IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

AI-Based Skill Recommendation System

S. A. Kurzadkar, D. A. Agrawal, Misha Thakre, Chandrashekhar Tidake, Aadinath Mundhe
Nayana Yembarwar, Ganesh Taynak, Om Kasadiwar, Vishal Patil
Department of Computer Science and Engineering
K. D. K. College of Engineering, Nagpur
skurzadkar@gmail.com, deepti161183@gmail.com,
mishathakre@gmail.com, chandrashekhartidake@gmail.com,
mundheaadinath158@gmail.com, nayanayembarwar@gmail.com,
ganeshtaynak10@gmail.com, kasadiwarom@gmail.com, vp071802@gmail.com

Abstract: *The expansion of digital work environments has increased the need for intelligent platforms that efficiently connect service providers and clients. This study introduces an AI-driven freelancing system designed to improve job matching, candidate selection, and user interaction. The platform supports essential operations such as user registration, profile creation, and job posting, along with advanced features like real-time communication and automated notifications. Artificial Intelligence techniques are employed to analyze user data, enabling accurate skill matching, candidate prioritization, and customized job recommendations. Freelancers are provided with a personalized job feed, while clients can easily identify suitable candidates through ranked suggestions. The integration of cloud technologies ensures secure data handling and seamless system performance. The proposed solution enhances productivity, transparency, and overall user satisfaction in modern freelancing environments*

Keywords: Artificial Intelligence, Freelancing Platform, Job Matching, Personalized Job Recommendation, Real-Time Communication, Cloud Computing, Skill Matching, User Interaction, Data Security

I. INTRODUCTION

The rapid advancement of digital transformation has led to the rise of flexible work models, with freelancing platforms playing a crucial role in the gig economy. These platforms connect skilled professionals with organizations, but often face challenges such as poor communication, delayed payments, limited transparency, and inefficient job matching. Such issues reduce trust and affect collaboration quality. To overcome these limitations, this study proposes an AI-enhanced freelancing platform developed using Flutter, Flask, and Firebase. The system improves communication, task management, and reliability through real-time data synchronization and secure authentication.

Furthermore, modern freelancing platforms require enhanced user interaction and automation to improve efficiency. The proposed system integrates features such as personalized job feeds, real-time chat communication, and notification systems to ensure seamless collaboration between clients and freelancers. These features not only improve usability but also increase transparency and responsiveness within the platform.

II. LITERATURE REVIEW

1. Overview - recommender systems for hiring

Recommender systems offer the basic technology for matching jobs and freelancers. Classical approaches (content-based, collaborative filtering, hybrid) remain the foundation; later works combine text understanding (NLP) with collaborative signals to increase accuracy. For general theory and algorithms, the Recommender Systems Handbook is a strong reference.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

Key reference (book): Ricci, F., Rokach, L., & Shapira, B. (eds.). Recommended Systems Handbook (2015). comprehensive chapters on CF, content-based methods and hybrid systems. (See introductory chapters on hybrid recommenders.

2. Text / NLP methods for job-resume matching

Modern job matching heavily relies on textual similarity between job descriptions and resumes/profiles. Techniques moved from TF-IDF + cosine similarity (classic IR) to modern contextual embeddings (BERT / SBERT) and fine-tuned Siamese networks to compute robust semantic similarity. For TF-IDF and vector models, see Manning et al., Introduction to Information Retrieval chapter on term weighting and vector space (TF-IDF).

For sentence embeddings and Siamese BERT for vacancy-resume matching, see Lavi et al. (2021) "Fine-tuned Siamese Sentence-BERT for matching jobs and resumes (Consultant BERT)" demonstrates practical improvement using SBERT embeddings.

Representative papers: D. Lavi et al., Fine-tuned Siamese Sentence-BERT for matching jobs and resumes, arXiv 2021.

3. Skill graphs / knowledge graphs for skill matching

Several studies propose building a skills knowledge graph (KG) to model relationships between skills, occupations and job postings. A KG helps to infer related or missing skills and supports "skill gap analysis" (what skills a freelancer lacks for a role).

de Groot et al., Job Posting-Enriched Knowledge Graph for Skills-based Matching uses skill & occupation KG to reduce skill gaps and improve match quality. generating unified candidate skill graphs for career path recommendation also provides useful methodologies to normalize and connect skills.

