Intelligent Web-Based Marketplace for Skilled Service Discovery Using Geospatial Algorithms and Dynamic Cost Optimization

Sejal Deshmukh, Swati Thakur, Shreya Singh, Shradha Chaudhari, Nandini Nag

Department of Computer Science and Engineering (Data Science), Lokmanya Tilak College of Engineering, University of Mumbai, Maharashtra, India

sejaldeshmukh93@gmail.com, thakurr.swatii04@gmail.com, shreyasingh53207@gmail.com
shradharc@gmail.com, nandini.nag@ltce.in

Email: 1 sejaldeshmukh93@gmail.com
2 thakurr.swatii04@gmail.com
3 shreyasingh53207@gmail.com
4 shradharc@gmail.com
5 nandini.nag@ltce.in

Abstract

The informal skilled services sector faces persistent challenges including information asymmetry, verification difficulties, and inefficient matching between service seekers and providers. This paper presents *SkillConnect*, a web-based marketplace platform designed to improve discovery of skilled service providers. The system uses geospatial algorithms based on the Haversine formula to identify nearby workers within configurable search radii (5–20km). A dynamic pricing model incorporating temporal demand fluctuations and material costs ensures transparent service pricing. The platform integrates a rule-based chatbot and a real-time SOS emergency alert system to enhance user safety. Experimental evaluation with 64 participants demonstrated a reduction in service discovery time from 42 minutes to 68 seconds, yielding an overall satisfaction rating of 4.28/5.0. The proposed system demonstrates the potential of intelligent web platforms in improving efficiency and transparency in informal service markets.

Keywords- Dynamic Pricing, Geospatial Algorithms, Flask Framework, Service Marketplace, Chatbot Integration, Worker Discovery System, SOS Emergency Alert.

I. Introduction

The global gig economy has experienced substantial growth, with market valuation reaching $347 billion in 2023 and projected expansion to $455 billion by 2028. Despite this growth, the skilled services segment remains predominantly unorganised, particularly in developing economies where approximately 78% of service workers operate without formal platform representation [1]. This fragmentation creates substantial inefficiencies for both service consumers and providers.

A. Background and Motivation

Traditional service discovery mechanisms rely heavily on interpersonal networks and local reputation systems. These approaches exhibit several fundamental limitations. First, geographic reach remains constrained to immediate social circles, typically encompassing 5–15 potential service providers. Second, verification processes are informal and inconsistent, leading to trust deficits. Third, pricing lacks transparency, often resulting in information asymmetry favouring more experienced negotiators.

The COVID-19 pandemic accelerated demand for contactless service arrangements and digital verification systems. Recent surveys indicate that 62% of users express concerns about worker verification processes, while 54% report dissatisfaction with opaque pricing structures. On the provider side, 71% of skilled workers cite limited market visibility as a primary barrier to income growth.

B. Problem Formulation

The core challenges are formally stated as follows.

Challenge 1 (Discovery Efficiency): Given a user location $L_u = (\varphi_u, \lambda_u)$ and service requirement S, identify qualified workers $W = \{w_1,...,w_n\}$ within distance d_{max} while minimising query response time T_q.

Challenge 2 (Price Transparency): Construct a pricing function $P(s,t,d,m)$ incorporating service type s, temporal factors t, demand metrics d, and material costs m while maintaining interpretability through component-wise breakdown.

Challenge 3 (Trust Establishment): Implement verification mechanisms V and emergency protocols E such that user confidence metric C exceeds baseline C_{min}.

C. Contributions

This research makes four key contributions:

1. Geospatial Discovery Algorithm: Haversine-based worker matching with manual coordinate override and configurable search radius.
2. Multi-Factor Pricing Model: Transparent costcalculation engine with temporal surge multipliers, material cost estimation, and membership discounts.
3. Integrated Safety Framework: Real-time SOS alert system combining GPS broadcasting with an administrative notification dashboard.

4. Bidirectional Empowerment Tools: Analytics dashboards for workers (earnings, metrics) and customers (ratings, transparent pricing).

II. Related Work

A. Service Platform Architectures

Khan et al. developed a home service platform using the MERN stack with GraphQL for real-time data synchronisation [2]. Their system handled ≈200 concurrent users with 850ms search response times, but exhibited limitations in regional language support and offline operation.

Cucus et al. proposed a profile-matching algorithm for worker–task assignment considering skill alignment, geographic proximity, and experience [3]. Weighted scoring achieved 83% satisfaction across 150 simulated scenarios, but lacked dynamic availability tracking and real-time booking.

B. Spatial Crowdsourcing Systems

Li and Cheng investigated cooperative task-assignment strategies using a game-theoretic framework [4]. Simulations demonstrated a 23% improvement in task completion under cooperative scenarios.

