IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

Automated Job Search Using Resume Keywords VIA APIs

Parth Dhake[1], Shreyash Kale[2], Preet Patel[3], Christina Padale[4], Prof. N. V. Keskar[5]
Department of Computer Engineering[1-5]
Matoshri Aasarabai Institute of Technology and Research Centre, Eklahare, Nashik Maharashtra, India

Abstract: *This project presents an Automated Job Search System that uses Natural Language Processing [2] (NLP) to extract key information from resumes and match it with real-time job listings through APIs. The system parses unstructured resume documents to identify structured details such as skills, qualifications, experience, and job roles. These extracted keywords are then compared with job descriptions retrieved from external Job Search APIs.*
Unlike traditional keyword-based search systems, the proposed solution incorporates semantic matching [9] techniques to analyze contextual similarity between resumes and job postings. This improves recommendation accuracy and reduces irrelevant results. The architecture combines resume parsing, keyword extraction, API-based job retrieval, and ranking algorithms to create a complete automated job-matching pipeline. By automating resume analysis [8] and job searching, the system reduces manual effort, enhances efficiency, and provides personalized job recommendations [6]. The scalable and API-driven design makes it suitable for integration with recruitment platforms and career guidance systems..

Keywords: Resume Parsing, Job Matching APIs, Natural Language Processing [2], Semantic Matching, Recommendation System

I. INTRODUCTION

The Tech Hire is an AI-powered [1] Automated Job Search System developed to address the growing challenges of modern recruitment. In today's competitive and fast-paced job market, candidates often spend significant time browsing multiple job portals, manually filtering listings, and customizing resumes for different roles. At the same time, recruiters face difficulties in screening large volumes of applications efficiently. The Tech Hire aims to bridge this gap by automating resume analysis [8] and improving job-to-candidate matching through intelligent technologies.

The system is built as a full-stack web application using React [14] for creating a dynamic and responsive user interface, Node.js [15] for handling backend services and API communication, and Supabase [16] as a secure and scalable database and authentication solution. Together, these technologies ensure high performance, data security, and seamless user experience.

The platform is structured into two primary modules: the User Module and the Admin Module. The User Module is designed for job seekers and provides features such as secure registration and login, resume upload and management, AI-powered [1] resume analysis [8], personalized job recommendations [6], profile editing, and real- time location-based job tracking. Once a resume is uploaded, the system processes the document using Natural Language Processing [2] (NLP) techniques to extract structured information, including technical skills, educational background, certifications, work experience, and project details. Intelligent matching algorithms then compare this extracted data with available job postings and generate ranked recommendations based on compatibility scores. The integration of real-time location tracking further enhances job discovery by suggesting opportunities within preferred geographic areas.

The Admin Module allows administrators to manage user accounts, add or update job postings, monitor system activities, review analytics, and maintain overall platform integrity. This centralized control ensures accurate job listings and consistent system performance.

DOI: 10.48175/IJARSCT-31977

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

Unlike traditional keyword-based job portals, The Tech Hire uses contextual analysis and similarity scoring to improve recommendation accuracy and reduce mismatches. By automating resume screening and job matching, the system minimizes manual effort, enhances recruitment efficiency, and supports fair, data-driven hiring decisions. Overall, The Tech Hire provides a scalable, secure, and intelligent solution that simplifies the recruitment process for both job seekers and administrators.

OVERALL DESCRIPTION

The Tech Hire is a smart and automated job-matching platform designed to make job searching faster and more efficient. Instead of relying on basic keyword searches, the system uses AI-based resume analysis [8] to understand a candidate's skills, qualifications, experience, and career interests. It then compares this information with available job listings to provide accurate and personalized job recommendations [6].

The platform is built using modern web technologies such as React [14] for the interactive user interface, Node.js [15] for backend operations, and Supabase [16] for secure data storage and authentication. It consists of two main components: a User Module and an Admin Module. Job seekers can create accounts, upload resumes, receive tailored job suggestions, and explore opportunities based on their preferred locations. Meanwhile, administrators can manage job postings, monitor platform performance, and ensure system reliability.

