

ARTIFICIAL INTELLIGENCE-BASED CHATBOT FOR PATIENT HEALTH CARE

Submitted in partial fulfillment
of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
Specialization in Data Science

By

BALAGUNDLA SUDHEERKUMAR (40733004)
KOPPURAVURI VENKAT (40733703)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BALAGUNDLA SUDHEERKUMAR (40733004) & KOPPURAVURI VENKAT (40733703)** who carried out complete project entitled "**ARTIFICIAL INTELLIGENCE-BASED CHATBOT FOR PATIENT HEALTH CARE**" under my supervision from November 2023 to April 2024.

G. Kavitha
10/4/24

Internal Guide

Ms.G.Kavitha, M.E.,(Ph.D),,

[Signature]
HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Dr. **SATHYABAMA** D.,
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva-voce Examination held on

12/4/24

[Signature]
Internal Examiner

[Signature]
External Examiner

(Dr. P. ASHA)

DECLARATION

I, **BALAGUNDLA SUDHEER KUMAR(40733004)**, here by declare that the Project Report entitled "**ARTIFICIAL INTELLIGENCE-BASED CHATBOT FOR PATIENT HEALTH CARE**" done by me under the guidance of **Ms. G. Kavitha, M.E.,(Ph.D).**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering with Specialization in Data Science**.

DATE: 10/04/2024

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, Dean, School of Computing, **Dr. L. LAKSHMANAN, M.E., Ph.D.**, and **Dr. S. Vigneshwari M.E., PH.D.**, Heads of the **Department of Computer Science and Engineering** for providing me with the necessary support, and details at the right time during the progressive reviews.

I would like to express my sincere and deep gratitude to my Project Guide **Ms. G. Kavitha, M.E.,(Ph.D)**, for his valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Medical care is very important for a healthy life. However, it is very difficult to seek medical attention if you have a health problem. The recommended notion is to develop a medical Chabot that can adopt ML to analyze the ailment and produce necessary information concerning the condition we are discussing with a doctor. Medical chatbots were built to reduce medical costs and improve access to medical knowledge. Some chatbots serve as medical manuals to help patients become aware of their illnesses and improve their health. Users can assuredly benefit from chatbots if they can diagnose several kinds of illnesses and render the required data. Text Diagnosis Bot enables sufferers to join in analyses of their medicinal matters and present a personalized analysis report concerning the symptoms. Therefore, people have opinions about their well-being and individual stability. Designing and implementing an artificial intelligence (AI)-based chatbot for healthcare involves a systematic approach encompassing various stages and components. This abstract provides an overview of the key aspects involved in creating such a chatbot, including requirements gathering, technology stack selection, data collection and preprocessing, AI model training, integration with healthcare knowledge bases, dialog management, user interface design, security and privacy considerations, testing and validation, deployment and monitoring, user training and support. The chatbot aims to enhance healthcare accessibility, provide medical advice, schedule appointments, offer medication reminders, and engage users in meaningful conversations related to their health and well-being. By following a structured implementation process, healthcare organizations can leverage AI-driven chatbots to improve patient outcomes, streamline healthcare services, and enhance overall healthcare delivery.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	3
	LIST OF ABBREVIATIONS	4
1	INTRODUCTION	5
2	LITERATURE SURVEY	
	2.1 Inferences from the Literature Survey	6
	2.2 Existing System and Proposed System	9
	2.3 Open Problems in Existing System	9
3	REQUIREMENTS ANALYSIS	
	3.1 Risk Analysis of the Project	10
	3.2 Software Requirements & Hardware Requirements	12
4	DESCRIPTION OF THE PROPOSED SYSTEM	
	4.1 Flow chart	13
	4.2 Selected Methodology or process model	16
	4.3 Architecture / Overall Design of Proposed System	23
	4.4 Description of Software for Implementation and Testing Plan of the Proposed Model/System	24
5	IMPLEMENTATION DETAILS	27
6	RESULTS AND DISCUSSION	29
	A. Working Principle	29
	B. Results	31
7	CONCLUSION	34
	REFERENCES	35

APPENDIX

A	SOURCE CODE	35
B	SCREENSHOTS	39
C	RESEARCH PAPER	41

LIST OF FIGURES

FIGURE NUMBER	FIGURE NAME	PAGE NUMBER
4.1	FLOWCHART	14
4.2	USE CASE DIAGRAM	15
4.3	SEQUENCE DIAGRAM	16
4.4	ARCHITECTURE DIAGRAM	24
5.1	HOME PAGE	30
5.2	BASIC DETAILS PAGE	31
5.3	MEDICAL CHATBOT PAGE	31

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AI	ARTIFICIAL INTELLIGENCE
DFD	DATA FLOW DIAGRAM
GUI	GRAPHICAL USER INTERFACE
ML	MACHINE LEARNING
NLP	NATURAL LANGUAGE PROCESSING
NLTK	NATURAL LANGUAGE TOOLKIT
SDN	SOFTWARE-DEFINED NETWORK
UML	UNIFIED MODELLING LANGUAGE

CHAPTER 1

INTRODUCTION

Medical chatbot has a high impact on the health culture of the state. It has improved reliability and is less prone to human errors. Today's people are more likely addicted to the internet but they are not concerned about their health. They avoid hospital treatment for small issues which may become a major disease in the future.

This proposed idea solves this problem. This idea focuses on creating a chatbot that is free of cost and available throughout the day. The fact that the chatbot is free and can be accessed wherever the user is, be it their working environment, prompts the user to have it and use it. It saves the overhead involved in consulting specialized doctors.

Machine Learning is now at its peak and chatbot is just one example of its use to contribute to progress. It should also be made clear that chatbots are not designed to diagnose patients. The idea presented was to use Machine Learning to create a medical chatbot that could provide some basic details about the disease before consulting a doctor. Text-to-text medical chatbots enable patients to discuss their medical problems and provide personalized diagnoses based on symptoms. The patient can then be transferred to a specialist.

This sequence of things fundamentally saves the doctor's time. In turn, chatbots are always available to help people in need of medical assistance. In addition, the virtual assistant may be responsible for reminding users to take medication and monitoring the patient's health status. Generally, chatbots communicate with real people.

Chatbots are used in applications such as e-commerce customer service, call centers, and internet games. However, a chatbot can communicate with a text or voice interface and get responses through Machine Learning.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

A literature survey is the most important step in the software development process. Before developing the tool it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need a lot of external support. This support can be obtained from senior programmers, books, or websites. Before building the system the above considerations are taken into account for developing the proposed system. The major part of the project development sector considers and fully surveys all the required needs for developing the project. For every project Literature survey is the most important sector in the software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, manpower, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine the software specifications in the respective system such as what type of operating system the project would require, and what necessary software is needed to proceed with the next step such as developing the tools, and the associated operations.

1. A Chabot for Medical Purpose using Deep Learning Vaibhav Tode, Himanshu Gadge, Sudarshan Madane, Prateek Kachare, Anuradha Deokar (2021).

Our Project aims to design a Conversational AI Powered Chatbot for Medical Diagnostics using Deep Learning which mainly focuses on rural parts as well as poor and needy people of our country. Our System can understand the symptoms of the patient and communicate with the Patient (End-user) through web UI. Our system tries to solve their problem with the help of the symptoms provided by the Patient itself and helps them to give the correct antibiotics/ medicines and precautions. NLTK (Natural Language Toolkit) is a module/ program in Python that can able to perform symbolic and statistical Natural Language Processing for English written in Programming.

2. Contextual Chatbot for Healthcare Purposes (using Deep Learning), Prathamesh Kandpal, Kapil Jasnani, Ritesh Raut, Prof. Dr. Siddharth Bhorge (2022).

In this paper we will be not only discussing the working of our model but also the applications and relevant work conducted in this domain, also there will be a discussion about the challenges and future scope of this technology. Healthcare plays a wide role in our daily lives, whenever a person feels sick he/she visits their family doctor or any nearby clinic just to get to know what issues they are facing, in recent years many companies & institutions have collaborated with hospitals to provide support which can help doctors and medical staff to deal with patients in a better way and reduce their efforts with the help of technology.

3. Healthcare Chatbot using Natural Language Processing, Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge (2023).

The system provides text or voice assistance, which means the user can use his convenient language, The Bot will provide which type of disease is based on the user's symptoms and provides a doctor and also provides food suggestions which means which type of food you have to take. Thus, people will have an idea about their health and have the right protection. Chatbots are programs that work on Machine Learning (ML) as well as Artificial Intelligence (AI) Natural Language Processing (NLP) techniques such as NLTK for Python can be applied to analyze speech, and intelligent responses can be found by designing an engine to provide appropriate human-like responses.

4. Santhanamahalingam, S., Alagarsamy, S., & Subramanian, K. (2022, October). A study of cloud-based VPN establishment using network function virtualization technique. In 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 627-631). IEEE.

