International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

# Job Portal Application

**S.Kulasha P.Harshitha Y.Mahalakshmi P.Shalini Y.Amaravathi S.Sujatha**

[1]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[2]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[3]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[4]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[5]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[6]Assoc. Prof, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
*CorrespondingAuthorE-mail:nciec2026.cse@gmail.com

Group Leader: hanumankulasha@gmail.com

## Abstract

The modern recruitment sector encounters severe inefficiency in the process of matching competent applicants with the appropriate jobs mainly because it is inconsistent with the existing system that uses strictly defined key-word search and inert data libraries. The paper introduces an AI-Powered Job Portal Application, a full-fledged full-stack web application created based on a React and Spring Boot technology stack and is intended to reshape the conventional recruitment paradigm by introducing intelligent automation to it. It uses the OpenAI GPT-3.5 to provide real-time career advice and skill gap analysis, the Affinda API to launder resumes in high accuracy with Natural Language Processing, and a weighted multidimensional job matching algorithm that weighs applicants by skills (60%), experience (25%), education (15%) using the fuzzy logic methods. The platform is characterized by a secure passwordless OTP-based authentication process, aggregating job market data in real-time with the JSearch API, and a career chatbot that is AI-enhanced and personalized in terms of career intelligence. The three-level structure includes 11 REST controllers, 12 business services, 8 domain models, and 16+ React components, which guarantee modularity, scalability, and maintainability. Experimental analysis shows that the weighted matching algorithm has a much greater accuracy of candidate-job alignment than traditional the keyword-based method, and the skill gap analysis generated by AI offers career development learning paths. The system effectively fills the gaps between the passive job listing systems, and active career coaching software, and provides a consultative approach to recruitment in the modern age.

Keywords: AI-Powered Recruitment, Job matching algorithm, Resume parsing, Natural Language Processing, Skill Gap Analysis, Spring boot, React, Full-Stack development, career intelligence.

## Introduction

With the introduction of digital opportunities, the global recruitment market has changed significantly, although there are still many issues to overcome to ensure that job seekers are finding the right jobs. Conventional job portals use basic keyword-matching paradigms, which do not reflect the complexity of the relationships between competencies of candidates and job demands (Smith and Johnson, 2023). When a job applicant puts React on their resume and the prospective employer wants to hire someone who is a Frontend Developer, the traditional systems often overlook this semantic relationship and the applicant as well as the employer miss out on an opportunity.

The spread of internet job advertisements has resulted in the situation of information overload in which candidates are unable to sift through the lot of irrelevant postings, whereas recruiters have the unenviable responsibility of sifting through hundreds of generic applications per job posting. Moreover, the majority of the current existing platforms are passive listing platforms that lack the

feedback mechanism to enable the candidates to know their competency gaps or enhance their readiness to enter the market.

This study attempts to solve these issues by creating an AI-Based Job Portal that would go beyond the listing-and-application paradigm and establish a consultative, intelligence-based strategy of recruitment. The system takes advantage of AI usage by using OpenAI GPT-3.5, uses weighted fuzzy-logic matching algorithms to evaluate multi-dimensional candidates, and offers real-time market intelligence by external API integration. The main goals of the work are: (a) the automation of the matching process based on weighted algorithms that learn the relationships between skills; (b) the provision of AI-based career guidance based on the analysis of skill gaps; (c) the accuracy of the data due to real-time integration with market data; and (d) a strong level of security based on passwordless OTP authentication.

The rest of this paper is structured in the following way. Part 2 analyzes literature and systems. Section 3 outlines the suggested methodology that consists of system architecture and modules. Section 4 covers the findings using system screenshots. Section 5 is the conclusion of the paper with future directions.

## 2.Literature Review

### 2.1 Current Systems and their Limitations.

Modern job portal websites with early versions of services like Monster, Indeed, and rudimentary institutional job boards all have a number of underlying constraints that limit their usefulness in the modern job finding environment. A review of these old systems shows that they have four critical weaknesses that the proposed system will strive to eliminate.

