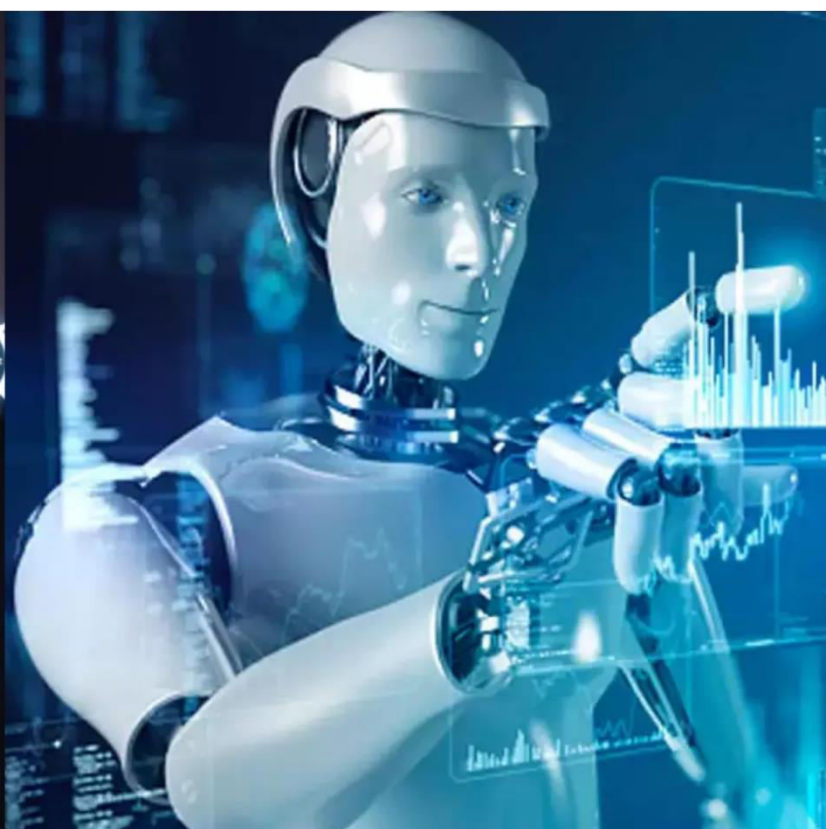




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

SmartMedi: A Role-Based Platform for Verified Medicine Discovery, Pharmacy Price Comparison and Safety Reporting

Pradeep M R G, Rishwanth M, Venkatalakshmi S

U.G. Students, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,
Sivagangai, Tamil Nadu, India

Assistant Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,
Sivagangai, Tamil Nadu, India

ABSTRACT: Patients who look for a medicine online usually find composition details, prices and opinions scattered over different websites, listings are rarely reviewed by anyone qualified, and suspected adverse reactions have no simple route to a reviewer. This paper presents SmartMedi, a role-based web platform for verified medicine discovery, pharmacy price comparison and safety reporting. Pharmacies and verified doctors submit formulations that stay in a pending state until an administrator approves them, professional accounts are cleared only after a documented credential check, and every approval, rejection and safety review is written to an audit table. Patients can search and compare approved medicines, view pharmacy listings, read reviews tagged by a lexicon-based sentiment analyzer, ask an English/Tamil assistant, and place cash-on-delivery orders whose prices are always computed on the server. A TF-IDF recommender suggests related medicines and a heuristic estimator gives an indicative price range. The platform is built with Flask, SQLAlchemy and Jinja templates on SQLite, MySQL or PostgreSQL. On a seeded demonstration catalogue, all 17 automated regression tests passed, and in-process requests were served in 0.8 to 9.0 ms on average. The evaluation also showed that the uncalibrated price heuristic and the small catalogue limit the price and recommendation output, and these limits are reported openly. SmartMedi is an information prototype and not a clinical decision tool.

KEYWORDS: Medicine Information System, Price Comparison, Role-Based Access Control, TF-IDF, Cosine Similarity, Adverse Event Reporting, Sentiment Analysis, Flask, SQLAlchemy.