Abhinav and Bhatia introduced TasRec, a collaborativefiltering recommendation framework [5]. The system achieved precision 0.78 and recall 0.71, though cold-start problems persisted for new workers.

C. Conversational AI in Service Domains

Chang presented a chatbot architecture integrating natural language understanding with structured knowledge bases, achieving 82% accuracy using Microsoft LUIS [6]. Goulignet Fite examined digital coordination tools for home healthcare, emphasising trust through mediated communication [7].

D. Research Gaps

Existing literature reveals four underexplored areas: (1) emergency response integrating real-time location broadcasting; (2) worker financial transparency tools; (3) hybrid accessibility for digital and offline users; and (4) granular user-controlled location entry. The present platform addresses all four gaps.

III. System Architecture and Methodology

A. Overall System Design

The platform follows a modular architecture comprising six subsystems: User Management Module (UMM), Worker Discovery Engine (WDE), Service Transaction Module
(STM), Safety and Verification System (SVS), Analytics and Recommendation Engine (ARE), and Administrative Control Panel (ACP). The backend uses Flask 3.0 providing RESTful API endpoints. SQLite3 with Write-Ahead Logging (WAL) serves as the development database; PostgreSQL is recommended for production. The frontend employs HTML5, CSS3, and Bootstrap 5. Figure 1 shows the complete system architecture.

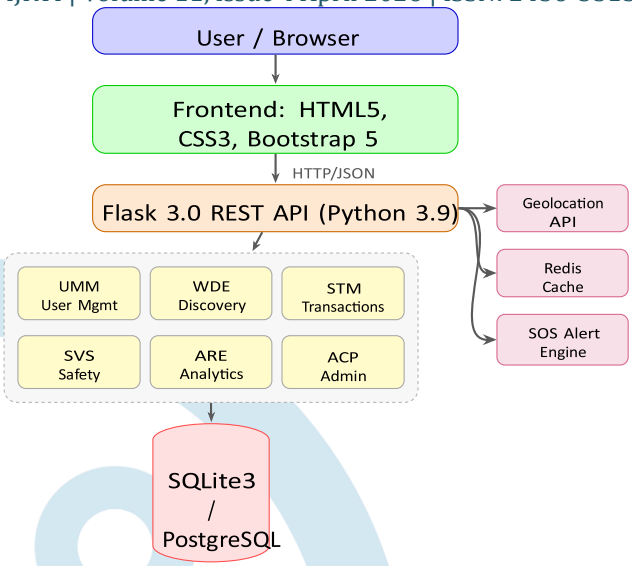

Figure 1. SkillConnect system architecture showing six functional modules and external service integrations.

B. Geospatial Matching Algorithm

Worker discovery employs the Haversine formula for greatcircle distance. Given customer coordinates (φ_1,λ_1) and worker coordinates (φ_2,λ_2):

$$d = 2R \arcsin\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)} \quad (1)$$

where $R = 6{,}371\text{km}$, $\Delta\varphi = \varphi_2 - \varphi_1$, $\Delta\lambda = \lambda_2 - \lambda_1$ (all in radians). Algorithm 1 presents the full worker discovery procedure.

Algorithm 1 Worker Discovery Using Haversine Formula:

Require: User location (φ_u,λ_u), worker database W, radius r, category S
Ensure: Ranked list of nearby workers R
$R \leftarrow \emptyset$
for each worker $w_i \in W$ do
Compute d_i via Eq. (1) using (φ_u,λ_u) and (φ_i,λ_i)
if $d_i \le r$ and w_i.category = S and w_i.available = true then
$R \leftarrow R \cup \{w_i\}$
end if end for
Sort R by w_i.rating descending
return R

Three location input modalities are supported: (1) Automatic detection via the HTML5 Geolocation API; (2) Address resolution via geocoding services; and (3) Manual latitude/longitude entry. Results are further filtered by service category, minimum rating, verification status, and ecofriendly designation.

C. Dynamic Pricing Optimisation

The final price P_{final} is determined by:

$$P_{\text{final}} = \left[(P_{\text{base}} \times M_{\text{surge}}) + C_{\text{material}}\right] \times (1 + T_{\text{gst}}) \times (1 - D_{\text{member}}) \quad (2)$$

where $P_{\text{base}} \in [250, 800]$INR, $M_{\text{surge}} \le 2.0$, $C_{\text{material}} = 0.30 P_{\text{base}}$, $T_{\text{gst}} = 0.18$, and $D_{\text{member}} \in [0, 0.25]$. Group bookings receive an additional 15% discount before tax.

