International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

AI-Based Fake Media Detection Using Machine Learning and Natural Language Processing

BALE HEMANTH KUMAR

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

B. Suryanarayana Murthy

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The integration of Artificial Intelligence (AI) in healthcare has significantly improved patient care, accessibility, and decision-making processes. This project presents an AI-Based Healthcare Chatbot with an integrated Patient-Doctor Management System designed to provide preliminary medical assistance and streamline communication between patients and healthcare providers.

The system is developed using the Django web framework and incorporates user authentication for three roles: patients, doctors, and administrators. Patients can register, log in, and interact with an AI-powered chatbot that provides medical advice based on user input. The chatbot utilizes Natural Language Processing (NLP) techniques to understand patient symptoms and generate appropriate responses.

A key feature of the system is the Intelligent Triage Engine, which evaluates patient input and classifies cases into severity levels such as EMERGENCY, URGENT, ROUTINE, and SELF-CARE. This classification helps prioritize patient cases and ensures that critical conditions receive immediate attention. The chatbot responses, along with severity levels, are stored in a database as chat logs.Doctors can access a dedicated dashboard where patient queries are displayed and sorted based on severity. This allows healthcare providers to focus on high-priority cases first, improving efficiency and response time. The system also supports real-time interaction through an API-based chat interface.

The application emphasizes automation, scalability, and usability. It reduces the burden on healthcare professionals by handling routine queries while ensuring that serious cases are escalated appropriately. The modular architecture allows for future enhancements such as integration with electronic health records, advanced AI models, and telemedicine features.This project demonstrates how AI can enhance healthcare delivery by providing quick, reliable, and accessible medical assistance while improving communication between patients and doctors.

Keywords: Healthcare Chatbot, Triage System, Django, Artificial Intelligence, Patient Management, Doctor Dashboard, NLP, Medical Decision Support

I. INTRODUCTION

Healthcare systems worldwide face challenges such as increasing patient demand, limited medical resources, and the need for timely diagnosis. Traditional healthcare services often require patients to visit hospitals or clinics, which can be time-consuming and inconvenient, especially for minor health concerns. With advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP), healthcare services can now be partially automated to improve accessibility and efficiency.This project focuses on developing an AI-Based Healthcare Chatbot integrated with a Patient-Doctor Management System. The system allows patients to interact with a chatbot that provides preliminary medical advice based on symptoms described in natural language. The chatbot uses NLP techniques to understand user input and generate relevant responses.

One of the core components of the system is the Triage Engine, which analyzes patient messages and assigns a severity level. This helps in prioritizing cases and ensuring that critical patients receive immediate attention. The system categorizes cases into EMERGENCY, URGENT, ROUTINE, and SELF-CARE, enabling efficient resource allocation.The application is built using Django, a powerful Python web framework that supports rapid development and scalability. It includes user authentication mechanisms for patients, doctors, and administrators. Patients can register and interact with the chatbot, while doctors can access a dashboard displaying patient queries sorted by severity.

The system also stores chat logs, which can be used for analysis and improving healthcare services. Additionally, the use of APIs enables real-time communication between the frontend and backend.The primary objective of this project is to provide an accessible and efficient healthcare solution that reduces the workload on medical professionals while ensuring timely assistance for patients. By leveraging AI technologies, the system enhances healthcare delivery and provides a foundation for future advancements such as telemedicine integration and predictive analytics.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

The application of Artificial Intelligence in healthcare has been widely studied, particularly in areas such as medical diagnosis, patient monitoring, and decision support systems. Healthcare chatbots have gained popularity as they provide instant responses and reduce the burden on healthcare professionals.Early chatbot systems were rule-based, relying on predefined scripts and keyword matching. While these systems were simple to implement, they lacked flexibility and were unable to handle complex queries. With the

advancement of Natural Language Processing (NLP), modern chatbots can understand context and generate more accurate responses.

Machine learning and deep learning models have been widely used in healthcare chatbots. Techniques such as Support Vector Machines (SVM), Decision Trees, and Neural Networks have been applied for symptom classification and disease prediction. Recent developments in transformer-based models have further improved chatbot performance by enabling better understanding of context and language nuances.Triage systems have also been extensively researched. Traditional triage methods involve manual assessment by healthcare professionals, which can be time-consuming and subjective. Automated triage systems use machine learning algorithms to classify patient cases based on severity. These systems help prioritize critical cases and improve resource allocation.