I. INTRODUCTION

Medicines in India are sold under the Drugs and Cosmetics Act, 1940 [1], and a patient who wants to know what a tablet contains, what it treats, what it costs at different pharmacies and what other patients experienced has to consult several sources. Brand names, strengths and prices differ between shops, composition and side-effect information is spread over package inserts, manufacturer sites and retail pages, and a visitor cannot easily tell whether a listing has been reviewed by anyone qualified.

Two further gaps matter in practice. First, manufacturers and pharmacies introduce new formulations continuously, and doctors need to hear about them in one place. Second, suspected adverse reactions are most useful when they reach a reviewer in a structured form. The Pharmacovigilance Programme of India [2] exists for this purpose, but an ordinary information portal rarely offers patients or clinicians a simple and private way to record such an observation. A portal intended for Tamil Nadu also benefits from supporting Tamil as well as English.

This paper presents SmartMedi, a role-based web platform that addresses these points in a prototype. The main contributions are: (i) a review workflow in which uploaded formulations remain pending until an administrator approves them, professional accounts are verified through a documented credential check, and every decision is stored in an audit table; (ii) a patient portal for searching and comparing approved medicines, comparing pharmacy listings, reading sentiment-tagged reviews and placing cash-on-delivery orders priced on the server; (iii) three transparent analytical modules, namely a TF-IDF recommender, a heuristic price indicator and a lexicon-based sentiment analyzer,

together with an English/Tamil assistant; and (iv) an open evaluation consisting of automated regression tests, response-time measurements and an examination of how the analytical modules behave on the seeded catalogue. SmartMedi is an information and demonstration system. It does not diagnose, prescribe or replace the advice of a doctor or pharmacist, it is not connected to any live retailer or courier, and its price and recommendation outputs are indicative only.

The rest of this paper is organized as follows. Section II reviews related work. Section III describes the methodology and system architecture. Section IV presents the user interfaces, experimental results and limitations. Section V concludes the paper and outlines future enhancements.

II. LITERATURE SURVEY

Information-retrieval methods are the usual basis for searching and relating text records. Salton and Buckley [3] compared term-weighting schemes and showed that weighting a term by its frequency in a document and by its rarity across the collection works well for ranking. This term frequency-inverse document frequency (TF-IDF) scheme, combined with cosine similarity between document vectors [4], remains a standard and easily inspected way to find similar items. Content-based recommendation, described by Ricci et al. [5], uses the attributes of the items themselves instead of the behaviour of other users, so it can work in a new system that has no interaction history. For medicines, similarity can therefore be computed from composition and indications, although similarity of text does not establish that two products are therapeutically interchangeable.

Reviews are a second source of information. Pang and Lee [6] survey opinion mining and sentiment analysis, and lexicon-based methods such as VADER [7] assign scores from word lists with rules for negation and intensity, which makes them transparent and usable without labelled training data. Their weakness is context: in a medicine review a word such as “pain” may describe the condition being treated and not dissatisfaction with the product, so a general lexicon needs domain care.

Estimating a price from product attributes is a regression problem for which ensemble methods such as random forests [8], available in scikit-learn [9], are a common choice. They need a documented dataset with consistent currency and pack sizes; without such data a model cannot be validated, and any fixed formula should be presented as a heuristic.

Because the platform handles accounts, orders and health-related reports, access control and web security are part of the design. Role-based access control [10] assigns permissions to roles instead of individuals, which suits a system with patients, doctors, pharmacies and administrators. Cross-site request forgery, in which the browser of a signed-in user is made to send an unwanted request, is prevented by tokens that are tied to the session and checked on every state-changing request [11]. Table I summarizes how this work relates to SmartMedi.

