

Job Hook- An Intelligent Job Aggregator Portal With AI-Based Recommendation Engine

¹Dr. A. C. Sawant, ²Aniket Solanke, ³Laxman Ghodke, ⁴Rohan Gore, ⁵Sumit Lohar

¹ Assistant Professor, Department of Information Technology, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra, India

²³⁴⁵ Undergraduate Students, Department of Information Technology, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra, India

Abstract- The Job Aggregator Portal is an intelligent platform that consolidates job listings from multiple sources to simplify the employment search process for users. It leverages APIs such as Adzuna, Jooble, and JSearch to provide real-time, up-to-date job information across various domains. The portal incorporates AI-ML- based resume analysis to recommend suitable jobs based on user skills and experience, enhancing personalization and efficiency. Built using React for the frontend and Spring Boot for the backend, the system ensures seamless data management, user authentication, and fast performance. By integrating smart algorithms and modern web technologies, the Job Aggregator Portal significantly improves the job search experience for users while assisting companies in reaching the right candidates efficiently. Furthermore, the Job Aggregator Portal addresses major challenges in the job search ecosystem—such as data redundancy, fake listings, and inefficient filtering—by implementing real-time synchronization, data validation, and intelligent ranking algorithms. The system's modular architecture allows easy scalability and integration of additional APIs, ensuring adaptability to evolving market trends. It also emphasizes data security and privacy by using secure API communication and encrypted user authentication. The AI-driven recommendation engine continuously learns from user interactions, improving job relevance over time. Additionally, the platform provides valuable insights through analytics dashboards, helping recruiters understand user preferences and optimize postings.

Keywords: Job Aggregator, Employment Technology, Job Search Portal, API Integration, Resume Parsing, AI-Based Job Recommendation

I. INTRODUCTION

In the rapidly evolving digital era, job searching has become one of the most critical yet challenging tasks for individuals seeking employment opportunities. With the rise of numerous online job platforms such as LinkedIn, Naukri.com, and Indeed, job seekers often face the inconvenience of visiting multiple websites and manually filtering through countless job postings. To address this issue, the Job

Aggregator Portal aims to unify and streamline the job search process by collecting listings from multiple sources and presenting them in one centralized platform. The proposed system integrates APIs such as Adzuna, Jooble, and JSearch to fetch real-time job data across different industries and

domains. Additionally, it incorporates AI-ML-based resume analysis to extract user skills and provide personalized job recommendations, enhancing both accuracy and relevance. The backend of the portal is developed using Spring Boot, ensuring secure data handling and API integration, while the frontend is built with Angular, providing a responsive and user friendly interface. This system not only simplifies the job search experience for candidates but also benefits employers by increasing the visibility of job listings and reaching suitable candidates efficiently IoT data and advanced analytics. This system enhances the touring experience for travelers while optimizing the operational efficiency of the agencies. It is also integrated with a GPS tracker for real-time navigation

Future enhancements may also include chatbot- based guidance and multilingual support to increase accessibility. Ultimately, this research contributes to creating an innovative, scalable, and intelligent platform that redefines the future of digital recruitment.

II. LITERATURE SURVEY

The job search process has evolved significantly with the advancement of web technologies, data aggregation techniques, and intelligent recommendation systems. Traditional employment portals such as LinkedIn, Indeed, and Naukri.com primarily focus on hosting job listings; however, they often operate as isolated systems, requiring users to visit multiple platforms to find suitable opportunities. This fragmentation has led to the emergence of Job Aggregator Systems, which collect, filter, and display job data from diverse sources through APIs and web scraping methods. The following section reviews the existing literature on related domains, including job aggregation, recommendation systems, AI-based job matching, API integration techniques, and resume parsing and skill extraction technologies.

Job Aggregation Systems Job aggregation platforms are designed to unify employment data from multiple sources and present it on a single platform. According to Sodhi et al. (2018), aggregators simplify the recruitment process by collecting data from various job portals and providing users with comprehensive results. These systems utilize web crawlers or RESTful APIs to fetch listings in real time, thus improving accessibility and convenience. However, challenges such as data duplication, inconsistent data formats, and verification of job authenticity still persist. Zhu et al. (2020) emphasized the need for intelligent filtering algorithms to eliminate redundant listings and improve the reliability of aggregated results. As the demand for automation in recruitment grows, job aggregation systems continue to evolve toward becoming AI-assisted employment ecosystems that combine scalability, intelligence, and efficiency to transform the future of digital hiring.

Evaluation Metrics Metrics by which recommendation systems normally are evaluated include accuracy, precision, recall, and F1score.

Recommendation Systems in Job Platforms Recommendation systems have become an integral component of modern job platforms, significantly enhancing the user experience by offering personalized and context-aware job suggestions. These systems analyze a variety of inputs, including users' past interactions, search history, educational background, professional experience, and skill sets, to generate highly relevant job recommendations. By automating the matching process, recommendation systems reduce the cognitive load on users, saving time and improving engagement with the platform. The integration of hybrid filtering strategies and AI- driven models ensures that modern job aggregators provide not only accuracy but also diversity and user satisfaction, thereby significantly transforming employment landscape.

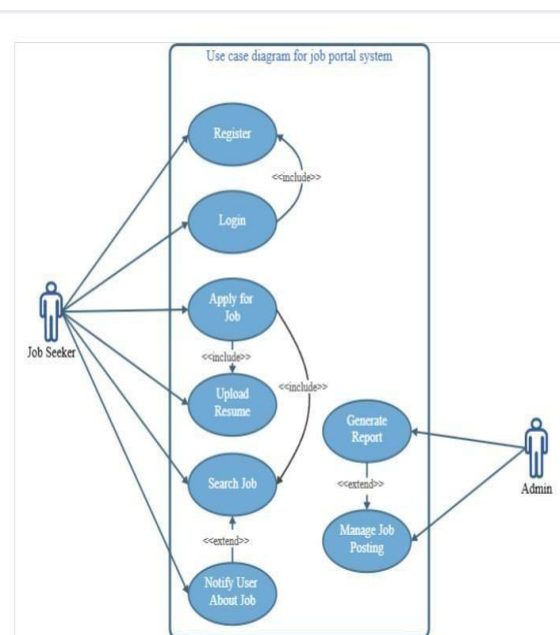
AI/Machine Learning in Job Matching Artificial Intelligence (AI) and Machine Learning (ML) have significantly enhanced the accuracy and personalization of job matching. Patel and Sharma (2021) demonstrated the use of Natural Language Processing (NLP) to analyze resumes and extract relevant skills automatically. Machine learning classifiers, such as Support Vector Machines (SVM) and Random Forests, are employed to predict job suitability. These approaches help match candidates to relevant opportunities with higher precision and efficiency. Techniques such as BERT (Bidirectional Encoder Representations from Transformers) and word embeddings like Word2Vec enable systems to interpret the meaning of skills, experience, and qualifications beyond simple keyword matching. Moreover, reinforcement learning models have been explored to continuously improve job recommendations based on user feedback and interaction patterns. These intelligent systems not only automate the recruitment process but also minimize human b

API Integration for Real-Time Data Modern job aggregators rely on APIs provided by external job listing platforms such as Adzuna, Joooble, and JSearch to collect structured job data. According to Rahman et al. (2022), API-based integration improves data reliability, reduces maintenance costs compared to web scraping, and ensures up-to-date listings. RESTful APIs, in particular, allow efficient communication between the frontend and backend, facilitating real-time synchronization and faster response times. However, one of the main challenges lies in handling different API response formats and ensuring scalability while maintaining optimal system performance.