By integrating automation, intelligent matching, and location-based features, The Tech Hire reduces the time and effort required for job searching and recruitment. It offers a scalable and secure solution that enhances hiring accuracy and improves the overall job discovery experience.

1. Objectives

- Automate resume analysis [8] using NLP techniques.
- Extract structured information from unstructured resume data.
- Match candidates with relevant job opportunities using AI algorithms.
- Integrate external job portal APIs for real-time job listings.

2. Applications

- Job Seekers
- Recruiters / HR Teams
- Recruitment Platforms
- Career Guidance & Academic Institutions

3. Architecture Design:

Fig. 1: Automated Job Search Pipeline – Scrape, Extract, Match, and Rank Job Opportunities

4. Challenges and Limitations

- Resume Parsing and Text Extraction:

A major challenge was accurately parsing resumes in various formats due to inconsistent layouts, tables, non- standard skill names, and noise, highlighting the need for a robust preprocessing pipeline with proper text cleaning, tokenization, and normalization before applying AI models.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

• Natural Language Processing [2] Complexity:
Extracting relevant skills required advanced NLP techniques since simple keyword matching was insufficient due to varied terminology (e.g., "ML" vs "Machine Learning"), highlighting the need for semantic similarity modeling and domain-specific tuning for accurate contextual understanding.
• API Integration Challenges:
Integrating external job APIs posed challenges such as rate limits, authentication errors, inconsistent data formats, and network latency, demonstrating the importance of strong error handling and reliable fallback mechanisms when working with third-party services.
• Data Normalization and Standardization:
Different job APIs return data in varying formats. Standardizing job listings into a unified internal structure required additional transformation logic. Lesson learned: Data normalization is essential for scalable system design.