Santhanamahalingam, Alagarsamy, and Subramanian conducted a study focusing on the establishment of a cloud-based VPN using the network function virtualization (NFV) technique. Their research aimed to establish a resilient and protected VPN infrastructure, ensuring secure connectivity and safeguarding network communications. The study was presented at the 2022 3rd International Conference on Smart Electronics and Communication, organized by the Institute of Electrical and Electronics Engineers (IEEE), with the paper being published in the conference proceedings (pp. 627-631).

5. Mozo, A., Karamchandani, A., de la Cal, L., Gómez-Canaval, S., Pastor, A., & Gifre, L. (2023). A Machine-Learning-Based Cyberattack Detector for a Cloud-Based SDN Controller. *Applied Sciences*, 13(8), 4914.

Mozo et al. (2023) present a machine-learning-based cyberattack detector for a cloud-based software-defined networking (SDN) controller. Their study focuses on establishing a resilient and protected virtual private network (VPN) infrastructure to safeguard network communications and ensure secure connectivity. By utilizing machine learning techniques, the proposed detector aims to enhance the security of the SDN controller against cyber threats, contributing to the establishment of a secure VPN infrastructure. The findings of this research hold significance in improving network security and protecting sensitive information transmitted through VPNs.

6. Nandhini, S., Vaishnav, S., & Vishwakarma, S. V. (2022, April). VPN blocker and recognizing the pattern of IP address. In *AIP Conference Proceedings* (Vol. 2405, No. 1). AIP Publishing.

Nandhini, Vaishnav, and Vishwakarma (2022) present their research in AIP Conference Proceedings, where they discuss the development of a VPN blocker and the identification of IP address patterns. Their work aims to establish a resilient and protected VPN infrastructure that safeguards network communications and ensures secure connectivity. By implementing mechanisms to block unauthorized VPN access and recognizing patterns in IP addresses, the authors contribute to enhancing the security and reliability of virtual private networks.

2.2 EXISTING SYSTEM AND PROPOSED SYSTEM

- **Existing System:**

A conversational agent that interacts with users using natural language is called a Chatbot. Many chatbots have been developed using text communication starting from ELIZA which simulates a psychotherapist to PARRY which simulates a paranoid patient. ELIZA is a well well-known artificial therapist. The bot attempts to rephrase the questions of the client and responds to certain keywords. If no keyword is found ELIZA replies with fixed phrases to keep the conversation going. Medicine is a field in which help is critically needed.

- **Proposed System:**

The proposed idea is to make a medical chatbot using AI which will diagnose the disease and supply basic details about the disease before consulting a doctor. to scaleback the healthcare costs and improve accessibility to medical knowledge the medical chatbot is made. A few chatbots act as reference books, which help the patient discover the illness and improve their well-being. The user can take important advantage of a chatbot only it can diagnose all quiet diseases and supply the necessary information. text-to-text diagnosis bot connects patients about their medical issues and gives a customized diagnosis to support their symptoms. Hence, people will have thought about their health and have the proper protection.

2.3 OPEN PROBLEMS IN THE EXISTING SYSTEM

- A conversational agent that interacts with users using natural language is called a Chatbot.
- Many chatbots have been developed using text communication starting from ELIZA which simulates a psychotherapist to PARRY which simulates a paranoid patient.
- ELIZA is a well-known artificial therapist.
- The bot attempts to rephrase the questions of the client and responds to certain keywords.
- If no keyword is found ELIZA replies with fixed phrases to keep the conversation going. Medicine is a field in which help is critically needed.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating, and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable, and traceable related to identified business needs or opportunities, and defined to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance, and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

USABILITY

It specifies how easily the system must be used. It is easy to ask queries in any format which is short or long, the Porter stemming algorithm stimulates the desired response for the user.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

ROBUSTNESS

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. The user can cope with errors for irrelevant queries during execution.

SECURITY

The state of providing protected access to resources is security. The system provides good security and unauthorized users cannot access the system thereby providing high security.

RELIABILITY

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even be capable of working around any failure.

COMPATIBILITY

It is supported by version above all web browsers. Using any web server like localhost makes the system a real-time experience.

FLEXIBILITY

The flexibility of the project is provided in such a way that it can run on different environments being executed by different users.

SAFETY

Safety is a measure taken to prevent trouble. Every query is processed securely without letting others know one's personal information.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS DOCUMENT

HARDWARE REQUIREMENTS

- System: Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor: 15" LED
- Input Devices: Keyboard, Mouse
- Ram: 4 GB.

SOFTWARE REQUIREMENTS

- Operating system: Windows 7/10.
- Coding Language: Python.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The proposed idea is to make a medical chatbot using AI which will diagnose the disease and supply basic details about the disease before consulting a doctor. to scale back the healthcare costs and improve accessibility to medical knowledge the medical chatbot is made. A few chatbots act as reference books, which help the patient discover the illness and improve their well-being. The user can take important advantage of a chatbot only it can diagnose all quiet diseases and supply the necessary information. text-to-text diagnosis bot connects patients about their medical issues and gives a customized diagnosis to support their symptoms. Hence, people will have thought about their health and have the proper protection.

4.1 FLOWCHART

- The DFD is also called a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as a bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

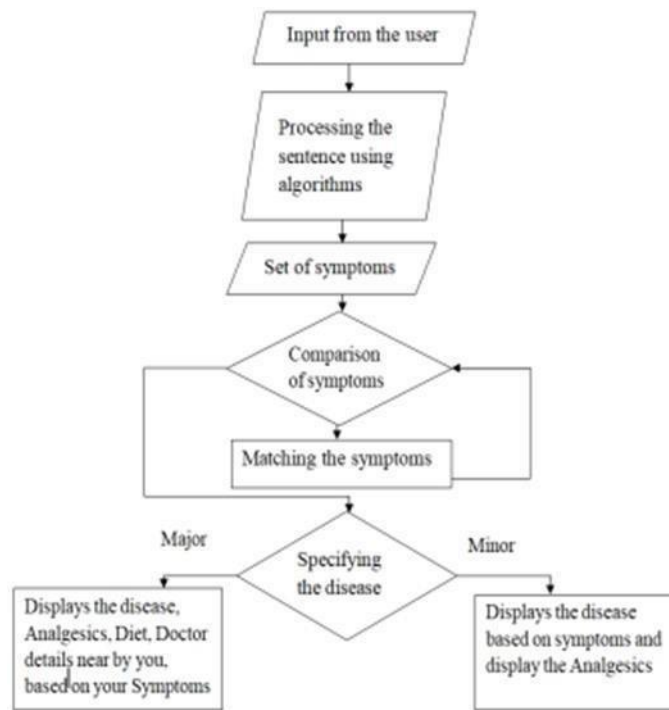


FIG 4.1 FLOW CHART

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form, UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

USE CASE DIAGRAM

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

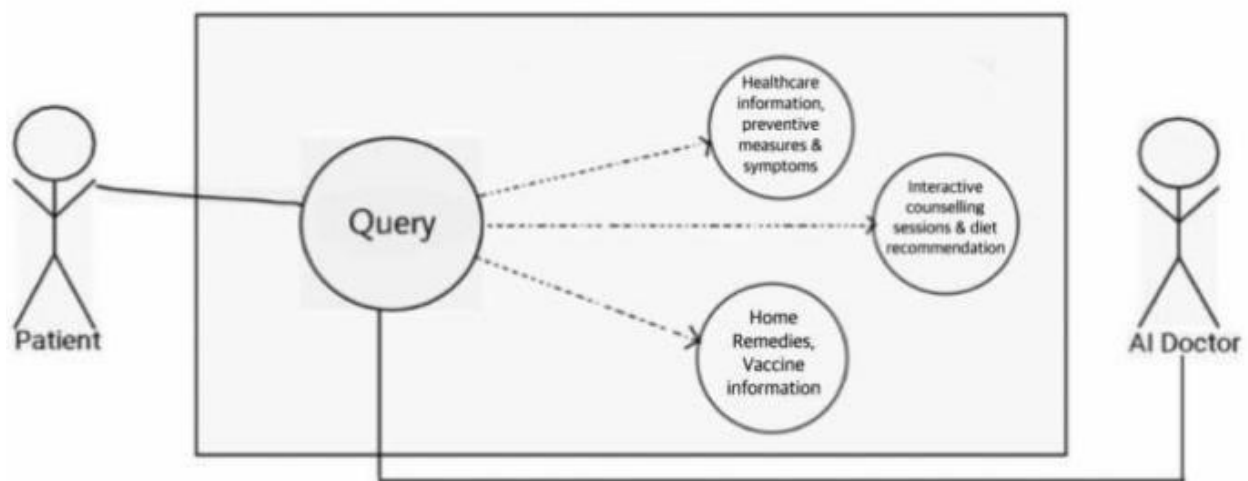


FIG 4.2 USE CASE DIAGRAM

SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extensibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development processes.

Provide a formal basis for understanding the modeling language.

Encourage the growth of the OO tools market.

