



International Journal of Multidisciplinary Research in Science, Engineering and Technology

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



Impact Factor: 9.864

Volume 9, Issue 5, May 2026



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

HealthGuardAI – Intelligent Multi-Disease Prediction and Health Monitoring System

Sakthivel R.M., Dr.B.Suresh

III B.Sc.-CS-A, Dept. of Computer Science, Vels Institute of Science, Technology and Advanced Studies, Chennai,
Tamil Nadu, India

Asst. Professor, Dept. of Computer Science, Vels Institute of Science, Technology and Advanced Studies, Chennai,
Tamil Nadu, India

ABSTRACT: HealthGuardAI is a web-based healthcare decision support system designed to predict and monitor multiple diseases using machine learning. The project addresses the growing need for early detection of chronic conditions such as heart disease, diabetes, and lung cancer by allowing users to enter health-related parameters through a simple interface. The system processes the input data, performs validation and preprocessing, and then generates predictions with risk scores. Along with these results, it also provides personalized suggestions such as diet guidance, exercise advice, and precautionary measures. The application is built with Flask, SQLite, SQLAlchemy, Scikit-learn, Pandas, NumPy, Chart.js, and xhtml2pdf, combining machine learning, database management, and visualization in one platform.

In addition to disease prediction, the system includes secure user authentication, prediction history, an analytics dashboard, a BMI calculator, chatbot support, and downloadable PDF reports. These features make the project more useful than a single-purpose prediction tool because they support continuous health monitoring and user awareness. The journal records the main ideas, technical environment, analysis, design, implementation, testing, and conclusion of the project in an organized format.

I. INTRODUCTION

The rise in chronic illnesses has made preventive healthcare more important than ever. Many diseases develop gradually and remain unnoticed until the condition becomes serious. Traditional healthcare methods depend mainly on periodic checkups, which can delay diagnosis and reduce the chance of timely intervention. This project was developed to support early awareness by combining web technology with machine learning.

HealthGuardAI focuses on multi-disease prediction rather than a single-condition model. It accepts key health values such as age, blood pressure, glucose level, heart rate, body mass index, and lifestyle information. After processing the data, the application provides prediction outputs that help users understand their health status more quickly.

The project was designed with a practical goal: to create a system that is easy to use, secure, and informative. The interface is intended for ordinary users who may not have technical knowledge. Through simple data entry, visual dashboards, prediction history, and downloadable reports, the platform supports both immediate feedback and long-term monitoring.

Another important aspect of the project is accessibility. Because the system is web-based, users can access it from common browsers without installing specialized software. This makes the solution suitable for academic use, personal health tracking, and future extension into broader healthcare applications.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

II. WORKING ENVIRONMENT

2.1 Hardware Requirements

The system is designed to run on standard computing equipment. A basic processor such as Intel Core i3 or AMD Ryzen 3 is sufficient for development, while 4 GB RAM is the minimum recommended memory. An SSD improves application responsiveness, especially when handling datasets, databases, and model files. Keyboard, mouse, monitor, and network connectivity are required for normal operation. The project does not depend on specialized hardware, which keeps the deployment simple and economical.

2.2 Software Requirements

The project uses Python as the main programming language and Flask as the web framework. HTML, CSS, Tailwind CSS, and JavaScript are used to build the interface. Scikit-learn supports the machine learning models, while Pandas and NumPy handle preprocessing and analysis. SQLite with SQLAlchemy stores user data and prediction history, Chart.js presents visual analytics, and xhtml2pdf generates downloadable reports. A browser such as Chrome or Edge is enough to access the system.

2.3 System Software

System software includes the operating system, Python runtime, database engine, file management, memory management, and security support. These components allow the application to execute smoothly and maintain user privacy. Together, the hardware and software environment provides a reliable base for running the HealthGuardAI application.

III. SYSTEM ANALYSIS

System analysis was performed to understand the problem, define requirements, and evaluate whether the proposed solution could be implemented effectively. The traditional healthcare approach often lacks real-time monitoring and early detection. Users usually depend on manual diagnosis or periodic medical visits, which may not provide continuous insight into changing health conditions.

The project therefore identifies a need for a more responsive and integrated solution. HealthGuardAI collects user health data, processes it, and uses trained models to estimate disease risk. Important attributes were selected based on their relevance to common chronic diseases. This improves prediction usefulness while keeping the interface compact and understandable.

