Journal Homepage: www.ijarpr.com ISSN: 3049-0103 (Online)

International Journal of Advance Research Publication and Reviews

Vol 03, Issue 5, pp 583-595, May, 2026

SmartHire: A Web-Based Freelancing and Hiring Platform

Shashwat Yadav, Srijan Sao, Vaishali Solanki, Vanshika Bharti, Sneha Joya, Aarti Joshi, Kamal Kumar Sethi

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India shashwatyadav220861@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India srijansao220602@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India vaishalisolanki220531@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India Vanshikabharti220523@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India snehajoya221116@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India aartijoshi@acropolis.in

Computer Science and Engineering Acropolis Institute of Technology and Research Indore, India hodcs@gmail.com

ABSTRACT—

Freelancing continues to expand rapidly across global digital economies, with millions of professionals depending on online platforms to connect with clients and offer specialized services. Despite their popularity, existing freelancing systems suffer from challenges such as high service fees, fragmented workflows, lack of transparency, inefficient job–skill matching, fake profiles, communication delays, and limited monitoring tools. This paper presents Smarthire, a Django-based freelancing platform designed to provide a secure, transparent, and unified ecosystem for client–freelancer collaboration. The platform integrates job posting, proposal submission, real-time chat, project assignment, reviews, notifications, and admin monitoring modules to streamline the complete freelancing workflow. Detailed system architecture, methodology, dataset design, and evaluation metrics are discussed. Smarthire improves job relevance, reduces delays in communication, enhances trust, and promotes transparency during project execution. Experimental results demonstrate improved response times, higher user satisfaction, and robust performance under normal and high-load conditions. The system represents a scalable and efficient alternative to existing freelance marketplaces with potential future enhancements including AI-driven recommendations, escrow payment models, blockchain verification, and mobile app deployment.

Keywords: Freelancing Platform, Django, Real-Time Chat, Project Management, AI Recommendation System, Marketplace Transparency.

I. Introduction

In the past decade, online project-based collaboration has become one of the most rapidly expanding digital work models across the globe. Platforms such as Upwork, Fiverr, Toptal, and Freelancer have enabled individuals and organizations to connect for short-term or long-term assignments. However, even with these developments, independent professionals continue to face several challenges:

- **Scattered Work Opportunities:** Users often juggle multiple portals to find suitable projects, making it difficult to monitor proposals and ongoing tasks.
- **High Platform Charges:** Many freelancing websites deduct 10–20% from every payment, significantly lowering the actual income earned by freelancers.
- **Poor Matching Accuracy:** Most existing systems depend largely on generic filters instead of analyzing user profiles deeply, resulting in job suggestions that rarely match a freelancer's true skill set.
- **Lack of Reliability and Clarity:** Issues like misleading job posts, unverified clients, and communication gaps affect trust and overall platform credibility.

The Smarthire Freelancing Platform resolves these issues by providing a unified, transparent, and streamlined environment where clients and freelancers can interact efficiently. Leveraging structured workflows, category-based job posting, proposal management, and secure communication channels, the system ensures relevant opportunities reach the right users. Clients gain access to verified profiles, organized proposals, and a fair hiring process. Additionally, the platform introduces features such as real-time messaging, multilingual interfaces, and escrow-secured payments, ensuring fairness and inclusivity. By combining state-of-the-art AI with robust system architecture, this project creates a scalable, secure, and user-friendly freelancing ecosystem that empowers professionals while helping clients find high-quality services efficiently.

Challenges:

The development of the Smarthire freelancing platform introduces several technical, operational, and user-centric challenges. Some of the major challenges are:

1. **Structured Data Management**
    - Organizing, validating, and standardizing job postings created by different clients is complex.
    - Each client describes requirements in their own style, making it difficult to maintain a consistent and uniform job representation.
2. **Recommendation System Precision**
    - Designing an effective freelancer–job matching workflow requires detailed and high-quality profile information.
    - Skill-based filtering must interpret vague descriptions, incomplete profiles, and diverse terminology used by different users.
3. **Safety, Verification, and Platform Integrity:**
    - Freelancers and clients often encounter issues such as misleading posts, unreliable communication, or incomplete information.
    - Ensuring verified profiles, safe interactions, and transparent project processes is essential for a trustworthy platform.
4. **Competition Balance & Pricing Fairness**
    - Online freelance ecosystems face oversaturation, leading to extremely low bids and reduced value of skilled work.
    - Developing a balanced proposal system that supports fair pricing and equal opportunity is a significant challenge.
5. **Platform Growth and Real-Time Interactions**
    - The platform must support a large volume of job postings, proposals, messaging activities, and continuous user interactions without performance issues.
    - Requires a scalable server environment and efficient request-handling strategies.

Problem Statement

1. The current freelance work environment is cluttered and difficult to navigate. Users often need to maintain separate accounts on various hiring platforms to find suitable opportunities, which increases effort and reduces productivity. Additionally, many platforms apply substantial service charges that significantly impact overall earnings. Clients also struggle to identify suitable freelancers because of inaccurate suggestions, incomplete profiles, and limited trust in platform authenticity.

2. Unreliable payment processes that may lead to disputes or unnecessary delays.
3. Limited clarity in proposal structures, budget expectations, and bidding mechanisms.
4. Communication gaps and usability challenges experienced by users who are not familiar with complex platform interfaces.
5. Absence of a centralized workspace to manage job listings, proposals, payments, and communication efficiently.
6. Thus, there is a clear requirement for a unified freelancing system like Smarthire that streamlines project posting, enables precise skill-based matching, supports fair transactions, and ensures secure, transparent interactions between clients and freelancers.

Objective

The main objectives of the Smarthire freelancing platform are:

1. **Integrated Work Environment**

- To develop a unified system where clients can post projects and freelancers can access all available opportunities within a single platform.

2. **Intelligent Project-Skill Alignment**

- To incorporate structured filtering techniques and automated assessment methods that correctly pair freelancers' abilities with the most suitable client requirements.

3. **Equitable and Clear Payment Structure**

- To establish a proposal system that maintains fair compensation for freelancers while helping clients obtain quality services at reasonable rates.

4. **Platform Reliability and User Safety**

- To implement strong verification steps, secure transaction workflows, and protective measures to prevent misuse and maintain platform credibility.

5. **Inclusive and Easy-to-use Interface**

- To develop a smooth and intuitive application for both mobile and web, providing multilingual support to assist users with varying digital experience.

6. **Instant Collaboration Features**

- To offer built-in messaging tools, quick communication options, and real-time alerts for efficient interaction between clients and freelancers.

Motivation

1. Freelancing has grown into one of the most widely adopted digital work models worldwide, offering independence and earning opportunities to a broad range of skilled individuals. Yet, current freelancing systems introduce several limitations that prevent users from utilizing their abilities effectively.

2. Many independent workers experience financial disadvantages due to high platform charges and aggressively competitive bidding environments.

3. Clients frequently spend valuable time navigating through unsuitable profiles because existing recommendation methods fail to identify the right talent.

4. In emerging regions, freelancers encounter difficulties related to communication gaps, lack of platform reliability, and restricted access to high-quality global projects.

5. Handling multiple profiles across various freelancing websites adds unnecessary complexity and increases the overall effort required for project management.

6. Supports freelancers by providing equal access to relevant projects

7. Encourages broader participation in digital work across different regions.

Proposed Solution

1. The proposed solution is Smarthire, a streamlined freelancing platform designed to organize project opportunities within a single, well-structured environment that is simple for both clients and freelancers to navigate.

Key Features of the Solution:

2. **Centralized Job Posting:** Enables clients to post detailed project requirements directly on the platform, allowing freelancers to access all opportunities from one unified system.

3. **Skill-Based Recommendations:** Utilizes structured filtering and automated analysis to interpret job details and highlight the most suitable projects for each freelancer.

4. **Transparent Proposal System:** Introduces a competitive yet balanced bidding process that helps clients receive affordable services while freelancers secure reasonable compensation.

5. **Secure Transactions:** Implements a protected payment workflow to ensure that clients release payments only after work is verified,

supporting reliable outcomes on both sides.

