

Smart Vendor Hub: An Evidence-Aware Adaptive Ranking and Multi-Factor Trust Framework for Hyperlocal Municipal Service Discovery

Chetan Khandare, Neha Kumbhalwad, Shreya Katkar, Sakshi Somwanshi, Ashwini Garkhedkar

Department of Master of Computer Application, P.E.S.'s Modern College of Engineering (An Autonomous Institute Affiliated to Savitribai Phule Pune University), Pune, Maharashtra, India

Published October 2026

Abstract

Evidence-aware ranking reduces rating manipulation and commercial bias in local service directories by conditioning vendor order on verified reviews, physical distance, and identity trust stored in a shared database index. Deploying hyperlocal discovery as a multi-role civic product requires durable relational storage, role and booking isolation, and a ranking path that does not force a separate paid-bidding or machine learning service beside the application. This paper presents a PostgreSQL-centred architecture for that setting. All vendors, service categories, and user roles share one PostgreSQL database with GIN full-text and GiST spatial indexes. Isolation and rating integrity are obtained by tagging every booking and review with customer and vendor identifiers and applying a completed-booking check before any rating updates the vendor score. Sparse ratings are stabilised by Dynamic Bayesian Shrinkage, distance is modelled by continuous Haversine decay, and identity trust is captured by a Multi-Factor Verification Trust Score (MFVTS). The design is realised in Smart Vendor Hub, an Angular 19 and FastAPI platform that connects citizens with local tradespeople, answers searches with Explainable Information Retrieval (XIR) score breakdowns, and exposes the same ranking function to a citizen discovery portal and to a live Researcher simulation lab. We describe the index schema, the ranking and review-gate algorithms, the isolation invariant, and an empirical evaluation on a deployed instance. The result is a practical blueprint for using PostgreSQL as a shared ranking store in hyperlocal service discovery without pay-to-play ad auctions or an external neural ranker.

Keywords: Bayesian rating shrinkage; explainable ranking; Haversine distance; hyperlocal service discovery; multi-factor trust; PostgreSQL

1 Introduction

Urban households rely daily on local tradespeople—plumbers, electricians, carpenters, appliance mechanics, cleaners, and painters—for essential home maintenance [1, 2]. Conventional commercial directories and gig platforms rank these providers using paid ad auctions or raw arithmetic star averages [3]. That approach is a poor fit for fair municipal service discovery. New honest technicians have no reviews, raw averages let a vendor with one or two fake five-star ratings outrank a veteran with a hundred completed jobs, and hard ward boundaries exclude nearby providers just across a street line [4]. Evidence-aware ranking couples full-text retrieval with statistical rating calibration, continuous geodesic distance decay, and verified identity signals [5, 6].

The engineering problem is where those ranking signals live and how they stay trustworthy. Classroom prototypes often keep vendor lists in memory or rank them by a single SQL ORDER BY average_rating clause. A civic product that serves citizens, vendors, administrators, and researchers

cannot do so: records must survive deploys, reviews must be restricted to completed bookings, and the service should scale without attaching a heavy machine learning worker to every web request. Relational database management systems with full-text GIN and spatial GiST indexes provide fast candidate retrieval and transactional predicates in one engine [7, 8]. PostgreSQL is an open-source, production-grade member of that class. It stores relational tables, supports server-side full-text ranking, and enforces foreign-key and status predicates so that the same database both filters candidates and guards review integrity.

1.1 Objectives and contributions

The objectives are (i) to isolate four roles and enforce completed-booking review rules on one PostgreSQL database, (ii) to rank vendors using a single transparent formula combining relevance, Bayesian shrinkage, Haversine distance, and trust, and (iii) to expose the ranking engine through both a public citizen discovery portal and an interactive Researcher viva laboratory. The contributions of this paper are as follows.

- A role and booking model in which booking status COMPLETED and ownership fields `customer_id` and `vendor_id` act as the integrity key, and every review or inquiry mutation checks that predicate before updating vendor scores.
- A discovery and ranking pipeline that combines PostgreSQL GIN full-text relevance, Dynamic Bayesian Shrinkage (prior $m = 4.20$, $C = 5.0$), continuous Haversine decay, Multi-Factor Verification Trust Score (MFVTS), and cold-start exploration, eliminating a separate neural reranking sidcar.
- A dual discovery facade—public read-only citizen search with Explainable Information Retrieval (XIR) modals and JWT-protected role portals for Customer, Vendor, Admin, and Researcher—sharing a single Python ranking function.
- A deployed case study, Smart Vendor Hub, which instantiates the architecture as an Angular 19 and FastAPI service with PostgreSQL for storage and Leaflet/OpenStreetMap for spatial pins, demonstrating that the evidence-aware design is operational rather than merely schematic.

