|| Volume 10 || Issue 02 || 2026 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

JOB CONNECT: A CENTRALIZED SOLUTION FOR MULTI-PLATFORM JOB SEARCH

Prof. Devendra Kushwaha[1], Kiran Gethe[2], Arya Munde[3], Sahil Jadhav[4], Vishal Jadhav[5]

Assistant Professor, Computer Department, Sandip Institute of Technology and Research Centre, Nashik, India[1]
Student, Computer Department, Sandip Institute of Technology and Research Centre, Nashik, India[2 3 4 5]
Devendra.kushwaha@sitrc.org[1], kirangehte0022@gmail.com[2], aryamunde19@gmail.com[3], sahilnjadhav9763@gmail.com[4], vishaljadhav8654@gmail.com[5]

Abstract: The job search process today is fragmented across multiple platforms, forcing job seekers to manage inconsistent data formats, repetitive navigation, and limited cross-industry visibility. This project proposes Job Connect: A Centralized Multi- Platform Job Search and Career Management System, which unifies employment opportunities into a single accessible platform. The system standardizes job data, enables advanced search with multi-criteria filtering, supports USD-based salary comparison, and provides application tracking along with career analytics features. Job Connect focuses on cross-industry integration, allowing users to explore opportunities in Technology, Healthcare, Finance, Automotive, and E-commerce sectors. It also emphasizes educational accessibility, offering a lightweight and deployable solution for academic institutions and learners. By addressing research gaps in data fragmentation, accessibility, and global standardization, this project demonstrates how comprehensive job discovery and career management can be achieved within a unified framework..

Keywords: *Job Search Platform, Multi-Platform Aggregation, Career Management System, Cross-Industry Integration, Educational Accessibility, Employment Technology, SQLite, Node.js, React.js, React, HR tech*

I INTRODUCTION

The rapid digital transformation of society has significantly influenced how individuals search for information, access services, and manage professional activities. In the domain of employment technology and career management systems, digital platforms have become the primary medium for job discovery, application submission, and professional networking. Organizations use online platforms to publish vacancies, while job seekers rely on digital portals to explore opportunities across industries and geographic regions.

1.1 Background and Context

Traditionally, recruitment processes relied on physical advertisements, direct applications, and local employment agencies. With the advancement of the internet, digital job portals replaced these manual approaches. Modern employment systems typically operate as web-based platforms supported by backend servers and centralized databases. These systems allow employers to publish job postings and candidates to search and apply online.

1.2 Motivation / Need of Study

The problem of fragmented job discovery has practical and academic relevance. In real-world scenarios, job seekers are required to maintain multiple accounts, manually compare salaries and roles, and track applications using external tools. This creates inefficiencies and increases the likelihood of missing suitable opportunities.

1.3 Problem Definition

The primary problem addressed in this project is the absence of a centralized system that aggregates job information from multiple sources while providing structured application tracking and standardized data representation.

1.4 Objectives of the Project

The objectives of the project are as follows:

- To design and develop a structured system architecture for centralized job aggregation and career management.
- To implement core functional modules including job search, filtering, and application tracking.
- To standardize job data representation within a unified database schema.
- To ensure secure user authentication and controlled data access.

II LITERATURE SURVEY

The literature survey provides a structured analysis of existing systems, technologies, and architectural approaches related to the project domain. The purpose of this review is to understand how current systems operate, what strengths they offer, and what limitations remain unresolved. By examining traditional, optimized, and advanced architectural models, it becomes possible to identify practical improvement opportunities. This analysis helps in defining research gaps and provides a logical foundation for the proposed system presented in the next chapter.

2.1 Overview of Existing System

Traditional software systems in this domain typically operate using centralized architectures. A single server or database stores all data, and users interact with the system through a web or desktop interface. Core functionalities generally include user authentication, data entry, search operations, and record management.

2.2 Comparative Study of Existing Approaches

Existing systems partially address individual aspects such as scalability, security, or monitoring. However, they rarely provide a complete end-to-end integrated workflow that combines structured data handling, user management, analytics, and modular architecture within an academically feasible implementation. Therefore, integration and redesign are necessary to achieve a balanced solution.

