International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Medi Diagnose AI

Sudha Laxmaiah[1], Pagidimarri Akhila[2], Ryaka Deekshita[3], Boyapally Hritika Reddy[4]

[1]Assistant Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India

[2,3,4]B.Tech Students; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India

Mail Id; sudha.nama@slv-edu.in[1],pagidimarriakhila20@gmail.com[2],deekshitaryaka@gmail.com[3], hrithikaboyapalli@gmail.com[4]

Abstract

The increasing demand for rapid and accessible healthcare information has led to the development of intelligent digital health systems. This paper presents MediDiagnose AI, a web-based healthcare platform designed to assist users in identifying potential diseases based on their symptoms. The system allows users to create accounts, log in securely, input symptoms, and receive preliminary diagnostic suggestions generated through rule-based reasoning mechanisms. In addition to symptom analysis, the platform integrates supportive healthcare features including a medical chatbot for basic health inquiries, patient information management, disease knowledge resources, and automated report generation.The application is implemented using modern web technologies, with React powering the frontend interface, Python with the Flask framework handling backend processing, and Supabase (PostgreSQL) providing secure and scalable database management. The architecture ensures efficient data processing, reliable authentication, and a responsive user experience.MediDiagnose AI aims to improve healthcare accessibility by providing users with immediate guidance regarding possible medical conditions, particularly for minor or early-stage symptoms. While the system is not intended to replace professional medical consultation, it functions as a supportive tool that encourages early awareness and informed decision-making. The proposed system demonstrates how intelligent web applications can contribute to digital health services by offering fast, user-friendly, and informative preliminary diagnosis support.

Keywords

Digital Healthcare, Symptom-Based Diagnosis, Web-Based Medical System, Artificial Intelligence in Healthcare, Medical Chatbot, Disease Prediction, Healthcare Information Systems, Flask, React, Supabase.

Introduction

MediDiagnose AI is an intelligent web-based healthcare application developed to support early disease identification through symptom analysis and optional CT scan image submission. In modern society, many individuals postpone visiting healthcare professionals due to limited time, insufficient awareness, or lack of immediate access to medical services. Such delays may result in the late detection of diseases and potentially serious health complications. The proposed system addresses this challenge by allowing users to input their symptoms and optionally upload medical images, enabling the platform to generate preliminary diagnostic suggestions using rule-based reasoning and simulated artificial intelligence techniques.The system is designed to simplify the diagnostic support process by minimizing manual efforts and improving efficiency in healthcare information management. MediDiagnose AI integrates several modules, including patient registration and management, symptom-based diagnosis, automated report generation, disease information access, and a medical chatbot that assists users with basic healthcare queries. Through these functionalities, the system provides an organized platform that supports both patients and healthcare providers in accessing health information and making timely decisions. By leveraging modern web technologies, the platform delivers a responsive and user-friendly interface that facilitates quick interaction and reliable data handling.

Scope of the System

The primary objective of MediDiagnose AI is to develop a digital healthcare platform capable of performing preliminary disease prediction based on user-provided symptoms and optional CT scan images. The system is intended to assist healthcare professionals in conducting faster initial assessments while also helping manage patient data efficiently. By automating parts of the diagnostic support process, the platform contributes to improved healthcare workflow and better information accessibility.The application is designed as a web-based system so that it can be deployed in hospitals, clinics, and rural healthcare environments where medical resources may be limited. Through internet accessibility, users can interact with the system from various locations and obtain preliminary health insights without immediate physical consultation. Furthermore, the system architecture is developed with scalability in mind, allowing future enhancements such as the integration of advanced artificial intelligence or

machine learning models for automated CT scan analysis and the inclusion of a broader range of disease prediction capabilities.

