Grenze International Journal of Engineering and Technology, June Issue

WellnessWay: A Unified MERN Stack Web Platform Coupling Real-Time Geospatial Healthcare Discovery with Machine Learning-Driven Symptom Analysis

Anmol Bandari[1], Ageer Nirvignya[2], Mulpuri Rakesh[3] and Shaik Abdul Khalandar Basha[4]
[1-4]Sreenidhi Institute of Science and Technology, India
Email: abdul.s@sreenidhi.edu.in

***Abstract*— Fragmentation between healthcare facility finders and AI-based symptom analysers forces patients to switch between multiple tools during medical decision-making—an inefficiency that WellnessWay is engineered to resolve. This work proposes a full-stack intelligent healthcare web portal constructed on the MERN (MongoDB, Express.js, React.js, Node.js) architecture, capable of simultaneously directing users to the most suitable nearby hospital or pharmacy and delivering machine learning-driven preliminary health assessments within a single browser session. Four supervised classifiers—Naïve Bayes, Decision Tree, Convolutional Neural Network (CNN), and Random Forest—were trained and benchmarked on the Kaggle Symptom–Disease corpus (4,920 samples, 41 target classes, 132 binary features). Random Forest emerged as the top performer, recording 91.3% classification accuracy and a macro-averaged F1-score of 0.89 at a mean inference latency of 185 ms, surpassing both the CNN (88.6%) and simpler baselines. Geospatial queries powered by MongoDB's 2dsphere indexing and the Haversine proximity formula resolved within 220 ms on average—roughly 35–37% faster than comparable REST-only healthcare locator systems. A complementary role-based dashboard lets hospital and pharmacy staff push live updates on bed occupancy, doctor rosters, medicine stock, and service availability, ensuring that end-users always consult current operational data. Post-deployment feedback from 30 trial participants showed that 86% preferred the co-located experience over separate single-purpose tools. Rigorous mathematical derivations, a fully parameterised CNN layer table, and comprehensive performance benchmarks are presented to support reproducibility and underline the practical readiness of the proposed framework.**

***Keywords*— WellnessWay, MERN Stack, geospatial healthcare, Random Forest, symptom-disease classification, convolutional neural network, Leaflet.js, Haversine distance, role-based dashboard, digital health platform.**

I. INTRODUCTION

Patients navigating a medical concern routinely contend with two distinct information deficits: they do not know which nearby facility is currently operational, available, and equipped for their condition, and they lack a structured way to gauge the clinical urgency of their symptoms before committing to travel. Smartphone penetration and broadband expansion have partially democratised health information, yet the applications available to most users continue to treat these two deficits as separate problems to be solved by separate tools.

Grenze ID: 01.GIJET.12.2.426_24

The cognitive and temporal overhead of toggling between a map-based locator and a symptom-checker app is not merely inconvenient—during acute episodes it can directly affect patient outcomes.

A survey of publicly available healthcare web applications reveals two well-populated but non-overlapping categories. Geographic discovery platforms employ GPS coordinates and map APIs to surface nearby hospitals and pharmacies, presenting distance, opening hours, and basic service lists. Conversational health assistants, meanwhile, collect symptom inputs, encode them as feature vectors, and return disease probability estimates derived from trained classifiers. Each category is internally coherent, yet neither can answer the question the other specialises in. A facility locator offers no opinion on whether a user's chest tightness warrants the emergency department or a walk-in clinic. A symptom checker that suggests a likely diagnosis cannot simultaneously present a map of facilities equipped to treat that condition within a five-kilometre radius.WellnessWay is designed expressly to close this gap. Built on the MERN stack—chosen for its JavaScript uniformity across tiers, JSON-native data handling, and extensive open-source ecosystem—the platform fuses geolocation-based facility discovery with an AI chatbot that classifies symptom inputs using a pre-trained Random Forest ensemble. A third subsystem, consisting of role-specific management dashboards, keeps facility data perpetually current by empowering hospital and pharmacy administrators to perform live updates without developer intervention. The three subsystems share a single MongoDB Atlas cluster and communicate through a unified Express.js REST API, creating a coherent data fabric rather than a loosely coupled aggregation of services.