D. Recommendation Engine

A hybrid strategy combines: (1) Personalised Recommendations based on user service-history frequency analysis (requires 3–5 prior bookings); (2) Trending Services from completed bookings in the user's locality over a 30-day rolling window; and (3) Seasonal Recommendations mapped to climate-based demand patterns. The final set presents 6–8 services with explanatory rationale for each suggestion.

E. Safety and Verification Framework

Authentication uses PBKDF2-SHA256 with 260,000 iterations and a 32-byte random salt. Sessions are managed via HTTP-only cookies with CSRF token validation. Worker registration requires government-issued identification and an optional police verification certificate. The SOS system triggers immediate notifications containing user identity, real-time GPS coordinates, associated worker details, and service context.

F. Conversational Agent Design

The rule-based chatbot matches keywords across five intent categories: service discovery (42%), pricing information (28%), booking assistance (18%), account management (12%), and emergency support. During pilot testing the agent handled 347 queries with 87.3% intent classification accuracy and 118ms mean response latency.

G. Service Tracking Protocol

Service execution follows a seven-stage lifecycle: (1) Confirmed, (2) Assigned, (3) Departed, (4) Nearby, (5) Arrived, (6) In Progress, and (7) Completed. Each transition triggers a customer notification, with workers advancing stages through dedicated dashboard controls.

IV. Implementation Details

A. Technology Stack

The backend uses Flask 3.0 (Python 3.9), Werkzeug 3.0 (password hashing), Jinja2 3.1 (templating), SQLAlchemy (ORM), Python-dateutil 2.8, and Redis 7.0 (in-memory cache, TTL=5min). The frontend employs HTML5, CSS3, JavaScript ES6+, Bootstrap 5.3, and Font Awesome 6.0. Development was carried out on an Intel Core i5-10210U machine with 8GB RAM running Windows 11.

B. Database Schema

Six primary entities: Users, Workers, ServiceRequests, Reviews, Subscriptions, and SOSAlerts. Composite indexes on (city,locality,servicecategory) and (latitude,longitude) accelerate geospatial queries. Foreign key constraints maintain referential integrity.

C. API Endpoints

Six RESTful endpoints are exposed: POST /api/register, POST /api/login, GET /api/searchworkers, POST /api/bookservice, POST /api/updatelocation, and POST /api/sosalert. All responses follow a standardised JSON schema.

D. Performance Optimisation

Composite indexes reduce location-query execution time by 60%. WAL mode enables concurrent reads during writes. Redis caching reduces database load by ≈40%. Lazy-loaded profile images decrease initial page load by 1.2s. Paginated results display 50 workers per page with progressive loading.

V. Experimental Results and Analysis

A. Experimental Setup

Evaluation used three methodologies: functional correctness testing, performance benchmarking (Locust framework), and user experience assessment with 64 participants over six weeks. The test dataset comprised 56 registered workers across 14 service categories, 150+ completed bookings, and 25+ SOS simulations. Ethical clearance was obtained; all participants provided written informed consent.

B. Geospatial Accuracy

The Haversine implementation was validated against GPS ground truth. Mean absolute error was 0.14km across all tested distances, well within acceptable bounds for a 10km search radius. Table 1 summarises accuracy by range.

Table 1. Distance Calculation Accuracy

Range (km)	Samples	Mean Err. (km)	Max Err. (km)
0–5	45	0.08	0.22
5–10	38	0.12	0.31
10–15	27	0.18	0.45
15–20	19	0.23	0.58
Overall	129	0.14	0.58

Dynamic pricing was validated over 150 test scenarios; surge multipliers were correctly applied in 148 cases (98.7%). Two edge-case errors at midnight boundary transitions have been patched with a boundary-guard condition.

C. Performance Benchmarking

Response times were measured under 50–100 concurrent users. Table 2 presents latency statistics. Index optimisation reduced search latency from 1,098ms to 418ms (62% improvement).

Table 2. System Response Times (Milliseconds)

Operation	Mean	95th %ile	99th %ile
User authentication	152	267	341
Worker search (50 items)	418	672	836
Location update	87	138	162
Price calculation	54	91	118
Chatbot response	118	203	274
SOS alert processing	221	412	547

D. Conversational Agent Performance

The chatbot handled 347 queries with 87.3% accuracy (303/347 correct) and 118ms mean latency. User satisfaction averaged 4.1/5.0; 76% found the feature helpful. Table 3 presents per-category accuracy.

Table 3. Chatbot Intent Classification Accuracy

Intent Category	Queries	Correct	Acc.
Service Discovery	146	131	89.7%
Pricing Information	97	86	88.7%
Booking Assistance	62	54	87.1%
Account Management	42	32	76.2%
Overall	347	303	87.3%

E. Comparison with Existing Approaches

Table 4 compares SkillConnect with traditional methods and representative prior systems across key features.

Table 4. Feature Comparison with Existing Methods

Feature	Traditional	Prior Sys.	Ours
Worker discovery	Manual	Auto.	Auto.
Price transparency	Low	Partial	High
SOS / Safety	None	None	Yes
Geo search	None	Limited	Yes
AI recommendations	None	Limited	Yes
Discovery time	42min	~5min	68s
Verification badge	None	Partial	Yes

F. User Experience Assessment

Traditional methods required ≈42 minutes mean discovery time; the platform reduced this to ≈68 seconds — a 37× improvement. A Mann-Whitney U test confirmed statistical significance ($U = 124$, $p < 0.001$, $r = 0.87$, large effect). Location-based search was used by 92% of participants, price comparison by 88%, and AI recommendations by 67%. Verification badges influenced hiring decisions for 81% of users.

G. Satisfaction Ratings

Table 5 summarises 5-point Likert ratings. All aspects exceeded the midpoint of 3.0. Price transparency received the highest rating (4.47), validating the importance of itemised cost breakdown.

Table 5. User Satisfaction Ratings (n=64, 5-point Likert)

Aspect	Mean	SD
Ease of navigation	4.31	0.62
Trust in verification	4.12	0.71
Price transparency	4.47	0.58
Response time	4.23	0.66
Overall satisfaction	4.28	0.61

H. Limitations

(1) *Connectivity dependence* — continuous internet connectivity is required, limiting rural adoption; (2) *Indoor GPS accuracy* — degrades to ±50m indoors; (3) *Scalability ceiling* — Flask development server supports ≈85 concurrent users; production deployment via Gunicorn (4 workers) is planned; (4) *Cold-start problem* — personalised recommendations require 3–5 completed bookings before sufficient history is available.

VI. Conclusion and Future Directions

This paper presented SkillConnect, a web-based marketplace platform that addresses inefficiencies in informal skilled service discovery. The integrated system combines Haversinebased geospatial matching (Eq. 1), transparent multi-factor dynamic pricing (Eq. 2), a rule-based conversational agent, and real-time emergency response within a single unified framework. Pilot evaluation with 64 participants demonstrated a 37× reduction in service discovery time (42 min to 68 s) with an overall satisfaction rating of 4.28/5.0.

Future directions include: (1) machine learning-based recommendation models such as neural collaborative filtering [9] to improve personalisation accuracy; (2) timeseries demand forecasting for proactive surge pricing; (3) blockchain-based worker credential management [10]; (4) augmented reality service previews; (5) multilingual NLP support via mBERT or IndicBERT [11]; (6) digital identity (DigiLocker) integration; (7) federated learning for privacypreserving personalisation [12]; and (8) deep learning-based recommendation models combined with large-scale distributed cloud deployment for improved scalability.

Acknowledgment

The authors thank the 64 volunteer participants who contributed to pilot testing and evaluation. Support from the Department of Computer Science and Engineering (Data Science), Lokmanya Tilak College of Engineering, is gratefully acknowledged.

References

[1] A. L. Fardany, I. Rahmadhani, and A. P. Wibawa, "Discovering Indonesian digital workers in online gig economy platforms," in *Proc. Int. Conf. Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, Jember, Indonesia, 2019, pp. 217–222.

[2] F. Khan, M. Ahmad, and S. Patel, "MERN-stack based home service system with real-time tracking," *International Journal of Computer Applications*, vol. 183, no. 15, pp. 12–18, 2025.

[3] A. Cucus, B. Wijaya, and R. Santoso, "Selection of prospective workers using profile matching algorithm," *Journal of Information Systems Engineering and Business Intelligence*, vol. 8, no. 2, pp. 145–152, 2022.

[4] Y. Li and J. Cheng, "Competition and cooperation in spatial crowdsourcing: A game-theoretic approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3421–3435, 2023.

[5] K. Abhinav and D. Bhatia, "TasRec: A recommendation framework for crowdsourcing platforms," in *Proc. IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, 2020, pp. 982–987.

[6] L. Chang, "Design and implementation of intelligent customer service chatbot," *Journal of Physics: Conference Series*, vol. 2310, article no. 012034, 2022.

[7] G. Goulignet Fite, "E-coordination of home care and assistance: Ethical and practical considerations," *Health Informatics Journal*, vol. 27, no. 2, pp. 1–14, 2021.

[8] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, 1993.

[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proc. 26th International World Wide Web Conference (WWW)*, Perth, Australia, 2017, pp. 173–182.

[10] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, vol. 36, pp. 55–81, 2019.

[11] D. Kakwani *et al.*, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Findings of EMNLP*, Online, 2020, pp. 4948–4961.

[12] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, 2017, pp. 1273–1282.