5. Data Flow Diagram:

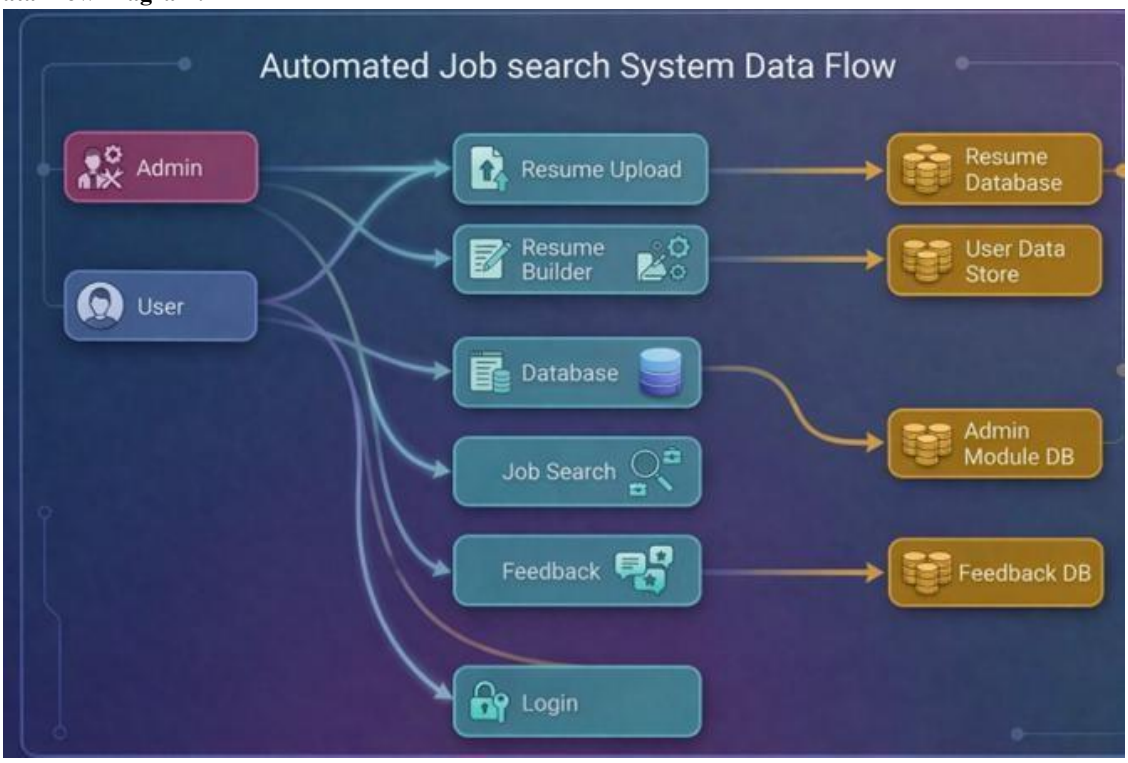

Fig. 2: Data Flow Diagram of Automated Job Search System

The data flow of the Automated Job Search System begins with two main actors: Admin and User. The user logs into the system and can either upload an existing resume or create one using the resume builder module. This information is securely stored in the Resume Database and User Data Store.
Once the data is stored, the system processes it through the database module to analyze user skills, qualifications, and preferences. Based on this data, the Job Search module retrieves relevant job listings and provides personalized recommendations to the user.
The admin manages system operations through the Admin Module Database, ensuring smooth functioning and data management. Additionally, users can provide feedback, which is stored in the Feedback Database to improve system performance and user experience.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

Overall, the system ensures efficient data flow by collecting, processing, and utilizing user information to deliver accurate and real-time job recommendations.

6. Architecture:

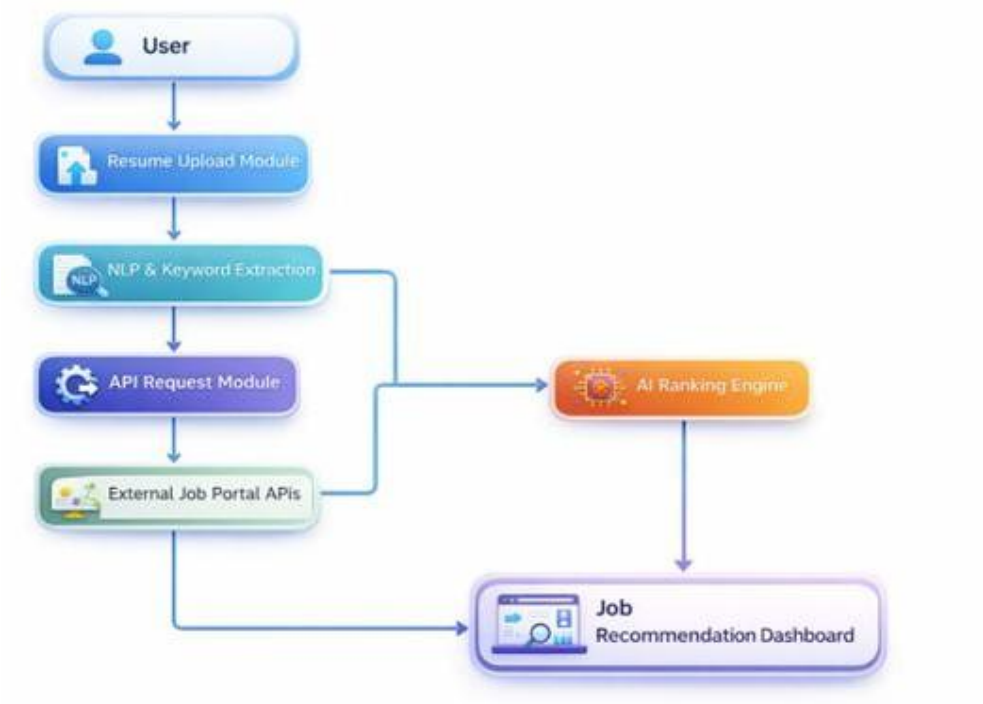

Fig. 3: Architecture of Automated Job Search System

The architecture of the Automated Job Search System follows a modular and intelligent pipeline approach. It starts with the user uploading a resume through the Resume Upload Module, where the system collects candidate information.

