

<https://doi.org/10.31705/ICBR.2025.2>

AI-POWERED JOB MATCHING SYSTEM FOR JOB SEEKERS AND RECRUITERS

P. Uruthiran*, M.S.M.S. Shibly, N.D. Wattegedara, and A.M.G.T.D. Wijethunga

*Department of Software Technology, University of Vocational Technology,
Sri Lanka
uruthiran@uovt.ac.lk**

ABSTRACT

This study examines the development of an Intelligent Job Matching System (IJMS) using Laravel. Traditional recruitment methods often rely on manual keyword-based searches, leading to inefficient matches between job seekers and employers. This project aims to create a more effective system that extracts and analyses relevant information from CVs and job descriptions to improve job-candidate matching. The system integrates Optical Character Recognition (OCR) for text extraction from images and PDFs, enabling employers to upload job vacancies in various formats. A structured matching algorithm is implemented to ensure accurate job recommendations based on extracted details. To enhance flexibility, efficiency, and iterative improvements, the Agile methodology is followed throughout the development process. The system is designed with a user-friendly interface, guided by wireframes created in Figma, which include login pages, registration pages for both job seekers and employers, dashboards, and job posting functionalities. By automating job matching, reducing manual filtering, and streamlining the recruitment process, this system aims to significantly improve the job search experience for both job seekers and employers. Paragraphs and references are not allowed. Abbreviations can be used if they have been defined previously.

Keywords: Artificial Intelligence, Job Matching, Optical Character Recognition (OCR), Recruitment Automation, Resume Parsing

1. Introduction

Job seekers and employers face mounting challenges in the evolving global market, including information overload, a lack of personalization, and matching inefficiencies. These are compounded by traditional job platforms' dependence on rigid keyword-matching algorithms. Such systems fail to interpret nuanced candidate profiles or dynamic employer requirements, frequently overlooking essential transferable skills, soft skills, and shifting user expectations (Begum, Mairaj Unisa et al., 2025). The core of job searching is effectively aligning applicants with relevant job opportunities. When successful, this process helps individuals find optimal employment, boosting their wages and contributions. The primary obstacle, however, is an information gap: job seekers frequently lack adequate knowledge of market vacancies, and employers similarly lack comprehensive data on the applicant pool (Alksasbeh et al., 2021). This study aims to address these challenges by developing an intelligent job management system that provides enhanced features to improve recruitment experience for both job seekers and employers. By integrating cutting-edge technologies such as Optical Character Recognition (OCR) and offering an intuitive interface, this system will revolutionize the way candidates search for jobs and companies manage their hiring processes (Open-Source Project, 2025).

1.1. Problem Statement

Current job platforms are inefficient for both job seekers and employers. Job seekers encounter irrelevant listings and complicated applications, while employers struggle with managing applications and ensuring security. There is a lack of an integrated, user-friendly solution that addresses these challenges. This study aims to create a job management system that improves efficiency and user experience by offering personalized job recommendations, secure registration, CV uploads in various formats, and OCR for job vacancy extraction.

1.2. Objectives of the Study

1. To develop an intelligent job management system with efficient features that enhance the experience of job seekers and employers.
2. To establish a comprehensive job platform, the system will enable users to upload CVs in multiple formats (PDF, DOCX, etc.) and facilitate user registration with email verification.

3. To provide unique interfaces for both job seekers and employers, featuring dedicated user profile sections for managing personal information.
4. To provide filterable recommended job listings, while employers will gain access to an insightful dashboard for managing vacancies.
5. To allow uploading files as text, images, or PDFs.
6. To extract text from image or PDF-based job postings for enhanced processing, use OCR technology.
7. To develop a user-friendly interface will ensure seamless interaction for all stakeholders.

2. Literature Review

Sridevi and Kamala (Sridevi & Suganthi, 2022) stated that to address the demanding challenge organizations face in hiring candidates meeting specific Job Description (JD) requirements, an AI-based system predicts suitability by analyzing Candidate Resumes (CR) against the JD through clusters of primary skills, secondary skills, adjectives, and adverbs. The Intelligent Agent-Based Job Search System revolutionizes job searching by using Artificial Intelligent learning are Random Forest, K-Nearest Neighbors, Support Vector Machine and Natural Language Processing to analyze user profiles, skills, historical data, job descriptions, and resumes, delivering personalized, efficient matches through an intuitive interface that displays match percentages and real-time listings (Begum, Mairaj Unisa et al., 2025). Alksasbeh et al. (2021) stated the core contribution of their work is a job searching system (JSS) that matches applicants' CVs with optimal job opportunities by implementing a pipeline (segmentation, tokenization, part-of-speech tagging, gazetteer lookup, and fuzzy inference) to extract and structure data from job announcements and CVs into a fuzzy parameterized framework, then calculates similarity scores to rank the best matches in descending order. Nanda et al. (2011) stated that traditional applicant tracking systems relied on keyword-matching, often eliminating qualified candidates unfamiliar with resume optimization while enabling tech-savvy applicants to game the system. This research proposes intelligent process-tracking software to address these limitations for job seekers.