III. WORKFLOW

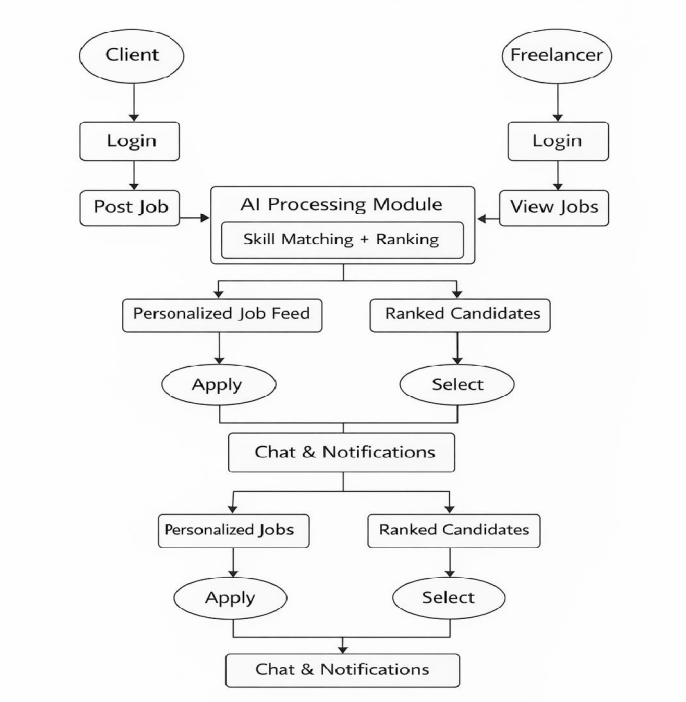

Fig. 3.1 System Architecture of Smart Hiring AI-Based Job Recommendation System

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, May 2026 Impact Factor: 8.2

1. User Registration and Login

The workflow begins with users (clients and freelancers) registering on the platform and logging in through secure authentication provided by Firebase. Role-based access control is applied to differentiate functionalities for clients and freelancers.

2. Profile Creation

Freelancers create and maintain detailed profiles by adding skills, experience, portfolio, and other relevant information. This data is stored in the cloud database and is used as input for AI-based analysis and matching.

3. Job Posting (Client Side)

Clients post job requirements by providing structured details such as job description, required skills, budget, and project scope. The job data is stored in the database and made available for further processing.

4. AI Processing Module

The core of the system is the AI module, which processes job and profile data using advanced techniques:

Skill Matching: NLP techniques extract and compare relevant skills from job descriptions and freelancer profiles.

Candidate Ranking: Machine learning models evaluate freelancers based on skill relevance, experience, ratings, and bid value.

Job Recommendation: The system generates job suggestions tailored to freelancer profiles using similarity analysis.

5. Personalized Job Feed

Based on AI analysis, freelancers receive a dynamic and personalized job feed that displays relevant job opportunities, reducing search effort and improving engagement.

6. Application Process

Freelancers apply for suitable jobs by submitting proposals, including bid amount and project details. These applications are stored and processed for further evaluation.

7. Real-Time Interaction

A real-time chat module enables seamless communication between clients and freelancers. This allows users to discuss project requirements, negotiate terms, and clarify expectations before final selection.

8. Notification System

The system generates real-time notifications to keep users informed about:

New job recommendations

Application status updates

Incoming messages

Project progress

This ensures timely updates and improves user responsiveness.

9. Selection and Completion

Clients review the AI-generated ranked list of candidates, interact if needed, and select the most suitable freelancer. The project is then initiated and completed through the platform, ensuring smooth workflow execution.

IV. METHODOLOGY

1. Flutter Frontend

The user interface is developed using Flutter, enabling cross-platform compatibility across Android, iOS, and web platforms. The frontend provides an intuitive and responsive environment where users can perform operations such as registration, login, job posting, profile management, and application tracking. The system also supports real-time dashboards, personalized job feeds, and interactive UI components, ensuring smooth navigation and enhanced user engagement. The integration of dynamic data rendering allows users to view updates instantly, including job recommendations, messages, and notifications.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

2. Flask Backend API

The backend is implemented using the Flask framework, which acts as an intermediary between the frontend, database, and AI modules. It handles API requests related to user authentication, job posting, proposal submission, and system processing. RESTful APIs using JSON format are used for communication between Flutter and Flask, ensuring modularity and scalability. The backend also manages business logic, validation, and secure data exchange, maintaining low latency and high system reliability.

3. Firebase Database

Firebase is used as a cloud-based backend service for real-time data storage, authentication, and synchronization. It enables seamless data exchange between clients and freelancers.

Firestore Database stores structured data such as user profiles, job details, proposals, and chat messages.

Firebase Authentication ensures secure login using email or third-party providers.

Real-time synchronization ensures that updates (messages, job status, notifications) are instantly reflected across the platform.

4. AI Service Modules (Python)

Artificial Intelligence is the core component that enables intelligent automation and decision-making. The AI modules are implemented using Python libraries such as scikit-learn and TensorFlow.

