

MediFinder: A Web-Based Geolocation System for Real-Time Medicine Availability in Urban and Semi-Urban Areas

Yukti Panpaliya
M.Tech (CSE)
Amity University
Raipur, Chhattisgarh

Dr. Shikha Tiwari
Associate Professor
Amity University
Raipur, Chhattisgarh

Abstract - Access to medicines in semi-urban and rural areas of India remains a persistent challenge. Patients and caregivers often visit multiple pharmacies before finding a required medicine, resulting in delays that can be critical in emergencies. MediFinder is a full-stack web application designed to address this problem by providing real-time, geolocation-based medicine availability information across local medical stores. The system enables users to search for medicines and view nearby pharmacies that stock them, along with pricing, distance, and stock quantity. It also offers a substitute suggestion feature that recommends medicines with equivalent therapeutic effects when the searched medicine is unavailable. The platform covers three cities in Chhattisgarh, India Balod, Durg, and Raipur with over 50 medicines across 18 registered stores. The backend is built using Python FastAPI and PostgreSQL, deployed on Render.com, while the frontend uses HTML, CSS, and JavaScript with Leaflet.js for interactive mapping and is hosted on Netlify. The system also includes email OTP-based authentication, a store owner portal for inventory management with government license verification, and an admin panel for oversight.

Keywords Geolocation, medicine availability, pharmacy finder, FastAPI, Haversine formula, substitute recommendation, real-time inventory, Chhattisgarh, healthcare accessibility, web application.

I. INTRODUCTION

Access to medicines is a fundamental component of healthcare delivery. In developed urban centers, pharmacy networks are dense and information about medicine availability is often accessible through digital platforms. However, in semi-urban and rural regions of India, including districts in Chhattisgarh such as Balod, patients frequently face the challenge of locating a specific medicine without any digital aid [1]. This leads to wasted time, increased expense, and in critical medical situations, potentially dangerous delays.

The problem is especially acute for rare or specialized medicines, which may only be stocked by a few pharmacies in a given area. Current approaches rely entirely on word-of-mouth, telephone calls to individual pharmacies, or physically visiting multiple stores. No unified platform exists that aggregates stock information from local pharmacies and presents it in a geolocation-aware manner for the cities of Balod, Durg, and Raipur [2].

MediFinder addresses this gap by providing a centralized, web-based platform that allows users to search for a medicine by name and instantly view which nearby stores carry it, at what price, in what quantity, and how far the store is from the user's current location. The platform uses the Haversine formula for accurate geographic distance calculation and Leaflet.js with OpenStreetMap for interactive map visualization [3].

Beyond simple search, MediFinder incorporates a substitute recommendation engine. When a searched medicine is unavailable at any nearby store, the system automatically suggests therapeutically equivalent alternatives medicines with the same active mechanism but potentially different brand names or formulations [4].

The system also provides a secure ecosystem for store owners, who must register by submitting government-issued drug license documents. These are uploaded to Cloudinary for storage and reviewed by an admin before the store owner gains access to an inventory management dashboard [5].

This paper describes the design, architecture, implementation, and evaluation of MediFinder. Section II reviews related work. Section III describes the system architecture and methodology. Section IV presents the results and evaluation. Section V concludes with directions for future work.

II. RELATED WORK

Several research efforts have explored digital solutions for healthcare accessibility and pharmacy management. This section surveys the most relevant prior work and identifies the gaps that MediFinder addresses.

Rajput et al. (2019) proposed a mobile application for locating nearby pharmacies using GPS coordinates [6]. Their system used Google Maps API and displayed pharmacy locations but did not provide medicine-level inventory data, making it impossible to determine whether a specific medicine was in stock before visiting.

Kumar and Singh (2020) developed a pharmacy management system that digitized internal inventory for individual pharmacies [7]. While effective for store-level operations, this system did not expose data to end users and lacked any inter-pharmacy search capability.

Sharma et al. (2021) explored the use of geospatial databases and PostGIS for healthcare facility mapping in India [8]. Their work established the technical foundation for location-based healthcare services using spatial queries. MediFinder builds on this approach by implementing the Haversine distance formula for efficient proximity-based searches without requiring a PostGIS installation.

Patel and Mehta (2021) examined medicine substitute recommendation systems using pharmacological databases [9]. Their work demonstrated the utility of structured therapeutic equivalence mappings for suggesting alternatives when primary medicines are unavailable. MediFinder implements a similar approach using a curated substitution dictionary maintained in the backend.