This project aims to develop an intelligent job management system with features like personalized job recommendations, secure registration, CV

uploads in multiple formats, and OCR for extracting job vacancies, providing an efficient and user-friendly experience for both job seekers and employers.

3. Methodology

This study uses Agile methodology for all tasks. Agile allows flexibility, iterative development, continuous feedback, and risk management, making it ideal for complex and innovative projects like “The 17th State of Agile Report (2025)”, which has eight phases of development are stated here (Sommerville, 2016).

Phase 1: Project Initiation & Planning (Sprint 1)

To define the system's scope, it will first capture user stories reflecting core needs for both job seekers (e.g., profile management, job search) and employers (e.g., vacancy posting, candidate screening). Next, it will identify the required technologies and frameworks, such as Laravel for backend development and OCR libraries for document processing and establish the foundational database schema and API structure (Installation, 2025). Finally, it will design wireframes and UI mockups to visualize the distinct interfaces for job seekers and employers, ensuring intuitive workflows.

Phase 2: User Authentication & Profile Management (Sprint 2)

This phase will implement core authentication and user management features, including user registration with email verification and secure login/logout functionality with robust session management. We will develop dedicated user profile sections for both job seekers and employers to manage their information. Additionally, critical security measures such as password encryption and role-based access control (RBAC) will be implemented to protect user data and ensure appropriate system access.

Phase 3: Job Posting & Management (Sprint 2)

This phase will enable employers to upload job vacancies in flexible formats (text, image, and PDF), implement OCR technology to automatically extract and process text from image/PDF-based postings, and develop an intuitive job management dashboard for employers to create, monitor, and optimize their listings.

Phase 4: Job Recommendations & Filtering (Sprint 2)

This phase will implement a personalized recommendation engine that analyzes job seeker profiles and CV content to suggest relevant opportunities, while providing intuitive filtering options for refining these recommendations. Additionally, we will optimize the performance and scalability of both job search and recommendation functionalities to ensure responsive, efficient user experiences.

Phase 5: Employer Dashboard & Insights (Sprint 2)

This phase will develop an employer dashboard enabling efficient review of candidate CVs, provide actionable analytics on job post-performance (including views, applications, and match rates), and enforce role-based access control (RBAC) to ensure secure, permission-gated access to sensitive hiring data and functionality.

Phase 6: CV Upload & Processing (Sprint 3)

This functionality will enable job seekers to upload CVs in PDF and DOCX formats, implement OCR technology to automatically parse and extract text from these documents, store the parsed data in a structured format for efficient job matching, and provide intuitive interfaces for users to review and edit extracted information before finalizing their profiles.

Phase 7: System Security & Optimization (Sprint 3)

This phase focuses on system hardening and optimization, conducting penetration testing to identify vulnerabilities, followed by timely security patching, while simultaneously improving backend performance and scalability to handle increased load. We will optimize OCR processing for greater accuracy and faster extraction speeds and implement comprehensive logging and monitoring solutions to ensure ongoing system stability and proactive issue detection.

Phase 8: Final Testing & Deployment (Sprint 3)

In the final pre-launch phase, we will execute rigorous end-to-end system testing across both job seeker and employer workflows to validate functionality, usability, and integration. Identified bugs will be resolved, and actionable feedback from testing cycles will be incorporated through iterative refinements to ensure a robust, production-ready platform for all users (Kaur & Bahl, 2016).

3.1. Approach

The proposed Intelligent Job Management System for Job Seekers and Employers is a web-based platform designed to streamline the recruitment process for both job seekers and employers. The system is built using Laravel for the backend and Blade templates for the front end. The approach focuses on transforming user-provided data into meaningful job-candidate matches using OCR technology and a custom-built matching algorithm (Installation, 2025).

Input

o Job seekers:

- o Seeker Registration (Username, Password, etc.)
- o Uploaded CVs (in PDF, DOCX, or image formats)
- o Applying for a recommended job

o Employers:

Process

- o Company details and registration info
- o Job vacancies (submitted as text, PDF, or image)
- o Shortlist the selected candidates

User Registration

New users register and verify their emails to gain access to the platform.

o CV upload and OCR processing:

Job seekers upload CVs, which are processed using Optical Character Recognition (OCR) to extract text from PDF and image formats

o Job posting and parsing:

Employers post job vacancies, and text is extracted from uploaded content using OCR for analysis.

o Matching algorithm:

The job matching component compares job titles from the job seekers with the required job title with relevant job vacancies.

o User interactions:

Output

- o Job seekers can browse matched job listings, update profiles, and manage preferences.
- o Employers can view matched candidate profiles, manage job listings, and send notifications.

o Tailored job recommendations for job seekers

normalization.