Proposed System

The proposed system, MediDiagnose AI, introduces a digital solution that addresses the limitations of traditional healthcare diagnosis methods by providing an intelligent and centralized healthcare platform. The system enables users to register securely, submit their symptoms, and optionally upload CT scan images for preliminary evaluation. Based on the provided inputs, the system generates possible diagnostic suggestions using rule-based analysis while also maintaining patient information in a structured database.The platform also incorporates additional features such as automated report generation, disease information resources, and a medical chatbot that assists users with basic health-related inquiries. These features help improve user engagement while offering quick access to essential healthcare information. By centralizing patient records, diagnosis support, and communication tools within a single platform, the proposed system enhances the organization and accessibility of medical data.Furthermore, the system is designed to support healthcare providers by streamlining patient data management and enabling faster decision-making during the initial stages of diagnosis. Although MediDiagnose AI is not intended to replace professional medical consultation, it serves as a supportive tool that promotes early health awareness and encourages users to seek medical attention when necessary. Through its scalable architecture, the system also provides opportunities for future integration of advanced artificial intelligence techniques for more accurate and automated medical analysis.

System Requirements

Functional Requirements

The MediDiagnose AI system is designed with two primary user roles: administrator and general user. The administrator module manages the overall functionality and monitoring of the platform. Administrators are responsible for registering and logging into the system securely and accessing the administrative dashboard that provides an overview of system activity. Through this interface, administrators can view and manage patient records, examine diagnosis reports generated by the system, and monitor system data to ensure proper operation. The module also provides secure logout functionality to maintain data security and session control.The user module focuses on enabling patients or healthcare users to interact with the diagnostic system efficiently. Users can create accounts and authenticate themselves through a secure login mechanism. After authentication, users can select or create patient records and enter symptom information. The system also allows optional uploading of CT scan images for enhanced diagnostic analysis. Once the necessary inputs are provided, the platform generates a preliminary diagnosis based on predefined rule-based logic. Users can then view the diagnostic results, download generated reports, and interact with a medical chatbot that provides basic health-related assistance. Secure logout functionality is also provided to protect user data and system access.

Non-Functional Requirements

Non-functional requirements define the quality attributes that ensure the system operates efficiently and reliably. Scalability is an important aspect of the system, as the application must be capable of supporting an increasing number of users without performance degradation. Usability is also essential, and the platform is designed with a simple and intuitive interface that allows users to navigate the system easily. Reliability is another critical factor, requiring the system to maintain consistent performance and availability while minimizing system failures.Security is implemented to protect sensitive patient data and to prevent unauthorized access through authentication and access control mechanisms. Performance requirements ensure that the system provides quick responses when generating diagnoses or interacting with the chatbot. Compatibility is considered during development so that the application functions correctly across modern web browsers. Additionally, portability ensures that the system can operate on different computing environments without requiring major modifications.

Computational Resources

Software Resources

Software resources define the technological components required to develop and operate the MediDiagnose AI system. The software requirements specification outlines what the system should accomplish without focusing on the internal implementation details. These requirements provide a foundation for system planning, cost estimation, development scheduling, and monitoring project progress throughout the development lifecycle.The implementation utilizes modern programming technologies and development tools. Python version 3.8 or higher is used as the primary backend programming language due to its flexibility and strong ecosystem of libraries. The frontend interface is developed using JavaScript with the React framework, which enables the creation of dynamic and responsive user interfaces. The system operates on standard operating systems such as Windows 11 or Linux environments. Supabase, which is built on PostgreSQL, is used as the database management system to store and manage patient records, diagnosis results, and related information. Visual Studio Code serves as the integrated development environment for coding and debugging, while

additional libraries such as jsPDF are used for generating downloadable reports.

Hardware Resources

Hardware resources refer to the physical computing components required to support the operation of the system. The MediDiagnose AI application is designed to run efficiently on standard computing hardware without requiring specialized equipment. A processor equivalent to an Intel i3 with a minimum clock speed of 2.0 GHz is sufficient to support the system operations. At least 8 GB of RAM is recommended to ensure smooth performance during application execution and data processing. Additionally, a storage capacity of approximately 512 GB is suggested to accommodate the operating system, development tools, and stored application data.