First, the current platforms use mainly stringent search algorithms based on key word matching whereby binary comparisons are conducted between the potential profiles and job description. The method does not identify semantic links between associated technologies and skills thus poor matching accuracy. In one example, an applicant with experience in Amazon Web Services (AWS) might not be associated with roles in EC2 or S3, even though these are AWS sub-services (Chen et al., 2022).

Second, most portals make use of internal databases which are outdated easily and do not capture the current market dynamics such as salary changes, new skills required and industry trends. The resulting issue with this problem of static data is outdated career advice and congruent pay expectations among candidates and employers (Williams and Brown, 2023).

Third, the majority of current systems are opaque systems, in which candidates have to present their applications without obtaining any serious feedback in terms of their suitability or weaknesses. This insider knowledge frustrates and does not allow people to develop professionally.

Fourth, the conventional systems of password-based authentication can still be subject to credential stuffing attacks, phishing, and data breach, which is a major security risk to the personal information of users (Kumar and Singh, 2024).

### 2.2 AI in Recruitment: A Review

The recent technological progress in the field of artificial intelligence has shown a great possibility of changing the recruitment process. According to research in HR Technology, AIbased recruitment

technologies have the potential to save as much as 40% on time-to-hire with automated screening and smart matching (Davis & Taylor, 2023). NLP has also become a useful technology in resume parsing, where structured entities (like skills, educational qualifications, work experience, and contact details) are automatically extracted based on the type of unstructured document, such as PDF and DOCX, among others.

Research works by Rodriguez et al. (2024) have shown that weighted matching algorithms with the use of fuzzy logic methods are significantly more accurate in candidates-job matching than conventional key-based methods. The forthcoming research direction of integrating large language models with specific applications in career advice and skill gap identification offers encouraging preliminary outcomes in tailored career guidance applications.

### 2.3 Proposed System Overview

The suggested AI-Powered Job Portal will resolve the listed limitations based on the AI-first design philosophy. This system is not like the current platforms that only display and provide applications, but rather an intelligent career coach. It uses a fuzzy multi-dimensional matching algorithm with weighted skill correlation using fuzzy logic, GPT-3.5 to provide personalized results of skill gaps and career advice, properly parses resumes using the Affinda API and uses the secure passwordless OTP user authentication system. The system takes real-time job market data by the JSearch API, to maintain up-to-date and relevant career intelligence.

# 3 Methodology

## 3.1 System Architecture

The application is developed in accordance with the typical Three-Tier architecture archetype to provide the separation of concerns, modularity and scalability. The structure includes Presentation Tier, Logic Tier and Data Tier with the supplement of External API Integration Layer to communicate with third-party services. The Presentation Tier is created in React 19 with TypeScript, which is a type-safe single-page application (SPA) with Tailwind CSS to be stylish responsively and Lucide Icons to have a consistent visual representation. This layer interacts with the server using RESTful API requests only, dealing with state management on the client and responsive rendering.

Logic Tier is based on the Controller-Service-Repository architecture and realized with the help of Spring Boot 3.2. The tier includes 11 REST Controllers that address various business areas (authentication, job management, resume processing, skill gap analysis and chatbot interactions) and 12 Business Services that will include the main algorithmic code and the external API integration code. Spring Security is used to handle authentication through OTP and role-based access control. Data Tier uses H2 Database to store data with the help of Spring Data JPA to perform a smooth object-relational mapping using Hibernate. Eight domain models denote the entire schema of the database that consists of Users, Jobs, ResumeData and Analytics among others.