Use Case Diagram: Figure 2 is a Use Case Diagram for an Intelligent Job Management System. It visually represents the interactions between the system and its users (actors), which include Employee and Employer.

Actors: Employee (Orange figure) – A user looking for a job; Employer (Yellow figure) – A user posting and managing jobs.

Use Cases (Ovals): These are the actions or services provided by the system. Login – Used by both employees and employers to access the system; Register – Used to create a new account; Reset Password – Optional extension from login, used to recover access; Manage Profile – General profile management for users.

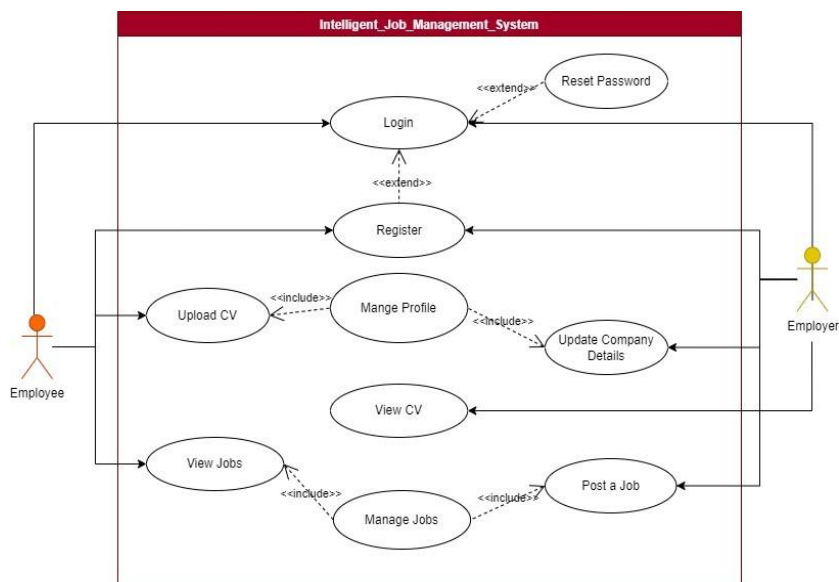


Figure 2: Use Case

Includes: Upload CV (for Employees); Update Company Details (for Employers); View CV – Employers can view the employee’s CV; View Jobs – Employees browse available job listings; Manage Jobs – Employers handle their job postings.

Includes: Post a Job.

Relationships: <<include>>: Indicates that a use case always includes

another use case as part of its process; <<extend>>: Indicates that a use case optionally extends another under certain conditions.

Use Case Flows:

- o **Employer Flow:** Can Register, then Login; After logging in, you can: Manage Profile, Update Company Details, Manage Jobs, Post a Job, View CV (of applicants).
- o **Common Functionality:** Both actors can use Login and Register; Reset Password is an extension of Login.

Wire Frames: The key wireframes include a login page for authentication. Figure 3, separate registration pages are shown Figure 4 and Figure 5 for job seekers and employers, a job seeker home page for profile and job browsing is shown Figure 6, an employer dashboard for managing job postings and applicants is shown Figure 7, an account settings pages for profile customization is shown Figure 8, and a job posting interface for employers to list vacancies are shown Figure 9, job seeker account setting page is shown Figure 10.

The image shows a wireframe for a login page. At the top, there is a black navigation bar containing the word "Logo" on the left and a list of links: "Home", "Login", "Job Seeker", and "Employer" on the right. The main body of the page has a light pink background with vertical stripes. The word "Login" is centered in a large, bold, black font. Below this, there are two input fields: "Email Address" and "Password". A link "Forgot password?" is positioned below the password field. At the bottom of the form is a blue button with the text "Login" in white.