Web-based healthcare systems have been developed using frameworks like Django, which provide features such as authentication, database management, and API integration. These systems enable secure and scalable healthcare applications.Despite advancements, challenges remain, including data privacy, accuracy of predictions, and handling complex medical conditions. Ensuring the reliability and safety of AI-based healthcare systems is crucial.The proposed system builds upon existing research by integrating chatbot functionality, triage classification, and patient-doctor interaction into a single platform. This unified approach improves efficiency and provides a comprehensive solution for modern healthcare needs.

III. EXISTING SYSTEM

The existing healthcare systems primarily rely on manual processes and direct interaction between patients and healthcare providers. Patients typically need to visit hospitals or clinics to consult with doctors, which can be time-consuming and inconvenient. This approach often leads to overcrowding in healthcare facilities and delays in providing medical attention.

Some digital healthcare solutions have been developed, including telemedicine platforms and basic chatbot systems. However, many of these systems are limited in functionality. Traditional chatbots are often rule-based and lack the ability to understand complex user queries. They provide generic responses and are not capable of performing intelligent analysis or classification.Existing triage systems are usually manual, requiring healthcare professionals to assess patient conditions and prioritize cases. This process can be subjective and prone to human error. Additionally, it may not be efficient in handling a large number of patients.

Another limitation of current systems is the lack of integration between patient interaction and doctor dashboards. Many systems do not provide real-time data sharing or prioritization of cases based on severity.Overall, existing systems face challenges such as limited automation, lack of scalability, and inefficient resource utilization. These limitations highlight the need for an AI-based solution that can automate patient

interaction, perform intelligent triage, and improve communication between patients and doctors.

IV. PROPOSED METHOD

The proposed system is an AI-driven Healthcare Chatbot integrated with a Patient-Doctor Management System designed to provide automated medical assistance and intelligent case prioritization. The system leverages Natural Language Processing (NLP) and machine learning techniques to understand patient queries and generate appropriate medical advice.The core component of the system is the **Triage Engine**, which analyzes user input and classifies cases into severity levels such as EMERGENCY, URGENT, ROUTINE, and SELF-CARE. This classification enables efficient prioritization of patients and helps doctors focus on critical cases first. Research shows that AI-based triage systems can significantly improve patient prioritization and healthcare resource allocation .

The system supports three types of users: patients, doctors, and administrators. Patients can interact with the chatbot to receive preliminary medical advice. Doctors can access a dashboard that displays patient queries sorted by severity, ensuring efficient decision-making.The chatbot uses NLP-based models to process symptoms and generate meaningful responses. Modern AI chatbots can understand human language and provide real-time assistance, improving accessibility to healthcare services . The system also stores chat logs, enabling data analysis and continuous improvement.

Additionally, the system includes a web-based interface built with Django, ensuring scalability and maintainability. It supports real-time communication through APIs and secure authentication for all users.Overall, the proposed system enhances healthcare delivery by automating routine tasks, reducing workload on doctors, and providing timely assistance to patients. It also lays the foundation for future enhancements such as integration with electronic health records and advanced AI models.

V. IMPLEMENTATION

The implementation of the AI-based Healthcare Chatbot system is carried out using Python and the Django web framework. The system follows a modular architecture to ensure scalability, maintainability, and efficient performance.The application is built using Django's Model-View-Template (MVT) architecture. The **models** define the database structure, including user roles (patient, doctor, admin) and chat logs. The **views** handle user requests, authentication, chatbot interaction, and triage processing. The **templates** provide the front-end interface for user interaction.

User authentication is implemented using Django's built-in authentication system. Separate registration forms are provided for patients and doctors. After successful login, users are redirected to role-specific dashboards.The chatbot functionality is implemented

using a dedicated module (chatbot.py) that processes user input and generates responses. The system uses NLP techniques to analyze text input and extract relevant information such as symptoms. Recent studies show that AI chatbots can effectively assist in healthcare by providing timely feedback and improving patient interaction .

The **Triage Engine** (triage.py) is a critical component that evaluates patient messages and assigns severity levels. This engine uses rule-based logic or machine learning models to classify cases. Research indicates that AI-based triage systems can perform comparably to medical professionals in prioritizing cases .