Software Process Model

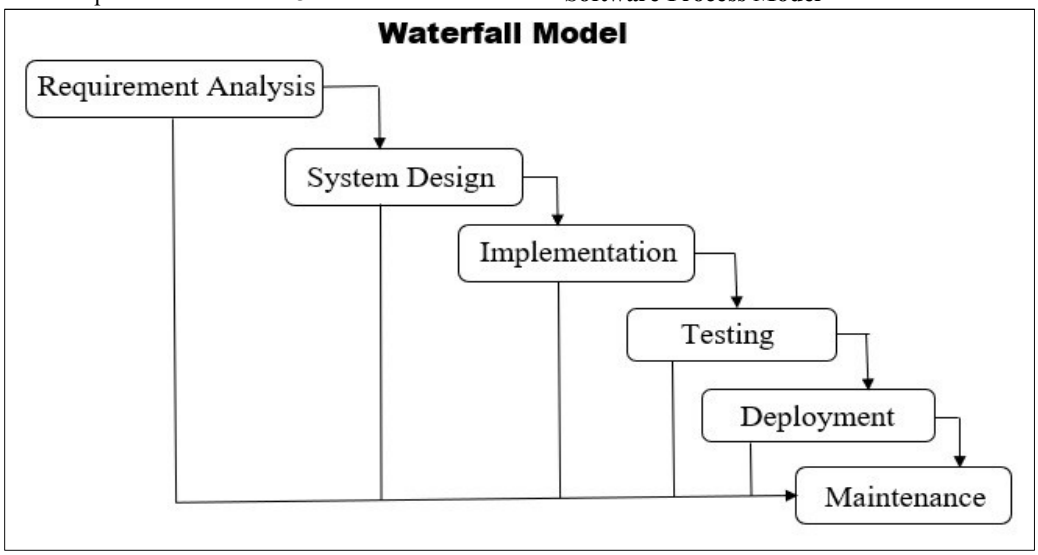


Fig.1 Spiral Model

A software process model provides a structured framework for organizing and managing the software development lifecycle. It outlines the sequence of activities involved in designing, implementing, testing, and deploying a software application. Various development models exist, each suited for different project requirements and complexities.Common process models include the Waterfall model, which follows a sequential development approach; the V-model, which emphasizes verification and validation throughout the development stages; the Incremental model, which develops the system in smaller functional units; the Spiral model, which combines iterative development with risk analysis; and the Agile model, which focuses on flexibility, collaboration, and rapid adaptation to changes. The selection of an appropriate development model depends on factors such as project scope, complexity, and development requirements. In many modern projects, iterative and flexible approaches are preferred to support continuous improvements during the development process.

System Design

System design focuses on defining how the proposed solution will be structured and implemented. This stage begins with identifying the problem and establishing the objectives of the system. It includes careful planning for data collection, storage, and processing to ensure efficient operation. Feature engineering and model design are also important aspects of this phase, as they determine how the system processes inputs and generates outputs. Selecting suitable algorithms and designing system architecture are essential for achieving accurate and efficient results.The design phase also involves establishing appropriate training and evaluation strategies to measure system performance using relevant metrics. Additionally, the development of intuitive user interfaces ensures that users can interact with the system easily. Deployment planning is another important component of the design stage, as it ensures smooth integration of the system with existing technological environments. A well-defined design process ultimately contributes to the reliability, scalability, and effectiveness of the application.