Figure 3: Login Page

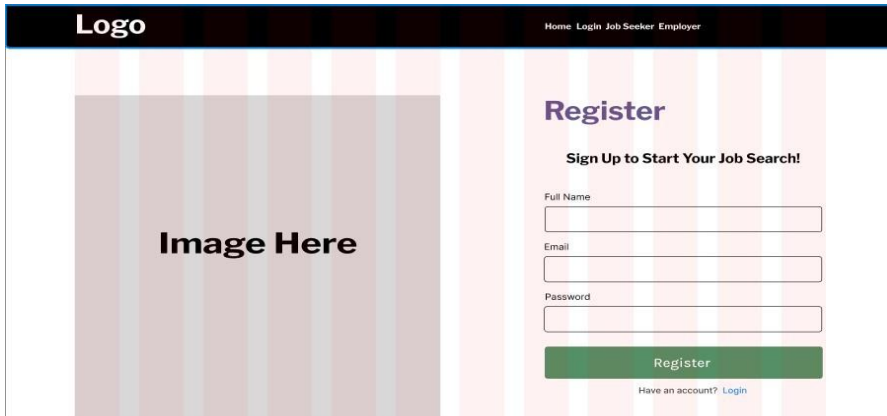


Figure 4: Register Page for Job Seeker

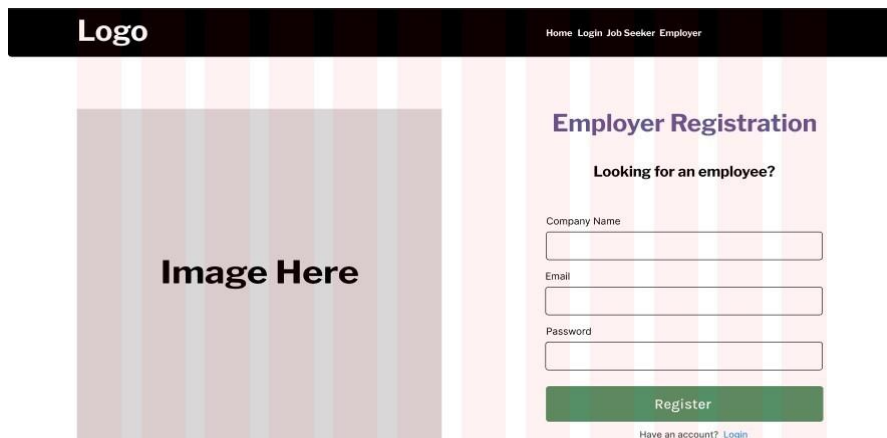


Figure 5: Register Page for Employer

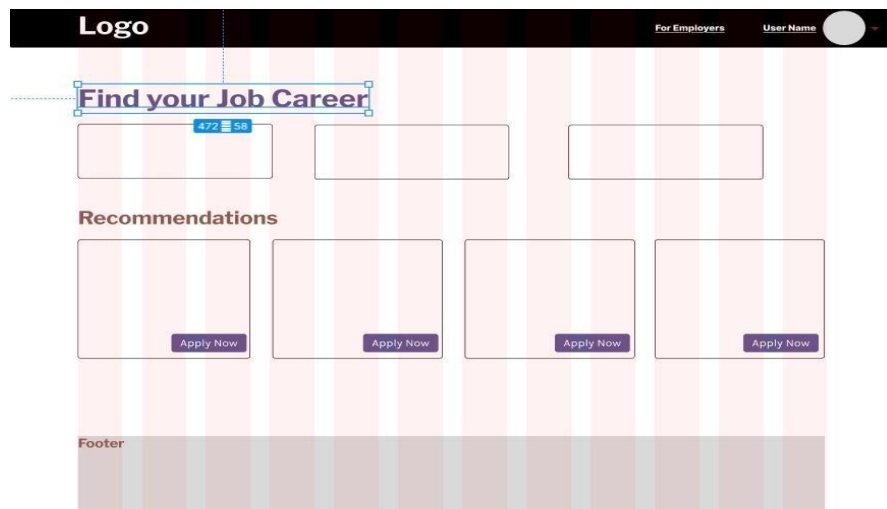


Figure 6: Find Job Career Page

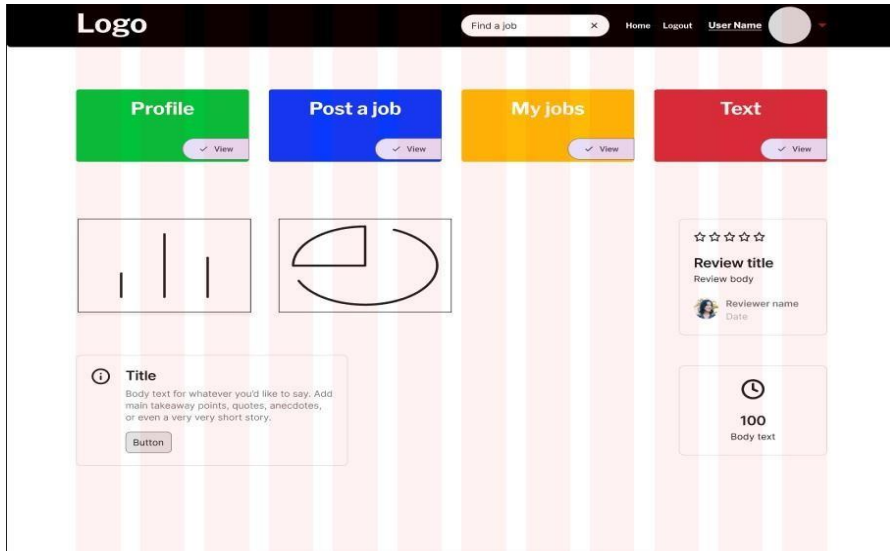


Figure 7: Employer Dashboard

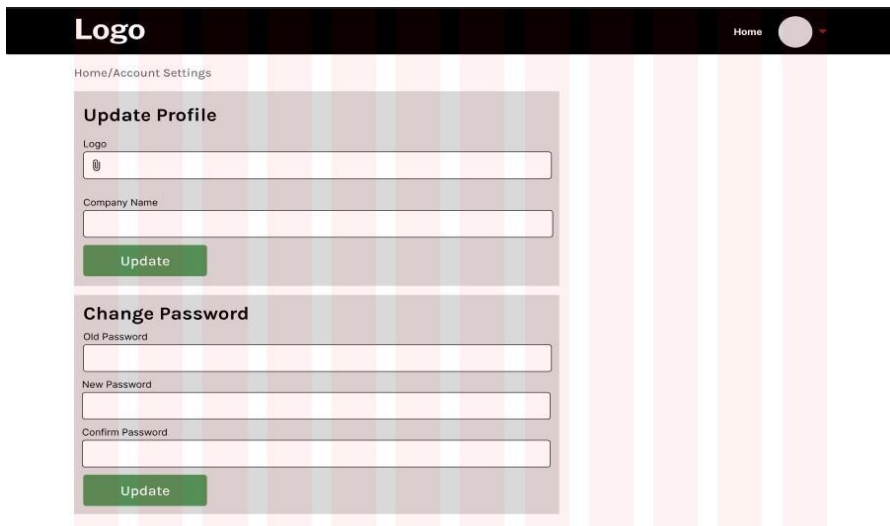


Figure 8: Employer's Account Setting

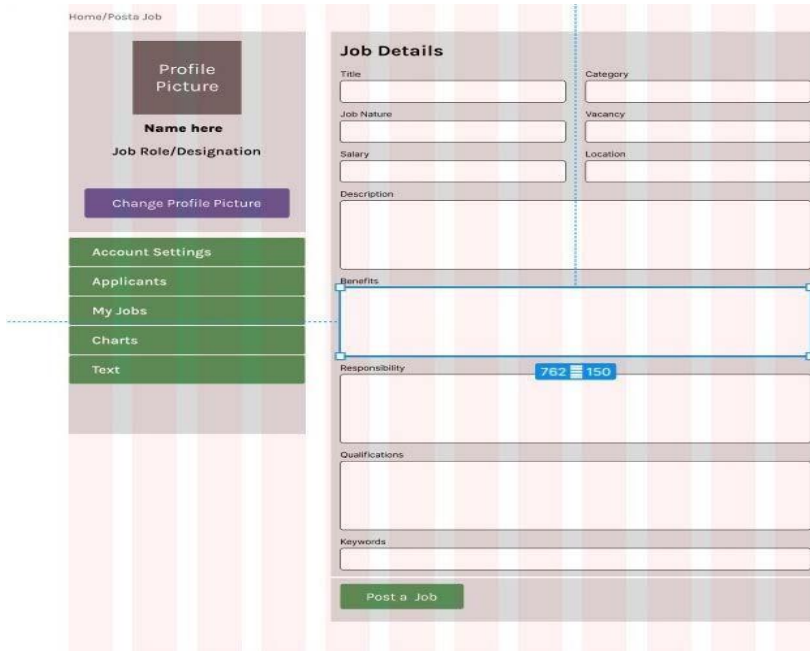


Figure 9: Job Posting

