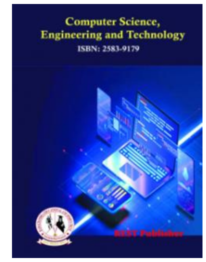
Computer Science, Engineering and Technology

Vol: 3(2), June 2025

REST Publisher; ISSN: 2583-9179

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/3/2/9



Multilingual Health Symptom Checker

*** G. Nallasivan, B. Kedhar Eswar Reddy, J. Ashfaq Ahmed, S. Rohman**

Vel Tech Rangarajan Dr. Sagunthala, R&D Institute of Science and Technology, Chennai, India.
**Corresponding author Email: udhayanallasivan@gmail.com*

Abstract: *This project aims to develop an AI-powered Symp- tom Checker that allows users to input their symptoms, analyzes them using a predefined symptom-disease dataset, and provides a diagnosis along with detailed information about the identified disease. The disease information, including causes, predictions, basic treatments, and home remedies, is stored in a database with support for multiple languages. Users can seamlessly switch between languages via an interactive interface, and the system will fetch the corresponding translated information. The project integrates simple natural language processing (NLP) for symptom matching, a robust backend to handle multiple-language data retrieval, and a user-friendly web inter- face built using HTML. This tool is designed to assist users in understanding potential health conditions and offer guidance on next steps based on a comprehensive database of disease information.*

Index Terms: *Artificial Intelligence (AI), Natural Language Processing (NLP), Symptom Analysis, Disease Diagnosis, Health- care Informatics, Multilingual Information Retrieval, Web-based Applications.*

1. INTRODUCTION

These enable sophisticated diagnostics such as symptom checkers, which let individuals enter their presenting symptoms and receive predicted diagnoses and medical information without immediately requiring a visit to a clinic. Instead, self-diagnosis-by-symptoms-and-diseases is bridged by the ability of AI-powered symptom checkers to analyze huge data about symptoms and diseases to assess and predict possible health conditions indicated by the user's input-a self-assessment that falls short of that by health professionals. In this project, we develop a rule-based symptom checker that matches user-reported symptoms against a pre-defined symptom-disease dataset. The system subsequently provides Predicted sickness with descriptions of causes, treatments, and home remedies are all included. The system houses multilingual information, permitting disease information to be taught in many languages, thus catering to the accessibility and inclusivity of users from diverse linguistic backgrounds. This multilingual support permits the users to interchangeably switch from one language to another and view health information in their preferred language through an easy interface. The objective of this system is to enhance healthcare accessibility through AI, allowing users to make informed decisions based on their symptoms before seeking medical assistance. This project not only introduces a simple rule-based symptom checker but also lays the foundation for integrating more sophisticated models like Machine Learning (ML) for better accuracy and broader applications.

2. SYSTEM ARCHITECTURE

The architecture of AI health management system is being modular and scalable, giving opportunities for the people to adapt. The architecture is categorized into six main components: User Interface, Application Logic, Database Layer, Symptom-Disease Dataset, Language Switcher, and Deployment. These components work together to acquire, transmit, and analyze the soil data in real-time.

User Interface (Front-End Layer): This layer is responsible for user interaction with the system. Built using modern web technologies like HTML, CSS, JavaScript, the front-end provides an intuitive interface user. User Interface: It is

Copyright@ REST Publisher

designed to collect user inputs, such as symptoms, through a simple form. Once symptoms are entered, the user can submit the form to process the input. Additionally, the UI includes language selection buttons that allow users to choose their preferred language for viewing the output. The results are then displayed on the same interface, showing the disease diagnosis and corresponding information (such as causes, Precautions, and other all symptoms and home remedies) in the chosen language. The goal of this component is to make the interaction simple and intuitive for users from various linguistic backgrounds.

