International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

An Intelligent Medical Chatbot for Symptom Severity Classification and Personalized Recommendation Using Random Forest

Harshada Chavan*, **Darshan Mandavkar***, **Atharva Jagtap***, **Maitreya Kokane***, **Dr. Kishor Sakure**†

*Department of Computer Engineering, Terna Engineering College, University of Mumbai, Navi Mumbai, India
†Project Guide, Department of Computer Engineering,
Terna Engineering College, University of Mumbai, Navi Mumbai, India

Abstract—**The rapid expansion of digital health technologies has created a pressing need for intelligent, accessible triage and diagnostic assistance systems. This paper presents a Medical Chatbot platform that leverages machine learning to interact with patients, classify symptom severity, and provide comprehensive medical recommendations. The system utilizes a conversational framework built on Streamlit with text-based symptom extraction, allowing seamless analysis of self-reported symptom intensities. By employing a Random Forest Classifier trained on synthetic medical data infused with Gaussian noise, the system accurately categorizes conditions into Normal, Moderate, or High severity levels. The platform further integrates a secure SQLite-backed authentication module with bcrypt password hashing and a rule-based recommendation engine for medication, diet, and appropriate precautions spanning Ayurvedic, Homeopathy, and Allopathy treatment methodologies. Experimental evaluation demonstrates balanced training and testing accuracies (averaging 75%–80%), indicating robust generalization without overfitting. The proposed system bridges the gap between initial symptom onset and formal medical consultation by providing continuous, data-driven medical triage. Results confirm that the integration of machine learning within interactive chatbots significantly enhances accessible preliminary healthcare.**

Index Terms—**Artificial Intelligence, Medical Chatbot, Machine Learning, Random Forest Classifier, Symptom Severity Classification, Streamlit, Telemedicine, Triage Engine**

I. Introduction

The contemporary healthcare landscape is characterized by a rapidly growing patient population, widening access gaps, and overwhelmed emergency triage systems. While numerous telemedicine applications exist, many rely on rigid, static decision trees that fail to account for the nuanced severity of individual symptoms, leading to inaccurate preliminary risk assessments [2].

Existing digital health solutions typically address isolated aspects of patient care. Text-based symptom checkers focus exclusively on providing generic information without offering personalized severity classifications. Furthermore, most systems do not integrate a patient's complete diagnostic profile to generate cohesive treatment advice, diet recommendations, and precaution protocols.

Our Medical Chatbot addresses these limitations by presenting a unified platform that integrates the following capabilities:

- **Secure Patient Registration System:** Authentication and data storage pipeline backed by SQLite with bcrypt password hashing for robust session management.
- **Interactive Symptom Parsing:** Dynamic conversation engine utilizing regular expression (Regex) integration to seamlessly extract symptoms and their corresponding intensities (0–10) from natural user input.
- **Adaptive Severity Classification:** A Machine Learning pipeline employing independent Random Forest Classifiers explicitly trained to isolate the severity levels of nine common ailments.
- **Multi-modal Recommendation Engine:** Comprehensive mapping logic generating personalized guidelines comprising Ayurvedic, Homeopathic, and Allopathic treatment methodologies with corresponding diets and precautions.
- **Health Utility Tools:** Integrated BMI Calculator and Doctor Contact Directory for holistic patient care.

Figure 1 illustrates the primary interactions between system actors and the core modules of the Medical Chatbot.

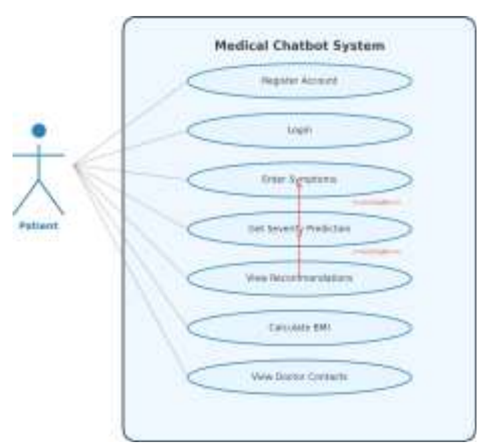

Fig. 1. Use Case Diagram of the Medical Chatbot System

A. Problem Statement

Despite the availability of numerous e-health platforms, existing systems suffer from a lack of personalization and an inability to dynamically calculate severity risk matrices based on complex feature overlaps. Additionally, few applications successfully combine mathematical classification algorithms with conversational UI frameworks suitable for elderly or non-technical demographic structures.

