engineering proceedings

MDPI

Proceeding Paper

AI-Driven Web-Based Project Allocation and Resource Matching System Using Fuzzy Similarity and Agile Development †

Maysaa Abd Ulkareem Naser *, Zainab Jaafar Kadheem, Sarah Salem Hazam and Zahraa Mohammed Jabar

Department of Computer Information Systems, University of Basrah, Basrah 61004, Iraq; it232100090@avicenna.uobasrah.edu.iq (Z.J.K.); sarah.salem.h330@gmail.com (S.S.H.); zahraa.mohammed.jabar22@gmail.com (Z.M.J.)

* Correspondence: maysaa.naser@uobasrah.edu.iq

† Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES2026), Bandirma, Turkey, 24–27 June 2026.

Abstract

The efficient allocation of project and resource assignment has been addressed as a major challenge in many modern-day organizations and engineering environments due to trends such as the increase in project complexity and heterogeneous competency requirements for project and personnel, along with an increasing need for transparency in decision-making mechanisms. This paper proposes a web-based project allocation and resource matching system based on AI that automates assignment workflow (using fuzzy similarity matching) coupled with agile software engineering in practice. The proposed framework of this research creates an intelligent decision-support solution that could be applicable for engineering management and organizational management.

Keywords: artificial intelligence; fuzzy matching; resource allocation; decision support systems; web systems; agile development; intelligent matching; project allocation

1. Introduction

With the increasing complexity of projects, heterogeneous competency demands and the dynamic evolution of operational workflows [1], efficient project allocation and intelligent resource matching are emerging as key challenges in contemporary engineering and organizational environments. Traditional manual allocation mechanisms are inefficient, inconsistent, and unable to scale effectively and require substantial administrative overhead [2]. As a result, intelligent allocation systems have gained considerable interest as realistic solutions not only for automating task assignment processes but also for enhancing transparency, operational efficiency and workflow optimization.

In recent times, breakthroughs in the fields of artificial intelligence (AI), decision-support systems and intelligent optimization have been able to provide computational frameworks for processing complex allocation and matching problems [3]. Stable matching algorithms [4], optimization models [5], approximation methods, and intelligent resource assignment techniques have been researched for directing process allocation and workflow management [6]. However, most existing approaches remain theoretical and are not accompanied by integration into deployable web-based architectures capable of enabling real-time organizational operationalization and workflow automation.

Several artificial intelligence approaches include fuzzy logic and fuzzy similarity analysis techniques; eventually these methods have proven to handle inconsistent, indeterminate, and heterogeneous data representations efficiently [7]. Fuzzy matching mechanisms [8]

Check for updates

Academic Editors: Teodor Iliev, Ivaylo Stoyanov, Grigor Mihaylov and Selahattin Kosunalp

Published: 22 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

allow more flexible evaluation of the similarity of competency descriptions against project requirements, which significantly improves allocation accuracy, particularly in environments where the terminology and the definition and scope of skills vary drastically. On top of that, intelligent recommendation and matching systems are gaining more attention in terms of AI-automated organizational management and workflow optimization.

Simultaneously, praxis in the development of scalable adaptive intelligent systems has granted great relevance to the integration of agile software engineering methodologies [9]. Agile methodologies provide features such as iterative development, offering rapid adaptation to requirement changes and improved system maintainability; hence they fit well into the modern web-based engineering platforms [10]. At the same time, lightweight web frameworks including Flask allow modular system designs, the creation of quick versions of deployed systems and more seamless connectivity between intelligent algorithms implemented by machine learning environments, database systems such as SQL or NoSQL and interactive user interfaces [11–14].

We propose an integrated system comprising a deployable decision-support platform developed using AI and fuzzy similarity, combined with intelligent matching algorithms, secure authentication mechanisms and agile software engineering. The proposed framework analyzes insider threats from the perspective of similarity in skills and utilizes it to automate the process of assignment with respect to projects, as well as to support secure workflow management features, real-time analytics and scalability around system operations.

Compared to traditional allocation methodologies, the proposed system leverages intelligent matching algorithms and full web deployment with database integration, authentication controls, analytics dashboards and automated workflow management in a single engineering architecture. This research is most significant for the development of a practicable intelligent decision-support framework based on fuzzy similarity matching using a more efficient and scalable web-based system in engineering, organizational and resource management applications.

2. Methodology

2.1. System Model Definitions

To provide clarity on the system structure and ensure uniform understanding of the proposed framework, we introduce the following definitions: Projects at the center of this study are considered as a collection of commutative and engineering tasks that upon completion require certain technical competences for their implementation. These projects include web development systems, data analytics platforms, intelligent recommendation systems and database optimization tasks. The overall architecture of the proposed intelligent resource allocation system is illustrated in Figure 1.

Resources are assignable entities in the system (such as users or development units) that can be assigned to projects through competency matching. Competencies are specialized technical skills that each user has, such as the programming languages a user knows and whether they know any frameworks, databases or data analysis. This allocation process matches user capacities to project requirements by means of a fuzzy similarity algorithm. If the percentage of matches is above a defined score (for example, 80%) the system computes similarity scores and assigns resources to projects.

https://doi.org/10.3390/engproc2026154098

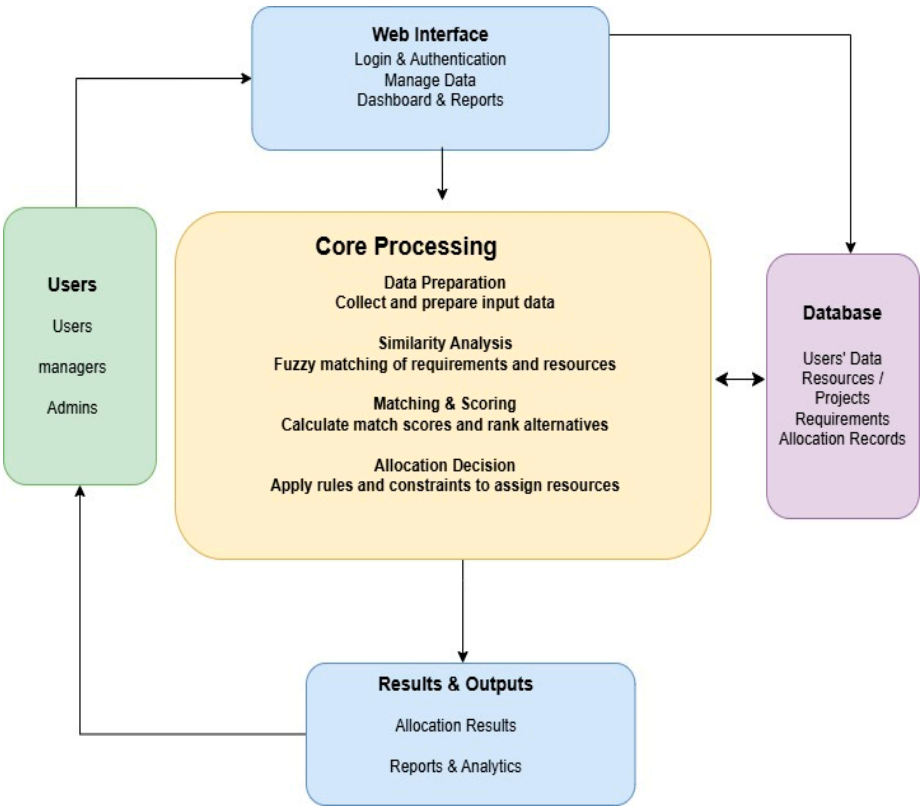


Figure 1. Architecture of intelligent allocation system.

2.2. System Design

Development of a preliminary database schema, workflow modeling, and interface development, a lightweight browser–server architecture was adopted to reduce computational overhead and support real-time interaction between users and the allocation platform. For the web-based application (cyber_spots), the platform backend was developed using Python 3.11 and the Flask 3.0 framework, because it is a lightweight WSGI web application framework that supports dynamic routing and provides functions for defining relationships between HTTP methods. The front-end interface was developed using HTML5, CSS3, JavaScript (ES6), and Bootstrap 5.3 to ensure responsiveness and compatibility across different platforms. SQLite 3.45 was selected as the relational database management system because of its lightweight architecture and ease of deployment. The integration of the presentation, application logic, and database layers enables secure user authentication, effective project management, automated task allocation, and real-time visualization of analytical results.

2.3. Requirement Analysis

Requirement analysis is pivotal in recognizing not just the system objectives, but also the operational workflows, user roles, and functional requirements of a system. The proposed solution outlines multiple system roles, with users, project managers and system administrators. The primary functional requirements include: secure authentication and role-based access control, project creation and management, competency registration and profile management, automated project–resource matching, real-time analytics and reporting, workflow automation and allocation tracking.

Manual allocation approaches to the problem are often inefficient and inconsistent and lack security. Thus, we propose a web-based platform that highlights workflow automation and intelligent decision-support mechanisms.

2.4. System Implementation

The dynamic rendering of content is aided by the user interface that supports AJAX techniques for asynchronous communication. Client-side validation mechanisms were employed to improve usability and reduce server-side processing overhead. We managed to integrate some libraries that made development more efficient and unified the interface: Bootstrap, Tailwind CSS, jQuery, Toastify, Choices.js, and Chart.js. Together these technologies enhanced the quality of responsiveness, maintainability, scalability, and user interaction.

User authentication is secured through bcrypt-based password hashing, with automatic salt generation and configurable computational cost. RBAC denies administrative functions, rolling back access to the system for improvements in security. JSON serialization mechanisms allow for structured data handling, and Local Storage is leveraged to store non-sensitive session information temporarily, improving performance while minimizing superfluous server requests, as shown in Figure 2.

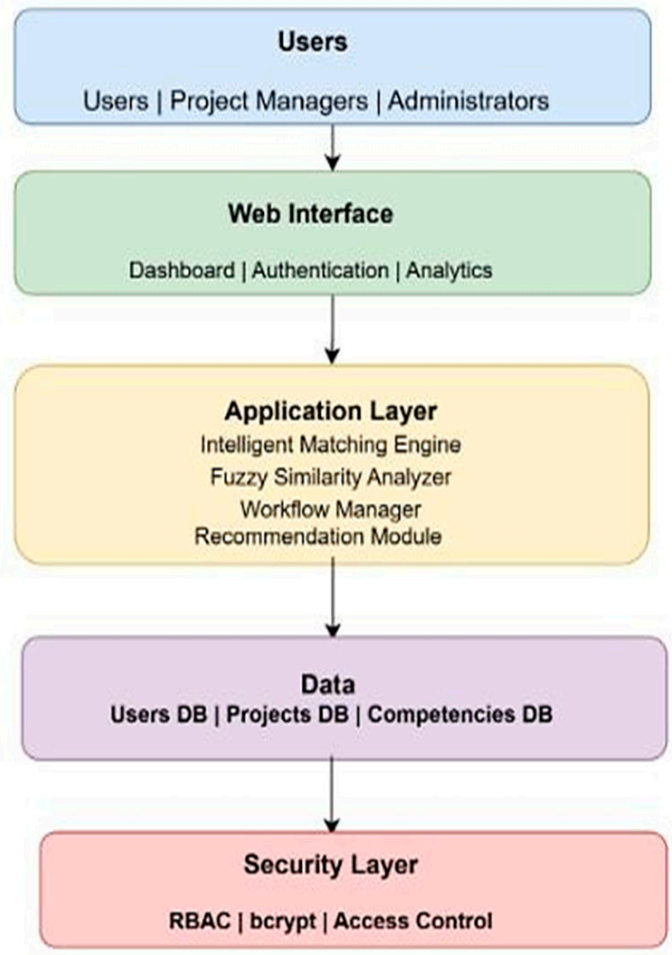


Figure 2. Multi-layer architecture of the intelligent project allocation and matching system.

Flask gives you a flexible and scalable way to build REST services as well as manage the logic behind your intelligent workflows. Clean project structuring and authentication: The framework supports clean project structuring, authentication integration, and secure communication between application layers. Bcrypt is a password hashing function that converts user passwords into secure encrypted values through random salt generation and computation-heavy hash functions. This significantly increases resistance to brute-force attacks and unauthorized access attempts. The database architecture comprises the following major tables:

- User Table: Contains user identity and competency data;
- Competency Table: Contains a list of pre-defined competency categories;
- Project Table: Contains the descriptions and requirement specifications of each project;
- Assignment Table: Contains allocation information and match scores;
- Authentication Table: Contains access credentials and authorization data.

The database design ensures increased performance, consistency, and more efficient read operations.

2.5. AI-Based Resource Matching Algorithm

The intelligent matching engine compares user competencies with project requirements using fuzzy similarity analysis. The algorithm transforms competency sets into structured lists and computes similarity percentages based on configurable threshold values. This approach improves matching robustness in scenarios involving inconsistent terminology, incomplete competency descriptions, and heterogeneous project specifications.

Algorithm Workflow: The matching process can be described through the following steps:

1. Initialize the matching counter;
2. Convert competency entries into structured lists;
3. Compare competency strings using fuzzy similarity functions;
4. Calculate similarity scores;
5. Apply threshold validation;
6. Generate matching percentages;
7. Store allocation results in the database;
8. Display matching recommendations through the dashboard.

The proposed system workflow is implemented through the following Algorithm 1:

Algorithm 1 Fuzzy Matching

```
from fuzzy wuzzy import fuzz
from fuzzy wuzzy import process
def fuzzy_compare_lists_percentage(list1, list2, threshold = 80):
    matched_count = 0
    for item1 in list1:
        best_match, score = process.extractOne(item1, list2)
        if score >= threshold:
            matched_count += 1
    if not list1:
        return 0.0
    percentage = (matched_count/len(list1)) × 100
    return percentage
```

3. Results and Discussion

The experimental results indicate that the intelligent allocation platform employed in this study greatly improved the overall process of allocating projects compared to traditional manual allocation approaches. An assessment was performed with 50 user competency profiles and 15 project requirements obtained from an institutional environment under different allocation conditions. Quantitative analysis showed that the proposed system reduced the average allocation time from 90 min in the traditional manual allocation process to 20 min after system deployment. Further, the automatic allocation mechanism helped save a large amount of manual administrative time by conducting repetitive assignment and retention tasks. These results further validate that the optimized

system successfully simplified workflow processes by minimizing human-centric manual operations while also providing scalable resource allocations.

Additionally, experimental results show that allocation accuracy increased considerably when using the fuzzy similarity matching algorithm. Using the algorithm resulted in better consistency in matching project requirements with user competencies given that it was able to deal with heterogeneous descriptions of competency types and non-uniform language shown by different users. Application of comparative analysis showed that allocation correctness rose from 68% with the manual allocation strategy to 91% by making use of the proposed fuzzy similarity device. The accuracy, recall and F1-score values of the proposed system are 0.89, 0.87 and 0.88 respectively, which shows that it provided strong matching performance under semantically inconsistent competency descriptions. The results validate the feasibility of applying fuzzy similarity methods to real-world matching domains where proficiency descriptions are often partial or semantically inconsistent.

Workflow efficiency metrics also performed significantly better post system implementation. The integration of automated allocation mechanisms, role-based access control and real-time analytics dashboards was shown to improve workflow efficiency from 60% to 88%, as demonstrated experimentally. Table 1 summarizes the performance evaluation results of the proposed intelligent allocation platform compared to the traditional manual allocation approach. These features improved the usability of the system and offered more transparent decision-making processes. They allowed administrators to monitor allocation activities effectively while enabling greater control over user permissions and operational flow management tasks. The visual examination was further confirmed by statistical analysis, which indicated that the observed improvement was statistically significant ($p < 0.05$) and reproducible in terms of evaluation results.

Table 1. Evaluation and achievement scores of the proposed allocation platform.

Metric	Traditional Method	Proposed System	Improvement (%)
Allocation Time (minutes)	90	20	77.8%
Allocation Accuracy	68%	91%	33.8%
Workflow Efficiency	60%	88%	46.7%
Precision	—	0.89	—
Recall	—	0.87	—
F1-Score	—	0.88	—

At the same time, the Flask and SQLite lightweight system architectures showed high affinity for both the rapid deployment of small-scale problems and low infrastructure complexity. The constructed system was deployed in a local institutional environment with moderate hardware resources showing stable operation and low response time during testing. The lightweight architecture reduced the need for system maintenance and also simplified implementation overhead, allowing for its use in small and medium-scale institutional environments. In addition, the design with a modular system made it easier to maintain and suitable for future expansions and possible integrations with other services.

Despite the positive outcomes of the allocation process, several limitations identified by the evaluators should be acknowledged. But, still, the overall utility of the allocation process relies largely on proper user competencies being provided. In some cases, absent or inconsistent competency descriptions might detrimentally impact both similarity scoring and allocation accuracy. Even worse, the proposed system is a simple fuzzy similarity-based one without the ability to employ any form of adaptive learning or behavioral analysis capabilities. Therefore, the system can have problems dealing with very dynamic environments or competency datasets that are always changing.

Future work may focus on integrating machine learning models and intelligent recommendation systems to improve allocation adaptability and predictive performance. Advanced optimization algorithms may also be incorporated to enhance global allocation efficiency and support more complex large-scale allocation scenarios. Furthermore, natural language processing techniques may be integrated to improve semantic interpretation of competency descriptions and reduce ambiguity in similarity matching operations. These enhancements could further improve system intelligence, scalability, matching accuracy, and decision-support capability in real-world institutional applications.

4. Conclusions

This study proposed an AI-powered web-based project allocation and resource matching system to automate intelligent assignment workflows in computational environments of organizations. The framework consists of an easily deployable decision-support platform that incorporates fuzzy similarity matching, secure web architecture (e.g., enabling users with the same capabilities to build dynamic local lists), role-based access control, and agile software engineering practices. The implementation showed how well fuzzy similarity algorithms can perform when faced with inconsistent competency representations, while also improving matching transparency, scalability and operational efficiency. Furthermore, the system decreased manual intervention and improved workflow automation with real-time insights and secure data handling mechanisms.

Author Contributions: Conceptualization, M.A.U.N. and Z.J.K.; methodology, M.A.U.N.; software, S.S.H. and Z.M.J.; validation, M.A.U.N., Z.J.K. and Z.M.J.; formal analysis, M.A.U.N.; investigation, Z.J.K.; resources, S.S.H.; data curation, Z.M.J.; writing—original draft preparation, M.A.U.N.; writing—review and editing, M.A.U.N., Z.J.K., S.S.H. supervision, M.A.U.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request. The data are not publicly available due to privacy and/or confidentiality considerations.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hashimzai, I.A.; Mohammadi, M.Q. The Integration of Artificial Intelligence in Project Management: A Systematic Literature Review of Emerging Trends and Challenges. *TIERS Inf. Technol. J.* **2024**, *5*, 153–164. [CrossRef]
2. Örnek, H.M.; Camci, A. Industry 4.0 and Agile Project Management: Evolution of Leadership Concepts and Roles. *Yönet. Bilim. Derg.* **2024**, *22*, 2601–2635. [CrossRef]
3. Ali, R.; Hussain, A.; Nazir, S.; Khan, S.; Khan, H.U. Intelligent Decision Support Systems—An Analysis of Machine Learning and Multicriteria Decision-Making Methods. *Appl. Sci.* **2023**, *13*, 12426. [CrossRef]
4. Hosseini, H.; Pathak, S. Strategic Aspects of Stable Matching Markets: A Survey. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI 2024), Jeju, Republic of Korea, 3–9 August 2024; pp. 8077–8085. [CrossRef]
5. Baardman, L.; Cristian, R.; Perakis, G.; Singhvi, D.; Skali Lami, O.; Thayaparan, L. The Role of Optimization in Some Recent Advances in Data-Driven Decision-Making. *Math. Program.* **2023**, *200*, 1–35. [CrossRef]
6. Pufahl, L.; Stiehle, F.; Ihde, S.; Weske, M.; Weber, I. Resource Allocation in Business Process Executions—A Systematic Literature Study. *Inf. Syst.* **2025**, *132*, 102541. [CrossRef]
7. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
8. Dubois, D.; Prade, H. *Fuzzy Sets and Systems: Theory and Applications*; Academic Press: Cambridge, MA, USA, 1980.

9. Palopak, Y.; Huang, S.-J. Perceived Impact of Agile Principles: Insights from a Survey-Based Study on Agile Software Development Project Success. *Inf. Softw. Technol.* **2024**, *176*, 107552. [CrossRef]
10. Babu, S. Trends in IT Project Management: Agile Methodologies and Their Impact on Project Success. *J. Recent Trends Comput. Sci. Eng.* **2024**, *12*, 73–81.
11. Grinberg, M. *Flask Web Development: Developing Web Applications with Python*; O'Reilly Media: Santa Rosa, CA, USA, 2018.
12. ForouzeshNejad, A.A.; Arabikhan, F.; Gegov, A.; Jafari, R.; Ichtev, A. Data-Driven Predictive Modelling of Agile Projects Using Explainable Artificial Intelligence. *Electronics* **2025**, *14*, 2609. [CrossRef]
13. Khanna, T.D.; Goyal, P.; Verma, S.; Singh, R.; Jain, A. Hybrid Work Meets Agile Software Development: A Systematic Mapping Study. *arXiv* **2024**, arXiv:2404.09983.
14. Islam, M.R.; Sandborn, P. Multimodal Generative AI for Story Point Estimation. In Proceedings of the 2025 IEEE Conference on Artificial Intelligence (CAI), Santa Clara, CA, USA, 25–27 March 2025; pp. 659–662. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.