Job Recommendation System: Uses Natural Language Processing (NLP) to analyze job descriptions and freelancer profiles, generating personalized job feeds based on skill relevance and user behavior.

Skill Matching Module: Identifies similarities between required job skills and freelancer capabilities to improve matching accuracy.

Proposal Ranking System: Applies a Logistic Regression model to rank freelancer proposals based on factors such as skills, experience, ratings, and bid value.

5. Communication and Interaction Module (New)

To improve collaboration and usability, the system integrates real-time communication features:

Chat System: Enables direct messaging between clients and freelancers for discussing project requirements.

Notification System: Provides real-time updates for job applications, messages, and project status.

Activity Tracking: Keeps users informed about system events, improving responsiveness and engagement.

6. Communication Flow

The system follows a client-server architecture:

The Flutter frontend sends requests to the Flask backend.

The backend processes requests, interacts with Firebase for data storage/retrieval, and communicates with AI modules for predictions and recommendations.

The processed response is returned to the frontend.

This ensures secure, fast, and efficient two-way communication across the platform.

7. System Features (Updated)

Real-time synchronization of projects, messages, and notifications

AI-driven job recommendation and candidate ranking

Personalized job feed for freelancers

Secure authentication and cloud-based storage

Real-time chat and communication system

Secure transaction monitoring

Role-based access control (Client, Freelancer, Admin)

V. RESULT

The system successfully enables efficient interaction between clients and freelancers through a user-friendly interface. AI-based skill matching improves the accuracy of job recommendations and candidate selection.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

The proposal ranking mechanism helps clients quickly identify the most suitable freelancers.
Personalized job feeds reduce the time required for freelancers to search for relevant opportunities.
The real-time chat system enhances communication and collaboration between users.
Notification features ensure users stay updated about job status, messages, and activities.
Firebase integration provides real-time data synchronization, improving system responsiveness.
Secure authentication and data handling ensure user privacy and system reliability.
Overall, the system improves efficiency, transparency, and user experience in freelancing operations.

VI. CONCLUSION

The developed AI-driven freelancing platform provides an effective approach for improving interaction between clients and freelancers in a digital environment. By incorporating intelligent mechanisms such as automated skill matching, candidate evaluation, and personalized job recommendations, the system enhances both accuracy and efficiency in the hiring process. The inclusion of real-time communication and notification features further strengthens user engagement and collaboration. Overall, the system demonstrates the practical application of Artificial Intelligence in simplifying and improving freelancing operations.

REFERENCES

[1] Ricci, F., Rokach, L., & Shapira, B. (eds.) Recommender Systems Handbook. Springer, 2015. chapters on collaborative filtering, content-based and hybrid recommenders.
[2] Manning, C.D., Raghavan, P., & Schütze, H. Introduction to Information Retrieval. Cambridge Univ. Press (2008). see chapter “Term frequency and weighting” (TF-IDF, vector space).
[3] Jurafsky, D. & Martin, J.H. Speech and Language Processing (2nd/3rd ed.). NLP foundations useful for resume/job text processing.
[4] Lavi, D., et al. (2021). Fine-tuned Siamese Sentence-BERT for matching jobs and resumes. practical SBERT use for vacancy/resume embeddings.
[5] De Groot, M., et al. (2021). Job Posting-Enriched Knowledge Graph for Skills-based Matching. building skills & occupation KG.
[6] Widodo, R.I.H., et al. (2024). Job Recommendation System Combining Collaborative Filtering and Content-Based Filtering. ResearchGate. hybrid job recommender example.
[7] Rosenberger, J., et al. (2025). Career BERT: Matching Resumes to ESCO jobs in a shared space. ScienceDirect domain-specific BERT for career matching.
[8] Tian, G. (2025). Employment recommendation system based on item-based collaborative filtering. ACM / research article (example recent work on CF for jobs).
[9] Feuling the Gig Economy: A Case Study Evaluation of Upwork. Upwork case study / academic analyses (useful for platform design insights).
[10] Facebook AI Research. FAISS: A library for efficient similarity search (2017) - practical system for embedding search at scale.
[11] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
NAACL-HLT. Foundational paper for BERT, useful for resume parsing, job descriptions, and skill extraction.
[12] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP. Core reference for SBERT, semantic similarity between freelancer profiles and job postings.
[13] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. IEEE Computer. Classic and highly cited work on collaborative filtering, useful for freelancer – client interaction modeling.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, May 2026
Impact Factor: 8.2

[14] Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems. Helpful for skills ontologies, knowledge graphs, and linking skills, roles, and experience.

[15] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys. Broad survey covering deep learning recommenders, hybrid systems, and future directions.