System Architecture

The architecture of a project represents the structural organization of system components and the sequence in which requests are processed. It describes how different modules interact and collaborate to perform system operations. Architectural descriptions provide a formal representation of system structure, enabling developers to understand system behavior and evaluate design decisions.The architecture of the MediDiagnose AI system can be categorized into two major types: software architecture and technical architecture.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Software Architecture

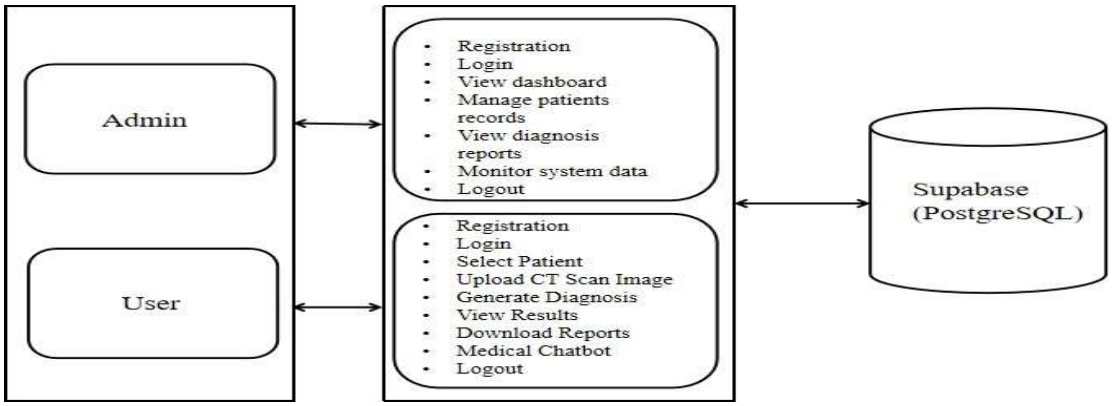


Fig.2 Software Architecture

Software architecture defines the structural design of the application and its internal components. Effective architectural design helps reduce software vulnerabilities and enhances the reliability of the system. During development, software architecture tools assist in identifying potential flaws or weaknesses in the system structure. These tools allow developers to analyze the design, evaluate potential security threats, and detect possible gaps in the application's security framework.By implementing well-designed architectural strategies, developers can ensure that the system remains stable, secure, and efficient. The architectural design also enables teams to detect and resolve issues during development rather than after deployment, thereby improving overall system quality.

Technical Architecture

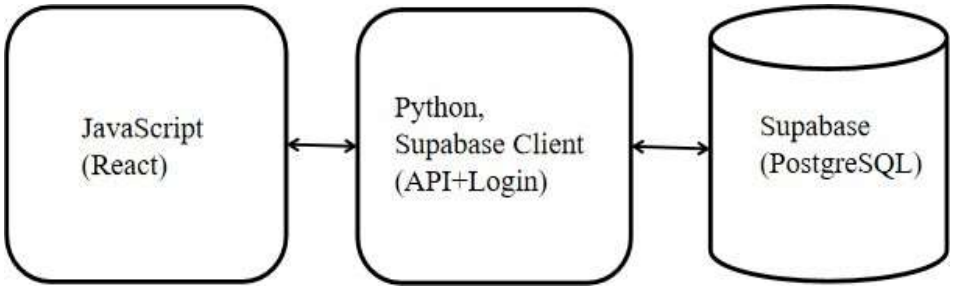


Fig.3Technical Architecture

Technical architecture focuses on the technological infrastructure required to support the system. It defines how hardware, software platforms, databases, and network components interact to deliver the required functionality. This architectural approach provides a blueprint that describes how different system components are arranged and how they communicate with each other to satisfy system requirements. Proper technical architecture ensures system scalability, reliability, and efficient resource utilization.

Database Design

The database component of the MediDiagnose AI system is responsible for storing and managing patient information, disease data, and diagnostic results. Patient data tables store personal details, symptoms, and associated diagnostic reports. Disease-related tables maintain structured information about medical conditions and their associated symptoms. The use of a structured relational database enables efficient data retrieval, storage, and management while ensuring data consistency and integrity.