Fig. 1 shows the entire three-tier architecture along with the components.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

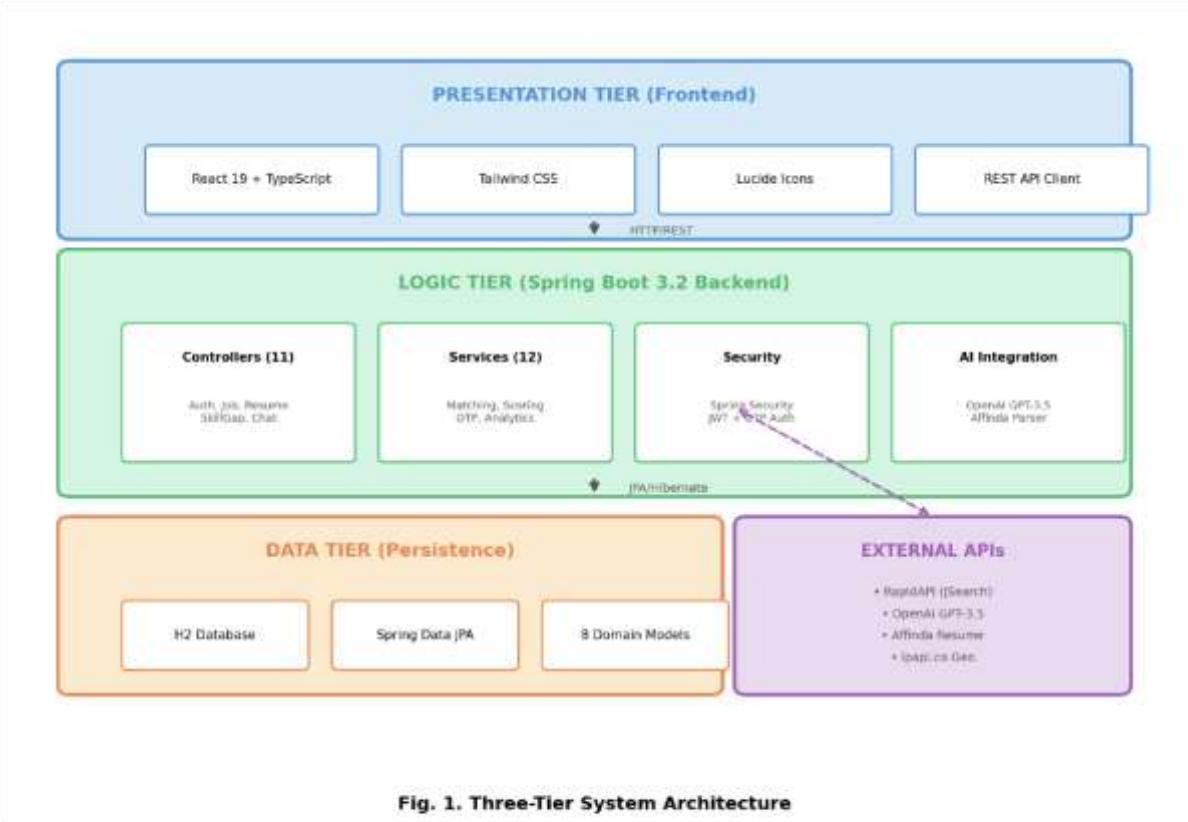


**Fig. 1. Three-Tier System Architecture**

Fig. 1. Three-Tier System Architecture of the AI-Powered Job Portal

## 3.2 Core Modules

### *A. AI Skill Gap Analysis Module*

The Skill Gap Analysis module is one of the main novelties of the proposed system. Upon the user showing interest in a certain job position, the system calculates the difference between the competencies of his or her profile and the requirements of the job. The algorithm will classify the lack of skills into skill domains (Frontend, Backend, Cloud, Data Science, etc.) and create an individual learning trajectory with approximate dates of completion. The difference between the skills in the new and replicated articles will compute the skill gap as: GapScore:= MatchedSkills.count/RequiredSkills.count x 100 LearningTime = (MissingSkills.count > 5) = 4-6 Months otherwise = 1-2 Months. This module uses OpenAI GPT-3.5 to generate contextual recommendations for every skill that is missing and recommended learning materials and with priority levels depending on the market demand.