TABLE I. SUMMARY OF RELATED WORK AND ITS RELEVANCE TO SMARTMEDI

Ref.	Focus of the work	Relevance to SmartMedi
[1] Govt. of India, 1940	Regulation of drugs sold in India	Context for pharmacy and professional verification
[2] IPC, PvPI	National monitoring of adverse drug reactions	Model for structured safety reports
[3] Salton and Buckley, 1988	Term weighting for text retrieval	TF-IDF vectors of medicine descriptions
[4] Manning et al., 2008	Vector space model, cosine similarity	Ranking of similar medicines
[5] Ricci et al., 2015	Content-based recommender systems	Recommendation without usage history
[6] Pang and Lee, 2008	Opinion mining and sentiment analysis	Tagging of medicine reviews
[7] Hutto and Gilbert, 2014	Rule-based sentiment with negation	Design of the review analyzer
[8] Breiman, 2001	Random forests	Planned price model (offline experiment)
[10] Sandhu et al., 1996	Role-based access control	Four-role permission model
[11] OWASP	CSRF prevention	Session tokens on all mutations

The surveyed work treats retrieval, recommendation, sentiment analysis, price modelling and web security separately. SmartMedi brings these elements together in one auditable workflow for medicine information in a Tamil Nadu setting, and it states clearly which components are heuristic and which have been tested.

III. METHODOLOGY

A. System Architecture

SmartMedi follows a four-tier architecture, shown in Fig. 1. The presentation tier consists of Jinja2 templates, CSS and a small amount of browser JavaScript for the login, registration and assistant widgets and for attaching the CSRF token to requests. The application tier is a Flask 3 [12] application built with the application-factory pattern; Flask-Login manages sessions, and six blueprints (authentication, medicines and orders, doctor portal, administration, research and safety, and the JSON API) separate the features. A security module is installed on every request. The intelligence tier is a set of plain Python modules (recommender, price estimator, sentiment analyzer and assistant) that use NumPy and scikit-learn [9]; when scikit-learn is missing, the recommender falls back to its own NumPy implementation of TF-IDF. The persistence tier uses SQLAlchemy [13] and works with SQLite by default, or with MySQL and PostgreSQL through the DATABASE_URL setting.