Implementation

Technologies Used

The implementation of the MediDiagnose AI system relies on modern web technologies to ensure efficiency and scalability. Python is used as the primary programming language for backend development due to its simplicity, readability, and extensive library support. The language supports object-oriented programming, offers automatic memory management, and provides platform independence.The frontend of the system is developed using the React.js library, which enables the creation of dynamic user interfaces. React allows the application to update content dynamically without reloading pages, resulting in improved performance and user experience.For backend processing, the Flask framework is used to implement server-side logic and manage communication between the frontend and the database. Flask handles the diagnostic logic,

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

processes user requests, and returns responses to the client application.Supabase, built on PostgreSQL, is used as the database management system. It stores patient records, diagnostic outcomes, and related data in a structured format while providing secure and centralized data management.

Testing

Overview of Testing

Software testing is an essential process used to evaluate whether a software system meets specified requirements and functions correctly. The objective of testing is to detect defects, improve reliability, and ensure that the system performs as expected. In the MediDiagnose AI system, testing is conducted to verify the correct functioning of modules such as user authentication, patient data management, diagnosis generation, report creation, and chatbot interactions.Testing ensures that the system produces accurate diagnostic results based on user input and maintains stable performance under different conditions. By identifying and correcting potential issues during development, testing contributes to improved system reliability and user satisfaction.

Dimensions of Testing

Different dimensions of testing are considered to ensure comprehensive system evaluation. Functional testing verifies that all system features operate according to their intended behavior. Performance testing evaluates how efficiently the system responds to user requests and manages multiple interactions. Usability testing ensures that the interface remains intuitive and easy to use. Security testing focuses on protecting sensitive patient information and preventing unauthorized access. Reliability testing confirms that the system operates consistently without unexpected failures.

Stages of Testing

Testing is conducted in several stages throughout the software development process. Unit testing focuses on evaluating individual modules independently to ensure they produce correct outputs for specific inputs. Integration testing examines the interaction between system components, verifying that the frontend, backend, and database communicate effectively. System testing evaluates the complete application to confirm that it meets all functional and non-functional requirements. Finally, acceptance testing is conducted to determine whether the system satisfies user expectations and is ready for deployment in real-world environments.

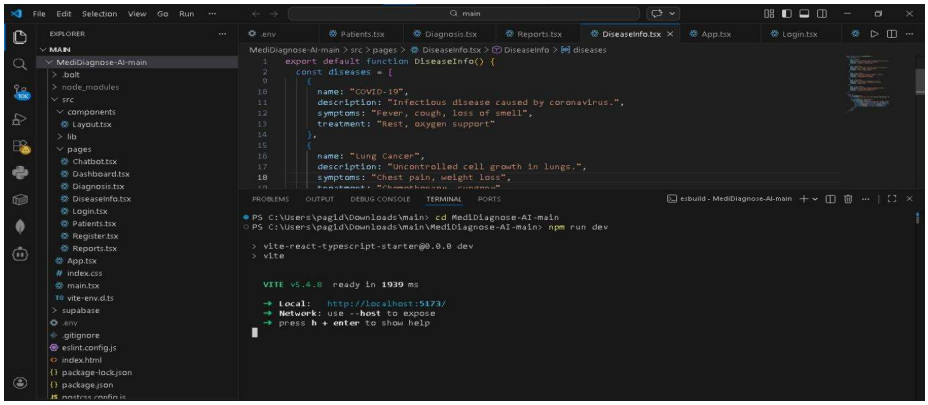
Types of Testing

Two major testing approaches are applied in this project. Black box testing evaluates system functionality without considering the internal code structure, focusing instead on input-output behavior. In contrast, white box testing examines the internal logic of the system, enabling developers to verify code execution paths and ensure proper implementation of algorithms.

Test Cases

Test cases are designed to validate system functionalities such as user registration, login, patient data entry, CT scan image upload, diagnosis generation, report creation, and chatbot responses. Each test case specifies the input conditions, expected outputs, and actual results obtained during testing. Successful execution of these test cases confirms that the system operates according to the specified requirements and is capable of supporting real-world usage.

