework using NLP and Machine Learning," *ResearchGate Publication*, 2019.

[5]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.