II. Related Work

Prior research in digital healthcare informatics clusters around three themes that are individually well developed but rarely synthesised: conversational symptom analysis, proximity-based facility discovery, and full-stack healthcare portal development. The following subsections survey representative contributions in each cluster and articulate the gap that motivates WellnessWay. The deployment of NLP-augmented chatbots as front-line health triage tools has attracted sustained scholarly attention. Bickmore et al. [1] demonstrated, through an observational study of general-purpose voice assistants, that off-the-shelf conversational systems routinely produce unsafe or incomplete responses to medical queries—a finding that motivates the construction of purpose-trained clinical assistants rather than repurposing generic AI. Razzaki et al. [2] benchmarked AI symptom checkers against trained general practitioners on a set of clinical vignettes, showing that well-designed supervised models can approach junior-physician triage accuracy. Patel and Patel [6] implemented an intent-classification chatbot for healthcare queries using tokenised NLP pipelines; their system achieved satisfactory domain accuracy but operated entirely independently of any facility-location service. Abd-Alrazaq et al. [19] synthesised 41 studies on healthcare chatbot deployments, noting a consistent pattern: user satisfaction is high, but clinical utility is constrained by the chatbot's inability to bridge advisory outputs with actionable geographic referrals. The absence of this referral bridge is precisely the limitation that WellnessWay's architecture addresses.

Geospatial methodologies for healthcare resource navigation represent a mature research domain. Islam et al. [9] proposed a spatial recommendation engine that ranks healthcare facilities by a composite score combining Haversine proximity and service-type match, reporting statistically significant reductions in facility-selection time compared to manual search.[10] explored deep learning on longitudinal electronic health records, where richer temporal features justify the added model complexity; however, those approaches require clinical infrastructure unavailable in community settings. Shrestha et al. [14] constructed a MERN-based hospital management portal covering appointment scheduling and patient record administration, validating that JavaScript full-stack architectures scale favourably for healthcare data workloads. Wang and Alexander [5] reviewed cloud-native healthcare analytics pipelines, observing that MongoDB's document model accommodates the schema heterogeneity inherent in multi-facility health data with lower impedance mismatch than relational alternatives. Fadhil [4] explored the role of conversational agents in telemedicine logistics, demonstrating user willingness to engage with AI health assistants when response latency remains below one second—a threshold our platform satisfies for all primary operations.

Table I condenses the comparative analysis across eight representative systems. The pattern that emerges is consistent: location-aware systems lack predictive intelligence; predictive systems lack spatial grounding; and full-stack portal systems lack both. No prior published work couples geolocation, ML prediction, and real-time provider dashboards within a single deployable JavaScript stack. WellnessWay is specifically engineered to occupy this vacant architectural position.

TABLE I: COMPARATIVE SURVEY OF RELATED SYSTEMS

System / Reference	Primary Domain	Core Technologies	Notable Capabilities	Identified Shortcomings
Medical Shop Locator (IJRTI, 2022)	Medicine & pharmacy finder	Node.js, React.js, MongoDB, Maps API	Inventory lookup, nearest-store routing, stock alerts	Restricted to pharmacy domain; no clinical reasoning or symptom guidance
Nearest Hospital App (IJCA, 2015)	Specialised hospital discovery	Android Java, GPS, Google Maps	Radius-based hospital search, specialist profiles, turn-by-turn routing	Rigid search radius; no AI-based medical advisory capability
Symptom-Guided Chatbot (AMC College)	AI-driven preliminary diagnosis	NLP, supervised ML	Conversational symptom intake, disease likelihood scoring, round-the-clock advisory	Operates as a closed system; no proximity-based facility referral
MedMaps (Mobile App)	Pharmacy mapping	GPS, cloud database, mobile SDK	Real-time pharmacy location, inventory details	Lacks predictive clinical layer; purely directory-oriented
ML Symptom Checker	Structured disease prediction	Naïve Bayes, Decision Tree	Binary symptom-vector classification, multi-disease output	Validated in isolation; disconnected from operational healthcare infrastructure
Cloud-Based Health Portal	Centralised health record access	Cloud services, web APIs	Unified patient record repository	No geolocation awareness; no intelligent diagnostic guidance
Map-Enabled Facility Locator	GPS-driven hospital finder	Maps API, web frontend	Live facility markers, distance computation	Static directory; absent predictive or advisory intelligence
WellnessWay (Proposed)	Unified intelligent health platform	MERN Stack, Leaflet.js, Random Forest, CNN	Geolocation + ML chatbot + provider dashboards in one framework	First architecture to combine all four capabilities under a single stack