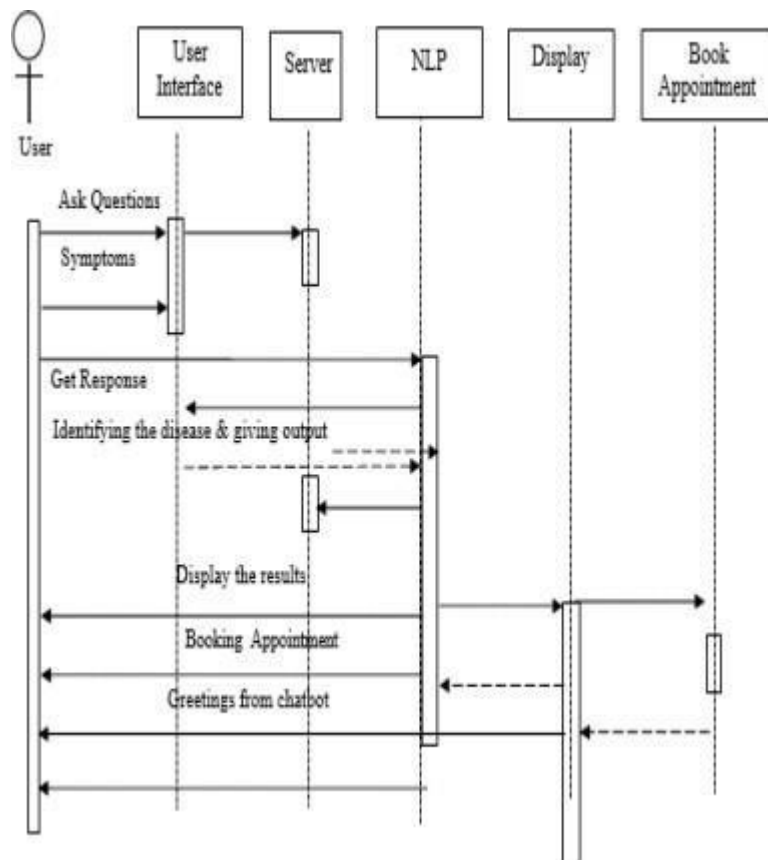


FIG 4.3 SEQUENCE DIAGRAM

4.2 SELECTED METHODOLOGY OR PROCESS MODEL

- User Validation And Extraction Of Symptoms
- Extracted Symptoms With Trained Datasets
- Specifying The Disease And Referring A Doctor

1. User Validation and Extraction of Symptoms

The validation of the user login details occurs here. Then Symptoms are extracted using the String Searching Algorithm where a substring representing the symptoms is identified in the natural language text input. When users give directly the symptom name the system will identify it. However, the system should also be able to handle input like, “When I read, I’m okayat first, but over time, my eyes seem to get tired, and I start to see double.” In this case, the system should extract words like “eyes tired” and “see double” (and not substrings like “read” or “okay”).

2. Extracted Symptoms with Trained Datasets

Given some substring from the user's input, we generate a list of suggested closest symptoms. We then ask the user to confirm if they have any of the symptoms from the suggested ones. Based on their reply few diseases are being shortlisted. Then further symptom clarification and symptom suggestions are done by asking the users a series of questions and the mapping of the symptoms to the exact disease is done.

3. Specifying the Disease and Referring a Doctor

This process carries the list of diseases in the database and each symptom being entered is compared to the symptoms of the common diseases. The next symptom is checked until a matching symptom is found. They are shortlisted based on the end users' input on the question evaluation. The accurate disease is identified and specified to the end user by the Chatbot. The chatbot checks whether the identified disease is a major issue or a minor issue based on the conditions built into the chatbot. If it is a major issue the chatbot refers a specialist to the end user by sending the doctor details. And if it is a minor issue the chatbot specifies the disease and alerts the end user with first aid or remedy and asks to visit a doctor shortly.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- Web development (server-side),
- Software development,
- Mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping or production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
 - Python has a simple syntax similar to the English language.
 - Python has a syntax that allows developers to write programs with fewer lines than some other programming languages.
 - Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
 - Python can be treated procedurally, in an object-orientated way, or in a functional way.
 - The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
 - In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, NetBeans, or Eclipse which are particularly useful when managing larger collections of Python files.
- ### Python Syntax compared to other programming languages
- Python was designed for readability and has some similarities to the English language with influence from mathematics.
 - Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
 - Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions, and classes. Other programming languages often use curly brackets for this purpose.

Python Install

Many PCs and Macs will have Python already installed.

To check if you have Python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have Python installed on a Linux or Mac, then on Linux open the command line, or on Mac open the Terminal and type:

```
Python --version
```

If you find that you do not have Python installed on your computer, then you can download

it for free from the following website: <https://www.python.org/>

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the Python interpreter to be executed.

The way to run a Python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your Python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
```

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in Python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any Python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19, 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19, 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Whenever you are done in the Python command line, you can simply type the following to quit the Python command line interface:

```
exit ()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Or by creating a Python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in

- Python the indentation is very important.
- Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
```

```
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
```

```
print("Five is greater than two!")
```

Comments

Python has commenting capability for in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
```

```
print("Hello, World!")
```

Docstrings

Python also has an extended documentation capability, called docstrings.

Docstrings can be one-line or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
""" This is a  
multiline docstring."""
```