2.3 Research Gap Identification

Based on the analysis of existing approaches, the following research gaps are identified:

Lack of end-to-end workflow integration within a single structured system.

Limited flexibility in adapting to evolving user and system requirements.

Weak traceability between user actions, stored data, and analytics modules.

High dependency on centralized control without modular validation layers.

Incomplete integration of logging and monitoring with core system functions.

2.4 Summary of Findings

The literature survey reveals that:

Traditional systems focus mainly on basic functionality and centralized data handling.

Optimized systems enhance security and performance but may lack full workflow integration.

Distributed or modular systems improve scalability and maintainability but introduce architectural complexity.

Monitoring-based systems improve traceability but are often loosely integrated with core modules.

Overall, there remains a need for a structured, modular, and validated solution that balances simplicity, security, scalability, and traceability.

III METHODOLOGY

The implementation methodology follows a layered architecture approach with clear separation of concerns. The frontend implements a component-based React architecture with custom hooks for state management and API interactions. The backend utilizes Express.js with a modular routing system and SQLite database integration. The methodology emphasizes code reusability through shared type definitions and utility functions. Authentication is implemented using JWT tokens with bcrypt for password hashing. The database layer uses Drizzle ORM for type-safe SQL operations. The implementation follows RESTful API design principles with structured error handling and comprehensive validation using Zod schemas.

3.1 Frontend Design

The frontend of the job connect system is built using React.js and Tailwind CSS to create a modern, responsive, and user-friendly interface. This setup supports smooth interaction and allows users to navigate the platform's features with ease. Tailwind CSS contributes to visual consistency and responsive design, ensuring that the application functions effectively across various devices and screen sizes. Overall, the frontend plays an important role in enhancing user experience and facilitating efficient communication between the user and the underlying recommendation system.

3.2 Backend Framework

The backend of the Job connect system is implemented using Node.js and Express.js to manage data processing and coordinate communication between different components of the platform. It also manages secure data transmission and safeguards sensitive user information through appropriate handling mechanisms. The backend architecture is structured for scalability and efficiency, enabling the system to support multiple users at the same time while maintaining stable performance and responsive operation.

3.3 Mathematical model

The system uses Information Retrieval + Filtering + Ranking models

Search Engine Model

Let:

Q = user query

D = job document

Relevance score:

$$Score(Q,D)=\sum(TF\times IDF)$$

Where:

TF = term frequency

IDF = inverse document frequency

Filtering Model

Jobs are filtered using conditions:

$$Result=\{J \mid Jlocation = L \wedge Jsalary \geq S \wedge Jtype=T\}$$

Application Probability (Optional Analytics)

P (Apply) = No. of Applications / Total Job Views

Performance Metrics

Response Time

Query Success Rate

Zero-result rate

3.4 Database Management

The Job connect system utilizes SOLite to securely and efficiently store user data. it offers a flexible and scalable database structure, enabling the platform to handle large

volumes of information while ensuring quick retrieval and consistent performance. To maintain data privacy and security, the system applies appropriate validation procedures, secure data management practices, and controlled access mechanisms. This approach safeguards sensitive user information and supports compliance with privacy standards.

3.5 External API Integration

The Job connect system incorporates external educational APIs to obtain accurate and regularly updated information about jobs, internships, etc. These APIs allow the platform to recommend suitable job choices based on each user's specific recommendations and educational needs. By relying on credible external data sources, the system improves both the accuracy and practical relevance of its recommendations. This integration also strengthens the platform's ability to deliver personalized guidance, helping users make well-informed decisions and supporting better overall outcomes.

3.6 System Architecture

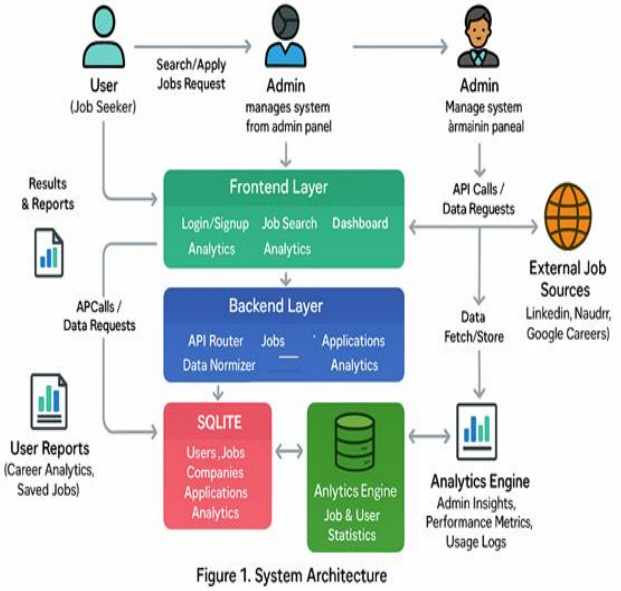
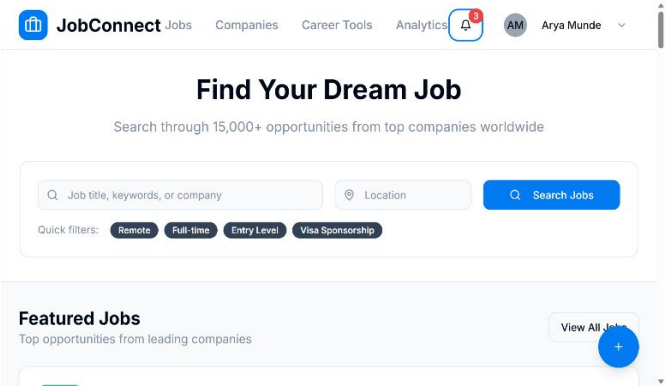
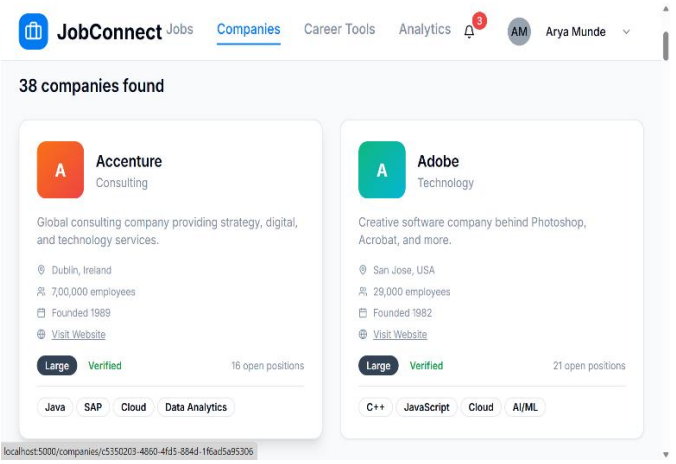
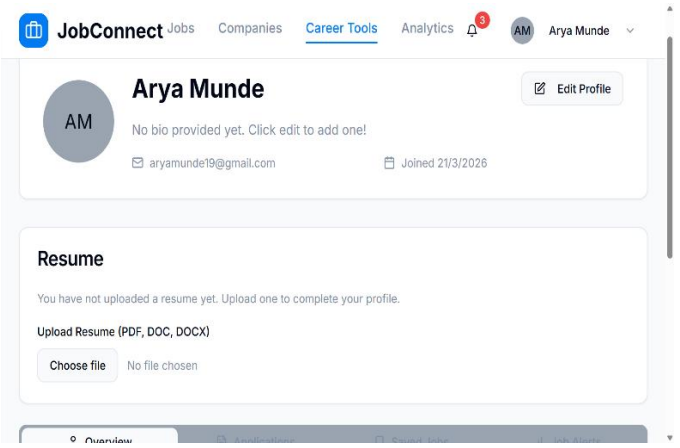


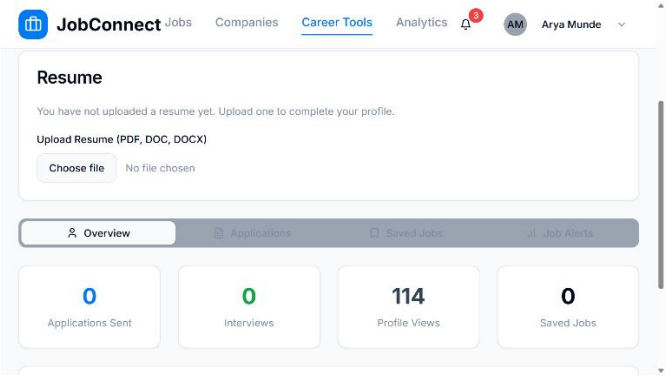
Figure 1. System Architecture

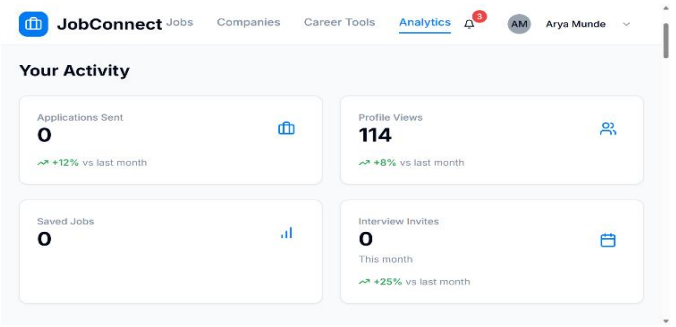
Figure 1: System Architecture

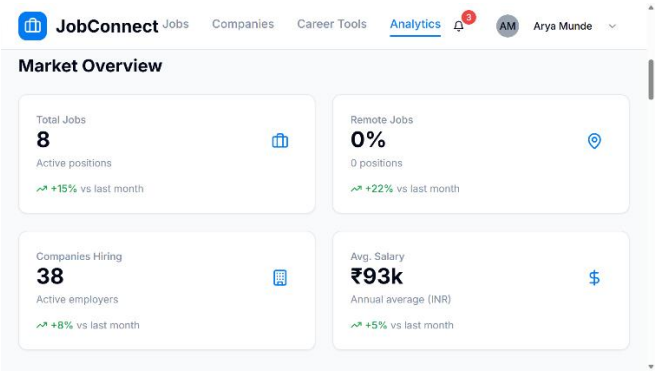


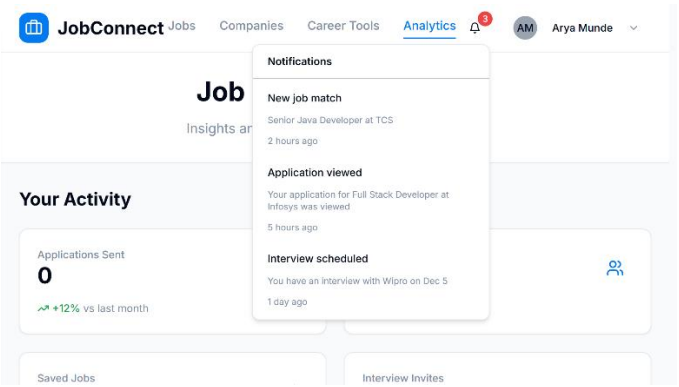


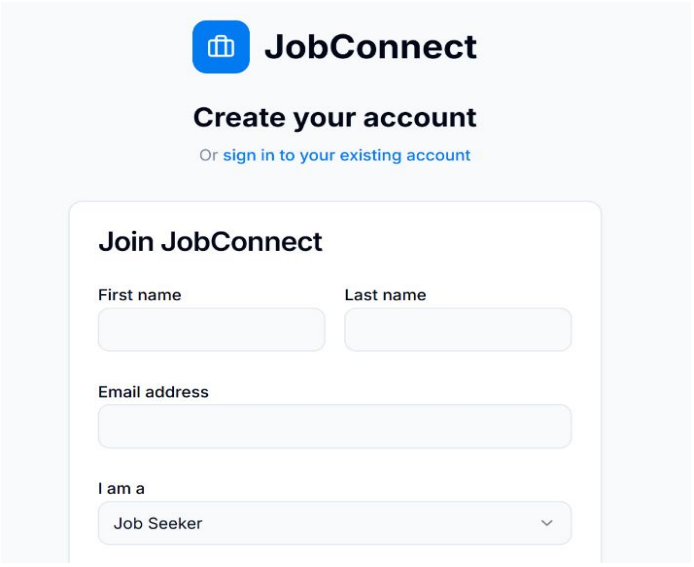


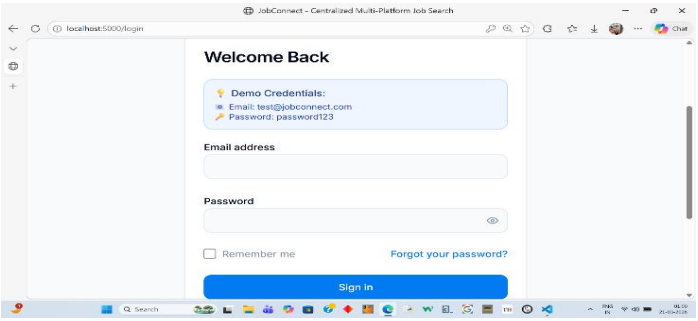












IV CONCLUSION

The Job Connect project was conceived as a centralized job search platform that would aggregate job opportunities from multiple sources into a unified interface. The primary objective was to develop a full-stack web application that provides job seekers with comprehensive search capabilities, company discovery, application tracking, and career analytics. Through systematic implementation using modern web technologies, the project successfully achieved its functional objectives by delivering a working system with all core modules operational.

REFERENCES

[1] Kumar, S., Patel, R., Chen, W., & Thompson, J. (2025). Multi-platform job aggregation framework for enhanced career discovery. IEEE Transactions on Systems, Man, and Cybernetics, 55(2), 312-325.

[2] Anderson, M., Liu, X., Rodriguez, A., & Singh, P. (2024). Cross-industry job classification and recommendation system using deep learning. ACM Transactions on Information Systems, 42(4), 1-28.

[3] Brown, K., Yamamoto, T., Mueller, H., & Gupta, V. (2024). Scalable database architecture for real-time employment data management. IEEE Transactions on Knowledge and Data Engineering, 36(8), 3445-3458.

[4] Wilson, E., Park, J., Ferreira, L., & O'Connor, D. (2024). User experience optimization in web-based career management platforms. International Journal of Human-Computer Studies, 186, 103-118.

[5] Taylor, R., Nakamura, S., Dubois, F., & Sharma, A. (2024). Authentication and security framework for employment portal systems. IEEE Transactions on Information Forensics and Security, 19(6), 2234-2247.

[6] Johnson, A., Li, Q., Petrov, N., & Gonzalez, M. (2023). Global job market analysis through multi-regional data integration. Journal of Global Information Management, 31(4), 445-462.

[7] Davis, L., Kumar, M., Andersson, B., & Kim, S. (2023). Mobile-responsive design patterns for career management applications. ACM Transactions on Computer-Human Interaction, 30(5), 1-24.

[8] Martinez, C., Wong, H., Ivanov, D., & Patel, S. (2023). Real-time job recommendation using machine learning and user behavior analysis. IEEE Transactions on Neural Networks and Learning Systems, 34(11), 8234-8247.

[9] Thompson, G., Chen, F., Rossi, M., & Ahmed, K. (2023). Distributed architecture for large-scale employment data processing. IEEE Transactions on Parallel and Distributed Systems, 34(7), 1823-1836.

[10] Wang, Y., Brooks, J., Kowalski, A., & Nair, R. (2023). Natural language processing for job description analysis and skill extraction. Computational Linguistics, 49(2), 387-412.

[11] Garcia, R., Tanaka, M., Smith, P., & Krishnan, V. (2022). Privacy-preserving data management in employment platforms. IEEE Transactions on Dependable and Secure Computing, 19(6), 3789-3802.

[12] Lee, H., Murphy, K., Zhao, L., & Okafor, C. (2022). API integration strategies for multi-platform employment data aggregation. ACM Computing Surveys, 55(8), 1-35.

[13] Foster, D., Yamada, K., Silva, J., & Rahman, S. (2022). User authentication patterns in modern web applications. IEEE Internet Computing, 26(4), 45-53.

[14] Mitchell, A., Chang, W., Popov, I., & Gupta, N. (2021). Cross-platform data standardization for employment information systems. Information Systems Research, 32(3), 789-806.

[15] Carter, B., Xu, J., Larsson, E., & Patel, M. (2021). Performance optimization techniques for job search applications. ACM Transactions on Internet Technology, 21(2), 1-28.