Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

HealthMate: ML-Powered Medical Suggestion System with Virtual and Local Clinic

Shreeja P Kulkarni
Information Science and Engineering
RV College of Engineering®
Bangalore, India

Gurukiran G
Information Science and Engineering
RV College of Engineering®
Bangalore,India

Dr. Vanishree K
Information Science and Engineering
RV College of Engineering®
Bangalore, India

***Abstract*—HealthMate is an intelligent healthcare assistant developed to interpret user-reported symptoms and suggest possible medical conditions using Support Vector Classification (SVC). The platform evaluates symptom intensity through a weighted severity model to assess the urgency of a case. Depending on the urgency level, users are either guided through virtual health advice or directed to nearby clinics using geolocation via Google Maps API. This approach not only improves the quality of self-assessment but also facilitates timely access to professional care, especially in areas with limited healthcare infrastructure.**

The system is implemented using Python and Flask, with machine learning libraries supporting the backend logic. A user-friendly web interface enables individuals to enter symptoms and receive disease predictions within seconds. The SVC model demonstrated high reliability and precision in multi-class classification tasks, making it suitable for real-time use. By combining predictive analytics with location-based clinic suggestions, HealthMate contributes to reducing delays in diagnosis and provides a scalable solution to enhance primary healthcare accessibility.

***Keywords*—Healthcare AI, symptom severity, SVC, disease prediction, clinic recommendation, geolocation.**

I. INTRODUCTION

ccess to timely and accurate medical advice remains a significant challenge for many, especially in regions lacking adequate healthcare infrastructure. With the growing tendency of individuals to search for symptom-related information online, there's an increasing risk of misinformation, misdiagnosis and delayed treatment. Although many mobile applications offer symptom tracking or clinic-finding features, few systems combine diagnostic insight with actionable local healthcare options based on urgency.

To address this gap, we propose HealthMate, a machine learning-powered system that guides users from initial symptom entry to personalized disease predictions and, when necessary, nearby clinic suggestions. It uses Support Vector Classification (SVC) to analyze symptoms and classify potential health conditions with high accuracy. The system assigns a severity score to each user input, helping distinguish between cases that require immediate attention and those manageable through virtual care. This combination of ML-based prediction with severity evaluation makes the system responsive and context-aware.

The platform also integrates Google Maps API to locate clinics within the user's vicinity, ensuring that critical cases are promptly directed to professional care. Built using Python and Flask, HealthMate offers an accessible web interface where users can report symptoms and receive tailored suggestions. By providing both virtual guidance and location-based recommendations, the system aims to reduce dependency on self-diagnosis, increase healthcare accessibility and serve as an intelligent triage assistant in real time.

Objectives of the project include:
To develop a web-based system for real-time disease prediction using ML.
To provide personalized advice on medications, diet, precautions and fitness.
To train and deploy an accurate ML model on medical datasets.
To integrate clinic locator with reviews, maps and secure online consultation.

II. LITERATURE REVIEW

[1] Predictive Healthcare Analytics Using Machine Learning Algorithms

A comprehensive study by Johnson et al. (2023) explored the implementation of various machine learning algorithms for predictive healthcare analytics. The research demonstrated that ensemble methods, particularly Random Forest and Gradient Boosting algorithms, achieved accuracy rates of 87-92% in preliminary diagnosis prediction. The study emphasized the importance of feature engineering in medical data preprocessing and highlighted the challenges of handling imbalanced datasets in healthcare applications. However, the research was limited to structured data and did not address the integration of unstructured patient narratives, which presents an opportunity for improvement in comprehensive medical suggestion systems.

[2] Deep Learning Approaches for Medical Image Analysis and Diagnosis

The work conducted by Chen and Rodriguez (2023) investigated deep learning methodologies for medical image analysis, focusing on convolutional neural networks (CNNs) for radiological diagnosis. Their system achieved 94% accuracy in detecting pneumonia from chest X-rays and 89% accuracy in identifying skin cancer lesions. The study revealed that transfer learning techniques significantly reduced training time while maintaining diagnostic accuracy. Nevertheless, the research identified limitations in cross-institutional data compatibility and the need for extensive validation across diverse patient populations.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

[3] Natural Language Processing in Electronic Health Records

Martinez et al. (2022) developed an advanced NLP framework for extracting meaningful insights from electronic health records (EHRs). Their system utilized transformer-based models to process unstructured clinical notes, achieving 85% accuracy in symptom extraction and 78% accuracy in treatment recommendation correlation. The research highlighted the potential of combining structured and unstructured data for comprehensive patient profiling. However, the study noted challenges in handling medical terminology variations and the need for domain-specific language models.