```
print("Hello, World!")
```

SYSTEM TESTING

Test plan

Software testing is the process of evaluating a software item to detect differences between a given input and expected output. Also to assess the features of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process of making sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per

customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of Testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing the expected output against the given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white-box testing and black-box testing.

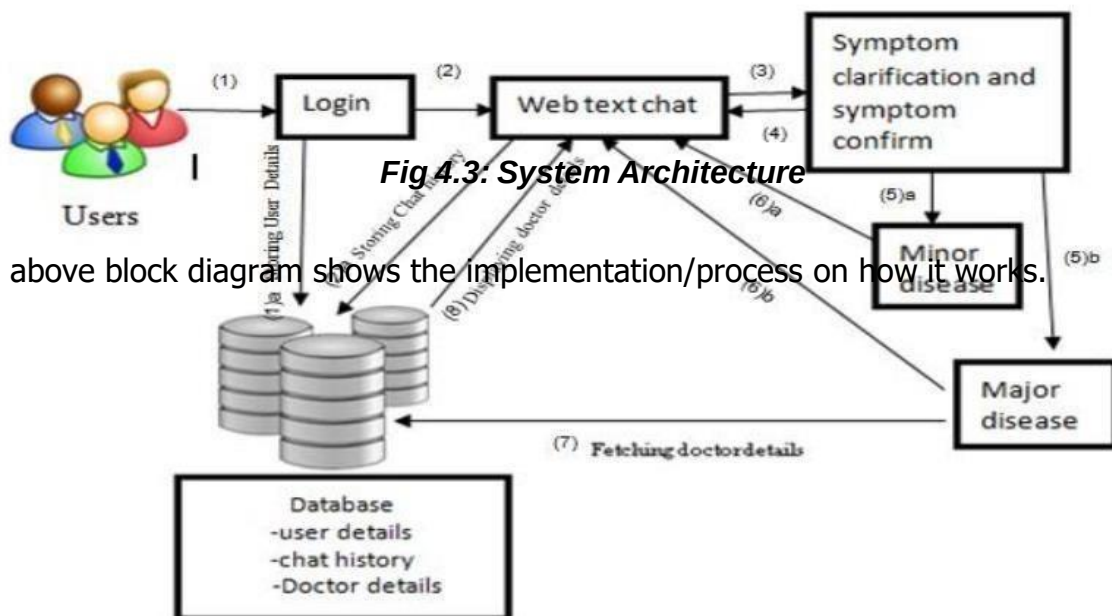
Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation.

4.3 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM



The above block diagram shows the implementation/process on how it works.

FIG 4.4 ARCHITECTURE DIAGRAM

4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING LAN OF THE PROPOSED MODEL/SYSTEM

To implement this model, the execution of the program is done through Google Colab. Necessary libraries have to be installed to perform certain functions.

DESCRIPTION OF PROGRAMMING LANGUAGE AND SOFTWARE

PYTHON

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports an Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, a simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries, and Windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on a Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.

Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.

- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is easy to use. Just accept the default settings, wait until the installation is finished, and you are done.

Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

CHAPTER5

IMPLEMENTATION DETAILS

The implementation of an artificial intelligence (AI)-based chatbot for healthcare involves several key steps and components. Below are the detailed implementation details for creating such a chatbot:

1. Requirements Gathering: - Define the objectives and scope of the chatbot, such as providing medical advice, scheduling appointments, medication reminders, etc. - Identify the target audience (patients, caregivers, healthcare professionals) and their specific needs and preferences. - Gather input from healthcare experts, clinicians, and potential users to understand the desired functionalities and features of the chatbot.

2. Technology Stack: - Choose a suitable programming language and framework for developing the chatbot. Common choices include Python (using libraries like Flask or Django), Node.js, or Java. - Select AI and NLP libraries and tools for natural language understanding and processing, such as TensorFlow, PyTorch, NLTK, spaCy, or Dialogflow (for Google Cloud-based solutions). - Consider additional tools for voice recognition and synthesis if voice-based interactions are part of the chatbot's functionality.

3.Data Collection and Preprocessing: - Collect a diverse dataset of medical questions, symptoms, diagnoses, treatments, medications, and other healthcare-related information. - Preprocess the data by cleaning, normalizing, and structuring it into a format suitable for training AI models. This may involve tokenization, stemming, lemmatization, and entity recognition.

4. AI Model Training: - Develop and train AI models for natural language understanding (NLU) and natural language generation (NLG) tasks. - Use machine learning algorithms such as deep learning (e.g., neural networks), decision trees, or ensemble methods to train NLU models to interpret user queries accurately. - Train NLG models to generate coherent and contextually relevant responses based on the input received from users.

5.Integration with Healthcare Knowledge Base: - Integrate the chatbot with a healthcare knowledge base or database containing medical information, guidelines, protocols, and treatment options. - Ensure the chatbot can access and retrieve relevant information from the knowledge base to provide accurate and up-to-date responses to user queries.

6.Dialog Management: - Implement a dialog management system to handle multi-turn conversations, context switching, and user intents effectively. - Use techniques such as state machines, dialogue trees, or reinforcement learning for managing conversation flows and maintaining context across interactions.

7. User Interface Design: - Design a user-friendly interface for the chatbot, suitable for web,

mobile, or messaging platforms (e.g., Slack, Facebook Messenger, WhatsApp). - Incorporate features like buttons, menus, quick replies, and rich media (images, videos) to enhance user interaction and engagement.

8. Security and Privacy: - Implement security measures to protect sensitive user data, comply with healthcare regulations (e.g., HIPAA), and ensure confidentiality and privacy. - Use encryption, authentication mechanisms, access controls, and data anonymization techniques to safeguard user information and maintain trust.

9. Testing and Validation: - Conduct thorough testing of the chatbot across different scenarios, user inputs, and edge cases to ensure robustness, accuracy, and reliability. - Use automated testing tools, unit tests, integration tests, and user acceptance testing (UAT) to validate the chatbot's functionality and performance.

10. Deployment and Monitoring: - Deploy the chatbot on scalable and reliable hosting infrastructure, such as cloud platforms (e.g., AWS, Google Cloud, Azure) or on-premises servers. - Monitor the chatbot's performance, response times, user satisfaction metrics, and system logs to identify issues, optimize performance, and iteratively improve the chatbot over time.