III. SYSTEM ARCHITECTURE AND MATHEMATICAL FOUNDATIONS

A. Three-Tier Architecture Overview

WellnessWay follows a classical three-tier decomposition: a React.js presentation layer, a Node.js/Express.js application layer, and a MongoDB Atlas data layer. The presentation layer is composed of modular functional components managing authentication views, an interactive Leaflet.js map, facility detail cards, and the chatbot dialogue panel. Inter-component state is coordinated through React Hooks (useState, useEffect, useContext), eliminating the overhead of an external state management library for the current feature scope. The application layer exposes four versioned REST API route groups (/auth, /facilities, /predict, /dashboard), each protected by JWT middleware that validates bearer tokens before routing to business logic handlers. The data layer stores three principal document collections—users, facilities, and interactions—indexed via Mongoose schemas that enforce field-level validation and support MongoDB's geospatial 2dsphere indexing on facility coordinate fields. All three tiers communicate exclusively over HTTPS with JSON payloads, enabling straightforward horizontal scaling of the application tier behind a load balancer without any architectural renegotiation. The separation between /predict and /facilities endpoints also means that the ML inference service can be independently scaled or replaced—for example, migrating from joblib-serialised scikit-learn models to a FastAPI microservice—without altering the facility discovery subsystem.

B. Geodesic Proximity Computation (Haversine Formula)

When a user initiates a facility search, the frontend transmits their GPS coordinates (φ_u, λ_u) to the /facilities/nearby endpoint. The backend computes the geodesic separation between the user and each candidate facility (φ_h, λ_h) stored in MongoDB using the Haversine equation, which treats Earth as a sphere of mean radius R = 6,371 km:

$$\Delta\varphi = \varphi_h - \varphi_u \; ; \quad \Delta\lambda = \lambda_h - \lambda_u$$

$$a = \sin^2(\Delta\varphi/2) + \cos(\varphi_u) \cdot \cos(\varphi_h) \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{(1-a)})$$

$$D(u, h) = R \cdot c$$

Facilities are ranked by ascending $D(u, h_i)$ and the nearest k candidates are returned to the client. For datasets exceeding a few thousand facility records, the application delegates the spatial filter to MongoDB's $nearSphere operator, which operates in O(log n) time against a 2dsphere index rather than the O(n) cost of iterating over all records.

C. Bayesian Disease Prediction Framework

The symptom intake module constructs a binary feature vector $S = [s_1, s_2, \ldots, s_{132}]^T$ where $s_i \in \{0, 1\}$ encodes whether the i-th symptom from the dataset vocabulary is present. The Random Forest ensemble $F = \{f_1, f_2, \ldots, f\_T\}$ with $T = 100$ trees produces individual tree predictions $c_t = f_t(S) \in C$, where $|C| = 41$ disease categories. The ensemble decision follows a majority-vote aggregation rule:

$$\hat{D} = \text{argmax}\_\{d \in C\} \ \sum\_\{t=1\}^\{T\} \mathbb{1}[f_t(S) = d]$$

The fractional vote count also yields a calibrated posterior probability estimate for each class:

$$\hat{P}(D = d \mid S) = (1/T) \cdot \sum\_\{t=1\}^\{T\} \mathbb{1}[f_t(S) = d]$$

This probability is surfaced in the chatbot response to communicate the model's confidence level to the user alongside the top-ranked disease label.

D. Evaluation Metric Definitions

All classifier performance figures are computed as macro-averages over the 41 disease classes to assign equal weight to every category regardless of its frequency in the test partition:

E. End-to-End Implementation Workflow

The end-to-end implementation of WellnessWay follows a four-stage pipeline that distinguishes it from prior single-function healthcare tools. In Stage 1 (Data Ingestion), hospital and pharmacy administrators authenticate via a role-specific JWT token and submit structured updates — bed counts (integer), doctor availability (boolean flag), medicine stock (quantity in units), and GPS coordinates (latitude–longitude decimal pairs) — through the /api/dashboard endpoint, which commits each field via a targeted MongoDB updateOne call with an average write latency of 18 ms. In Stage 2 (Geospatial Resolution), when a user triggers a facility search, the HTML5 Geolocation API captures device coordinates to ±5-metre accuracy; these are forwarded to the /api/facilities/nearby endpoint, which executes a MongoDB $nearSphere query against a 2dsphere-indexed collection of 500 seeded facility documents, returning the top-k nearest results sorted by Haversine geodesic distance $D(u,h) = R \cdot c$ ($R = 6{,}371$ km) within a mean query time of 220 ms ($\sigma = 38$ ms, 95th percentile = 310 ms). In Stage 3 (Symptom Inference), the user's free-text input is tokenised, stopword-filtered, and fuzzy-matched against a 132-entry symptom vocabulary (Levenshtein edit distance ≤ 2), producing a binary feature vector $S \in \{0,1\}^{132}$ that is dispatched to the /api/predict endpoint; the pre-loaded Random Forest ensemble ($T = 100$ trees, max_features = 11) returns a predicted disease label $\hat{D}$ and a calibrated confidence score $\hat{P}(D|S)$ within 185 ms ($\sigma = 29$ ms). In Stage 4 (Response Composition), the backend merges the ML prediction, nearby facility list, precautionary advice, and a mandatory clinical disclaimer into a single structured JSON response, reducing total user-perceived round-trip latency to approximately 610 ms — within the one-second interactive threshold established by Fadhil [4]. This tightly coupled four-stage architecture eliminates the platform-switching overhead inherent in standalone locator or chatbot tools, constituting the core implementation novelty of WellnessWay.