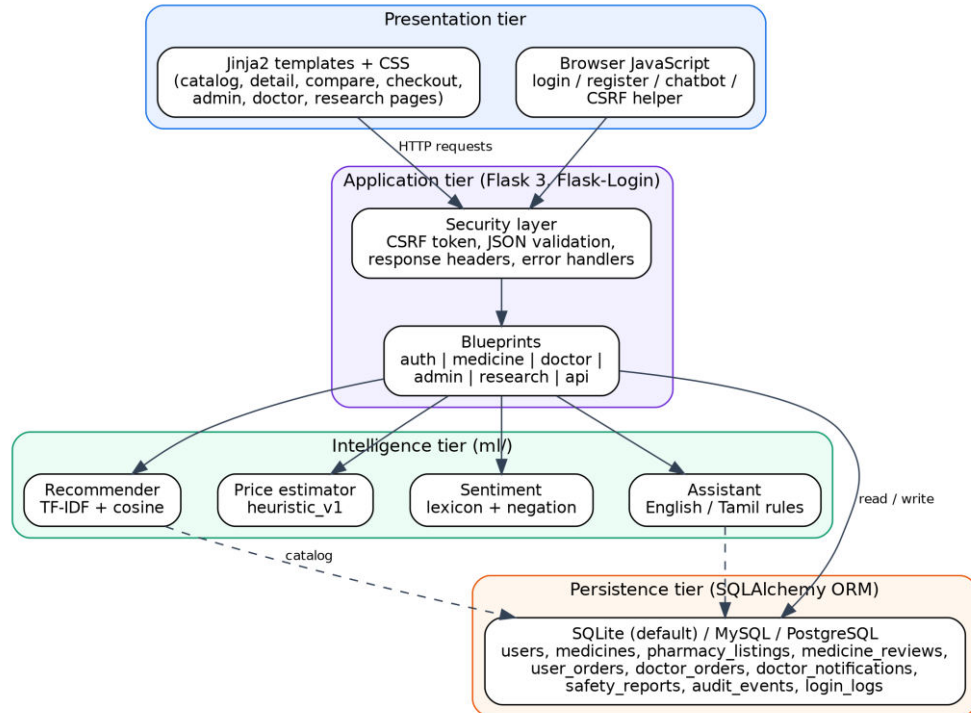


Fig.1.System architecture of SmartMedi

B. Roles and Review Workflow

There are four roles: patient, doctor, pharmacy (including pharmaceutical companies) and administrator. Public registration creates only patient, doctor and pharmacy accounts; any other requested role is replaced by the patient role, so administrator accounts cannot be self-registered. Doctors and pharmacies must enter a licence number and stay unverified until an administrator records a credential review of at least ten characters, which is stored in the audit table. This is a documented manual check; the system does not contact any regulator. Only verified professionals and administrators can upload medicines. An upload is stored with the status pending, an initial listing from the uploader and a risk level of “Unassessed”. The administrator can verify, reject or reset it, and on the first verification a notification is created for every registered doctor. Pending and rejected medicines are hidden from the public catalogue, search, comparison and ordering, and can be seen only by the administrator and the uploader. The complete workflow is given in Fig. 2.

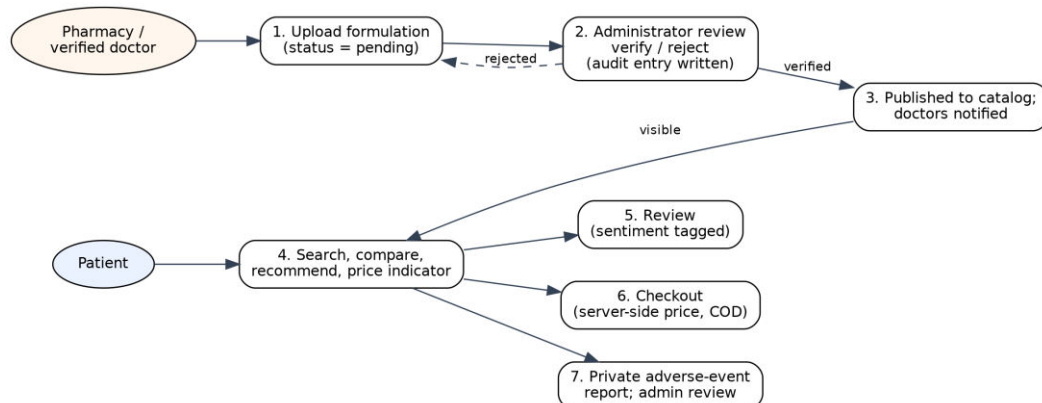


Fig.2.Review and usage workflow of SmartMedi

C. Data Model

The relational schema (Fig. 3) has ten tables: users, medicines, pharmacy_listings, medicine_reviews, doctor_notifications, doctor_orders, user_orders, login_logs, audit_events and safety_reports. A medicine record holds its composition, indications, benefits, disadvantages, base price and verification status, and each pharmacy listing holds the price, discount price and stock status of one shop. Login attempts are logged, and an audit event records the actor, action, entity and details of each administrative decision. The tables are created at start-up; versioned migrations have not been introduced yet.