11. User Training and Support: - Provide user training and onboarding resources to educate users about the chatbot's capabilities, functionalities, and best practices for interaction. - Offer ongoing support, troubleshooting assistance, and feedback mechanisms to gather user input, address user queries, and enhance the chatbot's usability. By following these implementation steps and considerations, you can develop and deploy an effective AI-based chatbot for healthcare that provides valuable support, information, and assistance to users in a user-friendly and secure manner.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 WORKING PRINCIPLE

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help", "copyright", "credits" or"license" for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running a new version of Python, then you would need to use a print statement with parenthesis as in **print ("Hello, Python!");**. However, in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins the execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have an extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have a Python interpreter set in the PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

HTTP protocol is the foundation of data communication on the World Wide Web. Different methods of data retrieval from specified URLs are defined in this protocol.

The following table summarizes different HTTP methods –

S.No	Methods & Description
1	GET Sends data in unencrypted form to the server. The most common method.
2	HEAD Same as GET, but without a response body
3	POST Used to send HTML form data to the server. Data received by the POST method is not cached by the server.

4	PUT Replace all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

To demonstrate the use of the **POST** method in URL routing, first, let us create an HTML form and use the **POST** method to send form data to a URL.

6.2 RESULTS

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

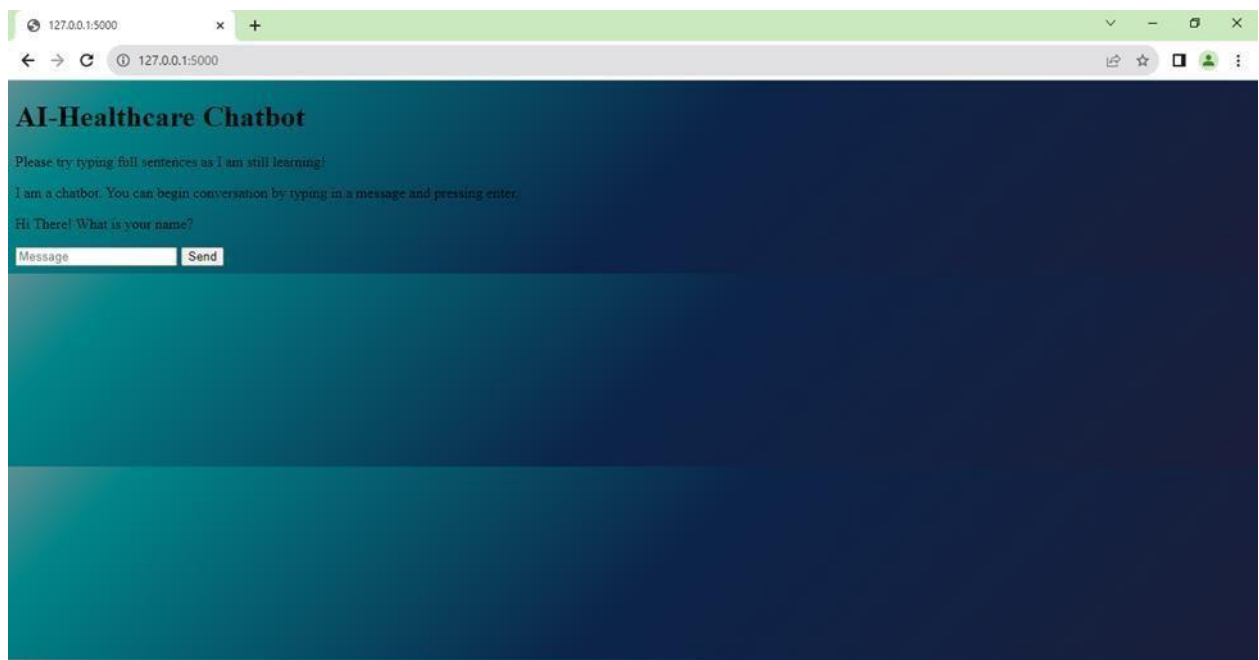


FIG 6.1 HOME PAGE

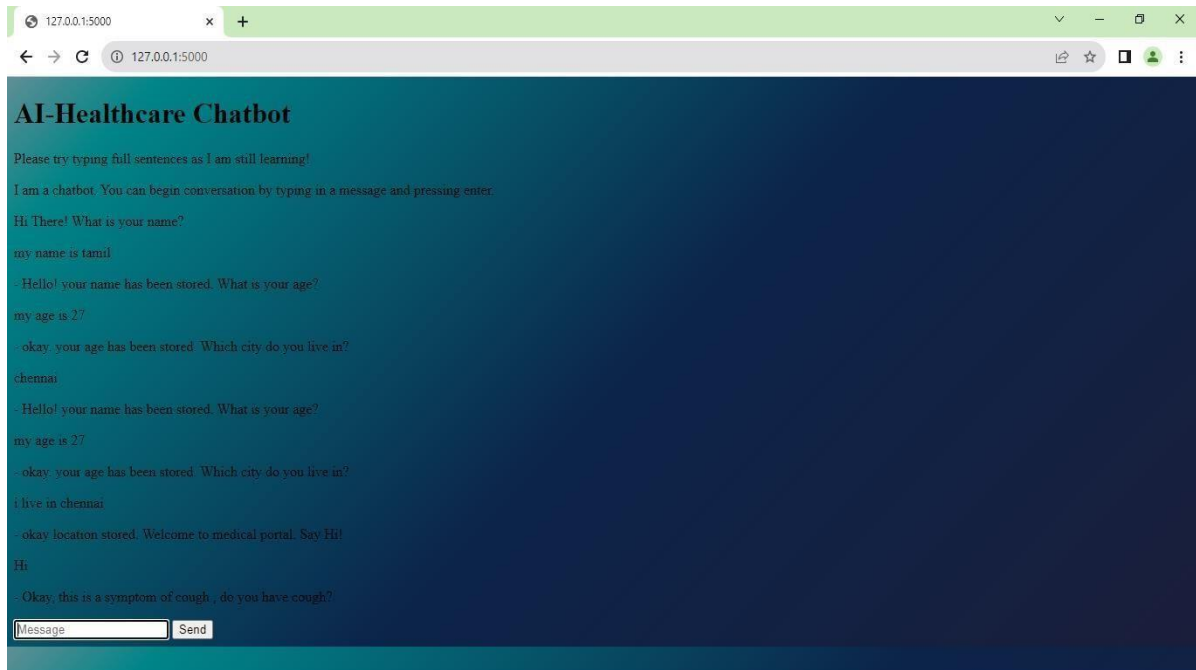


FIG 6.2 BASIC DETAILS PAGE

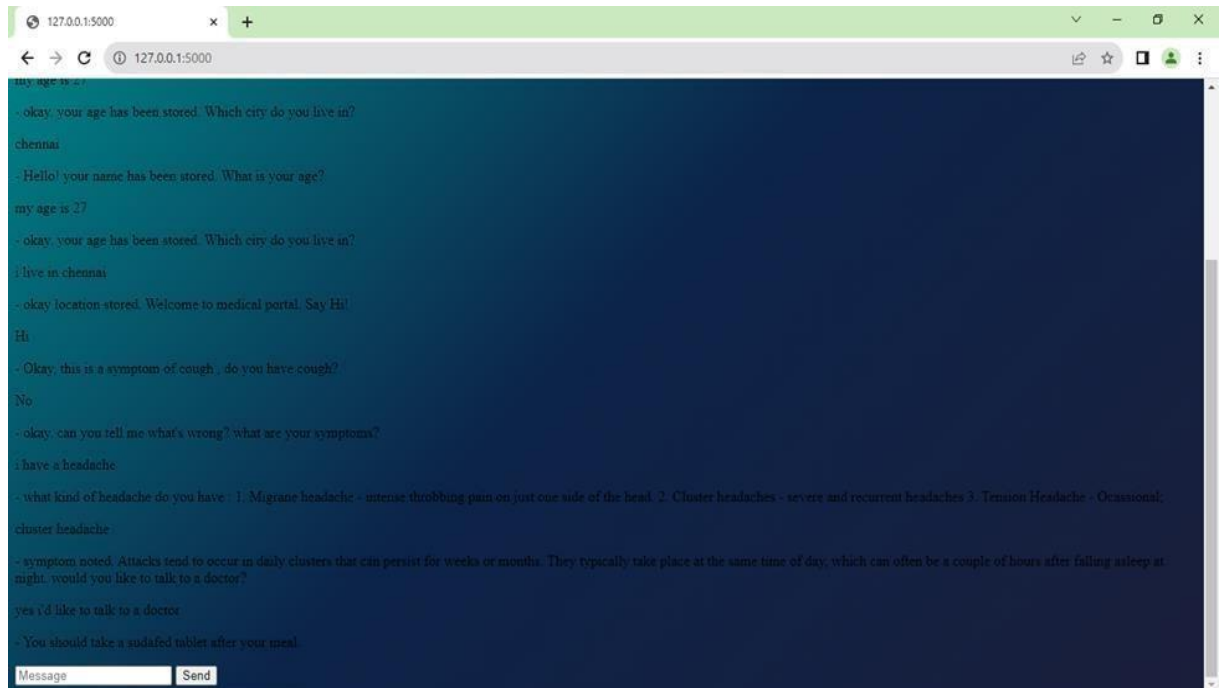


FIG 6.3 MEDICAL CHATBOT PAGE

CHAPTER 7

CONCLUSION

CONCLUSION

In conclusion, the development of an AI-based chatbot for healthcare holds great promise in improving patient engagement and accessibility to medical information. This technology offers round-the-clock support, efficiently addressing user queries and providing personalized health advice. Leveraging natural language processing, the chatbot ensures a user-friendly interaction experience, catering to individuals with varying levels of health literacy and empowering them to manage their well-being effectively. However, while the chatbot demonstrates significant potential, certain limitations and ethical considerations must be addressed. Its effectiveness hinges on the accuracy of underlying algorithms, and challenges remain in handling complex medical inquiries or emergencies. Additionally, ensuring user privacy and maintaining the confidentiality of health data are paramount in the deployment of such systems. Despite these challenges, the continuous advancement of technology holds the promise of further enhancing the capabilities of AI-based chatbots, ultimately revolutionizing healthcare interactions and improving overall patient care.

REFERENCES

1. [1] Vaibhav Tode, Himanshu Gadge, Prateek Kachare, and Sudarshan Madane, CureBot -An Artificially Intelligent Interactive Bot for Medical Diagnostics International Research Journal of Engineering and Technology (IRJET)., Vol.7, no.12 (Dec 2020).
2. [2] Satyendra Praneel Reddy Karri and Dr Santosh Kumar, Deep Learning Techniques for Implementation of Chatbots, 2020 International Conference on Computer Communication and Informatics (ICI), Coimbatore, INDIA, (2020) January 22-24.
3. [3] Dipesh Kadariya, Revathy Venkataramanan, Hong Yung Yip, Maninder Kalra, Krishnaprasad Thirunarayanan and Amit Sheth, kBot: Knowledge-Enabled personalized Chatbot for Asthma Self-Management, 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA,(2019) June 12-15.
4. [4] R. Dharwadkar, N.A. Deshpande, A medical chatbot. Int. J. Comput. Trends Technol. (IJCTT). **60**(1) (2018).
5. [5] S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari, S.K. Devi, A self-diagnosis medical chatbot using artificial intelligence. J. Web Dev. Web Designing **3**(1) (2018).
6. [6] N. Jyothirmayi, A. Soniya, Y. Grace, C. Reddy Kumar Kishor, B.V. Murthy Ramana, Survey on chatbot conversational system. J. Appl. Sci. Computer. **6**(1) (2019).
7. [7] Gorgulla, C., Boeszoermenyi, A., Wang, Z. F., Fischer, P. D., Coote, P. W., Padmanabha Das, K. M., ... & Arthanari, H. (2020). An open-source drug discovery platform enables ultra-large virtual screens. Nature, 580(7805), 663-668.
8. [8] Coley, C. W., Eyke, N. S., & Jensen, K. F. (2020). Autonomous discovery in the chemical sciences part II: outlook. Angewandte Chemie International Edition, 59(52), 23414-23436.
9. [9] Laranjo L, Dunn AG, Tong HL, Kocaballi AB, Chen J, Bashir R, Surian D, Gallego B, Magrabi F, LauAYS, Coiera E. Conversational agents in healthcare: a systematic

review. J Am Med Inform Assoc. 2018 Sep 1;25(9):1248-1258. doi: 10.1093/jamia/ocy072. PMID: 30010941; PMCID: PMC6118869.

- 10.[10]S. Ayanouz, B. Abdelhakim, and M. Benhmed, "Asmart chatbot architecture based NLP and machine learning for health care assistance", Association for Computing Machinery Proceedings, pp. 1-6, 2020.
- 11.[11]Seo Kyori, Analysis of Domestic and GlobalTrends and Development Prospects of Artificial Intelligence-based Chatbot Service, National Information Society Agency (NIA), D.gov Trend & Future, 2018.

APPENDIX

A. SOURCE CODE