The system uses Django ORM to store chat logs in the database. Each log contains the patient's message, chatbot response, severity level, and timestamp. This data is displayed on the doctor dashboard, where cases are sorted using custom ordering logic based on severity.

The chatbot interaction is implemented through an API endpoint (chat_api). This endpoint receives user input in JSON format, processes it using the triage engine, and returns a response along with severity classification. This ensures real-time communication between the frontend and backend.

Error handling mechanisms are included to manage invalid requests and system failures. Security features such as login authentication and CSRF protection are also implemented.

Overall, the implementation focuses on efficiency, modularity, and real-time performance, ensuring that the system can handle multiple users and provide reliable healthcare assistance.

VI. ALGORITHMS

1. Natural Language Processing (NLP)

NLP is used to process and understand patient input. It involves tokenization, text normalization, and feature extraction. NLP enables the chatbot to interpret symptoms and generate appropriate responses.

2. Rule-Based + Machine Learning Hybrid Model

The system may use a hybrid approach combining rule-based logic and machine learning. Rule-based systems handle predefined conditions, while machine learning models improve adaptability and accuracy.

3. Triage Classification Algorithm

The triage engine classifies patient cases into severity levels:

- EMERGENCY

- URGENT
- ROUTINE
- SELF-CARE

This classification is based on symptom keywords, severity indicators, and contextual analysis. AI-based triage systems have been shown to improve prioritization accuracy and efficiency in healthcare settings .

4. Decision Tree / Classification Models

Decision trees or similar classifiers can be used to map symptoms to severity levels. These models are simple, interpretable, and effective for healthcare applications.

5. Database Query Optimization

Django ORM is used to efficiently retrieve and sort chat logs based on severity and timestamp.

Recent research highlights that combining NLP with machine learning significantly enhances chatbot performance and healthcare decision support systems .

VII. SYSTEM DESIGN

The system design follows a modular and layered architecture to ensure scalability and efficient processing.

1. User Interface Layer

This layer includes web pages for:

- Patient registration and login
- Doctor dashboard
- Chat interface

It provides a user-friendly environment for interaction.

2. Application Layer

This layer contains core functionalities:

- Authentication module
- Chatbot module

- Triage engine
- API handler

It processes user input and manages system logic.

3. Chatbot Module

This module uses NLP techniques to analyze patient messages and generate responses. It acts as the primary interface between the user and the system.

4. Triage Engine

The triage engine evaluates patient input and assigns severity levels. It ensures that critical cases are prioritized.

5. Database Layer

The database stores:

- User information
- Chat logs
- Severity levels
- Timestamps

Django ORM ensures efficient data handling and retrieval.

6. Doctor Dashboard

The dashboard displays patient queries sorted by severity. This helps doctors focus on urgent cases first.

7. API Layer

The chat API enables real-time communication between frontend and backend. It processes JSON requests and returns responses.

8. Workflow

1. Patient logs in
2. Enters symptoms
3. Chatbot processes input
4. Triage engine assigns severity
5. Response is generated
6. Data stored in database