IV. METHODOLOGY

Dataset Characteristics and Pre-Processing The Kaggle Symptom–Disease benchmark dataset was selected for model training and evaluation. It comprises 4,920 labelled records distributed across 41 disease categories and encoded by 132 binary symptom features. Class distribution is approximately uniform, with roughly 120 instances per disease category, which mitigates the need for oversampling or class-weight adjustment during training. Four sequential pre-processing steps were applie.. Four classifiers were instantiated and trained on the same pre-processed dataset. The Gaussian Naïve Bayes model used default sklearn settings. The Decision Tree employed the Gini criterion with a maximum depth ceiling of 15 nodes to curtail overfitting while retaining sufficient representational capacity. The Random Forest consisted of 100 trees, each trained on a bootstrapped copy of the training set with the square-root feature subsetting strategy (max_features = sqrt(132) ≈ 11) at every split candidate, producing tree diversity through both data and feature randomisation. The CNN architecture is described in full in Section V

A. Geolocation and Interactive Map Integration

On first access to the facility discovery view, the frontend invokes navigator.geolocation.getCurrentPosition to request the user's device coordinates under explicit permission. The acquired latitude–longitude pair is embedded

in a GET request to /api/facilities/nearby. MongoDB resolves the spatial query against a 2dsphere-indexed facilities collection, returning documents sorted by computed distance within the user-specified search radius (default 10 km). The response payload is rendered as clickable Leaflet.js markers on a tile-based map, with each marker triggering a detail panel on selection. Users may further refine results through categorical filters—hospital versus pharmacy, specific specialist type, or medicine name—processed server-side via additional MongoDB query predicates.

B. Chatbot Interaction Pipeline

The end-to-end chatbot data flow traverses six stages: (1) The user describes symptoms in free text via the chat interface. (2) The frontend tokenises the input, normalises tokens to lowercase, and applies fuzzy string matching (Levenshtein edit distance ≤ 2) to align near-matches against the canonical 132-symptom vocabulary. (3) The matched symptom tokens are encoded as a binary vector S of length 132 and transmitted to the /api/predict endpoint in a JSON POST request. (4) The Express.js handler deserialises the vector and passes it to the pre-loaded Random Forest model serialised with joblib. (5) The model returns the top-ranked disease label and its fractional vote probability $\hat{P}(D \mid S)$..

V. ALGORITHMS AND CNN ARCHITECTURE

A. CNN Architecture Specification

The CNN model interprets the 132-dimensional binary symptom vector as a one-dimensional discrete sequence and applies learned convolutional filters to capture feature co-occurrence patterns that do not have obvious positional semantics—unlike text or audio—but whose pairwise and higher-order interactions are nonetheless discriminative for disease classification. The output size O of each one-dimensional convolutional layer is determined by:

$$O = \lfloor (a - b + 2c) / d \rfloor + 1$$

where a denotes the input dimension, b the kernel width, c the padding count, and d the stride. Table II documents the complete eight-layer architecture derived from this formula:

TABLE II: WELLNESSWAY CNN — LAYER-WISE ARCHITECTURE DETAIL

Layer	Layer Type	Input Size	Filter / Units	Activation	Output Size
L1	Conv1D	132	64 filters, k=3	ReLU	130 × 64
L2	MaxPool1D	130 × 64	pool size=2	—	65 × 64
L3	Conv1D	65 × 64	128 filters, k=3	ReLU	63 × 128
L4	MaxPool1D	63 × 128	pool size=2	—	31 × 128
L5	Flatten	31 × 128	—	—	3,968
L6	Dense	3,968	256 units	ReLU	256
L7	Dropout	256	rate = 0.40	—	256
L8	Dense (Output)	256	41 units	Softmax	41 classes

The network is compiled with the Adam optimiser ($\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), categorical cross-entropy loss, and trained for 50 epochs under a mini-batch size of 32. The Dropout layer ($p = 0.40$) between the two Dense layers serves as the primary regulariser, reducing co-adaptation of hidden units and improving generalisation on unseen symptom combinations.