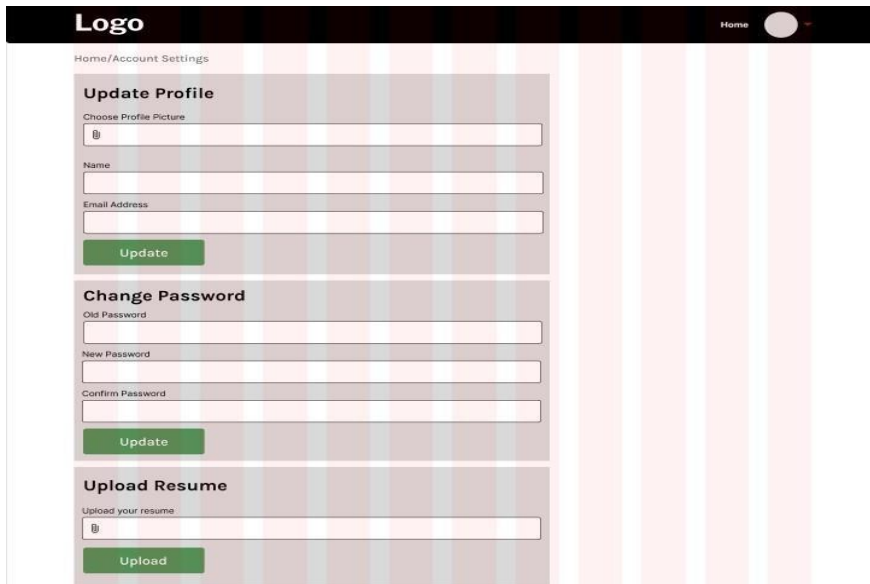


Figure 10: Job Seeker's Account Setting

3.3. Implementation

In the implementation phase of the Intelligent Job Matching System, wireframes were designed using Figma to provide a structured visual

representation of the system's user interface. These wireframes serve as the blueprint for the front-end development, ensuring a user-friendly and intuitive experience. These wireframes guided the development process, ensuring consistency in design and usability.