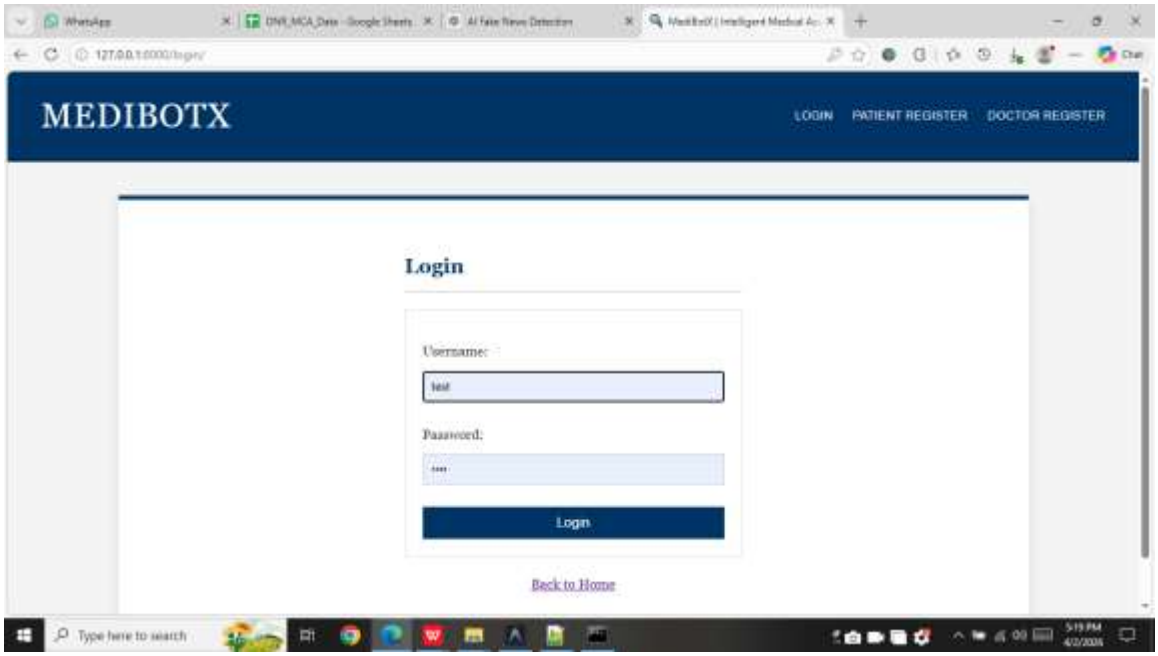
7. Doctor views prioritized cases

Design Features

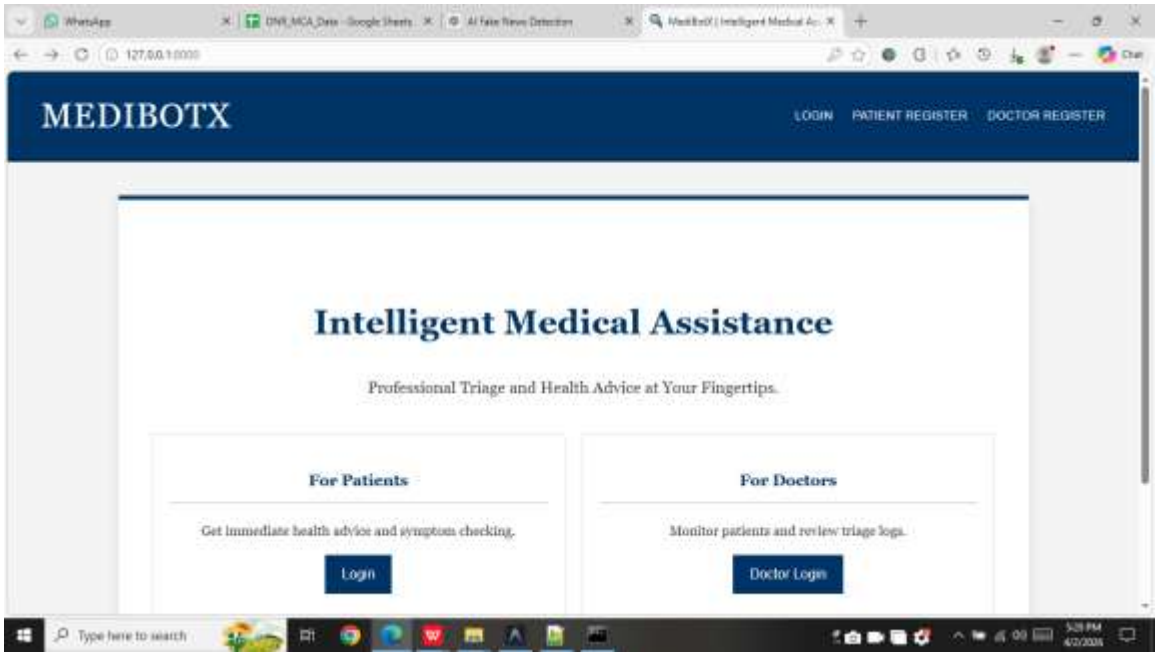
- Scalability for multiple users
- Modular architecture
- Real-time processing
- Secure authentication