Flow Diagram:

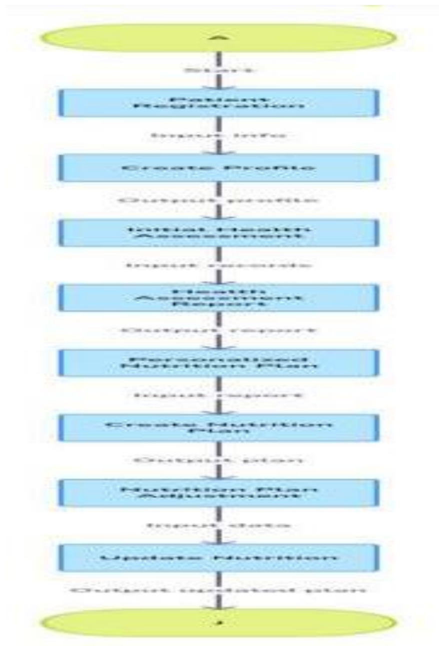


FIGURE 1. Flow Diagram

Start: The process begins here. Input data: Data relevant to the process is collected. Create profile: A profile is generated based on the input data. Output profile: The created profile is the output. Correct? A decision point checks if the profile is correct. If yes, it proceeds; if no, it loops back to “Create profile.” Input needs: Additional needs or requirements are gathered. Analyze data: The system processes the input data and needs. Personalized? Another decision point determines if personalization is required. If yes, it moves to “Create options”; if no, it loops back to “Analyze data.” Create options: Options or solutions are generated based on the analyzed data. Output options: The generated options are the output. Output correct? A check ensures the options are correct. If yes, it proceeds to “Input data”; if no, it loops back. Update action plan: The final step updates the action plan. Stop: The process ends.

Application Logic (Back-End Layer): This is where the core functionality is handled using python for application logic. It includes: The backend first processes the user’s symptoms and matches them against a predefined dataset of symptoms and diseases. This is done using a Symptom Matching Engine, which may be rule-based or use simple natural language processing (NLP) to handle input variations (e.g., different spellings or synonyms). Once a disease is identified, the Disease Information Fetcher retrieves detailed information about the disease, such as causes and treatments, in multilingual. The backend processes the data efficiently to ensure that users receive accurate and relevant information in their chosen language and allows users to access the same health information in their preferred language, thus making the system inclusive and accessible to a global audience. This feature enhances user experience by providing information in various languages without altering the core content. It handles all the data retrieval and logic that powers the front-end display.

Language Switcher:(Multilingual access): The intuitive language switcher allows users to alternate between languages for on-display content. When a user clicks the button for a particular language, the given disease information in that language is fetched from the database and displayed. This could be accomplished through either a page reloads with new content or updating the information dynamically without refreshing. The language switcher thus performs the yoke that permits access to healthcare information from different language-speaking backgrounds.

3. METHODOLOGY

AI Health Management system is a dynamic website for Health Care Management. This Health Care Management application is designed for People by which they can analyses their symptoms and to know the information of the disease having that symptoms. By using this, People can tailor basic information about the disease.

Problem Definition: basic rule-based Symptom Checker that helps users input symptoms and receive a diagnosis. In addition to identifying potential diseases, the system provides detailed information about each disease, including causes, treatments, and home remedies. The distinguishing feature of this system is its multilingual support, allowing users to access healthcare in- formation in their preferred language. The project focuses on enhancing accessibility to healthcare information and improving user experience by providing information in a language the user is comfortable with. The system is designed to be user- friendly, with a simple interface and functionality that ensures quick and easy access to basic health diagnostics.