The uploaded resume is processed using the NLP & Keyword Extraction module, which analyzes the content to extract important skills, qualifications, and keywords. This structured data is then passed to the API Request Module, which communicates with external job portals via External Job Portal APIs to fetch relevant job listings.

Simultaneously, the extracted data is sent to the AI Ranking Engine, which evaluates and ranks job opportunities based on user profile matching, relevance, and preferences. Finally, the processed and ranked job recommendations are displayed to the user through the Job Recommendation Dashboard.

This architecture ensures automation, intelligent matching, and real-time job recommendations by integrating NLP, APIs, and AI-driven ranking mechanisms.

7. Advantages:

- Reduces manual job searching
- Provides real-time job recommendations [6].
- Improves matching accuracy
- Offers scalable architecture.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, March 2026
Impact Factor: 8.2

8. Functional Requirements:

The system follows a structured workflow to process resumes and job data effectively. The first stage is Resume Collection, where resume files in PDF or DOC format are collected from users. These resumes are then processed using Natural Language Processing [2] (NLP) and parsing techniques to extract relevant information. The output of this stage is a structured candidate profile that includes details such as skills, education, and work experience.

The next stage is Job Data Collection, where job descriptions are gathered from recruiters. These descriptions undergo text cleaning and keyword extraction to remove unnecessary information and identify important terms. The result is a structured set of job requirements.

In the Feature Extraction stage, both resume data and job descriptions are analyzed. Techniques like TF-IDF [13], word embeddings [11], and tokenization are applied to convert textual data into numerical feature vectors, making it suitable for computational processing.

The Matching Process then compares candidate features with job features using cosine similarity [6] and machine learning [3] algorithms. This stage produces a matching score that indicates how well a candidate fits a particular job role.

Following this, the Recommendation Engine uses matching scores along with user preferences to rank and filter job opportunities. This results in personalized job recommendations [6] for the candidate.

Finally, the Skill Gap Analysis stage compares the candidate's current skills with job requirements. It identifies missing skills and suggests relevant courses or improvements, helping candidates enhance their profiles and increase their chances of selection.

9. Non-Functional Requirements:

The system is designed with several important non-functional requirements to ensure efficiency, reliability, and user satisfaction.

Performance is a key requirement, where the system should be capable of processing resumes, job searches, and recommendations quickly. Ideally, it should respond within 2–3 seconds for a single request to provide a smooth user experience.

Scalability ensures that the platform can handle multiple users at the same time without any decrease in performance. This is important as the number of users grows over time.

Reliability focuses on maintaining consistent uptime and ensuring that AI algorithms for job matching and recommendations function correctly without failures. This helps build user trust in the system.

Security is crucial as the platform deals with sensitive user data such as resumes and application history. The system must store this data securely using encryption and implement strong authentication mechanisms to prevent unauthorized access.

Usability emphasizes that the interface should be user-friendly and intuitive. Both job seekers and recruiters should be able to navigate the system easily without confusion.

Maintainability requires the system to be modular and easy to update. This allows developers to integrate new AI models or features with minimal disruption to the existing system.

Finally, Portability ensures that the system can be accessed across different devices such as web and mobile platforms, and supports various operating systems, making it widely accessible to users.

10. API's and Tools Used:

The system integrates various APIs and tools to efficiently process resumes and fetch job-related data.

The Resume Reader API is used to parse resumes and convert them into a structured JSON format. This makes it easier for the system to process and integrate the data into further stages of analysis.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, March 2026 Impact Factor: 8.2

The Sharp API Resume Parser plays an important role in extracting detailed resume information and calculating resume-job compatibility scores using advanced AI models. This helps in evaluating how well a candidate matches a job role.

The Arya.ai Resume Parser is utilized to convert resumes into structured candidate profiles that are suitable for Applicant Tracking Systems (ATS). This ensures compatibility with industry-standard recruitment systems.