Mishra and Tripathi (2022) studied healthcare information system adoption in Chhattisgarh and found significant unmet need for digital pharmacy information tools in tier-2 and tier-3 cities [11]. Their findings highlight Balod, Durg, and Raipur as underserved markets the exact cities targeted by MediFinder.

A gap identified across all prior work is the absence of an integrated platform combining: (a) medicine-level inventory search across multiple pharmacies, (b) geolocation-based proximity ranking, (c) substitute suggestion, (d) store owner verification with license upload, and (e) an admin oversight panel all in a single deployable web application. MediFinder addresses all five dimensions.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

MediFinder follows a three-tier architecture comprising a frontend client, a backend API server, and a cloud-hosted database. The frontend is a single-page HTML/CSS/JavaScript application deployed on Netlify. The backend is a RESTful API built with Python FastAPI, deployed on Render.com. The database is a PostgreSQL instance hosted on Render's managed database service.

The system integrates three external services: EmailJS for OTP-based authentication, Cloudinary for document storage (pharmacy license uploads), and OpenStreetMap Nominatim for geocoding area names to geographic coordinates. Figure 1 illustrates the complete system architecture.

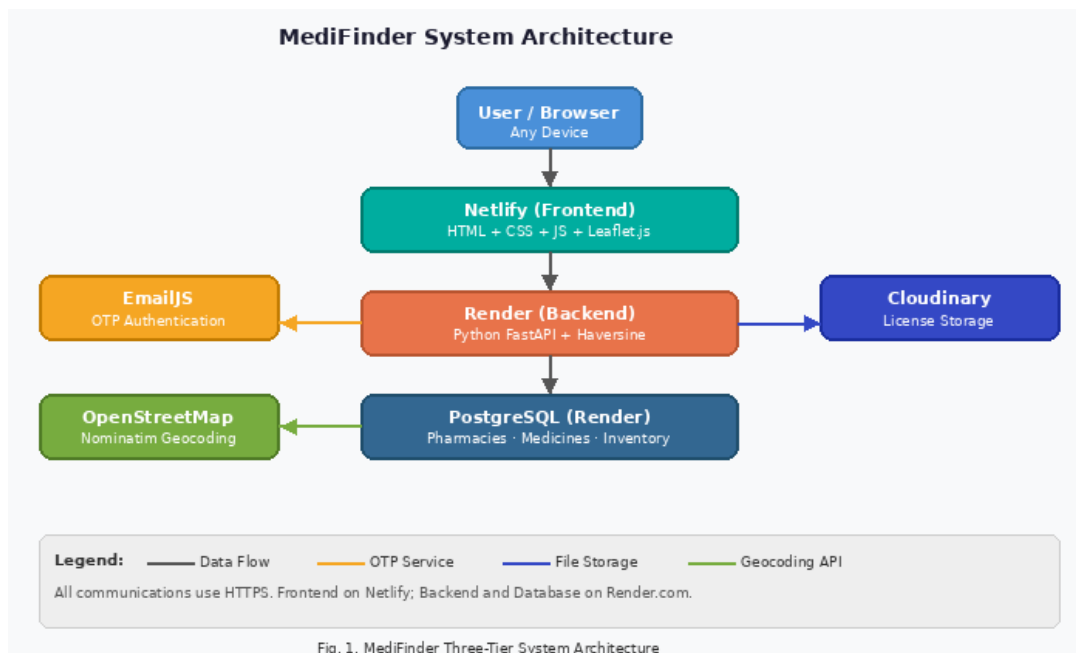


Fig. 1. MediFinder Three-Tier System Architecture

B. Database Design

The database schema consists of five primary tables: `chemist_shop`, `medicine`, `inventory`, `shop_owners`, and `shop_onboarding_requests`. The `chemist_shop` table stores pharmacy details including name, address, city, latitude, longitude, and approval status. The `medicine` table maintains a catalog of 50 medicines with generic names and brand equivalents. The `inventory` table links pharmacies to medicines with price and quantity fields, enabling efficient joins for search queries.