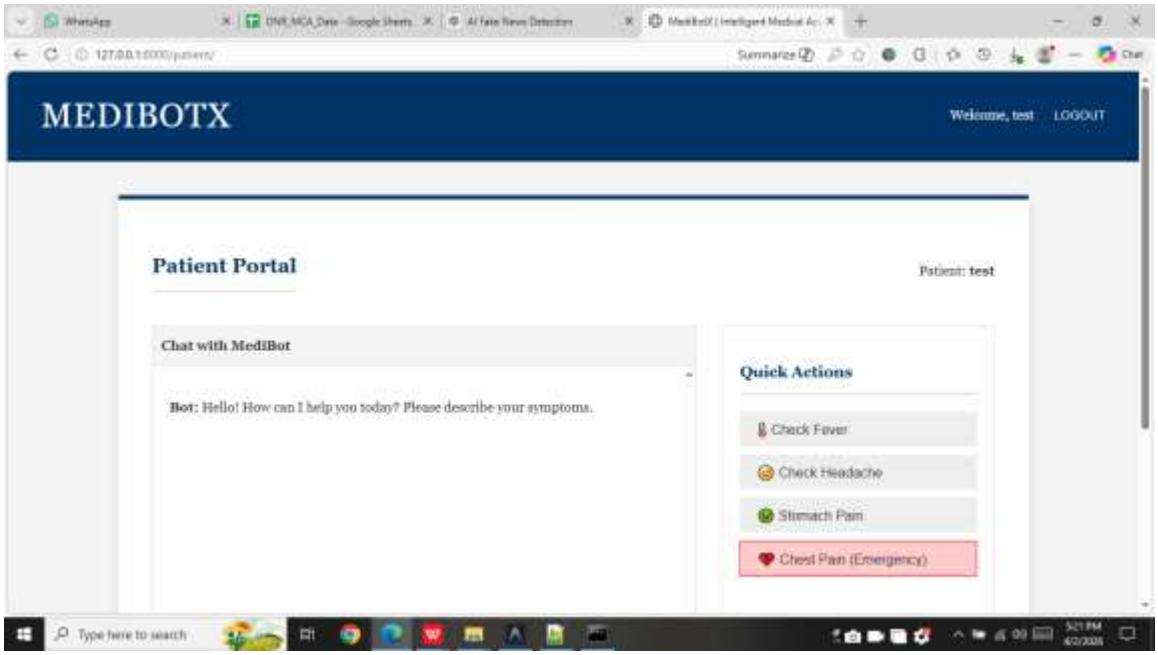
Modern AI healthcare systems emphasize modular design and real-time processing for efficient deployment