The feasibility study shows that the project is practical in technical, economic, operational, and schedule terms. It uses open-source technologies, a lightweight database, and modular features that can be built and tested in stages. Because of this structure, the project can be completed within an academic timeline without excessive cost.

The analysis also highlights the limitations of existing systems. Many earlier tools focus on a single disease, do not store health history, and lack visualization or personalized guidance. HealthGuardAI was proposed to fill these gaps with a broader and more user-friendly design.

IV. SYSTEM DESIGN

The architecture of HealthGuardAI follows a three-tier design: presentation layer, application layer, and data layer. The presentation layer contains the user interface built with HTML, CSS, Tailwind CSS, and JavaScript. This layer allows users to register, log in, enter health values, view predictions, and access dashboards.

The application layer is implemented using Flask in Python. It handles request processing, validation, preprocessing, prediction calls, BMI computation, session management, chatbot logic, and report generation. This layer acts as the core of the system because it connects the front end to the machine learning and database components.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

The data layer uses SQLite with SQLAlchemy ORM to store user credentials, health records, prediction results, and history. This helps maintain consistency and supports retrieval of past results. The database is organized in a way that makes future reporting and trend analysis straightforward.

The design is also supported by UML representations such as use case, activity, sequence, and ER diagrams. These diagrams clarify how users interact with the system, how data moves through the modules, and how the database is structured. The overall design encourages modularity, scalability, and easier maintenance.

V. DATABASE DESIGN AND MODULE DESCRIPTION

The database design centers on four main tables: User, Health_Data, Prediction, and Prediction_History. The User table stores registration details such as username, email, and password. Health_Data stores values entered by the user, including blood pressure, sugar level, heart rate, and age. Prediction stores the disease type, result, risk score, and prediction time, while Prediction_History links users and predictions for long-term tracking.

User Authentication: Manages registration, login, sessions, and access control so that only authorized users can use the platform.

Health Data Input: Collects user information through structured forms and checks values for completeness and correctness.

Data Preprocessing: Cleans missing values, transforms data into numerical format, and prepares input for machine learning models.

Disease Prediction: Uses trained models such as Logistic Regression, Decision Tree, or Random Forest to generate results.

Analytics Dashboard: Displays health trends using charts and graphs so users can understand their data visually.

VI. IMPLEMENTATION AND TESTING

Implementation began with medical dataset preparation and model training. The collected data was cleaned, checked for missing values, and converted into a form suitable for machine learning. Once the data was ready, the models were trained using Scikit-learn and the most suitable one was integrated into the Flask application.

The front end was then connected to the back end so that user input could be sent directly to the prediction engine. After a prediction is generated, the result is stored in the database and displayed in the interface. The system also produces recommendations based on the outcome so that users receive more than a simple label.

Testing was carried out to verify that all modules worked correctly. Input validation was checked to ensure that incorrect values were handled safely. Database operations were tested for proper storage and retrieval. The prediction workflow, dashboard display, BMI calculator, chatbot responses, and PDF report generation were also checked for stability.

The testing phase confirmed that the system could perform its main functions smoothly. Any small errors identified during integration were corrected to improve reliability and user experience.

VII. CONCLUSION AND FUTURE SCOPE

HealthGuardAI demonstrates how machine learning and web development can be combined to support preventive healthcare. By allowing users to input health data and receive disease risk estimates, the project provides a practical way to promote early awareness and health monitoring.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

The system is valuable because it combines several useful features in one platform: prediction, history tracking, visual analytics, BMI assessment, chatbot guidance, and PDF report generation. These functions make the application more complete than a single-purpose prediction tool.

The project also has room for future improvement. It can be expanded to support additional diseases, more advanced machine learning models, wearable device integration, mobile access, and doctor appointment features. These extensions would make the system even more useful in real-world healthcare settings.

Overall, the project shows that intelligent systems can play a strong role in modern healthcare by making health data easier to interpret and by encouraging timely action.

REFERENCES

1. Project dissertation text: HealthGuardAI – Intelligent Multi-Disease Prediction and Health Monitoring System.
2. Python Flask documentation and common open-source machine learning libraries such as Scikit-learn, Pandas, NumPy, SQLAlchemy, Chart.js, and xhtml2pdf.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com