```
from flask import Flask, render_template, request
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer
import os

from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

filenumber=int(os.listdir('saved_conversations')[-1])
filenumber=filenumber+1
file= open('saved_conversations/'+str(filenumber),'w+')
file.write('bot : Hi There! I am a medical chatbot. You can begin conversation by typing in
a message and pressing enter.\n')
file.close()

app = Flask(__name__)

english_bot = ChatBot('Bot',
    storage_adapter='chatterbot.storage.SQLStorageAdapter',
    logic_adapters=[
        {
            'import_path': 'chatterbot.logic.BestMatch'
        },
    ],
    trainer='chatterbot.trainers.ListTrainer')
english_bot.set_trainer(ListTrainer)
```

```

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/get")
def get_bot_response():
    userText = request.args.get('msg')
    response = str(english_bot.get_response(userText))

    appendfile=os.listdir('saved_conversations')[-1]
    appendfile= open('saved_conversations/'+str(filenameumber),"a")
    appendfile.write('user : '+userText+'\n')
    appendfile.write('bot : '+response+'\n')
    appendfile.close()

    return response

if __name__ == "__main__":
    app.debug = True
app.run()

from chatterbot import ChatBot, utils
from chatterbot.trainers import ListTrainer
import os

try:
    os.remove("db.sqlite3")
    print("Old database removed. Training new database")
except FileNotFoundError:
    print('No database found. Creating new database.')

english_bot = ChatBot('Bot')
english_bot.set_trainer(ListTrainer)

```

```

for file in os.listdir('data'):
    print('Training using ' + file)
    convData = open('data/' + file, 'r').readlines()
    english_bot.train(convData)
print("Training completed for " + file)

```

<style type="text/css">

```

.fixed-panel {
    min-height: 400px;
    max-height: 400px;
    background-color: #19313c;
    color: white;
    overflow: auto;
}

.media-list {
    overflow: auto;
    clear: both;
    display: table;
    overflow-wrap: break-word;
    word-wrap: break-word;
    word-break: normal;
    line-break: strict;
}

.panel {
    margin-bottom: 20px;
    background-color: #fff;
    border: 6px solid transparent;
    border-radius: 25px;
    -webkit-box-shadow: 0 1px 1px rgba(0,0,0,.05);
    box-shadow: 0 1px 1px rgba(0,0,0,.05);
}