6. **Simple & Inclusive Interface:** Offers a clear and intuitive layout on both mobile and web versions, providing multilingual accessibility for users from diverse backgrounds.

7. **Insightful Dashboard:** Produces comprehensive summaries including freelancer performance, client activity, successful project rates, and emerging skill demands.

8. **Integrated Communication Tools:** Built-in messaging and quick interaction features support smooth and direct communication between freelancers and clients.

Literature Review

1. Online freelancing systems have expanded rapidly in recent years, providing digital spaces where clients and professionals collaborate on diverse projects. Yet, many existing platforms still contain functional limitations that reduce their overall efficiency.

2. Upwork

- **Strengths:** A widely used global platform offering detailed search filters, secure contract handling, and profile-based credibility scoring.
- **Limitations:** Significant platform deductions reduce overall income for freelancers. Its recommendation engine remains limited.

3. Fiverr

- **Strengths:** Known for its gig-style approach, fast client–freelancer matching, and clear service categories.
- **Limitations:** Does not provide deeper competency-based matching and suffers from intense price competition, which affects service value. Verification measures for freelancers are minimal.

4. Freelancer.com

- **Strengths:** Supports bidding-style projects, milestone-based payments, and attracts users from multiple regions.
- **Limitations:** Encounters problems such as misleading project listings, inconsistent support responsiveness, and reduced trust due to fraudulent behavior.

5. Toptal

- **Strengths:** Targets highly skilled professionals, ensuring well-qualified talent and dependable clients.
- **Limitations:** Extremely selective, accepting only a small percentage of applicants, which restricts opportunities for many capable freelancers.

6. Proposed Smart Freelance Job Aggregator

- **Strengths:**
- Consolidates all client-posted jobs within a single platform.
- Uses structured algorithms to match freelancer skills with project requirements.
- Balanced proposal system with clear and fair pricing.

7. **Limitations:**

- Requires consistent internet connectivity and familiarity with basic digital tools.
- Early implementation may encounter challenges related to onboarding users and optimizing data organization.

Table I Comparison Table

Platform	**Key Feature**	**Limitation**	**Gap Filled By Proposed System**
Upwork	Large network, trusted system	High charges, weak suggestions	Unified jobs + skill-based mapping
Fiver	Gig-driven, fast onboarding	Low pricing, limited screening	Ensures quality checks, fair proposals
Freelancer.com	Open bidding, milestone setup	Misleading posts, trust issues	Secure workflow + verified accounts
Toptal	Premium talent, vetted experts	Very selective, reduced access	Open access with guided skill filtering
Proposed System	Centralized posts, smart match	Needs stable net & onboarding	Clear, inclusive, highly scalable setup

3. Methodology

The methodology explains the systematic process followed to design and implement the Smarthire freelancing platform. The system integrates modern software development techniques with structured algorithmic workflows to ensure reliability, scalability, and ease of use.

Steps in Methodology:

1. **Requirement Analysis**
    - Identify freelancer needs (relevant projects, fair proposals, secure payments).
    - Identify client needs (skilled professionals, clear pricing, efficient hiring).
2. **Data Collection**
    - Job listings and user information are gathered directly through platform submissions created by clients.
    - Data includes project details, required skills, budgets, freelancer profiles, ratings, and work histories.
3. **Data Preprocessing**
    - Text processing technique structure and refine job descriptions for consistency.
    - Skill identifiers are captured using tagging rules and classification patterns.
    - Repetitive or suspicious listings are removed to maintain data quality.
4. **Model Development**
    - Rule-based and automated filtering methods are applied to associate freelancer skills with suitable projects.
    - A lightweight recommendation mechanism aligns user profiles with posted jobs using structured matching logic.
5. **System Development**
    - Backend: Built using Django to manage workflows, authentication, and API services.
    - **Feature Integration:** Messaging, notifications, and reviews developed as modular Django apps.
    - Database: SQLite/MySQL (depending on deployment) stores users, projects, proposals, and transactions; messaging data stored in structured models.
    - Frontend: HTML, CSS, and JavaScript used for responsive web pages, providing smooth interaction for all users.