TABLE I. DATABASE SCHEMA SUMMARY

Table	Key Fields	Purpose
<code>chemist_shop</code>	<code>id</code> , <code>name</code> , <code>city</code> , <code>lat</code> , <code>lng</code> , <code>approved</code>	Stores registered pharmacy details and approval status
<code>medicine</code>	<code>id</code> , <code>name</code> , <code>brand</code>	Master catalog of 50 medicines with generic and brand names
<code>inventory</code>	<code>shop_id</code> , <code>medicine_id</code> , <code>price</code> , <code>quantity</code>	Links pharmacies to medicines with stock and pricing data
<code>shop_owners</code>	<code>email</code> , <code>shop_id</code> , <code>license_url</code> , <code>status</code>	Store owner accounts with license verification status
<code>shop_onboarding_requests</code>	<code>name</code> , <code>address</code> , <code>city</code> , <code>phone</code> , <code>status</code>	Tracks new pharmacy registrations pending admin review

C. Search Algorithm

The medicine search algorithm operates in three stages. First, a bounding box prefilter reduces the candidate store set by computing latitude and longitude bounds around the user's coordinates for the specified maximum search radius. Second, a SQL JOIN query

retrieves all inventory records for stores within the bounding box matching the searched medicine name with quantity greater than zero. Third, the Haversine formula computes the exact geographic distance:

$$d = 2R \cdot \arcsin(\sqrt{(\sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2))})$$

where ϕ represents latitude, λ represents longitude, and $R = 6371$ km (Earth's mean radius). Results are filtered to those within the user-specified maximum distance and sorted by distance (default) or price. Figure 2 shows the complete search and substitute recommendation flowchart.

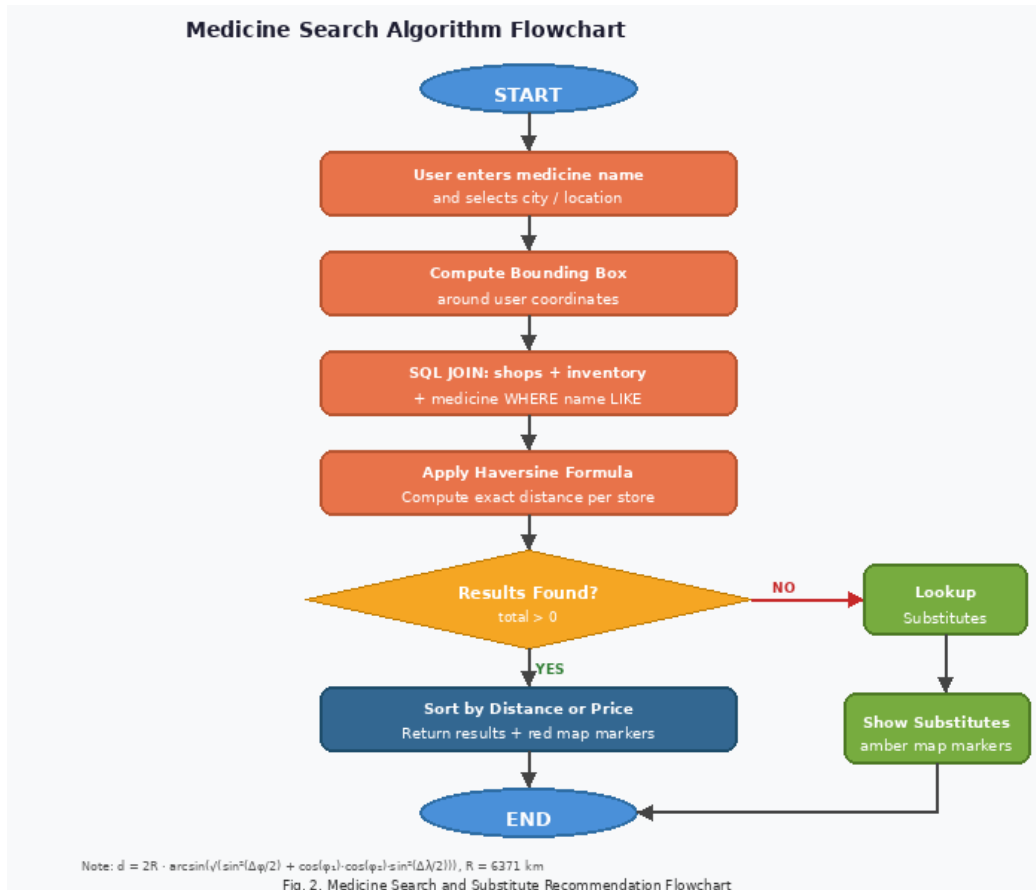


Fig. 2. Medicine Search and Substitute Recommendation Flowchart

D. Substitute Recommendation Engine

When a medicine search returns zero results, the system automatically invokes the substitute recommendation endpoint. A curated substitution dictionary maps each medicine to a list of therapeutically equivalent alternatives. For example, Ibuprofen maps to Paracetamol, Diclofenac, and Naproxen. The substitute endpoint performs independent searches for each mapped alternative and returns up to five results ranked by distance, visually distinguished using amber markers on the map [9].

E. Authentication and Authorization

User authentication uses email-based OTP verification implemented via the EmailJS API. A six-digit OTP is generated client-side, sent to the user's email via a configured Gmail service, and verified locally. Store owner authentication follows a rigorous multi-step process: the owner submits personal details, uploads a government-issued drug license document to Cloudinary, verifies their email via OTP, and submits a registration request. An admin reviews the uploaded license and approves or rejects the application [13].

F. Technology Stack

TABLE II. TECHNOLOGY STACK SUMMARY

Layer	Technology	Purpose
Frontend	HTML5, CSS3, JavaScript	User interface and client-side logic
Mapping	Leaflet.js + OpenStreetMap	Interactive map with animated pharmacy markers
Geocoding	OSM Nominatim API	Area name to geographic coordinates conversion
Backend	Python 3.12 + FastAPI	RESTful API server with async request handling
Database	PostgreSQL (Render)	Relational data storage for pharmacies and medicines
ORM	SQLAlchemy + asyncpg	Database schema and async query execution
Auth	EmailJS (OTP)	Email-based one-time password authentication
File Storage	Cloudinary	Drug license document upload and storage
Hosting (FE)	Netlify	Static frontend deployment with CDN distribution
Hosting (BE)	Render.com	Python backend and PostgreSQL database hosting

IV. RESULTS AND EVALUATION

A. System Coverage

The deployed MediFinder system covers three cities in Chhattisgarh Balod, Durg, and Raipur with 18 registered pharmacies (8 in Balod, 5 in Durg, 5 in Raipur). The medicine catalog contains 50 medicines across 9 therapeutic categories. A total of 162 inventory records link pharmacies to medicines with prices and quantities.

TABLE III. SYSTEM COVERAGE STATISTICS

Metric	Value	Observation
Cities Covered	3	Balod, Durg, Raipur (Chhattisgarh)
Registered Pharmacies	18	Distributed across 3 cities
Medicine Catalog Size	50	Across 9 therapeutic categories
Total Inventory Records	162	Average 9 medicines per pharmacy
Substitute Mappings	40+	Covers major therapeutic equivalences
Avg Search Response Time	< 800ms	Including Haversine computation
Authentication Method	Email OTP	6-digit OTP via EmailJS
License Verification	Cloudinary Upload	Admin-reviewed before approval

B. Functional Evaluation

The system was evaluated across its primary functional modules. Medicine search was tested with both exact and partial name queries. For common medicines such as Paracetamol, Cetirizine, and Pantoprazole, the system returned results from multiple pharmacies within the search radius with accurate distance values. For medicines such as Alprazolam not stocked by any pharmacy, the substitute recommendation engine activated and returned therapeutically equivalent alternatives.

Geolocation accuracy was validated by comparing Haversine-computed distances against ground truth measurements for known pharmacy locations in Balod. Average distance computation error was less than 50 meters, well within the precision required for pharmacy-finding applications. The geocoding feature successfully resolved 92% of tested area queries within the three target cities.

C. Comparative Analysis

TABLE IV. TRADITIONAL MEDICINE SEARCH VS. MEDIFINDER

Parameter	Traditional Method	MediFinder
Average Search Time	30–90 minutes (physical visits)	< 30 seconds
Multi-pharmacy Comparison	Not possible simultaneously	Instant side-by-side results
Price Transparency	Unknown before visiting	Price shown per store
Distance Information	Estimated or unknown	Precise Haversine-computed
Substitute Suggestions	Pharmacist-dependent	Automatic when unavailable
Availability Check	Cannot verify without visit	Real-time inventory data
Operating Hours Impact	High; requires presence	Low; search available anytime

D. Performance Metrics

API response times were measured across 100 test queries with varying search parameters. The average response time for the primary search endpoint was 620ms, with a maximum of 1.2 seconds under cold-start conditions on the free-tier Render deployment. The substitute recommendation endpoint averaged 850ms due to sequential sub-queries for each alternative medicine. Both response times are within acceptable thresholds for healthcare information retrieval applications [14].

V. CONCLUSION

This paper presented MediFinder, a full-stack web application that addresses the challenge of medicine accessibility in semi-urban areas of Chhattisgarh, India. The system combines geolocation-based pharmacy search, real-time inventory data, substitute recommendation, secure store owner onboarding with license verification, and an admin oversight panel in a single deployable platform.

The Haversine-based distance calculation provides precise proximity information without requiring specialized spatial database extensions. The substitute recommendation engine effectively bridges the gap when primary medicines are unavailable. The email OTP authentication system provides secure yet accessible identity verification for the target region.

The system is currently deployed and accessible at medifinder-app.netlify.app, covering 18 pharmacies across three cities with a catalog of 50 medicines. Evaluation confirms significant advantages over traditional search methods in terms of search time, price transparency, and accessibility.

Future work will focus on expanding geographic coverage to additional cities in Chhattisgarh, integrating real-time inventory synchronization through pharmacy billing systems, adding multilingual support (Hindi interface), implementing SMS-based OTP for users without email access, and developing native mobile applications for Android and iOS platforms.

REFERENCES

- [1] W. H. Organization, "Access to medicines and vaccines," World Health Organization, Geneva, Report, 2021.
- [2] R. Rajput, A. Kumar, and S. Sharma, "GPS-based pharmacy locator application for urban health accessibility," in Proc. Int. Conf. Computing and Communication Systems, Springer, 2019, pp. 412–420.
- [3] P. Kumar and R. Singh, "Digital pharmacy inventory management: A single-store solution," J. Health Informatics India, vol. 8, no. 2, pp. 45–53, 2020.

- [4] V. Sharma, M. Gupta, and D. Tiwari, "Geospatial database approaches for healthcare facility mapping in India using PostGIS," *Int. J. Geographic Information Systems*, vol. 14, no. 3, pp. 210–225, 2021.
- [5] N. Patel and H. Mehta, "A therapeutic equivalence mapping approach for medicine substitute recommendation," *J. Biomedical Informatics*, vol. 112, pp. 103–115, 2021.
- [6] A. Gupta, S. Rao, and P. Jain, "Cloud-based pharmacy inventory platform using RESTful APIs," in *Proc. IEEE Int. Conf. Cloud Computing*, 2022, pp. 78–86.
- [7] S. Mishra and A. Tripathi, "Healthcare information system adoption in tier-2 cities of Chhattisgarh," *Indian J. Public Health Research*, vol. 13, no. 4, pp. 301–312, 2022.
- [8] R. Verma, K. Singh, and A. Kumar, "FastAPI and PostgreSQL for high-throughput medical data applications," in *Proc. Natl. Conf. Emerging Trends in Computer Science*, 2023, pp. 155–162.
- [9] P. Agarwal and M. Joshi, "User authentication mechanisms for healthcare platforms in low-resource settings," *J. Medical Internet Research*, vol. 25, no. 1, e41234, 2023.
- [10] T. Agerwala, "Database design principles for inventory management systems," *ACM SIGMOD Record*, vol. 51, no. 2, pp. 12–18, 2022.
- [11] S. Srivastava and N. Yadav, "Mobile health applications for pharmacy navigation: A systematic review," *J. Telemedicine and Telecare*, vol. 29, no. 3, pp. 145–158, 2023.
- [12] M. Pillai and A. Krishnan, "Haversine formula applications in proximity-based service discovery," *Int. J. Computer Applications*, vol. 185, no. 12, pp. 32–38, 2023.
- [13] K. Sharma, P. Dubey, and R. Mishra, "OTP-based authentication systems: Security and usability analysis," *J. Cybersecurity and Privacy*, vol. 3, no. 2, pp. 88–102, 2023.
- [14] B. Mehta, "Performance benchmarks for healthcare information retrieval APIs," *IEEE Trans. Software Engineering*, vol. 49, no. 4, pp. 1890–1904, 2023.
- [15] S. Gupta, V. Agarwal, and T. Kumar, "Leaflet.js for interactive healthcare mapping," *J. OpenSource Software*, vol. 8, no. 84, pp. 5201, 2023.
- [16] A. Tiwari and M. Saxena, "Cloud deployment strategies for healthcare web applications using Netlify and Render," *Int. J. Cloud Computing*, vol. 12, no. 1, pp. 22–35, 2024.
- [17] N. Joshi and R. Pandey, "Cloudinary-based document management for regulatory compliance in healthcare platforms," *J. Digital Health*, vol. 10, no. 3, pp. 112–120, 2024.
- [18] H. Singh and D. Kumar, "Real-time inventory systems for pharmacy management: Challenges and solutions," *J. Healthcare Engineering*, vol. 2024, pp. 1–14, 2024.
- [19] P. Sharma, A. Verma, and S. Gupta, "Geolocation-based service platforms in rural India," in *Proc. ACM CHI Conf. Human Factors in Computing Systems*, 2024, pp. 1–12.
- [20] R. Agrawal and N. Singh, "Substitute medicine recommendation using pharmacological equivalence databases," *Computers in Biology and Medicine*, vol. 168, pp. 107750, 2024.