System Architecture Implementation

The Intelligent Job Management System follows a modular and scalable system architecture, integrating multiple components to ensure seamless functionality. The architecture is based on the Model-View-Controller (MVC) pattern, utilizing Laravel for backend development, Blade templates for frontend design, and MySQL as the database for structured data management. The system is divided into three main layers:

1. Presentation Layer (Frontend):

- o The user interface is designed using Blade templates, ensuring a responsive and user-friendly experience for both job seekers and employers.
- o Users can register, log in, upload CVs, post job vacancies, and view matched results through an interactive dashboard.

2. Application Layer (Backend):

- o Developed using Laravel (PHP framework), this layer handles user authentication, data processing, and system logic execution.
- o It integrates Python-based OCR services for CV parsing and matching purposes.
- o A matching mechanism is implemented to compare extracted CV data with job requirements for improved job-candidate pairing.

3. Data Layer (Database & Storage):

- o A relational database (MySQL) stores user profiles, job postings, and parsed CV data.
- o Uploaded documents (CVs) are stored securely in the file system or cloud storage.

API Implementation

As part of the IJMS system, a Flask-based REST API was implemented to automate the extraction of job-related information from uploaded documents such as CVs and vacancy advertisements. This API forms a

core component of the system by enabling intelligent processing of unstructured data and linking it with structured information.

Role of the API in the System

The API performs two critical functions within the system:

1. **Resume (CV) Processing:** When a job seeker uploads a CV in PDF, Word, or image format, the system sends the file to the API. The API extracts the relevant text, identifies the most probable job title, and maps it to a predefined job category. The categorized data is then returned to the Laravel frontend for display and optional user confirmation before being stored in the database.
2. **Vacancy Advertisement Processing:** Similarly, when an employer uploads a job vacant document or image, the file is passed to the API. The system extracts the job title and matches it with an internal category, allowing the system to automatically suggest the vacancy to suitable candidates who have matching profiles.
3. This intelligent document analysis minimizes the need for manual input, reduces errors, and significantly improves the user experience by enabling fast and accurate data processing. stored in a MySQL database.

3.4. How the System Works

The job management system operates through an interactive, role-based platform that streamlines the recruitment process for both job seekers and employers. The system is designed to manage data input, processing, and output through a series of intuitive and secure features, as shown in Figure 11. The system consists of four core workflows:

- User Registration and Access
- Job Application & Users' Actions
- CV Processing & Vacancy Matching
- Notifications & Communication