The Quick-Extract tool provides resume data extraction in both JSON and CSV formats, making it flexible for storage, processing, and data analysis.

The Textkernel Parser is an enterprise-grade multilingual resume parsing API that supports advanced recruitment analytics. It is especially useful for handling resumes in different languages and large-scale hiring processes.

For job data collection, the system uses the Google Jobs API, which automatically retrieves job listings from various online job feeds. This helps keep the platform updated with the latest job opportunities.

Additionally, the Indeed API is integrated to enable job search functionality directly within the application. It allows users to search and explore job listings without leaving the platform.

11. Future Scope:

The Automated Job Search System can be further enhanced with several advanced features to improve its effectiveness and user experience. One potential improvement is the integration of automated job application functionality, where users can apply to suitable job listings directly through the platform. This would significantly reduce manual effort and streamline the recruitment process.

Another important enhancement is the use of advanced deep learning models such as BERT for improved semantic understanding and more accurate job matching. This can help in better interpretation of resume content and job descriptions beyond basic keyword matching.

The system can also be extended by incorporating an AI-based chatbot for career guidance, which can assist users with resume building, interview preparation, and career suggestions. Additionally, integrating real-time labor market analysis can provide users with insights into trending skills and in-demand job roles.

Further improvements may include mobile application development for better accessibility, multilingual resume parsing to support diverse users, and integration with professional networking platforms for enhanced job discovery.

Overall, these enhancements will make the system more intelligent, scalable, and user-centric in future implementations.

II. CONCLUSION

The development of Tech Hire effectively transforms the traditional job search process into an intelligent, AI-driven system that addresses the challenges faced by modern job seekers. By utilizing advanced resume parsing, semantic skill matching, and real-time job API integration, the platform delivers accurate and personalized job recommendations [6] based on a candidate's skills, experience, and location preferences. Built using React [14], Node.js [15], and Supabase [16], the system ensures a secure, scalable, and high-performance full-stack architecture. It reduces manual effort, improves job-to-candidate matching accuracy, and enhances overall recruitment efficiency for both users and administrators. Additionally, its modular and flexible design allows for future integration of more advanced machine learning [3] models, enabling continuous improvement in recommendation quality and deeper career insights for users.

ACKNOWLEDGEMENT

I sincerely express my gratitude to our respected project guide for their continuous guidance, valuable suggestions, and encouragement throughout the development of the project titled "Automated Job Search Using Resume Keywords via APIs (Tech Hire)". Their support greatly contributed to the successful completion of this work. I am also thankful to the Head of the Department and faculty members for providing the necessary resources and academic environment. I

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 5, March 2026 Impact Factor: 8.2

extend my appreciation to my friends for their cooperation and assistance. Lastly, I am deeply grateful to my family for their constant motivation, understanding, and unwavering support during this project.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2020.
[2] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2022.
[3] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[4] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[5] J. Malinowski et al., 'Matching people and jobs: A bilateral recommendation approach,' HICSS, 2006.
[6] J. Paparrizos et al., 'Machine learned job recommendation,' ACM RecSys, 2011.
[7] L. Li et al., 'Person-job fit with joint representation learning,' IEEE TKDE, 2016.
[8] Y. Xu et al., 'Resume information extraction with deep learning,' IEEE Access, 2019.
[9] Y. Zhang and X. Chen, 'Explainable recommendation: A survey,' Foundations and Trends in IR, 2020.
[10] S. Kessler et al., 'Resume parsing using NLP techniques,' IJCA, 2012.
[11] T. Mikolov et al., 'Efficient estimation of word representations in vector space,' arXiv, 2013.
[12] J. Devlin et al., 'BERT: Pre-training of deep bidirectional transformers,' NAACL, 2019.
[13] G. Salton and C. Buckley, 'Term-weighting approaches in text retrieval,' Information Processing & Management, 1988.
[14] React Documentation, https://react.dev
[15] Node.js Documentation, https://nodejs.org
[16] Supabase Documentation, https://supabase.com