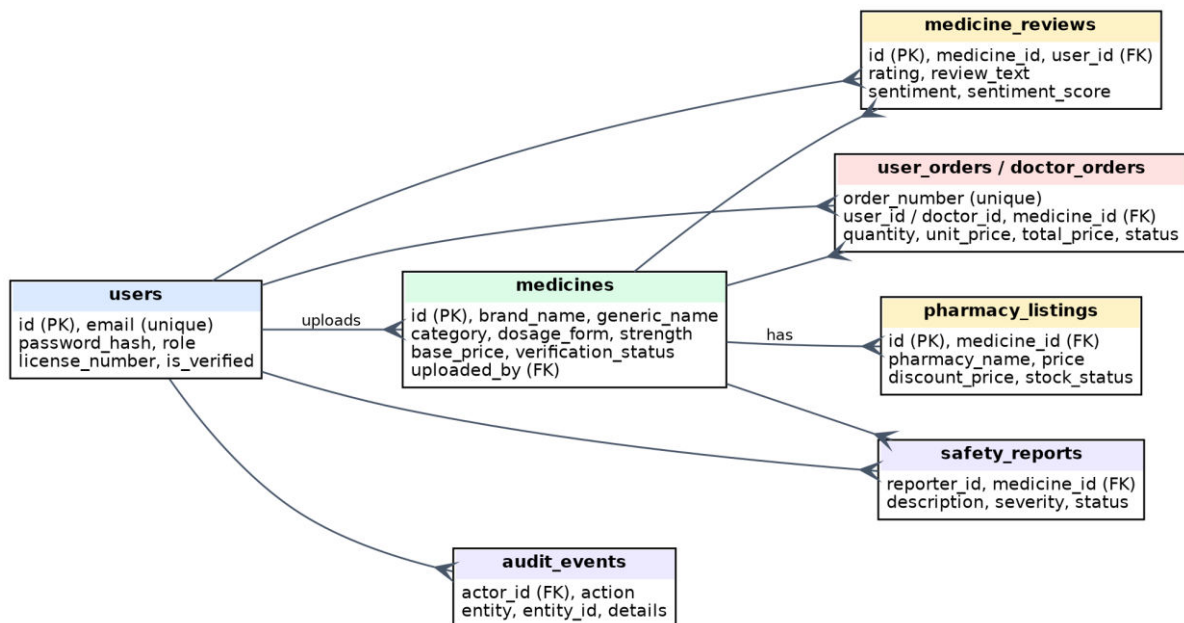


Fig.3.Main tables and relationships of the SmartMedi database

D. Recommendation Engine

For every verified medicine a text document is built from its active ingredients, category, indications, dosage form and chemical composition. TF-IDF vectors are computed over all verified medicines, the cosine similarity between the selected medicine and every other medicine is calculated, and items with a similarity above 0.15 are returned with the score as a percentage, a flag for the same category and the price difference. Symptom search treats the query as an additional document, compared with the indications, category, benefits and active ingredients of each medicine, and returns items with a similarity above 0.1. The vectorizer is refitted on every request, which is simple and adequate for a small catalogue. The scores are relevance indicators and not claims of therapeutic equivalence.

E. Price Indicator

The price estimator, named heuristic_v1 in the code, is a deterministic formula and not a trained model. The estimate is $8.50 + 2.85 \cdot \ln(1 + s) \cdot c \cdot f$ rupees, where s is the strength in milligrams read from the strength text, c is a multiplier for the therapeutic category (0.75 for vitamins and supplements up to 4.50 for oncology) and f is a multiplier for the dosage form (1.0 for tablets up to 2.80 for inhalers). The reported range is 0.85 to 1.20 times the estimate. When the listed price is compared with the estimate, a difference below -15% is labelled "Great Value", a difference within $\pm 15\%$ "Fair Market Price" and anything higher "Higher than Average". Every response carries validated = false and a notice that it is an illustrative estimate and not a validated market price. A separate offline script compares a median predictor with a random forest [8] by five-fold cross-validation on an exported catalogue, but that model is not used by the application.

F. Review Sentiment Analysis

A review is split into words and compared with two lists of 25 positive and 26 negative clinical terms. A negator such as "not" or "never" reverses the polarity of the next listed term, and an intensifier such as "very" multiplies its weight

by 1.5. The score is $(P - N)/(P + N)$, where P and N are the weighted positive and negative counts. A score above 0.15 is positive, below -0.15 negative and anything else neutral. Each stored review keeps its label and score, the detail page counts the three classes, and star ratings are validated to the range 1 to 5 independently of the text.

G. English and Tamil Assistant

The assistant accepts English or Tamil; Tamil is recognized by characters in the Unicode Tamil block (U+0B80 to U+0BFF). It handles greetings and a small set of intents (side effects, price, composition and category questions such as diabetes or fever) by filling reply templates in the matching language from the verified catalogue. A medicine is identified when its full brand name or full generic name appears in the message. There is no language model, and the replies advise consulting a physician.

H. Ordering and Server-side Pricing

At checkout the server selects the requested in-stock pharmacy listing, or the cheapest in-stock listing when none is requested, takes the unit price from that listing and ignores any price sent by the browser. The quantity must be between 1 and 20, a phone number and delivery address are mandatory, and only cash on delivery is accepted until a payment gateway is connected. The total is the subtotal plus a ₹49 fee for express delivery, minus a 5% discount when the subtotal is at least ₹500. An order number, tracking number and estimated delivery slot are generated for display; they are prototype records because no courier is integrated, and an order can be read only by the user who placed it. Verified doctors can place separate orders of up to 100 units, which remain in “Pending Approval”.