III. METHODOLOGY

1. System Design

The Job Aggregator Portal is designed with a modular architecture comprising three main components: the front-end, the back-end, and the database. The front-end is developed using Angular, ensuring a responsive, interactive, and user-friendly interface where job seekers can register, upload resumes, and browse personalized job recommendations. The back-end, implemented using Flask (Python), serves as the core processing unit that manages API requests, handles resume parsing, applies AI-based recommendation logic, and ensures secure data transactions. The database, built using MySQL, stores critical information such as user profiles, extracted skills, job listings, and system



logs. Together, these components form an integrated platform that provides a seamless job search and recommendation experience.

2. Data Collection

The system collects data from multiple external APIs such as Adzuna, Joooble, and JSearch, which provide structured and real-time job listings from various industries and locations. These APIs return job details like title, description, company name, location, and salary. Simultaneously, the system gathers user data through uploaded resumes, which are processed to extract essential information such as skills, education, and experience. The collected data is stored in the database and periodically updated to ensure freshness and accuracy. Additionally, feedback from users' interactions—such as job clicks or saved jobs—is recorded to enhance future recommendation accuracy through machine learning models.

3. Algorithm Development

At the core of the Job Aggregator Portal lies an AI-powered recommendation algorithm designed to match users with the most relevant job opportunities. The process begins with Natural Language Processing (NLP) techniques that extract and analyze skills from the user's resume. These extracted features are then compared against job descriptions obtained from multiple APIs using similarity measurement algorithms such as cosine similarity and keyword matching. The Machine Learning (ML) component refines recommendations based on user interactions and feedback, ensuring continuous improvement in job relevance. The algorithm prioritizes roles based on the user's skill set, location preference, and job popularity metrics to deliver highly accurate and personalized results.

4. System Integration

Once the individual modules—front-end, back-end, and database—are developed, they are integrated into a unified and dynamic system. The Flask back-end acts as the intermediary between the Angular front-end and the MySQL database, managing data flow and ensuring secure API communication. Real-time data from APIs such as Adzuna, Joooble, and JSearch are fetched, processed, and displayed dynamically on the user interface. The AI-ML model is seamlessly integrated into the back-end, enabling real-time skill analysis and job recommendation generation. This integration ensures that users experience a cohesive system that provides up-to-date job listings, personalized recommendations, and secure user authentication—all in one unified platform System Design:

IV. OVERVIEW

In the modern recruitment ecosystem, the role of job aggregators has evolved significantly with the integration of Artificial Intelligence (AI) and Machine Learning (ML). Unlike traditional job portals that rely on manual searches or basic keyword matching, the Job Aggregator Portal incorporates AI-driven resume analysis to extract user skills, qualifications, and experience automatically. Using this data, the system intelligently recommends personalized job opportunities that align with each user's profile, improving the accuracy and relevance of job matches.

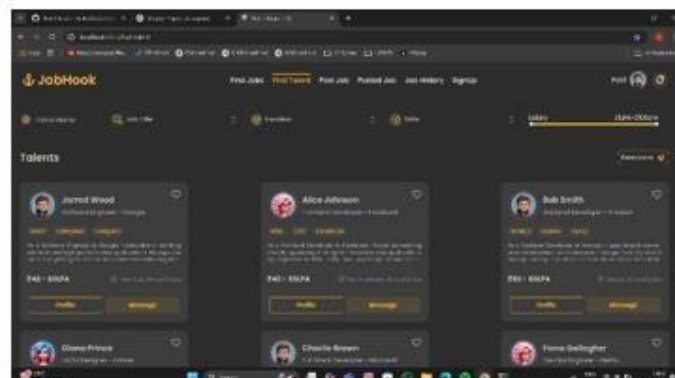


Fig1. Users Interface

The backend of the portal, developed using Flask (Python), manages real-time data fetching, resume parsing, and recommendation generation. The frontend, designed with Angular, provides an interactive and responsive user interface for easy navigation, job filtering, and profile management. The MySQL database stores user information, extracted skills, and API data efficiently, ensuring reliable access and fast response times. APIs such as Adzuna, Joooble, and JSearch are integrated to fetch up-to-date job listings from multiple platforms, ensuring comprehensive coverage across industries and regions.