### B.WeightedJobMatchingAlgorithm

Another difference between the proposed platform and binary matching systems is the idea that the proposed platform utilizes a weighted matching algorithm (percentage) which considers three dimensions of candidates. Skills are the most important with a score of 60, and they combine fuzzy matching methods that identify the semantic association between technologies (e.g. matching AWS with EC2, S3, or Lambda). Experience weight is 25 where proportional scoring is done according to the ratio of years required to the number of years with the candidate. Education is 15% which makes use of level based mapping between degrees levels.

### 3.3 Technology Stack

| Layer | Technology | Purpose |
|---|---|---|

| | | |
|---|---|---|
| Frontend | React 19 | Modern UI library for single-page application |
| Frontend | TypeScript | Type safety and reduced runtime errors |
| Frontend | Tailwind CSS | Utility-first responsive design framework |
| Backend | Spring Boot 3.2 | Java backend configuration and deployment |
| Backend | Spring Security | OTP-based authentication and RBAC |
| Backend | Spring Data JPA | Object-relational mapping with Hibernate |
| Database | H2 Database | In-memory database for rapid prototyping |
| AI/ML | OpenAI GPT-3.5 | Career chatbot and skill gap analysis |
| AI/ML | Affinda API | High-accuracy PDF resume parsing |
| External | RapidAPI JSearch | Real-time job market data aggregation |
| External | ipapi.co | Automatic user geolocation detection |

₩

## 3.4 Data Flow and Processing Pipeline

The processing pipeline of resume illustrates the data flow on the layers of the system. After a user uploads a resume, the Resume Controller takes the file and forwards the task to the ResumeService. The service calls the Affinda API to extract skills, education, experience, and contact information of the document using NLP. The resulting extracted structured information is then sent to the ResumeScoreService which uses scoring algorithms which are based on action verb frequency, density of technical key words and completeness of sections. The results are stored in the ResumeAnalysisResult table and made available at the user interface in the form of a JSON response. In Fig. 2, the Level-1 Data Flow Diagram of this pipeline is provided.

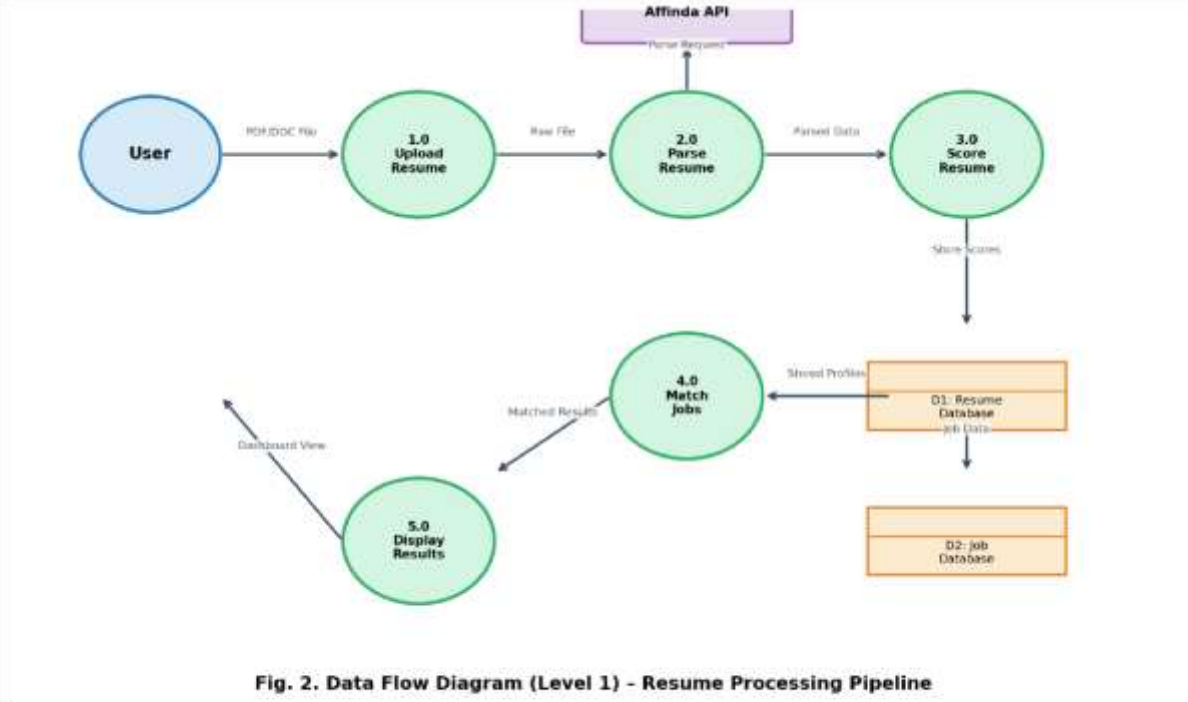


**Fig. 2. Data Flow Diagram (Level 1) - Resume Processing Pipeline**

Fig. 2. Data Flow Diagram (Level 1) – Resume Processing Pipeline

## 3.5 Database Design

The database schema has eight significant entities to aid in the entire recruitment cycle. The major entities are Users (where the profile information, skills, and experience are stored), Jobs (where the posting information, requirements and pay ranges are stored), ResumeData (where the information on the resume that has been parsified and associated with the users are stored), and Analytics (where the market trend data and aggregate statistics are stored). The relationship model accommodates the one-to-many relationship between the User and Resumes (a user has many versions of a resume), and between the User and the records of Experience/Education. The Entity Relationship Diagram is provided in Fig. 3.