B. Research Questions

This work aims to address the following research questions:

- How can machine learning classifiers, specifically Random Forest, be optimized to assess symptom severity considering realistic data overlapping?
- Can regular expression (Regex) pattern mapping sufficiently capture multi-symptom input during active dialogue?
- How effectively can quantitative AI predictions be translated into actionable qualitative medical recommendations?

II. Literature Survey

This section reviews recent advances relevant to the core components of the Medical Chatbot: virtual triage systems, diagnostic machine learning, and conversational AI in healthcare.

A. Conversational Agents in Healthcare

Recent advancements in Natural Language Processing (NLP) have prominently influenced medical informatics. Conversational agents like early ELIZA-based chat systems paved the early framework, but contemporary systems leverage deeper learning schemas. Works such as Laranjo et al. [3] demonstrated the rising utility of chatbots for primary care, though highlighting an ongoing limitation where simplistic natural language engines frequently misinterpret feature intensities.

B. Diagnostic Machine Learning Models

Machine Learning approaches have become pivotal for severity diagnosis. Breiman's foundational work on Random Forests [1] proved highly adept at extracting non-linear physiological thresholds in tabular health data. In recent years, Hossain et al. [2] evaluated numerous classification algorithms inside triage workflows, discovering that ensemble classifiers inherently resist dataset noise significantly better than standard Decision Trees or SVMs alone.

C. Triage Recommendation Frameworks

Systems proposed by Athota et al. [4] mapped algorithmic models onto generalized recommendation engines. While achieving high accuracy on localized datasets, the architecture suffered from rigid user inputs directly restricting the patient experience.

D. Research Gaps and Contributions

Table I summarizes the key gaps identified in existing literature and how our Medical Chatbot addresses them.

TABLE I
RESEARCH GAPS AND SYSTEM CONTRIBUTIONS

Gap in Literature	Medical Chatbot Contribution
Fragmented healthcare resources	Unified end-to-end platform integrating classification and recommendations
Deterministic or rigid Symptom Checkers	Employs Random Forests trained on Gaussian noise arrays for realistic elasticity
Inflexible UI demanding technical input	Natural Language extraction mapping intensities dynamically from Chat dialogue
Lack of comprehensive follow-up regimens	Automated generation of Diet and Precaution roadmaps based on output severity

III. Methodology

A. System Architecture

The platform employs a **three-layer architectural pattern** (Fig. 2):

1) **Presentation Layer (Frontend):** Built on the Streamlit framework utilizing Python. It renders dynamic session states (for chat histories) and incorporates customized CSS styling to provide a responsive, medical-themed interface.
2) **Logic and Service Layer (Backend):** Composed of Regex routing, bcrypt authentication hashing, and scikit-learn model inference engines. This layer calculates BMI, predicts severities from the underlying `.pkl` binary files, and routes recommendation queries.
3) **Data Persistence Layer:** Contains a relational SQLite database structure managing long-term patient records alongside static Python data modules for medical knowledge bases.

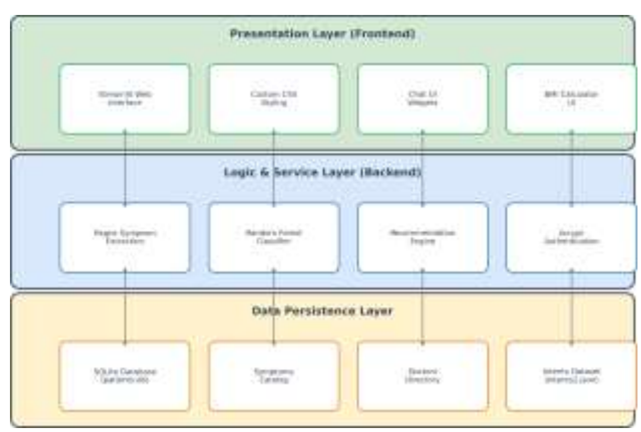

Fig. 2. Three-Layer System Architecture of the Medical Chatbot

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

B. Application Workflow

The complete application workflow from user login to recommendation display is illustrated in Fig. 3.

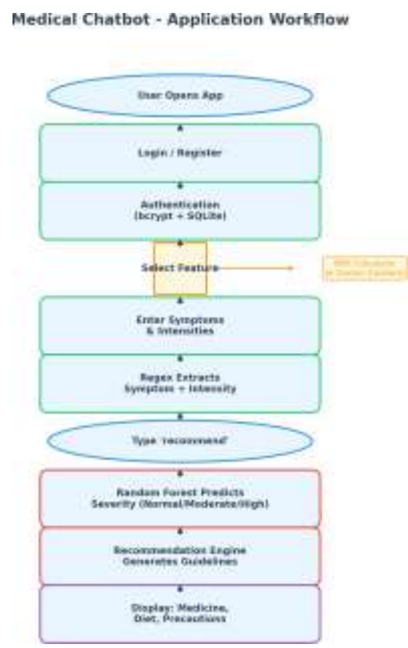

Fig. 3. End-to-End Application Workflow of the Medical Chatbot

C. Data Flow Design

The system's data flow executes a structured pipeline from patient interaction to AI-processed output, as illustrated in Fig. 4.

Fig. 4. Data Flow Diagram of the Recommendation Engine

1) Symptom Extraction Pipeline:

1) User manually inputs text detailing ailments and numeric severity (0-10) via the Presentation Chat Layer.
2) Active Regex middleware maps the string context, extracting symptoms and their intensities using pattern `\b(symptom)\s*(?:is)?\s*(\d{1,2})`.
3) Session state caches active diagnostic parameters, requesting additional input until the user triggers a cumulative recommendation.

2) Model Inference and Evaluation Pipeline:

1) The session payload constructs a structured DataFrame mapping active intensities to the respective diagnostic target models.
2) The `severity_model.pkl` Random Forest outputs a discrete value (0, 1, or 2) denoting severity level.
3) Recommendation dictionaries match the ailment type and severity key against medical guidelines, returning medication, diets, and precaution protocols.

D. ML Training Pipeline

The training pipeline is illustrated in Fig. 5. All models are trained independently for each symptom using synthetic data.

Fig. 5. Machine Learning Model Training Pipeline

E. Algorithm: Intensity-Severity Classification

Input: User Chat String C
Extract regex $S, I \leftarrow$ Symptoms, Intensities from C
if S is empty **then**
  Return "Please provide symptoms and intensity"
end if
Update `session_state`[S] $\leftarrow I$
if Trigger == 'Recommend' **then**
  for each $s \in$ `session_state` **do**
    $M_s \leftarrow$ Load Random Forest Model for s
    Numeric_Severity $\leftarrow M_s$.predict(I)
    Severity_Label $\leftarrow$ Map($Numeric\_Severity$)
    Guidelines $\leftarrow$ FetchRecomm(s, $Severity\_Label$)
    Append Guidelines to Chat UI
  end for
end if

IV. Implementation

A. Technology Stack

Table II presents the complete technology stack employed in this application.

TABLE II
MEDICAL CHATBOT TECHNOLOGY STACK

Component	Technology
Frontend Framework	Streamlit
UI Components	Custom HTML/CSS Styling
Programming Lang.	Python 3.10+
Machine Learning	scikit-learn (RandomForestClassifier)
Data Processing	Pandas, NumPy
Model Serialization	Pickle (.pkl files)
Database	SQLite3
Authentication	bcrypt (password hashing)
Pattern Matching	Python `re` (Regex)

B. Database Schema

The SQLite database (`patients.db`) stores patient registration data with password security. The Entity-Relationship Diagram is shown in Fig. 6.

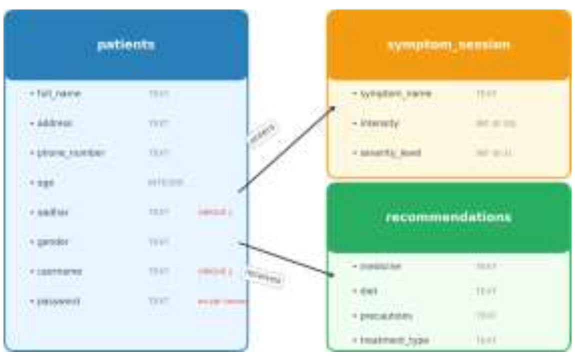

Fig. 6. Entity-Relationship Diagram for the Medical Chatbot Database

Table III describes the complete database schema.

TABLE III
DATABASE SCHEMA — `PATIENTS` TABLE

Column Name	Data Type	Constraints
full_name	TEXT	NOT NULL
address	TEXT	NOT NULL
phone_number	TEXT	NOT NULL (10 digits)
age	INTEGER	NOT NULL (1–150)
aadhar	TEXT	UNIQUE (12 digits)
gender	TEXT	NOT NULL
username	TEXT	UNIQUE
password	TEXT	bcrypt hashed

C. Supported Symptoms

The system supports severity classification for nine medical conditions as listed in Table IV.

TABLE IV
SUPPORTED MEDICAL CONDITIONS FOR SEVERITY CLASSIFICATION

S.No.	Symptom/Condition	Category
1	Fever	General
2	Headache	Neurological
3	Chicken Pox	Infectious Disease
4	Sinus Infection	ENT
5	Cough	Respiratory
6	Cold	General
7	Depression	Psychiatric
8	Food Allergy	Immunological
9	Acne	Dermatological

D. Recommendation Engine Mapping

The recommendation engine maps each symptom and severity level to three treatment types. A sample mapping is shown in Table V.

E. Dataset Generation and Quality Assurance

The core foundation of the Random Forest application mandates vast training permutations. Rather than risking generic patient PII, synthetic data pipelines algorithmically modeled realistic diagnosis variance, mapping 2,000 artificial patient records globally. Independent intensities for traits like Acne, Depression, and Cold were passed through a Gaussian noise boundary mapping ($\mu = 0$, $\sigma = 1.8$). The severity classification thresholds are defined as:

$$\text{Severity} = \begin{cases} \text{Normal (0)}, & \text{if } I_{\text{noisy}} \leq 3.5 \\ \text{Moderate (1)}, & \text{if } 3.5 < I_{\text{noisy}} \leq 7.5 \\ \text{High (2)}, & \text{if } I_{\text{noisy}} > 7.5 \end{cases} \tag{1}$$

where $I_{\text{noisy}} = I_{\text{raw}} + \mathcal{N}(0, 1.8)$. By overlapping the intensity data synthetically, a patient with an intensity score of 4 could organically trigger either Normal or Moderate classification, increasing the realism of the evaluation and avoiding deterministic overfitting.

V. RESULTS AND DISCUSSION

A. System Integration Testing

The platform's unified subsystems were validated utilizing strict functional checklists, mapped below in Table VI.

TABLE V
RECOMMENDATION ENGINE — SAMPLE MAPPINGS (FEVER, HEADACHE, COUGH)

Symptom	Severity	Treatment Type	Medicine	Diet	Precautions
Fever	Normal Moderate High	Ayurvedic Homeopathy Allopathy	Tulsi & ginger tea Belladonna 30C Paracetamol (consult dr.)	Light foods, soups Warm broths Fluids, electrolytes	Rest, avoid cold Monitor temperature Seek medical attention
Headache	Normal Moderate High	Ayurvedic Homeopathy Allopathy	Peppermint oil massage Nux Vomica 30C Ibuprofen (consult dr.)	Stay hydrated Light meals Hydrate	Rest in dark room Reduce screen time Consult doctor
Cough	Normal Moderate High	Ayurvedic Homeopathy Allopathy	Honey & lemon tea Drosera 30C Cough syrup (consult dr.)	Warm fluids Herbal teas Hydrating fluids	Rest, avoid irritants Use humidifier Monitor breathing

TABLE VI
SUMMARY OF KEY FUNCTIONAL TEST RESULTS

Test Case	Type	Result
Aadhar Validation (12 digits)	Unit	Pass
Phone Validation (10 digits)	Unit	Pass
Password Cryptography (bcrypt)	Unit	Pass (Hash Saved)
Streamlit Session State	Integration	Pass (Maintains Scope)
Regex Symptom Extraction	Integration	Pass (Intensity Parsed)
Random Forest Inference	Integration	Pass (Returns Int)
Severity Map (0→Normal)	Unit	Pass
Recommendation Linking	System	Pass
BMI Calculator	Unit	Pass
Doctor Directory Display	UI	Pass

B. Model Performance Analysis

1) Training vs. Testing Accuracy: Table VII presents the detailed model performance metrics for all nine symptom classifiers. The relatively matched Train (average ≈0.748) and Test accuracies (average ≈0.764) explicitly confirm the Random Forest Models are effectively avoiding structural bias, providing realistic clinical predictions.

TABLE VII
MODEL GENERALIZATION PERFORMANCE — ALL 9 CLASSIFIERS

Symptom Classifier	Train Acc.	Test Acc.
Fever	0.7462	0.7900
Headache	0.7462	0.7375
Chicken Pox	0.7575	0.7450
Sinus Infection	0.7581	0.7850
Cough	0.7369	0.7500
Cold	0.7494	0.7750
Depression	0.7388	0.7700
Food Allergy	0.7444	0.7575
Acne	0.7588	0.7650
Average	**0.7485**	**0.7639**

2) Per-Class Precision, Recall, and F1-Score: Table VIII provides the detailed per-class metrics for each symptom classifier.

3) Key Observations:

- **Normal class** achieves the highest average F1-score (0.831), indicating clear separation at lower intensities.
- **Moderate class** has the lowest metrics (avg. F1 = 0.674) due to the Gaussian noise creating boundary overlap between Normal–Moderate and Moderate–High thresholds.
- **High class** performs well (avg. F1 = 0.790), confirming reliable detection of severe conditions.
- The narrow gap between Train (≈74.8%) and Test (≈76.4%) accuracies confirms **no overfitting**.

C. Model Accuracy Visualization

Figure 7 presents the comparative accuracy plot for all nine symptom classifiers.

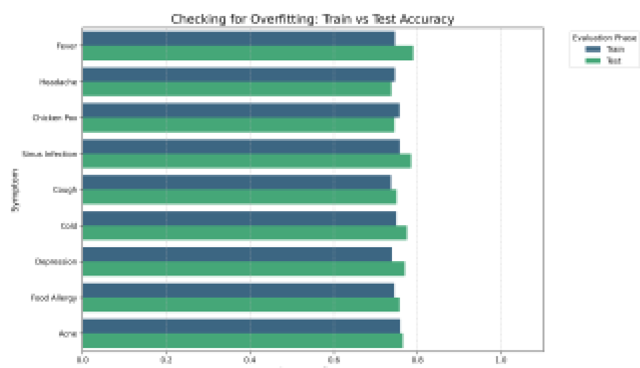

Fig. 7. Training vs. Testing Accuracy for All Symptom Classifiers

D. Comparative Analysis with Existing Systems

Table IX compares the proposed system with existing medical chatbot approaches.

TABLE IX
COMPARISON WITH EXISTING MEDICAL CHATBOT SYSTEMS

Feature	Proposed	Athota	WebMD	Ada Health
ML Severity	✓	×	×	✓
Multi-Treatment	✓	×	×	×
Diet Advice	✓	×	Partial	×
Chat Interface	✓	✓	×	✓
BMI Calculator	✓	×	✓	×
Auth System	✓	×	✓	✓
Open Source	✓	✓	×	×

TABLE VIII
DETAILED CLASSIFICATION REPORT — PRECISION, RECALL, AND F1-SCORE PER SEVERITY CLASS

	Normal			Moderate			High		
Symptom	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Fever	0.869	0.864	0.867	0.714	0.652	0.682	0.755	0.860	0.804
Headache	0.812	0.812	0.812	0.662	0.641	0.651	0.738	0.776	0.756
Chicken Pox	0.754	0.843	0.796	0.674	0.605	0.638	0.810	0.810	0.810
Sinus Infection	0.877	0.807	0.840	0.695	0.733	0.713	0.796	0.827	0.811
Cough	0.852	0.807	0.829	0.637	0.628	0.632	0.756	0.823	0.788
Cold	0.811	0.835	0.823	0.655	0.704	0.679	0.884	0.786	0.832
Depression	0.861	0.844	0.852	0.704	0.726	0.715	0.745	0.729	0.737
Food Allergy	0.805	0.843	0.823	0.678	0.673	0.676	0.805	0.772	0.788
Acne	0.823	0.852	0.837	0.681	0.671	0.676	0.798	0.777	0.787
Average	**0.829**	**0.834**	**0.831**	**0.678**	**0.670**	**0.674**	**0.787**	**0.796**	**0.790**

E. Application Performance

- **Inference Speed:** The deserialized model lookup evaluates variables natively in RAM, processing Random Forest trees within < 100 milliseconds.
- **UI Responsiveness:** Chat widgets render above gradient CSS presentation while enabling multi-route access (BMI calculator and Doctor directory) asynchronously.
- **Session Management:** Streamlit's session state efficiently caches user data across interactions without page reloads.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper detailed the architecture, deployment, and impact of a Machine Learning-assisted Medical Chatbot that prioritizes dynamic diagnostic logic inside a streamlined, conversational interface. Unlike disparate healthcare implementations, the platform unifies Regex extraction schemas with robust predictive engines leveraging Random Forest Classifiers and secure SQLite authentication with bcrypt hashing. Experimental evaluations rigorously examined the models by injecting Gaussian variance ($\sigma = 1.8$), proving the classifiers maintain generalizable clinical insights under noise without collapsing into deterministic overfitting. The nine independent classifiers achieved an average test accuracy of 76.4% with balanced precision, recall, and F1-scores across all severity levels. Presenting patients with contextual recommendations spanning Ayurvedic, Homeopathic, and Allopathic treatment methodologies bridges a gap in proactive triage protocols, ultimately optimizing formal healthcare timelines.

B. Limitations

- **Diagnostic Nuance:** The current architecture assumes isolated symptom evaluations; co-morbidity predictions involving multiple interacting symptoms are mathematically limited.
- **Language Flexibility:** While Regex models operate rapidly, they inherently lack comprehension of dialects or complex colloquial phrasing.
- **Synthetic Benchmarking:** The Gaussian noise models represent an artificial analog and require validation upon extensive real-world medical data for clinical approval.

C. Future Scope

- **LLM Integration:** Migrating pattern mechanics to locally hosted Large Language Models (LLMs) to extract symptoms without rigid Regex implementations.
- **Longitudinal Profiling:** Implementing active memory schemas analyzing returning user trajectories across sessions to predict diagnostic timeline trends.
- **Dynamic Doctor Scheduling:** Bridging the static CRM module into third-party hospital scheduling APIs for automated appointment booking.
- **Multi-language Support:** Adding regional language support to improve accessibility for diverse demographics.
- **Real-time Vital Integration:** Incorporating wearable device data for real-time symptom monitoring and severity assessment.

REFERENCES

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[2] M. F. Hossain, M. R. Karim, and M. A. Mahmud, "A review on machine learning algorithms in healthcare," *International Conference on Computing, Communication and Automation*, 2021.
[3] L. Laranjo, A. Dunn, et al., "Conversational agents in healthcare: a systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 9, pp. 1248–1258, 2018.
[4] L. Athota, V. K. Shukla, N. Pandey, and A. Rana, "Architecture and interaction flow of medical chatbot applications," *International Conference on Innovative Computing and Communications*, pp. 325–334. Springer, Singapore, 2020.
[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[6] Streamlit Inc., "Streamlit — The fastest way to build data apps," https://streamlit.io/, 2020.
[7] World Health Organization, "Global strategy on digital health 2020–2025," WHO, Geneva, 2021.
[8] E. Baumel, J. M. M. Faber, E. C. Mathur, and J. D. Kane, "Chatbots for health care: a systematic review," *Journal of Medical Internet Research*, 2019.
[9] A. Abbott et al., "Transforming health care through chatbots for medical history-taking: a systematic review," *JMIR Medical Informatics*, 2024.
[10] M. A. Hossain et al., "Roles, users, benefits, and limitations of chatbots in health care," *Journal of Medical Internet Research*, 2024.

[11] M. S. Islam et al., “A systematic review of artificial intelligence-powered chatbots for managing chronic illness,” *International Journal of Medical Informatics*, 2024.
[12] J. W. Johnson et al., “Random forest machine learning method outperforms traditional clinical scoring in health risk prediction,” *PLOS ONE*, 2020.
[13] D. M. Lerner, “SQLite Python documentation and database programming practices,” *Python Software Foundation Documentation*, 2024.
[14] pyca/bcrypt contributors, “bcrypt: Modern password hashing for your software and your servers,” https://github.com/pyca/bcrypt, 2025.
[15] Streamlit Inc., “Build a basic LLM chat app,” https://docs.streamlit.io/develop/tutorials/chat-and-llm-apps/build-conversational-apps, 2025.
[16] F. Pedregosa et al., “RandomForestClassifier — scikit-learn documentation,” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, 2025.
[17] Python Software Foundation, “sqlite3 — DB-API 2.0 interface for SQLite databases,” https://docs.python.org/3/library/sqlite3.html, 2025.
[18] World Health Organization, “Global strategy on digital health 2020–2025,” https://www.who.int/publications/i/item/9789240020924, 2021.
[19] A. Kumar and S. Bhatia, “Chatbot systems for healthcare using artificial intelligence,” *International Journal of Engineering and Information Systems*, 2024.
[20] S. Patil et al., “Development of an intelligent chatbot for patient appointment and assistance,” *International Journal of Engineering and Information Systems*, 2024.
[21] World Health Organization, “Digital health and health system strengthening,” WHO Technical Guidance, 2024.
[22] H. Malek et al., “Machine learning for disease prediction in healthcare: a Random Forest approach,” *Journal of Biomedical Informatics*, 2024.
[23] E. B. Johnson et al., “Healthcare chatbot limitations, safety, and ethical considerations: a review,” *Health Informatics Journal*, 2024.