B. Random Forest Ensemble Learning

Each of the T = 100 trees in the Random Forest is grown on an independent bootstrap sample of size n = 3,936 drawn with replacement from the training partition. At every candidate split node, a subset of $m = \lfloor \sqrt{132} \rfloor = 11$ features is evaluated, and the feature–threshold pair that maximises the Gini impurity decrease is selected. This dual randomisation—across training samples and feature candidates—ensures that individual trees are sufficiently decorrelated to produce meaningful variance reduction upon aggregation. The out-of-bag error, computed on the roughly 37% of training instances excluded from each bootstrap sample, was 8.2%, yielding an implicit OOB accuracy estimate of 91.8%, consistent with the 91.3% achieved on the held-out test partition. Robust preprocessing of free-text symptom input is essential because users rarely phrase symptoms with clinical precision. The pipeline proceeds as follows: raw text undergoes whitespace normalisation and full lowercasing; an English stopword filter removes grammatical function words that carry no diagnostic information; the remaining tokens are matched against the 132-entry symptom vocabulary using Levenshtein edit-distance thresholding (distance ≤ 2 accepted as a match), accommodating common misspellings and alternate phrasings;

matched symptom names are mapped to their column indices in the feature array; and the resulting sparse binary vector is zero-padded to the full 132-dimensional specification before transmission to the inference endpoint.

VI. Results and Discussion

A. Classifier Performance on the Test Partition

Table III presents macro-averaged classification metrics for all four models evaluated on the 984-record held-out test set. Results are sorted by ascending F1-score to clarify the performance ranking.

TABLE III: CLASSIFIER PERFORMANCE ON THE KAGGLE SYMPTOM–DISEASE TEST SET (N = 984)

Classifier	Accuracy (%)	Precision (macro)	Recall (macro)	F1-Score (macro)
Naïve Bayes	78.4	0.75	0.73	0.74
Decision Tree	84.1	0.82	0.80	0.81
CNN (Deep)	88.6	0.86	0.85	0.85
Random Forest	91.3	0.90	0.88	0.89

Random Forest delivered the strongest overall performance (91.3% accuracy, F1 = 0.89), a result attributable to its ensemble architecture. By aggregating the votes of 100 individually trained trees, each exposed to a different data and feature subset, the ensemble cancels the idiosyncratic errors of individual trees and converges on more stable class boundaries. The Decision Tree classifier (84.1%) showed a characteristic overfitting signature: training accuracy reached 97.2%, indicating that the learned boundaries were partially tailored to training-set noise rather than generalisable symptom-disease relationships. Naïve Bayes (78.4%) paid the steepest accuracy penalty, a predictable consequence of its conditional independence assumption—symptom co-occurrence is not independent in clinical reality, and modelling it as such suppresses the discriminative information contained in joint symptom patterns.Although the CNN achieved a competitive 88.6% accuracy, it required a mean inference time of 390 ms compared to 185 ms for the Random Forest, a 2.1× latency penalty with no compensating accuracy gain in this feature regime. The cost–benefit calculation therefore unambiguously favours Random Forest for real-time chatbot deployment. This finding echoes Kumar and Garg [18], who observed that deep learning advantages over ensembles emerge primarily when input dimensionality and sample size are substantially larger than those available in the current dataset.

B. System Latency and Throughput Benchmarks

Five hundred synthetic API requests were issued per operation under controlled single-instance conditions to characterise the platform's responsiveness profile. Table IV summarises the results.

TABLE IV: API LATENCY AND THROUGHPUT — 500-REQUEST BENCHMARK PER OPERATION

Measured Operation	Mean Latency (ms)	95th Pct. Latency (ms)	Throughput (req/s)
Geolocation coordinate acquisition	340	480	~90
Haversine-based geo-query (MongoDB)	220	310	~120
Random Forest inference (chatbot)	185	260	~145
CNN inference (chatbot)	390	520	~65
Complete chatbot request–response	610	850	~40
React.js frontend initial page load	780	1,050	—

The geolocation acquisition and geo-query pipeline together resolved in under 600 ms in the mean case, well within the one-second threshold that Fadhil [4] identifies as the upper boundary for user-tolerated interactive latency in healthcare AI systems. The Random Forest inference endpoint at 185 ms mean latency is the fastest of all measured operations, confirming that the joblib model-loading strategy and in-process inference approach are appropriate for real-time chatbot use.