.panel-info {
    border-color: #0c2735;

```

```

}
.panel-info>.panel-heading {
    color: white;
    background-color: #0c2735;
    border-color: #0c2735;
}
.panel-footer {
    padding: 10px 15px;
    background-color: #0c2735;
    border-top: 1px solid #0c2735;
    border-bottom-right-radius: 3px;
    border-bottom-left-radius: 3px
}
body {
    background: rgb(96,143,149); /* Old browsers */
    background:      -moz-linear-gradient(-45deg,      rgba(96,143,149,1)      0%,
    rgba(0,133,136,1) 9%, rgba(12,36,73,1) 52%, rgba(26,30,59,1) 100%); /* FF3.6-15 */
    background:      -webkit-linear-gradient(-45deg,      rgba(96,143,149,1)
    0%,rgba(0,133,136,1) 9%,rgba(12,36,73,1) 52%,rgba(26,30,59,1) 100%); /* Chrome10-
    25,Safari5.1-6 */
    background: linear-gradient(135deg, rgba(96,143,149,1) 0%,rgba(0,133,136,1)
    9%,rgba(12,36,73,1) 52%,rgba(26,30,59,1) 100%); /* W3C, IE10+, FF16+, Chrome26+,
    Opera12+, Safari7+ */
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#608f95',
    endColorstr='#1a1e3b',GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
}
</style>

```

B. SCREENSHOTS

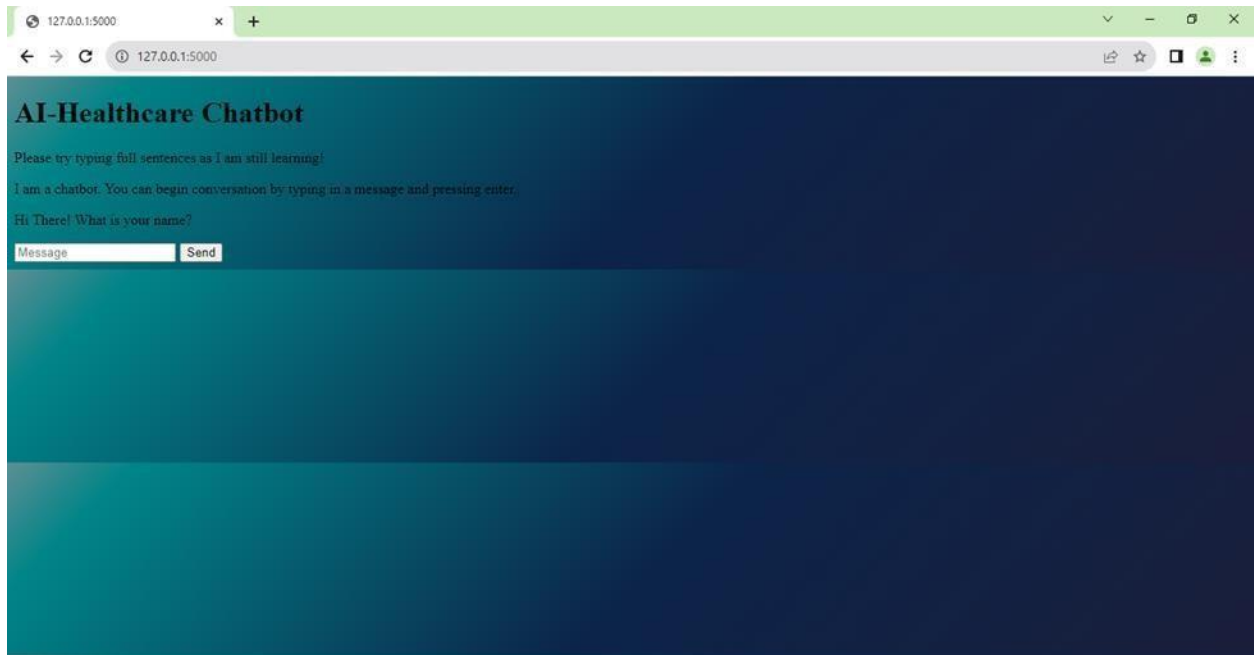


FIG : HOME PAGE

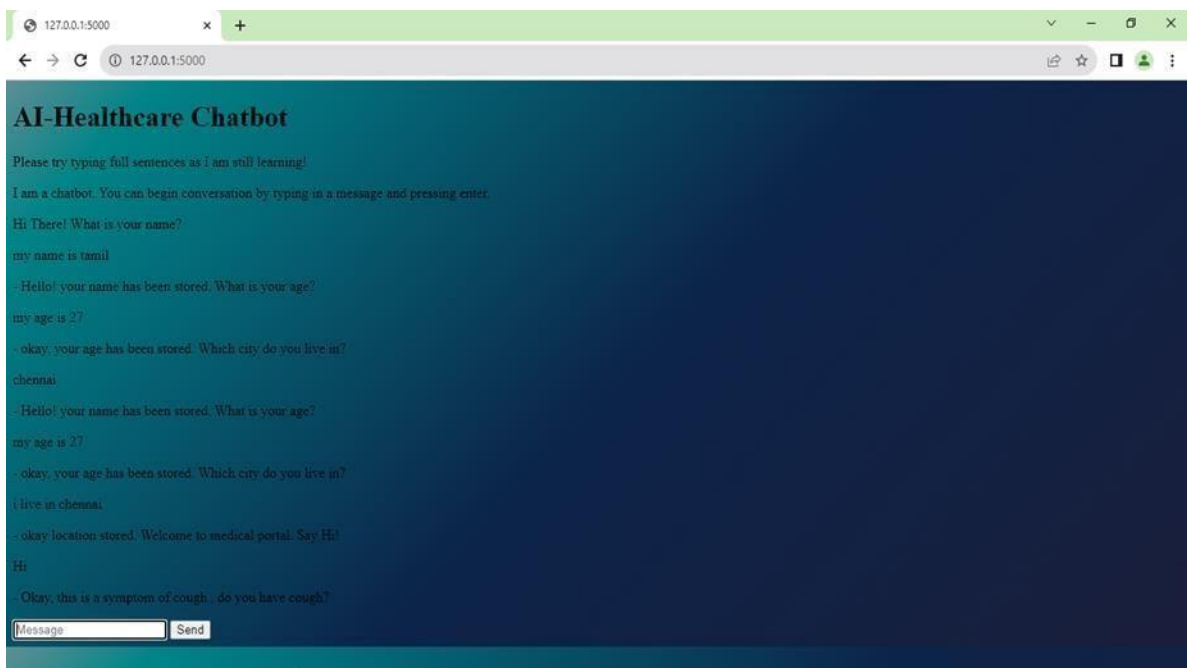


FIG : BASIC INFORMATION DETAIL PAGE

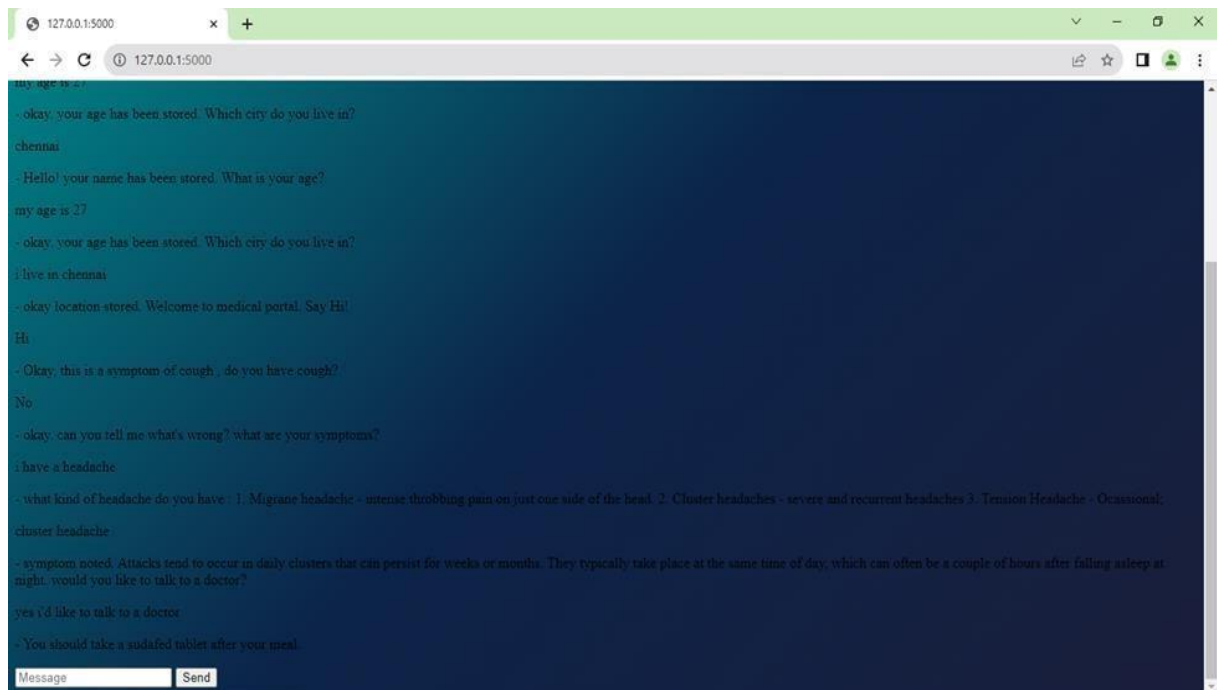


FIG : MEDICAL CHATBOT PAGE

ARTIFICIAL INTELLIGENCE-BASED CHATBOT FOR PATIENT HEALTH CARE

Balagundla Sudheerkumar¹, Koppuravuri Venkat², G Kavitha³

^{1,2}UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology,

Chennai, India

³Assistant professor, Dept. of CSE, Sathyabama Institute of Science and Technology,
Chennai, India

Email id: balagundlanswamy123@gmail.com

Email id: koppuravurivenkat197@gmail.com

Email id: kavithagovindasamy91@gmail.com

ABSTRACT- Medical care is vital to a healthful lifestyle. But it's far more difficult to get scientific help if you have health issues. The proposed idea is to expand a scientific Chatbot that could use ML to investigate the ailment and generate the essential facts to talk about the condition with the medical doctor. Medical discourses had been created to lessen scientific charges and improve get right of entry to clinical know-how. Some chatbots act as clinical publications, assisting sufferers study their disorders and enhance their health. Users can advantage of Chatbots if they can become aware of types of diseases and offer vital facts. Automated textual content-primarily based diagnostics allow sufferers to engage in the analysis of their medical issues and offer personalized evaluation with signs. Therefore, humans reflect on consideration on their very own proper and personal balance.

Keywords: AI, AI Chatbot, AI Chatbot structure, AI health Chatbot's, health Chatbot.

INTRODUCTION

Healthcare chatbots are having a massive impact on public health and our way of life. It increased faith and decreased human error. Today's people are addicted to the net however they don't care approximately their non-public protection. They keep away from hospitalization for minor troubles that could turn into severe illnesses in the future. The proposed

concept solves this trouble. The idea is to create a Chatbot that is free and available 24/7. The Chatbot is loose and to be had by the consumer anyplace there, whether they may be working in their surroundings, which encourages the consumer to keep and use it. This saves capital expenses for professional sessions. Machine mastering is at its height right now and is one example of the use of Chatbot to facilitate progress. It has to also be clarified that the reason for the Chatbot isn't always to diagnose sufferers. The concept became to create a medical chat with the use of system learning that might offer a few basic facts approximately the sickness before consulting a doctor. Text-based clinical chatbots help patients percentage their clinical worries and offer a customized diagnosis primarily based on symptoms. After this, the affected person maybe dispatched to a specialist. This collection of activities saves quite a few time for the instructor. Now, Chatbots are usually ready to help individuals who need scientific assistance. In addition, a virtual assistant may be chargeable for reminding customers of medicines and tracking the affected person's fitness popularity. By speaking fluently, the Chatbot interacts with real people. Chatbots are used in applications that include e-trade customer service, name facilities, and online games. However, with Chatbot, you can engage with textual content or voice interface and get answers through the system getting to know.

PROBLEM STATEMENT AND OBJECTIVE

This research aims to develop a safe and responsible healthcare chatbot that provides accurate health information, encourages early intervention for minor issues, and guides users toward seeking professional medical attention when necessary. This revision clarifies the problem and outlines the objective of your research. It highlights the limitations of current chatbots and proposes a solution that addresses patient safety and early intervention.

LITERATURE SURVEY

A comprehensive literature review is essential for developing a well-informed and effective healthcare chatbot. Here's why:

- **Understanding Existing Research:** It helps you identify existing research on healthcare chatbots, their functionalities, limitations, and potential risks.
- **Identifying Research Gaps:** By analyzing existing literature, you can pinpoint areas where current research falls short and identify opportunities for your chatbot to address those gaps.
- **Learning from Previous Work:** You can learn from the successes and failures of previous chatbot projects to design a more efficient and user-friendly system.
- **Developing a Strong Theoretical Foundation:** The review can help you establish a theoretical framework for your chatbot by exploring relevant concepts in healthcare communication, artificial intelligence, and human-computer interaction.

Focus on Healthcare Chatbots:

While the importance of literature reviews in software development is valid, tailor this section to your specific research. Focus on literature related to healthcare chatbots, their functionalities, effectiveness, and the challenges they address. By incorporating these changes, you can create a more focused and relevant literature review section for your research paper.

Clinical Chatbot Using Deep Learning

This research proposes a deep learning-powered clinical chatbot specifically designed to address the challenge of limited healthcare access in rural and underserved communities. In regions like India, a growing population strains healthcare resources, leading to delays in seeking treatment. This chatbot aims to bridge this gap by offering basic healthcare information through a user-friendly web interface. By analyzing user-reported symptoms, the chatbot, powered by deep learning, will provide relevant information about potential illnesses and encourage early intervention. However, it's crucial to emphasize that the chatbot is for informational purposes only and should not replace seeking professional medical attention when necessary. The Natural Language Toolkit (NLTK) will be a key tool in developing the chatbot's ability to understand and respond to user queries in natural language.

Contextual Care Chatbot (Through Deep Learning)

Yes, there's a revised paragraph on Contextual Care Chatbots using Deep Learning for your research paper: This research explores the potential of Contextual Care Chatbots powered by Deep Learning to revolutionize healthcare access. Chatbot technology has evolved from simple button-based interactions to sophisticated systems that leverage machine learning and artificial intelligence. Contextual Chatbots represent the next step, utilizing deep learning to retain and process user information, enabling personalized interactions. Beyond simply discussing functionalities, this research will delve into existing research on healthcare chatbots, identify future challenges and opportunities, and propose a deep learning-powered chatbot specifically designed to address the gap in healthcare access. This chatbot will integrate natural language processing with deep learning to offer users relevant health information, potentially leading to earlier intervention and improved health outcomes.

Medical Chatbot Using Natural Language Processing

While a healthy lifestyle is crucial, accessing medical professionals can be challenging. This research proposes a medical chatbot that leverages Natural Language Processing (NLP), a subfield of Artificial Intelligence (AI). However, it's important to clarify that the focus is

not on diagnosis, but on providing informative guidance. By analyzing user-reported symptoms through NLP techniques, the chatbot will offer relevant health information about potential illnesses and encourage early intervention when necessary. This can empower users to make informed decisions about seeking professional medical attention. Existing healthcare chatbots often serve as educational tools, but this research aims to go beyond by integrating NLP for a more comprehensive user experience.

EXISTING SYSTEM

Conversational agents, or chatbots, have been around for decades, with early examples like ELIZA utilizing basic text conversation to mimic a Rogerian therapist. These early chatbots often relied on keyword matching and generic responses to maintain a conversation. While these initial attempts laid the groundwork for chatbot development, their limitations in understanding complex language restrict their effectiveness in healthcare. Modern chatbots leverage advancements in Natural Language Processing (NLP) to provide more comprehensive and informative interactions. This research aims to build upon these advancements by developing a deep learning-powered chatbot that goes beyond basic information retrieval to offer users personalized guidance and encourage early intervention in healthcare.

Disadvantages of Existing System

While existing healthcare chatbots offer valuable functionalities, they have limitations that this research aims to address.

- **Limited Personalization:** Many AI-powered chatbots struggle to provide personalized recommendations tailored to individual health needs and medical history. This can hinder their ability to offer truly meaningful guidance.