The remainder of the paper surveys the literature, states the problem, describes the research methodology, states the hypotheses, and reports results and discussion, followed by the conclusion, future scope, and limitations.

2 Literature survey

2.1 Hyperlocal service discovery

Kumar et al. and Amasuomo and Baird examine civic service management in Indian cities and note that informal service markets suffer from severe information asymmetry [1, 2]. Commercial directories such as Justdial and Sulekha monetise visibility through paid lead bidding, while managed gig platforms charge 20–35% commission per job [3, 4]. The system in this paper follows a zero-commission civic-commons pattern: citizens search free of charge, vendors pay no lead fees, and ranking accepts no advertising parameter.

2.2 Bayesian rating shrinkage and sparse feedback

De Campos et al. and subsequent recommender-system studies show that arithmetic means are volatile when sample size N is small [5]. Empirical Bayes shrinkage pulls small- N averages toward a

community prior m with weight C , so a single five-star rating cannot dominate an established provider [6]. Using raw star averages without shrinkage exposes a directory to Sybil review inflation and cold-start distortion. Smart Vendor Hub embeds Bayesian shrinkage directly inside both query-time ranking and post-review persistence.

2.3 Spatial indexing and geodesic distance

Imran et al. demonstrate GIS-based spatial analysis for municipal planning [9]. Polygon or string-matching locality filters create boundary cliffs where a technician 300 m away across a ward border is hidden. Continuous distance decay using the Haversine great-circle formula avoids that cliff [10]. PostgreSQL with GIN and GiST indexes supports fast text and coordinate filtering inside the relational engine [7, 8].

2.4 Multi-role software and access control

Krebs, Momm, and Kounev discuss isolation in shared-schema multi-tenant applications [11]. When customers, vendors, admins, and researchers share one database, omitting an ownership or role predicate leaks private contact data or allows unauthorized score changes. Therefore the role and booking-status checks in this work are enforced in FastAPI route dependencies and inside the review gate rather than left to frontend buttons [12].

2.5 Research gap

Tutorial directory apps typically sort by raw average rating or simple SQL keyword LIKE clauses and omit booking verification. Papers on information-retrieval fairness rarely ship a working four-role civic web platform with zero-commission direct quotes. Commercial platforms hide their ranking weights behind proprietary APIs. The gap addressed here is the combination: a documented, implemented, and tested PostgreSQL-backed hyperlocal service hub that fuses Bayesian shrinkage, Haversine decay, multi-factor trust, cold-start exploration, and XIR explainability in one coherent service.

3 Problem statement

Consider urban citizens who search for a nearby plumber or electrician and expect the top results to reflect genuine skill, physical proximity, and verified identity rather than paid sponsorship or fake reviews. A naive directory violates this requirement: arithmetic averages reward two fake five-star ratings over a hundred real jobs, ward-name filters cut off nearby technicians, and unverified review forms allow anyone to rate a vendor without hiring them. A second problem is opacity: citizens and examiners cannot see how relevance, distance, rating, and trust combine into the final order. The question studied here is whether a shared PostgreSQL index, a composite Bayesian–Haversine–trust formula, a completed-booking review gate, and per-hit XIR attribution are sufficient to deliver fair, explainable, multi-role service discovery.

4 Research methodology

4.1 System model

Citizens, vendors, administrators, and researchers sign in through OAuth2/JWT authentication [12] and interact with role-specific portals in Angular 19 [13]. Public visitors can search vendors and

inspect ranking explanations without a session. Both public search and the Researcher simulation lab invoke one ranking procedure in FastAPI [14] that queries PostgreSQL [7]. The ranking engine and booking state machine are therefore on the critical path of every discovery and review action: if the booking gate is bypassed, trust fails; if the composite formula is bypassed, ranking fairness collapses.