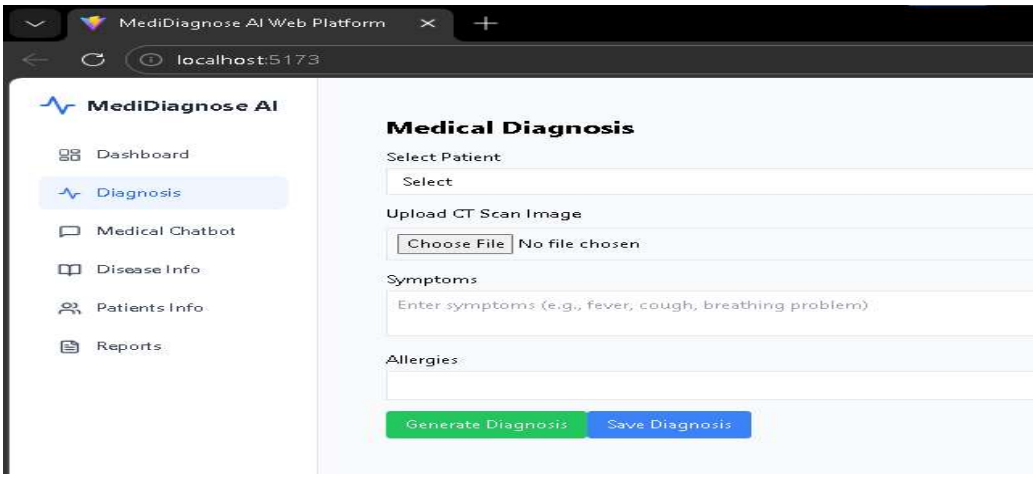
Screenshots



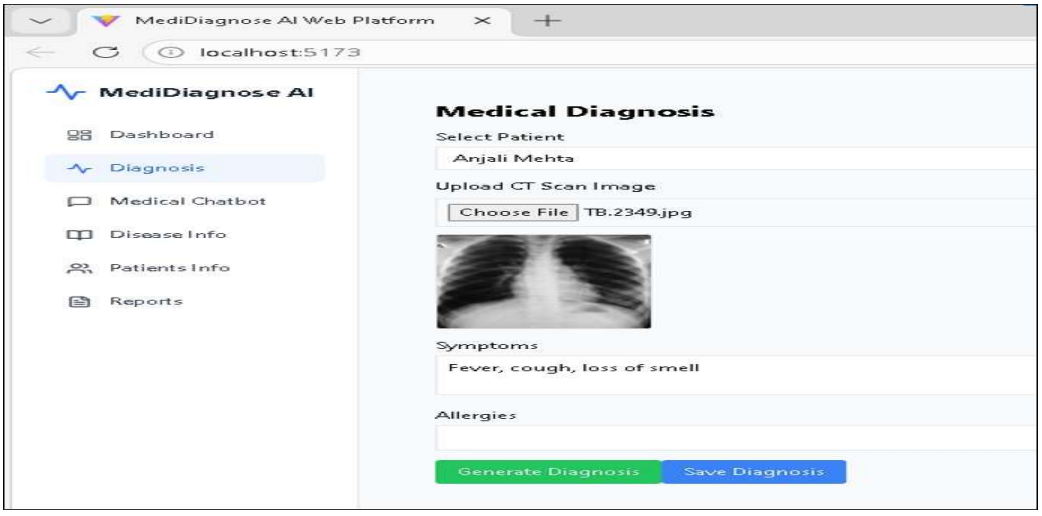
Screenshot 1 Run the Application & URL generated

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Screenshot 2 Registration Page