- **Data Privacy and Security Concerns:** Patient health information is highly sensitive, raising concerns about data privacy and security when interacting with AI chatbots. This research will prioritize robust security measures and ensure compliance with relevant data protection regulations like HIPAA

(Health Insurance Portability and Accountability Act) in the US.

PROPOSED SYSTEM

This research proposes a deep learning-powered clinical chatbot to improve healthcare access, specifically targeting early intervention. Unlike existing chatbots that focus solely on information retrieval, this system prioritizes guiding users toward seeking professional medical attention when necessary. The chatbot will leverage deep learning to analyze user-reported symptoms and provide relevant information about potential illnesses, but it will not attempt diagnoses. Data privacy and security are paramount, and the system will adhere to relevant regulations to ensure user trust. This approach aims to bridge the gap in healthcare access, particularly in underserved communities, by empowering users to make informed decisions about their health advantages of the Proposed System

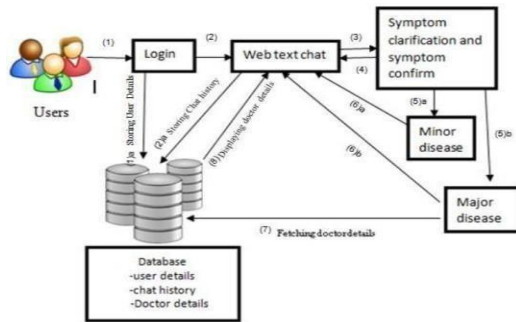
Advantages of the Proposed System

This deep learning-powered chatbot offers several advantages over existing healthcare chatbots:

- **Enhanced Accessibility:** The chatbot provides 24/7 availability, unlike human healthcare professionals who have limited working hours. This allows users to access health information and guidance anytime, regardless of location or time zone.
- **Cost-Effectiveness:** The chatbot can potentially reduce healthcare costs compared to traditional consultations. This makes it a potentially valuable tool for low-income families and medically underserved communities.
- **Scalability:** The chatbot can serve a large number of users simultaneously, addressing a critical challenge in areas with limited medical resources. It can efficiently handle multiple inquiries without compromising response time or quality.

By combining these advantages, the proposed chatbot has the potential to significantly improve healthcare access, particularly for those facing geographical or financial barriers.

SYSTEM ARCHITECTURE



SYSTEM REQUIREMENTS

Hardware Requirements

- System: Pentium Dual Core.
- Hard Drive: a hundred and twenty GB.
- Monitor: 15-inch LED
- Input gadgets: keyboard, mouse.
- RAM: 4 GB

Software Requirements

- Operating system: Windows 7/10.
- Programming Language: Python.

MODULES

- User Verification and Trait Extraction
- Features are extracted with a skilled dataset
- Explanation and commands of illnesses by a health practitioner.

User Verification and Trait Extraction

This module ensures secure user access through a verification process. Once a user logs in, the system extracts relevant health information from their descriptions of symptoms. A Natural Language Processing (NLP) technique, such as a rule-based or machine-learning approach, will be employed to identify key phrases and concepts related to the user's health concerns. This extracted information will then be used to guide the chatbot's responses and provide relevant healthcare information.

Improvements:

- Emphasizes NLP for symptom extraction, aligning with your deep learning focus.

- Removes potentially inaccurate references to string search algorithms.
- Focuses on the core functionality of extracting relevant health information.

Feature Extraction from a Trained Dataset

This module leverages a curated medical dataset to extract relevant features that can be used to identify potential health concerns. The approach avoids directly asking users yes or no questions about specific diseases. Instead, it utilizes Natural Language Processing (NLP) techniques to analyze user-reported symptoms. By comparing these symptoms to the trained dataset, the chatbot can identify patterns and generate a list of potentially related illnesses and their associated symptoms. This data-driven approach offers a more informative and user-friendly experience compared to a question-based system.

Explanation of Diseases and Commands from the Medical Doctor

The process includes listing diseases in a database and comparing every symptom to common sickness characteristics. Next, the characteristic is checked until a suit is observed. They are listed in line with the contribution of the give-up customers to the question rank. The chatbot identifies the precise sickness and communicates it to the stop person. The chatbot checks whether or not the identified disease is a chief or minor trouble based totally on the recommendations built into the chatbot. If the problem is critical, the chatbot sends the health practitioner's info and directs the end person to an expert. If the problem is minor, the chatbot identifies the disease, directs the end consumer approximately first aid or answers, and shows seeing a physician as quickly as feasible.

GOALS:

The fundamental objectives of UML design are:

1. Provide users with a geared up-to-use, obvious visual layout language to create and proportion significant fashions.
2. Provide extra practice and special coaching to beautify key principles.

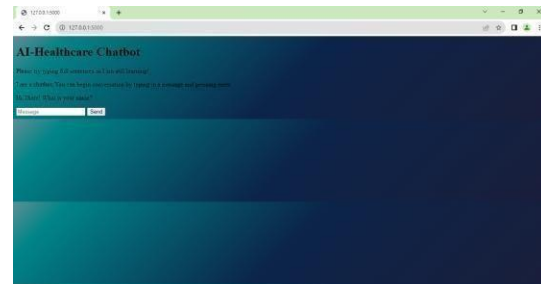
3. be impartial to specific programming languages and development approaches.
4. Provide a formal basis for know-how language formation.
5. To promote the development of market-oriented merchandise.
6. Supports excessive-level development standards including collaboration, composition, modeling, and additives.
7. Complete with the best competencies.

RESULTS AND DISCUSSION

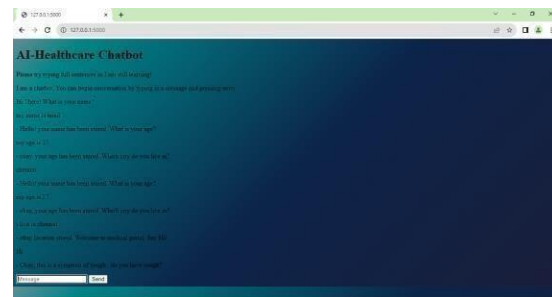
By the use of synthetic intelligence remedy chatbots, the consumer can be able to do voice communication and get particular desired outcomes within the shape of voice. The person can solve special and widespread questions saved within the database. Chatbot's overall performance relies upon the Raspberry Pi and database. The foremost reason for the fitness care chatbot machine is to offer offerings to humans in rural and authority hospitals who are unable to e-book an appointment with a doctor or acquire medical facts. They can solve their problem with the use of a chatbot. The synthetic intelligence-based chatbot gadget gives clinical tips on drugs or sicknesses and is to be had 24*7 hours. With the speech signal input through the microphone, it's miles compared to the database saved at the Raspberry Pi and remarks are obtained from the database. If the description of the query or the facts associated with the disorder and remedy is not available in the database, the answer will be dispatched from the Google servers. The input speech sign is first transformed from voice to textual content, then from text to speech, and the output signal is taken from the speaker.

SCREENSHOTS

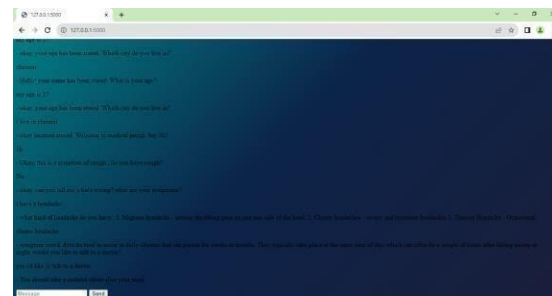
Home Page



Basic Information Detail Page



Medical Chatbot Page



CONCLUSION

The objective of this observation is to discover the definition and architecture of AI chatbots and the importance thing technology for implementing AI chatbots. In addition, this study additionally examines the kingdom of AI-based chatbot service improvement. Based on this studies study, AI-primarily based scientific schools have proven to be the only device in supplying healthcare offerings. This is proven by way of many examples of AI-powered healthcare chatbots, whose improvement has been extended through the face-to-face lifestyle that has end up the new norm because of the speedy unfolding of COVID-19. Additionally, effective medical faculty AI

has been determined to be exceedingly effective in improving healthcare offerings and enhancing the affected person's enjoyment. On the other hand, those scientific colleges that are made via synthetic intelligence have been discovered to have some obvious boundaries. Efforts must be made to triumph over these limitations and to supply better health services, in a spread of ways including increasing their best of statistics collection, the use of strategies evolved through synthetic intelligence, increasing the quantity of qualified health professionals, cooperating and many others. Implement security features.

REFERENCES

- [1] Vaibhav Tode, Himanshu Gadge, Prateek Kachare and Sudarshan Madane, CureBot -An Artificially Intelligent Interactive Bot for Medical Diagnostics International Research Journal of Engineering and Technology (IRJET)., Vol.7, no.12 (Dec 2020).
- [2] Satyendra Praneel Reddy Karri and Dr Santosh Kumar, Deep Learning Techniques for Implementation of Chatbots, 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, INDIA, (2020) January 22-24.
- [3] Dipesh Kadariya, Revathy Venkataramanan, Hong Yung Yip, Maninder Kalra, Krishnaprasad Thirunarayanan and Amit Sheth, kBot: Knowledge-Enabled personalized Chatbot for Asthma Self-Management, 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA,(2019) June 12-15.
- [4] R. Dharwadkar, N.A. Deshpande, A medical chatbot. Int. J. Comput. Trends Technol. (IJCTT). **60**(1) (2018).
- [5] S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari, S.K. Devi, A self-diagnosis medical chatbot using artificial intelligence. J. Web Dev. Web Designing **3**(1) (2018).
- [6] N. Jyothirmayi, A. Soniya, Y. Grace, C. Reddy Kumar Kishor, B.V. Murthy Ramana, Survey on chatbot conversational system. J. Appl. Sci. Comput. **6**(1) (2019).
- [7] Oh, Y. J., et al. "A systematic review of artificial intelligence chatbots for promoting physical activity, healthy diet, and weight loss." International Journal of Behavioral Nutrition and Physical Activity **18** (2021): 1-25.
- [8] Global Chatbot Market Size, <https://markets.businessinsider.com/news/stocks/global-chatbot-market-anticipated-to-reach-9-4-billion-by-2024-robust-opportunities-to-arise-in-retail-e-commerce-1028759508>
- [9] Laranjo L, Dunn AG, Tong HL, Kocaballi AB, Chen J, Bashir R, Surian D, Gallego B, Magrabi F, Lau AYS, Coiera E. Conversational agents in healthcare: a systematic review. J Am Med Inform Assoc. 2018 Sep 1;25(9):1248-1258. doi: 10.1093/jamia/ocy072. PMID: 30010941; PMCID: PMC6118869.
- [10] S. Ayanouz, B. Abdelhakim, and M. Benhmed, "Asmart chatbot architecture based NLP and machine learning for health care assistance", Association for Computing Machinery Proceedings, pp. 1-6, 2020.
- [11] Seo Kyori, Analysis of Domestic and Global Trends and Development Prospects of Artificial Intelligence-based Chatbot Service, National Information Society Agency (NIA), D.gov Trend & Future, 2018