I. Security, Privacy and Auditability

Passwords are stored as salted hashes, and session and remember-me cookies are HttpOnly with SameSite=Lax. Every POST, PUT, PATCH and DELETE request, including JSON requests, must carry the session’s CSRF token in a form field or in the X-CSRF-Token header, which is compared in constant time [11]. JSON bodies must be flat objects with string text fields, request bodies are limited to 2 MB, and invalid values return a JSON error. Responses carry no-sniff, frame-options, referrer and no-store cache headers, and the redirect after login accepts only local paths. The administrator catalogue export neutralizes cells that start with =, +, - or @ so that spreadsheets do not execute them, and it contains no patient records. Safety reports are visible only to the reporter and the administrators, and audit events record the actor, action, entity and time. Data are collected only to run the service, consistent with the purpose-limitation principle of the Digital Personal Data Protection Act, 2023 [14]. The audit table is not tamper-proof against database administrators, and HTTPS, rate limiting and password-reset e-mail are not provided by the application; they must be added before any public deployment.

IV. EXPERIMENTAL RESULTS

A. User Interfaces

The catalogue page (Fig. 4) offers search by brand name, generic name or active ingredient, category and dosage-form filters, sorting and a compare selector, and the navigation bar shows only the links that fit the signed-in role. The detail page (Fig. 5) shows the composition, an indicative price assessment, pharmacy listings ordered by price with their discounts, recommendations, and reviews with sentiment counts. The administrator verification page (Fig. 6) shows the number of pending, verified and rejected medicines and lets the administrator review each submission. All screenshots were taken from a running instance with the seeded demonstration data.

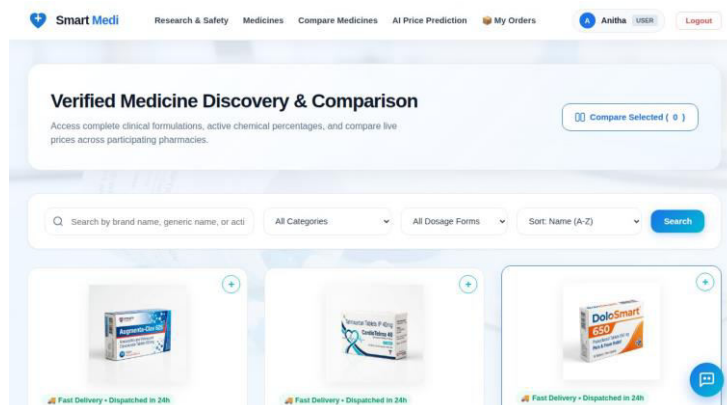


Fig.4.Verified medicine catalogue page (patient view)