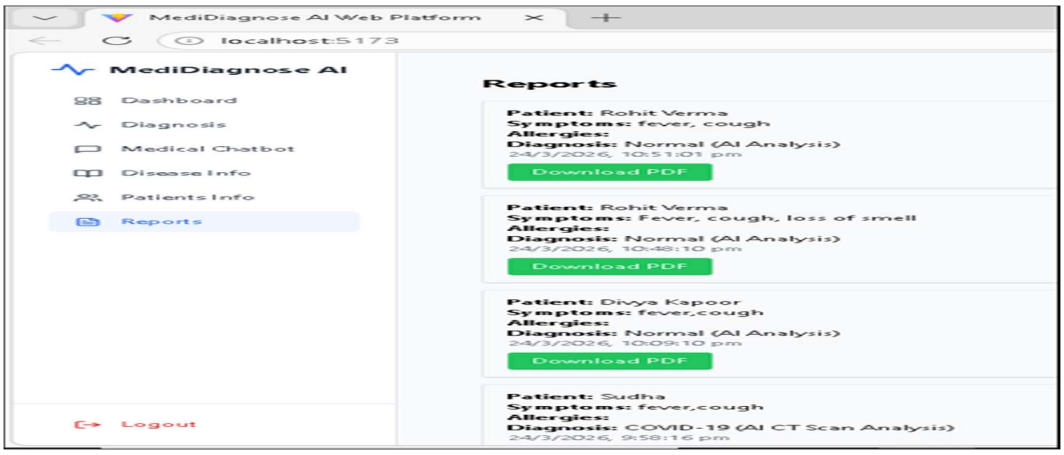


Screenshot 3 Diagnosis Page

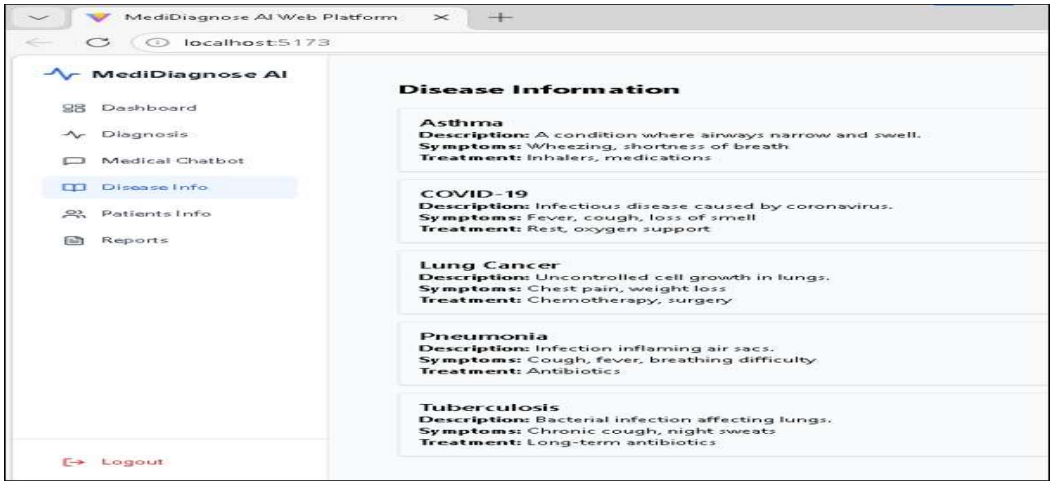


Screenshot 4 Diagnosis Result

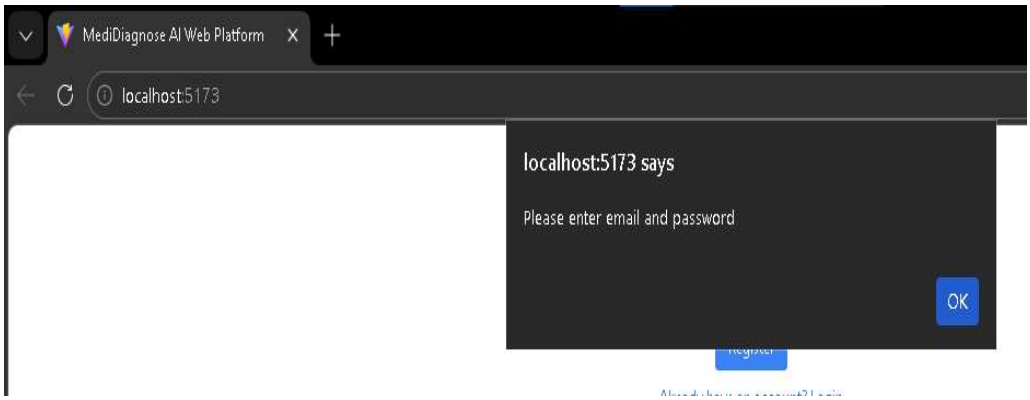
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026