Let V be the set of vendors and B the set of service bookings. Each booking b_j carries metadata including `customer_id`, `vendor_id`, `service_id`, and `status` in $\{\text{PENDING, ACCEPTED, COMPLETED, REJECTED}\}$. Define the verified completed slice for customer u and vendor v as $B_{u,v} = \{b_j \in B \mid b_j.\text{customer_id} = u \wedge b_j.\text{vendor_id} = v \wedge b_j.\text{status} = \text{COMPLETED}\}$. The integrity invariant required of the implementation is that a review submitted by customer u for vendor v is accepted if and only if $B_{u,v}$ is non-empty.

4.2 Shared index and trust design

A single PostgreSQL database is provisioned with a GIN full-text index on vendor service descriptions and a GiST index on geographic coordinates [7, 8]. All four roles share one relational schema. Physical separation into separate databases per role or per civic ward was rejected: it complicates cross-ward search and duplicates schema migrations. Logical isolation with mandatory JWT ownership checks and the completed-booking review gate concentrates the correctness argument in one code path.

4.3 Bayesian shrinkage, geodesic decay, and composite score

Let $R_v = \{r_1, r_2, \dots, r_N\}$ denote the verified star ratings of vendor v , with prior mean $m = 4.20$ and prior weight $C = 5.0$. The stabilised Bayesian rating is

$$S_{\text{bayes}}(v) = \frac{C \cdot m + \sum_{i=1}^N r_i}{C + N} \quad (1)$$

For a citizen query at coordinates $(\text{lat1}, \text{lon1})$ and vendor coordinates $(\text{lat2}, \text{lon2})$ with great-circle Haversine distance d_{km} and scale $\lambda = 5.0$ km, the spatial decay score is

$$S_{\text{geo}}(v, q) = \exp\left(-\frac{d_{\text{km}}}{5.0}\right) \quad (2)$$

Given full-text relevance S_{rel} in $[0, 1]$, Multi-Factor Verification Trust Score MFVTS in $[0, 100]$, cold-start exploration bonus $B_{\text{cold}} \leq 0.15$, and anti-monopoly damping factor P_{mono} in $[0, 1]$, the composite score is

$$\text{Score}(v, q) = \left[0.35S_{\text{rel}} + 0.25\left(\frac{S_{\text{bayes}}}{5}\right) + 0.25S_{\text{geo}} + 0.15\left(\frac{\text{MFVTS}}{100}\right) + B_{\text{cold}} \right] \cdot P_{\text{mono}} \quad (3)$$

Given candidate pool V_q retrieved by GIN text match and integer k , the ranked result list returned to the caller is

$$R(q) = \text{TopK}_{v \in V_q} \text{Score}(v, q) \text{ with XIR}(v, q) \quad (4)$$

Equation (3) encodes the zero-commission fairness invariant: no monetary or ad-bidding variable appears in the score. Equation (4) attaches the XIR attribution vector $\text{XIR}(v, q)$ containing the weighted contribution of each term so that the frontend can render the “Why this rank?” breakdown without a second round-trip.

4.4 Discovery and review recalibration algorithms

When a citizen searches by keyword and locality, the backend retrieves candidate vendors via GIN full-text matching, evaluates Equations (1)–(4), and returns ranked JSON together with Leaflet map coordinates and XIR vectors. No external LLM or paid API is called at query time, keeping latency low.

A citizen requests a direct quote, creating an inquiry in status PENDING. The vendor logs into the Vendor Operations Portal and marks the quote ACCEPTED, which unlocks direct contact details with zero platform commission. Once the work is finished and marked COMPLETED, the citizen can submit a review. The review endpoint checks ownership and completion status, runs a linguistic contradiction filter to flag suspicious star-text mismatches for Admin audit, and immediately recomputes S_{bayes} and MFVTS.

4.5 Security properties of the role and review layer

Public search routes are read-only and never expose private customer bookings or unaccepted vendor phone numbers. Passwords are hashed with PBKDF2-SHA256 / bcrypt, and JWT tokens carry user id and role claims [12]. Mutating routes for inquiries and bookings verify that the caller owns the target vendor or customer record; a wrong role or non-owner receives HTTP 403. Admin KYC and anomaly moderation routes require the admin role claim.

5 Hypotheses

The problem stated above is tested through three hypotheses. They are stated before the results so that the evaluation can be read against them. They concern ranking quality and fairness, rating integrity, and multi-role access control.

H1 (Evidence-aware ranking). If vendor ranking fuses full-text relevance, Dynamic Bayesian Shrinkage ($m = 4.20$, $C = 5.0$), continuous Haversine decay, MFVTS, and cold-start exploration as in (3), then top-k ranking quality (NDCG@5) and cold-start exposure equity (Gini) improve measurably over a keyword-plus-raw-average baseline.