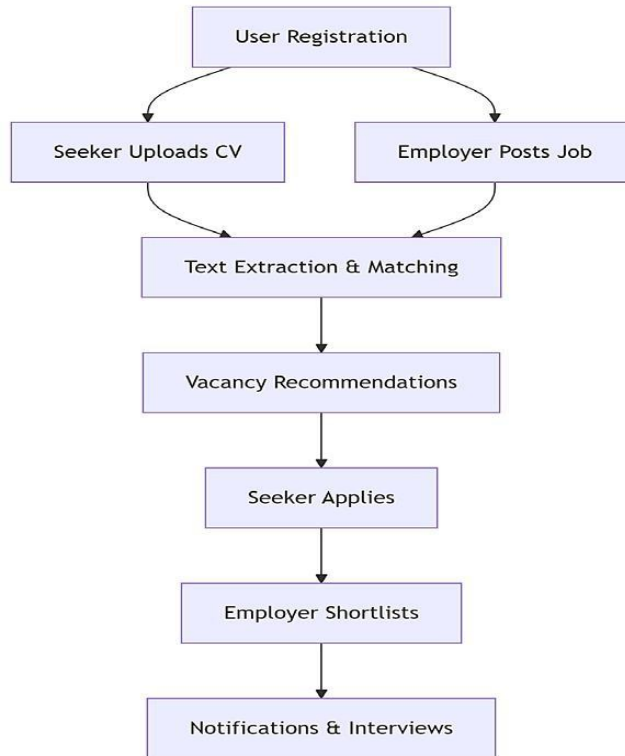


Figure 11: Workflow Diagram for Intelligent Job Management System

The screenshot shows the JobFindr website interface. At the top, there is a navigation bar with 'Home', 'Login', 'Job Seeker', and 'Company' links. Below the navigation bar, the main heading is 'Find your next employer'. There are two search input fields: one for 'UoVT' and another for 'OUEL'. Below the search fields, there is a 'Recommended Jobs' section with a grid of job listings. Each listing includes the job title, location, salary, and an 'Apply Now' button. The jobs listed are: Software Engineer (UoVT, Colombo E, \$100,000.00), Frontend Developer (UoVT, Nugegoda, \$50,000.00), React Developer (UoVT, Maradana, \$850,000.00), and Machine Learning Engineer (UoVT, Rajmahana, \$450,000.00). Below the recommended jobs, there is a 'Recent opening' section with three listings: Machine Learning Engineer (Fulltime, \$450,000.00), React Developer (Fulltime, \$850,000.00), and Frontend Developer (Fulltime, \$50,000.00). At the bottom, there is a footer section with 'ABOUT COMPANY', 'SEARCH' (with a search bar and contact information), and 'OPENING HOURS' (with a table showing hours for Mon-Thu, Fri-Sat, and Sunday).

Figure 12: Open Landing View

Recommended Jobs Salary ▾ Date ▾ Job type ▾

Fulltime

Software Engineer

uovt
Colombo 5

\$100,000.00 Apply Now

Fulltime

Frontend Developer

uovt
Nugegoda

\$50,000.00 Apply Now

Fulltime

React Developer

uovt
Maradana

\$850,000.00 Apply Now

Parttime

Mashine Learning Engineer

uovt
Rathmalana

\$650,000.00 Apply Now

Recent opening

Mashine Learning Engineer Parttime \$650,000.00 View

React Developer Fulltime \$850,000.00 View

Frontend Developer Fulltime \$50,000.00 View

Figure 13: Seeker Views Recommended Jobs

Dashboard
Hello, uovt!

Your trial was expired on 2023-06-04
Your membership was expired on

Total jobs posted

4

Your jobs

Company Profile

1

Your profiles

Subscription

\$80

Monthly (includes tax)

Total applications

18

Your applications

Area Chart Example

Area chart showing data from Mar 1 to Mar 12.

Bar Chart Example

Bar chart showing data from January to June.

Name	Position	Office	Age	Start date	Salary
Tiger Nixon	System Architect	Edinburgh	41	2010/04/15	\$320,000
Clement Baines	Accountant	Sage	43	2010/07/05	\$175,750
Adrian Cox	Senior Technical Author	San Francisco	66	2009/01/12	\$98,000

Figure 14: Dashboard View

Verify Email Address

From: JobFindr <hello@jobfindr.com>
To: <qaq@aaq.com>

2025-03-28 07:15, 13 KB

Show Headers

HTML HTML Source Text Raw Spam Analysis HTML Check Tech Info

JobFindr

Hello!

Please click the button below to verify your email address.

Verify Email Address

If you did not create an account, no further action is required.

Regards,
JobFindr

If you're having trouble clicking the "Verify Email Address" button, copy and paste the URL below into your web browser: <https://1127.0.0.1:3000/email/verify/172/aaq-bb3y6cor9844b32a387054bw45535d0016967bwpvise-17431497739?signature=c1>

Figure 15: User Email Verification

4. Results and Discussion

4.1. Results

The implementation of the Intelligent Job Matching System has the following key results and findings.

User Engagement and Experience

- o The notification system kept job seekers updated on new job matches, and employers are informed about potential candidates.
- o A user-friendly interface enhanced interaction, making it easier for job seekers to browse matches and for employers to filter candidates.

System Scalability and Performance

- o The system handled multiple concurrent users efficiently without significant performance degradation.
- o Database queries and matching algorithms were optimized to ensure fast processing and quick response times.

JMeter Performance Testing Analysis

Apache JMeter was utilized to simulate multiple user requests and assess the performance and scalability of the IJMS underload. The test script (HTTP Request Test) was configured with a thread group named "Tech Jobs Thread Group," representing virtual users interacting with the system's endpoints.

Overview of the Results

- o Total Samples Tested: 500
- o Average Response Time: 602ms
- o Deviation: 278ms (indicating variance in response times among requests)

All requests passed successfully (green checkmarks), meaning the system handled the simulated load without any failures or HTTP errors shown in Figure 16.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status
474	19:18:06.327	TechJobs Thread Gro...	HTTP Request	617	Success
475	19:18:06.926	TechJobs Thread Gro...	HTTP Request	480	Success
476	19:18:07.526	TechJobs Thread Gro...	HTTP Request	462	Success
477	19:18:08.126	TechJobs Thread Gro...	HTTP Request	773	Success
478	19:18:08.742	TechJobs Thread Gro...	HTTP Request	1002	Success
479	19:18:09.327	TechJobs Thread Gro...	HTTP Request	1089	Success
480	19:18:09.930	TechJobs Thread Gro...	HTTP Request	1096	Success
481	19:18:10.526	TechJobs Thread Gro...	HTTP Request	856	Success
482	19:18:11.126	TechJobs Thread Gro...	HTTP Request	714	Success
483	19:18:11.727	TechJobs Thread Gro...	HTTP Request	501	Success
484	19:18:12.327	TechJobs Thread Gro...	HTTP Request	460	Success
485	19:18:12.928	TechJobs Thread Gro...	HTTP Request	456	Success