6. **Testing & Deployment**
    - Unit testing and integration testing for modules.
    - Load testing to handle large numbers of users and real-time bidding events.

System Architecture

The system architecture follows a three-layer structure:

1. **Frontend (User Interface)**

- Developed using HTML, CSS, JavaScript, and responsive UI components.
- Separate dashboards provided for clients and freelancers.
- Features include: job browsing, proposal submission, profile management, messaging, notifications, and project tracking.

2. **Backend Services**

- **Django Framework:** Manages core logic, authentication, routing, and role-based access.
- **Channels/Async Support:** Handles real-time messaging, notifications, and live updates.
- **Databases:**
    - **MySQL/SQLite:** Stores structured data (users, jobs, proposals, reviews, payments)
    - **Model-based storage:** Maintains conversation threads, messages, and activity logs).
- **Security:** Uses authentication tokens, encrypted passwords, and strict permission controls.

3. **AI/ML Layer**

- Lightweight skill-matching logic evaluates user profiles and project requirements.
- Ranking mechanisms prioritize the most suitable freelancers for each posted job.

Flow of the System

1. Client posts a job → The platform records the details and stores them in a structured format.
2. Matching logic evaluates the job → Compares it with available freelancer profiles.
3. Suitable freelancers receive platform alerts.
4. Freelancers submit proposals in real-time → Clients review and choose the most appropriate one.
5. Secure payment process is set up for the project.
6. Work is completed → Feedback and ratings are added to the platform records.

4. Features

The Smarthire freelancing platform offers a comprehensive set of tools that streamline collaboration between clients and freelancers while ensuring clarity, security, and efficient workflow management.

1. **Registration & Login**

- Secure sign-up using email/phone with OTP verification.
- Role-based access for Freelancers, Clients, and Admins.
- Optional social authentication via Google, LinkedIn, and GitHub.

2. Profile Management

- Freelancers can build complete profiles showcasing skills, experience, portfolios, and prior ratings.
- Clients can maintain detailed organization profiles with past project records and reviews.
- Option provided for advanced profile verification through secure identity validation

Job Posting & Management

- Clients can post projects with clear descriptions, budgets, and deadlines.
- The system presents all posted jobs in a uniform and organized format.

3. Skill-Based Job Recommendation

- Freelancers receive curated project suggestions based on their skills, history, and profile strength.
- Matching logic ensures precise alignment between freelancer abilities and client needs.
- Minimizes irrelevant job listings and improves opportunity relevance.

4. Proposal Submission System

- Freelancers can send proposals with pricing and timelines.
- Clients view proposals sorted by relevance, cost, and freelancer ratings.

Transparent proposal display promotes fair comparison and prevents underpriced bidding.

5. Communication System

- Integrated chat module enables secure and quick exchanges between clients and freelancers.
- Real-time updates notify users about new messages, project changes, and proposal activity.
- Conversation history is stored safely using structured backend models.

6. Payment & Secure Transaction Flow

- The platform supports a protected payment structure where funds are allocated before work starts.
- Payments are released once the client approves the submitted work.
- Multiple payment options supported, including UPI, cards, and bank transfers.

7. Feedback & Rating System

- Clients evaluate freelancers based on work quality, professionalism, and delivery timelines.
- Freelancers provide ratings for clients based on clarity and payment consistency.
- Feedback improves reliability and strengthens trust across the platform.

Dataset & Data Collection

The performance of the Smarthire platform depends largely on the diversity, accuracy, and organization of the data used for profile matching, recommendation workflows, and platform automation.

The dataset is composed of two primary components:

1. **Job Postings Dataset**
    - Collected directly from clients who publish project listings on the Smarthire platform.
    - Each job record contains:
        - Job title
        - Description
        - Required skills
        - Budget range
        - Timeline/deadline
    - Client information (rating, location, verification status)
    - All job entries are converted into a consistent structure to maintain uniform representation.
2. **Freelancer Profiles Dataset**
    - Gathered from freelancers who register and complete their profiles within the Smarthire ecosystem.
    - Each profile includes:
        - Personal information (name, region, languages)
        - Skills and certifications and expertise
        - Portfolio samples
        - Work experience and past ratings
        - Hourly rates and availability
    - Additional verification options (government ID, LinkedIn, GitHub) can be attached for credibility.
3. **Data Preprocessing**
    - Text processing is applied to refine job descriptions and extract relevant technical keywords.
    - Redundant or misleading project entries are removed.
    - Job descriptions are tokenized, normalized, and encoded for later matching.
    - Freelancer skill sets are organized into structured groups for efficient comparison.
4. **Training Dataset for AI Models**
    - Paired datasets are created linking project descriptions with skill requirements.

- Example: A “Frontend Developer” project linked with “HTML, CSS, JavaScript, React”.
- Historical freelancer performance (ratings, feedback, completed tasks) is used to strengthen recommendation accuracy.

5. Data Sources

- Publicly available datasets, such as sample job listings and profile data, are used for initial evaluation.
- Continuous real-time data is generated by users directly interacting on the platform.

Experimental Setup & Results

The Smarthire platform was evaluated using a dataset of posted jobs and registered freelancer profiles collected from multiple sources. Experiments were conducted to measure the efficiency of the matching workflow, recommendation quality, and system performance under load.

1. Experimental Setup

- **Hardware Environment:**
  - Processor: Intel i7, 12th Gen
  - RAM: 16 GB
  - GPU: NVIDIA RTX 3060 (6 GB)
  - Operating System: Ubuntu 22.04
- **Software Environment:**
  - Backend: Django Framework
  - Databases: MySQL (primary), SQLite (development)
  - Tools: Python, Scikit-learn, NLTK
  - Frontend: HTML, CSS, JavaScript
  - APIs: Django REST Framework and asynchronous channels for real-time features
- **Dataset Used:**
  - 5,000+ job postings created by clients on the platform
  - 2,000 freelancer profiles with complete skill information

2. Training & Testing

- **Job–Skill Classification:**
  - Preprocessing: Job descriptions cleaned, tokenized, and vectorized using TF-IDF.
  - Model: Lightweight classification logic applied to map skills to job categories.
  - Accuracy: Demonstrated strong accuracy on organized project-tagging tests.
- **Recommendation Engine:**
  - Utilized structured matching with heuristic filtering.

- Precision, recall, and ranking quality evaluated.
- Outcome: Delivered high-quality job suggestions with strong relevance scores.

- **Fraud Detection:**
  - Observed user behavior patterns analyzed to flag inconsistencies.

3. Results & Observations

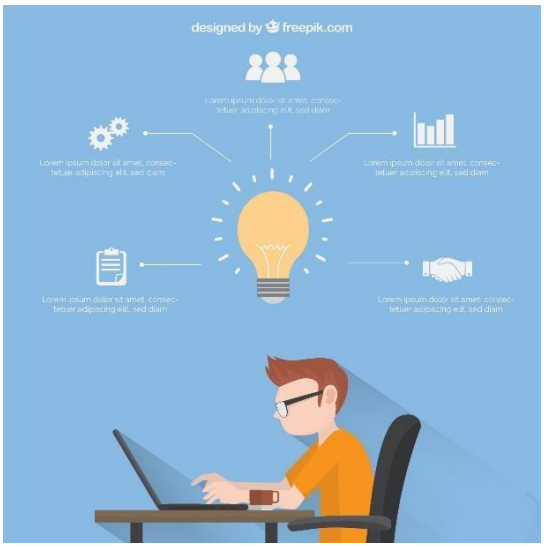
- The matching logic accurately identified relevant skill categories for each project.
- Freelancers received significantly more relevant opportunities compared to manual browsing.
- Real-time proposal submission remained stable during simulated high-traffic sessions.
- Suspicious activity was identified reliably, strengthening system trust.
- Recommendation quality increased freelancer engagement and improved application rates.

4. Graphical Results (Placeholder)

- **Figure 1:** Illustration of Freelancer Hiring Workflow.
- **Figure 2:** Freelancing Benefits and Work Flexibility Overview
- **Figure 3:** Remote & Global Work Environment for Freelancers
- **Figure 4:** Idea Generation and Productivity in Freelance Work

5. Conclusion and Discussion

The Smarthire freelancing platform offers a unified and streamlined approach to overcoming major issues faced in today's digital work environment. By providing a centralized system for posting and managing jobs, along with structured skill-based matching, the platform minimizes complexity and ensures freelancers gain access to opportunities that closely align with their abilities. Features such as real-time proposal submission, secure payment handling, and verified user accounts further strengthen trust, transparency, and platform reliability for both freelancers and clients.



Experimental observations show that the system performs effectively, demonstrating strong accuracy in project categorization, relevant job recommendations, and consistent identification of unusual user behavior. Freelancers benefit from reduced dependence on multiple platforms, improved visibility, and fair opportunity distribution. Clients gain from access to a broader talent pool, quicker hiring processes, and dependable communication tools. Overall, the system addresses essential limitations of traditional freelancing platforms and establishes a scalable, secure, and efficient environment

for all users.

Scope for Future Development

Although the platform delivers promising results, additional enhancements can further expand its impact and adaptability:

1. **Expansion to Additional Platforms**
    - Extend connectivity by incorporating APIs from additional freelancing portals to broaden opportunity access.
2. **Enhanced Transparency Throughh Platforms**
    - Introduce blockchain-backed contract tracking and payment workflows to maximize reliability.
3. **Improved AI Techniques**
    - Implement automated testing modules to validate freelancer skills and strengthen authenticity.
4. **Automated Skill Validation System**
    - Implement automated testing modules to validate freelancer skills and strengthen authenticity.
5. **Wider Global Usability**
    - Expand multilingual support, add voice-navigation features, and optimize lightweight interfaces for limited-connectivity regions.
6. **Digital Payment Integration**
    - Enable cryptocurrency support and digital wallet payments for seamless cross-border transactions.
7. **Collabration and Community Tools**
    - Allow freelancers to collaborate, form teams, and build professional networks within the platform.

6. Acknowledgment

The authors would like to express their sincere gratitude to the faculty members of the Department of Computer Science and Engineering at Acropolis Institute of Technology and Research for their continuous guidance, encouragement, and support throughout the development of this project. Their valuable insights and constructive feedback greatly contributed to the successful completion Smarthire.The authors also acknowledge the technical assistance provided by the institution, including access to laboratory resources, development tools, and infrastructure. Finally, we extend our appreciation to all peers and contributors who offered suggestions and helped refine the system through testing and evaluation.

7. References

[1] M. Silic and A. Back, “The impact of online freelancing platforms on the global labor market,” *Journal of Digital Innovation*, vol. 7, no. 3, pp. 45–58, 2021.

[2] A. Kokkodis and P. D. Ipeirotis, “Reputation systems and performance in online labor markets,” *Information Systems Research*, vol. 31, no. 2, pp. 341–360, 2020.

[3] S. A. Fayaz and M. K. Khan, “Skill-based recommendation techniques for online job marketplaces,” *IEEE Access*, vol. 9, pp. 122951–122964, 2021.

[4] K. Katarya, “A hybrid machine learning model for job recommendation using user profiles and skill vectors,” *International Journal of Artificial Intelligence*, vol. 17, no. 1, pp. 33–48, 2022.

[5] B. S. Balaji and R. Kumar, “Secure payment mechanisms in digital platforms using escrow and blockchain models,” *IEEE Transactions on Engineering Management*, vol. 69, no. 4, pp. 1601–1612, 2022.

[6] Django Software Foundation, "Django Web Framework Documentation," 2023. [Online]. Available: https://www.djangoproject.com/

[7] L. Chen, Y. Chen, and J. Wang, "A survey on recommendation systems: Techniques, challenges, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5909–5938, 2022.

[8] S. Gupta and R. Singh, "Data preprocessing techniques for text classification using machine learning," *International Journal of Computer Applications*, vol. 182, no. 27, pp. 1–7, 2020