Moreover, the system emphasizes user privacy, data validation, and real-time synchronization to eliminate issues such as duplicate listings or outdated job data. The AI-based recommendation model continuously learns from user feedback, improving its performance over time to deliver more refined results. By automating the process of job collection, filtering, and recommendation, the portal enhances user satisfaction and provides a smarter, faster, and more personalized job search experience.



In addition to empowering job seekers, the platform also benefits employers by expanding the visibility of job postings and helping them connect with the most relevant candidates. The integration of AI, data analytics, and modern web technologies ensures that the Job Aggregator Portal not only streamlines the job search process but also contributes to transforming the traditional recruitment landscape into a more data-driven, intelligent, and efficient ecosystem.

V. PROPOSED SYSTEM

1. AI-Powered Resume Analysis

The proposed Job Aggregator Portal utilizes Artificial Intelligence (AI) and Natural Language Processing (NLP) to analyze the resumes uploaded by users. The system automatically extracts essential information such as technical skills, educational qualifications, and work experience. These extracted features are then matched with job descriptions fetched from integrated APIs. This automation eliminates the need for manual filtering and ensures that users receive job suggestions that align closely with their professional profile.

2. API Integration for Real-Time Job Listings

The system integrates multiple external APIs, such as Adzuna, Jooble, and JSearch, to fetch real-time job data from different platforms. This allows the portal to display the most recent and relevant job openings across various industries and geographic locations. By consolidating job data from multiple sources, the system ensures diversity, reliability, and freshness in job listings while reducing redundancy and outdated information.

3. User-Friendly Web Interface

The frontend of the portal is developed using Angular, offering a clean, responsive, and interactive interface. Users can easily register, upload resumes, apply filters, and view personalized job recommendations. Recruiters or companies can also post job openings through a dedicated dashboard. The user interface focuses on simplicity and accessibility, ensuring that both beginners and professionals can navigate the system effortlessly.

4. Secure Back-End and Database Management

The back-end of the system, developed using Flask (Python), handles data processing, API communication, and recommendation logic. It ensures secure authentication and manages requests between the frontend and the database. The MySQL database stores all user data, extracted skills, and API responses in a structured and secure manner. Proper indexing and normalization techniques are applied to ensure high performance and data consistency.

5. Real-Time Synchronization and Continuous Learning

The system continuously synchronizes with the integrated job APIs to update job listings in real time. It also employs feedback learning, where user responses—such as applying, skipping, or saving a job—are used to fine-tune future recommendations. This continuous improvement mechanism ensures that the AI model evolves with user preferences, delivering increasingly accurate results.

6. Scalability and Integration Capabilities

The Job Aggregator Portal is designed with scalability in mind, allowing easy integration of additional APIs, AI models, or modules in the future. Its modular architecture enables it to adapt to new technologies or expand into advanced features such as chatbot-based job guidance, resume scoring, and employer analytics. This flexibility ensures long-term sustainability and relevance in the rapidly evolving job market.

VI. CONCLUSION

Please text here conclusion

REFERENCES

1. M. Sodhi et al., "Job Aggregation Systems and Recruitment Optimization," 2018.
2. Y. Zhu et al., "Data Filtering and Deduplication in Job Aggregators," 2020.
3. R. Patel and K. Sharma, "AI-Based Resume Analysis and Job Matching Using NLP," 2021.
4. A. Rahman et al., "API Integration Techniques for Real-Time Data Systems," 2022.
5. Adzuna API, "Adzuna API Documentation." [Online]. Available: <https://developer.adzuna.com>
6. Jooble API, "Jooble API Documentation." [Online]. Available: <https://jooble.org/api/about>
7. JSearch API, "JSearch API Documentation." [Online]. Available: <https://rapidapi.com>