Screenshot 5 Reports Page



Screenshot 6 Disease Information Page



Screenshot 7 Invalid credentials

Conclusion

MediDiagnose AI presents a web-based intelligent healthcare support system developed to assist in the early identification of diseases through symptom analysis and optional CT scan image input. The system provides preliminary diagnostic suggestions that help users gain quick insights into possible health conditions without immediately relying on manual consultation processes. By automating parts of the diagnostic workflow, the platform helps reduce the time required for initial health assessment and supports both patients and healthcare professionals in making informed decisions.The platform integrates multiple healthcare functionalities within a unified environment, including patient information management,

automated diagnosis generation, report creation, and a medical chatbot for basic health assistance. These features collectively contribute to improved operational efficiency and more organized healthcare data management. The system also enhances accessibility to health-related information, particularly for users who may have limited access to immediate medical consultation.Overall, MediDiagnose AI demonstrates how modern web technologies and intelligent diagnostic support tools can contribute to improving digital healthcare services. The system offers a user-friendly interface, reliable data handling, and efficient processing of user inputs, making it a practical solution for preliminary healthcare guidance. Although it is not intended to replace professional medical diagnosis, it serves as a supportive platform that promotes early awareness and encourages timely medical consultation, thereby contributing to better patient care and decision-making.

Future Scope

The MediDiagnose AI platform has significant potential for future improvements through the integration of advanced technologies and additional healthcare features. One possible enhancement involves incorporating machine learning and artificial intelligence algorithms trained on large medical datasets to improve the accuracy and reliability of disease prediction. Such improvements would allow the system to analyze complex patterns in medical data and provide more precise diagnostic insights.Another important direction for future development is the creation of a mobile-based version of the system. A mobile application would improve accessibility by enabling users to interact with the platform through smartphones and other portable devices, making healthcare support available anytime and anywhere. Furthermore, integrating real-time communication features such as online doctor consultations could strengthen the system by allowing users to connect directly with healthcare professionals when necessary.Additional enhancements may include the implementation of voice-enabled chatbot functionality to improve user interaction and accessibility for individuals who prefer voice-based communication. Deploying the system on cloud infrastructure could also improve scalability, performance, and system availability. Moreover, advanced data protection mechanisms and security frameworks can be incorporated to ensure the confidentiality and safety of sensitive patient information. These future developments would further strengthen the system's capability to support modern digital healthcare services.

References

[1] A. Mabrouk, et al., "Pneumonia detection on chest X-ray images using an ensemble of deep convolutional neural networks," *arXiv preprint arXiv:2312.07965*, 2024.
[2] T. Rahman, M. Rahman, S. M. Z. Ali, et al., "Exploring the effectiveness of transfer learning with convolutional neural networks for pneumonia detection from chest X-ray images," *PLOS ONE*, vol. 15, no. 7, 2024.
[3] T. Sanida, et al., "A lightweight convolutional neural network for lung disease detection using chest X-ray images," *Applied Intelligence*, 2024.
[4] S. Singh, et al., "Efficient pneumonia detection using vision transformers on chest X-ray images," *Scientific Reports*, 2024.
[5] H. Lamouadene, et al., "Detection of COVID-19 and lung diseases using deep learning and transfer learning techniques," *Computers in Biology and Medicine*, 2025.
[6] World Health Organization, "Tuberculosis and pneumonia statistics 2025," WHO, 2025. [Online]. Available: https://www.who.int
[7] Kaggle, "Chest X-ray Images (Pneumonia) dataset," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia