

Smart HealthCare: AI-Powered Symptom-Oriented Disease Prediction, Intelligent Health Advisory, and Personalized Monitoring System.

Asst. Prof. Mayuri Shripati Rane¹, Asst. Prof. M. S. Bhandigare², Prajakta Krishna Patil³

¹Assistant Professor, Master of Computer Applications,

²Head of Department, Master of Computer Applications

³Student, Master of Computer Applications,

Sant Gajanan Maharaj College of Engineering, Mahagaon, Shivaji University, Kolhapur, India.

kagwademayu1016@gmail.com, msbhandigare@gmail.com, patilprajakta267@gmail.com,

Abstract: *Smart Healthcare is a AI-based web application designed to make basic health guidance more accessible, especially for people who may not always have quick access to doctors. The system allows users to enter their symptoms and get an instant idea of possible diseases along with helpful health advice. The project uses a combination of two approaches. The first is a rule-based system that matches user symptoms with a medical database to give quick results. The second uses an AI model to provide more detailed insights, such as how serious the condition might be, what precautions to take, and simple home remedies. Apart from diagnosis, the application also helps users manage their health by storing past reports, setting medicine reminders, and offering a chatbot that can answer basic health-related questions. It also includes a feature to find nearby hospitals, making it more useful in real-life situations. The system is built using Angular for the frontend, ASP.NET Core for the backend, and SQL Server for storing data. Overall, the goal of this project is to create a simple, smart, and user-friendly healthcare platform that helps people make better health decisions at the right time.*

Keywords: *Smart Healthcare*

I. INTRODUCTION

Access to timely and reliable healthcare remains a major challenge, especially in rural and semi-urban areas where medical facilities and qualified professionals are not always easily available. Many people rely on internet searches or unverified sources for understanding their symptoms, which can often lead to confusion or incorrect self-diagnosis. There is a growing need for a system that can provide quick, basic health guidance in a simple and trustworthy way.

Smart Healthcare is developed as a solution to this problem. It is a web-based application that allows users to enter their symptoms and receive instant feedback about possible health conditions. The aim is not to replace doctors, but to assist users in understanding their health better and deciding when to seek medical attention.

The system uses a combination of a rule-based method and artificial intelligence to generate results. The rule-based approach quickly matches user symptoms with a structured medical database, while the AI-based approach provides more detailed insights such as severity, precautions, and basic remedies. This combination helps balance speed and accuracy while keeping the system easy to use. In addition to diagnosis, the platform offers features like health history tracking, medicine reminders, a chatbot for basic health queries, and a tool to find nearby hospitals. These features make it more than just a symptom checker and turn it into a small personal health management system.

The application is built using modern web technologies including Angular for the frontend, ASP.NET Core for the backend, and SQL Server for data storage. The overall goal of Smart Healthcare is to create a simple, accessible, and intelligent platform that helps users make informed decisions about their health in a timely manner.



II. RELATED WORK

Many applications are already available that help users understand their health based on symptoms. For example, tools like WebMD and Mayo Clinic allow users to enter symptoms and get possible disease suggestions. These platforms are useful, but they mostly follow fixed rules and often give very general results without considering personal context.

In recent years, researchers have also used machine learning techniques to improve disease prediction. Different algorithms are trained on medical data to identify patterns between symptoms and diseases. These systems can provide better accuracy, but they usually depend on large datasets and are not always easy for normal users to understand.

Some modern healthcare systems use artificial intelligence and chat-based interfaces to provide more interactive support. These systems can understand user input better and give more detailed responses. However, relying only on AI can sometimes lead to inconsistent or unclear results if the system is not properly controlled.

There are also separate applications for features like medicine reminders, health tracking, or finding nearby hospitals. While these are helpful, they are usually not combined into a single platform.

The Smart Healthcare system improves on these existing solutions by combining a simple rule-based approach with AI-based analysis. This helps provide both quick results and more detailed guidance. In addition, it brings multiple features like diagnosis, history tracking, reminders, chatbot support, and hospital search into one system, making it more practical and useful for everyday healthcare needs.

III. LITERATURE REVIEW

1. Symptom-Based Disease Prediction Using Machine Learning Techniques

Authors: R. Kumar, A. Mehta

Explanation: This paper presents a system that predicts diseases based on user symptoms using machine learning algorithms. It helps users perform basic health assessments without immediately visiting a hospital. The system improves diagnostic speed and provides early health guidance.

Additional Issues: Requires improvement in real-time accuracy and integration with clinical validation systems.

2. Web-Based Healthcare Recommendation System Using Symptom Analysis

Authors: P. Sharma, S. Patel

Explanation: This research introduces a web-based platform that analyzes symptoms and provides personalized health recommendations and precautions. It uses weighted symptom matching instead of simple rule-based methods, improving prediction quality and user engagement.

Additional Issues: Needs stronger authentication mechanisms and secure handling of sensitive healthcare data.

3. Secure Smart Healthcare Platform with Role-Based Access Control

Authors: A. Singh, R. Gupta

Explanation: This study proposes a secure healthcare system that combines disease prediction, patient record management, and analytical dashboards. It uses role-based access control to protect sensitive data and improve system reliability.

Additional Issues: System scalability and integration with real-world hospital databases need enhancement.

4. Data Mining Techniques for Disease Prediction in Healthcare Systems

Authors: N. Gupta, V. Reddy

Explanation: This paper explores the use of data mining techniques to analyze patient data and predict diseases. Classification algorithms are used to identify patterns and support early diagnosis, improving prediction accuracy.

Additional Issues: Requires high-quality and large datasets for better performance and reliable predictions.



5. Smart Healthcare Monitoring System Using Web Technologies

Authors: K. Reddy, D. Verma

Explanation: This system provides a web-based platform where users can enter symptoms and receive disease predictions along with precautions. It also maintains patient history for future reference, improving healthcare accessibility.

Additional Issues: Needs real-time monitoring features and integration with wearable health devices.

6. Healthcare Diagnostic Support System Using Symptom Scoring Methods

Authors: M. Patel, S. Shah

Explanation: This research focuses on a scoring-based system where symptoms are assigned weights to calculate disease probability. It helps users understand their health condition before consulting a doctor.

Additional Issues: Accuracy depends heavily on correct symptom weighting and structured database design.

7. Digital Healthcare Assistant for Early Disease Awareness

Authors: S. Khan, T. Ahmed

Explanation: This study introduces a digital assistant that allows users to select symptoms and receive possible diagnoses along with precautionary advice. It promotes early disease awareness and reduces the burden on healthcare facilities.

Additional Issues: Requires stronger security, authentication, and better patient history tracking for reliability.

IV. PROBLEM STATEMENT

Human-based health assessment and scattered medical resources create challenges in obtaining quick and reliable preliminary diagnosis. Many individuals ignore early symptoms due to lack of time, awareness, or easy access to healthcare services. Existing platforms often provide unstructured and non-personalized information, making it difficult for users to understand their condition. Additionally, there is no single system that combines symptom analysis, remedies, and health tracking in one place.

V. PROPOSED SYSTEM OVERVIEW

The Smart Healthcare system is designed as a web-based platform that helps users understand their health conditions based on the symptoms they experience. The system follows a simple flow where users enter their symptoms, and the application processes this information to provide possible disease predictions along with useful health advice.

The core of the system is the diagnosis engine, which uses a hybrid approach. First, a rule-based method matches the selected symptoms with a structured medical database to quickly identify possible diseases and calculate a match percentage. Then, an AI-based module is used to generate more detailed insights such as severity level, precautions, and basic home remedies. This combination helps the system provide both fast and meaningful results.

The system is divided into two main parts: frontend and backend. The frontend is developed using Angular and provides an interactive and user-friendly interface where users can search symptoms, view results, and manage their health data. The backend is built using ASP.NET Core Web API, which handles all the logic, processes user requests, and communicates with the database. Microsoft SQL Server is used to store data such as symptoms, diseases, user details, and diagnosis history.

Apart from diagnosis, the system includes additional features to improve usability. Users can view their past diagnosis history, set medicine reminders, and interact with a chatbot for basic health queries. The system also provides a feature to find nearby hospitals using location-based services, making it more practical in real-life situations.

Security is also considered an important part of the system. User authentication is handled using JWT tokens, and sensitive data is protected to ensure privacy. Overall, the proposed system aims to provide a simple, fast, and reliable platform that supports users in making better health-related decisions.



VI. SYSTEM ARCHITECTURE

The Smart Healthcare System is a web-based platform that connects users and administrators within a single integrated system. Its main objective is to provide quick and reliable health assessment by analyzing user-selected symptoms and predicting possible diseases. The system also offers remedies, precautions, and maintains user health records securely using a centralized database. It ensures efficient data processing, user-friendly interaction, and supports better health awareness through a modern and scalable architecture.

1. Modules

1.1 User Authentication Module

In this module, users can register and securely log into the system. During registration, users provide basic details such as name, email, age, and password. The system verifies login credentials and allows access to authorized features. This module ensures data security and protects user information.

1.2 User Profile Module

Users can manage their personal information such as age, gender, and basic health details. They can update their profile and maintain accurate data, which helps the system provide better and more relevant health predictions.

1.3 Symptom Selection Module

In this module, users select the symptoms they are experiencing from a predefined list. Multiple symptoms can be selected, and this information is sent to the system for analysis and disease prediction.

1.4 Diagnosis Module

This module is the core of the system. Users select symptoms, and the system processes them using a hybrid approach. The rule-based engine matches symptoms with the database to calculate a match percentage, while the AI module provides more detailed insights such as severity level, precautions, and health advice. The final result is displayed in a clear and understandable format.

1.5 Severity Analysis Module

This module evaluates how serious the condition is based on selected symptoms. The system categorizes the condition into levels like low, medium, or high severity and suggests whether medical consultation is needed.

1.6 Health Recommendation Module

Based on the prediction, the system provides basic remedies, precautions, and health advice. It helps users take early preventive actions and manage their condition better.

1.7 Diagnosis History Module

This module stores all previous diagnosis results of the user. Users can view their past records, including symptoms, predicted diseases, and dates, helping them track their health over time.

1.8 Health Assistant Module (Chatbot)

This module provides an AI-based chatbot that helps users with basic health-related queries. Users can ask questions in simple language, and the chatbot responds with helpful information, suggestions, and follow-up guidance. It also supports multiple languages for better accessibility.



1.9 Location-Based Services Module

This module helps users find nearby hospitals and clinics. Users can search based on their location, view available healthcare facilities, and access directions using map services. This feature makes the system more practical in emergency or real-life situations.

1.10 Admin Module

Admins can log into the system and manage data such as diseases, symptoms, and remedies. They can add, update, or delete records and ensure the system remains accurate and up to date.

1.11 Data Management Module

This module handles the storage and organization of all system data in the database, including users, symptoms, diseases, and history. It ensures efficient data retrieval and secure management.

2. Backend Architecture

The system uses ASP.NET Core Web API and SQL Server to manage authentication, data processing, and storage:

- **Authentication (JWT-Based):** Ensures secure login and identity verification for users and admins. It generates tokens to maintain user sessions and protect sensitive data.
- **Business Logic Layer:** Handles core functionalities such as symptom analysis, disease prediction, severity calculation, and health recommendations. It processes user input and applies logic to generate results.
- **SQL Server Database:** Stores structured data including users, symptoms, diseases, remedies, and diagnosis history. It ensures secure and efficient data management.
- **API Layer (REST APIs):** Enables communication between the Angular frontend and backend. It handles requests like login, symptom submission, prediction results, and data retrieval.
- **AI Integration:**

The system connects to an external AI service to generate advanced health insights, including severity and recommendations.

3. Diagnosis and Notification Workflow

- The user logs into the system and selects symptoms.
- The frontend sends the selected symptoms to the backend API.
- The backend processes the data using the rule-based algorithm and AI module.
- The system generates a diagnosis result with match percentage and health advice.
- The result is stored in the database and displayed to the user.
- Additional features like reminders, chatbot interaction, and hospital search can be accessed as needed.

VII. IMPLEMENTATION DETAILS

The implementation of the Smart Healthcare system consists of several key steps: User Authentication, Symptom Input, Diagnosis Processing, Result Generation, and Additional Health Services. These steps work together to provide a smooth and efficient user experience.

User Authentication and Profile Management

Users first register and log into the system securely using their credentials. Authentication is handled using JWT tokens to ensure safe access. After logging in, users can manage their profile by adding personal details such as age, gender, blood group, and basic health information. All user data is securely stored in the database to maintain privacy and integrity.



Symptom Input and Selection

Once logged in, users can search and select symptoms from a predefined list available in the system. The application provides filtering options based on body systems and a search feature to make selection easier. Users can select multiple symptoms, and the system ensures that at least a minimum number of symptoms are selected before proceeding to diagnosis.

Diagnosis Processing

After symptom selection, the system processes the data using a hybrid approach. The rule-based engine compares selected symptoms with the stored medical database and calculates a match percentage for possible diseases. At the same time, the AI module analyzes the symptoms to generate deeper insights such as severity level, precautions, and possible health risks. This combination ensures both speed and meaningful results.

Result Generation and Display

The system generates a final diagnosis result that includes the predicted disease, match percentage, severity level, and a brief description. It also provides recommended precautions and basic home remedies. The results are displayed in a clear and user-friendly format so that users can easily understand their condition.

Notifications and Health Services

The system includes additional features such as medicine reminders and health notifications. Users can set reminders for medications, and the system provides timely alerts. The chatbot module allows users to ask basic health-related questions and receive instant responses. The nearby hospital feature helps users locate healthcare facilities based on their location.

History and Data Management

All diagnosis results are stored in the system for future reference. Users can view their past diagnosis history along with details such as symptoms, predicted diseases, and timestamps. This helps users track their health over time and make better decisions. The stored data also improves system usability by providing quick access to previous records.

Conclusion of Implementation

This step-by-step implementation ensures a secure, efficient, and user-friendly healthcare system. It helps users quickly understand their health condition, take preventive actions, and maintain a record of their medical history, while acting as a supportive tool alongside professional medical advice.



VIII. SYSTEM AECHTURE

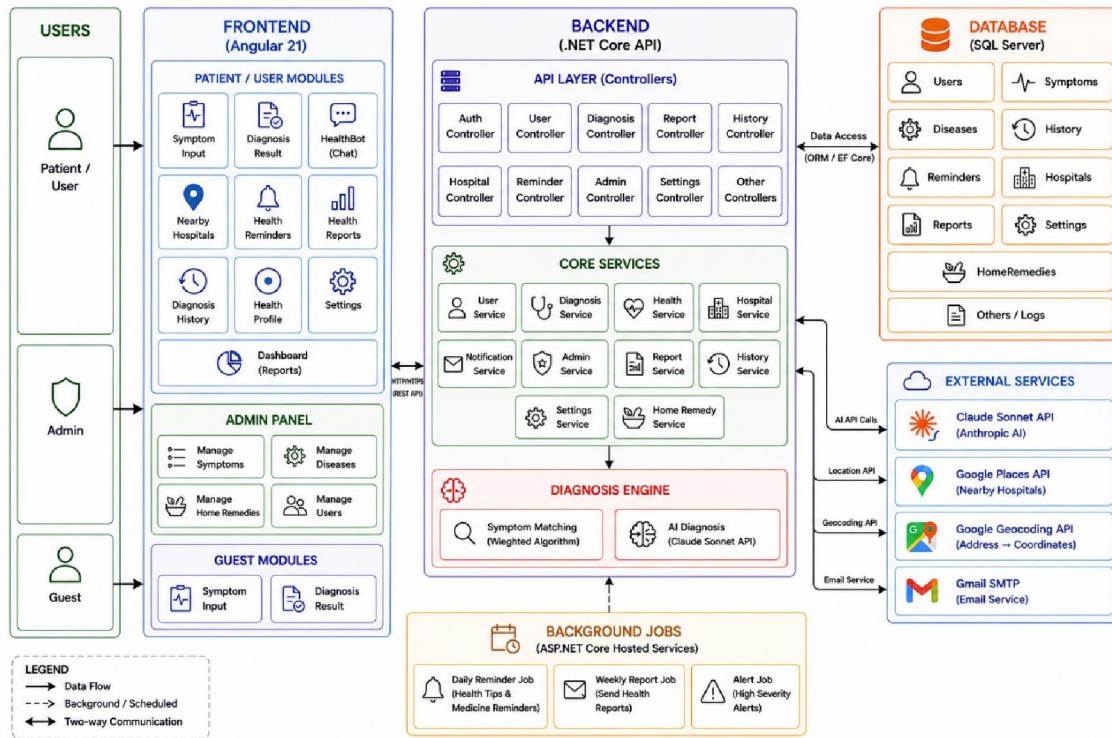


Fig. System Architecture

IX. PROPOSED SYSTEM

System Architecture:

System Architecture is shown in Figure 5.1

Our proposed system will function in the following steps:

Step 1: User Registration

Users register in the web application by providing their basic details such as name, email, and password. The system stores this information securely for future access.

Step 2: Login & Authentication

The system verifies users through secure authentication using JWT tokens and OAuth. Only authorized users can access the application features.

Step 3: Symptom Selection

After logging in, users select symptoms from a predefined list. The system provides search and filter options to make symptom selection easy and user-friendly.

Step 4: Diagnosis Processing

The system processes the selected symptoms using a rule-based matching algorithm to identify possible diseases. At the same time, the AI module analyzes the symptoms to generate deeper insights such as severity level and health advice.



Step 5: Result Generation

The system generates a diagnosis result that includes predicted disease, match percentage, severity, precautions, and basic home remedies. The result is displayed in a clear and understandable format.

Step 6: History Storage

The system saves the diagnosis result along with selected symptoms in the database. Users can view their past diagnosis history anytime.

Step 7: Additional Services

The system provides additional features such as medicine reminders, chatbot assistance for health queries, and a nearby hospital search feature to improve usability.

Step 8: Notification System

Users receive notifications for medicine reminders, important health alerts, and updates related to their health activity.

X. ANALYSIS OF PROPOSED SYSTEM

1. Enhanced Efficiency and Real-Time Response:

The proposed Smart Healthcare system improves the speed and accessibility of basic health guidance. Users can enter symptoms and receive instant results without waiting for manual consultation. The system processes data quickly and provides real-time feedback, helping users make timely health decisions.

2. Intelligent Diagnosis with Hybrid Approach:

The system uses a combination of rule-based logic and AI-based analysis. The rule-based engine ensures fast and structured matching of symptoms with diseases, while the AI module provides deeper insights such as severity level, precautions, and personalized advice. This hybrid approach improves both accuracy and usability compared to traditional systems.

3. Secure User Authentication and Data Privacy:

The platform implements secure authentication using JWT tokens and OAuth. User data such as personal details and health records are stored securely in the database. Role-based access ensures that only authorized users can access specific features, maintaining privacy and system security.

4. Improved User Experience and Accessibility:

The system provides a simple and user-friendly interface where users can easily select symptoms, view results, and access additional features. The inclusion of chatbot support, medicine reminders, and nearby hospital search improves usability and makes the system practical for everyday use.

4. Real-Time Results and Notifications:

The system generates instant results after analysis and provides notifications with disease details, severity level, and advice. High-risk conditions trigger alert messages encouraging users to consult a doctor immediately.

5. Centralized Data and System Reliability:

All user data, symptoms, diseases, and diagnosis history are stored in a centralized database. This improves data consistency, allows easy retrieval of past records, and ensures smooth system operation. Compared to traditional methods, the system reduces dependency on manual processes and improves reliability.



XI. MODULES

The Smart Healthcare system is divided into five main modules: User, Diagnosis, Admin, Chatbot, and Notification. Each module handles specific functionalities to ensure smooth operation of the system.

User Module

This module manages user registration, login, and profile information. Users can update their personal details and access all system features such as symptom selection, diagnosis results, and history tracking.

Diagnosis Module

This is the core module of the system. It processes user-selected symptoms and generates disease predictions using both rule-based and AI-based approaches. The module provides match percentage, severity level, and health recommendations.

Admin Module

The admin module manages system data, including symptoms, diseases, and remedies. Administrators can perform CRUD operations and monitor system usage. This ensures that the database remains accurate and up to date.

Chatbot Module

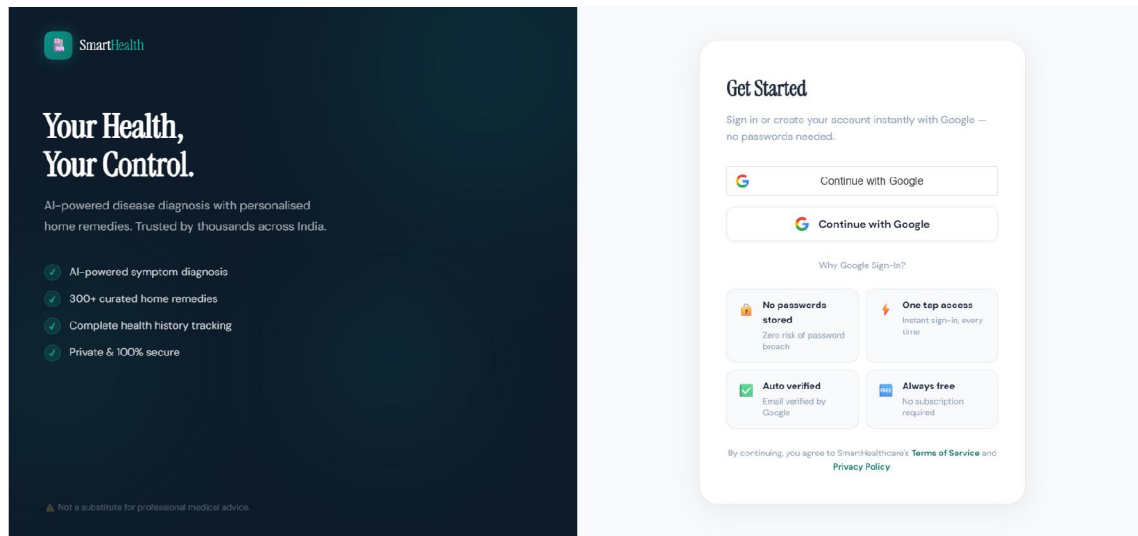
The chatbot module provides an AI-based assistant that helps users with basic health-related questions. It offers instant responses, suggestions, and guidance, improving user interaction with the system.

Notification Module

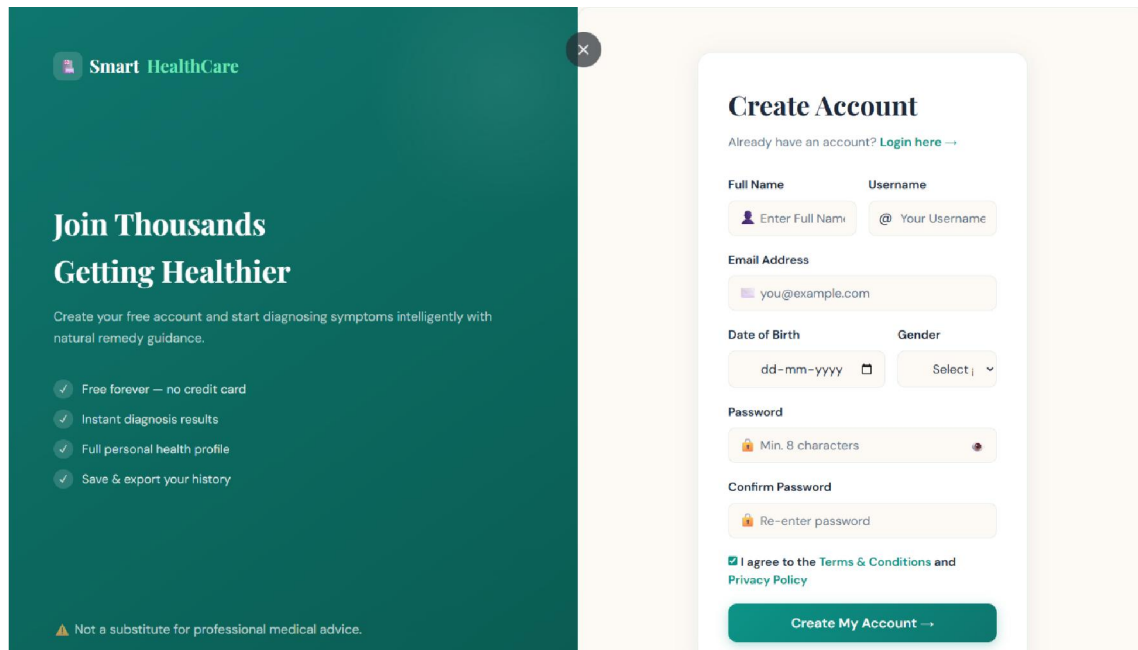
The notification module sends alerts and updates to users. It includes medicine reminders, health alerts, and system notifications. This module ensures timely communication and improves user engagement.

XII RESULTS

USER LOGINS



REGISTRATION PAGE & USER DASHBOARD



XIII. CONCLUSION

The Smart Healthcare system is an AI-based web application that helps users understand their health based on symptoms. It uses a hybrid approach combining a rule-based algorithm and artificial intelligence (AI) to provide both quick predictions and detailed health insights such as severity, precautions, and remedies. Along with diagnosis, features like history tracking, medicine reminders, AI chatbot support, and nearby hospital search make the system practical and user-friendly. Overall, it helps users make better and faster health decisions, and its accuracy can be further improved in the future with advanced AI models and larger medical datasets.

REFERENCES

- [1]. "A Symptom-Based Disease Prediction System Using Machine Learning" This paper explains how disease prediction can be done using symptom analysis and intelligent algorithms to improve early diagnosis.
- [2]. "Healthcare Chatbot for Disease Prediction and Treatment Recommendation" This research presents a system that uses symptoms to predict diseases and provide basic remedies and health suggestions.
- [3]. "An Intelligent Web-Based System for Disease Diagnosis" This paper discusses the use of web technologies and data analysis for predicting diseases based on user input.
- [4]. ASP.NET Core Documentation – Provides guidelines for building secure and scalable backend APIs. <https://learn.microsoft.com/en-us/aspnet/core/>
- [5]. Angular Documentation – Used for developing the frontend user interface and handling client-side logic. <https://angular.io/docs>
- [6]. SQL Server Documentation – Used for database design, data storage, and query management. <https://learn.microsoft.com/en-us/sql/>
- [7]. MDN Web Docs – Provides information on HTML, CSS, and JavaScript used for UI development. <https://developer.mozilla.org/>



- [8]. Kumar, A., & Clark, M. – Kumar and Clark’s Clinical Medicine, 10th Edition, Elsevier, 2020. This book provides medical knowledge about diseases, symptoms, and diagnosis methods.
- [9]. Harrison’s Principles of Internal Medicine, 20th Edition, McGraw Hill, 2018. It explains various diseases, symptoms, and clinical approaches used in healthcare.