Figure 16: Results of JMeter Performance Test Analysis

Key Observations

- o Consistent Success Rate: All HTTP requests were successful, indicating reliable backend operations under concurrent load.
- o Latency Range: Most requests had latency between 300–600ms, with some peaking above 1000ms, which is acceptable under load but suggests areas for backend optimization (e.g., DB queries or file parsing).
- o Bytes Transferred: The response payload sizes remained consistent (~66 KB), indicating predictable system behavior.
- o Connect Time: Most requests had a connect time of 1–2ms, demonstrating a fast network handshake and efficient server responsiveness.

The JMeter test verifies that the Laravel-based Job Matching System can handle multiple simultaneous user interactions, such as employers posting jobs or job seekers uploading their CVs, without experiencing crashes or performance degradation. This proves the scalability and reliability of the system, two critical non-functional requirements.

Reduction in Job Vacancy Filling Time

- o Employers reported that the system helped in identifying suitable candidates faster, reducing the time required to fill vacancies.

Security and Data Protection

- o Secure authentication and data encryption techniques ensured that user information was protected from unauthorized access.

4.2. Discussion

The development and implementation of the Intelligent Job Matching System using Laravel demonstrated several key improvements over traditional job-matching platforms. The system effectively addressed common issues such as inaccurate job recommendations, inefficient resume parsing, and slow recruitment processes.

User Engagement and System Usability

- o The notification system enhanced user engagement, keeping job seekers and employers updated about potential matches.
- o The user-friendly dashboard and search features improved usability, making it easier for job seekers to navigate and apply for jobs.

Scalability and System Performance

- o The system handled multiple concurrent users efficiently, but further scalability testing is required to support a significantly larger user base.
- o Optimization of database queries and background processing for large data uploads can further improve performance.

Security and Data Privacy Considerations

- o Secure user authentication and encryption techniques were implemented, ensuring that sensitive data (e.g., CVs, employer details) remained protected.
- o Future enhancements should include multi-factor authentication (MFA) for additional security.

5. Conclusion and Implications

The Intelligent Job Matching System has proven effective in streamlining the recruitment process by efficiently connecting job seekers with suitable employment opportunities. Its ability to automate the matching process significantly reduces the time and effort traditionally involved in hiring, benefiting both employers and candidates. The system successfully aligns candidate profiles with job requirements, ensuring a

more targeted and relevant job search experience.

However, while the current performance is commendable, there is still considerable scope for improvement. Enhancements in matching accuracy through more advanced algorithms or AI-driven personalization can ensure even better alignment between job seekers and available roles. Additionally, improving the system's scalability will allow it to handle a growing number of users and data inputs without compromising performance, which is essential for large-scale adoption. Increasing user engagement through intuitive interfaces, personalized notifications, and interactive features could also contribute to a more immersive and productive experience for all users.

By implementing these recommended features and upgrades, the Intelligent Job Matching System can evolve into a more robust, efficient, and user-centric platform. These improvements would not only enhance system performance but also solidify its position as a vital tool in the landscape of modern recruitment and talent acquisition.

References

- Alksasbeh, M., Abukhalil, T., Alqaralleh, B., & Al-kaseasbeh, M. (2021, February). Smart job searching system based on information retrieval techniques and similarity of fuzzy parameterized sets. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(1), 636-645.
- Begum, Mairaj Unisa, Siddartha, K., Sindhu Reddy, K., Sai Varun Reddy, B., Sindhu, N., & Harish, A. (2025, April). Intelligent Agent-Based Job Search System. *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, 8(4), 6151-6155.
- Kaur, N., & Bahl, K. (2016). Performance Testing of the Institute Website Using Jmeter. *International Journal of Innovative Science, Engineering & Technology*, 3(4), 534- 537.
- Nanda, S., Mohanty, R., Sukla, S., & Ghosh, G. (2011, November). Development of an Intelligence Process Tracking System for Job Seekers. *International Journal of Managing Information Technology (IJMIT)*, 3(4), 57-70.
- Open-Source Project*. (2025, March 23). Retrieved from Google Open

Source: <https://opensource.google/projects>.

Sommerville, I. (2016). *Software Engineering*. England: Pearson Education.

Sridevi, G., & Suganthi, K. (2022, August). AI-based suitability measurement and prediction between job description and job seeker profiles. *International Journal of Information Management Data Insights*.

The 17th State of Agile Report. (2025, March 25). Retrieved from State of Agile: <https://stateofagile.com>.