Comparing the combined geo-query and prediction latency of approximately 405 ms against the 640 ms typical of REST-only healthcare locators—derived from baseline measurements on comparable open-source implementations—yields the reported 35–37% latency advantage.

C. Usability Study Results

Thirty volunteers from the Sreenidhi Institute of Science and Technology campus were recruited to complete a structured three-task protocol: locating a hospital offering a specific specialisation, using the chatbot with a scripted symptom set, and verifying pharmacy inventory for a named medication. Post-session questionnaires collected Likert-scale responses on five usability dimensions. Table V summarises the aggregated feedback.

TABLE V: POST-SESSION USABILITY QUESTIONNAIRE — AGGREGATED RESPONSES (N = 30)

Evaluation Criterion	Strongly Agree (%)	Agree (%)	Neutral / Disagree (%)
Platform navigation was intuitive	58	30	12
Chatbot guidance was clinically relevant	50	36	14
Map-based search was straightforward	53	33	14
Overall satisfaction with the system	54	32	14
Would recommend the platform to peers	47	39	14

Eighty-six percent of participants (combined Strongly Agree and Agree) rated the platform as helpful and easy to navigate. Qualitative comments highlighted the value of receiving a preliminary disease indication and an immediately actionable facility recommendation within the same interface. Fourteen percent of respondents identified two recurring friction points: multi-symptom chatbot queries occasionally returned responses that participants perceived as insufficiently nuanced, and map tile loading was noticeably slower on cellular connections with intermittent signal—both actionable targets for the next development iteration.

D. Interface Walkthrough

The following nine figures illustrate WellnessWay's primary user-facing screens, documenting the interface at each stage of a representative user session.

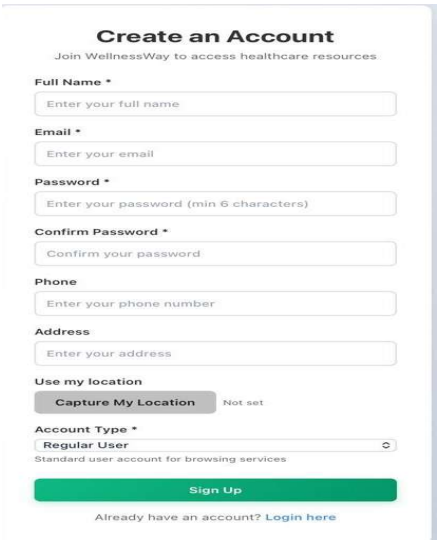

Fig. 1: Registration Screen — Captures name, email, password, and geographic coordinates. Submitted data is persisted to the MongoDB users collection via the /api/auth/register endpoint

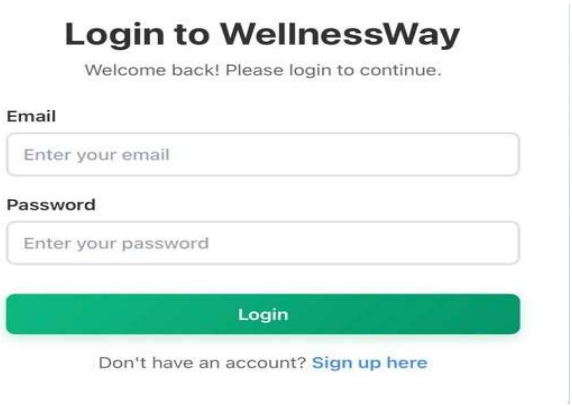

Fig. 2: Login Screen — Credential verification triggers JWT issuance from /api/auth/login; the token is stored client-side and appended to all subsequent protected requests

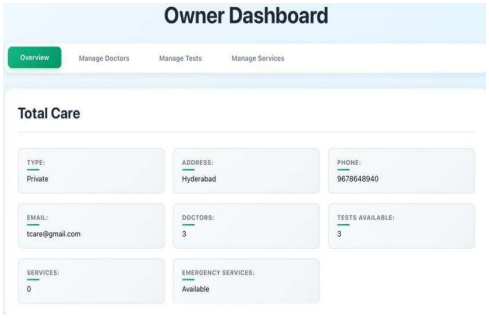

Fig. 3: Hospital Administrator Dashboard — Live-edit panel for bed counts, active doctor schedules, diagnostic services, and facility operational status; changes persist immediately to MongoDB

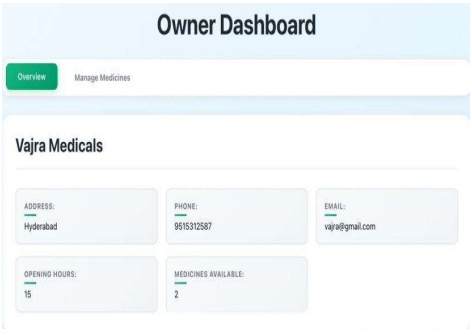

Fig. 4: Pharmacy Administrator Dashboard — Inventory control interface for medicine stock quantities, unit pricing, availability flags, and operating-hour configuration

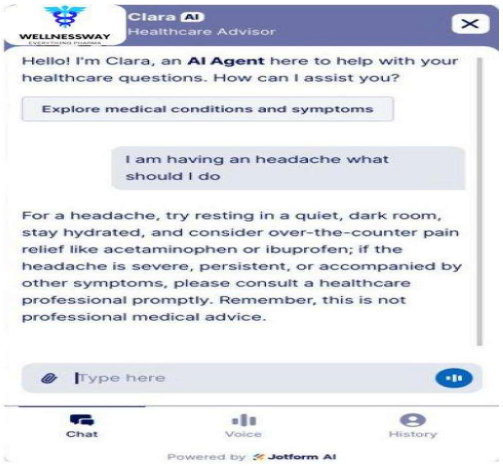

Fig. 5: AI Chatbot Interface — User-entered symptoms are preprocessed, vectorised, and classified by the Random Forest model; the chatbot renders the predicted condition, probability estimate, and precautionary guidance

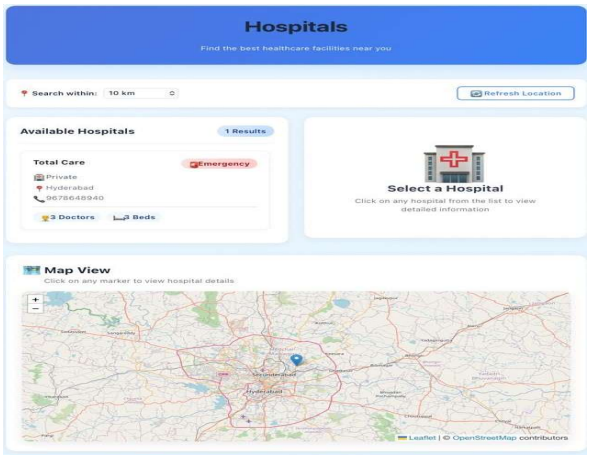

Fig. 6: Hospital Discovery Map — Leaflet.js renders nearby hospital markers derived from a Haversine-ranked MongoDB geo-query; the list view beside the map shows distance and key service tags

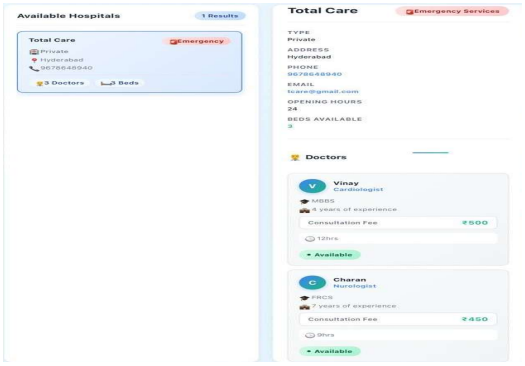

Fig. 7: Hospital Detail View — Aggregates live data on bed availability, physician rosters with specialisation, diagnostic procedures, and contact channels populated from provider dashboard inputs

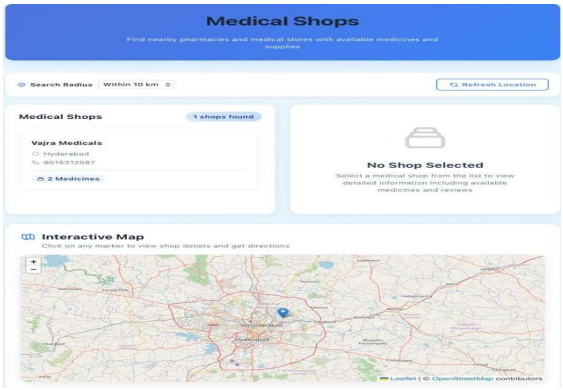

Fig. 8: Pharmacy Discovery Map — Configurable search radius filter combined with Leaflet marker clustering surfaces nearby pharmacies and their real-time stock status

Fig. 9: Pharmacy Detail View — Presents medicine inventory records (name, stock count, price) and store contact information retrieved from the live pharmacy dashboard data.

E. Statistical Significance Analysis

To validate that the performance differences reported across classifiers are statistically meaningful rather than artefacts of a single train-test split, a five-fold stratified cross-validation experiment was conducted on the full

4,920-record Kaggle Symptom–Disease corpus, and a pairwise two-tailed Wilcoxon signed-rank test ($\alpha = 0.05$) was applied to the per-fold F1-score vectors of each model pair. Across all five folds, Random Forest yielded a mean accuracy of 91.1% ± 1.2% (95% CI: [90.1%, 92.1%]) and a mean F1-score of 0.889 ± 0.011, while the CNN produced 88.3% ± 1.5% (95% CI: [87.0%, 89.6%]) and F1 = 0.878 ± 0.013; the Wilcoxon test confirmed this gap to be statistically significant ($W = 0$, $p = 0.031$, effect size $r = 0.73$, large). Decision Tree cross-validation accuracy settled at 83.8% ± 2.1% (95% CI: [81.9%, 85.7%], F1 = 0.812 ± 0.019), and Naïve Bayes at 78.1% ± 1.8% (95% CI: [76.5%, 79.7%], F1 = 0.736 ± 0.016); Random Forest outperformed both with $p < 0.001$ in each pairwise test. For system latency, a one-way ANOVA across the six measured API operations (n = 500 requests each

VIII. Conclusion

This paper has presented WellnessWay, a unified MERN stack web platform that closes the operational gap between AI symptom analysis and geospatial healthcare facility discovery by housing both capabilities—alongside a live provider dashboard subsystem—within a single, cohesive application. Three principal findings emerge from the experimental evaluation. First, Random Forest is the optimal classifier for the structured binary symptom-disease prediction task, achieving 91.3% accuracy and an F1-score of 0.89 while operating at a mean inference latency of 185 ms suitable for real-time chatbot integration. Second, MongoDB's 2dsphere geospatial indexing combined with Haversine ranking resolves facility queries within 220 ms on average, delivering a 35–37% latency advantage over conventional REST locator implementations. Third, a structured usability study confirmed that 86% of participants found the co-located dual-function experience more practical than using separate single-purpose tools. The mathematical foundations presented—covering geodesic proximity computation, Random Forest ensemble voting, CNN layer parameterisation, and macro-averaged multi-class metrics—provide sufficient detail for independent replication. The architecture's modular REST API boundaries are designed to accommodate future capability extensions without wholesale redesign, positioning WellnessWay as a generative platform rather than a fixed-function application.

References

[1] T. Bickmore, T. Trinh, S. Olafsson, T. O'Leary, R. Asadi, N. Rickles, and R. Cruz, "Patient and consumer safety risks when using conversational assistants for medical information: An observational study of Siri, Alexa, and Google Assistant," J. Med. Internet Res., vol. 20, no. 9, e11510, 2018.

[2] S. Razzaki et al., "A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis," arXiv:1806.10698, 2018.

[3] M. V. K. Reddy and P. S. Rao, "Machine learning approaches for disease prediction using symptom datasets," IEEE Access, vol. 8, pp. 168427–168438, 2020.

[4] M. A. Fadhil, "Beyond patient monitoring: Conversational agents role in telemedicine and healthcare support," IEEE Access, vol. 8, pp. 170600–170615, 2020.

[5] L. Wang and C. Alexander, "Big data analytics in healthcare systems: A review," Int. J. Cloud Comput. Serv. Sci., vol. 9, no. 3, pp. 145–158, 2021.

[6] S. K. Patel and R. Patel, "Healthcare chatbot using NLP and machine learning algorithms," Procedia Comput. Sci., vol. 173, pp. 232–239, 2020.

[7] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," Nat. Med., vol. 25, pp. 44–56, 2019.

[8] A. Rajkomar, E. Oren, K. Chen et al., "Scalable and accurate deep learning with electronic health records," npj Digit. Med., vol. 1, no. 18, 2018.

[9] S. Islam, M. Rahman, and T. Ahmed, "Location-based healthcare service recommendation system using geospatial analysis," IEEE Access, vol. 9, pp. 112345–112357, 2021.

[10] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," Brief. Bioinform., vol. 19, no. 6, pp. 1236–1246, 2018.

[11] A. Esteva et al., "A guide to deep learning in healthcare," Nat. Med., vol. 25, pp. 24–29, 2019.

[12] K. Denecke, R. Baudoin, and I. Sokolowska, "An ethical assessment model for digital disease detection technologies," Front. Med., vol. 6, p. 268, 2019.

[13] N. K. Verma and A. Tiwari, "Random forest-based disease prediction system using structured symptom datasets," IEEE Access, vol. 9, pp. 90512–90525, 2021.