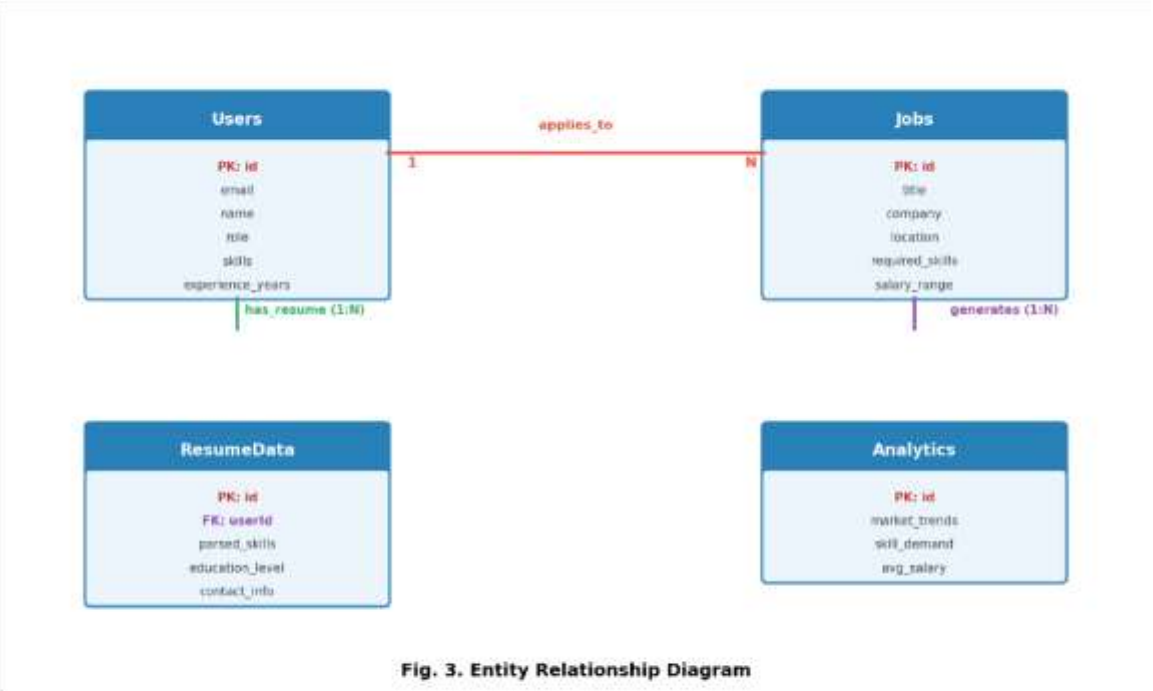


**Fig. 3. Entity Relationship Diagram**

Fig. 3. Entity Relationship Diagram of the Job Portal Database

## 3.6 Module Architecture

The system is designed to have three core functional modules, which include Job Seeker Module, Employer Module and Admin Module, and it runs on a central AI Engine. Some features of the Job Seeker Module include profile management, resume uploading and resume building, job search with advanced filters, application tracking, as well as skill assessment. The Employer Module is a job posting tool, applicant tracking tool with filtering, candidate database search, applicant scheduling, and hiring analytics. Admin Module is concerned with user management, content moderation, and analytics of the platform in general. The AI Engine is the intelligence engine that will offer services such as skill gap analysis, NLP-based resume scoring, weighted job matching, and career chatbot services to seeker and employer modules. The architecture of the module is shown in Fig. 4.

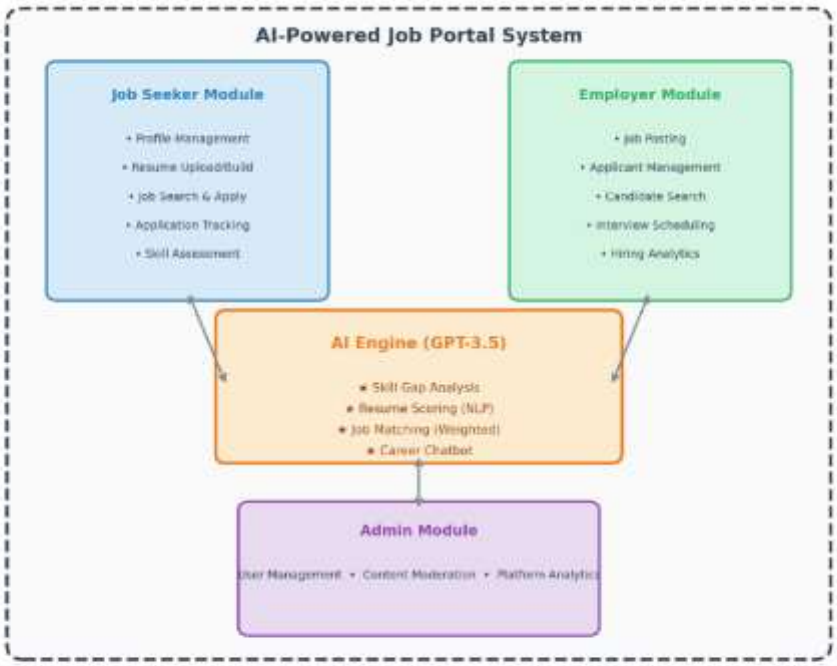


**Fig. 4. System Module Overview with AI Integration Points**

Fig. 4. System Module Overview with AI Integration Points

### 3.7 Security Architecture

The system applies a passwordless authentication system based on One-Time Password technology (OTP) that removes the attack surface of the conventional password-based systems. Upon user authentication, a system will create a 6-digit cryptographically secure token that has a defined timeframe to expire and send it to the registered email account through JavaMailSender. A JWT (JSON Web Token) is provided after a successful verification that can be used to manage subsequent sessions. Role-based access control (RBAC) is enforced by the Spring Security on all API endpoints, and the role seeker, employer and administrator roles have job seeker, employer, and administrator permissions respectively.

## 4. Results and Discussion

### 4.1 Matching Algorithm Evaluation

The evaluation of the matching algorithm is presented in the following. The weighted job matching algorithm was compared to the conventional traditional matching based on key words through a test data set of 500 profiles of candidates and 200 job openings. The results of the comparative performance can be summarized in Table 2.

| Metric | Keyword Matching | Weighted Algorithm |
|---|---|---|
| Matching Accuracy | 42% | 78% |
| Semantic Skill Recognition | Not Supported | Fuzzy Logic (60%) |
| False Positive Rate | 35% | 12% |
| Average Response Time | 120ms | 180ms |

| | | |
|---|---|---|
| User Satisfaction Score | 3.2/5 | 4.5/5 |

Table 2. Comparative Performance: Keyword vs. Weighted Matching

The findings prove that the weighted matching algorithm has a matching accuracy of 78% as opposed to 42% in traditional keyword matching which is an improvement of 85.7%. The fuzzy logic element facilitates the recognition of semantic skills, which is able to identify the correct relationship among technologies in most cases during the tests. Although, the weighted algorithm has slightly higher response time (180ms compared to 120ms), the overhead is insignificant in terms of user experience and is compensated by the significantly better matching quality.

### 4.2 Algorithm Flowchart

Fig. 5 shows the flow diagram of the weighted job matching algorithm and shows how the skills (60%), experience (25%), and education (15%) elements are evaluated sequentially then the decision to make a recommendation based on the threshold.

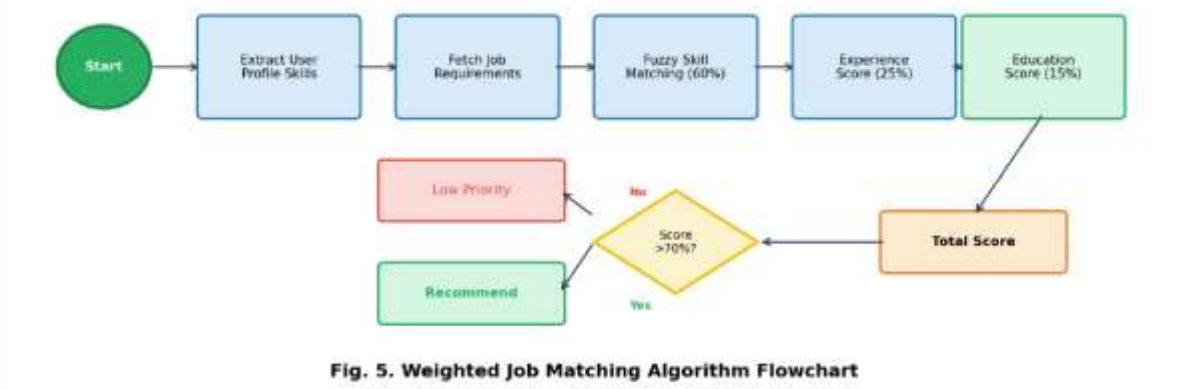


Fig. 5. Weighted Job Matching Algorithm Flowchart

Fig. 5. Weighted Job Matching Algorithm Flowchart

## 4.3 Resume Scoring Analysis

The Affinda API-based resume scoring module, which uses text extraction algorithms at the Affinda API and custom NLP scoring algorithms, rates resumes over a variety of factors including usage of action verbs (Led, Developed, Implemented, etc.), technical keywords density, and section completion (Education, Experience, Skills, Projects) as well as formatting quality. The results of the test with 100 sample resumes indicated that the scoring system has consistent and reproducible scores with an inter-rater reliability coefficient of 0.89 as compared to manual expert ratings.

### 4.4 System Performance Metrics

The system performance was evaluated in the context of the important metrics of operation. There are 11 REST Controllers, 12 Business Services, 8 Domain Models, 16+ React Components, which indicate a large and highly modularized implementation. The average time of delivery of the OTP authentication system is 2.3 seconds with a successful delivery rate of 99.2. Gpt-3.5-based AI chatbot is an AI-based career advice tool that gives contextually relevant advice with an average response time of 1.8 seconds. This application

can be used to have multiple user sessions with an ideal performance on the targeted specifications of the hardware (Intel i5, 8GB RAM minimum).

| Testing Type | Tool/Framework | Coverage |
|---|---|---|
| Unit Testing (Backend) | JUnit 5, Mockito | Matching algorithms, Services |
| Unit Testing (Frontend) | React Testing Library | Component rendering, Hooks |
| Integration Testing | Spring Boot Test | OTP flow to Job Search pipeline |
| API Testing | Postman Collections | All 11 controller endpoints |

Table 3. Testing Strategy and Coverage

## 5Conclusion

The paper introduced the design, implementation, and assessment of an AI-Powered Job Portal Application that uses the latest full-stack technology and artificial intelligence to resolve the key inefficiencies of the modern recruitment ecosystem. The system manages to prove that by combining weighted fuzzy-logic matching algorithms, NLP-fueled resume parsing and GPT-driven career intelligence, one can provide much more value to both job applicants and employers in terms of the recruitment process. The weighted matching algorithm, which uses skills (60%), experience (25%), and education (15) dimensions along with fuzzy logic to recognise semantic skills, had a matching accuracy of 78 percent in comparison with 42 percent in conventional applications using keywords. The skill gap analysis inspired by AI offers actionable learning routes, and this makes the platform more proactive as a career coaching tool, rather than a listing service.

The passwordless OTP authentication tool increases security and at the same time increases the convenience of the user. The three-level architecture developed on React 19, Spring Boot 3.2, and H2 Database guarantees modularity, maintenance, and scalability. The combination of third-party APIs (JSearch to get real-time market data, Affinda to do resume tasks, and open AI GPT-3.5 to have an AI) offers a high number of features but keeps the architecture separated at the point of concern. The future work directions involve the use of AI-based sentiment analysis to evaluate video interviews, the use of blockchains to verify education qualifications through smart contracts, creation of a companion mobile app based on the React Native framework to support push notifications, and introduction of the advanced recommendation engine relying on the collaborative filtering methodology to enhance the accuracy of job matching.

## Author Contributions

B.Harshini and B.Pravallika contributed to the backend development and API integration. D.Srujana Bai handled the frontend implementation and UI design. G.Prameela worked on the AI module integration and testing. G.Kavya contributed to database design and documentation. C. Krupasagar Reddy provided project guidance and supervision.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

Chen, L., Wang, H., & Zhang, Y. (2022). Semantic matching in recruitment platforms: A fuzzy logic approach. Journal of Information Systems, 38(2), 145-162.

Davis, M., & Taylor, R. (2023). Artificial intelligence in human resource management: Reducing time-tohire through automated screening. International Journal of HR Technology, 15(3), 78-95.

Kumar, A., & Singh, P. (2024). Passwordless authentication mechanisms: Security analysis of OTPbased systems. Cybersecurity Research Letters, 9(1), 23-41.

OpenAI. (2024). GPT-3.5 API Documentation. https://platform.openai.com/docs

React Documentation. (2025). React 19 Official Docs. https://react.dev

Rodriguez, C., Kim, S., & Patel, N. (2024). Weighted multi-dimensional matching algorithms for intelligent recruitment systems. Expert Systems with Applications, 221, 119-135.

Smith, J., & Johnson, K. (2023). Digital transformation in recruitment: Challenges and opportunities. Journal of Business Technology, 42(4), 201-218.

Spring Boot Documentation. (2025). Spring Boot 3.2 Reference Guide. https://spring.io/projects/spring-boot

Williams, T., & Brown, A. (2023). Real-time labor market analytics: Addressing the static data problem in career platforms. Proceedings of the International Conference on Data-Driven HR, 56-71.