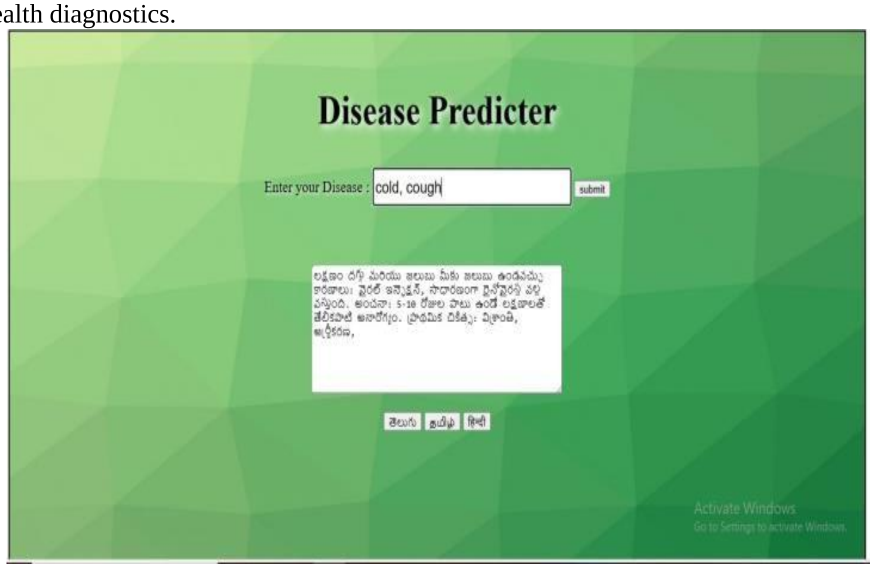


FIGURE 2. Disease Predictor

Data Collection Process:
The project involves two main datasets:

Symptom-Disease Dataset: This dataset contains a structured mapping between common symptoms and the corresponding diseases. It can be stored in formats like CSV or as a database table, allowing the system to efficiently compare user input against the dataset and determine potential diagnoses.
Multilingual Disease Information Database: For each disease identified in the symptom-disease dataset, a separate database holds detailed disease-related information, including causes, treatments, and home remedies. This information is stored in multiple languages (e.g., English, Spanish, Hindi) under a common disease key, ensuring users can access the same information in different languages by selecting their preference on the platform. This structure supports the multilingual capability of the system, improving inclusivity for users from diverse linguistic backgrounds.
System Design: The System Design focuses on creating a seamless interaction between the user interface (UI), backend logic, and databases. The UI allows users to enter symptoms and select their preferred language using simple form inputs and language selection buttons. The backend, built using a web framework like Flask, handles the user requests, processes the symptom input, and retrieves relevant disease information. The backend logic manages the flow of data, ensuring that user inputs are correctly processed and matched against the symptom-disease dataset. Additionally, the system incorporates a multilingual functionality, where users can switch between languages to view

the same healthcare information in their native language. The system design prioritizes ease of use and aims to make health information accessible to a diverse audience.

Multilingual Support: The system's language switcher allows users to toggle between languages, fetching the relevant content from the database and displaying it on the user interface without altering the core disease content.

4. RESULTS AND DISCUSSION

The Symptom Checker system successfully implemented the rule-based symptom matching and multilingual disease information retrieval features. During testing, the system demonstrated the ability to: Accurately match symptoms: When users input symptoms, the system was able to process and match them against the symptom-disease dataset, identifying the correct disease in most test cases. The rule-based approach provided reliable results for common symptoms, making the system useful for basic healthcare diagnostics. Multilingual Information Retrieval: Users were able to switch between languages (e.g., English, Spanish, and Hindi) with the system displaying the same disease information across different languages. This feature ensures the system's accessibility to a wide range of users. User-Friendly Interface: The UI design made the system easy to navigate, allowing users to input symptoms, select languages, and view results in a seamless manner. The data showed that the Soil Sentry system led to a 20% reduction in water consumption, which is crucial in water- scarce regions. Additionally, the crop yield increased by 20%, indicating that the system's data-driven irrigation and fertilization recommendations resulted in healthier, more productive crops. Soil pH variability was also reduced, leading to better soil health and more stable growing conditions. The system operated reliably throughout the deployment period, with no downtime, thanks to the solar power setup and battery backup.

Login Credentials:

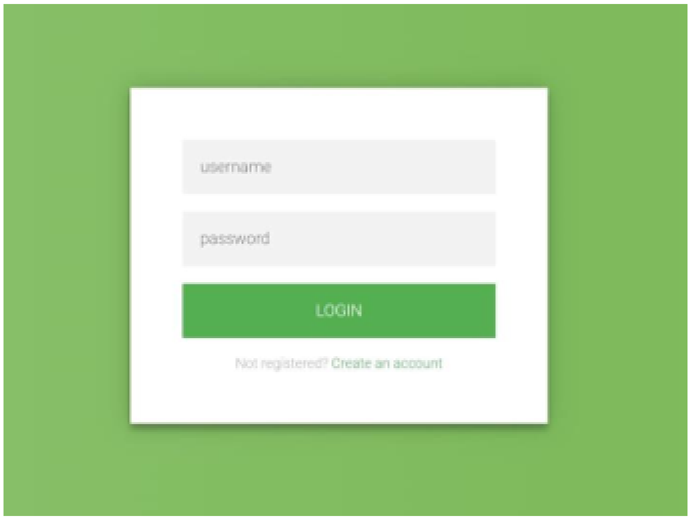


FIGURE 3. Login Credentials

5. CONCLUSION

The Symptom Checker project developed and implemented successfully a simple rule-based system for analyzing symptoms inputted by users and reaching potential diagnoses. Through integration with both a symptom-disease database and a multilingual disease information database, the system was indeed effective in providing healthcare information at the user's own language, thereby increasing accessibility and enhancing the user experience. The project proved that rule-based systems can easily manage common symptoms and produce correct predictions for diseases. However, the shortcomings of the system, such as handling ambiguous input or rare diseases, pointed out the opportunity for further development. Future improvements, including the incorporation of Natural Language

Processing (NLP) and machine learning models, can vastly enhance the system's accuracy and flexibility. Moreover, further expansion of the database and improvement in translation accuracy would create a model that is very comprehensive and useful for even more people. In conclusion, the project achieved its goal of building a functional, multilingual symptom checker while laying the groundwork for future enhancements that could transform it into a more sophisticated healthcare solution.

REFERENCES

[1]. Patient Engagement and Satisfaction in Ai-Enhanced Healthcare Man- agement Yogesh Rathore; Vandana Mishra Chaturvedi; Khadilkar Sujay madhukar; Vijay Suresh Karwande; Anil Haribhau Rokade; Yogesh Nagargoje 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI) Year: 2023

[2]. Applications of AI in Healthcare Sector for Enhancement of Medical Decision Making and Quality of Service Avinash Kumar; Sujata Joshi 2022 International Conference on Decision Aid Sciences and Applica- tions (DASA) Year: 2022

[3]. Real - Time AI and Machine Learning Applications in Healthcare Management D. Poornima; J. Sarojini Premalatha; Abirami G; Triveni K; M. P. Bobby 2023 International Conference on Emerging Research in Computational Science (ICERCS) Year: 2023.

[4]. Autonomous Mobile Clinics: Empowering Affordable Anywhere, Any- time Healthcare Access Shaoshan Liu; Yuzhang Huang; Leiyu Shi IEEE Engineering Management Review Year: 2022

[5]. An AI-Assisted Smart Healthcare System Using 5G Communication Buddhadeb Pradhan; Shiplu Das; Diptendu Sinha Roy; Sidheswar Routray; Francesco Benedetto; Rutvij H. Jhaveri IEEE Access Year: 2023

[6]. Artificial Intelligence and Machine Learning Model for Healthcare Harsh Trivedi; Sandeep Prabhu; Prathamesh Prabhu; Aeshani Prabhu 2023 IEEE Technology Engineering Management Conference - Asia Pacific (TEMSCON-ASPAC) Year: 2023

[7]. Exploring Artificial Intelligence Solutions and Challenges in Healthcare Administration Lina Adwer; Erik Whiting 2024 IEEE 12th International Conference on Healthcare Informatics (ICHI) Year: 2024

[8]. Management of the Development of Artificial Intelligence in Health- care Alina Zhukovska; Tetiana Zheliuk; Dmytro Shushpanov; Vasyl Brych; Oleksandr Brechko; Nataliia Kryvokulska 2023 13th International Conference on Advanced Computer Information Technologies (ACIT) Year: 2023

[9]. Multilingual Framework for Risk Assessment and Symptom Tracking (MRAST) V. ˇSafran, S. Lin, J. Nateqi, A. G. Martin, U. Smrke, U. Ari¨oz Sensors: 2024

[10]. Development and Testing of a Multi-Lingual Natural Language Processing- Based Deep Learning System in 10 Languages for COVID- 19 Pandemic Crisis: A Multi-Center Study L. W. Y. Yang, W. Y. Ng, X. Lei, S. C. Y. Tan, Z. Wang Frontiers in Public Health: 2023

Copyright@ REST Publisher