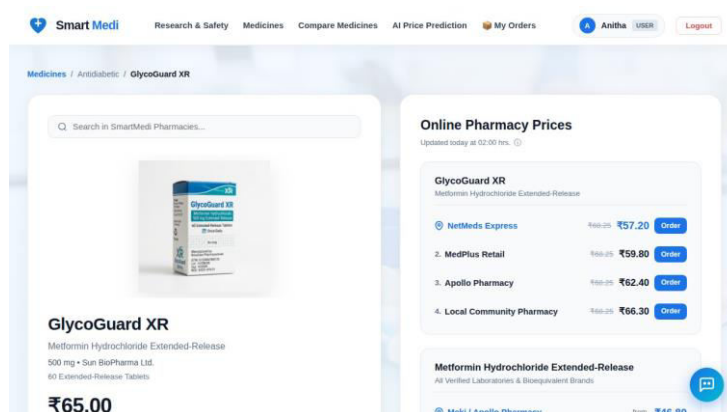


Fig.5.Medicine detail page with pharmacy price comparison

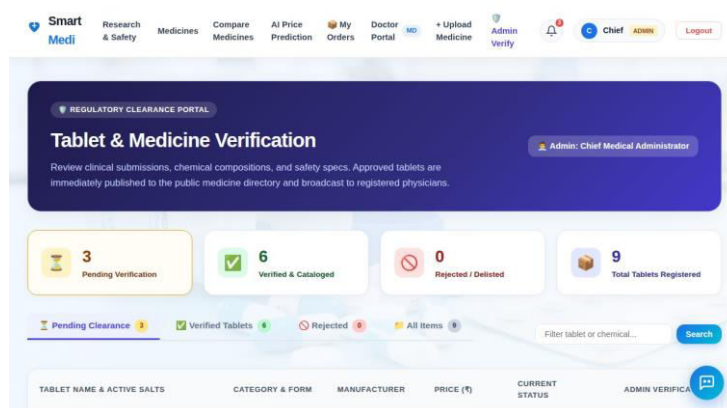


Fig.6. Administrator verification dashboard

B. Experimental Setup

The experiments used the demonstration database created by the project's seed scripts: five users (patient, doctor, pharmacy, demo and administrator), nine medicines of which six are verified and three are pending, 24 pharmacy listings from four pharmacies, and 18 reviews. All records were written by the developers; none are real market observations or patient data. The software ran on Python 3.12 with Flask 3.1, SQLAlchemy 2.1 and scikit-learn 1.8 on SQLite in a general-purpose Linux container. Response times were measured in-process with the Flask test client, 100

repetitions per request after login, so they exclude network transfer and browser rendering.

C. Functional and Security Testing

The regression suite contains 17 tests and runs on an in-memory SQLite database that never touches the configured database. All 17 tests passed in about 12 s. Table II groups them by what they verify.

TABLE.II. AUTOMATED REGRESSION TESTS BY AREA

Test area	What is verified	Passed
Order and price integrity	Server-side price ignores client values; orders are private and unknown orders return not found; only cash on delivery; out-of-stock items refused	3 / 3
CSRF and session	Requests without a token are rejected and pages render the token; logout needs POST and a token; external redirects after login are refused	3 / 3
Roles and verification	Doctor role and verification checks; uploads need a verified account and stay pending; credential review needs documented notes	3 / 3
Catalogue visibility and input	Unverified medicines are hidden; malformed payloads give errors; main pages render	3 / 3
Safety, audit and export	Safety reports are private and reviewable; approvals write audit entries and notify doctors; CSV export is admin-only without patient data	3 / 3
Research script	Reproducible metrics with duplicate grouping; datasets that are too small are rejected	2 / 2
Total		17 / 17

D. Response Time

Table III gives the mean and 95th-percentile time for the main requests on the seeded catalogue. The 95th-percentile time was under 16 ms for every request. Inside the intelligence tier, the recommender took about 2.4 ms on average for the six verified medicines, while the price estimator, the sentiment analyzer and the assistant each took well under 1 ms. Because the vectorizer is refitted on every request, the recommender time will grow with the size of the catalogue; this was measured only on six medicines.

TABLE.III. RESPONSE TIME OF MAIN REQUESTS (100 RUNS EACH)

Request	Mean (ms)	95th percentile (ms)
GET /medicines (catalogue)	8.4	11.2
GET /medicines/1 (detail, price analysis, recommendations)	9.0	15.4
GET /compare (three medicines)	5.0	8.0
POST /api/recommend (symptom query)	5.4	6.3
POST /api/predict-price	0.9	1.3
POST /api/analyze-sentiment	0.8	1.1
POST /api/chatbot	2.0	2.8

E. Behaviour of the Analytical Modules

Table IV reports what the analytical modules returned on the seeded data. These are observations on a small demonstration set and not accuracy measurements. The symptom search found the correct verified medicine for fever

and body pain, high blood pressure and diabetes, but the vectorizer does not stem words, so the query “infection” returned nothing while “infections” matched the antibiotic. The price indicator estimated ₹15.39 to ₹37.86 for medicines listed at ₹32 to ₹185 and therefore labelled all six as higher than average, which shows that its multipliers are not calibrated to the catalogue.