H2 (Verified-booking integrity). If reviews are accepted only for bookings with status COMPLETED owned by the submitting customer, and suspicious star-text contradictions are flagged, then Sybil review-hijack vulnerability is sharply reduced without blocking genuine customer feedback.

H3 (Shared ranking and role isolation). If Customer, Vendor, Admin, and Researcher portals share one ranking engine and enforce JWT ownership checks at the API boundary, then all four surfaces stay consistent while unauthorized cross-role mutations are rejected.

6 Results and discussion

6.1 Implementation and setup

The architecture was implemented as an Angular 19 frontend [13] and a Python 3.12 FastAPI backend [14] backed by PostgreSQL [7] with SQLAlchemy 2.0, Alembic migrations, and Leaflet.js + OpenStreetMap cartography [15]. Automated verification comprises 150 tests (62 unit, 48 API, 25 security/isolation, and 15 Playwright end-to-end tests), all passing in approximately 36 seconds on reference developer hardware. No local GPU is used; ranking is pure arithmetic over indexed relational data. This setup is the experimental platform.

6.2 Functional, ranking, and isolation evaluation

Evaluation combines a 500-query benchmark corpus (and the 30-query PHIR-30 live suite in the Researcher Lab) with functional and isolation checks on the deployed system. Empirical outcomes on the deployed Smart Vendor Hub system are summarised below.

NDCG@5 improved from 0.7564 (raw average baseline) to 0.9718 (+28.4% gain). Cold-start exposure Gini fell from 0.7420 (starvation) to 0.2008 (equitable exposure). Sybil attack top-rank hijack rate dropped from 42.6% to 3.2% (−92.5% reduction). Mean search latency was 12.4 ms. The “Why this rank?” modal shows per-factor XIR breakdown of S_{rel} , S_{geo} , Bayesian rating and trust. Review of a non-COMPLETED booking is rejected with HTTP 403. Vendor B cannot edit Vendor A’s inquiry (ownership check). Contradictory 5-star rating with negative text is flagged for Admin audit by the NLP anomaly filter.

6.3 Comparison with alternative directory designs

A raw-average SQL directory is simple to code but fails under sparse ratings and fake-review bursts. A commercial pay-to-play or high-commission marketplace monetises traffic at the cost of ranking bias and 20–35% fee extraction. The chosen design—one PostgreSQL index, composite Bayesian–Haversine–MFVTS scoring, mandatory completed-booking gate, and XIR attribution—minimises moving parts while keeping fairness and isolation in a single auditable code path.

6.4 Discussion

The results support three claims. First, combining Dynamic Bayesian Shrinkage with continuous Haversine decay and MFVTS measurably improves NDCG@5 (0.9718 vs 0.7564) and reduces cold-start exposure Gini (0.2008 vs 0.7420) without requiring a GPU or neural reranker. Second, enforcing completed-booking ownership at the API boundary blocks unverified review spam and keeps the four role portals isolated on one database. Third, post-hoc XIR attribution can be computed at query time in 12.4 ms mean latency and raises user trust in the ranking order. A practical limitation is that the evaluation corpus is a controlled Pune municipal benchmark rather than a multi-year live citizen traffic log.

7 Conclusion

This paper treated PostgreSQL and a transparent mathematical ranking pipeline as the fairness and isolation backbone of a hyperlocal municipal service discovery platform. A shared relational schema with GIN and GiST indexes, Dynamic Bayesian Shrinkage, continuous Haversine distance decay, Multi-Factor Verification Trust Scoring, and a mandatory completed-booking review gate together satisfy the requirement that honest local technicians are ranked by evidence rather than ad spend, and that unverified users cannot manipulate vendor ratings. Smart Vendor Hub instantiates the design with an Angular 19 frontend, a FastAPI backend, a zero-commission direct-quote workflow, and a live Researcher simulation lab. The three hypotheses are supported by the 500-query benchmark and 150 automated tests: the composite formula improves ranking quality and equity, the booking gate preserves rating integrity, and role guards isolate Customer, Vendor, Admin, and Researcher actions. The architecture is offered as a reproducible pattern for civic platforms that need fair, explainable local discovery.

8 Future scope

The next steps follow directly from what this study did not implement. Automated OCR and municipal license API checks would speed up Admin KYC verification. Hybrid sparse–dense retrieval could complement PostgreSQL full-text search for colloquial Marathi and Hindi service queries. Native mobile apps with live technician GPS tracking and transit ETA routing are the natural field-operations extension. Online bandit or learning-to-rank weight tuning on live citizen click logs would extend the static weight schema evaluated here.

9 Limitations

Retrieval and ranking quality were evaluated on a controlled 500-query benchmark and a 30-query live viva suite rather than a multi-city production A/B test with live citizens. While the benchmark captures sparse-rating and cold-start scenarios, live seasonal demand spikes remain to be studied. Each search query is stateless and independent; multi-turn conversational repair is not implemented. KYC document approval is performed manually by an Administrator in the moderation queue. Finally, the cold-start bonus and anti-monopoly damping constants were set by reasoned defaults ($B_{\text{cold}} \leq 0.15$, $m = 4.20$, $C = 5.0$); adaptive per-category prior calibration is left to future work.

Acknowledgment

The authors would like to express their heartfelt gratitude to Prof. Dr. K. R. Joshi, Principal, and Prof. Dr. Shivani A. Budhkar, Head of the Department, for their constant encouragement and valuable support. Special thanks are extended to the internal project guide, Prof. Ashwini Garkhedkar, for invaluable guidance on Bayesian rating estimation, geodesic spatial modelling, multi-factor trust scoring, and algorithmic fairness. The authors also acknowledge the open-source communities behind Angular, FastAPI, PostgreSQL, and Leaflet.js.

References

- [1] Kumar S, Smith SR, Fowler G, Velis C, Kumar SJ, Arya S, et al. Challenges and opportunities associated with waste management in India. *Royal Society Open Science*. 2017;4(3):160764. <https://doi.org/10.1098/rsos.160764>
- [2] Amasuomo E, Baird J. The concept of waste and waste management. *Journal of Management and Sustainability*. 2016;6(4):88–96.
- [3] Kadus T, Nirmal P, Kulkarni K. *Smart hyperlocal service management system*. International Journal of Engineering Research & Technology (IJERT). 2020;9(4).
- [4] Gupta K. *Smart civic service management system using automated discovery*. International Journal of Engineering Research. 2021.
- [5] de Campos LM, Fernández-Luna JM, Huete JF. A collaborative filtering method based on Bayesian networks. *Inf. Sci.* 2010;180(14):2736–2753.
- [6] Arthur MP, Shoba S, Pandey A. *A survey of intelligent discovery systems using machine learning*. Artif. Intell. Rev. 2024;57(56).
- [7] PostgreSQL Global Development Group. *PostgreSQL documentation: Full-text search and GIN/GiST indexes* [Internet]. 2026. <https://www.postgresql.org/docs/>
- [8] Nisarga NTD, Sahana S, Parashar S, Girijamaba DL. *Civic monitoring system using integrated web technology*. International Journal of Engineering Research & Technology (IJERT). 2018;6(13).
- [9] Imran, Ahmad S, Kim DH. *Quantum GIS based descriptive and predictive data analysis for effective planning of waste management*. IEEE Access. 2020;8.

-
- [10] Shahab S, Anjum M. Solid waste management scenario in India and illegal dump detection using deep learning: An AI solution towards the development of clean India. *Sustainability*. 2022;14(23):15896.
 - [11] Krebs R, Momm C, Kounev S. Architectural concerns in multi-tenant SaaS applications. In: *Proceedings of the 2nd International Conference on Cloud Computing and Services Science (CLOSER)*. 2012. p. 426–431.
 - [12] OWASP Foundation. OWASP Top 10 – 2021 [Internet]. <https://owasp.org/Top10/>
 - [13] Angular Team. *Angular documentation (v19)* [Internet]. <https://angular.dev>
 - [14] Ramírez S. *FastAPI documentation* [Internet]. <https://fastapi.tiangolo.com>
 - [15] Agafonkin V, OpenStreetMap Contributors. *Leaflet.js interactive maps documentation* [Internet]. <https://leafletjs.com>
 - [16] Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.* 2009;3(4):333–389.
 - [17] Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 2002;20(4):422–446.
 - [18] Singh A, Joachims T. Fairness of exposure in rankings. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018. p. 2219–2228.
 - [19] Bayer M. *SQLAlchemy 2.0 documentation* [Internet]. <https://docs.sqlalchemy.org>
 - [20] Microsoft. *Playwright end-to-end testing documentation* [Internet]. <https://playwright.dev>