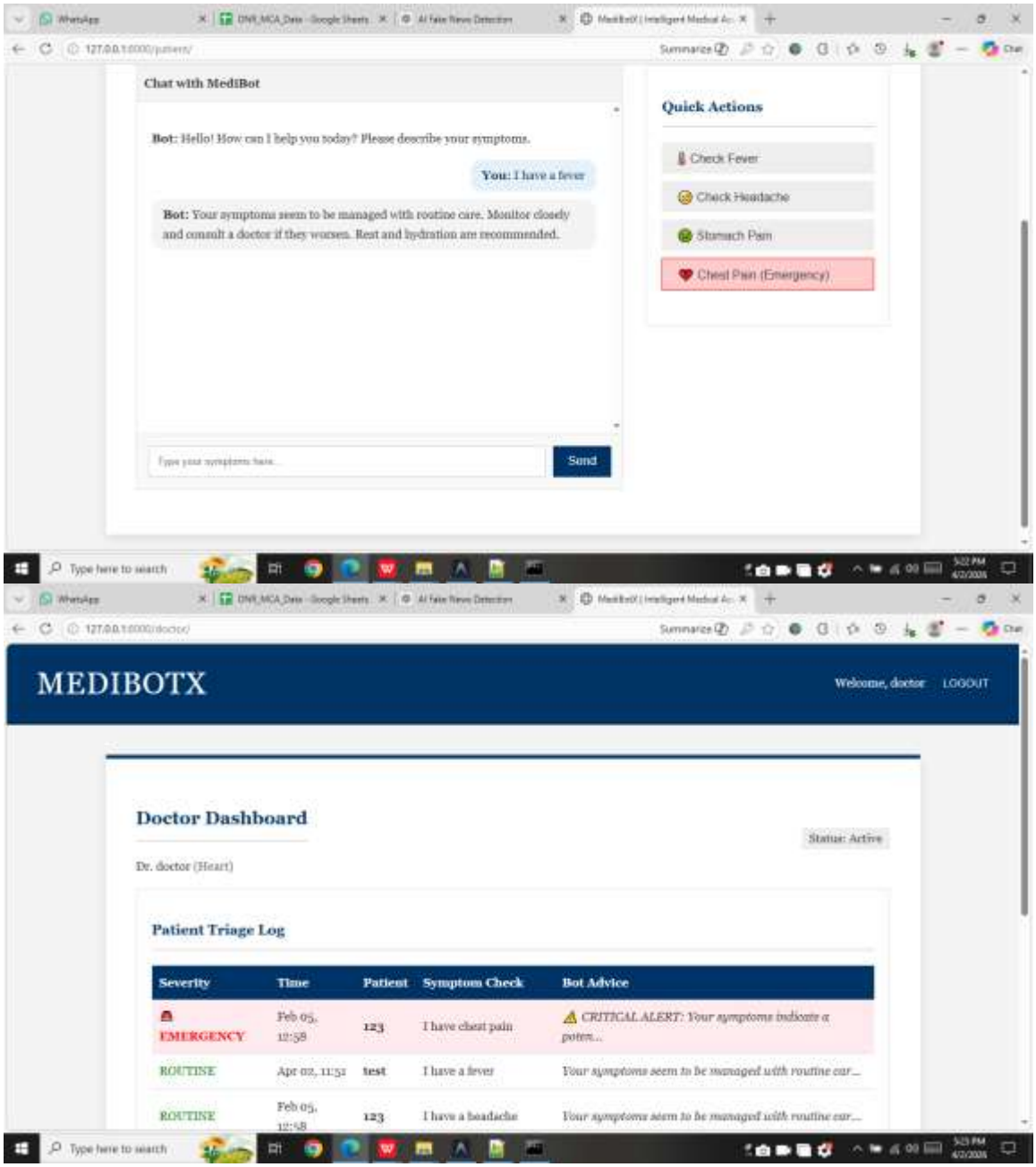
SYSTEM DESIGN IMAGES



International Journal of Engineering Research and Science & Technology







VIII. CONCLUSION

The AI-Based Healthcare Chatbot system developed in this project provides an efficient and scalable solution for improving healthcare accessibility and patient management. By integrating Natural Language Processing and machine learning techniques, the system enables automated interaction between patients and healthcare providers.One of the key contributions of this system is the implementation of an intelligent triage engine, which classifies patient cases based on severity. This ensures that critical cases receive immediate attention, improving overall healthcare efficiency. Studies have shown that

AI-based triage systems can enhance decision-making and resource allocation in healthcare environments .

The use of Django as the development framework ensures a robust and scalable architecture. The system supports multiple user roles, secure authentication, and real-time communication through APIs. The doctor dashboard further enhances usability by providing prioritized patient data.However, the system has certain limitations, including dependency on predefined rules and potential inaccuracies in complex medical cases. Future improvements may include integrating advanced deep learning models, expanding medical datasets, and incorporating real-time clinical data.

In conclusion, this project demonstrates the potential of AI in transforming healthcare services by providing automated, accessible, and efficient solutions. It serves as a foundation for future advancements in AI-driven healthcare systems.

REFERENCES

1. Jain, M. et al. (2024). *AI Healthcare Chatbot*
2. Dadheech, V. et al. (2025). *Advanced AI Healthcare Chatbot using NLP*
3. Idrees, S. et al. (2024). *Chatbots in Healthcare – A Study*
4. Chow, J. et al. (2025). *Large Language Models in Medical Chatbots*
5. Gomez, C. et al. (2024). *LLMs in Medical Triage Systems*
6. AI Chatbot Triage Study (2024)
7. Emergency AI Triage Evaluation (2025)
8. Shi, Q. et al. (2025). *C-PATH Healthcare AI System*
9. Bhatt, A. et al. (2024). *Med-Bot AI Assistant*
10. Ke, Y. et al. (2024). *PEACH AI Healthcare Chatbot*
11. Bhatt, D. et al. (2024). *AI Diagnostic Benchmarking*
12. Nature Medicine AI Chatbot Study (2024)
13. IEEE Healthcare AI Systems Review (2023)
14. WHO Digital Health Report (2024)
15. Springer AI in Healthcare Systems (2024)