TABLE.IV. OBSERVED BEHAVIOUR OF THE ANALYTICAL MODULES ON THE SEEDED CATALOGUE

Module	Input	Observation
Symptom search	Five queries	Fever and body pain, high blood pressure and diabetes each returned the correct verified medicine (15.7%, 25.6%, 16.9%); “infection” returned nothing (no stemming); “acidity” returned nothing because the only gastric medicine is still pending
Similar medicines	Six verified medicines	None exceeded the 0.15 threshold; the six medicines belong to different therapeutic classes
Price indicator	Six verified medicines	Estimates of ₹15.39 to ₹37.86 against listed prices of ₹32 to ₹185; all six labelled “Higher than Average” (+32% to +810%)
Sentiment	18 seeded reviews	12 neutral and 6 positive, identical to labels derived from the star ratings (3 stars, 5 stars); the data contain no negative review, so negative handling is untested
Assistant	Twelve messages	Correct for full medicine names, for diabetes in English and Tamil, and for a greeting; no match for partial names (“Metformin”, “DoloSmart”); a greeting was returned for “fever with chills” and “which tablet is good” because “hi” occurs inside other words

F. Discussion and Limitations

All results come from demonstration records and a controlled environment, so they show that the workflow, access control and modules behave as designed, and not how accurate the platform would be on real catalogues. The price heuristic must not be read as a market judgement: its formula was set by hand, and the offline random-forest experiment has not been run because the six-row catalogue is below the ten distinct combinations the script requires. The recommender is only as useful as the catalogue is large and varied, and it needs stemming or synonym handling. The assistant needs partial-name matching and whole-word matching for greetings. Symptom words must also not be treated as diagnosis: the message “chest pain” returned the fever and pain medication list, which is why every reply carries advice to consult a physician.

Other limits follow from the prototype scope. Retail prices are locally entered and not live, orders do not progress beyond placement because no courier or fulfilment service exists, credential verification depends on an administrator’s external check, and no usability study, ethics review or clinical evaluation has been carried out. HTTPS, rate limiting, managed audit retention and versioned migrations are needed before any public service. SmartMedi therefore works as an information and screening aid, and decisions about treatment stay with qualified professionals.

V. CONCLUSION AND FUTURE ENHANCEMENT

This paper presented SmartMedi, a role-based platform for verified medicine discovery, pharmacy price comparison and safety reporting. By combining an administrator-approved publication workflow, documented credential checks, an audit trail, server-side pricing, session-bound CSRF protection and transparent TF-IDF, lexicon and rule-based modules, the platform gives patients, doctors and pharmacies a single place for medicine information. All 17 automated regression tests passed and requests were served in a few milliseconds on the seeded catalogue. The evaluation also exposed real weaknesses in the price heuristic, the vocabulary matching and the assistant, which are reported so that they can be corrected before any wider use.

In future work, the price model will be calibrated on a lawfully sourced dataset and compared with a median baseline and a random forest using the existing offline protocol. Stemming and synonym handling will be added to the recommender, and the assistant will be changed to match partial medicine names and whole words. A payment gateway, courier tracking and an order-status workflow will replace the prototype records, and HTTPS, rate limiting, e-mail password reset and versioned migrations will be introduced. Credential verification could be linked to an official register where one is available, and a usability study with ethics approval and relevance judgements from qualified reviewers will be used to evaluate the recommender and the review analysis.

REFERENCES

1. Government of India, "The Drugs and Cosmetics Act, 1940," Act No. 23 of 1940.
2. Indian Pharmacopoeia Commission, "Pharmacovigilance Programme of India (PvPI)." [Online]. Available: <https://ipc.gov.in>
3. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
4. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
5. F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
6. B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008.
7. C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. AAAI Conf. Weblogs Soc. Media (ICWSM)*, Ann Arbor, MI, USA, 2014, pp. 216–225.
8. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
9. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
10. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.
11. OWASP Foundation, "Cross-Site Request Forgery Prevention Cheat Sheet." [Online]. Available: <https://cheatsheetseries.owasp.org>
12. Pallets Projects, "Flask documentation." [Online]. Available: <https://flask.palletsprojects.com>
13. SQLAlchemy authors, "SQLAlchemy documentation." [Online]. Available: <https://docs.sqlalchemy.org>
14. Government of India, "The Digital Personal Data Protection Act, 2023," Act No. 22 of 2023.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details