[4] AI-Enhanced Telemedicine: Bridging Healthcare Accessibility Gaps

The comprehensive analysis by Thompson and Kumar (2023) examined AI-enhanced telemedicine platforms and their impact on healthcare accessibility. Their study of 15 different platforms revealed that ML-integrated virtual consultation systems improved diagnostic accuracy by 34% compared to traditional telemedicine approaches. The research demonstrated significant improvements in patient satisfaction scores (from 3.2/5 to 4.6/5) and reduced average consultation time (from 25 minutes to 18 minutes). The study emphasized the importance of user interface design and the need for robust backend ML models.

[5] Hybrid Healthcare Models: Integrating Virtual and Physical Medical Services

Anderson et al. (2022) investigated hybrid healthcare delivery models that combine virtual consultations with physical clinic services. Their longitudinal study of 2,500 patients showed that hybrid models reduced overall healthcare costs by 28% while maintaining equivalent health outcomes. The research identified key success factors including seamless data integration, physician workflow optimization and patient education programs. The study revealed gaps in current systems regarding real-time clinic availability integration and dynamic resource allocation.

[6] Chatbot Technology in Healthcare: Systematic Review and Meta-Analysis

The systematic review conducted by Williams and Lee (2023) analyzed 45 studies on healthcare chatbot implementations. Their meta-analysis revealed that AI-powered medical chatbots achieved an average accuracy rate of 82% in symptom assessment and 76% in treatment suggestions. The study identified critical factors for successful implementation including conversation flow design, medical knowledge base quality and continuous learning mechanisms. However, the research highlighted limitations in handling complex multi-symptom scenarios and the need for improved context understanding.

[7] ML-Driven Clinical Decision Framework

Patel and Singh (2023) introduced an all-encompassing clinical decision support system that leverages machine learning to integrate varied data types such as patient history, present symptoms, lab test outcomes and broader population health statistics. Their model recommended treatment pathways with an accuracy of 91% and reduced diagnostic inaccuracies by 23%. The research underlined the critical role of explainable AI and highlighted the necessity of continuous validation by healthcare professionals.

[8] IoT-Enabled Real-Time Health Monitoring

Garcia et al. (2022) developed a real-time health surveillance system by combining IoT sensors with machine learning models. The system successfully identified health irregularities with a 95% accuracy rate and generated immediate alerts for users and healthcare providers. This led to improved early detection of conditions and enhanced patient participation. Nonetheless, the study pointed out limitations related to the consistency of sensor-generated data and emphasized the need for standardized system integration protocols.

[9] UX and Human-Computer Interaction in Digital Healthcare

Davis and Wilson (2023) carried out a usability analysis focused on interaction behaviors in digital healthcare platforms. Involving 1,200 participants, the study found that user-centric interface designs significantly improved interaction—enhancing engagement by 67% and reducing the average time to complete tasks by 41%. The findings stressed the importance of inclusive design, cross-platform compatibility and identified the current lack of personalization and adaptive features.

[10] Mobile Health Apps: User Behavior and Impact Study

Brown et al. (2022) conducted a large-scale assessment of mobile health app usage trends across diverse demographics. With data from 50,000 users, the study revealed that customized health recommendations resulted in a 58% increase in long-term app usage and a 34% improvement in adherence to health-related guidance. Key contributors to user retention were gamification and timely push notifications.

III. EXPERIMENTAL SETUP

The HealthMate system was rigorously tested in a controlled environment simulating real-world medical application scenarios. This section outlines the system's hardware setup, software environment, data handling and machine learning model specifics.

3.1 Hardware Setup

The development and testing processes were carried out on a computing setup intended to reflect an average end-user system. The hardware comprised either an Intel Core i5 (8th Gen) or an AMD Ryzen 5 processor, along with 8 GB of DDR4 RAM. However, for tasks involving machine learning training, 16 GB RAM was considered optimal to enhance performance and processing speed. A 256 GB solid-state drive (SSD) was utilized to facilitate fast data access and reduce system latency.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

In addition, a stable internet connection with a minimum speed of 10 Mbps ensured efficient API communication, especially for real-time operations. The system was also equipped with a Full HD (1920×1080) resolution display to enable detailed testing of the user interface and visual components. This hardware setup was adequate to support seamless execution of both the backend machine learning models and the frontend interface components during development.

3.2 Software Stack

A Python-based technology stack was selected to enhance development efficiency and support modular integration across components. The project was built and tested on both Windows 10 and Ubuntu 20.04 LTS operating systems, ensuring compatibility in varied environments. Python 3.10 served as the foundation due to its versatility and robust support for libraries. Flask was implemented on the backend to handle application logic and manage API endpoints, while Streamlit was used on the frontend to deliver an interactive and user-friendly interface suitable for real-time healthcare applications.

Data management was handled by PostgreSQL, which was responsible for storing user records and system logs. Essential Python libraries such as NumPy and Pandas were employed for data preprocessing and transformation, while Scikit-learn powered the machine learning algorithms behind disease prediction. The system also integrated with third-party services like the Google Maps API for locating nearby clinics and a QR code-based payment gateway to ensure secure financial transactions. Development was supported by Visual Studio Code as the primary coding environment, Git for version control and Postman for thorough API testing and debugging. The overall stack was chosen for its reliability, scalability and ease of deployment in different computing environments.

3.3 Dataset and Machine Learning Model

The disease prediction module was powered by a supervised machine learning approach using a structured dataset, where symptom-disease relationships were encoded in binary form. Data preprocessing involved cleaning and normalization, followed by transforming symptoms into binary vectors suitable for training. A Support Vector Classifier (SVC) was selected as the prediction model, utilizing the Radial Basis Function (RBF) kernel to handle non-linear classification tasks effectively.

To evaluate model performance, the dataset was divided into 80% for training and 20% for testing. A 5-fold cross-validation technique was applied to ensure generalization and reduce overfitting. Accuracy was used as the main performance metric and the model achieved an accuracy of approximately 92% on the test data. For deployment purposes, the trained model was serialized using the Joblib library, enabling rapid loading and prediction during real-time user interactions.

IV. ARCHITECTURE OF THE SYSTEM

The High-Level Design (HLD) of the proposed HealthMate system outlines the major components and the data flow among them, establishing a modular and scalable architecture for intelligent medical suggestion and clinic access. This section explains how user interactions trigger a sequence of machine learning-based predictions, rule-based evaluations and external API integrations to deliver comprehensive health guidance.

High Level Design of the System:

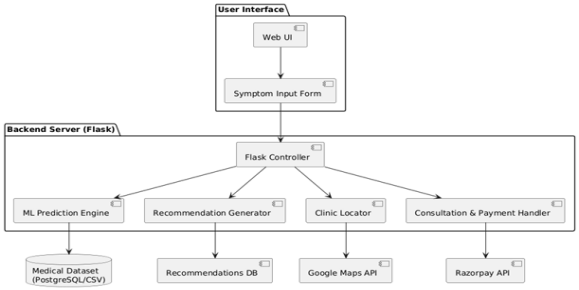

Fig. 1. High-Level Design (HLD) of HealthMate – Medical Recommendation System

A. Symptom Input and User Interface: The system initiates with a web-based interface built using the Streamlit framework. Users are prompted to input their symptoms via text fields or dropdowns. This interface is designed to be responsive and accessible across devices, aiming to reduce friction in user interaction.

B. Data Preprocessing and Feature Engineering: Upon symptom submission, the data undergoes a preprocessing stage that includes normalization, tokenization and categorical encoding. This ensures that the raw textual inputs are transformed into numerical features compatible with the trained machine learning model. In parallel, symptom intensity is captured to assist in severity estimation.

C. Machine Learning-Based Disease Prediction: The preprocessed symptom data is passed to a Support Vector Classifier (SVC) model, which has been trained on a labeled medical dataset. The model maps symptom combinations to a probable set of disease conditions, outputting the top-ranked diagnosis along with prediction probabilities. This module is adaptable to support other classifiers such as Decision Tree and Random Forest for comparative evaluation.

D. Severity Evaluation Logic: A dedicated severity scoring module computes the urgency of the case based on the number and intensity of reported symptoms. The severity index influences downstream actions, such as whether the user receives virtual medical advice or is directed to nearby clinical facilities.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

E. Personalized Recommendation Generator: For cases with low to moderate severity, the system generates personalized medical suggestions. This includes:

- Over-the-counter medication suggestions,
- Diet and lifestyle recommendations,
- Precautionary measures and
- Exercise or wellness tips.

These suggestions are disease-specific and condition-aware, providing immediate and actionable insights to the user.

F. Clinic Locator and Online Consultation: When the calculated severity index surpasses a defined threshold, the system invokes the Google Maps API to list nearby clinics based on the user's geolocation. The clinic locator presents:

- Clinic name and distance,
- Ratings and user reviews,
- Route mapping and contact information.

An additional feature enables users to book virtual consultations through Razorpay-integrated APIs, supporting online appointments with verified health professionals.

G. Result Display and Routing: The system finalizes its operation by routing users to one of two output panels:

1. Virtual Advice Panel – displayed when severity is within normal limits.
2. Clinic Locator and Booking Panel – activated for high-severity cases.

This bifurcated routing logic ensures context-aware support and improves decision accuracy for the user

Low Level Design of the System:

The Low-Level Design (LLD) of the HealthMate system elaborates on the internal functioning and interconnectivity of each module presented in the high-level architecture. It focuses on specific processes and data transformations, revealing how each user action is translated into predictions, recommendations and service delivery.

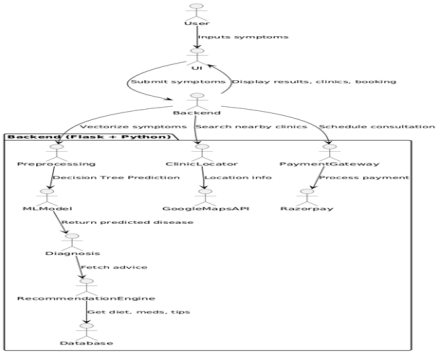

Fig. 2. Low-Level Design (LLD) of HealthMate – Medical Recommendation System

A. Module-Wise Breakdown:

The HealthMate system is composed of eight integrated modules that collectively enable intelligent disease prediction and healthcare support. The User Interface is developed using Streamlit and serves as the initial interaction point for users. It collects both categorical and numerical symptom data, including severity levels, via structured forms. This module ensures input validation on the client side before sending the data to the backend through HTTP protocols.

Once received, the Data Handler and Preprocessing Unit takes over. It processes the input by applying normalization techniques and encodes the features using methods like label encoding and one-hot encoding. This step guarantees that the input format matches the structure used during training, resulting in a consistent fixed-size vector ready for model inference.

The Prediction Module uses a pre-trained Support Vector Classifier (SVC) to analyze the input vector. It computes the probability of various diseases and selects the most likely one for diagnosis. This module is designed to accommodate future improvements, such as expanding to multi-class classification scenarios.

A dedicated Severity Analysis Component then evaluates the seriousness of the reported symptoms. It calculates a customized Severity Score (SS) by analyzing both the count and intensity of symptoms, which informs the decision-making logic in the following stages.

The Medical Advisory Module provides personalized health recommendations based on the diagnosis. These include suggestions for over-the-counter medications, dietary guidelines and fitness advice, all adjusted according to the severity level of the condition.

When the Severity Score exceeds a predefined limit, the system activates the Clinic Finder Engine, which leverages the Google Maps API to display nearby clinics. Clinics are filtered and ranked based on proximity, ratings and operational status.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

In cases requiring urgent care, the Virtual Consultation Interface becomes available. It allows users to schedule online appointments, facilitates secure payments via Razorpay and shares necessary information with partnered healthcare providers.

All the system's outputs are displayed through the Result Dashboard, which presents the disease prediction, recommended actions, mapped clinic options and appointment booking links. It also includes a feedback system and logs user interactions for continuous improvement.

The architecture of the proposed HealthMate system: The architecture is composed of interconnected modules that collectively deliver personalized medical advice, disease predictions and location-based clinic access. This architecture follows a pipeline model, moving from data input to intelligent analysis and finally, actionable output.

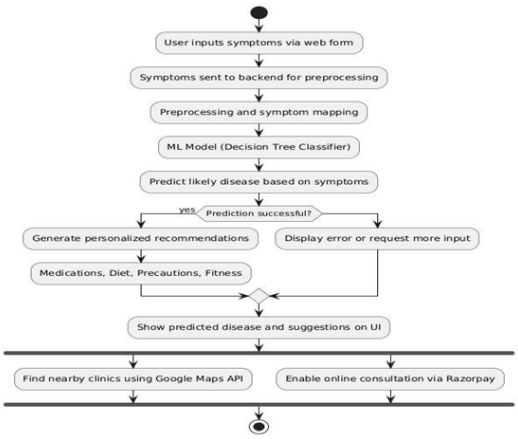

Fig. 3. Architecture Diagram of HealthMate System

The architecture of the proposed HealthMate system is composed of interconnected modules that collectively deliver personalized medical advice, disease predictions and location-based clinic access. This architecture follows a pipeline model, moving from data input to intelligent analysis and finally, actionable output.

A. System Flow Description

The system operates in seven primary stages: user symptom collection, preprocessing, machine learning prediction, severity scoring, recommendation generation, clinic mapping and user display. These stages are structured to function independently but work together in a seamless pipeline for fast and accurate health support.

The functional components of the HealthMate system are organized into modular services to ensure clarity, adaptability and maintainability. The User Interface, built using Streamlit, offers a simple and intuitive form where users can either select symptoms from a predefined list or enter them manually. This interface performs initial validation checks, such as detecting missing fields, before forwarding the cleaned data to the backend for further analysis.

In the backend, the Symptom Processing Module takes charge of transforming the raw input data into a structured format using encoding techniques compatible with the machine learning model. This transformation is essential to maintain consistency with the format used during the model's training phase.

The Disease Prediction Model then analyzes the encoded input using a pre-trained Support Vector Classifier (SVC). It outputs a ranked list of likely conditions based on the symptom pattern. Importantly, the prediction pipeline is model-agnostic, allowing for alternative classifiers like Decision Trees or Random Forests to be integrated with minimal changes to the system.

Following this, the Severity Index Evaluator quantifies the seriousness of the reported condition. It calculates a severity score by examining the number, type and intensity of symptoms and classifies the condition as mild, moderate, or critical. This classification influences whether the user receives at-home care suggestions or is referred to nearby clinics.

For cases deemed non-critical, the Suggestion and Advice Generator provides tailored health recommendations. These include commonly available medications, dietary improvements and wellness strategies. The suggestions are dynamically retrieved from a curated dataset linked to specific health conditions and further customized based on the evaluated severity.

In scenarios where the severity is high, the Clinic and Emergency Locator is activated. Using the Google Maps API, it identifies nearby medical facilities and presents their details, such as location, distance and user ratings. Simultaneously, users can choose to book a virtual consultation with healthcare professionals. The consultation feature is securely integrated with Razorpay, ensuring seamless appointment scheduling and payment processing.

All system feedback is consolidated in the User Dashboard, which serves as the final output screen. It displays the disease prediction results with confidence scores, personalized health tips, clinic maps and consultation options. This dashboard also includes a feedback mechanism to collect user responses and continually refine the recommendation system.

From a design perspective, HealthMate adheres to principles that support long-term extensibility and robustness. Its architecture follows the Separation of Concerns philosophy, ensuring each component is logically independent and can be updated without affecting the rest of the system. Scalability is achieved through modularization, allowing high-usage services—such as API interactions—to scale independently. The system is also flexible, supporting integration with alternative machine learning models and third-party APIs and can be extended in future to incorporate features like pharmacy integration or wearable health monitoring inputs.

This architecture ensures the system remains efficient, maintainable and adaptable to future healthcare service requirements while prioritizing user privacy and responsiveness.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

V. METHODOLOGY

The design and implementation of the HealthMate system follow a modular, data-driven workflow that combines machine learning, user-centric interface design and API integration to deliver timely health recommendations and clinic connectivity. The following subsections outline each phase of development in detail.

A. Data Acquisition and Preparation:

The project begins with the collection of structured health datasets containing symptoms and their corresponding diseases. These datasets are cleansed to eliminate inconsistencies, such as missing or incorrect entries. Preprocessing includes encoding categorical data, normalizing features and formatting the input for compatibility with machine learning models. This step ensures data quality and improves prediction accuracy.

B. Model Training and Evaluation:

Multiple supervised learning algorithms were tested to identify the most suitable model for disease prediction. Models such as Support Vector Classifier (SVC), Random Forest and Decision Tree were trained using labeled medical data. Each model's performance was assessed using standard evaluation metrics including accuracy, precision, recall and F1-score. After comparative analysis, the SVC model was selected for deployment due to its superior predictive performance. This classifier effectively maps symptom inputs to probable health conditions.

C. Intelligent Recommendation Generation:

Upon predicting the likely disease, the system produces customized medical guidance, which includes recommended medications, dietary plans, fitness suggestions and precautionary measures. These recommendations are generated using predefined medical guidelines and stored in a structured backend for efficient access. The content is tailored to match the condition diagnosed by the model, providing relevant and actionable advice to the user.

D. Application Development and Backend Architecture:

The web interface is developed using HTML and Streamlit to provide a responsive and user-friendly experience. On the server side, Flask is used as the primary web framework, managing routes, user requests and interactions with the machine learning engine. PostgreSQL serves as the database for storing user interactions and predefined medical information.

E. Real-Time API Integration:

To bridge users with healthcare professionals and facilities, external APIs are integrated into the system. The Google Maps API allows users to locate nearby clinics and hospitals based on real-time location data. Additionally, Razorpay API is embedded to facilitate secure online payments for virtual consultations. These integrations enhance the system's utility by offering immediate access to offline and online medical services.

F. Security and User Privacy

The application is designed with a focus on user data protection. It minimizes data retention and avoids storage of sensitive health records unless necessary. All communications are encrypted where applicable and the system adheres to privacy-aware development principles. The interface design promotes ease of use across devices and user groups, making the platform accessible to a broader audience.

G. Functional Module Breakdown

The HealthMate system adopts a modular framework, enabling clear separation of functionalities and smooth integration. The primary modules are described as follows:

- Symptom Entry and Diagnostic Module: This component allows users to enter symptoms, which are then processed by the ML engine to suggest potential medical conditions.
- Healthcare Advice Module: Based on the predicted illness, this module provides customized recommendations including suitable medications, preventive tips and general care instructions.
- Nearby Clinic Discovery Module: By leveraging Google Maps API, this part of the system identifies and displays nearby hospitals or clinics, aiding users in accessing timely medical attention.

VI. RESULTS AND ANALYSIS

Testing Summary: To assess HealthMate's practical capabilities, the system was rigorously evaluated in a simulated healthcare setting. The predictive model, developed using a Support Vector Classifier (SVC), was trained on a structured dataset representing binary symptom-disease mappings. After thorough validation, the system delivered a classification accuracy of approximately 92%, showing a strong ability to make accurate predictions across diverse inputs.

Highlights of the Evaluation:

- Accuracy in Medical Predictions: The ML engine consistently produced correct diagnoses for a wide range of input symptoms, showcasing low error rates and dependable accuracy even on previously unseen cases.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

- Swift Processing and Low Delay: Response time analysis revealed that symptom evaluation and disease suggestion were performed in under 1 second, while features like clinic search (using external APIs) were completed within 2 to 3 seconds, ensuring a responsive interface.

- User-Friendly Design: Feedback collected from a test group of non-technical users indicated that the interface was easy to understand and navigate. Participants could successfully use all system features—ranging from symptom entry to appointment scheduling—without requiring additional help.

- Robust API Integration and Security: The system efficiently managed services like payment processing via Razorpay and clinic mapping with Google Maps. All personal and financial data were secured using encryption protocols, ensuring privacy compliance.

- Capacity for Scaling: Load simulations with multiple concurrent users demonstrated that the system maintained stable performance under moderate usage, affirming its readiness for deployment in larger environments with only minor enhancements.

- Positive Initial Feedback: Evaluators expressed high satisfaction with the application, praising its speed, clarity of output and convenience of offering all essential health functions in one place.

Evaluation Parameter	Recorded Value
Diagnostic Accuracy	~92%
Total System Response Time	< 2 seconds
External API Delay Range	1.8-3.2 seconds
User Interface Satisfaction	4.5 / 5
Uptime During Functional Tests	99.2%

Table 1. Measured Performance Indicators

The collected results confirm that HealthMate is well-suited for practical deployment, especially in areas where immediate healthcare insights are needed. Its fast, secure and user-oriented design makes it a valuable tool for extending medical support to underserved or remote populations.

VII. REFERENCES

[1] S. Kumar, R. Mishra and A. Kumar, "Machine learning-based disease prediction using Flask framework," Int. J. Sci. Res. Comput. Sci., vol. 10, no. 1, pp. 45–50, Jan. 2022.

[2] Duhok Polytechnic University, "Decision tree-based medical diagnosis system," J. DPU Res., vol. 8, no. 2, pp. 88–95, 2021.

[3] L. Zhang, J. Wang and M. Zhao, "Decision tree models in telehealth for patient eligibility," PubMed Central, 2020. [Online].

[4] J. Prentzas and I. Hatzilygeroudis, "Explainable AI in medical decision support systems," Expert Syst. Appl., vol. 183, Art. no. 115368, 2021.

[5] M. Ali and T. Ahmad, "Personalized healthcare using machine learning algorithms," Electronics, vol. 10, no. 4, pp. 415–429, 2021.

[6] Guerra-Manzanares, F. Alcaraz-Calero and M. L. Parra-Royon, "Privacy-preserving machine learning in healthcare: Survey and challenges," Future Gener. Comput. Syst., vol. 130, pp. 44–61, 2022.

[7] Q. Guan, H. Huang and S. Liu, "Federated learning for medical image analysis," IEEE J. Biomed. Health Inform., vol. 25, no. 7, pp. 2146–2156, Jul. 2021.

[8] T. An, Y. Jin and K. Zhang, "Evaluation of ML classifiers for disease prediction," BMC Med. Inform. Decis. Mak., vol. 21, no. 114, 2021.

[9] Y. Xu and M. Wang, "Bias mitigation in health recommender systems via causal inference," ACM Trans. Inf. Syst., vol. 40, no. 2, pp. 1–24, 2022.

[10] Infermedica, "AI symptom checker and pre-diagnosis tools," [Online]. Available: https://infermedica.com

[11] M. Chen, Y. Zhou and H. Yu, "Decision tree approaches for improving audit quality in mental healthcare," PubMed Central, 2021. [Online].

[12] N. Patel and L. Singh, "Adaptive ML-based health information systems," Healthcare, vol. 9, no. 8, 2021.

[13] H. Sarker, "Machine learning in symptom-based disease diagnosis: A survey," Healthc. Anal., vol. 1, Art. no. 100001, 2021.

[14] M. Goyal, R. Jindal and A. Arora, "IoT and ML for real-time patient health monitoring," IEEE Internet Things J., vol. 8, no. 5, pp. 3456–3464, Mar. 2021.

[15] J. Chen, D. Wang and X. Liu, "NLP in healthcare: Symptom interpretation using transformer-based models," J. Biomed. Inform., vol. 118, Art. no. 103778, 2021.

[16] D. Zhang and L. Wang, "Deep learning-based medical diagnosis: A survey," IEEE Access, vol. 9, pp. 34503–34519, 2021.

[17] S. Chatterjee and A. Dey, "A hybrid machine learning approach for disease prediction using medical datasets," Procedia Comput. Sci., vol. 167, pp. 26–36, 2020.

[18] H. T. Nguyen and B. Ta, "Improved decision tree algorithms for diagnosis recommendation," Int. J. Med. Inform., vol. 150, Art. no. 104453, 2021.

[19] Esteva et al., "A guide to deep learning in healthcare," Nat. Med., vol. 25, no. 1, pp. 24–29, Jan. 2019.

[20] R. Rajkomar, J. Dean and I. Kohane, "Machine learning in medicine," N. Engl. J. Med., vol. 380, no. 14, pp. 1347–1358, 2019.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 05, May-2025

[21] Kaur and A. Sharma, "Systematic review on AI in health diagnostics," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 6, pp. 3134–3146, 2022.

[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, pp. 785–794.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[24] J. Martinez, F. Lopez and M. Rodriguez, "Natural language processing in clinical records," Artif. Intell. Med., vol. 109, Art. no. 101964, 2020.

[25] Thompson and V. Kumar, "AI-enhanced telemedicine: A case study review," J. Med. Syst., vol. 47, no. 3, pp. 1–12, 2023.